NEURAL NETWORKS FOR LEGAL QUANTUM PREDICTION

by

Andrew J. Terrett

B.A., University of Warwick, England, 1990

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF LAWS

in

THE FACULTY OF GRADUATE STUDIES

(Faculty of Law)

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

October 1994

FACULTY ~~Department~~ of LAW.

The University of British Columbia
Vancouver, Canada

Date NOV, 30th 1994.

ABSTRACT

This thesis argues that Artificial Neural Networks (ANN's) have applications within the domain of law and can be built using readily available Artificial Intelligence software for the Personal Computer. In order to demonstrate this, I have built working ANN's using data made available to me by the Faculty of Law Artificial Intelligence Research Project (FLAIR) and Windows™-based ANN software that is commercially available. The reasons for building a system are three-fold. First, a working system is the most graphic demonstration of my above assertion. Secondly, given my background as a practitioner in law, I am concerned to ensure the quick, efficient movement of law-related technology from research laboratory to marketplace. Thirdly, the building of a neural network avails me with the opportunity to analyze comparatively the performance of the ANN's with statistical and expert system models which used the same data and were also built at the FLAIR project.

The theoretical foundation for this thesis is the view that, although many legal decisions are often reducible to a set of doctrines, policies or sub-doctrinal rules, certain domains of legal decision-making evade analysis using a rule-based paradigm. Thus, although relational patterns between any given facts and the law exist, they cannot always be described. Therefore in order to build "intelligent" computer programs that can assist the lawyer in his or her work, we should explore the potential of and utilize those tools that can find relational patterns automatically. Having done this, we should attempt to combine the same with those software tools that we understand more fully, namely expert systems and traditional programming methods. The use of hybrid neural network/expert system programs is well developed in many other domains. Legal researchers, however, have yet to even thoroughly examine neural networks as an isolated technology. This thesis is an attempt to right this imbalance.

TABLE OF CONTENTS

Chapter 2 Issues of Jurisprudence

LIST OF FIGURES

Chapter 1

Chapter 4

A NOTE ABOUT THE SOFTWARE

I have used Brainmaker Neural Network Simulation Software (Versions 3.0 and .3.1 for Windows™) developed by California Scientific Software of Nevada City, CA., Borland dBASE™ Version III Plus as the database, Microsoft Excel Version IV and CorelDRAW Version 2.01 for graphing throughout this thesis. I cannot, in all honesty, endorse either Brainmaker or dBASE™. In my opinion, although dBASE™ is an reasonable product as a database, for my purposes, its programming language was "clunky", very "user-unfriendly" and lacked the mathematical functionality necessary to build neural networks. Brainmaker although economically priced, also lacked certain functionality of other similar neural network software. Version 3.0 contained bugs which caused unnecessary delay to the progress of my research. However, the book that accompanies the software makes an excellent introduction to neural network technology (and is almost worth the money for this feature alone.) The actual software manual, however, is bewildering (at least initially) to someone who not skilled in the art of developing neural networks. To someone contemplating a similar project I would offer this advice; explore the neural network software available over the Internet. The most important factor in a neural network is the algorithm. Everything else, such as graphs, user-friendliness, etc. is of secondary significance (once the theory has been mastered.) Shop around, don't use the first thing you think looks appropriate and I wish you the very best of luck.

ACKNOWLEDGMENT

Thanks are due to the following people:

1. To the FLAIR programmers, (Bruce Atherton, Keith MacCrimmon and John McClean,) for patiently answering an endless stream of questions, (more often, elementary; less often, interesting) regarding neural networks, the dBASE™ programming language and whiplash injuries in British Columbia.

2. To Professors J.C.Smith and Marilyn MacCrimmon for their time and input.

3. To my parents for their support, financial and otherwise in this venture.

4. To Kelly, who while providing constant distractions in her quests for caffeine also showed immense patience in attempting to understand this obscure technology and in listening to the daily ups and downs in writing this thesis (of which there were many.)

INTRODUCTION

This thesis is about jurimetrics[1] or the scientific investigation of legal problems. In particular, it addresses the potential application of Connectionism or Neural Computing (also referred to as "brain- style" computing to a specific type of legal problem.) The task set is to consider what role, if any, neural computing might play in a program that can mimic the legal reasoning processes of a lawyer and in particular to what extent neural computing can mimic a legal actor, whether adjuster, lawyer or judge, in determining the likely level of damages or quantum for personal injury cases involving whiplash in British Columbia, Canada. This domain of legal knowledge was chosen for both practical and theoretical reasons; practically speaking, the main reason was that FLAIR[2] has developed and maintains a database of approximately 1700 cases involving whiplash injuries (the "Whiplash Database."). Thus there was a readily available source of (largely) numerical data with which to build a system. Theoretically speaking, the assessment of quantum is a legal reasoning process that occurs throughout the development of a case; a lawyer constantly assesses what an injury might be worth in court and also assesses how a given piece of evidence affects previous assumptions and estimates. Attempts have also been made by members of FLAIR to develop an expert system and a conventional computer program (in the C programming language,) to predict quantum using the same data. Thus

[1] A term coined by Lee Loevinger in his article *Jurimetrics - The Next Step Forward*, Minnesota Law Review, 33, 1948, 455-493 at 483.
[2] Faculty of Law Artificial Intelligence Research Project, University of British Columbia.

useful comparisons could be made between the various models to see which was the more accurate and why.

The impetus for this thesis comes from the fact that despite much rhetoric to the contrary, expert systems have failed to make the impact on the legal profession that many researchers believed possible. Richard Susskind, who is arguably the best qualified researcher on this area, commented in 1986 "there has yet been developed a fully operational expert system in law that is of utility to the legal profession.[3]" At the time of writing, expert systems have made little further impact on the profession.

It is arguable and indeed argued by many[4], that the revived interest in Connectionism is an indication of a new emergent paradigm within (or parallel to) traditional Artificial Intelligence, an example of scientific revolution made known to us by Kuhn[5]. It is also arguable that when historians begin to compose the history of AI from the 1960's to the 1990's many will apply a subtitle entitled "The delusion of the successful first step in research.[6]" Supporters of Connectionism come in many shapes and forms. Equally robust responses have followed from Connectionist skeptics, notable from Fodor and

[3] Susskind, R.E., *Expert Systems in Law: A Jurisprudential Approach to Artificial Intelligence and Legal Reasoning,* Modern Law Review, 49, 1987, p.168.

[4] See Schneider, W., *Connectionism: Is it a paradigm shift for psychology?* Behavior Research Methods, Instruments and Computers, 19, 73-83; See also Fodor, J.A., *Why there STILL has to be a language of thought*; Smolensky, P., *Connectionism and the Foundations of AI*, Wilks, Y., *Some comments on Smolensky and Fodor*, Churchland,P.M., *Representation and high-speed computation in neural networks*, all in Partridge, D, & Wilks, Y., The Foundations of artificial intelligence - A sourcebook, Cambridge, Cambridge University Press, 1990.

[5] Kuhn, T.S., *The Structure of Scientific Revolutions*, Chicago, University of Chicago Press, 1970, 2nd edition, taken from the International Encyclopedia of Unified Sciences, Vol. 2, Number 2.

[6] Some authors have done just that. For instance see Graubard, S., (ed.) *The Artificial Intelligence Debate: False starts, Real foundations*, New York, Basic Books, 1988.

Pylyshyn[7]. This thesis does not attempt to develop further the theoretical debates. They are mentioned only in order to facilitate an broader understanding of the issues surrounding this area. The emphasis of this thesis is the practical; the time is right to assess the value of pure Connectionism using real-world data in the legal domain.

The first task of this thesis is to adequately describe the development, ancestry, functioning of and philosophical basis for neural network architectures and techniques. This is provided in Chapter 1. Chapter 2 details the various jurisprudential theories that may assist our analysis. In order to build valid models of law and legal operations using expert systems, one must have regard for legal theory. With Connectionism, one does not have to subscribe to a particular view of how the law functions, except to the extent that one must view the legal reasoning process as example-driven.

Given that neural computing is a relatively new phenomena, having re-arisen phoenix-like from the ashes of computational theory in the mid 1980's, there has been a surprisingly significant amount of material written on the application of neural networks in law and this will be reviewed and explored in Chapter 3. Finally, Chapter 4 describes the empirical basis of this thesis, describing the experiments run, the domain of whiplash injury quantum analysis, the software and the results obtained.

[7] Fodor, J., and Pylyshyn, Z.W., *Connectionism and cognitive architecture: A critical analysis*, Cognition, 28, 3-71, 1988.

CHAPTER 1

AN INTRODUCTION TO NEURAL NETWORKS

1.1 FUNDAMENTAL DEFINITIONS

Given that this thesis is primarily about artificial neural networks[8], it seems unnecessary to provide a comprehensive history of the Artificial Intelligence discipline, in which neural networks (at least to date) play only a minor supporting role. The past of this divided discipline is adequately catalogued elsewhere[9]. Many commentators describe AI research as divided into two competing paradigms - the symbolic and the sub-symbolic (or connectionist.) If one were to describe the history of AI to date in the form of a short fictional story, one could be forgiven for casting Connectionism (and indeed, previous emerging technologies) as a younger child within the AI family who brought great excitement at its birth, but was spurned at an early age, cast out into the wilderness during its adolescence only to returning hero-like, when the older and seemingly more successful and mature sibling failed to live up to expectations.

Whilst steering away from such melodramatic parodies, we cannot ignore the fact that the so-called symbolicist approach has gradually re-trenched, from bold schemes such as the "General Problem Solver"[10] and bold assertions[11] towards more humble goals[12]. The

[8] This thesis refers to artificial neural networks, ANN's and neural networks. These terms are used interchangeably throughout.

[9] See for instance, Dreyfus, H.L., - *What Computers Can't Do*, New York, Harper & Row, 1972, also Dreyfus, H.L., *What Computers Still Can't Do*, Cambridge, MA., MIT Press, 1992.

[10] The "General Problem Solver" was the impressive title given to a landmark computer project led by Newell and Simon, the culmination of their Cognitive Simulation (CS) approach. By 1957, computers,

irony of symbolic AI has been that the so-called "difficult" problems (i.e. those involving a high degree of specialized knowledge) have proved easy and visa versa. For example, two of the most famous expert systems are MYCIN and DENDRAL. MYCIN is an expert system built in 1972[13] which was used to diagnose and prescribe treatments for bacterial infections of the blood. DENDRAL, built at Stanford University in 1964[14] was used to determine the structure of organic molecules. We might conclude from these examples that expert systems are suited to the higher cognitive functions that humans possess and therefore possess high levels of intelligence. However, expert systems have failed to cope adequately with human functions that require native intelligence or what we call "common sense." This includes abilities such as pattern recognition and the instantaneous assessment of appropriate courses of action. Good examples of this type of skill include the ability to assess the speed of passing cars when crossing a road. There are no rules applied - we just know in the circumstances what would be the right thing to do - to walk or not to walk. Moreover, this type of ability reveals itself through an expert's articulation

acting as general symbol manipulators were seemingly providing the physical evidence of intelligence. Using the GPS, these researchers discovered a number of original proofs for theorems in *Principia Mathematica*. Such evidence seemed to suggest the correctness of the dominant reductionist worldview, i.e. the widely held belief that intelligence could be understood in terms of symbol manipulation

[11] "Within ten years a digital computer will be the world chess champion"; "within ten years a computer will discover and prove an important new mathematical theorem"; "within ten years, most theories in psychology will take the form of computer programs or of qualitative statements about the characteristics of computer programs." Simon, H.A., & Newell, A., *Heuristic problem Solving: The Next Advance in Operations Research*, Operations Research, Vol. 6, 1958.

[12] For example, the words of Susskind, a leading advocate of expert systems in law; "Legal expert systems are computer programs that have been developed with the aid of human legal experts in particular and highly specialized domains of law...It must be emphasized that such an outline today amounts to no more than the research aspirations of workers in the field of expert systems in law..." Susskind, R.E., *Expert Systems in Law: Theory and Practice*, Cambridge Lectures, 1987.

[13] See Harmon, P., and King, D., *Expert Systems, Artificial Intelligence in Business*, New York, John Wiley and Sons, 1985, p.5.

[14] See also Waterman, D.A., *A Guide to Expert Systems*, Menlo Park, CA., Addison-Wesley Publishing Co., 1986.

of an area of expertise as well as in the mundane. Experts often find it easier to describe a domain by reference to examples rather than by reference to hard rules.

It would be wise to for us to avoid being drawn into the lengthy narrative that is the history of AI, or being drawn into equally lengthy and inconclusive debates regarding whether there are two competing paradigms within the discipline. The reality seems to be that Connectionism adopts certain aspects of the symbolic approach and visa versa. For instance, in the domain of law, one might construct the argument that formalizing an area of law into rules involves the recognition of patterns and that Connectionism is the repetitive application of a rule, albeit in algorithmic form. Obsessive theorizing in AI tends to obscure purpose. However, Connectionism may also be seen as an element of complexity theory (considered further below,) a scientific hypothesis relating to non-linear phenomena. Debates regarding competing paradigms are likely to continue for some time. It is more important in a thesis relating neural networks to the domain of law merely to mark the major developments in the movement known as Connectionism so that the reader is given a sense of historical perspective. Therefore, the second part of this chapter chronicles the development of neural networks from the 1940's to the present day and also describes some of the applications for which neural networks are being used.

The primary task of this chapter is to define the fundamental concepts and methodologies in building neural network software. Such analysis will be assisted by briefly describing the two computing techniques of deduction and induction to supplement our description of the two apparent paradigms within AI. Therefore, the first part of this chapter is designed to describe the context of terms within which Connectionism is situated.

1.2 DEDUCTION -v- INDUCTION

The power of deductive thinking has been understood and emphasized as a model of rationality by metaphysical thinkers since Parmenides and Plato[15]. Deduction is the process by which a person or computer reasons in steps towards a conclusion based upon given premises. Deduction requires the sequential provision of answers. Thus when accountants fill out one's tax returns, they reason deductively. For instance, if one were asked for one's marital status on a form relating to one's personal income, the answer - "single" is likely to lead to a different and separate set of calculations from the answer "married." Many statutes in law are structured in a manner that encourages deductive thinking. Hence, one can see the immediate appeal of the legal domain to computer programmers interested in modeling real world phenomena and the need for the skills of a logician in legal analysis. The history of Artificial Intelligence is dominated by attempts to use largely deductive and symbolic processing techniques to model and solve problems of the real world.

Induction is the process by which a computer or a person processes data at once and draws a conclusion. The recognition of patterns, symbols and numbers are all examples of induction - a process that appears to occurs instantaneously. Pattern recognition

[15] For a good introduction to the differences between metaphysical and scientific thinking, see Sayegh, N.S., *Philosophy and Philosophers*, Ottawa, Academy Books, 1988.

techniques in particular have eluded AI researchers for decades - "Pattern-matching has been the Achilles heel of computer scientists for several decades," writes David Hertz[16]. Neural networks, such as those built in conjunction with this thesis, use induction to draw their conclusions and in certain circumstances[17], appear to excel at such tasks.

1.3 CONVENTIONAL COMPUTING TECHNIQUES

Conventional computers use symbolic programming; programs work step-by-step through a problem. The right data must be found at the right time or the program will cease to function, a so-called "catastrophic degradation." A computer program written in a linear form is not able to cease the operation in progress and re-compute at short notice. It must run its commands to their conclusion. Whilst highly accomplished at performing mind-boggling numerical calculations or re-arranging immense amounts of data quickly and without error, a conventional computer program cannot easily deal with the "fuzzy" data that surrounds us in the real world, governed and defined by the vague boundaries of natural language, which do not easily fit into precise mathematical sets.

1.4 ARTIFICIAL INTELLIGENCE (AI)

AI is "the science of making machines do things that would require intelligence if done by man.[18]" Such a simply stated goal has proved to be very costly in terms of time and

[16] Hertz, D., *Thinking Computers*, The Miami Herald, 5 February, 1989.
[17] See Section 1-20.
[18] Minsky, M., *Semantic Information Processing*, Cambridge, MA., MIT Press , 1968.

money[19]. Researchers seek to model human expertise and, in providing the user with computer-generated conclusions, offer useful explanations. AI evolved as a concept during the 1950's when it was considered that computers would be able to perform functions to emulate the workings of the human brain in an "intelligent" manner. The word intelligence is derived from the Latin "legere" meaning to collect, gather or assemble. The word intelligence has come to mean to understand, to know or perceive[20], hence a computer program that encompasses these functions could be described as artificially intelligent. However, would such a machine actually be intelligent? Would it have any conscious appreciation of what it is doing?

Instead of developing and analyzing the somewhat sterile debate of whether or not Connectionism represents a Kuhnian-style scientific revolution, we will accept at face value that AI consists of two significant paradigms - the symbolic and sub-symbolic. The symbolic view states that intelligence is derived from the explicit manipulation of symbols, usually through a sequential ordering. The sub-symbolic states that intelligence is "an emergent property of the inter-connection of a very large number of very simple processing elements, all operating in parallel and communicating only sub-symbolic information.[21]"

[19] Bart Kosko, a strong advocate of fuzzy thinking states "Something like $100 billion worldwide has gone into AI... and you can't point to a single AI product in the home, office, automobile or anywhere" (quoted in the forthcoming book "Fuzzy Logic" by Daniel McNeill and Paul Freiberger.) This may be overstating the position somewhat but it is undeniable that AI is perceived by the business community as a very expensive flop.

[20] Its full definition is "The ability to perceive logical relationships and use one's knowledge to solve problems and respond appropriately to novel situations." The New Lexicon Webster's Dictionary of the English Language, Lexicon Publications, Inc., New York, 1988.

[21] Rumelhart, D.E., & McClelland, J.L., *Parallel Distributed Processing: Explorations in the Micro-structures of Cognition*, Cambridge, MA., 1986, Vol. 1, p.4.

What is the difference between these paradigms in practice? Smolensky writes;

"Nowhere is the contrast between symbolic and subsymbolic approaches to cognition more dramatic than in learning. Learning a new concept in the symbolic approach entails creating something like a new schema. Because schemata are such large and complex knowledge structures, developing automatic procedures for generating them in original and flexible ways is extremely difficult. In the subsymbolic account by contrast, a new schema comes into being gradually, as the strength of the atoms slowly shift in response to environmental observation, and new groups of coherent atoms slowly gain important influence in the processing. During learning, there never need be any decision that "now is the time to create and store a new schema.[22] ""

The connectionists are those AI researchers who believe that in order to build truly intelligent machines, researchers must draw, at least in part, upon the complex adaptive model that we know works extremely well, namely the brain. However, strict adherence to any particular model of the brain is to be avoided. Pagels[23] makes a useful analogy with the legend of Daedalus and Icarus. A compulsive copying of how birds fly brought disaster, although the adaptation of certain elements may bring success. For example, modern planes, like birds, have wings and tails. Thus, say the connectionists, we should be informed by, but not chained to the biological analogy. Indeed, the software used in conjunction with this thesis bears little relation to the functioning of the brain as it is presently understood. The computationalists are those AI researchers who believe that the analogy of the brain as a model for building intelligent machines is an unnecessary diversion.

[22] Smolensky, P., in Rumelhart and McClelland, ibid., p.261.

[23] Pagels, H., *Dreams of Reason*, New York, Simon & Schuster, 1988, p.114.

1.5 EXPERT SYSTEMS

There is no accepted definition of an expert system. For convenience, we will adopt the generalized and accepted view that an expert system is a computer program that provides the user with the expertise of an expert through that program[24]. Although expert systems are not the main focus of this thesis, they are "predecessors-in-title" to neural networks within the AI discipline. This thesis attempts to make useful comparisons between the two types of software with reference to the domain of law. Thus some explanation of expert systems will assist our discussion.

Expert systems seek to capture both the raw data/information on any given domain and combine this with the rule-based procedures for deductive reasoning that are used by those actors skilled in, and knowledgeable of an area of expertise, i.e. those who might be loosely defined as "experts." Having combined data with the replicated reasoning procedures gleaned from the expert, the ideal expert system provides advice to the user as would an expert. At the same time, it should exhibit its internal reasoning process - the user can see the "what" and the "why" concurrently. This system transparency is a vital attribute and in the discipline of law, is of particular importance; few, if any, legal decisions are made without reference to the legal principles that underlie the decision.

[24] There are more long winded definitions such as that adopted by the British Computer Society Special Interest Group on Expert Systems:

> " An expert system is regarded as the embodiment within a computer of a knowledge-based component from an expert skill in such a form that the system can offer intelligent advice or make an intelligent decision about a processing function...The style adopted to attain these characteristics is rule-based programming."

An expert system consists of a knowledge base, an inference engine and a user interface. It may also be linked to a database. The knowledge contained in the knowledge base is captured in the form of *if-then* rules. Knowledge may be in the form of either facts or heuristics[25], (or a combination thereof.) The user interface acts as the intermediary between the user and the knowledge-base/inference engine in order to elicit answers to a sequence of pre-determined questions. The inference engine reasons logically using the rules in the knowledge-base and the facts discerned. Unlike conventional programs, the path taken by the program (and consequently the advice provided) is entirely dependent upon the facts provided by the user.

In the mid to late 1980's, there was much talk that this new domain of applied AI would give rise to a new profession, that of the "knowledge engineer." This ill-suited term was used to describe a person who obtains the knowledge from the expert and converts it into a form that the system can use, including inputting it into the system. In reality, the knowledge engineering profession has been notable only by its absence (as have the number of working expert systems, both in legal and non-legal fields.) Why is this? The creation of any expert system will involve finding solutions to three inter-related problems; knowledge acquisition, representation and utilization. The problems with each of these steps with reference to the domain of law will be analyzed in Chapter 2. At this point it

[25] Heuristics are rules of thumb that are perceived by experts in the domain to contribute to the overall solution of a given problem. Such solutions are based purely on experience. One often quoted example of a heuristic is in the game of chess; as a rule, one should try to control the centre of the board in the game. However, this does not in itself guarantee victory.

will suffice to acknowledge that each of the three steps were and remain more complex than they might appear at first.

1.6 FUZZY LOGIC

At what point does a person cease to be young and start to be old? The graphical representation of "young" will be a curve gradually decreasing over time and "old" will be represented graphically as a curve gradually increasing over time. Fuzzy logic is a recognition that everyday language is filled with terms that refuse to be bounded by classical mathematical sets and that traditional binary logic cannot fully incorporate "real-world" factors. This enlightened approach originally developed by Lotfi Zadeh, applying mathematically what had been recognized some years earlier by the eminent philosopher and linguist C.S.Peirce[26], attempts to overcome the black and white qualities of traditional logic approaches in representing knowledge. It also attempts to address the issue of commonsense knowledge. For instance, classical logic is defeated by the Cretan riddle, in which a Cretan asserts that all Cretans are liars. Both logical conclusions derived from this statement lead to contradiction. Classical logic cannot deal with such a paradox because the statement is both true and false. Fuzzy logic will say that the answer is 50% true and 50% false - i.e. on a scale of 0 to 1, truth value equals 0.5 and the falsity value

[26] Pierce, C.S.(1839-1914.) There is a developing body of literature relating to Pierce and neural networks; for instance see the chapter on Semiotics and Connectionism in Aparicio, M., & Levine, D., *Neural Networks for Knowledge Representation*, Hillsdale, NJ., Lawrence Erlbaum Associates, 1993. There is also an interesting body of literature that relates Pierce's work on Semiotics with the Critical Legal School; see for instance Kevelson, R., (ed.*) Pierce and the Law: Issues in Pragmatism, Legal Realism and Semiotics*, New York , Peter Lang, 1991. The relationships within this triad of theory makes an interesting theoretical under-pinning for a Connectionist theory of law, which is briefly addressed in Chapter 2. A fuller exposition is beyond the scope of this thesis.

also equals 0.5. Similarly, one can fuzzify the age of a person from 26 to "old with a fuzzy membership of 0.4" and "young with a membership of 0.4" on scales of 0 to 1.

The comp.ai.fuzzy Newsgroup on the Internet defines fuzzy logic as follows: "Fuzzy logic is an superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth - truth values between "completely true" and "completely false.[27]" Zadeh argues that rather than regarding fuzzy theory as a single theory we should regard the process of "fuzzification" as a methodology to generalize any specific theory from a crisp (discrete) to a continuous (fuzzy) form. This is known as the "extension principle.[28]" Thus the fuzzy representation of the number 7 might be as follows, (although this is only one of numerous possible representations;)

NUMBER	FUZZY VALUE
5	0
6	0.5
7	1
8	0.5
9	0

1.7 FUZZINESS AND LEGAL LANGUAGE

The language of the law is filled not only with binary (true-false) and converse predicates (grandparent-grandchild), but also gradable and contradictory predicates (hot-cold and

[27] Kantrowitz, M., Horstkolte, E., and Joslyn, C., *Answers to Frequently Asked Questions about Fuzzy Logic and Fuzzy Expert Systems*, comp.ai.fuzzy, July ,1994, ftp.cs.cmu.edu, usr/ai/pubs/faqs/fuzzy/fuzzy.faq.
[28] See Zadeh, L., *Fuzzy Sets* in Information and Control, 8, 1965, pp.338-353.

alive-dead (i.e. impossible for both to be true.)) Reed writes of this problem, "the main type of concept that a purely rule-based system cannot adequately represent is the "judgmental question." Judgmental questions are those where the so-called rule cannot work on binary data ("yes" or "no", "present" or "absent") but requires an assessment of one party's behaviour in relation to the norm. Questions such as "Did the Defendant take reasonable care?" are not really yes/no questions.[29]" Zadeh writes of fuzzy logic "it is widely agreed that an important problem in AI is the representation of commonsense knowledge. In general this knowledge may be regarded as a collection of such propositions as "snow is white," "icy roads are slippery," "most Frenchmen are not very tall," "Virginia is very intelligent"...[30]" The reality of the real world is that people use fuzzy language - we say " it is cold out today." This may be used as a relative or an absolute term i.e. it may mean that the temperature ranges from perhaps, minus to plus ten degrees Celsius. It may also depend on the context - a temperature of fifteen degrees Celsius on a July day where average temperatures are 23-28 degrees Celsius might also merit the above comment. Disorderliness of everyday language does not necessarily lead to incomprehension because language inevitably carries with it nuances and connotations created by past use and present context. Yet as Goodrich writes "The professional language and legal language in particular, however, strives to construct a unitary language which stands above the conflicting usage and differently oriented accents of social dialogue.[31]" If one describes the time of the day as dusk instead of night in casual

[29] Reed, C., *Computer Law*, London, Blackstone, 1990.

[30] Zadeh, L., *Commonsense Knowledge Representation Based on Fuzzy Logic*, Computer, Vol.16, No.10, 1983.

[31] Goodrich, P., *Reading the Law*, Oxford, Basil Blackwell, 1986, p.188.

conversation then no legal consequences arise. This is in contrast with the legal definition of night which is the time for users of the road to switch on their headlights. Failure to do so leads to possible legal consequences. A dispute may arise between the law enforcement agency and the "offender" as to whether the time of day might properly be considered to be "night-time" when the offender was stopped. Judicial interpretation may be swayed by a wide variety of factors. Thus for jurists, the borders of meaning are more important than the centre - "The word starts out to free thought and ends by enslaving it," writes Levi, paraphrasing Cardozo[32]. This "problem", which Hart describes as the "core" of certainty and the "penumbra" of doubt[33] is considered further in Chapter 2. Popper, on the other hand warns against the seeking of precision. He writes with reference to the domain of the philosophy of science:

> "..both precision and certainty are false ideals. They are impossible to attain, and therefore dangerously misleading if they are uncritically accepted as guides. The quest for precision is analogous to the quest for certainty, and both should be abandoned...it is always undesirable to make an effort to increase precision for its own sake - especially linguistic precision - since this usually leads to lack of clarity, and to a waste of time and effort on preliminaries which often turn out to be useless...one should never try to be more precise than the problem situation demands.[34]"

Unfortunately legal problem situations can be very demanding. Zadeh has been criticized by some traditionalists as using mere probability under a smoke-screen of logic[35] and by

[32] Levi, E., *An Introduction to Legal Reasoning*, Chicago, University of Chicago, 1949, p.8, paraphrasing Judge Cardozo in *Berkey v. Third Ave Ry. Co*.,, 244 N.Y. 84,94,155 N.E. 58,61 (1926).

[33] Hart, H.L.A., *Positivism and The Separation of Law and Morals*, 71 Harvard Law Review, 1958, pp.593-629, p.607.

[34] Popper, K., *Autobiography of Karl Popper*, in Schilpp, P.A., *The Philosophy of Karl Popper*, La Salle, IL., Open Court, 1974, p.1-181.

[35] "Fuzziness is probability in disguise. I can design a controller with probability that could do the same thing that you could do with fuzzy logic." Professor Myron Tribus, quoted in Kosko, B., *Fuzzy Thinking*, New York, Hyperion, 1993, p.3.

others for developing a system that is not logic at all[36]. Traditional logicians insist that there are discreet values or variables which can be quantified. However, fuzzy logic should be likened to vagueness rather than probability. It claims that the language of probability lacks the necessary expressions to deal with the real world. Therefore values can be non-quantifiables i.e. variables such as words, as well as numbers. Thus, say fuzzy logic theorists, we can add together fuzzy variables such as "young" and "a few years older" to calibrate "middle-aged." Not surprisingly, fuzzy logic offers much appeal to the "real-world" modeller in overcoming the rigidities of classical mathematics. Why are fuzzy systems important to neural network modellers? Lawrence sums up its importance succinctly - "Neural networks and fuzzy systems are similar in that they both process inexact information inexactly.[37]"

1.8 THE SCIENCES OF COMPLEXITY

Complexity theory is an emerging scientific movement, based on what scientists now see revealed in real-world phenomena. Examples of complex systems include the body, its internal organs, the weather, the market economy and population development. Indeed any phenomenon that has a non-linear aspect to it, might be described as complex. Given the emergent qualities of Complexity theory, its boundaries of application are not yet fixed. Pagels describes some of its key themes as;

[36] "Fuzzy theory is wrong, wrong and pernicious. What we need is more logic not less. The danger of fuzzy thinking is that it will encourage the sort of imprecise thinking that has brought us so much trouble. Fuzzy thinking is the cocaine of science." Professor William Kahan, quoted in Kosko, B., *Fuzzy Thinking*, New York, Hyperion, 1993, p.3.

[37] Lawrence, J., *An Introduction to Neural Networks*, California Scientific Software Press, Nevada City, CA., 1992, p.11.

"the importance of biological organizing principles, the computational view of mathematics and physical processes, the emphasis on parallel networks, the importance of non-linear dynamics and selective systems, the new understanding of chaos, experimental mathematics, the connectionist's ideas, neural networks and parallel distributive processes.[38] "

It is arguable that fuzzy logic is a notable omission from the above list. It is also arguable that certain domains of law exhibit characteristics of complexity[39] . Using John Holland's definition of complex adaptive systems, namely that they "form and use internal models to anticipate the future, basing current actions on expected outcomes[40] " we can see that both neural networks and the law qualify under such a heading. For instance, the entire body of Equity appears to act upon the body of the Common law in a non-linear fashion; contradictory cases co-exist side by side whilst new cases swing the dominant view from one perspective to another before eventual abandonment of a given legal understanding and the adoption of an entirely new paradigm to describe our understanding of a given legal concept.

However, this thesis is concerned merely with the connectionist's ideas and neural networks. Although Complexity theory is currently fashionable within the scientific community, we should be careful not to dismiss non-complex phenomena out of hand. It is likely that when we have understood the sciences of complexity a little further that we will look at the non-complex with a new perspective. Hence it is important not to lose sight of one's ancestry both within science and within the discipline of AI.

[38] Pagels, H., ibid., p.13.

[39] See Braithwaite, M., *Prolegomena to a Post-Modernist Theory of Law*, LL.M. thesis, University of British Columbia, October 1993, who develops a theory of law as a Complex Adaptive System.

[40] Holland, J., *Complex Adaptive Systems*, Daedalus, Vol.121, No.1, Winter 1992, pp.17-30 at 23.

1.9 NEURAL NETWORKS

An artificial neural network is a combination of artificial neurons, the connections between them and a learning rule[41] or algorithm. Like people, neural networks learn from experience, not from programming. They do not require the intervention of experts to describe the nature of their expertise in the form of an computer program/algorithm. Traditional non-parallel programs, whether of conventional or expert system variety, require the program builder to describe his/her domain. Ideally, neural networks require no description of the domain from the user; their only requirement should be large quantities of numerical data from which the neural network will induce the connections between functions.

Neural networks are good at pattern recognition, generalization and trend prediction. They require no rules or formulas with which to work and are tolerant of imperfect data. They are computational devices which, through the combined computing power of a very large number of artificial "neurons" (which deal with only very small chunks of information and carry out only very small computations by communicating along links with other neurons) can deal with large complex real-world problems. These so-called Connectionist machines make use of a parallel architecture of computing rather than traditional linear programmable code. However, and this point should be stressed, neural computing should be considered complimentary rather than in competition with

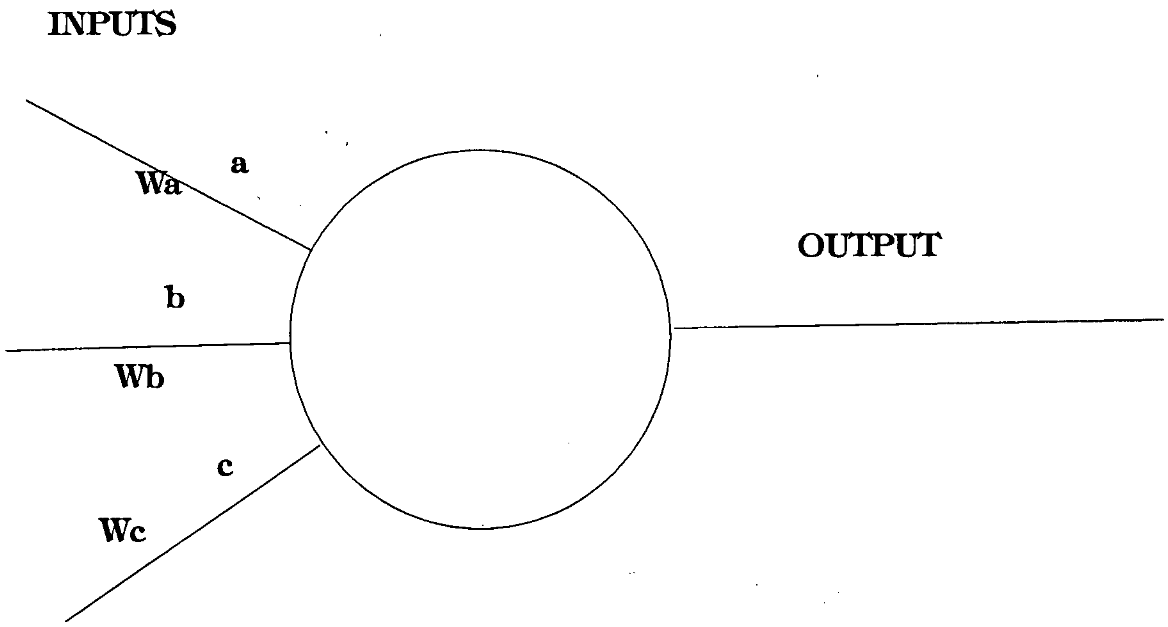
[41] Lawrence, J., ibid., p.79.

conventional computing techniques. Discussion of how connectionist and computationalist architectures may be used in tandem is developed later in the thesis.

1.10 THE ARTIFICIAL NEURON

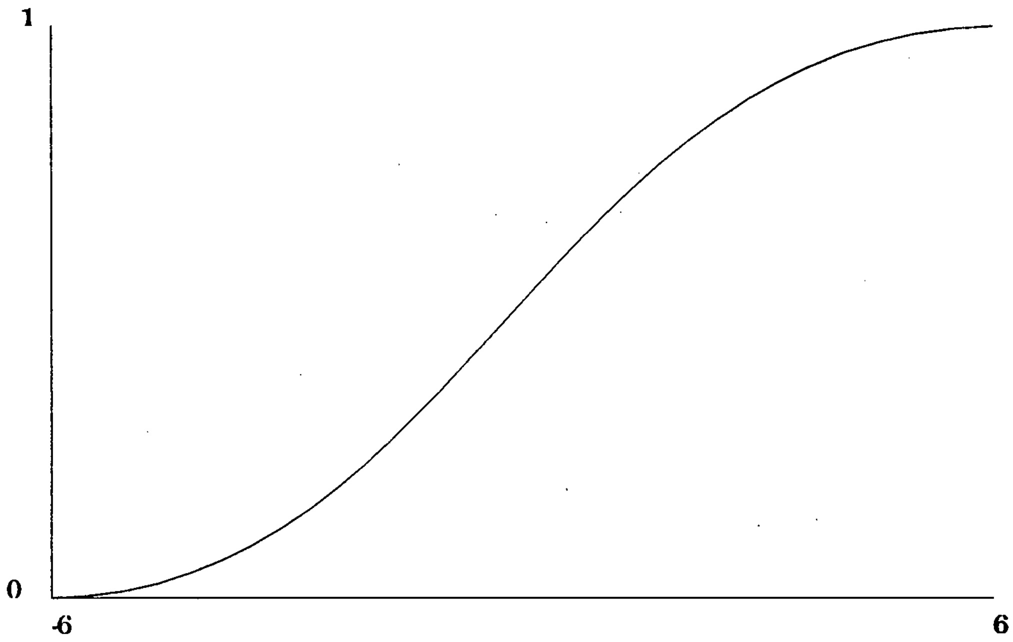
The most basic processing unit of a neural network is called a neurode or neuron which are combined into a "network" or structure. The structure of an artificial neuron is described in Figure 1.1.

Figure 1.1 An Artificial Neuron (x = name of connection, Wx = its associated weight)



At each neuron, every input has an associated weight, which modifies the strength of the input(s) connected to the neuron. The single neuron simply adds together the sum of all the inputs and calculates the output, through vector multiplication and addition. This is done by applying what is known as the "activation function" or "transfer function" to the sum of the input values. The most popular "activation function" used by neural network modellers is called the Sigmoid activation function[42] (see Figure 1.2.) This function is also known as the semi-linear or S-shaped function. Using the Sigmoid function, outputs vary continuously (but not in a linear manner) as inputs change. The most successful type of function can change according to the purpose of the network. According to authority on this issue[43] if the problem involves learning about average behaviour, Sigmoid transfer functions work best. If the problem involves learning about deviations from the average, for example in areas such as bankruptcy prediction and stock picking, the Hyperbolic

Figure 1.2 - The Sigmoid Activation Function

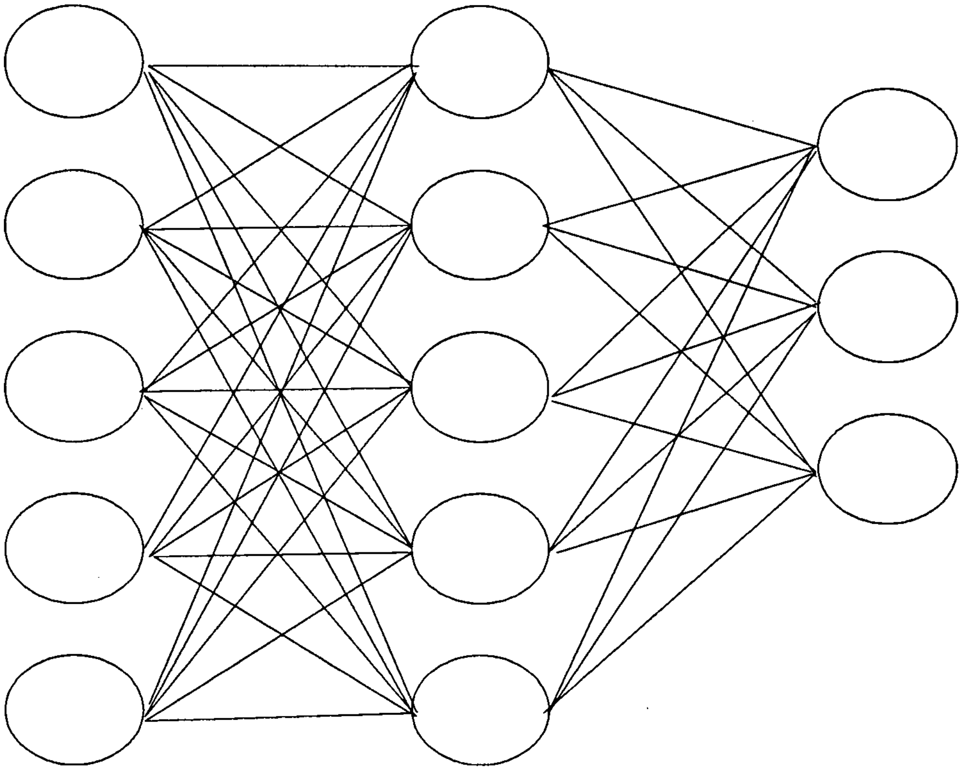


[42] There are a variety of other functions such as the Linear Transfer Function, the Linear Threshold Function, the Step Transfer Function, and Guassian Transfer Function. Each of these have different qualities and are represented by a different curve. An examination of each is beyond the scope of this thesis. By default the Brainmaker software used in this thesis uses a Sigmoid Activation/Transfer Function.

[43] Klimasauskas, C., *Applying Neural Networks,* PC AI, May-June 1991, p.20- 24

Tangent function tends to work the best. (Earliest neural networks tended to use the so-called Step Transfer Function where the output was limited to two values (1 and 0.) It acted in a manner similar to a digital logic chip.) We are concerned with determining average behaviour, therefore the only activation function used throughout the experiments detailed in this thesis is the Sigmoid activation function. Under this function, output values are restricted to a value between zero and one. If the sum of the input values gives a negative value, the output value will equal zero i.e. the neuron will not fire and no learning will occur in this area. A large positive input will yield the maximum output of one. Between these, the output rises in a S-shaped curve to a maximum gain of 1. Upon activation, each neuron releases an electronic impulse as output which is fed forward to the next layer. The "neurons", which are the elementary processing units, are arranged in a particular manner. The most common network architecture is that of three layers, in which the neurons of each layer are connected to all of the neurons of the next layer (as shown in Figure 1.3.)

Figure 1.3 - Common Neural Network Architecture (showing 5 input, 5 hidden and 3 output neurons.)



This is only one of many possible topologies although, again according to authority on this area, it appears to be the most effective. Additional hidden layers may be added to the network although it appear that this is no guarantee of success if simpler three-layer networks failed. I have not questioned this approach and all experiments conducted in conjunction with this thesis have used the Sigmoid transfer function in a three layer network.

How does the neural network learn? Data is fed to the network at the input layer of neurons. (Each neuron at the input layer represents a real-world factor.) The input layer neuron takes the numerical input through a connecting link and produces an output which in turn becomes the input for the hidden layer (or first hidden layer if there are more than one.) The middle layer is termed "hidden" because it is internal to the computer. The user sees only the input and the output. Describing what exactly happens in the hidden layer, beyond a form of vector matrix multiplication is one of the problems that neural network developers are still grappling with.

Neural networks are primarily pattern recognizers. They learn through repetitive exposure to relevant examples by developing associations between input and output data. These associations are stored in the form of weights on connections between neurons. Thus, when the network "sees" something that appears to be similar to a phenomenon it has come across before, it responds with similar output data. To describe fully how a neural network works, we will use a very simple example of a neural network that can recognize dogs based on colour, size and liveliness. The inputs and outputs for the network are described in Figure 1.4

Figure 1.4 - Inputs and Outputs of a very simple neural network

INPUTS		**REQUIRED OUTPUTS**
CHARACTERISTICS		**TYPES OF DOGS**
LARGE		PEKINESE
SMALL		LABRADOR
BLACK		ALSATIAN
BROWN/TAN		NEWFOUNDLAND
YELLOW/GOLDEN		WELSH TERRIER
LIVELY		ST. BERNARD
SLOW		

An association that we might hold is that, of all the above, a small, lively, brown/tan dog is likely to be a Welsh terrier. Thus, the Welsh terrier could be represented in a database as follows;

			INPUTS				OUTPUT	
LARGE	SMALL	BLACK	BROWN	GOLD	LIVELY	SLOW	WELSH	LA
0	1	0	1	0	1	0	1	0

Each of the other animals will be similarly represented in the database. For example, a Newfoundland might be characterized as large, black and slow. A medium sized dog might be represented as 0.5 in each of the "large" and "small" fields, indicating that the predicates "large" and "small" are only 50% true. One can immediately see the application of fuzzy logic theory here. We might wish to "fuzzify" the above data further so that the Welsh terrier is represented by a value of 0.2 in the "Large" database field and 0.8 in the "Small" database field and do likewise for all other inputs, implying that the dog is not small to an absolute degree.

1.11 TRAINING A NEURAL NETWORK

Before the process of back-propogation is described, it should be mentioned that neural networks can be classified into two distinct categories; feed-forward and feedback. The difference is this; if signals/impulses go only one way and output is dependent only on signals from incoming neurons (i.e. there is no looping back,) then the network may be said to be feed-forward. In a feedback neural network, the output signals of neurons directly feed back into neurons in the same or preceding layer. That is to say, they become inputs. It is important for us to recognize at this point that back-propogation is *not* a feedback "looping" network.

1.11.1 BACK - PROPOGATION

Having decided what it is that we wish the neural network to do and collected sufficient amounts of data (in a network-readable form) to allow the network to train, the next step is to train the network. Invented in 1974[44], the most popular, successful and therefore most common form of neural network training is called "back-propogation," (an abbreviation of "back-propogation of error") a form of supervised-learning algorithm which is a generalized form of the so-called Delta-Rule.

1.11.2 THE DELTA RULE

This is the simplest of all learning rules that states:

- Feed the input pattern into the network;
- For every output, determine the error by subtracting the actual output from the desired output.
- Change the weights of the connections proportionately to the product of the incoming signal;
- Repeat until a sufficient level of accuracy has been achieved.

[44] by Paul .J.Werbos, who at the time was a doctoral candidate at Harvard University. It was re-discovered in 1985 or thereabouts by David Rumelhart and David Parker independently of one another.

Thus each neuron processes the piece of data that it receives by finding the sum of all inputs multiplied by the weights matrix (using vector matrix mathematics of multiplication and addition.) The resultant values percolate through the system to the output layer. The actual output is compared with the expected output for a given set of training facts and that error is "propogated" back through the network. In the early stages of training, the difference/discrepancy between expected and actual outputs (the error value) will be very large as the weights on connections between neuron are initially set to random values. The error signal is fed back through the network, altering connection weights as it goes and (hopefully) preventing the same error from repeating itself . On the basis of the error value[45], the network retrains itself on the same set of examples by adjusting the connection weights automatically until gradually the number of "good" results (those results which match or are sufficiently close to the expected output) outweigh the "bad." Correctness of output is judged not on an absolute criteria but as the "best set of conclusions" subject to a standard deviation. The process of weighting adjustment can take a matter of minutes, hours or days depending on the amount of training information, the size of the network, the processing speed of the computer and the required accuracy of the trained network. Once this process of weighting adjustments has occurred a sufficient number of times so that the network is providing intelligible and reasonable outputs, the network is then said to have "learnt." We should note however, that the back-propogation rule can only change the weights of connections - it cannot change the network topology in any way. Alternative network topologies may alter the learning paradigm significantly.

[45] See Appendix F, Glossary of Neural Network Terms.

If we go back to our earlier example and imagine that the network is now training, for the facts “small”, “”brown” and “lively”, the network produces an output of 0.5 for “Welsh,” and 0.5 for “Pekinese” and an output of 0 for all other types of dog. This is telling us that the network is as yet undecided as to which type of dog it is being asked to identify although it has managed to eliminate all but two. The network will automatically make adjustments to its internal weightings of the connections between neurons. It will then move onto the next fact, (in this case, the St.Bernard) and present the facts for that type of dog. If the guess made by the network is incorrect it will make adjustments and continue to the next fact. Once the network has been through all the facts, it will return to the top of the list and re-present the facts once again. It will continue to re-present all the facts until it gets the “right” answer for all inputs. In this application, an output of 0.9 for the type of dog might be acceptable if all other outputs are approximately 0.1. (The tolerance of error is changeable according to the type of application.) Thus a neural network might not actually come to a final conclusion, rather it settles into a pattern that appears to satisfy all given inputs.

1.11.3 SUPERVISED -v- UNSUPERVISED LEARNING

It was mentioned above that back-propogation is a type of supervised learning. In supervised learning there is a human “trainer” (i.e. in this case, the author) who presents data in the form of pairs of inputs and outputs, either inputting the data manually or presenting the data in the form of a database. Thus we know what the results (outputs) should be for the inputs.

With a single layer network, the trainer can actually monitor specific corrections to the weights. However, this is not possible with a multi-layer network. For instance, if we think the network is mistaken in developing significant relationships between a given input and output, it is difficult to correct the weighting of the connections in the hidden layers as we cannot see what is happening. Back-propogation with a single hidden layer of neurons is the primary form of learning used by the software in conjunction with this thesis[46].

In the unsupervised learning model, there is no trainer. The network simply receives inputs. The network is required to come up with its own classification of those inputs and reflects them in the outputs. Such approaches are rarely used in real-world applications because if we cannot provide outputs for the inputs this means that we do not understand the domain. At present it seems that unsupervised learning neural network models are confined to the research laboratory.

1.12 WHY NOT USE EXPERT SYSTEMS?

Those scholars familiar with expert system applications might argue that a dog recognition system could as easily be implemented by inputting *if-then* rules into an expert system

[46] However, there are a wide variety of other such supervised-learning models, Among other Feed-Forward Models, there are the Perceptron, Adaline, Back-Percolation, and Adaptive Logic Networks to name but a few. Feedback models include such as Fuzzy Cognitive Map, Boltzmann Machine, Mean Field Annealing and the Learning Vector Quantization models.

shell. For instance, using the Welsh terrier example, we might input these factors in the following form; IF the dog is brown/tan AND it is small AND it is lively THEN it is a Welsh terrier. The user interface would ask the user; "Is the dog small? - Yes or No", "Is it brown/tan- Yes or No?" and so on, until the expert system deduced that the dog in question must be, of all the above possibilities, a Welsh terrier.

Admittedly one could use either technique in the above example. However, it is one of the primary arguments of the so-called Connectionists that neural network can excel in real world domains where the relationship between the inputs and output is either not apparent or not describable to another human being. This argument will be tested. Neural networks can learn directly from facts or cases, rather than from an experts articulation of a problem and is therefore, one of their primary advantage over expert systems. Very often, an expert in a given domain may understand his/her area comprehensively. However, constraints of language prevent that intuitive power from being articulated to another unskilled in the domain. We consider this phenomenon further in the Conclusion.

1.13 FURTHER ADVANTAGES OF NEURAL NETWORKS

Neural networks require no programming - it is not necessary for a programmer to analyze the problem in sufficient depth in order to write down a set of instructions that the computer can follow. A further attribute of neural networks is that given that they can learn directly from data; they remove the interpretive element that will accompany articulation in the form of rules. Clearly in interpretative disciplines such as law this is an important factor. The ability to learn directly from examples means that in legal application, much less consideration can be given to the jurisprudential framework and implications of the program. This is an important feature for legal applications, for as we shall see in Chapter 2, many expert systems in law projects have faltered because of either an unquestioning adherence to Legal Positivism leading to forms of mechanical jurisprudence or to excessive jurisprudential "navel gazing", where debates regarding the nature of law tend to dominate the empirical aim of producing workable expert systems.

A neural network can adapt its internal configuration during training. They can (at least in theory) deal with incomplete and/or "fuzzy" data - in theory, they can continue to operate with only a partial picture. A traditional program would inform its programmer of a error in the program and cease to function until rectified, the so-called "catastrophic degradation." Failure of any part of a traditional program dooms it to failure. Neural networks adapt their output to accommodate inconsistency and suffer so-called "gradual degradation." Good examples of gradual degradation can be found in our own bodies. Kurswell writes "In animal vision the failure of any neuron to perform its task is irrelevant. Even substantial portions of the visual cortex could be defective with relatively little impact on the quality of the end result.[47]" The network may still function although output becomes less specific. Importantly for legal applications, neural networks also have the potential to generalize with data. Thus until proved otherwise it is fair to believe that the prospect of computerized legal prediction, a goal which is as yet unattained by other forms of AI, may be achievable using this connectionist architecture.

[47] Kursweil, R., *The Age of Intelligent Machines*, Cambridge, MA., MIT Press, 1990, p.231.

1.14 WHERE NEURAL NETWORKS FALTER

A neural network is only as good as the set of examples that are fed to it. It can, in theory, learn the relationships between the number of times a person turns over in their sleep and its seeming effect on interest rate movements if one is minded to present such data as inputs and outputs. It is not intelligent in the sense that it will tell the user that clearly there is no possible relationship. Also neural networks cannot deal easily with classical logic problems, although one can see a superficial similarity between the functioning of a rule-based system and a neural network. For instance, rules are linked together starting with premises which lead to conclusions e.g. If V and W then X; If X and Y then Z. Such premises could be represented in a neural network by having V, W and Y represented by first layer neurons. X would be a hidden layer neuron and Z would be a final layer neuron. V and W combined exert influence on X, and X and Y together exert influence on Z. However, ANN's are imprecise - if one trained a neural network in non-fuzzy mathematics and asked it for the sum of 3+3, the answer given would be roughly 6, (although if enough examples were presented to the network it might eventually provide an absolutely accurate answer.) Since many humans with their complex biological neural networks have problems with certain tasks where traditional programming excels, (such as balancing a check-book, performing logic functions and keeping fine details,) expecting an ANN to be able to do likewise would be unwise. Furthermore, ANN's are "black boxes" - the knowledge that is contained within the program is contained in the strength of the weighted links between neurons and in their topology. None of this is visible to the user. It may be possible to view the numeric weighting data that the network assigns to particular neurons, but even in the smallest networks, this can be unintelligible. This is a serious shortcoming. The "programming" i.e. the continued internal weighting adjustments performed by the network, takes place unseen. Thus, although the user may discover an answer to his or her problem, the reasons why that particular answer is given may not be discernible. It is immediately apparent, therefore that this may have limits on the value of using connectionist architecture in law. No judge hands down a judgment without copious reference explaining his or her reasoning. If the output is one's only concern, then neural networks are likely to have utility, but even in an area such as quantum assessment for whiplash injuries, where the client is often heard to cry in exasperated tones "but what is it worth?" the next question is usually "why?" Therefore any belief that neural networks can be used to replace symbolic programming in its entirety should be dispelled at the earliest opportunity. This is particularly so in an area such as law, so rich in symbolism.

1.15 MINDS AND MACHINES

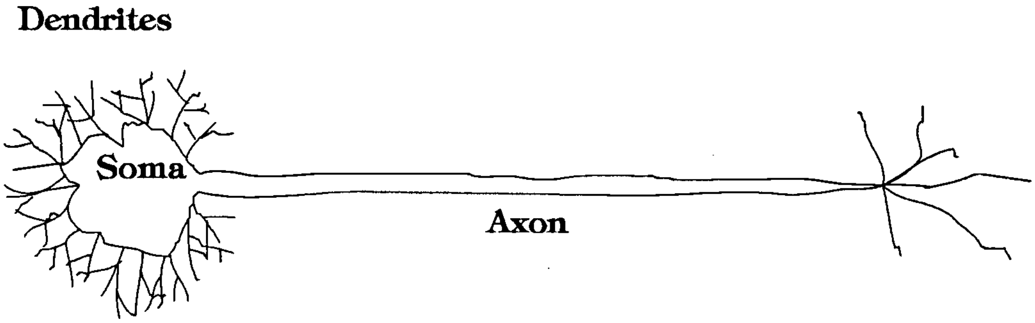
As the name implies, neural networks are modeled on the organizational processes believed to occur in the human brain. Although the human carbon-based version is many times slower than its silicon logic-gate counterpart, even the largest artificial neural networks in modern computer laboratories have about the same number of neurons as those contained in the "brains" of the lower order species. A simple organism may have a few dozen neurons; an insect as simple as a fly has over one million and a human brain has in the region of 100,000,000,000[48]. If we take this number as an estimate of the number of neurons needed to produce a self-conscious entity, then it is clear that we are a long way from building artificial brains. Indeed, at the present rate of progress, it is estimated that humans will not be able to build an artificial brain analogous to that of a human being for 120 years[49].

1.16 THE BIOLOGICAL ANALOGY - THE NEURON

Biological and artificial neural networks contain neurons, either real or simulated respectively. Biological neurons consist of a cell body plus one axon and many dendrites. These can be seen in Figure 1.6. The axon is the protuberance that delivers the neurons output to connections with other neurons. Dendrites are the protuberances that provide surface area to allow contact with other axons. i.e. they maximize the inputs and minimize the outputs. The average biological neuron receives impulses from up to 10,000 other neurons. (In certain parts of the brain such as the cerebellum which is crucial for motor functions, the single neuron may receive impulses from as many as 100,000 axons.)

[48] Lawrence, ibid., p.164.
[49] Lawrence, ibid., p.154.

Figure 1.5 A simplified biological neuron



A given neuron appears to be dormant unless the collective influence of all the inputs reaches a threshold level. Whenever the threshold is reached, the neuron delivers a full-strength charge in the form of a pulse down the axon and into the axon branches. This is known as "firing." Because a neuron either fires or remains dormant, it is said to be an "all-or-nothing" device. The artificial neurons in computer software can function as all or nothing devices as well as continuous output devices. Each neuron (whether carbon or silicon) receives signals from a great many other neurons which will either excite or inhibit the level of activation in the given neuron. The level of activation is the sum of the number of connections, their weighting/size, and the strength of the signals. Thus whilst slower than a parallel computer, the human brain makes up for this deficiency through the sheer scale of this parallel network.

1.17 NEURAL NETWORKS - A BRIEF HISTORICAL ANALYSIS

Given the nature of this thesis, it is impossible to provide a comprehensive analysis of the history of neural networks from the 1940's onwards. Whilst attempting to avoid a lengthy descriptive narrative on the development of neural computing, we should however, take note of a number of important dates in its development. It is accepted that the birth of AI as a discipline took place at the Dartmouth Conference of 1956. (Notable by his absence at that conference was the eminent computer scientist John von Neumann, an avid proponent of neuro-biologically inspired computer architecture[50].) The conclusion of the Dartmouth Conference was that the neurobiological model were an unnecessary diversion, given the apparent rapid development using linear methodology. A good example of the attitude that prevailed at the time is the following quotation: "Whether logical reasoning is really the way the brain works is beside the point," McCarthy says. "This is AI and so we don't care if it's psychologically real.[51]"

1.17.1 McCULLOCH AND PITTS

The first proponents of the neurobiologically-inspired computer were two mathematicians, W.S. McCulloch and W. Pitts who made the first attempt to define an artificial neuron that could be used to provide a mathematical analysis of how networks behave. They were largely overshadowed by conventional machine computing, from the 1960's onwards.

"*A Logical Calculus of the Ideas Immanent in Nervous Activity*[52]," is a technically complex paper, written as much for the neurophysiologist as for the computer scientist and in this sense, is fairly representative of much of their work. McCulloch and Pitts

[50] See for instance, Neumann, J. von, *The Computer and the Brain*, New Haven, Yale University Press, 1958, pp.66-82.

[51] Quoted in Israel, D.S., *The role of logic in knowledge representation*, Computer, Vol. 16, No.10, 1983.

[52] McCulloch, W.S., and Pitts, W., *A Logical Calculus of the Ideas Immanent in Nervous Activity*, 5 Bulletin of Mathematical Biophysics, 1943, pp.127 -147.

describe a computational device which they believe is an adequate representation of how the smallest particle in the biological nervous system might actually work, now known as the "McCulloch-Pitts neuron." This was a binary device which could have only one of two possible states (on or off), possessing a fixed threshold, receiving inputs from excitatory synapses that have identical weights or inhibitory synapses which act as a bar to any action on the part of the neuron (i.e. if the inhibitory synapse is active the neuron cannot fire.) They concluded that the structure of the network does not change with time. Their primary argument with reference to what we now call the discipline of computer science was that, if this type of neuron could be artificially implemented in an information processing machine, it could perform simple logic calculations. It followed from this, that first, any logical expression could be realized using this neuron and thus a network of neurons could perform computations of very great complexity and secondly, that the mind could be understood in computational terms.

(Although the Pitts and McCullough duo are usually described as the forefathers of neural computation, they were not the first to suggest that the activation of one brain cell by another increases the likelihood of such activation in the future. This honour is bestowed on William James and his chapter of a text entitled "Psychology" written in 1890[53].)

[53] James, W., *Psychology* (Briefer Course), New York, Holt, Chapter XVI "Association", pp.253-279.

1.17.2 THE HEBBIAN RULE

Donald Hebb's article "The Organization of Behavior" was the first description of a learning rule for neural network modellers. (It is also the first known use of the term "Connectionism.") Again, written more for the neurophysiologist than the computer scientist, Hebb described what he believed happens to the nerve synapses when used; "when an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency as one of the cells firing B, is increased.[54]" This pattern of re-enforcement has come to be known as the "Hebbian rule" of "synaptic modification." Computer scientists were quick to extend this theory. They argued that it follows from this rule that the strength of connections between neurons that excite or inhibit one another should be increased or decreased respectively, proportionate to activation values.

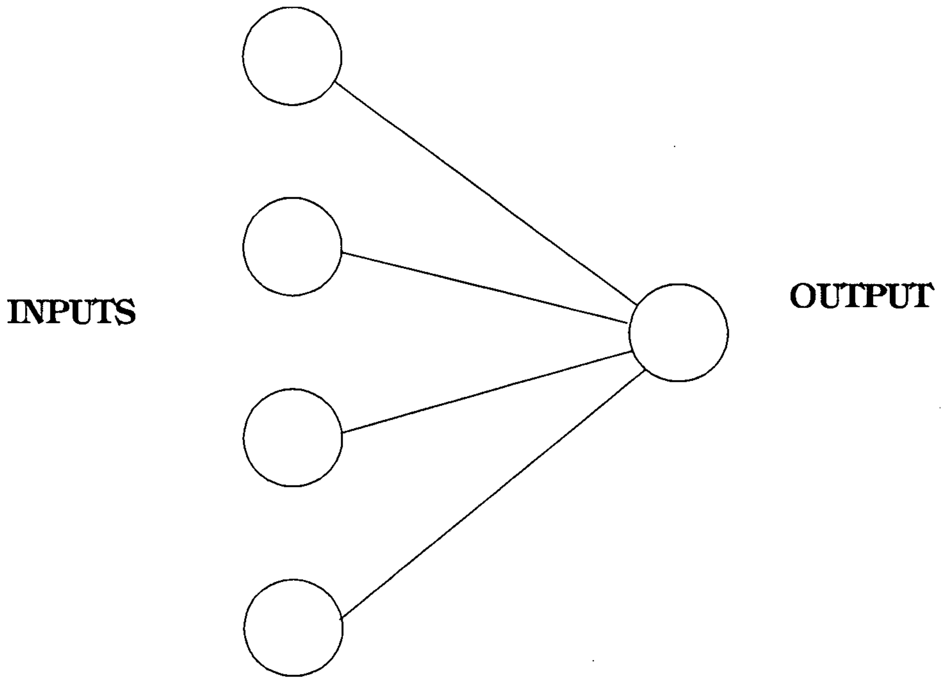
1.17.3 THE PERCEPTRON

Envisaged by Rosenblatt, this one layer feed-forward model of a neural network which utilized the Delta rule mentioned above, created much interest in the community of AI scholars. In this model inputs fire directly into outputs as seen in figure 1.7. In his highly scientific paper describing the Perceptron, Rosenblatt's enthusiasm for the project is evident; "...the system described is sufficient for pattern recognition, associative learning, and such cognitive sets as are necessary for selective attention and selective recall....with

[54] Hebb, D.O., *The Organization of Behavior*, New York, Wiley, 1949, p.50.

proper reinforcement it will be capable of trial and error learning, and can learn to emit ordered sequences of responses...[55] "

Figure 1.6 - The Perceptron



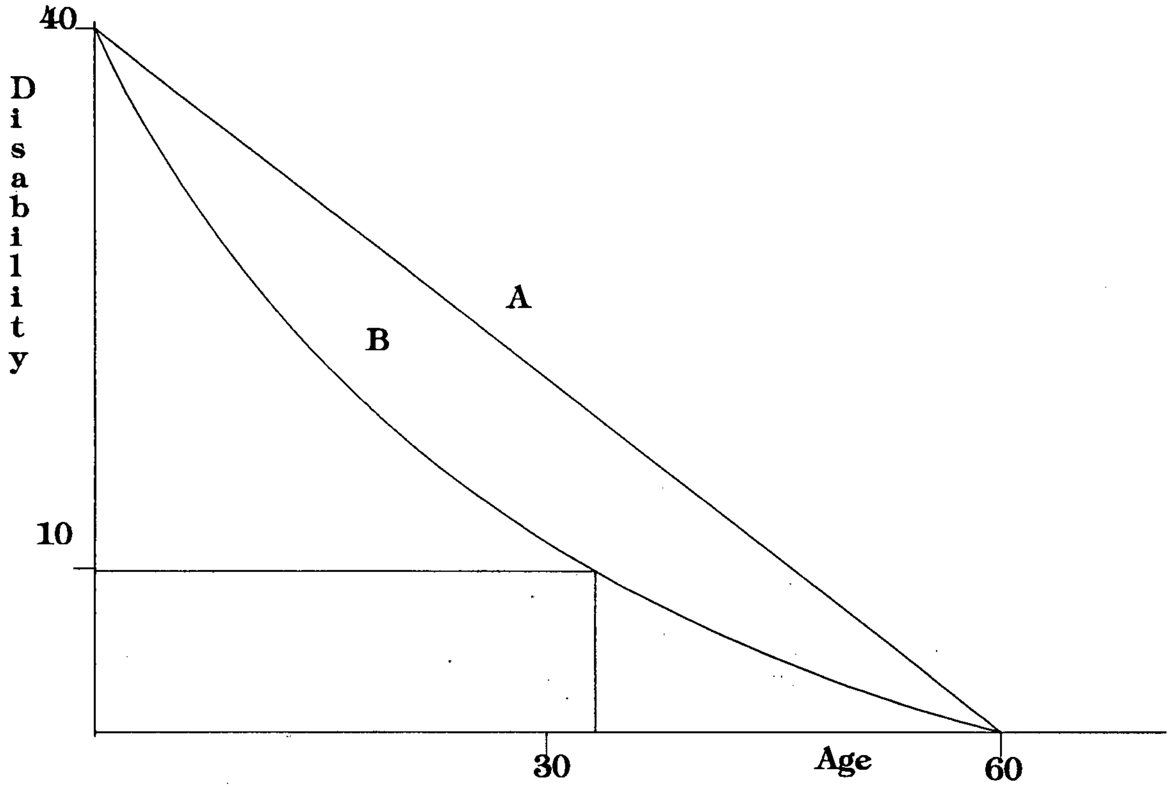
However, Rosenblatt was not cognizant of the problems that the Perceptron presented. For the layperson, there is no better description of the problems Perceptrons develop in relation to the representation of the real-world than the summary of their properties by Opdorp and Walker[56]. They use the example of an apartment without an elevator that is considered suitable for tenants dependent upon their age and degree of disability. If we use a basic premise of the rule, perhaps developed by Welfare Agencies, that after the age

[55] Rosenblatt, F., *The Perceptron: a probabilistic model for information storage and organization in the brain*, Psychological Review 65: p.404.

[56] Van Opdorp, G.J., and Walker, R.F., *A Neural Network Approach to Open Texture*, in Kasperson,H.W.K., & Oskamp, A., *Amongst Friends in Computers and Law*, Deventer, Kluwer Law and Taxation Publishers, 1990, pp.299-309.

of 60 or a disability in excess of 40%, the apartment is unsuitable for a tenant. We could represent these factors graphically as a straight line (see Figure 1.7 - marked "A").

Figure 1.7 Graphic Representation of Hypothetical Welfare Agency Rule.



If in addition we understand that the Welfare Agencies consider an apartment without an elevator unsuitable for a person of (for example) 35+ with a 10% disability then this rule could be presented graphically as a curve (marked "B" above) rather than a line. An increase in disability adversely affect age and visa versa. This curvature of the dividing line between suitable and unsuitable cannot be expressed by a perceptron, even though both age and disability are obviously inhibitory with respect to suitability thus "whenever the contribution of one input neuron should depend on the input values of other receptors, the perceptron will not suffice.[57]" In examples such as this, there must be either a loop between input neurons or at least one hidden layer of neurons to calculate the influence of the two factors upon each other. Such architectures were not envisaged by Rosenblatt.

1.17.4 "THE DARK AGES" - MINSKY AND PAPERT

> "In the popular history of neural networks, first came the classical period of the perceptron, when it seemed as if neural networks could do anything...Then came the onset of the dark ages, where, suddenly, research on neural networks was unloved, unwanted and most important, unfunded. A precipitating factor in this sharp decline was the publication of the book "Perceptrons" by Minsky and Papert[58] ."

Such an analysis is quite dramatic and somewhat overstated. The role played by Minsky and Papert in casting neural networks into the wilderness of underfunded computer science research is not that of key villains. However in their book, they argued that the next logical step for neural networks, that of multi-layer networks of Perceptrons would exhibit the same problems as the single layer network as described above. Such arguments were enough to kill off already flagging interest and funding. In retrospect, it seems that their conclusions are wrong. Rather than dwelling on their excellent, but misconceived analysis of why neural networks in any shape or form would fail, we should merely mark this event as an important turning point in the development of neural networks. The result of this book was that rule-based or so-called symbolic AI was founded and flourished for nearly 30 years. Despite all of this, a "cottage industry" of researchers such as Kohonen,

[57] Van Opdorp,G.J., and Walker,R.F., ibid., p.

[58] Anderson, J.A., and Rosenfeld, E., *Neurocomputing: Foundations of Research*, Cambridge, MA., MIT Press, 1988, Chapter 13 p.161.

Anderson and Hinton continued to work on connectionist models making notable improvements to the notion of the Perceptron. For instance, Anderson and Kohonen (independently of one another in 1972) were successful in developing an artificial neuron that could vary the frequency of its firing, rather than acting as a simple binary device[59]. Also, the notion of feedback (mentioned above) was developed by Hopfield in 1982.

1.18 THE REBIRTH OF CONNECTIONISM

The First Annual Conference on Neural Networks (ICNN) held in San Diego in June, 1987 which attracted nearly 2000 participants may be seen as the neural network equivalent of the Dartmouth Conference of 1956. Along with the work of Hopfield and the emergence of several commercial neural networks, most of the credit for the new found impetus in the neural network research project must go to the PDP group.

1.19 PARALLEL DISTRIBUTED PROCESSING (PDP)

One of the main reasons neural computing has re-appeared as a force within (or, depending on ones perspective, in competition with) AI is the contribution made to the field by a group of researchers, notably McClelland, Rumelhart and Hinton, known as the Parallel Distributed Processing group[60]. They represent the most plausible of all current connectionists' ideas and these ideas have been utilized by many manufacturers of neural network software. Brainmaker, the software used throughout this thesis (and most other

[59] See Kohonen, T., *Corrolation Matrix Memories,* IEEE Transactions on Computers, C-21: pp.353-359 and Anderson, J.A., *A simple neural network generating an interactive* memory, Mathematical Biosciences, 14: pp.197-220.

commercially available neural network software) draws on the work performed by this group. Their fundamental view is that we humans are smarter than computers because we have developed a computational architecture more suited to the tasks that surround us (i.e. those that require common sense) and those that appear to require the simultaneous consideration and manipulation of many pieces of information and the drawing of conclusions by induction. A key element to their theory is that understanding is derived from the interplay of various sources of information. One of the examples that they use in this context is of a child's birthday party at a restaurant. They write "We know things about birthday parties, and we know things about restaurants, but we would not want to assume that we have explicit knowledge about the conjunction of the two." However, from our experiences we are able to conceive and develop a likely scenario in such instances. Thus, although traditional AI approaches can capture knowledge in the form of structures whether scripts, frames or otherwise, "any theory that tries to account for human knowledge using script-like knowledge structures will have to allow them to interact with each other to capture the generative capacity of human understanding in novel situations[61] ." They are of the belief that PDP can satisfy this requirement. Using the back-propogation model with a hidden layer, they are able to demonstrate that neural networks can develop many solutions to any given problem simultaneously, instead of the traditional form of computation where one answer leads to the next.

[60] See Rumelhart,D.E., and McClelland,J.L, ibid. This is now the standard reference work on Connectionism.

[61] Rumelhart,D.E., and McClelland,J.L., ibid., p.9.

1.20 REAL WORLD APPLICATIONS OF NEURAL NETWORKS

So what can neural networks now do? Neural network technology appears to be capturing the imagination of academics and business people world-wide. Not surprisingly the first users of neural networks were electrical engineers in the large defense-related research universities of the United States. For instance, the ALVINN system ("Autonomous Land Vehicle In a Neural Net") developed by Carnegie-Mellon University learns to drive a vehicle along roads viewed through a television camera. It must first be trained on a particular road. Once this has been achieved, it can guide itself at speeds of up to 40 mph[62]. Another research project that has caught the imagination of the outside world is NetTalk, a back-propogation neural network that takes text and learns to turn it into speech. It reads a string of characters and turns them into phonemes. It then converts these into audible signals which are outputted via the computer's speaker. The two developers, Sejnowski and Rosenberg trained networks using 500 words from a Grade 1 reading level book, then later with 20,000 words from a dictionary. Using a three layer network but enormous computing power and many training hours, they were able to train a network that could produce phonemes output at a level of accuracy of 95%. Users of neural computing now include manufacturers[63], investment companies[64], direct marketers[65] and even makers of burglar alarms[66]. Governments have also been quick to seize the initiative. For instance the UK Government's Department of Trade and Industry

[62] Winston, P.H, *Artificial Intelligence*, Reading, MA., Addison-Wesley Publishing Co., 1992, p.458.

[63] See for instance, Kinoshita, J., *Neural Nets at Work*, Scientific American, Vol.259, No.5, pp.134-135.

[64] Swales, G.S., and Yoon, *Applying Artificial Neural Networks to Investment Analysis*, Financial Analysts Journal, Sept - Oct., 1992, Vol.48, No.5,p.78.

[65] See for instance, Classe, A., *Little Grey Cells*, Accountancy, May 1993, v.111 (n.1197), p.67.

[66] Somers.J. et al., *Catching Intruders in a Neural Net*, Security Management, Feb. 1994, v.38 (n2), p.68.

announced a 6 million GBP (CAN$12 million) scheme to promote the use of neural networks in response to a similar initiative by Japan's Ministry of International Trade and Industry. The Canadian Imperial Bank of Commerce (CIBC) recently announced that they will be using neural networks (and in particular the Professional version of Brainmaker, in order to provide "better forecasts of Canada's Gross Domestic Product and inflation than the usual econometric models[67]."

The increase in number of publications and products might suggest a fundamental paradigm shift within AI towards Connectionism. However, we have yet to conclude whether any of these products will make a more significant impact than traditional AI approaches upon business. At present we are riding the crest of a media-inspired wave of neural network "mania." It is difficult to draw rational conclusions regarding their potential in such an environment. The fundamental questions to be tested in this thesis are therefore; *given the increasing use of neural computing applications in business and finance, does Connectionism have any role to play in the artificial modelling of law and more particularly, in legal reasoning and case prediction, and if so, what role(s)?* Before we can answer this question, we must explore further the domain of law and examine the problems that this discipline presents.

[67] See unnamed article in Wired, September, 1994, p.39.

CHAPTER 2

ISSUES OF JURISPRUDENCE

2.1 INTRODUCTION

This chapter seeks to define the parameters of a sub-discipline of Jurimetrics known as Artificial Intelligence (AI) and Law - the marriage of a much-maligned science and a much-considered pseudo-science. In doing so, it reflects upon the two major trains of legal theory - Positivism and Anti-Positivism within jurisprudence and seeks to set the application of artificial neural networks within this context. Scholars in this sub-discipline have been quick to note the bilateral nature of the relationship between the parent disciplines. First, they point out that AI applications such as expert systems and other such software, present an excellent opportunity for the jurisprudential scholar to test theory in terms of practical application. Secondly, they describe features of the legal domain that make law an excellent representative model of the real-world and thus ideal for computer modeling. This Chapter focuses on the former at the expense of the latter.

Before we become overly concerned with what may be described as the predicament of AI applications in law, we should bear in mind that the vast majority of legal disputes, the so-called "easy cases" are concluded without recourse to the court. They are concluded satisfactorily based upon what both sides believe the law would say were they to ask. Despite the monumental production of legal material by the courts, the number of disputes heard in Courts of Law are minimal compared with the number of disputes that occur within Western societies. Nevertheless, the number of disputes heard by the courts is sufficient to merit an investigation of the viability of building programs that can potentially provide answers to the so-called 'hard cases." This last statement also merits investigation. We should be asking ourselves - are there correct answers, a range of correct answers to these so-called 'hard cases" or should the word "correct" have no place in the vocabulary of this discussion.

2.2 AI AND LAW - FROM WHERE TO WHERE...

The history of AI and law can be traced to an article by Buchanan and Headrick entitled "*Some Speculation About Artificial Intelligence and Legal Reasoning*[68]." In this article, the authors note that interdisciplinary work between the fields of computer science and law has foundered due to the misconceptions that each carry about the other. It was argued in that article that lawyers, in assessing the clients likelihood of success in litigation, tried to analyze the fact pattern in issue with those of previous cases. The lawyer required an ability to search through large amounts of data and needed to predict what could affect the potential of a favourable settlement. In all of the above, it was argued, computers could have significant implications. However, the authors pointed to other problems such as "the search for programmable rules used by lawyers" and "the disparity between the natural statement of the rule and the formal statement within the programming language.[69]" For almost a decade these and other problems appeared to be insurmountable.

[68] Stanford Law Review, 23, November 1970, pp.40-61.
[69] ibid., p.46.

Until the mid-1980's computer-based information technologies in law were little noted, outside of their immediate circle of users in legal practice. Although heavily used for data/information retrieval, they did not seek to assist, second guess or usurp the reasoning faculties of the legal practitioner, administrator or court official and were thus considered to be relatively unremarkable. The arrival of the expert system appeared to alter this perspective radically. They allowed the computer novice to manipulate natural language to formulate models of legal systems previously only comprehensible through the medium of text.

For legal academics the arrival of the expert system in law aroused much interest. It appeared that the modeling of law using expert systems could inform jurisprudential debate. For the Positivists, legal expert systems represented the physical embodiment of the view that law can be represented as a system of rules; in its purest form law could be objective, determinate and neutral.

However, it is argued that such a view is merely a further manifestation of the adage "when the only tool you have is a hammer, every problem tends to resemble a nail.[70] " It was thought that AI might possibly assist in the so-called Teleological/Positivist debate, possibly explaining reasons for inconsistencies and/or developing methods to deal with this such issues.

[70] Abraham Maslow, 1908- 1970, Psychologist.

2.3 LEGAL REASONING AND THE EXPERT SYSTEM

Law is concerned with what is acceptable and unacceptable behaviour in a given society. In Western legal culture, there are two ways of communicating standards - either by rules or examples. Not surprisingly, researchers interested in emulating legal reasoning model use both of these forms. However, there are numerous questions to be answered such as - what is the nature of a rule, what form does it take - (i.e. doctrine/principle/policy), how do lawyers use those rules or examples? Thus, the fundamental problem for creating working expert systems in law is founded in an antecedent problem of the creation of adequate legal theory. If we compare the problems of creating expert systems in law with those of creating expert systems in law's nearest professional relative, namely medicine, we see that the latter suffers few of the difficulties that law possesses. For instance, one does not have to consider the nature of medicine before one builds an expert system to perform medical diagnosis. Susskind sums up this problem for us;

> 'It is beyond argument, however, that *all expert systems must conform to some jurisprudential theory because all expert systems in law necessarily make assumptions about the nature of law and legal reasoning.* To be more specific all expert systems must embody a theory of structure and individuation of laws, a theory of legal norms, a theory of descriptive legal science, a theory of legal reasoning, a theory of logic and the law and a theory of legal systems as well as elements of a semantic theory, a sociology and psychology of law (theories that must all themselves rest on more basic philosophical foundations.[71])"

This is no small task. Sadly, however few researchers heed this advice. Unquestioning adherence to rule-based formalism appears to predominate[72]. Researchers tend to omit

[71] Susskind, R.E., *Expert Systems in Law - A Jurisprudential Inquiry*, Oxford, Clarendon Press, 1987, p.20, (his emphasis.)

[72] For instance, Dave Brown (of the University of Melbourne Law School) comments "Few, if any of the papers at this conference questioned the basic assumptions of the field...A strong legal and philosophical input is needed in any major project in this area if its aims are to produce any worthwhile and useable applications." (Taken from Brown,D., "The Third International Conference of Artificial Intelligence and

from their modeling the view that rules in law are not merely to be followed; order and predictability in law, although perhaps desirable are not two of its primary attributes. Rules are to be extended, adapted and interpreted. The law remains permanently in a state of flux, constantly altering the boundaries between its concepts. A lawyer is obliged to present the most favorable interpretation of the rule, in pursuance of the goals of the client, thus shifting the boundaries of a given legal concept. Courtrooms rarely witness the direct application of a pre-existing rule, otherwise the matter would be settled long before trial. Thus the use of an objective formalization methodology severely underestimates the function of rules in the legal context.

2.4 LOGIC-BASED APPROACHES

Any system of logic is concerned with meaning, with propositions and their interaction. It will consist of notations and a set of rules. The most rigid form of rule-based formalism is that of classical logic. We should begin this brief review with what Posner describes as "the apt and famous syllogism[73]," "All men are mortal; Socrates is a man; Therefore Socrates is mortal." The validity of the argument is undeniable. Legal reasoning/thinking appears to work on occasions, in such a manner. One can point to large bodies of statutory law in domains such as labour and immigration law, where at least, superficially, the law functions in such a manner. Thus, runs the theory, legal judgments can be viewed as logical deductions from accepted principles, abstract and divorced from experience.

Law; A Report and Comments", Journal of Law and Information Science, Vol.2 No.2, 1991, pp.223- 239, at p.238.

[73] Posner, R.A., *The Problems of Jurisprudence*, Cambridge, MA., Harvard University Press, 1990.

Yet, what this approach fails to grasp is that logic and rational action need not necessarily equate.

Earliest forays into the area by researchers such as Kowalski and Sergot et al.[74] adopted a strict rule-based isomorphic[75] approach. Whilst subscribing to the rule-based "worldview," it would appear that they were not close readers of Hart's work and his warnings against the use of deductive reasoning. Hart was very much alive to the folly of purely deductive reasoning in law. He writes;

"...deductive reasoning, which for generations has been cherished as the very perfection of human reasoning cannot serve as a model for what judges or indeed anyone should do in bringing particular cases under general rules.[76] "

Deductive reasoning cannot be entirely dismissed however. The drafting of a legal document owes much to deduction. When drafting a document such as a contract, the lawyer is effectively writing an algorithmic program to be performed by humans. Where deduction fails to play a significant role is in the process of reasoning with precedents. Unfortunately, the deductive functioning of certain statutes proved too seductive to the team from Imperial College, London, who displayed proficiency as skilled logicians but naiveté vis-à-vis jurisprudence. Notable for their absence of lawyers, this team of computer scientists used PROLOG, a logic programming language, to implement the British Nationalities Act 1981 in the form of an expert system. This approach has been

[74] Sergot M.J., Sadri, F., Kowalski, R.A., Kriwaczek,F., Hammond, P., and Cory, H.T, *The British Nationality Act as a logic program*, Communications of the ACM, 29, 370-383.

[75] Isomorphism refers to an attempt to represent the letter or substance of the law, rather than an attempt to deal with the issues of open texture.

[76] Hart, H.L.A., *Positivism and The Separation of Law and Morals*, 71 Harvard Law Review, 1958, pp.593-629

widely criticized by legal researchers for a naive appreciation of the jurisprudential implications of their work. In their defense however, it should be pointed out that such applications were built "to demonstrate and test developing techniques of logic programming[77]" rather than to prove the validity of logic/rule-based approaches to legal reasoning modeling.

2.5 JURISPRUDENTIAL THEORY AND ARTIFICIAL MODELS

So much has been written on the topic of Western jurisprudence - a full survey would consume the majority of this thesis and would also deflect us from our main task. Since our first glimpse of codified law provided by the Code of Hammurabi about 2000 B.C., we have pondered upon the nature and substance of law. Aristotle made the first attempts to distinguish the rule of law from the ruler and emphasized a unemotional exercise of human reasoning as the ultimate paragon. More recently, we have seen the emergence of a feminist jurisprudence, critiques from the post-modernist, Marxist, race-based and psycho-analytic perspectives (to name but a few.) As yet they appear to have made little impact upon the development of artificially intelligent models of legal reasoning. While acknowledging the myriad of critiques, we will focus upon those perspectives and critiques that have directly inspired researchers in our field. The task is to plot a course through the morass of words and ideas, to "tear through the veil of pretentious verbiage[78]" that surround theories of law to extract those aspects of each jurisprudential theory that actually inform the theoretical basis for the building of a neural network in law.

[77] Kowalski, R., and Sergot, M., *The Use of Logical Models in Legal Problem Solving*, Ratio Juris, Vol.3, No.2, July 1992, 201-218 at 201.
[78] Loevinger, L., *Jurimetrics - The Next Step Forward*, 33 Minnesota Law Review, p.455 at p.466.

In this regard, we must seek the substantive conclusions that have emerged from the debate between Positivism and the plethora of anti-Positivist views.

2.6 THE USE OF RULES

2.6.1 POSITIVISM

Positivism has a remarkably resilient character that re-manufactures itself in new guises. With the benefit of hindsight, we can see that H.L.A. Hart's book "The Concept of Law" (owing much to the work of Wittgenstein and Russell,) caused a significant paradigm shift in the nature of Legal Positivism. Prior to its publication in 1961, jurisprudence was dominated by the views promulgated by, inter alia, Bentham and Austin, who argued that, whilst the law should be divorced from morality, it should be seen as a set of commands or rules, the "positive" morality backed by a sanction, reducible to an empirical base. Hart made arguably the most important contribution to jurisprudential thinking this century by reformulating the views of Austin into the view that laws are legitimate unless there is a violation of some higher law[79].

Of most importance to the computer modeler is Hart's analysis of the nature of rules which is dominated by the notion of "open texture" - the indeterminacy that overshadows both rules and precedents. We might wish to equate the term with "fuzziness" as described

[79] Hart describes law as a unity of primary and secondary rules. Primary rules govern conduct. Secondary rules specify how ,if necessary primary rules should be modified, eliminated, used or adapted. Hart identifies as the most important secondary rule, a "rule of recognition" which identifies some features that primary rules must exhibit in order to be considered valid. Thus, in Canada, the Charter of Rights serves as the rule of recognition.

in Chapter 1. According to Hart, rules have a core of settled meaning and a penumbra of uncertainty, whose extent can vary from case to case. The example used by Hart to describe the problem of indeterminacy and "open texture" is a law that prohibits vehicles from a public park. There are certain vehicles that clearly fall within the core meaning of the rule such as a motor car. Other vehicles might fall into the penumbra and become those borderline cases on which the judicial actors are required to adjudicate. However, Fuller rightly point out "Professor Hart's...extended discussion of the core and penumbra is not just a complicated way of recognizing that some cases are hard, while others are easy,[80] " although this would seem to be a major theme of Legal Positivism in its various guises. (The distinction between hard and easy cases is for most lawyers meaningless. All cases that come across a lawyer's desk tend to be hard, else the issue would be settled. Thus any expert system that deals only with easy cases is worthless.) It is a question of determining the range of reference of a word - a analysis of the fuzzy set that contains the word in issue. Fuller remarks that words are rarely interpreted in isolation. When a judge does consider the penumbral case, she/he is forced into a "more creative role." Fuller also highlights the issue of the purposive or teleological function of law and in doing so, an inadequacy of Hart's theory. He uses the example of a vagrancy law, which states "it shall be a misdemeanor...to sleep in a railway station," a law which most rational thinkers would argue is designed to prevent homeless people or travelers from sleeping in a railway station. Fuller then posits the situation where the first arrested offender is sitting upright but asleep and the second has brought a blanket and pillow with the intention of sleep, but

[80] Fuller, L., *Positivism and Fidelity - A Reply to Professor Hart*, 71 Harvard Law Review 1958, pp.630-672.

was still awake at the time of arrest. Fuller asks 'Which of these cases presents the "standard instance" of the sleep?" The problem here is teleological rather than semantic - we know what sleep is and we also believe we appreciate the purpose of the law. He suggests that in cases such as these the purposive interpretation of the law must therefore override its literal interpretation.

This example is only one part of the so-called Hart-Fuller debate. We can see that, whilst highly informative, the debate can have no conclusion. Such a debate is false as the two parties were working from entirely different axioms. It has, however, acted an important springboard for further discussion on the nature of rules.

Hart's successor to the Chair of Jurisprudence at Oxford, the more contemporary jurisprudential figure of Ronald Dworkin is insistent that in law, as with literary interpretation, every case has a right answer and that discretion has no role to play. However, the right answer may not be found through the application of rules. A so-called "rights theorist", Dworkin maintains that in hard cases legal premises consist of rules, backed by principles and policy. Thus when the rules fail to address the issue in hand, the judge must "weigh and balance[81]" the competing principles and conflicting interests, implying an objective test. The notion of principles should be distinguished from public policy, which Dworkin believes has little part to play. Dworkin's writing uses the

[81] Dworkin, R., *Taking Rights Seriously*, Cambridge, MA., Harvard University Press, 1977, p.22. This echoes Hart's view that "there are ...areas of conduct where much must be left to be developed by courts or officials striking a balance...between competing interests which vary in weight from case to case," Hart, H.L.A., *The Concept of Law*, Oxford, Clarendon Press, 1961, p. 24.

terminology of "discovery" rather than "creation" in this context. Thus, when the rules "run out", a judge applies certain fundamental principles to "discover" the law. An example of such a principle might be - a man shall not profit from his own wrong. However, upon what objective criteria does the judge perform this process of weighing and balancing? This is an unconvincing metaphor and potentially misleading as there can be no objective criteria. It would seem that such action relies largely upon the discretion of the judge.

2.6.2 LIMITATIONS OF THE RULE-BASED APPROACH

It is easy to see the superficial similarity between the model of legal reasoning from the perspective of symbolic-based AI approaches to problem-solving. Positivism considers the majority of legal decisions to be bounded within the intellectually limited confines of doctrinal rules. The Dworkin theory outlined above, argues that, beyond the limit of rules, discretion (for want of a better title) is exercised by a weighing of factors. However, Coval and Smith show that this metaphor bears little relation to the predictable (and rule-governed) fashion that they believe to be judicial method[82]. Expert systems are rule-based inference engines. Thus any excursion into the realm of discretion must sentence the system to failure, unless we can develop a method by which to analyze the seemingly unanalyzable.

[82] Coval, S., & Smith,J.C., *Law and its Presuppositions: Actions, Agents and Rules*, London, Routledge and Kegan Paul, 1986.

The experience of many researchers in the field is summed up by Leith. He writes;

'My own history in the field (of expert systems and law) is of someone who began, relatively optimistically, to build such a system ...However, during this building, I grew more and more skeptical of being able to handle legal knowledge in a computer system. This skepticism can be reduced to the simple statement that real legal skill is a much more amorphous quantity than the builders of legal expert systems believe.[83] "

Clearly mechanical jurisprudence, the so-called 'judicial slot-machine" theory, has little to offer and should be abandoned as an underlying methodology. However, undeniably rules do play a role in law. The question is at what level?

There are further problems with the rule-based paradigm generally. Expert systems tend to deal with highly specialized and often obscure domains. Given that it is necessary for the expert to articulate a theory of how the law works in this domain, why should the user of the system trust the interpretation of an expert, A, who has built an expert system, as opposed to the views held by expert B, who has no such technical prowess? We can use Hart's in support here;

"Any honest description of the use of precedent in English law must allow a place for the following pairs of contrasting facts. First, there is no single method of determining the rule for which a given authoritative precedent is an authority. Notwithstanding this, in the vast majority of decided cases there is very little doubt. The head-note is usually correct enough. Secondly, *there is no authoritative and uniquely correct formulation of any rule to be extracted from cases.* On the other hand, there is often very general agreement, when the bearing of a precedent on a later case is in issue, that a given formulation is adequate.[84] "

Raz, however, finds few of these difficulties; 'In the main, however, the identification of the ratio of the case is reasonably straightforward.[85] " Williams also sees few problems;

[83] Leith, P., *The Computerized Lawyer - A Guide to the use of computers in the legal profession,* London, Springer-Verlag, 1991, p.189-190.
[84] Hart, H.L.A., *The Concept of Law,* Oxford, Clarendon Press, 1961, p.131, (my emphasis.)
[85] Raz, J., *The Authority of Law, essays on law and morality,* Oxford, Clarendon Press, 1979.p.79

"The Ratio Decedendi of a case can be defined as the material facts of a case plus a decision thereon.[86]" The problem therefore, may not be extracting the rule from the case; rather it is the problem of deciding whether the case in issue is sufficiently similar to the so-called "precedent" case and if so, in what respects, so as to permit the application of the extracted rule.

2.6.3 RULE SKEPTICISM & LEGAL REALISM

Again, it is unnecessary to explore the plethora of views held at the opposite end of the jurisprudential spectrum, unless they impact upon artificial legal reasoning modeling. Rule skeptics take a variety of forms[87] from those who merely find difficulty with the view that legal reasoning is predominantly rule-governed to those of a radically Critical orientation, who become almost nihilistic in their belief that law is entirely functional in upholding the capitalist hegemony - in essence, law is power. Rule-Skeptics question the purpose of engaging in legal reasoning. Rather than being a utilitarian method by which legal actors can predict the outcome of a given fact situation, they see it as playing a justificatory role, either obscuring or focusing attention upon existent dominant power relations. Legal realists insist that legal doctrine plays little or no role in the determination of disputes, rather it is a justificatory smoke screen for conclusions reached intuitively. Thus, they argue, it is more worthwhile to focus on the factual outcome of cases and their effects on wider society. The only real law say the Realists, is the decision of one particular judge in

[86] Williams, G., *Learning the Law*, London, Stevens & Sons, 1978, p.62

[87] Indeed, most lawyers who have ever had occasion to present a case before a judge tend to exhibit some form of rule-skepticism.

a given specific case. Even if there is a rule that would appear to apply to the case in issue, the lawyer is still well advised to consult the relevant precedents. If any such precedent reveals that the judge in that case erred from the accepted rule, the lawyer must analyze in what ways that case might be different and whether the case is issue exhibits any similar features.

Any discussion of the so-called 'Realist" movement must pay respect to the figure of Oliver Wendell Holmes, whose place in the development of American jurisprudence is un-paralleled[88]. An extreme member of the Pragmatic philosophical school, his work was largely a reaction against rule formalism, the legal positivist obsession - "The prophesies of what the court will do in fact and nothing more pretentious, are what I mean by the law.[89]"

Posner writes;

> 'Holmes shows the absurdity of supposing, as did the nineteenth century formalists against whom he was writing, that legal doctrines were unchangeable formal concepts like the Pythagorean theorem. He enforced the lesson of ethical relativism, thereby turning law into dominant public opinion in much the same way that Nietzsche turned morality into public opinion.[90]"

However, Holmes found agreement with his philosophical adversary, Hart, regarding the lawyers obligation to deal only with the law as it is, rather than as it ought to be and accepted the possibility of a scientific valuation in law. This is of little importance to our

[88] Those theorists who build on Holmes' work include, inter alia, Cardozo, B., *The Nature of the Judicial Process*, New Haven, Yale University Press, 1921, and Llewellyn, K., *Jurisprudence; realism in theory and practice*, Chicago, University of Chicago Press, 1962.

[89] Holmes, O.W., *The Path of the Law*, in Collected Legal Paper, New York, Harcourt, Brace & Co., 1920, p.173.

[90] Posner, R.A., ibid., p.240.

purpose as we seek to embody a descriptive rather than prescriptive view of the law. Unfortunately, from our point of view, Holmes was unable to articulate any such theory for the operation of legal reasoning. Thus, although highly critical of traditional jurisprudential approaches, and of great significance to those theorists concerned with law and its interaction with the social and policy contexts, Holmes (and Legal Realism as a jurisprudential movement) offer little other than vague clichés and platitudes for the Jurimetrician as to how the process of legal reasoning may be explained.

2.6.4 THE ANALYTICAL - TELEOLOGICAL APPROACH

We should be careful not to dismiss all rule-based approaches as completely without merit. Rather, they require a combination with a theory that takes account of the alternative discourses of economic, political and gender-based power relations existent within law. Critical scholarship is informative in this regard. The only theory to date that appears to take account of the accepted importance of rules whilst also acknowledging the importance of these alternative discourses is the analytic-teleological theory or "deep-structure" theory of law as expounded by Professor J.C. Smith. Deep structure argues that beneath doctrine there exists a non-conceptualized rule-based structure that governs decision-making within law. (Doctrine has, at most, an indirect influence.) This accounts for how given fact situations relate to the social values that are reflected within existing legal discourse. Thus judges decide hard cases in a rule-based manner however these rules are more likely to be unconsciously developed meta-rules. "According to deep structure theory, social goals teleologically justify legal doctrine," writes Braithwaite[91]. When reading cases on the chosen domain, the expert system developer is encouraged to use his/her analytical skills and possibly his/her intuition to rationalize the law based upon goals of the law and reason inductively by scanning all available cases. For Smith, law is explained in terms of actions and goals. For example, in the law of torts, there are conflicting interests between liberty of action, (formalized in the "remoteness" doctrine) and the prevention of harm to a third party. The law will structure these goals in terms of importance. The goals are not so much "balanced" as Dworkin would suggest, as ranked with one meta-rule dominating the second in a given case or in an entire domain. This goal-based analysis is then formalized in the form of "fuzzy rules" where sub-doctrinal heuristics may be brought to bear. "Once these patterns have been identified a broadly stated meta-rule or principle may be formulated which explains the general direction of the case law in the domain independently of the "surface discourse" of law at the doctrinal level.[92]" Whilst utilizing a rule-based theory, deep structure overcomes the inadequacies of Positivist analyzes of law at a doctrinal level. It is both Critical in that it accepts the socio-economic context in which law operates, whilst also avoiding the tendency towards nihilism that rule-skepticism exhibits. This approach has also shown a considerable degree of success in practice[93]. Therefore, in considering existing rule-based paradigms, we

[91] Braithwaite, M.J., ibid., p.10.

[92] Kowalski, A., *Case-Based Reasoning and the Deep Structure Approach to Knowledge Representation,* Proceedings of the Third International Conference of Artificial Intelligence and Law, ACM Press, Oxford, 1991, p21 at p.22.

[93] For example, Kowalski's systems appear to have achieved over 90% accuracy in predicting outcomes in 10 out of 11 cases. See also Deedman,G.C., *Building Rule-Based Expert Systems in Case-Based Law,* LL.M. thesis, UBC Faculty of Law, Vancouver, B.C and MacCrimmon, M.T., *Facts, Stories and the Hearsay Rule,* Pre-Proceedings of the III International Conference on Logica, Informatica, Diretto, Florence, 1989.

should be careful not to dismiss every attempt to emulate the real-world in such a manner as ill-founded. What should be criticized is the naive philosophical presuppositions of some researchers in law and AI that law is no more than a discreet set of doctrinal rules, a unitary model unconnected from the economic and political values and prejudices prevalent in our society.

However, there are factors that the analytic-teleological approach cannot address. It is undeniable that there is still an interpretive element involved in the formalization of the case generalizations. Some of the subtleties of the legal argument may be lost in the attempt to force the legal domain into the narrow confines of the logic of an expert system shell. Although Andrej Kowalski found that "the deep structure approach successfully explains and accounts for legal decisions in areas of case-based law generally considered to be unstructured and indeterminate[94]," some legal problems may be so complex as to defeat ones analytic skills, by for example exhibiting an apparent randomness. However, to date the approach has shown considerable success and appears to be the only credible coherent rule-based technique sufficient to explain the actions of the law whilst also taking account of non-doctrinal factors.

2.6.5 GARDNER

One of the most notable rule-based research efforts in AI and law was by Anne von der Lieth Gardner. It was not overly successful. She used LISP (list processing) language to

[94] Kowalski, A., ibid., p.22.

build a program that was designed to reason as to whether a particular set of circumstances constituted an offer and acceptance under the law of contract. Her conclusion was that ;

> 'Legal philosophy tells us, among other things, that legal rules do not dictate legal outcomes; there is more going on in the decision of a case than ordinary deduction. One extra element is that there is often room for choice about which, of several possible decisions, is to be preferred. Another is that the choice, once made, sets a precedent, which may change the space of choices available in later cases.[95] "

It would appear that two elements defeat her research. The first is the fact that there comes a point where her software concludes that the judge has a choice or discretion. Sadly, she was not able to program her software to emulate the legal reasoning process in this regard. The second element is open texture. She states:

> 'Underlying both the use of examples and the provision for expert disagreement is the idea that legal predicates have open texture. The phrase *open texture* is a catchall...For some predicates there may be a clear prototype case with many possible variations. For others there may be a definition that looks analytical but is in fact defeasible. For still others the concept may be so abstract that its range can be worked out only piecemeal. But these are all phenomena of natural language.[96] "

Given such conclusions, perhaps research funding might be better spent exploring the problems of the structure of natural language, of which legal language is after all, only a sub-category. It would be beyond the mandate of this thesis to attempt to answer questions regarding whether language reflects or shapes reality. Nevertheless, we would be wise to contemplate the wider context including the meaning of such questions when evaluating attempts to impose the models of empirical science upon non-empirical phenomena.

[95] Gardner, A.v.d.L., *An Artificial Intelligence Approach to Legal Reasoning*, Cambridge, Mass., MIT Press, 1987, p.190.

[96] Gardner, ibid., p.190.

2.6.6 THE WHIPLASH EXPERT SYSTEM

We cannot leave this area without making reference to an attempt to build a rule-based expert system to predict quantum for whiplash injuries during the summer of 1990, by Deena McLeod, (a law student at the University of British Columbia.) Unfortunately, the program is no longer working thus further dissection is precluded. However, it is understood that the expert system never functioned as expected and the project was abandoned. Why did it fail? It would seem that the researcher failed to discover the rule-based relationships between factors that appear to influence the decision making process. What does one do then, if one cannot discover the relationships between factors that are believed to unify the cases in the given domain? Expert systems cannot function in such domains. What if the cases offer no unifying structure, but instead exhibit an apparent randomness that evades analysis? A further problem common to all expert systems, including those based on the analytic-teleological approach is that the law is seldom static. Thus any problem solving model should ideally have a mechanism that can be easily modified.

2.7 CASE BASED REASONING AND THE IMPORTANCE OF ANALOGY

'The life of the law has not been logic; it has been experience" writes Holmes[97]. An alternative to the rule-based approaches, whether based on doctrinal or sub-doctrinal rules is Case-Based Reasoning (CBR). If we revert to the medical analogy once again, a physician who diagnoses the illness of a new patient is reminded of the illness of a past

[97] Holmes, O.W., ibid., p173.

patient to see if the former diagnosis is of relevance. In essence, the problem solver, whether medical, legal or otherwise, uses a past experience to solve a new problem. Analogy is the appreciation of the resemblance of several characteristics between two cases, situations or states of affairs, a discovery of the community of language, an aspect of the principle of universalization, which argues that like cases should be treated alike. Connectionism, like CBR utilizes this principle of universalization and is a close relative of CBR in that they are both models of computer-based problem solving and human cognition.

2.7.1 LEVI

The jurisprudential foundation for this approach can be found in the work of Edward Levi[98] who may be seen as a writer largely within the Frank[99] (rule-skeptic) tradition. Levi writes:

> 'The basic pattern of legal reasoning is *reasoning by example*. It is reasoning from case to case. It is a three-step process described by the doctrine of precedent in which a proposition descriptive of a first case is made into a rule of law and then applied to a next similar situation. The steps are these: similarity is seen between cases; next the rule of law inherent in the first case is announced; then the rule of law is made applicable to the second case. This is a method of reasoning necessary for the law, but it has characteristics which, under other circumstances, might be considered imperfections."
>
> 'Reasoning by example in the law is a key to many things. It indicates in part the hold which the law process has over the litigants. They have participated in the law-making. They are bound by something they helped to make.[100]"

[98] Levi, E., *An Introduction to Legal Reasoning*, Chicago, University of Chicago Press, 1949.

[99] Judge Jerome Frank, *Courts on Trial,* Princeton, Princeton University Press, 1949 and *Law and the Modern Mind*, Garden City, NY., Doubleday Press, 1963.

[100] Levi, E., ibid., p.5.

Cardozo offers a helpful analogy at this point. He describes the process of discovering analogy as a process of search and comparison. Of some judges, he argues that their notion of duty is to match the colours of the case in hand against the colours of many sample cases in front of the judge. The sample nearest in shade supplies the applicable rule. Levi augments this view. He does not dismiss the view that the law is a system of rules. However, he argues that the rules are unknown. Rather they evolve over time, in the process of determining similarity or difference with the precedent cases. However, there are problems with this approach; Lashbrooke writes of Levi, " ...the system (described by Levi) is imperfect unless there is some overall rule that provides that the finding of similarity is decisive. No such rule exists.[101] " Again, the decision as to whether similarity exists between two given cases is within the discretion of the judge. What then is similarity? Upon what basis does a judge decide that there is sufficient similarity so as to be guided by the precedent case? Are similar cases to be likened to certain legal concepts such as malice or pornography - very difficult to define but 'you know it when you see it'? Alternatively, there may be specific criteria which justify a particular form of categorization; if so what are they? There appears to be no objective process by which one performs this task.

Mital and Johnson conclude " it is problems such as the above (theories of similarity) which have made neural computing technology look attractive, both to developers of practical systems and to those who are primarily interested in clarifying issues in legal reasoning...there are certain features available with neural networks which help a machine

[101] Lashbrooke, E.C., *Legal Reasoning and Artificial Intelligence*, Loyola Law Review, 34, 1988, p.287.

to go beyond the constraints of having to follow a pre-defined set of rules or a priori relevancies.[102] " With these similarity problems in mind, we can review various attempts to build Case-Based Reasoners in law.

2.8 APPLICATIONS OF CBR IN LAW: ASHLEY

The best example of CBR[103] in law and one that avoids the empirical weighting of cases is the HYPO system built by Ashley and Rissland[104]. Having appreciated the inevitable complexity of the judicial decision-making process, they shied away from case prediction as a goal for the system. Nevertheless, this system is one of the most impressive and efficient available. The underlying rational is that 'reasoning by analogy is the key method of legal argument.[105] "

The system works as follows; a case is represented by a frame consisting of up to 13 dimensions each representing a factual situation, known as a factual predicate. The use of only 13 dimensions might suggest that the case analysis functions lack a necessary depth. However, the system appears to have been developed successfully with so few. Ashley describes dimensions as representing 'the features of a legal case that are important for the strength of a Plaintiff's position on a particular kind of claim. They represent the legal

[102] Mital, V., and Johnson, L., *Advanced Information Systems for Lawyers*, London, Chapman & Hall, 1992, p.257.

[103] For an excellent review of the variety of CBR projects in law, readers are referred to Chapter 14 of Mital, V., and Johnson, L., ibid.

[104] Rissland, E., and Ashley, K., *HYPO: A Precedent-Based Legal Reasoner*, in Vandengurghe, G.P.V., *Advanced Topics of Law and Information Technology*, Hillsdale, NJ, Lawrence Erlbaum, 1989, p.327.

[105] Ashley, K., *Reasoning by Analogy: A Survey of Selected AI Research with Implications For Legal Expert Systems*, in Walter, C., *Computing Power and Legal Reasoning*, St.Paul, Minn., West Publishing Co., 1985. p.105.

relationship between various clusters of operative facts and the legal conclusions that they support or undermine.[106]" Examples of such dimensions might be "an employee has switched employer." Once all dimensions have been adequately described, the system searches through its internal database of cases and retrieves those cases which most closely match the dimensions of the case, including those which potentially favour the other side, and lists them hierarchically. (This approach closely bears certain similarity to the so-called "Quadrant" feature of the FLEXICON system as developed by the FLAIR project under the guidance of J.C.Smith[107] although the latter system can be potentially applied to any domain of the law as opposed to merely trade secrets law. (However, FLEXICON has no pretensions vis-à-vis AI. It is not a Case-Based Reasoner, but a tool to enhance legal information retrieval.) One of the most important functions of HYPO is its ability to identify the legal dimensions of any given case, including those which either favour the Plaintiff or Defendant. Hypotheticals can be created and examined by varying values in the problem case, thus extending the system's ability to create legal argument.

Although an important advance in the development of law-based CBR, Mital and Johnson point to a deficiency of the HYPO project, which is also mirrored by current Neural Networks techniques. They state "the reasoning by which the decision in the case was reached - or supposed reasoning, considering that a common law court tends not to state

[106] Ashley, K., *Modelling Legal Argument: Reasoning with Cases and Hypotheticals,* Ph.D thesis, Univ. of Mass., 1988.

[107] See Smith, J.C., & Gelbart, Daphne, *Beyond Boolean Search: FLEXICON, A Legal Text-Based Intelligent System,* Proceedings of the Third International Conference on Artificial Intelligence and Law, Oxford, ACM Press, 1991

definitively how a particular decision was arrived at - is not represented explicitly.[108]" The second way in which Neural Networks may be seen to mirror the HYPO approach is that the term "dimension" appears to bear close resemblance to the connections between neurons in the neural network. The key difference between HYPO and neural networks is that, in theory the ANN can learn the important dimensions of a domain and also learn to ignore the trivial factors.

2.9 SUSSKIND

Susskind's book "Expert Systems in Law[109]", (a book based on his doctoral thesis and relating to his practical experience with the "Oxford Project," a proto-type expert system dealing with grounds for divorce under Scottish matrimonial law) attempts to shed some much-needed light upon the jurisprudential debate in respect of AI and law. He attempts to develop a core of consensus between jurisprudential thinkers such as Dworkin, Raz, Ross and Hart, on key to the jurisprudential questions that relate to the problems of building expert systems in law. With regard to the question of the primary purpose of engaging in legal reasoning, whether justification, prediction or persuasion, (one of the primary points of divisions between Positivist and Rule-Skeptics,) Susskind cuts the Gordian knot by claiming that the legal knowledge engineer does not have to decide if the program is to assist in justification, prediction or persuasion. He argues that all parties share some fundamental assumptions. These include a shared acknowledgment of "the possibility of legal reasoning, that is, an activity whereby legal consequences can be

[108] Mital, V., and Johnson, L., ibid., p.233.

[109] Susskind, R.E., *Expert Systems in Law A Jurisprudential Inquiry*, Oxford, Clarendon Press, 1987.

attached to acts, events, and states of affairs of our world," that legal reasoning is guided, if not by principles of logic then by principles of rationality and finally that all three functions of law presuppose a more fundamental function "namely that of stating what is true or false within the universe of legal discourse...together with that of deriving the implications of such truth or falsity in respect of particular facts.[110]" Consequently, runs his contention, once derived some may justify it, others use it for prediction and others bring it to bear in their rhetorical deliverances. Such arguments are unlikely to convince the rule-skeptics. Nevertheless, this impressive intellectual 'footwork" neatly sidesteps many of the problems that tend to beset jurisprudential debate and permit Susskind to further develop his contentions.

He goes on to identify the important characteristics of a legal domain for the building of an expert system; the domain should be relatively autonomous, sources of law should be limited and well defined, there should be agreement between experts as to their scope and content and the use of so-called commonsense knowledge should not be necessary in the reasoning process. Speaking from experience, experts may agree on the interpretation of cases yet the weight to be assigned to a given precedent with regard to issues of substantive law is often in issue. Although the scope of the law is usually reasonably defined (in terms of which cases are relevant) the application of the laws/rules to the facts in issue are often disputed. Indeed, this should not surprise given that the adversarial process and the various legal institutions actually encourage disagreements of this nature to be aired.

[110] Susskind, R.E., ibid., p.42.

The factors mentioned above would seem to severely limit the domains in which expert system in law may be applicable. Moreover, as time elapses and the limitations of the legal domain for computer modeling remain, it seem increasingly less likely that we will succeed in capturing the elusive qualities of legal reasoning through deontic logic. Seven years have elapsed since the publication of Susskind's book, yet workable expert systems have still failed to materialize in the legal office. Thus, in summary, his book not only reveals the inadequacies of expert systems in law based on traditional views within jurisprudence, but also reveals how little we know about how legal actors combine human judgment with the complexities of legal formalism.

2.10 WAHLGREN

We cannot leave the domain of AI and law without making mention of Peter Walhgren's influential book 'The Automation of Legal Reasoning[111] " which must rank alongside the works of Susskind, Ashley and Gardner in terms of the contribution made to understanding in this field.

It would be difficult to find a more comprehensive review of legal theory and legal reasoning than Walhgren's. However, the importance of this work lies not with the literature review but with the methodological approach. Wahlgren settles with the view that legal reasoning is primarily rule-based. The obvious criticism to direct at Wahlgren is

[111] Wahlgren,P., *The Automation of Legal Reasoning - A study on Artificial Intelligence and Law*, Deventer, Boston, Kluwer Law and Taxation Publishers, 1992.

that the adoption of this view neglects the bahavioural analysis approach of the rule-skeptics outlined above. However, he is able to counter this criticism, by pointing to his coherent methodological approach. He argues that his purpose is to create a *model* of legal reasoning that can be represented on the computer, rather than attempting to capture all forms of legal phenomena in his programming. Thus the model does not need to reflect all competing theories. Models of law are simplifications; as Smith writes 'in creating models we select certain features and leave out others. We can circumvent the problem of not having a single theory of law, by selecting a particular theory and creating a model on the bases of the presuppositions of that theory.[112]" Thus the validity (or otherwise) of legal positivism is no longer in issue. This is a novel and intelligent approach, given the morass of complexities and (to state the matter more crudely,) utter verbiage that characterizes a large degree of legal theory. However, the models created would be of no more than academic interest, as a result.

Although Wahlgren does not follow this argument through, his methodology suggests that it may be possible to adopt any theory of law, (i.e. Marxist, feminist, etc.,) select certain features of the domain and build an expert system model of that domain. The deficiencies of the adopted theory are of little relevance as one could argue that the expert system is merely a simplified model. Nevertheless, if the selected theory, for example feminist legal theory was successful in explaining the institutional bias against females in the domain of, for example child custody law, then one might argue that one had gone some way in

[112] Smith, J.C., *Review of The Automation of Legal Reasoning*, Stockholm, 5 Juridisk Tidskift, Nr.1, 1993-1994, p.247- 256 at 253.

providing empirical proof that validates the foundational theory. However, this argument could be countered by an argument (yet again) that such computer representation are merely models not real-world representations.

Wahlgren argues that the process of legal reasoning is highly complex. It may be divided, he says, into six sub-processes. First, the *identification* process whereby the legal actor defines the facts in terms of a legal description - defining the relevant facts, discovery of important missing facts and establishing uncertain or disputable facts. Secondly, the *law-search* process, or the discovery and inspection of relevant and possibly contradictory legal materials, at documentary, rule or concept levels. Thirdly, the *interpretation* process using context and analogy. Fourthly, the *application of rules* guided by methodological rules that explain the presuppositions for rule-application. Fifthly, the *evaluation process*, the process by which the legal actor determines the likely conclusion from the application of the chosen rule(s) and may involve a reformulation if the application indicates an outcome that is not in the best interest of the client. Finally, the process of *formulation*, the process by which the legal actor decides how best to describe the matter in issue. Speaking from the position of some experience in legal practice, commercial lawyers would rarely go through this process in its entirety[113]. It seems that for all our efforts, the development of a unified theory of legal reasoning may be a misconceived project. The reasoning processes of a non-litigious commercial lawyer dedicated to formalizing joint venture agreements would seem to have little relevance to

[113] It is difficult to discern what thought processes occur. Moreover, there is no reason to believe that all lawyers reason in the same way.

the work of a tax administrator or judge of the Admiralty Court. The processes used by different types of legal actors must be distinguished.

It is hardly surprising that Wahlgren should adopt a Legal Positivist approach to legal reasoning, as this view tends to predominate European jurisprudential thinking. Nevertheless common law case-based reasoning plays a substantial role in law, at least in the Anglo-American legal context. To ignore these complexities, is to provide a shallow theory for the purposes of modeling the law of North America and the UK. Furthermore, Wahlgren's theoretical analysis of the process of legal reasoning, although comprehensive, appears complex and unwieldy and is likely to prove difficult in implementation by others.

2.11 A CONNECTIONIST THEORY OF LAW?

Perhaps it is the case that prior modeling of law using Artificial Intelligence techniques has failed because of our insistence on the correctness of our view that legal reasoning is essentially a sequential process? Perhaps we are guilty of a blinkered vision; to re-use the adage; 'if the only tool you have is a hammer, then everything tends to resemble a nail." Is it the case that legal reasoning is, in fact, a parallel process?

The practical model described herein uses a parallel architecture, therefore we should posit the question - is there a parallel-processing/Connectionist theory of law? Before we attempt an answer to this, we should ask an antecedent question, albeit rhetorically, namely - do we need yet another theory of law to supplement the mire of existing literature on the nature of law? In answering this latter question, we should remember that the purpose of examining the nature of law serves a very practical need in this sub-discipline. This is not merely an intellectually rigorous exercise performed for its own merit. Hence if a theory can be found, then it should be sought. We cannot (and should not) offer a definitive answer until we have workable models as empirical proof. Even then it is still arguable that the parallel processing model works only in chosen discreet domains of law and cannot be utilized in the wider body of sequentially-based law.

Certain authors, notably Warner, seem to have few doubts on the matter. Although the pattern of legal reasoning appears to be the "decomposition of a problem into issues or unit problems, the resolution of each unit problem, and the synthesis of the unit solutions into a resolution of the whole problem," i.e. a seemingly sequential model of problem-solving, he argues that the "...resolution of the unit problem will impose a "state change" on the entire problem domain. This state change will render invalid earlier resolutions of unit problems, requiring their recalculation, and alter the calculations required for the resolution of remaining unit problems. This approach to problem solving appears inherent in the reasoning process used by lawyers.[114]" In support of this contention, Warner directs us back to Levi ; "...it appears that the kind of reasoning involved in the legal process is one in which the classification changes as the classification is made. The rules change as the rules are applied. More importantly, the rules arise out of a process which,

[114] Warner, D.R., *A Neural Network-Based Law Machine: Initial Steps*, 18 Rutgers Law and Technology Journal, 1992, p.52.

while comparing fact situations creates the rules and then applies them.[115]" Such conclusions cry out for empirical supporting evidence.

However, as stated above, it is important, if not vital, to distinguish the processes used in legal practice. Legal analysis does not necessarily equate with legal reasoning. By legal reasoning, we mean the processes by which judges decide cases, involving the use of precedent and the finding of analogy. Legal analysis is defined by Meldman as "the logical derivation of a legal conclusion from a particular factual situation in the light of some body of legal doctrine[116]" or the process by which lawyers draw conclusions from the description of a given problem. (The term rational is to be preferred to logical in this context.) What Warner does not make clear is, in referring to the "state change", he is referring to legal reasoning or legal analysis.

In any event, Warner is of the opinion that a sequential problem solving approach is insufficient. He writes "The sequential problem solving model fails largely because it overlooks the dichotomy between doctrine and utility. It fails because law cannot be rigidly defined in most instances." One interpretation of this criticism would be to view this description of the dichotomous character of the law as no more than a rephrasing of Fuller's criticism of Hart - i.e. a denial of the purposive content of law and therefore, an inadequate analysis. Warner continues "...the legal system has been described as "open textured" which means it is infected with varying degrees of indeterminacy. When lawyers

[115] Levi, E., *An Introduction to Legal Reasoning*, Chicago, University of Chicago Press, 1949, pp.3-4.
[116] Meldman, J.A., *A Structural Model for Computer-Aided Legal Analysis,* Rutgers Journal of Computers and Technology Law, 6, 1977, p.30.

engage in the problem solving process, they do not strive to find the one answer to the problem, but rather their objective is the identification of a range of solutions.[117]" Whilst experience suggests that this argument has an element of truth, lawyers do tend to adopt one position. They pursue the best of the potential range of solutions to meet their clients' needs, while also attempting to leave their position unfettered in order to pursue an alternative should the first option fail to materialize. A good example, might be exploring the possibilities of a "without prejudice" settlement while also serving pleadings/affidavits on the other side to show the opposing side that lawyer and client take the dispute seriously. Does this amount to parallel processing or is Warner trying to bend his analysis to suit his purposes?

Given that we have yet to build adequate models of law using a Connectionist architecture, it seems to be a little premature to define law using a Connectionist vocabulary. Instead of developing a theoretical foundation with which to justify his intuitive belief that Connectionist architecture is an adequate descriptive model of the legal reasoning processes, scientific method dictates that Warner should first develop working models, observe the results and then tune his theoretical arguments regarding the Connectionist attributes of law. However, we should also be minded of the irony that some of the most important development in the so-called logical, rational sciences were products of intuition. For instance, Einstein wrote of his discovery of the elementary laws

[117] Warner, D.R., *Towards a Simple Law Machine*, Jurimetrics Journal, 29, 1989, p.460.

of physics '‘to these elementary laws there leads no logical path, but only intuition, supported by being sympathetically in touch with experience.[118]"

Once again, Warner's work implicitly calls for a sociological analysis of legal procedures; solid empirical research that shows where lawyers use analysis and where they use reasoning would greatly enhance the level of debate. Leith argues that our pre-occupation with the legal text may be deflecting our attempts at comprehension of the legal environment. He writes "...law is an interconnected body of practices, ideology, social attitudes and legal texts, the latter being in many ways the least important.[119]" Development of a hierarchy of importance of the above, although of significance, is nowhere near as important as an overall appreciation of the complex interaction between these factors.

2.12 LEGAL REASONING AS A SKILL

The difference between connectionist or parallel thinking and sequential thinking is evident in the area of learning. Law School provides the student an appreciation of no more than a skeletal outline of the law. Although legal academics explain to their students that the law operates within a distinct political and socio-economic environment, the lesson of how that environment operates is something best taught in the real-world, rather than in any law school. In addition legal practice teaches the novice lawyer when to use the law, when not to use it and how to use it.

[118] Quotation taken from Holton,G., *Thematic Origins of Scientific Thought; Keplar to Einstein*, Cambridge, Harvard University Press, 1973, p.357.
[119] Leith,P., ibid., p.213.

If we consider the practice of law to be no more than a developed skill, then we can make useful comparisons between learning law in practice and any other skill that a person might pick up. For instance, driving a car is a skill that most people learn by practice. After a few months, the skill develops from a deliberate conscious sequential set of actions to becoming natural and seemingly intuitive. Some thinkers, notably the Dreyfus brothers[120] have concluded that in the process of mastering a skill, we progress from sequential to parallel thinking. A sudden reflection upon what the person is doing is likely to bring with it a sudden degradation in performance. For example, although we all know that there are rules for the point at which a person should change gear in a standard/stick-shift car, (in terms of RPM,) few drivers rely on their rev counter for an indication. They know when it 'feels right." They conducted research into how a person acquires a skill. They conclude that there are five[121] stages to skill acquisition, whether the playing of chess, flying of planes, riding a bike or learning to read. These are novice, advanced beginner, competence, proficiency and expertise. During the first stage, they write, "the novice learns to recognize various objective facts and features relevant to the skill and acquires rules for determining actions based upon those facts and features.[122]" They acquire "context-free" facts and rules. For instance, in a legal context, the law student who wishes to practice in litigation learns "context-free" facts and rules such as the court

[120] Dreyfus, H.L., & Dreyfus, S.E., *Mind Over Machine*, New York, Free Press, 1986. For more on this, see also Dreyfus, S.E., and Dreyfus, H.L., *Towards a reconcilation of phenomenology and AI*, in Partridge, D., and Wilks, Y., *The foundations of artificial intelligence - A sourcebook*, Cambridge, Cambridge University Press, 1990.

[121] Whether there are four, five or even ten stages of learning is not important. Instead, we should be aware that different skills are employed according to our stage of development.

[122] Dreyfus.H.L., & Dreyfus, S.E., ibid, p.21.

procedures, the meaning of "without prejudice" negotiations, methods of cross-examination, etc. During the second stage, the beginner starts to recognize context-free features by a "perceived similarity with prior examples[123]." Rules become located within a given situation. Using our legal example, an advanced beginner in legal practice might learn spot a problem with the client's evidence, having seen similar problems previously. Thus being able to spot the problem through experience becomes more important than being able to describe the problem in terms of rules. The key factor for Dreyfus and Dreyfus in the third stage is the sense of what is missing as well as what is there.

The fourth stage is proficiency. At this point, the proficient performer has a wealth of experience. Knowing what to say in a court room in any given situation that will appeal to a judge's understanding and beliefs is intuitive and can only by gained by experience - it is an holistic situation-based similarity recognition process - the sense of "deja vu" and recalling what worked last time. The lawyer will discriminate factors in any given situation as conspicuous against those to be ignored. "The relative salience of features will change[124]" with events. This is intuition. Dreyfus and Dreyfus write "intuition or know-how, as we understand it, is neither wild guessing nor super-natural inspiration, but the sort of ability we all use all of the time as we go about our everyday tasks... (The irony is not missed.)..an ability that out tradition has acknowledged only in women usually in interpersonal situations, and has adjudged inferior to masculine rationality[125]." Thus true experts can act in a seemingly irrational manner and yet still succeed.

[123] Dreyfus, H.L, & Dreyfus, S.E., ibid. p.22.
[124] Dreyfus, H.L, & Dreyfus, S.E., ibid. p.28.
[125] Dreyfus, H.L, & Dreyfus, S.E., ibid. p.29.

2.13 CONCLUSION

In jurisprudence as with other complex philosophical domains, one should not seek answers but merely attempt to clarify increasingly complex questions. Having accused other theorists of excessive "navel gazing" regarding the issue of the jurisprudential foundations for the building of AI models in law, this thesis is also in danger of surrendering itself to such problems. In the war of attrition between the army of AI researchers aimed with their weapons of choice and the citadel that contains that nebulous phenomenon known as the complexity of law, the latter is standing firm against almost all offensives.

Speaking from my own experience of working in a law firm, although there may be a variance of opinion amongst lawyers as to what the law actually is in a given instance, there is usually an agreement over what the appropriate action in each case should be. The lawyers tend to be aware of the range of solutions. This is not so much an awareness of what the law will say in the final instance but rather an agreement as to what would be the rational (but not necessarily logical) action for our client to take. Equally, there is usually agreement as to what action the opposing side should take and to prepare accordingly. Could we, therefore, build computer models of law based largely on legal intuition and ignore the formalistic complexities of the law. This would undoubtedly prove to be an interesting research topic.

In any event, from the review outlined in this Chapter, it is clear that with a few notable exceptions, this application of expert systems in law is beset with theoretical difficulties. The problems with symbolic representations of law are mirrored by what is now accepted as the failure of the traditional AI reductionist paradigm. Over ten years of research has been expended on AI in law. Given that the Connectionist paradigm is being exploited in other domains, it seems appropriate to now evaluate its potential with reference to the domain of law.

CHAPTER 3

NEURAL NETWORKS AND LAW - A LITERATURE REVIEW

3.1 INTRODUCTION

Open texture is a problem of everyday language, although its problems carry particular significance in law, as open texture often becomes become a matter of day-to-day debate in any Court of Law. Problems of open texture are not only reflected in law, they are also magnified. Therefore, before reviewing connectionist applications in legal language, we should ask if connectionist models can overcome the difficulties of natural language where symbolic approaches have largely failed. We will then look at the progress to date on topics within Connectionism and law and also look at the research on the use of connectionist architectures for alternative applications in the legal field such as for information retrieval.

3.2 CONNECTIONISM AND NATURAL LANGUAGE

The view that until we can understand natural language we will toil in vain trying to comprehend legal language, although not often aired in public, is a fear that almost all research workers in this field must have felt at one time or another. Studies of natural language are particularly complex and controversial . We know that we are able to expand our vocabulary as we use language. We also know from a reading of Chomsky[126] that

[126] Chomsky, N., *A Review of Verbal Behavior by B.F.Skinner* in Language, 35, 26-58, 1959.

natural language cannot be described in simple linear terms. Similarly, it is arguable that we change the legal language as we use it. Therefore if connectionist approaches to natural language prove successful, they may also prove successful with reference to legal applications which as we saw earlier are often concerned with meaning on the borders of language. Without wishing to be drawn into a review of all AI approaches to natural language which would inevitably involve a review of a substantial body of literature beyond the scope of this thesis, we should acknowledge the efforts of researchers such as Rumelhart and McClelland and the PDP research group[127], who represent efforts from the Connectionist perspective. McClelland maintains that "conventional symbolic approaches only partially characterize human language, because human language is not strictly compositional.[128]" i.e. a word does not make the same semantic contribution to the meaning of every expression in which it occurs. In short, the whole cannot be explained as the sum of the parts. The parts are useful insofar as they are clues to the meaning of the whole. Meaning is also determined by context. Thus words exert an influence over one another. We can also examine the structure of the sentence to analyze its influence on meaning. McClelland has developed a technique for learning natural language by presenting constituents of sentences to a connectionist network one at a time, taken from a small universe of vocabulary. (Constituents of a sentence might include agent, action, patient (a noun), instrument, location, co-agent and recipient. An example of the type of sentence that McClelland might feed to the network might be "the boy ate the soup with haste." Such sentences are fairly simple and could not adequately incorporate the

[127] See note to Chapter 1.

[128] McClelland, J.L., *Can Connectionist Models Discover the Structure of Natural Language? in Minds ,Brains and Computers*, Norwood, NJ., Ablex Publishing Co., 1992, Chapter 7, p.168-189.

metaphors, slang and context dependency that we encounter every day. However, using a limited vocabulary and a back-propogation rule, McClelland has found a degree of success in developing his model. However, he stresses that this neural computing application is likely to be most successful when used in conjunction and co-operation with traditional AI "compositional" methods. Early steps have proved promising using a PDP approach to "fill in" gaps that are missed by conventional symbolic approaches to natural language. The obvious question to ask of this research is - can they take the next step in their research and expand their model? As yet we have no answer.

3.3 CONNECTIONISM AND LEGAL OPEN TEXTURE

At present North American researchers lag behind their European counterparts in this area. This part of the thesis briefly presents various attempts by French, Dutch and English researchers to analyze the nature of open texture using neural computing.

3.3.1 BOCHEREAU ET AL.

The French trio of Bochereau, Bourcier and Bourgine[129] address an aspect of the domain of French case-based administrative law using a training set of 331 bye-laws and 47 bye-laws as the test data. In this highly technical paper, their purpose is to examine controversial mayoral decisions, by looking also at the legal domain, the use of legal norms and factual events to determine if the Council of State (administrative court)

[129] Bochereau, L., Bourcier, D., and Bourgine, P., *Extracting Legal Knowledge By Means of A Multi-Layered Neural Network Application to Municipal Jurisprudence*, , Proceedings of the Fourth International Conference on Artificial Intelligence and Law , Amsterdam, ACM Press, p.288 - 296.

decisions are correct in annulling or validating municipal laws that come before them. Their research is made all the more interesting because they attempted to perform the same tasks using a series of expert systems, written in PROLOG.

In their system, they use 49 input variables, divided into 4 subsets of regulations, bye-laws, factual standards and normative standards. Factual standards are those aspects of a case which involve an appreciation of circumstances. Normative standards include aspects such as Public order, Public Safety, Public Decency, etc. A mere 4 neurons were used in the hidden layer and there were 2 possible outputs - either annulment or confirmation of the bye-law. (Only 4 neurons were used in the hidden layer because of the few examples available.) Using the back-propogation method, they achieved a success rate when presenting new cases of around 80%.

They also achieved some success with extracting logical rules, although extracted rules such as those showing no relationship between "the standard "breach of public order" and "mayoral decisions in the field of traffic policy,[130]" would be dismissed as mere common sense reasoning and exposes a simplicity unlikely to be of value to policy makers and enforcers.

[130] ibid., p.293.

3.3.2 VAN OPDORP AND WALKER

Aside from providing one of the best explanations of neural networks for the uninitiated, Van Opdorp and Walker[131] offer an examination of the problems that one faces with various types of neural network architectures. They make the very valid point that simply listing the facts of a case may not be sufficient - some facts are more important than others. Therefore to work in open-textured domains, the system developer should involve an expert in the area who, although may not be able to describe their expertise in terms of rules, may possess a high degree of legal intuition and therefore know a bad example of the law when he or she sees it. Their example with regard to issue of open texture is as follows; "An apartment without an elevator is considered suitable depending on the age and degree of possible disability of the tenant." Thus, an expert in assigning sheltered housing would know if a given apartment is suitable for the tenant in question. Equally, with our domain of whiplash injury quantum assessment, an expert (whether lawyer or adjuster for ICBC) would know whether a given case is representative of how the domain works even though they may not be able to rationalize it in the form of rules. However, pure connectionists might argue that this conclusion undermines the value of neural computing because the great advantage of Connectionism over traditional AI was precisely that (ideally) one does not have to understand or articulate an understanding of the domain in order to build intelligent systems in that domain. I have not used an expert in the development of any of my neural networks except for a determination of which factors ought to be used as inputs and which should not be used at all.

[131] Van Opdorp,G.J., and Walker,R.F., ibid.

3.3.3 BENCH-CAPON

Ironically, we also find that Trevor Bench-Capon, a long-time proponent of the rule-based paradigm, is one of the few other research workers rising to the challenge of this issue of open texture. He sets himself three inter-related questions: "can a net classify cases successfully; can an acceptable rationale be uncovered by an examination of the net; and can we derive rules describing the problem from an examination of the net?[132]" i.e. acquire symbolic knowledge of open texture through a sub-symbolic methodology. Bench-Capon uses the fictional domain of welfare benefits paid to senior citizens to cover their costs of visiting close relations in a hospital. The domain envisaged, although showing some evidence of open texture, is governed by rules which could as easily be implemented using Boolean variables; e.g. A person should be of pensionable age, (English law still discriminates between the sexes on the issue of pensionable age; for women, the retirement age is 60, for men it is 65), have paid contributions in four out the five relevant years, should be the spouse of the patient, should have savings of no more than 3000GBP, if the spouse is an in-patient should live X miles from the hospital, if an out-patient, X + Y miles.

2400 test fictional cases were generated from a LISP program, consisting of 1200 cases where the randomly generated outcome satisfied all relevant conditions. Of the remaining 1200, each case would fail on *at least* one of the six conditions. He justifies the use of fictional rather than real-world cases on the grounds that "a real world domain would not

[132] Bench-Capon, T., *Neural Networks and Open Texture*, Proceedings of the Fourth International Conference on Artificial Intelligence and Law, Amsterdam, ACM Press, 1993, p.292-297.

provide the best test since, if *ex hypothesi* the domain is not well understood, there is no way of evaluating the rationale arrived at by the network.[133]" Bench-Capon constructed several neural networks, using one, two and three hidden layers. His results were that all three of the test networks converged producing success rates of 99.25%, 98.90% and 98.75% respectively. (Success rate means that a network discovers and settles into connection patterns that satisfy the relationships between input and output data.) He writes "This was a very encouraging level of performance...acceptable, even in a legal application" However, unlike Bochereau et al., Bench-Capon found that the "...analysis of the networks proved disappointing.[134]" The networks did not appear to appreciate the effect of age for men and women on entitlement; another condition - the distance/in-patient status seemed always assumed to be satisfied. He concludes "...an apparently acceptable level of performance can be achieved without any identification of some of the significant features of the problem.[135]" Thus he tried a second time, using data where, of the 1200 failing cases, only one condition failed. For one and two layer neural networks, he achieved a convergence in excess of 95%. This time the network appreciated that there was a discrimination between men and women of 5 years with regard to pensionable age, (although for some inexplicable reason, it concluded that women become senior citizens at 45 and men at 50.) The distance/patient status condition also appeared to be satisfied, providing a "picture...acceptably close to the truth and almost exact in the case of in-patients." What this experiment proves is that successful training is not necessarily indicative of the correctness of the underlying rationale. A neural network can come to

[133] ibid. at p.293.
[134] ibid. at p.294.
[135] ibid. at p.295.

the right answer for the wrong reasons. This is an important point for legal theorists; it confirms empirically what I argued earlier, namely that any belief that neural networks could somehow reveal the inherent "deep structure" of a given legal domain, should be dispelled.

As to whether we can derive rules from sub-symbolic architectures, he found some encouraging results (albeit combined with some significant deviations.) Thus Bench-Capon remains skeptical of our ability to infer rules from the results of a neural network, particularly if the domain involves complex combinations of factors that are perceived to have significance which will undoubtedly be the case within the domain of law.

3.4 RAGHUPATHI, LEVINE, BAPI AND SCHKADE.

This team of research workers (who are prolific advocates of Connectionism in law[136]) may be unfamiliar to many in the domain of computers and law. Their enthusiasm and belief in the task is refreshing; they write "In the long run, results from applications of neural processes to the legal domain will not only lead to a deeper understanding of fundamental legal decision processes but also enable study of the normative aspects of the legal system.[137]" Recalling what we concluded earlier in Chapter 3 in relation to the work

[136] See for example, Raghhupathi, W., Levine,D.S., Bapi, R.S., and Schkade,L.L., *Towards Connectionist Representation of Legal Knowledge*, in Levine,D.S, and Aparicio, M., *Neural Networks for Knowledge Representation and Inference*, Hillsdale, NJ., Lawrence Erlbaum, 1994, pp.267 - 282. See also Raghupathi, W., Schkade, L.L., Bapi, R.S., and Levine, D.S., *Exploring Connectionist approaches to Legal Decision Making*, Behavioral Science, 36, 1991, pp.133-139.

[137] Raghupathi,W., Schkade, L.L., Bapi, R.S., and Levine, D.S., *Exploring Connectionist Approaches to Legal Decision Making*, 36 Behavior Science, 2, April 1991, p.134.

of Warner[138], regarding legal reasoning being to a large extent about the selection of one option from a range of many, Raghupathi et al. wholeheartedly embrace and develop this view. In support they refer to work by Thagard[139] on explanatory coherence and MacLean[140] who developed the so-called triune theory of the brain. So far no other researchers have applied these theories. Thagard's connectionist network developed in LISP allows the user to compare and explain alternative hypotheses in the pure sciences as well as in law. Applying his network to factual and evidentiary issues of criminal cases, Thagard built a network called ECHO in which nodes represented hypotheses about a given case based on newspaper reports of a given trial. (The examples he addresses are murder cases in which there were no witnesses and juries had to infer on the basis of circumstantial evidence as to what actually happened.) His conclusion was that if there was an explanatory coherence between potential hypotheses, the nodes exhibited a excitement otherwise absent. Using such methodology, Thagard ran networks that could imply the guilt of a Defendant in a given case. It should be mentioned that Thagard does not propose to introduce this technology in any form within the legal discipline, at least not in its present state. He was merely using law as an example of how humans reason in parallel using competing hypotheses, choosing between two or more explanatory versions of events. One can imagine an extension of this so-called localist connectionist network to actual legal texts involving civil law rather than merely newspaper reports. The ECHO

[138] See note 114 in Chapter 2.

[139] Thagard, P., *Explanatory coherence*, in Behavioral and Brain Sciences, 12, 1989, pp.435-502.

[140] MacLean,P., *The triune brain, emotion and scientific bias*, In F.O.Schmitt (ed.) *The Neurosciences: Second study program*, New York, Rockerfeller University Press, 1970, pp.336-349.

program would appear to possess the ability to map out the analogical relationships that exist between a multitude of precedents and the case in issue.

3.5 THE TRIUNE THEORY OF THE BRAIN

MacLean's hypothesis is that the human brain may be divided into three layers, based largely on our evolutionary development. The first, "reptilian" brain focuses on the survival of the species through instinctive behaviour. The second, "old mammalian" brain is responsible for survival of the individual through emotions such as fear, love and anger. The third "new mammalian" brain is responsible for rational thoughts and strategies. The team of researchers led by Raghupathi maintain that this theory of the integrated instinctive, emotional and rational brain provides an appropriate conceptual framework for human thought processes. Thus reliance on pure rationality as a model to determine choice in legal reasoning or economic behaviour modeling is too shallow. It follows from this that any AI application in law that ignores the relevance of the instinctive and intuitive character of humankind will inevitably prove inadequate. Although developing a firm theoretical base, to date, Raghupathi et al. have yet to produce substantive models in support of their contentions with regard to Connectionism and law.

3.6 PHILIPPS

Another seemingly undaunted advocate of neural computing is Lothar Philipps[141]. He is of the belief that rather than supplying these decision-support tools to particular parties in

[141] See Philipps, L., *Distribution of Damages in Car Accidents Through The Use of Neural Networks*, 13, Cardoza Law Review, 1991, pp.987-1000, and *Are Legal Decisions Based on the Application of Rules or*

a dispute, they should be looked upon as modern science's "tool for justice, an especially subtle set of scales.[142]" The system that Philipps describes interestingly uses the domain of traffic accidents. However, he does not propose to develop a system designed to predict the value of quantum but rather its purpose is to assist the legal official in determining the percentage of the fault to be attributed to each party to the accident. He rightly points out "this calculation is not easy as fault does not have to be proportionate to the damage. The smallest mistake - may cause enormous damage. Thus the relationship between cause and effect is typically non-linear[143]." The data used is taken from a catalogue suggesting ratios for common types of accidents under German law. Some commentators accuse the judicial authors of these catalogues of being too rigid. Philipps argues that they can be made more flexible through the use of a neural network. Several networks are used; one for each type of accident. The inputs are ten different scenarios for traffic accidents involving A and B, such as "A changed lanes," "A turned into a parking space", "A changed lanes and B was driving extremely fast", etc. For each input, there is an output of a number between -1 and +1. -1 means that B pays everything; +1 means A pays everything. Very often the statutory statements read in isolation appear to be contradictory. For instance, there may be general law of the road which we shall call A, which has no exceptions, caveats or "loopholes." In the neural network, as in real world, a more specific rule, B, overrides such general rules. Why is this? " Philipps

Prototype Recognition? Legal Science on the Way to Neural Networks, 2 Pre-Proceedings of the Third International Conference on Logica, Informatica, Diretto , Florence, Martino, A.A. (ed.), 1989.

[142] Philipps, L., *Distribution of Damages in Car Accidents Through The Use of Neural Networks,* 13, Cardoza Law Review, 1991, p.990.

[143] Philipps, L., ibid., p.992.

argues that this is the way the law operates - "it is not a question of logic but of "psychology.""

Of most interest from a methodological point of view is Phillips' conclusion that the outputs varied according to the learning rate prescribed. The learning rate determines how large an adjustment the network algorithm will make to the connection strengths when it gets the output wrong. Philipps uses the useful metaphor of a game of golf. - you reach the target more quickly but may miss it several times. At different learning rates, he discovered outputs of .87, .97 and .61 for a given input statement. I performed similar experiments. The results are set out later in Chapter 4. In his experiments, different topologies did not provide any noticeable improvement in performance.

3.7 INFORMATION RETRIEVAL

One of the most pressing needs of the legal worker is quick access to a manageable number of relevant legal documents, whether cases, statutes or other secondary legislation. There are a number of existing methods of retrieving information using Boolean, vector space or probabilistic search methods. The most popular forms of information retrieval uses keywords with Boolean connectors. This is used by Quicklaw in Canada, Lexis in the UK, Australia and New Zealand and by West Publishing in the USA. However, there are serious deficiencies with full-text retrieval using single or complex Boolean search operators, such as AND, OR, and NOT, etc. One of the most important studies performed in this area[144] was performed on an IBM full text retrieval system, known as STAIRS. It revealed that its retrieval effectiveness was "surprisingly poor." The two most widely used measures of document-retrieval effectiveness are Recall and Precision. The study showed that "STAIRS could be used to retrieve only 20 percent of the relevant documents[145]" although the users believed that they were retrieving a far higher percentage. The study was not presented as a critique of the individual system but as a critique of the principles on which it and other full-text document-retrieval systems are based. The problem is succinctly summarized by Rau: "...words and lexical items in general are a poor approximation to meaning. Even with the addition of thesaurus or a phrasal lexicon, the approximation is not good...It is insufficient to treat words as indices irrespective of conceptual relationships between them.[146]"

Given that determining relevance is a gradable rather than binary operation and heavily reliant upon context, what are the alternatives? Systems which rely on manual extraction and coding of information are feasible up to a certain point. However, this approach would rapidly become uneconomic when dealing with large bodies of legal material. Thus, methods which rely on the automatic codification of text that has already been analyzed or is easily analyzable are to be preferred. Neural networks have features that could feasibly provide access to documentary information in new ways. This approach

[144] Blair,D.C, & Maron,M.E, *An Evaluation of Retrieval Effectiveness for a Full-Text Document Retrieval System*, Communications of the ACM, New York, 28,3, 1985, pp.289-299.
[145] Blair,D.C, & Maron,M.E, ibid., p.293.
[146] Rau, L.F., *Knowledge organization and access in a conceptual information system*, Information Processing and Management, 23, pp.269-283.

builds on views held by Bing[147] who argues that the words and lexical items by which documents are indexed should be weighted. In "normal" full text retrieval, all words, whether keywords or merely "noise" words are weighted similarly. One theory is that the relative frequency of the occurrence of words bears some relation to the meaning of a given paragraph. Thus words should "carry" a weight that reflects their importance.

3.8 NEURAL NETWORKS FOR INFORMATION RETRIEVAL

Both Belew's AIR (adaptive information retrieval) and Fernhout[148] adopt this weighted-word approach[149], the theory being that the number of times a given word appears reflects its importance in giving the document meaning. Thus the number of times a word appears in any given document is reflected in the strength of the connections between neurons. Automatic text analysis provides the neural network with the requisite numbers, showing how many times a word appears in a document and also how many times it appears in the document collection as a whole. This means that no training is required and that the system is usable very quickly. In large legal databases this approach may not be feasible as the network would take a very long time to train. Moreover, every time a new document is added there are major difficulties. Either the system must be re-initialized entailing a loss of the system's previous "learning" and a dilution of the strengths of the links between a

[147] Bing, J, *The law of the books and the files: possibilities and problems of legal information retrieval,* in G.P.V. Vandenburghe (ed.) *Advanced Topics in Law and Information Technology,* Hillsdale,NJ, Lawrence Erlbaum. 1990, pp.151-182.

[148] Fernhout, F., *Using a Parallel Distributed Processing Model As Part Of A Legal Expert System.* 1 Pre-Proceedings of the Third International Conference on Logica, Informatica, Diretto , Florence, Martino, A.A. (ed.), 1989, pp.255-268.

[149] Belew,R.K., *A Connectionist approach to conceptual information retrieval,* Proceedings of the First International Conference on Artificial Intelligence and Law, Boston, MA., 1987, pp.116-126.

word and a document, or the links remain unchanged by the addition of a new document. Neither of these approaches appear satisfactory.

An approach that seems to largely follow in the footsteps of the work of Belew, is that of Mital and Gedeon[150]. These apparently fearless advocates of Connectionism find fault with vector retrieval approaches because they allow for only one link between a word and a text unit. They argue "this (single link) does not deal with the indirect connectivities, such as are taken into account in a neural network based information retrieval system...In a neural network there can be many paths between a text unit and a word; some direct, some through other text units or words. It is postulated that this extra connectivity can discover and capture the semantic significance at a sub-symbolic level.[151]" The system they suggest is complicated. It involves the use of more than one link between any given word and document where that word appears in the document. The first type of links (known as Textual-Associative) are bi-directional and the weight assigned to the link is determined by a normalized word frequency. That is to say, extreme examples of this factor i.e. very high or very low word frequency measure are screened out as being of little value in discriminatory retrieval.

[150] Gedeon, T.D., and Mital, V., *Information Retrieval in Law Using a Neural Network Integrated With Hypertext*, Proceedings of the International Joint Conference on Neural Networks (ICJNN), Singapore, ACM Press, 1991, pp.1819-1824.

[151] Gedeon, T.D., and Mital, V., *Information Retrieval in Law Using a Neural Network Integrated With Hypertext*, Proceedings of the International Joint Conference on Neural Networks (ICJNN), Singapore, ACM Press, 1991, p.1820.

In addition they suggest the use of a second type of link called a "latent link." These also exist between every word and every document where the word occurs. These links are assigned an initial weight of zero. The strength of the "latent link" is derived from training during actual use of the system, rather than learning from a training set of materials. The latent link strengths are modified during use based upon user responses supplied by query and answer, gleaned through a hypertext-based user interface. Degrees of relevancy are monitored accordingly and reflected in the strength of the latent links. Thus situations where a judge makes an exclusionary statement (i.e. states "what is not in issue here are issues of privacy,") over time the system will be able to assign negative relevance to the term "privacy" and the given document, even though it is mentioned. Although interesting, one major criticism of the system must be that unless operated on an extremely powerful computer, the response time might be quite slow on a large database. The value of the empirical study must be questioned as the researchers tested their system on a mere 19 documents.

3.9 THE HYBRID APPROACH

In a later paper[152] Belew is joined by Daniel Rose to argue that both symbolic and sub-symbolic knowledge is embedded in the law. In order to demonstrate this thesis, they describe their system entitled SCALIR (Symbolic and Connectionist Approach to Legal Information Retrieval), a legal information retrieval system that combines sub-symbolic

[152] Rose, D.E., and Belew, K.B., *Legal Information Retrieval: A Hybrid Approach*, Proceedings of the Second International Conference on Artificial Intelligence and Law, Vancouver, ACM Press, 1989, p.138-146.

and symbolic paradigms. In this system two types of links between neurons are used which allow for both logical and associative queries and inferences to be drawn. Thus the user can ask for cases "like *A v.B*" as well as cases that include required keywords. The first type of link are called C-links which embody sub-symbolic data that is apparent to the system without human intervention and interpretation. Examples of this might be the fact that two cases are often retrieved together or that certain words frequently occur in the document. The second type of link (S-links) are labeled and unweighted symbolic representations of factual relationships, showing for instance, the dependencies between different sections of a given statute. It is arguable that the system described above might also work equally well in litigation support contexts.

3.10 NEURAL NETWORKS FOR DOCUMENT ASSEMBLY

Only one group of researchers, (again Mital and Gedeon[153]) have been so bold as to suggest that neural networks might also have a role to fulfill in the process of document assembly. Document assembly is the process by which lawyers develop documents from precedents with the aid of computer software. There are two main models of document assembly in use - template-based drafting and interactive assistance. The simplest form of template-based document assembly is the "merge" feature on any existing word-processing package, which can be used to formulate simple pleadings, tax forms and minor court forms. More advanced template-based drafting tools can create complex but

[153] Mital V., and Gedeon, T.D., *A Neural Network Integrated with Hypertext for Legal Document Assembly*, Proceedings of the Twenty-Fifth Hawaii International Conference on Systems Sciences, Hawaii, IEEE Computer Society Press, 1992, p.533-539.

relatively commonplace legal documents using so-called "boilerplate" clauses that require variation to suit the current purpose. Computerized interactive assistance in document assembly is desirable where a significant amount of legal knowledge is required. Commercial programs such as Blankety Blank[154] and Expertext[155] have used logic-based approaches to develop powerful tools for document assembly. Mital and Gedeons suggest that as an alternative to the use of templates or the use of logic-based tools, lawyers should leave the text in the precedent documents unaltered and instead the system should "simply allow the user to adapt the information contained in them to the current situation.[156]" They then go on to outline how their Textual-Associative and Latent links system could be used to augment hypertext to create a full-text retrieval system, the only difference being that the content of the database would be precedent documents as opposed to cases or statutes. This approach misses the point with regard to existing document assembly. The purpose of a document assembler is to lessen the amount of expertise and time taken in preparing first drafts of complex legal documents. The neural network that they suggest should be implemented does no more than retrieve the precedent in its skeletal form. Much time and effort would still be required to apply the document to its context. What Mital and Johnson suggest here is no more than a document delivery system. As we have seen, more document assemblers are used to remove some of the hindrances in drafting complex legal documents. For instance, neural networks do not offer users interactive assistance to ensure global consistency in terms of

[154] Produced by Softstream Technologies Inc., Hollywood, Florida, USA.
[155] Produced by Expertext of Toronto, Ontario, Canada.
[156] Mital V., and Gedeon, T.D., ibid., p.534.

structure and grammar of the document or allow younger, less experienced fee-earners to deal with matters that were previously the domain of more senior lawyers. Mital and Gedeon seem to be stretching the limits of the application of neural networks with this type of application.

3.11 CONCLUSION

We can summarize research to date on neural network applications in legal reasoning or legal "open texture" as follows; either researchers are skeptical of the ability of neural networks to rise to the tasks within the domain of law and the researchers' results tend to compound that skepticism, (e.g. Bench-Capon.) Alternatively, the researchers wholeheartedly embraces the theoretical perspective developed from philosophy and cognitive science, that artificial neural networks are likely to prove more successful at tasks which have proved insurmountable for traditional AI. However they have yet to develop results that confirm their suspicions. This thesis falls firmly in the camp of the latter. The task of this thesis is develop a neural network that is successful, to develop results that confirm my suspicion that neural networks are as good as other forms of computer programming in certain specified tasks within the domain of law. However, an important caveat should be entered at this point. Of the 5 networks relating to legal reasoning (as opposed to indirectly related legal tasks such as document assembly and information retrieval,) only 2 used real world data. Of these two, neither is as complex in terms of number of input neurons or connections, as those networks developed by myself and described in Chapter 4.

With regard to present applications of neural networks in law, it would appear that neural network technology is finding immediate and successful usage in information retrieval. Regardless of whether they can out-perform traditional statistical computing techniques at the task of predicting quantum, they have some immediate uses and we are likely to see more of this type of technology.

CHAPTER 4

THE WHIPLASH NEURAL NETWORK

4.1 INTRODUCTION

The domain in issue is the assessment of awards for whiplash injuries in British Columbia. After briefly identifying the role of the insurance company and the nature of a whiplash injury and how an assessment is made by the insurance company, this chapter details the attempts made to build a neural network using the legal domain of personal injuries and in particular utilizing the data contained in the "Whiplash database."

4.2 THE ROLE OF ICBC.

To non-residents of British Columbia (B.C.), motor vehicle insurance arrangements within the Province are slightly unusual. All motor vehicle insurance is provided and administered through the Insurance Corporation of British Columbia (ICBC). Set up in 1973 by the Insurance Corporation of British Columbia Act, (R.S.B.C., Chapter 201,) ICBC is a Crown agency, a non-profit organization and monopoly provider of basic liability and accident benefit coverage to all owners of vehicles licensed in the Province. The rationale behind this arrangement is that having a single licensing and insuring agent bring with it numerous advantages. First, ICBC can almost guarantee the population of B.C. that every resident in the Province has motor insurance. (It is estimated that there are between 250,000 and 1 million uninsured vehicles being driven on the roads of Ontario[157].)

[157] *Comparative Aspects of the Insurance Corporation of British Columbia and Private Sector Insurance Companies in Canada*, ICBC, Vancouver, B.C., 1990, p.6.

Secondly, all vehicle owners contribute to the Autoplan fund out of which claims are paid. The reasoning behind this goal is that no single group of road-users should feel treated harshly. According to ICBC, premiums appear to compare favourably with the private sector (although admittedly the comparative figures published are usually only provided by ICBC.) Premium increases appear to be minimal despite Vancouver drivers reputedly having a poor driving record[158]. This situation of one insurance company for the entire province leads to unusual circumstances whereby ICBC acts for both sides in a dispute. To many, it appears that there is a conflict of interests.

The motor vehicle insurance industry in B.C. is immense. ICBC employs nearly 4000 staff, has 900 brokers and in 1993 wrote over $2 billion worth of vehicle insurance. From that fund, in the fiscal year 1993, a total of 804,115 claims were made and $1,021,000,000 was paid out in injury related claims[159]. In 1989, there were 587 fatalities in road accidents and 47,471 people injured[160]. According to the Ministry of the Solicitor General, Motor Vehicle Branch of B.C., that averages an accident every 3.63 minutes and an injury in an accident every 11 minutes. Needless to say, the task of assessing the cost of those accidents falls on the shoulders of an army of ICBC adjusters. Should the matter turn litigious, the final arbiter of the matter will be a judge.

[158] An article in the Report on Business magazine, (June 1989) provided statistics compared 27 accidents per 1,000 people to 36 in Montreal and 70 in Vancouver "which has the worst drivers in the country."
[159] Insurance Corporation of British Columbia, *ICBC Annual Report*, Vancouver, B.C., 1993, p.5.
[160] Ministry of the Solicitor General, Motor Vehicle Branch, *Motor Vehicle Statistics- 1989*, 1989, Victoria, B.C.

4.3 WHAT IS WHIPLASH?

Whiplash is one of the commonly diagnosed syndromes in patients who have been involved in motor vehicle accidents. The term was first used in a medical paper in 1928[161]. Since that time it has proved to be an ambiguous condition which although common, has not been thoroughly researched by the medical profession. Part of the reason for the lack of interest in the condition is what is described by Croft as "its rather unsavoury reputation, owing to its common association with litigation, discouraging most researchers from delving into this "Pandora's box.[162]"" The term covers a wide variety of complaints such as neck pains, blurred vision, head-aches, dizziness, nausea, back pain and numbness. It may also result in surface hemorrhages of the brain and cerebral concussion. Whiplash may be combined with other injuries such as fractured or broken bones or skull, and crushed vertebrae.

When a car is struck from behind, the torso of the occupant is accelerated whilst the head is snapped backwards. It may also be complicated by other movement of the head i.e. from side to side or the head may be at an angle when the injury occurs. As the force of the collision dissipates the head then snaps forward and is forced onto the chest. If the car strikes any further objects such as a car in front, then the injuries may be exacerbated. However, whiplash is also an extremely easy injury to fake and patients who complain of any of the above symptoms after a car accident are often perceived by the medical

[161] Crowe,H.E., *Injuries to the Cervical Spine*, paper presented at the meeting of the Western Orthopedic Association, San Francisco, 1928.
[162] Foreman,S.M., & Croft, A.C., *Whiplash Injuries - The Cervical Acceleration/Deceleration Syndrome*, 1988, Wilkins & Wilkins, Baltimore.

profession as either malingerers or hypochondriacs. This argument is, according to certain Plaintiffs' Counsel[163], one commonly made by Defense attorneys.

4.4 HOW ARE DAMAGES ASSESSED?

An adjuster is the representative of the insurer (i.e. ICBC) who seeks to determine the extent of the ICBC's liability for loss when a claim is submitted. There are three main types of investigation - coverage i.e. determining under what head the loss falls; wage loss investigation; and medical investigation - the so-called core from which all damages flow. These three types of investigation provide the adjuster with a quantum figure to offer to the Plaintiff. If the figure offered by the adjuster is thought by the Plaintiff to be too low, the process of obtaining an award will become adversarial as the Plaintiff is required to seek independent legal advice. It will be the job of Plaintiff's Counsel to present the clients case in the most comprehensive manner, consistent with the truth, by emphasizing the nature and duration of the symptoms, the progress and present status of the patient. Defense Counsel who will represent ICBC will focus on those factors which adversely affect the recovery process and elucidate on the degree to which the symptoms have been resolved.

4.5 THE INITIAL ASSESSMENT

The job of making an initial assessment is given to an ICBC-employed adjuster. Employees of ICBC can spend their entire careers assessing injury. Therefore, many of them are very skilled in the art. However, a junior adjuster begins to develop experience

[163] Keith and Laura Magee of McGee and Company, Vancouver, B.C., a firm of lawyers that specializes in Plaintiff personal injury.

of the value of a whiplash injury by using what Plaintiffs' Counsel commonly describe as "the meat chart", a table of values developed internally by ICBC showing what figure they believe an adjuster should settle with in any given case. From this, the adjuster goes on to develop experience of likely quantum figures. How then does an adjuster arrive at a figure? Does she/he use a rule-based mechanism, does she/he simply know a $30,000 case when she/he see one or is there some other complex reasoning system at work? (At this point we should have in mind the arguments developed by the Dreyfus brothers, mentioned in Chapter 2.) Evidence gained from interviewing Plaintiffs' Counsel[164] suggests that lawyers use a combination of the above methods, although the balance between them depends on the case. It was suggested to me that one must also have a feeling for the social, economic and political environment in which legal actors operate. These factors are more thoroughly explored in the conclusion. But how does a judge reason? We have very little evidence because judges are generally very unwilling to permit third parties to scrutinize their practical reasoning processes, partly for fear that it might permit Counsel or other legal actors to "second-guess" them.

4.6 THE STATISTICAL QUANTUM ADVISOR

This is the system against which the whiplash neural network is to be judged. Members of the FLAIR project have been quick to recognize the importance of an automated quantum estimator which would work in conjunction with an information retrieval mechanism to retrieve relevant cases. Although the software developed as part of this thesis does not

[164] Keith Magee, see note 163.

include an information retrieval mechanism it is clear from the review of neural computing applications in law, in Chapter 3 that the potential also exists for a neurally-inspired information retrieval mechanism. Using a statistically-based model and implemented in the C programming language under DOS, database parameters are used to assess the weight given to each field in the database and the relationships that may exist between them. To ensure fairness, the Whiplash neural network uses identical original data.

The Statistical Quantum Advisor is methodologically problematic. The neural computing advocates would point out that case law has developed organically. How then, can a rigid mathematical formula that is no more than the summation of one person's belief as to how quantum is assessed, possibly capture the seeming complexity of the domain? On the face of it, the neural network ought to out-perform the Statistical Quantum Advisor for two reasons; First, the neural network develops its conclusions "organically" by trial and error; Secondly, the data is untouched - it does not involve any human interaction or interpretation, before entry into the computer. This is not to say that a computer's reasoning is inherently superior to that of a human being. Indeed, the experience of AI over the last forty years suggests quite the reverse. However, neural networks develop their own interpretation of data. Thus there is one less level in which interpretational mistakes can be made. The neural skeptics would counter such arguments by pointing out that law is a hermeneutic science - its very essence is interpretation. To argue that the involvement of human reasoning in some way infects the pure body of reason that is the law is to misconceive the nature of law. Rather than adopting one side or the other of such a debate, we should focus on the empirical data, such as that described in this Chapter.

4.7 METHODOLOGY

Having briefly discussed the context, in this part of the Chapter, I outline how the data in the Whiplash database was manipulated, data problems and the process of building a neural network.

There are six stages to the design process of building an artificial neural network. These are as follows;

- Decide what it is that the Network is required to do. The project that incorporates the practical aspect of this thesis is the development of an artificial neural network that attempts to predict quantum in a given case involving whiplash injuries.
- Decide what information is to be used. The prediction that is made by the ANN is based on the relationship(s) that exist between various categories of data (the input) and the quantum figure in previous reported cases of whiplash (the output).
- Get the data - The Whiplash Database
- Build a network.
- Train it
- Test it.

4.7.1. TEST2.PRG

The first task was to decide which cases could be used to train the ANN. This was discussed with John McClean, a programmer with FLAIR and a qualified lawyer in B.C.

who has significant experience in dealing with personal injury cases involving whiplash injuries. With the assistance of Keith MacCrimmon, who is also a programmer at FLAIR, was able to build a short program using a BASIC-like programming language in dBASE™ that was able to strip out all of the unwanted cases and provide myself with a relatively clean data set. A copy of the code is attached, with comments at Appendix A (entitled Test2.Prg.) Having removed all cases that mentioned any of the above factors I was left with a data set of approximately 660 cases, from an original 1661 consisting of approximately 60 fields which would be used as inputs. While my experiments continued more cases were added to the database, which in turn increased the set of useful cases to 705.

Within the existing whiplash database there are 115 fields[165] Many of these fields were irrelevant for my purposes. (For instance, whether FLAIR has a hard copy of the case in its records is of no consequence to the purpose of predicting quantum.) After consultation it was agreed that a total of 54 fields should be deleted. These included, inter alia; Court registry location; File number; Judge's title; B.C. decision citation; Pain and suffering award - (a character string, sometimes mentioned in the reports); Pain and suffering award - a numeric range; Case comments 1-3 (this were fields in which descriptions of symptoms could be included e.g. "The Plaintiff has felt slight neck pains on and off..."); Whether legal issues of proof or remoteness were mentioned in the case; Whether the case included cross references to other cases in the database; Whether there

[165] A database field is defined in the dBASE manual as "An individual data item in a record." See Ashton-Tate, *Learning and Using dBASE III Plus*, 1987, p.G-8.

was a hard copy of the case in FLAIR's records, etc. Some of these fields such as file number were clearly irrelevant to the overall award. Others, such as age and sex of the Plaintiff were considered by myself to be material. However, on advice from John McClean, they were also deleted.

Other database fields were used to act as filters i.e. if the case included any of the factors mentioned below[166], the case was skipped because it was felt that the information would initially "confuse" the network. Other factors used as to "filter out" potentially difficult cases were; If the case involved a Jury trial - a rare event which is believed by lawyers to "buck the trend" that is believed to exist in Whiplash damages awards. At one time, it was a commonly-held belief amongst lawyers that juries tended to be far more sympathetic to the ailments of the Plaintiff than case-hardened judges. (It now appear that this trend has reversed. Indeed according to Plaintiffs Counsel[167], ICBC now look for opportunities to use jury trials because they believe that, in these times of harsh car insurance premiums, juries are proving to be less sympathetic than the judges. In any event, jury trials tend to upset the accepted trend of awards whatever it may be.) A full list of those facts which were used as inputs is attached at Appendix B.

[166] For instance, if the database record mentioned that the case made mention of the fact that different types of damages were included in the non-pecuniary award (such as economic loss); if there were multiple accidents (i.e. if there were more than two parties involved in the accident that would complicate the award because some damages would be awarded against one party and some against another); if any mention was made of the notion of the "thin skull' concept; if there were any pre-existing injuries. If any of these factors were mentioned the case was not used for training purposes.

[167] Keith Magee, see note 163

4.7.2 DATA PROBLEMS

The problems with the database were many. For instance, the whiplash database has evolved and continues to evolve over time. Many people have been involved in inputting the data and we have no guarantee that they did not interpret cases in different ways from one another. Despite the fact that the fields contain largely numerical data, the human data processors are required to exercise a degree of discretion. Also new fields and codes have emerged with use.

The first problem encountered was that very often certain field had no entries. For instance, sometimes the date of the injury would not be mentioned in the judgment. However, the processors has only the written case judgment to work with. Therefore we do not possess a full account of all the evidence presented at trial. We have only a truncated version. The question was - what should one do in order to maintain the accuracy and scientific legitimacy of this experiment? Was it safe to presume that if a particular matter (such as a visit to a physiotherapist) was not mentioned in the database record that it had not been raised at trial? There are a number of possible scenarios. There may have been one or more visits to a physiotherapist, but the lawyer for the Plaintiff omitted to mention the matter for one of many reasons, the most likely scenario being that the lawyer decided that the judge was not one who takes much account of visits to physiotherapists. Alternatively the judge, despite hearing evidence of visits to physiotherapists and such like, might have decided that this was indeed irrelevant to the overall award. A final alternative is that the judge thought it was important but omitted to mention it from the judgment. We have no way of knowing without going back to the original trial transcript and accompanying documentation for each case record. Thus a pragmatic decision was taken that if a particular factor was not mentioned then it should be assumed that that factor was not present. When an entry in a field was considered to be important, the hard copy of the case was consulted for clues to the missing data. For instance the date of the injury was often omitted. From a perusal of those cases, it was concluded that it would be safe to imagine a delay of no more than 1 year from injury to trial. In order to avoid possible data errors made through intelligent guesswork, as a rule, those fields were not used.

Researchers on neural networks have made numerous claims about the ability of neural networks to work with incomplete data. This was a golden opportunity to test such assertions. Although these issues would be highly problematic if one wished to build a system for commercial exploitation by ICBC or B.C. lawyers, for the purposes of comparison, these issues are irrelevant, because the Statistical Quantum Advisor also worked with the same incomplete data.

4.7.3 DECEPTIVE DATA

In order to fine tune the network further, I felt that it was necessary at least initially to take account of inflation and therefore the year in which the judgment was handed down. All dates in the database were in a form "--/--/--" which is clearly unusable. Days and months were less important to the experiment. However year figures could provide a rough estimator of the inflation. However, dates can be deceptive data for neural networks. There are two ways of interpreting them in databases. We might wish to use dates in a way so that 1986 is no more important than 1985 or less important than 1987. That is to say that each is a unique year. They merely need to be distinguished from one another. Therefore to place all dates in the same column of data would be a mistake. However, in the discipline of law, dates are important in deciding whether a given case is still precedent - e.g. depending on the nature of the dispute, in a given case one party might argue that because a case is from, say 1926, it is too old to be relevant. Also, given two factually similar cases, each with different outcomes but one decided in 1985 and the other in 1994, the fact that one is more recent than the other is an important consideration.

If we take the view that each year is different rather than more or less important than another, there are two solutions. The first is to rearrange the data so that each year or groups of years have their own columns and given either a positive (1) or negative (0) value. The second is to convert the number into a symbol. By executing a Brainmaker command one can make unique neurons of each different number in the same column of data. If however, we take the view, and we do here, that a more recent case is a more valuable precedent than an older case, then we can place all the years in the same column so that the network interprets 1986 as a number more important than 1985 by an increment of 1. This view was adopted and utilized in the Test2.prg program. However, early experiments with the neural network software did not use this data.

4.8 THE SOFTWARE

The Brainmaker ANN software is one of the most popular neural network applications for the PC available[168]. Used throughout this project, it comes with a built-in database entitled Netmaker that, amongst other things, allows the user to identify which columns of data are to be used as inputs and outputs. This facilitates the quick creation of files which are used by the neural network software.

4.8.1 HISTOGRAMS

Brainmaker includes a number of tools that allow the user to test how well the network is training as the network trains. These tools are known as the Network Progress Display (NPD) and the Column Histogram. The NPD consists of two graphs.

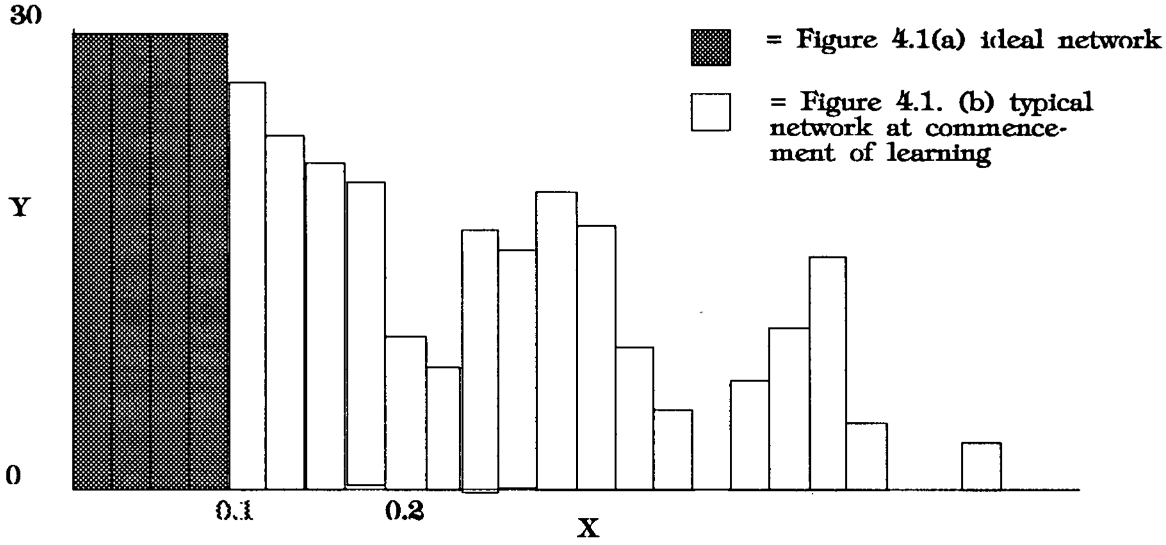
The first graph is described as "a histogram of the errors over the whole (single training) run." According to the software manual "the error for each neuron is the absolute value of the difference between the pattern (expected value) and the output (actual) value. The error display is a histogram where the height of each bar corresponds to the number of error values within the range of the bar[169]." That is to say that the horizontal axis is the error level and the vertical axis is the number of outputs that were within an acceptable range. Thus the ideal NPD diagram shows columns bunched together on the left side of

[168] According to the manufacturers, California Scientific Software, by September 1994, over 18,000 copies of the software have been sold.

[169] California Scientific Software, *Brainmaker User Guide and Reference Manual*, California Scientific Software, Nevada City, CA., 1993, p.10-34.

the Histogram as shown by Figure 4.1(a) overleaf where all the error values are within a tolerance of less than 0.1. Graph 4.1(b) shows a typical graph when the network commences training with error values up to 0.5 (i.e. 50%) off target. As training progresses, the bars gradually move leftwards across the graph.

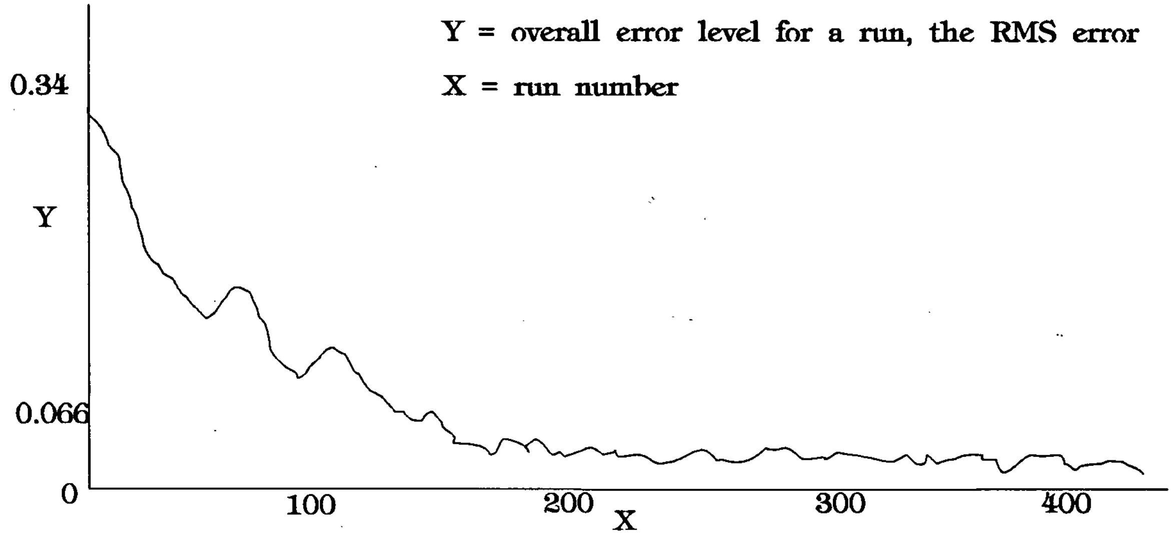
Figure 4.1 (a) & (b) - Network Progress Display Graph 1



X = error level, Y = number of outputs within error range.

The second NPD graph below is a display of the progress of the RMS (Root Mean Squared) error. According to the software manual, "the RMS error is much like an average error In a healthy learning network, this value should decrease from one run to the next. A good training progression is represented by an overall downwards trend from left to right. A network that is not learning as easily would show a more gradually declining curve or a curve that showed occasional increases in value.

Figure 4.2 Network Progress Display Graph 2



4.8.2 TRAINING, TESTING AND RUNNING

Once the user has decided which data fields to use as inputs and outputs, Brainmaker automatically divides the database in question into a set of training records (by default, set at 90% of the total database, written to ****.fct) and a set of testing records (the remaining 10%, written to ****.tst.) Every tenth case is placed in the Testing file automatically. Brainmaker takes every 6th, 16th, 26th case, etc.

Training is the process by which the input/output pairs are delivered through the network repeatedly. The learning algorithm (the back-propogation rule mentioned in Chapter 1) corrects the network until the tolerance parameters for all facts in the training set are satisfied. Testing is the process where those 10% of the cases set aside are put through the network. This allows the user to track results without correcting the network. It is important to note that for a fact to be considered "good" it must be a figure within the Tolerance band (i.e. 10% if the Tolerance is set to 0.1) of the maximum figure within the output set. Thus, using the Whiplash database which has a maximum quantum figure of $100,000 (in the case of Adachi v. Westhaver) the required output must be within $10,000 (10% of $100,000) of the actual output in order to be considered a "good" fact. Tolerance is defined as a percentage of the range of output.

Testing should be distinguished from running. Running is the process whereby the user presents new, unseen data to the network and requests an output. Of the many features provided by Brainmaker, only training and running are required although, for a developer not to test the trained network before presenting it with new data would be extreme folly. The testing tolerance is set by default at 0.4 (i.e. 40%) Thus clearly unacceptable results such as an output of $28,000 against a required pattern output of $14,000 would be considered a "good" fact by the network because it was within $40,000 (40% of $100,000) of the required output.

4.8.3 SEVERITY RATING - AN EXPLANATION

My first idea was to have two outputs - Quantum and the Severity Rating. This field is a numerical factor used in the database to identify how the judge perceives the case before him/her in terms of severity i.e. if the judge says in his/her summary - "this is a mild case of whiplash," this would merit a "1" in the appropriate database field , whereas if the injuries are described as severe, this would merit a "6." Clearly this field cannot be used as an input as it is a finding of the judge. However, I felt it would be both interesting and helpful to see if the network could distinguish severe from mild whiplash injuries based on the 60 or so fields of data used as inputs.

4.8.4 THE FIRST NEURAL NETWORK

The first neural network failed to yield any results of value. Randomness is the kindest adjective one could use. After 40 hours of training during which time the network failed to finish training on the training set , the network had still to distinguish the most severe from the mildest cases of whiplash. The results appeared to be very arbitrary indeed. However, the network appeared to have made more progress with the Quantum factor. Thus I felt it would be more prudent to not bother with the Severity Rating as a conclusion until I had crossed the first and more important hurdle of predicting quantum.

A cursory glance at the 30-40 new cases that were not used for either training or testing of the neural network gave some clues as to why attempting to predict the Severity Rating was problematic if not almost impossible. For instance, in the case of Zizic v. Rinta (1) (BCSC), Judge Boyd describes the whiplash injury sustained as moderate to severe and yet awards damages of $10,000 whereas only four months earlier, Judge Hunter awarded $25,000 for a "moderate" case of whiplash in Stewart v. Mitchell. Equally paradoxical is the earlier case of Dhut v. Dhaliwal (1987) (unreported) who, for a whiplash injury which merited a "6" on the Severity Rating scale, received a measly $850. However, perusal of the hard copy of the case reveals that at the time of entry into the database, "6" referred to an "unclassified" rating and in fact the range within that field numbered from 1 to 5. Small wonder therefore that the neural network appeared to show a degree of what the manufacturers describe as simulated "brain damage. But what of neural networks' famed ability to cope with conflicting data? It should be said that the network did settle into a pattern, but in order to accommodate the data, the network had to be generalize so much as to be worthless. This type of contradictory information also suggested at least one reason why the rule-based approach to Whiplash quantum assessment was doomed to failure from the outset.

4.8.5 THE NEXT EIGHT NEURAL NETWORKS

Having decided not to use the Severity Rating as a Pattern Output[170], Network #2 trained comparatively well. The tolerance factor was set fairly high (0.2) so that the network was more likely to form some kind of conclusion regarding the relationships between inputs and outputs. This policy succeeded. By the 222nd run of the 595 training records, some 572 were "good" and a mere 23 were "bad." However testing revealed large errors between expected and actual outputs.

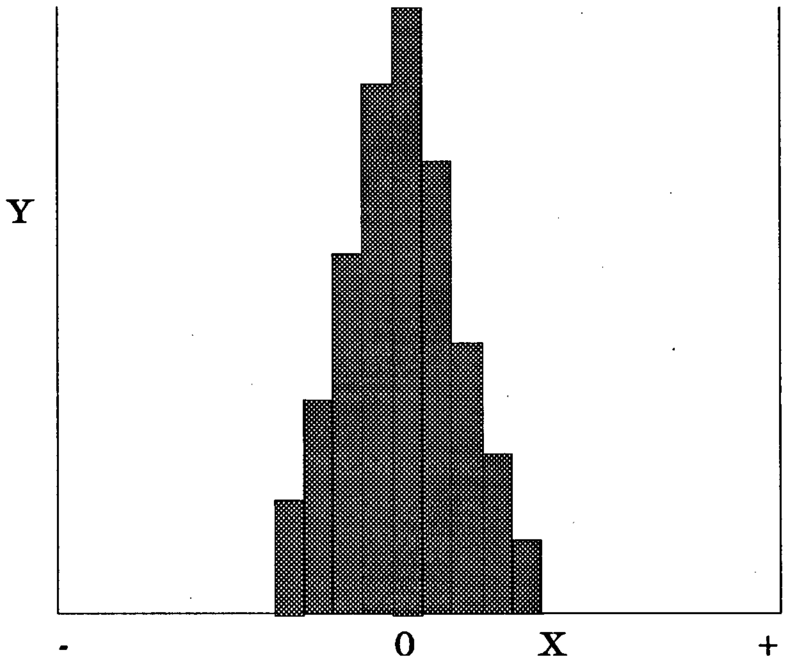
I wanted to stop the training early, (i.e. within the first few hundred epochs/data runs) because writers in this area, (and in particular, the manufacturers of the Brainmaker software) warn against creating a network, which whilst satisfying the tolerance criterion, fails to generalize well; i.e. it performs well in training but tests poorly. What I wanted to develop was a network that could generalize extremely well even though it might not train correctly on every given training fact.

[170] See Appendix G - Glossary of Brainmaker Terms.

How does one know whether the neural network is "learning" properly? The software possesses a further graphing capacity which allows the user to check on the mental health of the network. A healthy network with a capacity to learn is typified by a histogram resembling a "New York skyline" as shown in Figure 4.3, where most weights are clustered around zero.

Figure 4.3 A "healthy" learning neural network

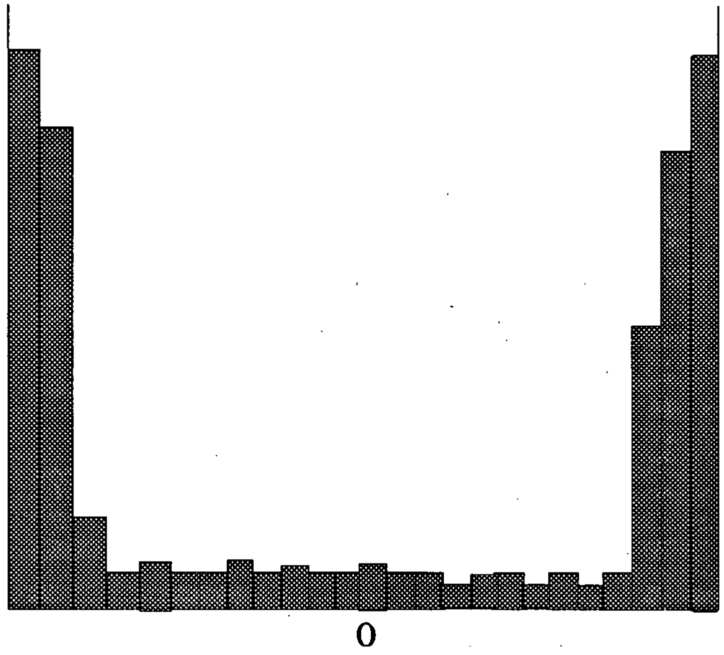
(Y = Number of Neuron Weights; X = Weight Values)



A network with a histogram as shown in Chart 4.4 below (which might be likened to a U-shaped valley) reflects a "brain-dead" network; that is to say, a network that has memorized rather than learnt from the facts, with connection weights piled at both extremes.

Figure 4.4 A "Brain-Dead" neural network

(Y = Number of Neuron Weights; X = Neuron Weights.)



Networks #3 and #4 were left to run for many hours as a test of the performance degradation curve. Early examination of the Histograms showed a healthy learning graph but later results e.g. runs 200-500 suggested a "brain-dead" network that had "memorized" rather than learnt the required results. From this I learnt that although the amount of time taken for the network to learn was not important, but the number of runs was; Using this data, on over 200 runs it seemed that the network could learn nothing further from the data. I found this result quite surprising given that it was so complex. However given the results it seemed likely that the network that provided the best generalization also would provide the best predictions. The Network Progress Display (NPD) curves suggested that the 1st 20-40 runs of the training data removed the majority of the errors. The next 80 removed more and the next 100-150 were fine tuning. Therefore, on this data, the network developed in the 100th-200th run seemed most likely to provide the network that would satisfies our requirements in terms of the most accurate predictions. This seemed very surprising indeed as neural networks often can take days to learn the required patterns in order to produce satisfactory outputs. I wanted to test the network by training it on thousands of runs of the data. Brainmaker would take days to perform such a task. Therefore I resorted to an alternative neural network written in C which ran on the SUN computer. The results relating to this experiment are provided further on in this chapter.

The results of Network #5 were invalidated because of my own human error. The column entitled Severity rating was included as a input. This is an inappropriate use of such data because this field reflects a judicial conclusion and, if anything should be used as an output.

4.9 HOW MANY HIDDEN NEURONS?

Knowing how many hidden neurons to use is part of the art of developing a good neural network. It seems that there is no definitive answer as the number of neurons is application-dependent. As a general rule, if one uses more facts and the fewer hidden neurons, the network will “generalize” more easily. However, this rule does not always hold true. Very often it is possible to “prune” out superfluous hidden neurons (or adding neurons) to improve performance without losing previous training. Literature on neural networks provided me with some guidance. For instance one idea mentioned by the developers of Brainmaker recommends the following formula;

No. of hidden neurons = (inputs + outputs) /2

In this scenario, with 60+ inputs and 1 output this means in the region of 30+ hidden neurons. An alternative rule of thumb is that the number of hidden neurons should be equal to the error tolerance times the number of training facts. However this would be equal to 60 hidden neurons - almost as many inputs. Other literature[171] suggested that the upper bound for the number of hidden neurons can be determined by the number of training examples according to the rule - five examples for each weight. We have nearly 600 examples therefore we easily satisfy this criterion. Early experiments were made with the number of hidden neurons around the 30 mark.

Network #6 used 27 hidden neurons in the middle layer. After 196 runs of the data, it successfully trained on a Tolerance of 0.1. Network #7 used 37 and trained successfully after 340 runs of the data. Network #8 used 17 neurons and trained even faster with a mere 102 runs of the data. All other factors remained the same. The makers of Brainmaker warn the user that reducing the numbers of hidden neurons can impair the networks performance. Thus whilst network #8 trained very quickly it did not test at all well. Each network was tested by taking random sample results from the 66-case testing file which Brainmaker generates automatically. The results are set out in Figure 4.5 below and the most accurate output for each fact is marked with a star (*);

[171] For example Klimasauskas, C., *Applying Neural Networks,* PC AI, May-June 1991, pp.20- 24.

Figure 4.5 Results of experiments with varying number of hidden neurons

CASE NUMBER	REQUIRED OUTPUT	NETWORK #6	NETWORK #7	NETWORK #8
		(27)	(37)	(17)
9	$8,000	$10,165*	$10,311	$11,019
19	$18,000	$11,532*	$34,945	$24,594
29	$15,000	$28,329	$16,073*	$25,570
39	$12,000	$11,679*	$17,147	$24,374
49	$7,500	$8,334*	$12,606	$14,926
59	$2,500	$5,599*	$9,359	$8,749
66	$30,000	$11,434	$16,195	$36,410*

Given that all of the above three networks showed significant errors in testing, I decided to run a separate trial of networks using similar data, with various numbers of hidden neurons but concentrating purely on the training statistics that the network made available to me. I used networks with 60, 50, 40, 30 20 and 10 neurons respectively in the hidden layer and trained them over 100 runs of the data. Tables showing the number of "good" and "bad" runs together with the RMS error are provided below. A network that trains perfectly would attain 598 "good" facts and 0 "bad" facts in the shortest possible time. It should also provide an RMS error as close to zero as possible which would be represented graphically as a steep declining curve from left to right as in Figure 4.2. However, we still need to bear in mind that the network that trains the best does not necessarily also test the best. As Lawrence points out "a network that can't get all the training examples correct may still test well and be more useful than the one that trains well[172]"

[172] Lawrence, J., ibid., p.205.

Figure 4.6 Results of further experiments varying the number of hidden neurons

60 Hidden Neurons, Tolerance of 0.11

RUN	GOOD FACTS	BAD FACTS	RMS ERROR
10	498	100	0.0774
20	541	57	0.0668
30	556	42	0.0627
40	564	34	0.0594
50	571	27	0.0564
60	567	31	0.0568
70	557	41	0.0624
80	574	24	0.0564
90	564	34	0.0626
100	558	40	0.0627

50 Hidden Neurons, Tolerance of 0.11

RUN	GOOD FACTS	BAD FACTS	RMS ERROR
10	513	85	0.0744
20	547	51	0.0667
30	549	49	0.0656
40	561	37	0.0614
50	554	44	0.0642
60	563	35	0.0627
70	560	38	0.0626
80	574	24	0.0585
90	568	30	0.0606
100	558	40	0.0609

40 Hidden Neurons, Tolerance of 0.11

RUN	GOOD FACTS	BAD FACTS	RMS ERROR
10	533	65	0.0718
20	546	52	0.0641
30	561	37	0.0622
40	561	37	0.0627
50	558	40	0.0639
60	567	31	0.0615
70	563	35	0.0621
80	561	37	0.0618
90	557	41	0.0604
100	594	4	0.0519

Figure 4.6 (continued)

30 Hidden Neurons, Tolerance of 0.11

RUN	GOOD FACTS	BAD FACTS	RMS ERROR
10	524	74	0.0725
20	547	51	0.0626
30	555	43	0.0650
40	556	42	0.0646
50	557	41	0.0633
60	551	47	0.0649
70	556	42	0.0647
80	555	43	0.0635
90	547	51	0.0648
100	560	38	0.0652

20 Hidden Neurons, Tolerance of 0.11

RUN	GOOD FACTS	BAD FACTS	RMS ERROR
10	524	74	0.0769
20	542	56	0.0679
30	562	36	0.0521
40	554	44	0.0516
50	556	42	0.0648
60	552	46	0.0628
70	559	39	0.0626
80	557	41	0.0625
90	548	50	0.0631
100	564	34	0.0600

10 Hidden Neurons, Tolerance of 0.11

RUN	GOOD FACTS	BAD FACTS	RMS ERROR
10	515	83	0.0820
20	524	74	0.0759
30	528	70	0.0718
40	538	60	0.0693
50	545	53	0.0692
60	542	56	0.0702
70	540	58	0.0689
80	548	50	0.0678
90	551	47	0.0682
100	562	36	0.0669

From the above tables we can see that networks with extreme numbers of hidden neurons do not train as well as those with neurons with 20-40 neurons. The network with 60 hidden neurons trained erratically as did the network with 10 hidden neurons. One of the most significant training results provided by this experiment was the severe dip in the number of "bad" facts in the 40-neuron network on the 100th run of the data set. The RMS error also dropped dramatically at the same time from 0.0604 to 0.0519. I took this change as an extremely good omen and decided that in future experimental networks, the number of hidden neurons would be set to 40. (It should be pointed out that this type of experiment is a fine-tuning device, best done when a moderately useful network has been developed. Thus I return to this experiment later in the Chapter.)

How many hidden layers? The rule would appear to be "more is not better.[173]" According to the manufacturers of the software used in this set of experiments, adding another layer of neurons offers no guarantee that the network will find a solution where a network with a single layer has failed. Also the network takes much longer to train as the errors have to filter across a further layer of neurons. Therefore I did not perform any experiments where I changed the number of layers of neurons.

4.10 EXTREME DATA REMOVAL

The distribution of data can drastically affect the networks ability to learn. Performance of the network is likely to improve if the network is forced to look more at the common values. Whilst I was aware that the neural network needs a large training set of data with

[173] Lawrence, J, ibid., p.194.

which to learn, it is advised by the makers of the software that isolated facts at the extreme minimum or maximum of the range of data are to be avoided and therefore certain unusual cases such as the case of Adachi v. Westhaver (where an exceptional damages quantum of $100,000 was awarded) were identified for removal from the database once all other factors had been tested. Other cases, such as those where quantum had been identified at $0 were also identified. I was aware that the Statistical Quantum Advisor program did not use certain problematic cases and so far as possible, I wanted to ensure that the same cases were removed. We return to this issue later in this chapter.

4.11 TOLERANCES

Tolerance specifies how accurate the output must be to be considered correct or "good." It is defined as a percentage of range of the output. It may be likened to a camera being focused very slowly. A high tolerance gives a blurred image of the pattern that we are trying to create. One may liken the process of tightening the tolerance to focusing a camera on an object and adjusting the lens so that the object is clearer through the view finder. If the tolerance level is set at 0.1 (as it is by default) and the network provides an output that is within 10% plus or minus of the maximum output then the fact is considered good. Not all of the above networks converged during training; in other words, they were having trouble learning/focusing. However, when the tolerance factors were loosened the networks seemed to train more easily. Even when the tolerance was loosened only slightly to 0.12 from 0.1 i.e. 12% instead of a 10% margin of error, the network trained far more easily requiring fewer runs of the data. By gradually tightening the tolerance factor the network trained quickly on the small remaining set of "bad" facts. This was a minor but nevertheless useful discovery if speedy training is an important criterion for the user. If the user is having difficulties running a trained network, one of the ways it can be improved is by starting training anew with a very loose tolerance (for instance, 0.4) and gradually tightening it over successive data runs. I tried running one network using such a style of training. I found that the mental health of the network could be preserved for much longer using this process of gradual reduction of the tolerance level. However, there was no mechanism for the automatic adjustment of the tolerance level and therefore, during training, the network required constant supervision. It would have been an excellent addition to the functionality of the software had this been possible. Using this system I began training with a high initial tolerance of 0.4 (40%) and over successive runs lowered it gradually to more acceptable levels. However, after 2000 data runs, and a lowering of the tolerance to 0.1 (10%) the networks mental health began to degrade. Also it did not seem to test any better than networks trained more quickly.

4.12 HOW MANY OUTPUT NEURONS?

This may seem to be a very odd question. After all, is it not the case that this neural network's purpose is to provide one figure signifying quantum in any given case? Why then should one need more than one output neuron? It would appear that one of the main reasons why earlier networks provided such erroneous results was precisely because I had used only one output neuron on a data set that contained extremes of data i.e. the vast majority of awards are between $5,000 and $40,000 although some "maverick" awards in the region of $60,000 to over $100,000 also exist. There are no hard and fast rules described in any of the relevant literature on how many output neurons should be used, although the general sentiment as regards input and output neurons is "more is better." Therefore I had to rely on intuition and experimentation. There are a variety of ways in which the quantum figure could be represented; First I could set up neurons representing bands of possible quantum e.g. $1- $10,000, $10,001 - 20,000, etc.; Secondly, I could arrange the data into a binary form with one neuron representing $1,000, one representing $2,000, one for $4,000, one for $8,000 and so on, building up to a final neuron for $64,000. Thus, for example, the quantum figure of $29,000 would be represented by a binary figure of 0011101. This output would then have to be reconverted into a decimal form to be readable by the end user. The problem with this approach would be that, at the higher end of the scale a difference of $5,000 may be dismissed as *de minimis*. However, at the lower end, an error of $5,000 is far more significant. Such is the nature of the data that one would need the network to train strongly on facts at the lower end of the quantum scale and less strongly on facts at the higher end. For instance, if the network provided an output of $45,000 as opposed to a required pattern output of $40,000 many lawyers would view such a result as sufficiently close as to make the network created of significant interest and value. However, an error of $5,000 between a required output of $1,000 and pattern output of $6,000 is a far more serious error. In order to perform this type of training function, one would have to write the neural network code from scratch rather than using an off-the-shelf package such as Brainmaker. Such a task was beyond my capabilities as a programmer and beyond the parameters of this thesis. (However this would make a very interesting research project for a programmer interested in the internal workings of a neural network.)

Another possible alternative would be to take the logarithm of the quantum figure so that the effect of extremes quantum figures would be lessened. However having spoken with the FLAIR programmers on this point I decided against this notion, the reason being as follows; If I used log (base 10), the sum of $100,000 would map to a value of 5 and the sum of $10,000 would map to a value of 4. That would mean that most of the awards would be bunched together between 4 and 5. It seemed that using this solution, I would be going from one extreme to another. The purpose of the data manipulation exercise is to "clip" the extreme values rather than using a metaphorical sledgehammer on all awards, extreme or otherwise. At their suggestion I decided to attempt a less powerful data manipulation exercise by taking the square root of all quantum values instead.

4.12.1 BANDS

The output values in the original database do not fall neatly within bands of $10,000. A brief analysis of the quantum figures in the Whiplash database shows that awards in excess of $50,000 are quite rare. The majority of the awards are between $5,000 and $40,000. The bands in the adapted program, Test3.prg, (a segment of which can be seen in appended form at Appendix C) reflect this fact. Initially, a total of 10 output neurons were used representing the following values; less than $150,000, less than $100,000, less than $80,000, less than $60,000, less than $40,000, less than $30,000, less than $20,000, less than $15,000, less than $10,000 and less than $5,000. The network was trained with 40 hidden neurons and a tolerance of 0.12. The first observation made was that the network trained much more slowly than previous networks. The RMS error curve began at a higher point (0.3410) and took longer to descend - an indication of slower training. The Column Histograms which reflect the mental health of the network degraded surprisingly quickly. By the 60th run of the data, they were showing characteristics of a "braindead" network, suggesting that nothing further could be learnt. However, for the first time, the network seemed to test extremely well. (Changing the number of hidden neurons made little difference to the training cycle.) From this experiment I drew two possible conclusions; either the use of 10 output neurons was too much for the network to handle and therefore I needed to find a compromise solution, of between 1 and 10 neurons that would allow the network to continue to train well whilst maintaining the newly discovered accuracy during testing or alternatively that the data needed further manipulation. The former conclusion was not convincing. Nowhere in any neural network literature is the possibility of too many output neurons mentioned. Indeed, most authors suggest using more input and output neurons rather than less; however they do suggest "pruning" the number of hidden neurons. It was suggested to me that I using large figures such as $40,000 would require the network to devote much of its computing power to solve calculations. Networks seems to train better when the numbers used are largely between 0 and 1.

To satisfy my curiosity, I decided to try a new experiment with 6 output neurons, maintaining the cluster of neuron values at the lower end of the dollar-value scale and using fewer neuron values at the top end. The values represented by these 6 neurons were as follows; less than $120,001, less than $60,001, less than $40,001, less than $25,001, less than $15,001 and less than $7,501. The Test3.prg program had to be changed and became Test4.prg. (A copy of the amended part of that program can be found at Appendix D.) At commencement of training the network revealed an even larger RMS error and the network again trained very slowly. However a higher degree of accuracy was achieved at the testing stage.

There is a methodological problem with this type of neuron architecture. The categories I picked (e.g. less than $25,000, less than $15,000) were chosen by intuition to reflect what I felt were helpful categories. However, in a given case, the required Pattern output might be $16,000, clearly closer to $15,000 than to $25,000. However, with the data set-up left unamended, it would be classified in the "less than $25,000" category. This problem was reflected when testing this network. Very often the results would show an output of approximately 0.95 against the neighbour of the required output neuron suggesting to me that the network was learning extremely accurately but that it was being thwarted by the data manipulation process. The problem was one of fuzzy values; the dBASE™ program used formalistic mathematical functions. In dBASE™ it is difficult to represent "around $16,000" as a number or a symbol; the programming language has no facility for the automatic fuzzification of data. Given the above results, it seemed pointless to pursue this banding method any further. I was being defeated by a "less than" sign. I wanted a network that would say to the user "the quantum is $x,000 give or take $2-3,000 either way." To add ironic insult to injury, when the learning "fired" the neighbouring neuron to that which was required, the network statistics facility told me that, in such situations the given fact was a "bad" fact. However, when the network showed a "good" fact, it was very "good" indeed. This gave me some encouragement that my purpose was achievable. It was a question of finding the right mixture of inputs and outputs and the appropriate form of data manipulation. In retrospect, this has proved to be no small achievement.

4.12.2 THE SQUARE ROOT

I was advised to try taking the square root of the output (instead of the logarithm) in order to lessen the effects of extreme examples. A simple command inserted in the Test3.prg program[174] allowed me to produce a new set of figures, between 0 and approximately 320. I then trained a network with 61 input, 40 hidden and 1 output neuron. Using an initial tolerance of 0.12 which was lowered gradually to 0.10 as the network continued to train, the network trained successfully on the training set in 160 data runs. The accompanying "New York skyline" Histogram suggested that the network remained "healthy" and was able to learn more at completion of training. The network also appeared to test well; a full table of results is set out below. I squared each output figure that the network produced and was able to quantify the error between pattern output and actual output. The table shows that of the 70 facts reserved for testing, the largest error

[174] "IF T->QU >=0, REPLACE B->QU WITH SQRT (T->QU) ENDIF

between output and pattern output was $70,228 and in one case, the network trained to within less than $1 of the required pattern. This was one of the more successful network developed, as judged by the testing criterion. (Values in Columns 4 and 5 are rounded up or down to the nearest dollar.)

Figure 4.7 Comparative Table of Outputs Using Square Root of Quantum.

CASE NUMBER	√ OF ACTUAL AWARD	NETWORK OUTPUT√	ACTUAL AWARD ($)	OUTPUT SQUARED ($)	DIFFERENCE ($)
1	100.04	122.33	10,008	14,721	4,713
2	130.03	147.60	16,907	21,785	4,879
3	63.05	99.96	3,975	9,992	6,017
4	97.00	151.06	9,409	22,819	13,410
5	112.04	97.94	12,553	9,592	2,961
6	81.04	97.94	6,568	9,591	3,023
7	100.04	94.47	10,008	8,925	1,083
8	112.04	116.68	12,553	13,614	1,061
9	71.07	116.68	5,051	13,614	8,563
10	97.00	110.43	9,410	12,195	2,785
11	123.10	126.06	15,154	15,891	737
12	87.04	84.42	7,576	7,127	449
13	100.04	147.26	10,008	21,686	11,678
14	92.02	112.80	8,468	12,724	4,256
15	141.01	102.66	19,884	10,539	8,655
16	157.65	157.99	24,854	24,960	106
17	39.06	66.17	1,526	4,378	2,852
18	126.06	132.82	15,891	17,641	1,750
19	148.02	224.98	21,910	50,616	28,706
20	55.02	106.46	3,027	11,334	8,307
21	87.04	109.67	7,576	12,028	4,452
22	112.04	194.57	12,553	37,896	36,343
23	212.05	213.57	44,965	45,612	647
24	141.01	134.17	19,884	18,002	1,882.
25	122.01	107.98	14,886	11,660	3,226
26	84.00	138.39	7,056	19,152	12,096
27	100.04	93.12	10,008	8,671	1,337
28	173.03	179.45	29,939	32,202	2,263
29	92.02	106.63	8,468	11,364	2,896
30	134.00	177.00	17,956	31,329	13,373
31	122.04	82.65	14,894	6,831	8,063
32	97.01	117.87	9,411	13,893	4,482
33	130.03	100.55	16,908	10,110	6,798

34	100.04	102.75	10,008	10,558	550
35	112.04	182.07	12,553	33,149	20,596
36	141.01	108.66	19,892	11,807	8,085
37	122.01	84.76	14,886	7,184	7,702
38	100.04	106.12	10,008	11,261	1,253
39	110.01	64.23	12,102	4,125	7,977
40	110.01	102.15	12,102	10,434	1,668
41	200.06	223.76	40,024	50,068	10,044
42	173.03	114.23	29,939	13,048	16,891
43	200.06	155.54	40,024	24,193	15,831
44	122.01	99.71	14,886	9,942	4,944
45	66.05	99.46	4,363	9,892	5,529
46	157.99	162.13	24,961	26,286	1,325
47	187.05	164.75	34,988	27,143	7,845
48	134.00	74.45	17,956	5,543	12,413,
49	141.01	140.42	19,884	19,718	166
50	224.05	271.02	50,198	73,452	23,254
51	173.03	173.28	29,939	30,026	87
52	173.03	202.68	29,939	41,079	11,150
53	157.99	96.98	24,961	9,405	15,555
54	132.06	131.30	17,440	17,240	200
55	173.03	107.64	29,939	11,586	18,353
56	141.04	158.67	19,892	25,176	5,284
57	83.98	122.85	7,052	15,092	8,040
58	128.09	132.56	16,407	17,572	1,165
59	157.99	122.85	24,961	15,092	9,869
60	157.99	321.23	24,961	103,189	78,228
61	157.99	126.99	24,961	16,126	8,835
62	187.05	183.25	34,988	33,581	1,407
63	173.03	116.60	29,939	13,596	16,343
64	126.06	139.07	15,891	19,340	3,449
65	187.05	212.14	34,988	45,003	8,015
66	141.01	114.57	19,884	13,126	6,758
67	87.04	108.07	7,576	11,679	4,103
68	46.99	95.65	2,208	9,148	6,940
69	100.04	229.03	10,008	52,455	42,447
70	246.03	233.42	60,531	54,485	6,049

The ideal testing network, in my opinion, would be one that provided results within $2-3,000/3,500 of the pattern output. Slightly more than this is still acceptable although an error in prediction of greater than $4-5,000 either way is likely to be enough to persuade a Plaintiff one way or the other on whether to pursue the matter all the way to court. Of the above 70 test cases, 26 cases could satisfy the testing criterion of less than $3,500 either way. This translates to 37% of the cases. However, I spoke with two Law students[175] who had worked on inputting cases into the Whiplash database. They agreed that they had developed an intuition of what the case might be worth which was accurate within a $7-10,000 range. Using this scale, 52 of the 70 cases or 74% of the test cases could be considered satisfactory. As can be seen from the above table some of the results were remarkably accurate, some coming within $500 of their required target (e.g. .Facts 12, 16, 51 and 54) whilst others were wildly inaccurate, the worst situation being Fact 60 which was nearly $80,000 off its target output. Clearly there was still a long way to go.

There was also a debt to be paid for using the square root function. The tolerance level of 0.1 was no longer acceptable; if I wanted an maximum error of no more than 10% then the tolerance had to be set at an absurdly low level, because all the outputs would have to be multiplied by themselves to provide an actual dollar figure.

(For some unknown reason the pattern output, when used in training tends to approximate the actual quantum values very slightly. For example, a quantum figure of $10,000 might become $10,008. In the larger scheme of things, I felt that such alterations were *de minimis.*)

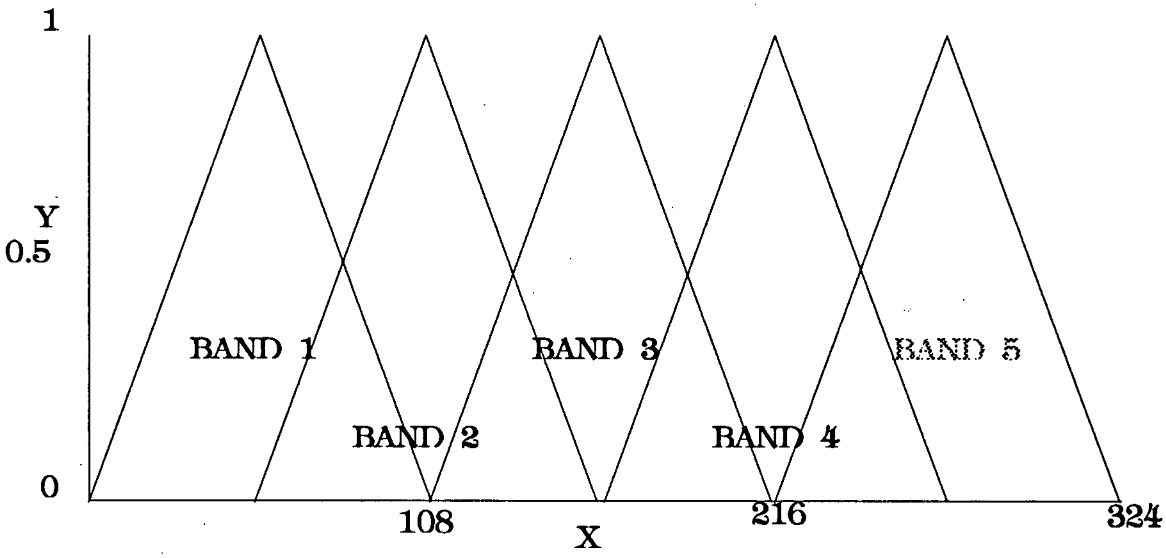
The next task was to see if I could combine the seeming advantages of square rooting the pattern output values with the use of more than one output neuron. My first idea was to develop some form of "fuzzy" banding mechanism which allows for a degradation of data,

[175] Lois Patterson and Serena Chandi.

which could overcome the rigidities of banding arranged by reference to the "less than" symbol. The ideal situation would be developing a database that provided values in the following type of graphical form;

Figure 4.8 - Graphical Representation of Fuzzy Banding of Quantum

(Y= Pattern Output Neuron Value; X = Square Root Value)



I wanted to be able to represent discrete numbers as a continuous variable using a number of neurons that would each partially fire. I decided to divide the square-root of the quantum figures into 5 bands of 108 units with an overlap of 54 (i.e. half) in each band. The above diagram (which shows the use of 5 output neurons each represented by a triangle) is a graphical representation of this design. The number 54 would be represented by neurons 1 and 2 having values of 1.0 and 0 respectively, whereas the number 81 would be represented by values of 0.5 in both the first and second neurons. Similarly, the value of 100 (which represents the square root of $10,000) is given a value of 0.15 in Band 1 and 0.85 in Band 2. Having consulted further with the FLAIR programmers, I became convinced that I could represent a single square-root of quantum figure using more than one neuron. My initial thought was that this could be achieved by a simple coding procedure. In fact the coding proved to be a little more complex. (An appended form of the code with accompanying comments can be found at Appendix E.) The hope was that the neural network, when trained would produce two figures in adjacent bands, one representing the downside slope of a band, the second representing the upside slope of its neighbour. If the trained network could also produce two figures in adjacent neurons for every testing fact then this was a good indication that the network had trained well. As can be seen from the table of outputs in Figure 4.9 below, this was not always achieved.

The network developed using this procedure used 61 input neurons, 40 hidden neurons and 5 output neurons. I trained the network over approximately 240 data runs. It did not train fully to completion. Indeed, the statistics provided by the software during training suggested that it trained really quite badly - the RMS error never fell below 0.08 as opposed to minimum values of approximately 0.04 in previous networks and the descent of the RMS error curve was slow and erratic. However, I was able to map the neuronal values back to a rough quantum figure by using a conversion graph similar to Figure 4.8. This proved to be an incredibly slow process but the alternative of writing a calculation program in the C programming language was equally unappealing, particularly as this was an experiment which might not be used again. The testing results for the first 35 of the 70 testing facts used in this network are set out in Figure 4.9 below;

Figure 4.9 - Table of Outputs from Multi-Output Neural Network

CASE NUMBER	PATTERN	OUTPUT	FIRST BAN VALUE	SECOND BAND VALUE	AVERAGE OUTPUT
	(TARGET VALUE)	(1 OUTPUT NEURAL NETWORK)			(5 OUTPUT NEURAL NETWORK)
1	100	122.33	152	159	155.5
2	130.03	147.60	161	No Value	161
3	63.05	99.97	73	84	78.5
4	97.00	151.06	102	85	93.5
5	112.04	97.94	111	108	109.5
6	81.04	97.94	81	81	81
7	100.04	94.47	91	70(124)	80.5(97)
8	112.04	116.68	101	No Value	101
9	71.07	116.68	91	79	85
10	97.00	110.43	99	99	99
11	123.10	126.06	138	162(195)	150(178.5)
12	87.04	84.42	87.5	92	90
13	100.04	147.26	123	142.5	133
14	92.02	112.80	111	109	110
15	141.01	102.66	86	55	70.5
16	157.65	157.99	85	60	72.5
17	39.06	66.17	?	?	?
18	126.06	132.82	75	81	78
19	148.02	224.98	160	No Value	160
20	55.02	106.46	81	84	82.5
21	87.04	109.67	94.5	82.5	88.5
22	112.04	194.57	105	111	108
23	212.05	213.57	115	129	122
24	141.01	134.17	102.5	102	102.2
25	122.01	107.98	134	118	126
26	84.00	138.39	83	84	83.5
27	100.04	93.12	85	96	90.5
28	173.03	179.45	93	156	124.5
29	92.02	106.63	79	96	87.5
30	134.00	177.00	109	157	133
31	122.04	82.65	75	73	74
32	97.01	117.87	52	185	118.5
33	130.03	100.55	122	138	130
34	100.04	102.75	81	76	78.5
35	112.04	182.07	101	116	108.5

Using a table similar to that in Figure 4.8, I was able to find the average $ value for each output. If there were two values, I could take an average. If there were three values for any given fact, I was advised by the FLAIR programmers to think of the first two figures as a set and the second and third figures as a separate set and take averages of each. Generally, there were two large values with a very small third (and possibly fourth) value so calculations of this nature were not problematic. If the two values were very close then it was an indication that the network had trained well on that fact and that it would be safe to have a high degree of confidence in the network. If the values showed a high degree of variance, I could have no such confidence. Certain testing facts caused problems. A good example of this is Fact 11 for which the network gave outputs of 0.452 on Band 1 which converts to 138 (a dollar value of $19,044), 0.999 on Band 2 which converts to 162 ($26,244) and 0.655 on Band 3 which converts to 195($38,025); each of these outputs has a fairly significant values on a scale of 0 to 1 and therefore could not be ignored *as de minimis*. Thus I had to include 2 values, by taking the average of the square root values for the first and second outputs, then repeating this for the second and third values. (The less accurate of the two numbers is in brackets.) Sometimes the network showed characteristics of indecision by providing very low output values in all output neuron bands. A good example of this phenomenon is Fact 17 for which the network provided outputs of 0.0094 in Band 1, 0.0768 in Band 2 and 0.0445 in Band 3. These values were beyond quantification using the above type of graph. One possible reason for the failure of the network in this particular instance was that this network did not take account of the dates of any of the cases. As stated above, Brainmaker takes every 10th case, starting with the 6th, the 16th, the 26th, etc. Fact 17 equates to the 176th case of 705 training cases. Therefore it was possible that this was quite an old case and that factor combined with a unusually low quantum figure of less than $2,000 made it impossible for the network to form any sort of conclusion. Although there was little I could do to compensate for a low quantum figure, apart from not using the case, I decided to include the date factor in future networks. It was pointed out to me subsequently that another reason why the 5 output network might have failed to deal adequately with extreme quantum figures was a subtle design flaw. Values of less than 54 had only 1 output neuron to fire, whereas all other values above this figure had 2. Equally at the opposite end of the spectrum, values of over 270 also had only one output neuron to fire. There was nothing against which to balance the first figure. What I needed was two further bands from 0 to 54 represented as a downwards slope and from 270 to 324 as a upwards slope. This might have overcome the problems of extreme values.

(Given the scale of the graphs I was not able to use figures with a values less than two decimal places (e.g. a value of 0.08 is too small to quantify on any graph. The alternative would be to write a short computer program. The results of this experiment were that in 19 or approximately 56% of the 34 usable testing cases, the network with 5 output-neurons proved to be more accurate than previous networks using only 1 output neuron and in 15 or approximately 44% of the 35 cases, the 5 output-neuron network proved to be less accurate. The above results also show that in certain circumstances the 5 output neuron network outperformed the 1 output neuron network by a significant margin[176].

[176] For example, Facts 26, 30 and 33.

Other facts showed the opposite[177]. Still other facts showed little difference under either network[178]. In short, I could find no compelling reasons to use the 5 output-neuron network. The ideal situation would be to take the best of each network and combine them into one network. But building a network with 6 outputs, (i.e. 5 output neurons for Bands 1 to 5 and a single output neuron for an unaltered square root value) offered no guarantee that the best features of each network would be used. Nevertheless, I felt that it would be a interesting and potentially worthwhile experiment.

This next network that I built consisted of 61 inputs, 40 hidden neurons and 6 outputs. It trained on over 280 runs of the training data. The RMS error curve was slow in descent and very erratic. I tested the network on the 283rd run of the data. My analysis revealed that the 6 neuron network to be less accurate than previous networks, using any criteria for analysis. It was clear that, far from combining the best parts of previous networks, this 6 output neuron network was more confused than ever. There may be a way around this problem by developing two separate neural networks, one which has 5 outputs and one which has only one output and using the latter network for verification. I did not explore this option further as I felt that I was being drawn into diversionary experiments away from the real purpose of the thesis. Therefore, I decided to abandon this combined banding and square-root data structure approach and return to the use of a single output neuron and attempt to improve that network, by including the use of dates, changing the

[177] For example, Facts 11, 16 and 23

[178] For example, Facts 1 and 2.

number of hidden neurons, changing the learning rates and training for a longer period of time.

Experiments with banding proved to be costly in terms of time taken to manipulate data but showed a few improvements on the single output neuron networks developed previously. Although the 5 output-neuron network reduced the number of wayward results, problems with interpreting output data had to be weighed against this. Also in this domain, for the time taken to build the networks, they offer little advantage. However, given a slightly different data structure, it may be that they would prove considerably better suited.

4.13 CHANGING OTHER FACTORS

4.13.1 INCLUSION OF DATES AS INPUT NEURONS

I returned to the use of a single-output (square root of quantum) neural network. I decided to include the date of the case as part of the input data. There were two potential methods of dealing with dates; either taking the view that each year was more important than previous years by an increment of 1 and thereby implementing the year of a given case through the use of 1 input neuron or banding the years together. As we have seen from the above experiments with banding of output values, banding (fuzzy or otherwise) is an arbitrary method of dealing with awkward data. Therefore, I decided to start initially with 1 input neuron to represent case dates, with a 1994 case being represented by the number 11, a 1993 case by the number 10 and so on. I trained this network on 62 input neurons, 40 hidden neurons and 1 output neuron, for over 250 data runs. The 104th data run had the lowest RMS error therefore I retrieved the 100th network and trained it over 4 more data runs.

I trained the second of these networks with 64 input neurons (utilizing 3 input neurons to reflect case date bands of 1993/1994, 1991/1992 and 1990 or prior, instead of 1), 40 hidden neurons and 1 output neuron which was the square root of quantum. I decided to train this network for a longer period to see how it trained. With over 6 hours of training and nearly 1,400 data runs, the network showed a very unusual RMS curve that was erratic in its initial descent and which then curved upwards in later runs. (The Histograms informed me that the networks "mental health" was irrecoverable by the 1000th run.) The network statistics revealed that the 221st data run provided the lowest RMS error and the least number of "bad" facts. (After the 300th run the results suggested that the network had started to memorize results rather than trying to learn.) I retrieved the 220th network and trained it on one further data run. The comparative results of testing of all three networks, (the "date = 3 inputs" network, the "date = 1 input" network and the "no date input" network) are set out below;

Figure 4.10 Comparative Table of Outputs Using Different Date Configurations

(#1) = most accurate network; (#2) second most accurate;

(#3) third most accurate (or least accurate) network;

CASE NUMBER	PATTERN	(1 OUTPUT & NO DATE INPUT) NEURAL NETWORK	(1 OUTPUT & 1 DATE INPUT) NEURAL NETWORK	(1 OUTPUT & 3 DATE INPUTS) NEURAL NETWORK
1	100.04	122.33(#1)	136.96(#3)	137.13(#4)
2	130.03	147.60(#1)	97.85(#3)	99.03(#4)
3	63.05	99.96(#4)	96.92(#2=)	96.92(#2=)
4	97.00	151.06(#4)	93.46(#1)	89.15(#2)
5	112.04	97.94(#3)	115.25(#1)	102.35(#2)
6	81.04	97.94(#3)	87.97(#1)	69.21(#2)
7	100.04	94.47(#1)	177.33(#3)	116.35(#2)
8	112.04	116.68(#1)	131.30(#3)	97.94(#2)
9	71.07	116.68(#3)	87.63(#2)	67.95(#1)
10	97.00	110.43(#3)	104.18(#1)	106.63(#2)
11	123.10	126.06(#1)	216.19(#3)	157.74(#2)
12	87.04	84.42(#2)	79.44(#3)	85.77(#1)
13	100.04	147.26(#3)	109.33(#1)	88.56(#2)
14	92.02	112.80(#2)	117.27(#3)	111.87(#1)
15	141.01	102.66(#3)	123.1(#2)	147.26(#1)
16	157.65	157.99(#1)	203.25(#3)	152.16(#2)
17	39.06	66.17(#2)	124.37(#2)	46.49(#1)
18	126.06	132.82(#1)	116.35(#2)	72.00(#3)
19	148.02	224.98(#3)	313.42(#2)	170.66(#1)
20	55.02	106.46(#2)	102.66(#1)	111.28(#2)
21	87.04	109.67(#3)	95.40(#2)	88.21(#1)
22	112.04	194.57(#2)	200.06(#3)	114.40(#1)
23	212.05	213.57(#1)	178.26(#2)	148.36(#3)
24	141.01	134.17(#1)	111.28(#2)	111.45(#3)
25	122.01	107.98(#3)	126.74(#1)	136.11(#2)
26	84.00	138.39(#3)	118.8(#2)	100.97(#1)
27	100.04	93.12(#2)	101.82(#1)	75.13(#3)
28	173.03	179.45(#2)	171.68(#1)	108.32(#3)
29	92.02	106.63(#3)	106.29(#2)	99.79(#1)
30	134.00	177.00(#3)	175.81(#2)	100.89(#1)
31	122.04	82.65(#3)	94.30(#1)	84.42(#2)
32	97.01	117.87(#2)	124.03(#3)	102.24(#2)
33	130.03	100.55(#3)	122.60(#2)	129.69(#1)
34	100.04	102.75(#1)	105.87(#2)	85.26(#3)
35	112.04	182.07(#3)	154.19(#2)	86.11(#1)
36	141.01	108.66(#2)	117.78(#1)	98.27(#3)
37	122.01	84.76(#3)	132.99(#2)	122.43(#1)
38	100.04	106.12(#1)	136.70(#3)	87.63(#2)
39	110.01	64.23(#3)	122.77(#1)	87.55(#2)

40	110.01	102.15(#1)	157.40(#3)	114.32(#2)
41	200.06	223.76(#1)	248.63(#2)	135.94(#3)
42	173.03	114.23(#3)	142.36(#2)	142.79(#1)
43	200.06	155.54(#2)	126.82(#3)	199.97(#1)
44	122.01	99.71(#2)	106.12(#1)	84.47(#3)
45	66.05	99.46(#2)	113.98(#3)	84.59(#1)
46	157.99	162.13(#3)	150.47(#1)	160.86(#2)
47	187.05	164.75(#2)	178.26(#1)	102.24(#3)
48	134.00	74.45(#3)	150.73(#1)	172.10(#2)
49	141.01	140.42(#1)	159.51(#3)	123.78(#2)
50	224.05	271.02(#3)	216.23(#1)	263.50(#2)
51	173.03	173.28(#1)	189.25(#2)	122.68(#3)
52	173.03	202.68(#2)	208.42(#3)	179.87(#1)
53	157.99	96.98(#3)	146.5(#1)	142.79(#2)
54	132.06	131.30(#1)	167.87(#3)	119.98(#2)
55	173.03	107.64(#3)	126.14(#2)	143.21(#1)
56	141.04	158.67(#2)	130.37(#1)	162.13(#3)
57	83.98	122.85(#3)	118.80(#2)	93.04(#1)
58	128.09	132.56(#2)	129.35(#1)	117.28(#3)
59	157.99	122.85(#2)	144.73(#1)	44.127(#3)
60	157.99	321.23(#3)	178.01(#1)	198.22(#2)
61	157.99	126.99(#2)	130.87(#1)	111.53(#3)
62	187.05	183.25(#1)	193.81(#2)	142.87(#3)
63	173.03	116.60(#2)	141.27(#1)	113.73(#3)
64	126.06	139.07(#2)	164.50(#3)	127.83(#1)
65	187.05	212.14(#3)	188.15(#1)	194.48(#2)
66	141.01	114.57(#1)	89.826(#3)	102.83(#2)
67	87.04	108.07(#2)	131.30(#3)	84.00(#1)
68	46.99	95.65(#1)	161.45(#2)	178.18(#3)
69	100.04	229.03(#3)	176.49(#1)	179.62(#2)
70	246.03	233.42(#1)	214.42(#2)	160.52(#3)

4.13.2 A COMPARATIVE ANALYSIS OF DATE INPUTS

The second network (using only one neuron for date inputs) was the most accurate network most of the time (27 times out of a possible 70) and least accurate the least number of times (18 out of a possible 70.) The "banded date" network was the second most accurate network; of the 70 testing facts, 39 fact outputs (56%) were more accurate than the network with no dates and 31 facts (44%) were less accurate. Of these, 20 testing fact outputs were significantly better[179] and 14 tested significantly worse[180].

The results seemed to justify the use of one input neuron to represent the dates of the case. However, one can also find in the data evidentiary support for using more than 1 neuron e.g. the problematic Fact 17 with its quantum square root figure of 39 ($1,521) was matched by an output from the "banded date" network of 46.49 ($2,161) as opposed to an output of 66.17 ($4,378) from the "no date" network, an output of 124.37 from the one input network (and we will recall, no calculation at all from the 5 output neural network.) Bearing in mind the accepted rule amongst neural network developers regarding the number of input neurons that of "the more inputs the better," I decided to try one last variation on this theme and build another network with 1 input neuron for each different year. Thus there would be 11 input neurons to represent the dates of all the training cases. This network was trained on over 170 data runs. This network seemed to suffer from "mental health" degeneration earlier than the other networks for reasons which were not immediately apparent. The network with the lowest RMS error and least number of "bad" facts was the network developed on the 130th data run. This was tested. The results suggested that the performance surpassed that of the "no-date network" but did not compare favourably with the two networks which also took account of date inputs. Thus with this particular aspect of the data set, I was <u>not</u> able to confirm the belief

[179] With an accuracy improvement of over 20 points, equating to an increased accuracy of over $400. Facts 4,9,13,15,17,19,22,26,35,37,42,43,48,52,53,55,60 and 69 all tested significantly better

[180] Using the same 20 point criterion, Facts 2,11,18,23,24,28,41,47,49,51,59,62,68 and 70 all tested significantly worse.

that with regard to input neurons "more is better." However, I was able to confirm that the use of one input neuron to represent dates worked best.

4.14.3 LEARNING RATES

We recall from Chapter 3 that the learning rate[181] determines how large or small an adjustment the network will make during training. Philipps[182] used the useful metaphor of a game of golf. Changing the learning rate may be likened to changing one's golf club - you may reach the target more quickly with a driving wedge but may miss the hole several times if one persists with its use. At different learning rates, Philipps discovered different outputs. I performed various experiments using different learning rates. The learning rate must be greater than zero; (Brainmaker sets the learning rate at 1.0 by default.) Using the same data set as before and the same number of hidden neurons throughout, I experimented by changing the learning rate to 0.8, 0.4 and also up to 2.0 The results were as follows;

Figure 4.11 Comparative Table of Outputs Using Varying Learning Rates

(#1) = most accurate network; (#2) second most accurate;

(#3) third most accurate (or least accurate) network;

CASE NUMBER	PATTERN	LEARNING RATE = 1	LEARNING RATE = 0.8	LEARNING RATE = 0.6
1	100.04	176.49(#2)	107.90(#1)	187.47(#3)
2	130.03	166.94(#3)	151.82(#2)	125.30(#1)
3	63.05	103.34(#2)	105.79(#3)	85.94(#1)
4	97.00	107.64(#3)	100.30(#2)	92.53(#1)
5	112.04	156.47(#3)	126.40(#1)	155.54(#2)

[181] See Appendix G, The Glossary of Brainmaker terms.

[182] See note 141 to Chapter 3.

Figure 4.11 (continued)

6	81.04	85.52(#1)	100.30(#2)	71.50(#3)
7	100.04	134.34(#2)	152.92(#3)	65.84(#1)
8	112.04	123.61(#2)	169.65(#3)	110.69(#1)
9	71.07	94.81(#2)	100.40(#3)	82.05(#1)
10	97.00	111.28(#2)	127.16(#3)	110.94(#1)
11	123.10	200.40(#2)	260.88(#3)	140.42(#1)
12	87.04	95.401(#2)	100.30(#3)	84.92(#1)
13	100.04	111.11(#2)	94.89(#1)	74.87(#3)
14	92.02	141.43(#3)	83.49(#2)	106.80(#1)
15	141.01	160.86(#2)	146.08(#1)	112.04(#3)
16	157.65	176.74(#1)	132.31(#2)	128.43(#3)
17	39.06	87.72(#3)	59.42(#1)	59.67(#2)
18	126.06	135.69(#2)	132.82(#1)	101.73(#3)
19	148.02	203.86(#2)	204.96(#3)	177.67(#1)
20	55.02	128.26(#3)	123.89(#2)	100.97(#1)
21	87.04	106.72(#3)	102.83(#2)	80.28(#1)
22	112.04	173.03(#3)	136.20(#2)	112.67(#1)
23	212.05	195.92(#1)	146.90(#3)	179.45(#2)
24	141.01	103.42(#3)	133.41(#1)	119.64(#2)
25	122.01	104.01(#2)	133.66(#1)	167.79(#3)
26	84.00	128.43(#3)	116.01(#2)	115.59(#1)
27	100.04	115.50(#3)	97.51(#1)	90.67(#2)
28	173.03	131.55(#1)	113.90(#2)	70.29(#3)
29	92.02	130.28(#3)	112.80(#2)	96.67(#1)
30	134.00	217.97(#3)	164.00(#2)	139.66(#1)
31	122.04	103.17(#1)	87.98(#2)	76.56(#3)
32	97.01	134.17(#3)	85.18(#1)	82.73(#2)
33	130.03	144.56(#1)	95.82(#2)	170.24(#3)
34	100.04	104.27(#1)	91.35(#2)	81.80(#3)
35	112.04	141.01(#1)	153.34(#1)	144.31(#2)
36	141.01	132.14(#1)	131.13(#2)	116.01(#3)
37	122.01	144.81(#3)	99.96(#2)	110.77(#1)
38	100.04	112.04(#3)	131.38(#2)	102.49(#1)
39	110.01	129.61(#3)	110.26(#1)	113.90(#2)
40	110.01	175.22(#3)	151.15(#2)	126.48(#1)
41	200.06	123.69(#2)	252.43(#1)	117.02(#3)
42	173.03	106.88(#1)	91.18(#2)	64.991(#3)
43	200.06	149.37(#3)	173.62(#1)	154.36(#2)
44	122.01	69.64(#3)	87.97(#1)	76.90(#2)
45	66.05	119.22(#3)	96.16(#2)	84.17(#1)
46	157.99	140.34(#2)	156.47(#1)	172.44(#3)
47	187.05	140.50(#2)	144.90(#1)	96.92(#3)
48	134.00	172.10(#2)	189.50(#3)	144.98(#1)
49	141.01	116.77(#2)	217.29(#3)	140.76(#1)
50	224.05	286.73(#3)	213.74(#1)	179.11(#2)
51	173.03	166.78(#1)	132.23(#2)	159.43(#3)
52	173.03	188.99(#1)	197.19(#2)	200.14(#3)
53	157.99	147.09(#1)	209.35(#3)	189.84(#2)

Figure 4.11 (continued)

54	132.06	156.05(#3)	139.58(#1)	143.04(#2)
55	173.03	149.54(#2)	108.74(#3)	155.20(#1)
56	141.04	148.87(#1)	163.73(#2)	115.41(#3)
57	83.98	103.00(#2)	112.80(#3)	91.18(#1)
58	128.09	143.38(1=)	143.38(1=)	155.54(#3)
59	157.99	74.28(#3)	125.30(#1)	123.44(#2)
60	157.99	158.41(#1)	192.99(#2)	119.64(#3)
61	157.99	131.89(#3)	137.80(#2)	138.39(#1)
62	187.05	194.15(#2)	198.54(#3)	182.15(#1)
63	173.03	138.65(#1)	132.48(#2)	79.77(#3)
64	126.06	162.30(#2)	145.15(#1)	165.42(#3)
65	187.05	181.64(#1)	185.61(#2)	187.05(#1)
66	141.01	95.91(#2)	104.01(#1)	91.01(#3)
67	87.04	101.65(#1)	141.94(#3)	102.07(#2)
68	46.99	123.1(#3)	105.45(#2)	81.63(#1)
69	100.04	165.59(#2)	134.25(#1)	169.82(#3)
70	246.03	158.67(#2)	114.66(#3)	170.49(#1)

(N.B. The above results should be looked at for comparative purposes only. The reader should not be concerned that certain results appear far from the required pattern output.)

The network with the learning rate of 2.0 trained very erratically indeed. It exhibited some very unusual behaviour. Indeed after a mere 60 or so runs of the training set, the Column Histogram looked very similar to that representing a "brain dead network" i.e. a network that had memorized rather than learnt. The results obtained from this network have not been included because they were nonsensical. Clearly, the network was making far too large adjustments with every run and this meant that it was failing to settle into a satisfactory learning pattern. Equally, the network with a learning rate of 0.4 took so long to show progress during training that I began to wonder if it would ever train to completion. I took the best network (i.e. the one with the lowest RMS error and smallest number of "bad" facts) that each of the 3 training sessions produced and ran the testing data through it. The results are set out above. The network with a learning rate of 0.8 should have trained more slowly than the network with a learning rate of 1.0. Interestingly, the 0.8 learning rate network trained to its best network *more* quickly rather than less, with under 100 runs of the data as opposed to 120 to reach its lowest RMS error figure. I was (and remain) unable to explain this phenomenon, although from this one might conclude that the learning rate of 1.0 was too high. Not surprisingly, the network with the 0.6 learning rate took much longer to reach its lowest RMS-error network. The best network was produced on the 150th run of the data.

As regards outputs, we can see from an examination of the above table that, like Phillips, I obtained different outputs using different learning rates. Of the three rates, the network with the learning rate of 0.6 provided me with the most accurate outputs, with 30 out of the 70 testing facts testing the best of the three networks. However, it also came third in 26 of the 70 possible testing cases. The network with the learning rate of 0.8 came third the least number of times (17) and still had a high number of first places. Despite being unable to establish unequivocally that the lower learning rate is better, it is clear from the above table that with this data, the learning rate of 1.0 is not appropriate and that outputs can be improved by a lower learning rate. It is open for debate as to the exact level of the most appropriate learning rate in this context. I decided to compromise and set the learning rate at 0.7 in my future work.

4.15 HOW MANY HIDDEN NEURONS - PART 2

I stated earlier in this Chapter that I would return to this experiment. My earlier experiment with this variable showed that the optimum number of hidden neurons is likely to lie somewhere between 20 and 40 and that networks can test reasonably well without having to train on all the given facts. However, I wanted to provide myself with outputs for networks with extreme numbers of hidden neurons, as sometimes (depending on the type of data,) some neural networks can work adequately on as little as one-tenth of the total of inputs and outputs[1], which in this situation (with 64 inputs and 1 output) would equate to 6-7 hidden neurons.

I tested a number of networks taking the best results for other variables from other experiments that I had run. For example, the learning rate was set at 0.7; there was one date neuron; one output neuron represented the square root function of quantum, etc. I trained 7 networks with the number of hidden neurons set at 10, 15, 20, 25, 30, 35 and 40 respectively. It was hoped that some kind of trend in the testing output might reveal itself. I already knew that with a larger number of neurons, the network would train well, possibly to completion (i.e. satisfaction of the tolerance level,) but it might not necessarily test well. However, it may be that in such circumstances the tolerance level is set too high. The results are set out in Figure 4.12 below. The most accurate network for any given fact is indicated by a star (*).

[1] The FLAIR project have been working on a neural network for information retrieval purposes. Their proto-type network works adequately and consists of 50 inputs, 99 outputs and 15 hidden neurons.

Figure 4.12

CASE	PATTERN	HIDDEN NEURONS = 10	HIDDEN NEURONS = 15	HIDDEN NEURONS = 20	HIDDEN NEURONS = 25	HIDDEN NEURONS = 30	HIDDEN NEURON S = 35	HIDDEN NEURONS = 40
1	**100.04**	163.65	191.44	152.25	55.02	132.65*	136.96	147.60
2	**130.03**	113.98	99.79	163.40	157.03	138.98*	97.85	146.00
3	**63.05**	78.93	84.58	77.83	87.38	86.79	96.92	80.027
4	**97.00**	94.81	84.00	93.04	82.14	92.53	93.46	80.79
5	**112.04**	117.95	152.75	149.12	154.19	109.84*	115.25	138.90
6	**81.04**	56.46	77.16	70.99	77.58	89.57	87.97	69.81
7	**100.04**	105.20*	62.54	95.82*	122.01	123.44	177.33	147.26
8	**112.04**	157.48	63.89	66.00	103.25	77.408	131.30	113.81*
9	**71.07**	53.59	77.92	67.95*	81.04	91.35	87.63	67.02*
10	**97.00**	104.01	106.21	100.04	108.07	99.71*	104.18	98.19
11	**123.10**	178.35	180.54	130.20*	172.86	142.70	216.19	216.53
12	**87.04**	74.03	82.98	67.27	88.56	74.03	79.44	63.98
13	**100.04**	56.21	62.88	98.02	106.04*	120.23	109.33	134.93
14	**92.02**	112.63	107.05	146.84	87.71*	100.72	117.27	106.38
15	**141.01**	201.33	106.80	40.83	95.40	111.36	123.10	131.04*
16	**157.99**	148.19	158.58*	140	177.84	121.33	203.52	183.42
17	**39.06**	59.25	70.99	41.68*	66.00	68.37	124.37	54.43
18	**126.06**	119.64	134.08	83.07	124.62*	119.81	116.35	162.64
19	**148.02**	166.94*	101.73	254.02	240.27	210.45	313.42	237.48
20	**55.02**	96.42	103.59	97.01	115.84	93.97*	102.66	97.01
21	**87.04**	92.28	77.24	74.87	79.27	83.24	95.40	86.70*
22	**112.04**	135.77	124.29	54.35	140.42	121.08*	200.06	174.63
23	**212.05**	177.84	175.81	168.42	209.86*	122.01	178.26	157.23
24	**141.01**	117.61	132.14	118.12	102.83	118.20	111.28	126.65
25	**122.01**	117.95	141.52	176.15	133.92*	101.31	126.74	113.22
26	**84.00**	93.46	121.08	122.09	112.80	99.54	118.80	126.31
27	**100.04**	104.52	96.25	109.17	101.06	115.50	101.82*	123.10
28	**173.03**	100.63	145.66	110.94	105.20	81.04	171.68*	100.30

Figure 4.12 (continued)

CASE	PATTERN	HIDDEN NEURONS =10	HIDDEN NEURONS = 15	HIDDEN NEURONS = 20	HIDDEN NEURONS = 25	HIDDEN NEURONS = 30	HIDDEN NEURON S = 35	HIDDEN NEURONS = 40
29	**92.02**	76.48	81.46	107.73	88.31	94.39*	106.29	117.36
30	**134.00**	124.62*	153.94	157.91	179.70	166.94	175.81	139.24
31	**122.04**	77.75	79.94	82.39	82.39	83.41	94.30*	80.79
32	**97.01**	132.65	97.34*	113.22	81.21	83.83	124.03	117.95
33	**130.03**	165.26	143.97	109.67	140.84	128.93*	122.60	92.19
34	**100.04**	75.63	82.14	96.84	82.73	94.30	105.87	119.81
35	**112.04**	112.80*	140.84	123.19	130.54	171.51	154.19	155.71
36	**141.01**	94.64	104.01	134.51*	125.89	104.10	117.78	127.16
37	**122.01**	104.94	101.39	126.82*	92.11	104.01	132.99	124.37
38	**100.04**	126.65	105.79	116.94	102.83*	122.17	136.70	166.69
39	**110.01**	105.79*	78.08	115.16*	88.31	85.94	122.77	96.08
40	**110.01**	232.16	136.20	166.27	91.60	123.95*	157.40	70.74
41	**200.06**	160.19	171.25	208.84	188.82	195.16	248.63	197.95*
42	**173.03**	90.00	136.03	109.08	134.42*	104.86	142.36	127.66
43	**200.06**	157.23	166.02	198.88*	131.21	170.41	126.82	235.96
44	**122.01**	78.08	106.55	96.16	80.53	100.21	106.12*	85.60
45	**66.05**	100.04	98.53	94.30	100.04	91.68*	113.98	111.62
46	**157.99**	159.68*	131.89	108.24	134.59	131.47	150.47	141.69
47	**187.05**	83.66	243.48	128.85	100.80	176.32	178.26*	239.68
48	**134.00**	123.69	134.08*	174.38	127.58	105.36	150.73	147.01
49	**141.01**	171.00	170.75	202.51	155.96	144.39*	159.51	156.64
50	**224.05**	247.53	197.27	166.78	214.25*	207.15	216.36	244.66
51	**173.03**	153.60	162.47*	193.55	156.22	137.13	189.25	161.62
52	**173.03**	193.72	146.00	148.11	194.22	182.15*	208.42	199.47
53	**157.99**	105.03	175.39	153.34	110.85	160.02*	146.50	215.77
54	**132.06**	147.85	121.84*	98.44	157.91	170.83	167.87	173.70
55	**173.03**	76.48	117.70	157.48	180.54	172.35*	126.14	187.64

Figure 4.12 (continued)

CASE	PATTERN	HIDDEN NEURONS =10	HIDDEN NEURONS = 15	HIDDEN NEURONS = 20	HIDDEN NEURONS = 25	HIDDEN NEURONS = 30	HIDDEN NEURON S = 35	HIDDEN NEURONS = 40
60	**157.99**	126.74	179.19	163.57*	167.28	198.54	178.01	171.68
61	**157.99**	112.80	137.13*	116.68	120.99	123.10	130.87	133.83
62	**187.05**	130.37	169.06	116.68	138.48	146.16	193.81	187.89*
63	**173.03**	130.79	81.55	167.87*	141.18	151.06	141.27	166.86
64	**126.06**	232.41	149.29	134*	167.54	212.14	164.50	164.16
65	**187.05**	213.91	192.96	235.79	191.44	231.23	188.15*	241.87
66	**141.01**	95.74	89.74	140.93*	121.58	73.02	89.826	84.00
67	**87.04**	106.04	95.232*	105.03	113.56	103.59	131.30	146.67
68	**46.99**	131.64	110.60	118.63*	112.54	74.54	161.45	104.77
69	**100.04**	137.72	150.04	223.54	139.49	234.61	176.49	123.10*
70	**246.03**	203.10	141.27	148.70	118.54	114.57	214.42*	129.02

On some facts all networks trained adequately therefore the accuracy of any given network is less crucial. For example, on Fact 3, the most accurate network (hidden neurons = 20) produced a translated $ error of $2,083, whilst the least accurate network (hidden neurons = 35) produced a translated $ error of $5,418. This is a fairly negligible difference compared with, say Fact 22 where the most accurate network (hidden neurons = 30) produced a translated $ error of $2,107 and the least accurate network (hidden neurons = 35) produced a translated $ error of $27,471. Therefore, where all outputs for a given fact are bunched closely together, there is no emboldening as clearly all of the networks had no problems dealing with the data relating to that given fact. If two networks produced outputs with almost identical errors and these two networks also happened to be the most accurate for that fact, they are both emboldened. Using these criteria, the network with 30 hidden neurons appeared to be the most accurate, (15 outputs being the most accurate out of possible 70.) However, in my opinion, in this type of comparative analysis, being the most accurate of seven networks for any given fact is not as important as providing a consistently accurate output for all 70 facts. Even the best performing network was showing wayward results that seemed incomprehensible. Trying to discern which network is the most consistent was a particularly difficult process, short of translating all the output figures to a $ value.

One must be very careful when thinking of these figures in terms of true dollar figures as they can be deceptive. For instance, the difference between 90 squared ($8,100) and 100 squared ($10,000) is not 10 squared ($100) but 43.59 squared ($1,900), whereas the difference between 210 squared ($44,100) and 220 squared ($48,400) is 65.57 squared ($4,300.) Therefore, one cannot pick a figure at random (e.g. 30 points) and use it as a test of accuracy because as we can see, in true $ values, an error of 10 at the lower end of the output spectrum is much less than an error of 10 at the higher end. I was hoping that I would find one network that was both consistent and accurate. The network with 30 hidden neurons came closest to satisfying this demand, but even this network had occasional hiccups[1]. A further perusal of the results suggested to me that I should use more than 1 network to be tested in comparison with the Statistical Quantum Advisor program. In the comparative tests described below I used 2 networks (one with 30 hidden neurons and one with 25 neurons.) I also developed a network that used an identical number of training facts as the SQA and used it for comparative analysis.

4.16 A HIGHER NUMBER OF DATA RUNS?

In order to perform this experiment, I had to transfer my data onto a SUN Sparc Workstation that had a much higher computing capacity than that of Brainmaker™ running under Windows™ on a 386/25Mhz PC. My neural network software would take 6-7 hours to perform the same operation that could be accomplished on the SUN computer in 20 minutes. However, the program used on the SUN computer was written in C, was fairly user-unfriendly and required a high degree of experience in the art of programming neural networks. Unfortunately, my programming skills were minimal. However, using this faster but more basic technology, I was able to run my training set over approximately

[1] See for instance, the outputs for this network with facts 19, 23, 28, 35 and 42.

48,000 data runs. There were also significant problems with this program. The most significant of these was the problem of over-training[2], the process whereby the network trained very well, in fact too well on the training facts and fails to generalize well. It was an inadequacy of this software that prevented me from finding the optimum network. The only solution available to me was to stop the training process at random and examine the results and hoping to stumble across a useful network. The key to obtaining a useful neural network is being able to stop the network before it starts to memorize the facts. Neural networks possess what we might describe in humans as laziness. If a solution can be found to a problem in the shortest possible time it will be sought by the neural network. If that solution should happen to involve memorization, then that solution will be sought nevertheless. The results of this experiment were that the network trained very well but was useless when given cases that it had not seen.

4.17 COMPARISON WITH THE STATISTICAL QUANTUM ADVISOR

This is probably the most important experiment documented in this thesis. I decided to put my best two neural networks up against the Statistical Quantum Advisor (SQA.) The SQA was written in the C programming language in 1992 by John McClean who is a programmer at FLAIR. Like myself, he took the Whiplash database and subjected it to a degree of pre-processing by removing difficult cases such as ones that involved subsequent injuries, multiple injuries, other injuries, etc. This reduced the set of cases to from an initial 1102 to 477, less than my training set of 635 cases. One of McClean's biggest

[2] See Appendix [ ], Glossary of Neural Network terms.

complaints regarding his program was one familiar to me, namely that there were not enough training cases. Training with this smaller amount of data would have one of two effects for neural networks; either the training set might be so small as to preclude learning altogether or it might make a cleaner data set that would lead to a more accurate network.

In order to perform this comparative analysis, I had to take his set of training data and manipulate it into a form that could be read by my neural network program. This involved taking the original database, deleting all the records that the SQA was unable to deal with and then running my Test*.prg program over that data file. I decided initially that I should test his program against mine in its present form, using 635 cases and then train it again using the same 477 cases as the SQA. Once this had been accomplished I then had to create what the makers of Brainmaker describe as a running fact file - a file consisting only of inputs, upon which the neural network could predict an output for each fact. My two most accurate networks which used 635 training facts did not perform as well as the SQA. Figure 4.13 below shows the outputs for my 43 of the 70 testing facts compared with the SQA outputs. A graphical representation of the same results can be found in Appendix H. All output figures have been rounded up or down to the nearest dollar. The most accurate network/program output for each fact is marked with a star. As we have seen above, there is a more important test than this. We are concerned to find the most consistent programming structure. The two neural networks should not be thought of as competing against one another. We have seen from experiments above that they have different attributes.

Figure 4.13 - Comparative Table of Neural Network and SQA Outputs

CASE NUMBER	ACTUAL AWARD	SQA OUTPUT ($)	NETWORK OUTPUT ($) (30 HIDDEN NEURONS)	NETWORK OUTPUT ($) (25 HIDDEN NEURONS)
1	10,000	19,727	17,596*	3,027
2	16,900	27,564	19,315*	24,775
3	4,000	5,755*	7,531	7,634
4	9,500	6,518	8,561*	6,746
5	12,500	7,838	12,064*	23,775
6	6,500	5,531	8,023	6,081*
7	10,000	11,474*	15,237	14,886
8	12,500	9,529	5,991	10,660*
9	5,000	5,492*	8,344	6,567
10	9,500	8,106	9,942*	11,679
11	16,000	13,186*	20,263	29,880
12	7,500	6,751	5,480	7,843*
13	10,000	8,897	14,455	11,244*
14	8,500	10,246	10,144	7,694*
15	20,000	12,807*	12,401	9,101
16	25,000	NOT USED	-	-
17	1,500	10,315	4,674	4,356*
18	16,000	23,558	14,354	15,530*
19	22,000	25,495*	44,289	57,600
20	3,000	11,188	8,829*	13,418
21	7,500	8,194	6,928*	6,283
22	12,500	9,947	14,660*	19,600
23	45,000	29,252	14,886	44,041*
24	20,000	16,826*	13,971	10,574
25	15,000	11,487	10,263	17,935*
26	7,000	NOT USED	-	-
27	10,000	16,132	13,340	10,213*
28	30,000	13,529*	9,568	11,067
29	8,500	10,066	8,908*	7,798
30	18,000	13,078*	27,869	32,292
31	15,000	11,945*	6,957	6,788
32	9,500	9,905*	7,027	6,594
33	17,000	9,575	16,662*	21,561
34	20,000	NOT USED	-	-
35	12,500	18,910	29,415	17,040*
36	20,000	NOT USED	-	-
37	15,000	16,806*	10,818	8,483
38	10,000	10,548*	14,925	10,574
39	12,000	9,559*	7,385	7,798
40	12,000	NOT USED	-	-
41	40,000	32,982	38,087*	35,652
42	30,000	24,368*	10,995	18,069
43	40,000	25,398	29,039*	17,216

Readers will note that in previous tables, 70 cases have been used for testing. This experiment allowed for an analysis of no more than 43 facts because the SQA program used 477 cases taken from the Whiplash database when it consisted of only 1,100 cases as opposed to its present total of over 1,700 cases. Thus, I was able to compare only the first 43 of my 70 testing cases in this analysis. The above table shows that, of the 38 cases where a comparison was possible, the SQA program was the most accurate in 15 cases. The 1st neural network was the most accurate in 12 cases and the 2nd network was the most accurate in 11 cases. Consistency was a more important attribute than accuracy in any given case. Using this criterion, both networks performed almost as well as the SQA. In 18 of the 38 cases, both networks were accurate within a range of $3-4,000. The SQA was consistent in 21 of the 38 cases, (using the same judgment criterion.) The SQA seemed to perform better with high-end awards, although it was not without an occasional hiccup as in case number 43. The neural networks seemed to cope better with the very low awards (for instance, the problematic case number 17 with an award of $1,500) suggesting that the connectionist architecture can cope better than linear programming with contradictory data. Now and again, however, the networks would provide upsets. For example, in case number 19, the networks provided outputs of around $44,000 and $58,000 against an actual award of $22,000. One would imagine that both neural networks and the SQA would be inaccurate on the same cases. Interestingly, and inexplicably, this was not so.

4.18 CONCLUSIONS

In the course of writing this chapter I have developed between 40 and 50 neural networks. I have not been able to develop a neural network that is likely to be of immediate commercial interest to ICBC or to lawyers who deal with whiplash injuries on a daily basis. This was not to be expected but would have been pleasantly surprising if it had been achieved. However, I have been able to build networks that come close to performing as well as a program built by a programmer experienced both in the domain of whiplash injuries and who is familiar with a range of computer programming techniques. I have familiarity with neither of these areas and yet by giving data to a neural network, have witnessed the evolution of a program that comes close to (but cannot as yet surpass) the performance of the SQA. I believe that, in the circumstances, one should be quite satisfied with such a conclusion.

One of the most important lessons to be learnt from experimenting with neural networks is the importance of finding an adequate search method. I started working with this data set changing items such as the representation of the dates of the cases, changing how the quantum figure might be represented, altering the learning rate, changing the number of hidden neurons, etc. I tried to work through all the possible variables as methodically as possible. However, time constraints and the limitations of the software have prevented me from making an exhaustive search of all possible variables. It may be the optimum network variables for this data set are a learning rate of 0.6, an error Tolerance of .0.8, a training over 12,000 runs, 30 hidden neurons, 61 inputs and 1 output neuron. However finding this optimum combination might take one person a lifetime. Alternative measures for dealing with this problem are considered further in the Conclusion to this thesis.

CHAPTER 5

CONCLUSION

5.1 WHERE NOW?

The performance of the Whiplash neural network appeared to be less successful, (albeit by a very marginal performance) than the Statistical Quantum Advisor. However, one should posit the question - in any given award, what error should be attributed to that particular judge? Arguably, judges do not hand down identical awards in identical cases. I have made the presumption throughout this thesis that if a number of different judges heard exactly the same case they would not necessarily hand down the same award. Equally, in different courtroom circumstances, the same judge might hand down a different award. Therefore the only measure I have been able to use is the particular award in a particular case. Alternative measures of accuracy might offer more favourable results. One such method might be to take cases with similar profiles, eliminate extreme awards and measure the success of the neural network in terms of its conformity with the overall band of cases. However, with the current data, the only measure available to me was a single case instance. Bands of awards may be a better indication than a single award criterion as a judgment of my success and undoubtedly my results would look far more impressive using this variable.

I have explored only one of the various possible neural network configurations, namely the back-propogation function using the Sigmoid Transfer function in a slow-learning software application. I believe that the most important conclusion to be drawn from this set of experiments is that, given the large number of variables that can be changed in a neural network, a superior search strategy must be found in order to discover the optimum combination of factors. With variables such as the learning rate, the number of hidden neurons, a variety of different data presentations, various transfer functions and the like, there are an immense number of configurations to be explored. I have built less than one hundred neural networks using what I believed to be the most obvious architectures. Yet there may be thousands, if not millions of different configurations of networks. Within this population there may (or may not) be one, ten, one hundred or one thousand networks that perform as well as (or better than) the statistical model. If we imagine a three dimensional graph of peaks and valleys which represent good and bad networks respectively, my models might be located in the foothills over on the North-East side of the graph, whereas the range of solutions actually reside somewhere in the mountain range in the South West. However, using my current search strategy I have no way of finding this possible range of solutions (if they exist), other than by stumbling upon them by luck more than judgment. As yet, there appears to be no procedure for identifying "good" networks other than through identifying those that adequately perform the task in question.

5.2 IMPROVING THE SEARCH STRATEGY

The only method available using the software that I used throughout this thesis is trial and error, by either wild guessing based on intuition (the so-called "Monte Carlo" method) or the more scientific method of hill-climbing - the process of testing potential improvements achieved by changes to each parameter. The problem with this latter search method is compounded by the fact that neural networks do not neccessarily react in a linear manner. That is to say that by changing the learning rate, the optimum number of hidden neurons may also have changed. Although more likely to succeed than wild guessing, this method is very time consuming. I have largely relied on hill-climbing using what I have read in software manuals, certain practice-orientated magazines[183] and sought advice from the FLAIR programmers[184] to set initial parameters and then vary a single parameter in both directions. This has not proved to be very successful. We must therefore explore some of the possible alternatives.

5.3 FUTURE RESEARCH

5.3.1 GENETIC ALGORITHMS

One alternative may be the use of genetic algorithms in conjunction with the neural network. These are another aspect of the so-called "evolutionary computation" research project that is affecting the Artificial Intelligence discipline. Like neural networks they are

[183] The magazines that I found to be most help have been PC-AI and AI expert.

[184] In particular Bruce Atherton.

finding immediate uses in the real world[185]. As the name suggests, the notion behind genetic algorithms is that it is possible to simulate certain characteristics of animal evolution in computer programs to find the "best fit" solution to a particular problem. Just as living species pass on certain crucial genetic information regarding characteristics through genetic strings called chromosomes during reproduction, mathematical algorithms can be passed down between generations and are then "mutated" or recombined to perform a given operation more effectively. Those that perform the best are selected and the remainder are disposed of. Genetic algorithms must have the following attributes; mutation, reproduction and selection. With respect to this problem, a population of neural networks is generated then the most successful are selected. These are then recombined/mutated and tested. Of this set, again the most successful are selected. These are, in turn, recombined and tested. This process continues until the search parameters have been satisfied and can take many generations before the optimum algorithms are found. Although not specifically designed for the process of function optimization, this is proving to be their primary application in engineering, manufacturing and elsewhere. In the context of this neural network research project I might wish to know the optimum number of hidden neurons, and the best learning rate. A search strategy to find this solution might be implemented using genetic algorithms. Hinton mentions the possibility of combining hill-climbing with genetic learning. He writes;

"It is possible to combine genetic learning with gradient descent (or hill-climbing) to get a hybrid learning procedure called "iterated genetic hill climbing" or "IGH" that works better than either learning alone. IGH is as a form of multiple restart hill climbing in which

[185] Examples include finance - Goldman Sachs of New York are known to be working in the area; in the military, the US Naval Surface Weapons Center at Bethesda, MD., are using genetic algorithms to advance propeller design.

the starting points, instead of being chosen at random, are chosen by "mating" previously discovered local optima.[186] "

This would seem to be the most obvious next step in the development of Connectionism in fuuzy domains such as legal reasoning. Otherwise we can resign ourselves to endless experimentation. Of course, even with this new search strategy, there is no guarentee that the networks would out-perform or even improve on previous neural networks or on the statistical program. It may be that I have stumbled unwittingly upon the optimum network, although I find such a scenario highly unlikely.

5.3.2 HYBRIDS - COMBINING EXPERT SYSTEMS WITH NEURAL NETWORKS

One of the problems that I have encountered in attempting to build neural networks in this domain has been that the network has no guidance as to which inputs might be more important than others. (This was, in part, a deficiency of the software. Some neural network software packages allow the user to manually increase the connection weightings on particular input neurons. The version of Brainmaker that I used, did not.) It may be that by providing the network with a degree of guidance would assist the network further. Also a knowledge-based system might be able to explain the reasoning contained within the network. We saw in Chapter 1 some of the limitations of expert systems; the facts that they do not have an adaptive reasoning functions, their performance does not improve with experience, etc. We have seen in Chapters 3 and 4 limitations of neural networks, namely that they have very limited explanatory functions, their requirement for a particular

[186] Hinton, G.E., *Connectionist Learning Procedures*, Artificial Intelligence, Vol.40., 1989, pp.189-234 at p.224.

presentation of data, etc. Liebowitz writes "Nowadays many people in the AI world believe that it's critical to integrate intelligent systems with such techniques as simulation and optimization, inter-active multimedia, neural networks and genetic algorithms.[187]" There is also some theoretical support for the notion;

> "Most probably, we think, the human brain is, in the main, composed of relatively small distributed systems, arranged by embryology into a complex society that is controlled in part (but only in part) by serial, symbolic systems that are added later. But the sub-symbolic systems that do most of the work from underneath must, by their very character, block all other parts of the brain from knowing much about how they work. And this, itself, could help explain how people do so many things yet have such incomplete ideas of how those things are actually done.[188]"

There are various types of hybrid architectures that may be utilized, such as loose-coupling, where the expert system and neural network stand alone but communicate via data files, tight coupling where they communicate via data passing[189]. Rule-based systems tend to work best with large areas of the problem domain but cannot always deal effectively with the boundary areas, where the subtleties of the domain prevent articulation in the form of rules. One possible application of a hybrid architecture in this domain might be the use of a rule-based system to deal with the major issues, so that rule-base defines the case as being a clear high-end award, (say $40,000-60,000) and then the neural network is used to provide a precise award within that band, based on its training with such cases.

[187] Leibowitz, J., Roll Your Own Hybrids, BYTE, July 1993, p.113.

[188] Minsky, M., and Papert, S., (epilogue to Perceptrons, Revised ed., Cambridge, MA, MIT Press, 1988)

[189] See Leibowitz, ibid., p.114.

5.4 PRACTICAL PROBLEMS OF NEURAL NETWORKS

5.4.1 FINDING A SOURCE OF DATA

Law is a domain rich in data. However, with the exception of social science data pertaining to the law (such as crime statistics) most of this data is contained in legal cases in a linguistic rather than numeric form. Computer programs have had great difficulty in dealing with issues of natural language and although inroads are being made to reduce linguistic knowledge to a more malleable form, we still have a way to go in this regard. For neural networks to be of significant further value in law, we need to discover a mechanism by which data can be easily transformed from linguistically indeterminate symbols to network-readable symbols without loosing any of their subtlety or character. As yet the only method is to put all the factors of the case into a numerically precise database. In the process much of the subtleties and the linguistic indeterminacies of the case are lost. We would be assisted by a systematic "fuzzification" of linguistic factors by using methods such as that by which I attempted to fuzzify the quantum values in the whiplash database in Chapter 4. However, this will involve a significant research undertaking to determine the optimum database structure. There would be a significant number of questions to be answered such as - should all inputs be fuzzified, if so to what extent, if not then which factors should be manipulated, etc.

5.4.2 TECHNICAL - EXTERNALIZING THE INTERNALITIES

Researchers in the field of law and economics often comment on their struggle to quantify and build into their equations those factors that evade numerical quantification. They have in mind such factors as a view of an ocean from land, the enjoyment people get from walking across an open landscape undisturbed by development and many other such intangibles. They call this process of attempted quantification "internalizing the externalities." Connectionist have the exact opposite problem; what I have called "externalizing the internalities." The knowledge, value, or intellectual property is contained in the weightings on the neurons in numerical form. We saw in Chapter 2 that the Paris-based trio of Bochereau et al.[190] achieved some success in extracting symbolic rules (albeit of very minimal use and they did not describe their methodology for doing so.) Even if we have discerned a meaning from the weighting of connections, this does not necessarily mean that we have discovered *the* relationship between inputs and outputs. It means that we have found *a* relationship, possibly one of many that may satisfy the requirements of the data. It is clear that judges view cases differently. It may also be that different lawyers reason in different ways. For instance, some people may rely more on their intuitive nature, others may prefer the rational logical analysis. There is no reason to assume that lawyers are any different. Moreover, there is no empirical proof that shows that judges necessarily come to similar conclusions by the same reasoning processes. Thus any notions that neural networks can in some way discover the "deep structure" of an area of law should be dispelled.

[190] See note [ ] of Chapter 2.

5.4.3 INCOMPLETE DATA

There are two distinct problems in this area. First, there are in the region of 60 potential inputs. It would be unfair to imagine that adjusters would be prepared to input all 60 factors in order to discover a quantum figure. Equally it is highly unlikely that the adjuster would have all the information at his or her fingers. Very often the purpose of predicting a possible quantum figure is to avoid going to the trouble and expense of accumulating all of the facts (the input data) in the first place. A neural network can function in such circumstances although the output that it provides is likely to be far more generalized.

The second problem relating to incomplete data is that only 5% of all actions actually go to court. Therefore the database contains only a very truncated "snapshot" of all disputes involving whiplash in British Columbia. A database showing all awards settled or litigated would have made a far richer data source.

There are other areas of research that would make interesting projects. One such project would be the development of a network from the bottom up. i.e. take a very small number of cases, (say 10) all from a given year and teach the network with these alone. Then add another 10 cases and let the network adjust its performance accordingly. This would continue until all 630 training cases are trained upon. I was not able to perform this experiment because aspects of the software prevented me from doing so. Another interesting experiment would be to isolate those cases that involve certain judges and train networks on only those cases, then compare the results and see if they correspond with the opinions of lawyers experienced in appearing before those judges.

5.5 THE SOCIOLOGICAL CONTEXT

5.5.1 WOULD COMPANIES WANT THESE MACHINES?

The reality of corporate culture is that, even with significant advances in the development of neural networks, the level of sophistication is unlikely (at least in the near future) to be able to replace the expertise of an adjuster with 10 or so years experience at ICBC. The best opportunities for the exploitation of this type of software may well be with its use by more junior adjustors. Although they might appreciate such innovations with open arms, not least because the software could provide the user with an empirically sound "second-guess." However, their managers may be less welcoming. Supervisors may not want their trainees to have a "crutch" to lean on, to justify their opinions; they would want their trainees to develop an expertise of their own. (This problem is not exclusive to neural networks - it is likely to be reflected in the adoption (or otherwise) of any other form of software that purports to offer substantive advice, such as expert systems.) Moreover, if the neural network is trained on a training set that includes all cases including highly unusual awards, the likelyhood is that, if adjusters blindly follow the network output as if its answers were "carved in stone," then those quirks of precedent will be re-inforced and replicated. The network will effectively loop those problem cases into its internal makeup. There would be a major perception problem to be overcome in any commercial application. Many computer users still do not realize that a computer is no more than a very high speed idiot with a well-developed memory.

5.5.2 WOULD SOCIETY WANT THESE MACHINES?

There are a number of political, ideological and ethical implications of developing machines that appear to predict outcomes in legal case, not least of which is the Orwellian implication for machine-made justice. These have been explored elsewhere[191] and are mentioned only for the sake of completeness. Mainstream jurisprudential commentators have been quick to pick up on this theme. Lloyd's Introduction to Jurisprudence offers these comments:

> "It is easy to throw up one's hands in horror, but is the "inscrutable" jury such a rational institution. You can at least program a computer with biases acceptable to the community, or at least to a majority of it, but juries, and judges for that matter retain their own caprices.[192]"

After 20 years of research into computers and law and 10 years of using expert systems, we remain a very great distance from these Orwellian scenarios and as the information revolution continues, enhancing access to information further, the likelyhood of this events coming to pass seems to be, if anything receding. Neural networks, for all their advances seem unlikely to alter this forecast dramatically.

[191] See Spengler,J.J., *Machine Made Justice - Some Implications* in Baade, H.W.(ed.), *Jurimetrics*, New York, Basic Books, 1963 and Beutel, F.K., *Experimental Jurisprudence*, Lincoln, University of Nebraska Press, 1957.

[192] Lloyd, of Hampstead, and Freeman, M.D.A., *Lloyd's Introduction to Jurisprudence*, London, Stevens & Sons, 1985, p.703.

5.5.3 THE NATURE OF THE NEGOTIATION PROCESS

Adjusters act on behalf of ICBC (possibly with the assistance of Counsel) against the Plaintiff and his/her lawyer. The reality of the negotiation process would appear to be that the adjuster predicts what the claim is worth objectively and often offers less because they know that the lawyer in all likelihood is going to ask for much more. This is commonly described as "low-balling" the offer. If neural networks such as the one developed in this thesis were to come into use by ICBC and the lawyers acting on behalf of their Plaintiffs were to discover that such a tool existed, they would undoubtedly wish to know the conclusion reached by such a system. Revelation of that figure might seriously undermine the traditional negotiation process based on the adversarial system. The response to this argument is therefore to offer the system to judges. However, they too are likely to possess experience superior to the pattern recognition powers of the network, at least in its current form. The network might, however, act as an intelligent assistant. A divergent award produced by such a neural network might cause a judge to reconsider his or her position.

There are further problems. The neural network (in its present form) cannot take account of sociological issues such as the particular dynamics of the social relationships between a given lawyer and adjuster, the role of the doctor in the trial process, etc. Some doctors are not supportive and may not perform well on the witness stand. Others are considered to be very credible. Equally adjusters may have some bias either for or against the Plaintiff depending on how they perform through the negotiation process. Who the judge may also have a dramatic effect on the outcome. Some judges are known to be sympathetic to the complaints of the Plaintiff; others are known to be highly skeptical[193]. Lawyers often tell their clients that going to court is a lottery. Thus, the data that I have used in this experiment i.e. injuries to the Plaintiff represent only half of the equation. The sociological interaction between key participants in the process cannot be analysed numerically, (at least not yet.)

5.6 AI AND THE MEDIA

For the sociologist, AI has been an interesting scientific paradigm. The advent of a new technology is heralded in the popular press as much as in the scientific journals. There then follows an enormous amount of media interest in the potential applications of this new techology. The problem with the popular media is that they tend to speak of technological advances as if they have already occurred rather than reporting on what a given scientist believes to be feasible with the technology. After the media interest has peaked, nothing further is heard for sometime while the real research continues. During this period, articles appear in the popular press asking questions along the following lines; Whatever happened to that new technology, X? Finally research begins to filter through which suggests that X is not the problem solving panacea that everyone thought it was. This thesis is part of that research momentum as the crest of media wave begins to break. It endorses the use of the technology but is realistic about the problems it presents. What is so remarkable is that one need know very little about the domain in issue and nothing

[193] One Vancouver lawyer stated in interview "If we find out the name of the judge the previous day and we know he is one who will not be interested in what we have to say, we immediately phone the adjuster and accept their offer." How can any computer program, neural network or otherwise, capture the dynamics of such human interaction?

about computers to build a program that has some predictive capacity, albeit currently insufficient for utilization. Neural networks are not intelligent. As the unnamed author of a recent article on neural networks used in financial forecasting, in Wired magazine so rightly noted "..the net will just as merrily try to find relationships between the cycles of the moon and inflation as between wage rates and inflation[194]." However, researchers should be optimistic about the potential of these machines. Like Leith[195] I started out on this research project relatively optimistically to build a neural network that could out-perform former programming methods. However during this building I too grew skeptical of being able to handle legal knowledge in this type of computer system. Leith describes his skepticism thus: "real legal skill is a much more amorphous quantity than the builders of legal expert systems believe.[196]" This thesis has shown that one cannot simply present a database of numeric factors which represent a complex set of linguistic, social and political relationships that make up a domain of law, to a neural network and hope that it will be able to work everything out. Bench-Capon has shown that it is possible with hypothetical domains but the real world remains nebulous[197]. My skepticism towards the ability of neural networks to deal with legal knowledge remains in part for all the reasons I have outlined above (despite some encouraging results.) However, this skepticism is tempered by the realization that the challenge of finding the optimum network(s) also present marvellous opportunities to explore the complexity of relationships between sociological factors in law; we have in our possession a far more complex tool than expert

[194] Wired Magazine, September 1994, p.39.
[195] See note 83 in Chapter 2.
[196] See note 83 in Chapter 2.
[197] See note 132 in Chapter 3.

systems. This is supplemented by an appreciation that I have only scratched the surface of neural network theory and application. Hypothetically, at least, these political, sociological and economic factors that affect legal reasoning and decision-making may be built into the network. Such research offers exciting potential for both lawyer and social scientist. I have been able to produce results almost of equal value as those produced by the Statistical Quantum Advisor with a relatively simple piece of software, no programming experience and no familiarity with the domain of whiplash. Furthermore, if the developments that I have outlined above are utilized i.e. the use of genetic algorithms to optimize searching, the use of banding as a measure of success and the fuzzification of input data then I believe that this technology could be utilized in a wide number of instances, faciliatating earlier settlements, substantial cost savings through alternative dispute resolution and out-of-court settlements through enhanced access to complex legal information. Plaintiffs are generally willing to trade-off a certain amount of accuracy for substantial savings in information costs. This process could be transformed by the application of neural networks. For all its pitfalls, I believe that I have identified an important new area for research in the discipline of Law and Information Technology and am both confident and excited about its future.

BIBLIOGRAPHY

BOOKS & THESES

Anderson, J.A., and Rosenfeld, E., *Neurocomputing: Foundations of Research*, Cambridge, MA., MIT Press, 1988.

Aparicio, M., & Levine, D., *Neural Networks for Knowledge Representation and Inference*, Hillsdale, NJ., Lawrence Erlbaum Associates, 1994.

Arbib, M.A., and Robertson, J.A., (eds.), *Natural and Artificial Parallel Computation*, Cambridge, MA., MIT Press, 1990.

Ashley, K., *Modelling Legal Argument: Reasoning with Cases and Hypotheticals*, unpublished Ph.D thesis, University of Massachussetts, 1988.

Baade, H.W., (ed.), *Jurimetrics*, New York, Basic Books, 1963.

Beutel, F.K., *Experimental Jurisprudence*, Lincoln, University of Nebraska Press, 1957.

Braithwaite, M., *Prolegamena to a Post-Modernist Theory of Law*, unpublished LL.M. thesis, University of British Columbia, October 1993.

Cardoza, B., *The Nature of the Judicial Process*, New Haven, Yale University Press, 1921.

Continuing Legal Education Society of British Columbia, *ICBC - Motor Vehicle Accident Claims*, Vancouver, B.C., 1988.

Continuing Legal Education Society of British Columbia, *ICBC Defence Materials*, Vancouver B.C., 1993.

Coval, S., & Smith, J.C., *Law and its Presuppositions: Actions, Agents and Rules*, London, Routledge and Kegan Paul, 1986.

Deedman,G.C., *Building Rule-Based Expert Systems in Case-Based Law*, unpublished LL.M. thesis, University of British Columbia, April, 1987.

Department of Trade & Industry, *Neural Computing- Learning Solutions*, London, 1993.

Dreyfus, H.L., *What Computers Can't Do*, New York, Harper & Row, 1972.

Dreyfus, H.L., *What Computers Still Can't Do*, Cambridge, MA., MIT Press, 1992.

Dreyfus, H.L., & Dreyfus, S.E., *Mind Over Machine*, New York, Free Press, 1986.

Dworkin, R., *Taking Rights Seriously*, Cambridge, MA., Harvard University Press, 1977.

Foreman, S.M., & Croft, A.C., *Whiplash Injuries - The Cervical Accelaration /Deceleration Syndrome*, Baltimore, Williams & Wilkins, 1988.

Frank, J., *Courts on Trial,* Princeton, Princeton University Press, 1949.

Frank, J., *Law and the Modern Mind*, Garden City, NY., Doubleday Press, 1963.

Gardner, A.v.d.L., *An Artificial Intelligence Approach to Legal Reasoning*, Cambridge, MA., MIT Press, 1987.

Goodrich, P., *Reading the Law: A Critical Introduction to Legal Method and Techniques*, London, Basil Blackwell, 1986.

Graubard, S., (ed.*) The Artificial Intelligence Debate: False starts, Real foundations*, New York, Basic Books, 1988.

Harmon, P., and King, D., *Expert Systems, Artificial Intelligence in Business*, New York, John Wiley and Sons, 1985.

Hart, H.L.A., *The Concept of Law*, Oxford, Clarendon Press, 1961.

Hebb, D.O., *The Organization of Behavior*, New York, Wiley, 1949.

Holmes, O.W., *Collected Legal Paper*, New York, Harcourt, Brace & Co., 1920.

Holton,G., *Thematic Origins of Scientific Thought; Keplar to Einstein*, Cambridge, Harvard University Press, 1973.

Insurance Corporation of British Columbia, *Comparative Aspects of the Insurance Corporation of British Columbia and Private Sector Insurance Companies in Canada*, Vancouver, B.C., 1990.

Insurance Corporation of British Columbia, *Annual Report*, Vancouver, B.C., 1993.

James, W., *Psychology*, New York, Holt, 1890.

Kevelson, R., (ed.*) Peirce and the Law: Issues in Pragmatism, Legal Realism and Semiotics*, New York , Peter Lang, 1991.

Kosko, B., *Fuzzy Thinking*, New York, Hyperion, 1993.

Kursweil, R., *The Age of Intelligent Machines*, Cambridge, MA., MIT Press, 1990.

Lawrence. J., *Introduction to Neural Networks*, California Scientific Software Press, Nevada City, CA., 1993.

Leith, P., *The Computerised Lawyer - a guide to the use of computers in the legal profession*, London, Springer-Verlag, 1991.

Levi, E.H., *An Introduction to Legal Reasoning*, Chicago, University of Chicago Press, 1949.

Leyh, G, (ed.) *Legal Hermeneutics: Theory, History and Practice*, Berkeley, University of California Press, 1992.

Llewellyn, K., *Jurisprudence; realism in theory and practice*, Chicago, University of Chicago Press, 1962.

Lloyd, of Hampstead, and Freeman, M.D.A., *Lloyd's Introduction to Jurisprudence*, London, Stevens & Sons, 1985.

McNeill, D., & Freiberger, P., *Fuzzy Logic* (forthcoming.) Section quoted in Success Magazine, September 1994.

Minsky, M., *Semantic Information Processing*, Cambridge, MA., MIT Press , 1968.

Minsky, M., and Papert, S., *Perceptrons*, Revised ed., Cambridge, MA, MIT Press, 1988.

Mital, V., and Johnson, L., *Advanced Information Systems for Lawyers*, London, Chapman Hall, 1992.

Ministry of the Solicitor General, Motor Vehicle Branch, *Motor Vehicle Statistics*, Victoria, B.C., 1989.

Naylor, C., *Build Your Own Expert System*, , London, Halstead Press, 1983.

Neumann, J. von, *The Computer and the Brain*, New Haven, Yale University Press, 1958.

Pagels, H., *The Dreams of Reason*, New York, Simon and Schuster, 1988.

Partridge, D. & Wilks, Y., *The Foundations of Artificial Intelligence - A Sourcebook,* Cambridge, Cambridge University Press, 1990.

Penrose, R., *The Emperor's New Mind,* New York, Oxford University Press, 1989.

Posner, R.A., *The Problems of Jurisprudence*, Cambridge, MA., Harvard University Press, 1990.

Raz, J., *The Authority of Law, essays on law and morality*, Oxford, Clarendon Press, 1979.

Reed, C., *Computer Law*, London, Blackstone Press, 1990.

Ringle, M., *Philosophical Perspectives on Artificial Intelligence*, The Harvester Press, Brighton, Sussex, 1979.

Rumelhart, D.E, and McLelland, J.L, (eds.) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Cambridge, MA., MIT Press, 1986.

Sayegh, N.S., *Philosophy and Philosophers*, Ottawa, Academy Books, 1988.

Schilpp, P.A., *The Philosophy of Karl Popper*, La Salle, IL., Open Court, 1974.

Susskind, R.E., *Expert Systems in Law - A Jurisprudential Inquiry*, Oxford, Clarendon Press, 1987.

Trial Lawyers Association of British Columbia, *Conducting a Whiplash Case: Winning Trial Strategies*, Vancouver, B.C., Canada, February 23, 1990.

Vandenburghe, G.P.V., (ed.) *Advanced Topics in Law and Information Technology*, Hillsdale, NJ., Lawrence Erlbaum, 1990.

Wahlgren,P., *The Automation of Legal Reasoning - A study on Artificial Intelligence and Law*, Deventer, Kluwer Law and Taxation Publishers, 1992.

Waterman, D.A., *A Guide to Expert Systems*, Menlo Park, CA., Addison-Wesley Publishing Co., 1986.

Williams, G., *Learning the Law*, 5th ed., London, Stevens & Sons, 1978.

Winston, P.H., *Artificial Intelligence*, Reading, Mass. USA, Addison-Wesley Publishing Co., 1992.

DICTIONARY

The New Lexicon Webster's Dictionary of the English Language, New York, Lexicon Publications, Inc., 1988.

ARTICLES

Anderson, J.A., *A simple neural network generating an interactive* memory, Mathematical Biosciences, 14: pp.197-220.

Ashley, K., *Reasoning by Analogy: A Survey of Selected AI Research with Implications For Legal Expert Systems,* in Walter, C., *Computing Power and Legal Reasoning,* St.Paul, MN., West Publishing Co., 1985. pp.105-130.

Belew, R.K, *A Connectionist Approach to Conceptual Information Retrieval* , Proceedings of the First International Conference on Artificial Intelligence and Law, Boston, MA., 1987, pp.116-126.

Bench-Capon, T.E., *Neural Networks and Open Texture*, Proceedings of the Fourth International Conference on Artificial Intelligence and Law, Amsterdam, ACM Press, 1993, pp.292-297.

Bench-Capon, T.E., & Sergot, M., *Towards a Rule-Based Representation of Open Texture in Law*, Journal of Law and Information Technology, 1992.

Bing, J, *The law of the books and the files: possibilities and problems of legal information retrieval*, in G.P.V. Vandenburghe (ed.) *Advanced Topics in Law and Information Technology*, Hillsdale,NJ, Lawrence Erlbaum, 1990, pp.151-182.

Blair, D.C., & Maron, M.E., *An Evaluation of Retrieval Effectiveness for a Full-Text Document Retrieval System*, Communications of the ACM, New York, 28,3, 1985, pp.289-299.

Bochereau, L., Bourcier, D., and Bourgine, P., *Extracting Legal Knowledge By Means of A Multi-Layered Neural Network Application to Municipal Jurisprudence*, , Proceedings of the Fourth International Conference on Artificial Intelligence and Law , Amsterdam, ACM Press, 1993, pp.288 - 296.

Brown,D., “The Third International Conference of Artificial Intelligence and Law; A Report and Comments”, Journal of Law and Information Science, Vol.2, No.2, 1991, pp.223- 239.

Buchanan, B.G., and Headrick,T.E., “*Some Speculation About Artificial Intelligence and Legal Reasoning* Stanford Law Review, 23, November 1970, pp.40-61.

Chomsky, N., *A Review of Verbal Behavior by B.F.Skinner* in Language, 35, 1959, pp.26-58.

Churchland,P.M., *Representation and high-speed computation in neural networks*, in Partridge, D, & Wilks, Y., *The Foundations of Artificial Intelligence - A Sourcebook*, Cambridge, Cambridge University Press, 1990.

Classe, A., *Little Grey Cells*, Accountancy, May 1993, v.111 (n.1197), p.67.

Crowe, H.E., *Injuries to the Cervical Spine*, paper presented at the meeting of the Western Orthopaedic Association, San Francisco, 1928.

Dreyfus, S.E., and Dreyfus, H.L., *Towards a reconcilation of phenomenology and AI*, in Partridge, D., and Wilks, Y., *The foundations of artificial intelligence - A sourcebook*, Cambridge, Cambridge University Press, 1990, pp.396-410.

Fernhout,F., *Using a Parallel Distributed Processing Model As Part Of A Legal Expert System*. 1 Pre-Proceedings of the Third International Conference on Logica, Informatica, Diretto , Florence, Martino,A.A. (ed.), 1989, pp.255-268.

Fodor, J.A., *Why there STILL has to be a language of thought,* in Partridge, D, & Wilks, Y., *The Foundations of Artificial Intelligence - A Sourcebook,* Cambridge, Cambridge University Press, 1990, pp.289-305.

Fodor, J., and Pylyshyn, Z.W., *Connectionism and cognitive architecture: A critical analysis*, Cognition, 28, 1988, pp.3-71

Fuller, L., *Positivism and Fidelity - A Reply to Professor Hart*, 71 Harvard Law Review 1958, pp.630-672.

Gideon, T.D., and Mital,V., *Information Retrieval in Law using a Neural Network integrated with Hypertext*, Proc. of International Joint Conference on Neural Networks (IJCNN), Singapore, 1991, pp.1819-1824.

Hart, H.L.A., *Positivism and The Separation of Law and Morals*, 71 Harvard Law Review, 1958, pp.593-629.

Hertz, D., *Thinking Computers*, The Miami Herald, February 5, 1989, p.

Hinton, G.E., *Connectionist Learning Procedures*, Artificial Intelligence, Vol.40., 1989, pp.189-234.

Holland, J., *Complex Adaptive Systems,* Daedalus, Vol.121, No.1, Winter 1992, pp.17-30.

Israel, D.S., *The role of logic in knowledge representation,* Computer, Vol. 16, No.10, 1983.

Kantrowitz, M., Horstkolte,E., and Joslyn, C., *Answers to Frequently Asked Questions about Fuzzy Logic and Fuzzy Expert Systems,* comp.ai.fuzzy, July ,1994, ftp.cs.cmu.edu, usr/ai/pubs/faqs/fuzzy/fuzzy.faq.

Kinoshita, J., Neural Nets at Work, Scientific American, Vol.259, No.5, pp.134-135.

Klimasauskas, C., *Applying Neural Networks,* PC AI, May-June 1991, pp.20- 24

Kohonen, T., *Corrolation Matrix Memories,* IEEE Transactions on Computers, C-21: pp.353-359

Kowalski, A., *Case-Based Reasoning and the Deep Structure Approach to Knowledge Representation,* Proceedings of the Third International Conference of Artificial Intelligence and Law, ACM Press, Oxford, 1991, pp.

Kowlaski, R., and Sergot, M., *The Use of Logical Models in Legal Problem Solving,* Ratio Juris, Vol.3, No.2, July 1992, pp.201-218.

Kuhn, T.S., *The Structure of*, in *Scientific Revolutions,* Chicago, University of Chicago Press, 1970, 2nd edition *International Encyclopedia of Unified Sciences,* Vol 2, Number 2.

Lashbrooke, E.C., *Legal Reasoning and Artificial Intelligence*, Loyola Law Review, Vol.34, No.2, 1988, pp.287-310.

Leibowitz, J., *Roll Your Own Hybrids,* BYTE, July 1993, pp.113-115.

Loevinger, L., *Jurimetrics - The Next Step Forward,* Minnesota Law Review, Vol.33, 1948, pp.455- 493.

McClelland, J.L., *Can Connectionist Models Discover the Structure of Natural Language?* in *Minds ,Brains and Computers,* Norwood, NJ., Ablex Publishing Co., 1992, Chapter 7, pp.168-189.

McCulloch, W.S., and Pitts, W., *A Logical Calculus of the Ideas Immanent in Nervous Activity,* 5 Bulletin of Mathematical Biophysics, 1943, pp.127 -147.

MacCrimmon, M.T., *Facts, Stories and the Hearsay Rule*, Pre-Proceedings of the III International Conference on Logica, Informatica, Diretto, Florence, Martino, A.A., (ed.) 1989, pp.461-475.

MacLean,P., The triune brain,emotion and scientific bias, In F.O.Schmitt (ed.) *The Neurosciences: Second study program*, New York, Rockefeller University Press, 1970, pp.336-349.

Meldman, J.A., *A Structural Model for Computer-Aided Legal Analysis,* Rutgers Computer and Technology Law Journal, 6, 1977, pp.27-71.

Mital V., and Gedeon, T.D., *A Neural Network Integrated with Hypertext for Legal Document Assembly,* Proceedings of the Twenty-Fifth Hawaii International Conference on Systems Sciences, Hawaii, IEEE Computer Society Press, 1992, pp.533-539.

Opdorp, van,G.J., and Walker,R.F., *A Neural Network Approach to Open Texture*, in Kasperson,H.W.K., & Oskamp, A., *Amongst Friends in Computers and Law*, Deventer, Boston, Kluwer Law and Taxation Publishers, 1990, pp.279-309.

Philipps, L., *Distribution of Damages in Car Accidents Through The Use of Neural Networks*, 13, Cardoza Law Review, 1991, pp.987-1000.

Philipps, L., *Are Legal Decisions Based on the Application of Rules or Prototype Recognition? Legal Science on the Way to Neural Networks,* 2 Pre-Proceedings of the Third International Conference on Logica, Informatica, Diretto , Florence, Martino,A.A. (ed.), 1989.

Raghhupathi, W., Levine,D.S., Bapi, R.S., and Schkade,L.L., *Towards Connectionist Representation of Legal Knowledge*, in Aparicio,M., and Levine,D.S, *Neural Networks for Knowledge Representation and Inference*, Hillsdale, NJ., Lawrence Erlbaum, 1994, pp.267 - 282.

Raghupathi, W., Schkade, L.L., Bapi, R.S., and Levine, D.S., *Exploring Connectionist approaches to Legal Decision Making*, Behavioral Science, 36, 1991, pp.133-139.

Rau, L.F., *Knowledge organization and access in a conceptual information system*, 23 Information Processing and Management, pp.269-283.

Rose, D.E, and Belew, R.K, *Legal Information Retrieval: a Hybrid Approach*, Proceedings of the First International Conference on Artificial Intelligence and Law, Boston, ACM Press, 1989, pp.138-146.

Rissland, E., and Ashley, K., *HYPO: A Precedent-Based Legal Reasoner*, in Vandengurghe, G.P.V., *Advanced Topics of Law and Information Technology*, Hillsdale, NJ, Lawrence Erlbaum, 1989, pp.327- 356.

Rosenblatt,F., *The Perceptron: a probabilistic model for information storage and organization in the brain*, Psychological Review 65,1958, pp.386-408.

Schneider, W., *Connectionism:Is it a paradigm shift for psychology?* Behavior Research Methods, Instruments and Computers, 19, 1988, pp.73-83.

Schwartz,E.I & Treece, J.B, *Smart Programs Go To Work*, in Business Week, March 2, 1992.

Sergot M.J., Sadri, F., Kowalski, R.A., Kriwaczek,F., Hammond, P., and Cory, H.T, *The British Nationality Act as a logic program*, Communications of the ACM, 29, pp.370-383.

Simon, H.A., & Newell, A., *Heuristic problem Solving: The Next Advance in Operations Research*, Operations Research, Vol 6, 1958.

Smith, J.C, & Gelbart,Daphne, *Beyond Boolean Search: FLEXICON, A Legal Text-based Intelligent System*, Proceedings of the Third International Conference on Artificial Intelligence and Law, Oxford, England, ACM Press, 1991, pp.225-233.

Smith, J.C, *Law, Language and Philosophy*, Best-Printers, Vancouver, 1968.

Smith, J.C., *Review of The Automation of Legal Reasoning*, Stockholm, 5 Juridisk Tidskift, Nr.1, 1993-1994, pp.247- 256.

Smolensky, P., *Connectionism and the Foundations of AI*, in Partridge, D, & Wilks, Y., *The Foundations of Artificial Intelligence - A Sourcebook*, Cambridge, Cambridge University Press, 1990, pp.306-326.

Somers.J. et al., *Catching Intruders in a Neural Net*, Security Management, Feb 1994, v.38 (n2), pp.68-74.

Spengler, J.J., *Machine Made Justice - Some Implications* in Baade, H.W., *Jurimetrics*, 1960, pp.

Stone, J., *Fallacies of the Logical Form in English Law*, in Sayre, P., *Interpretations of Modern Legal Philosophy*, 1947, pp.696-735.

Susskind, R.E, *Expert Systems in Law: Theory and Practice*, Cambridge Lectures, 1987.

Susskind, R.E., *Expert Systems in Law: A Jurisprudential Approach to Artificial Intelligence and Legal Reasoning,* Modern Law Review, 49, 1987, pp.168-183

Swales, G.S., and Yoon,Y., *Applying Artificial Neural Networks to Investment Analysis,* Financial Analysts Journal, Sept -Oct, 1992, v.48 (n5), pp.78-81.

Thagard, P., *Explanatory coherence*, in Behavioral and Brain Sciences, 1989, 12, pp.435-502.

Warner, D.R., *A Neural Network-Based Law Machine: Initial Steps*, 18 Rutgers Law and Technology Journal, 1992, pp.51-

Warner, D.R, *Towards a Simple Law Machine*, Jurimetrics Journal, 29, 1989, p451-487.

Wilks, Y., *Some comments on Smolensky and Fodor*, in Partridge, D, & Wilks, Y., *The Foundations of Artificial Intelligence - A Sourcebook*, Cambridge, Cambridge University Press, 1990, pp.327-336.

Zadeh, L., *Fuzzy Sets,* Infomation and Control, 8, 1965, pp.338-353.

Zadeh, L., *Commonsense Knowledge Representation Based on Fuzzy Logic*, Computer, Vol.16, No.10, 1983.

unnamed author, unnamed article in Wired, September, 1994, p.39.

APPENDIX A

TEST2.PRG

(* symbol denotes commentary)

```
SET ECHO OFF
SET TALK ON
SELECT 1
USE TARGET2 ALIAS T
SELECT 2
USE BRNMAK3 ALIAS B
SELECT 1

RET=.F.
RET1=.F.
RET2=.F.

DO WHILE .NOT. EOF()
  ? RECNO()


  RET=T->DMGS_INCL .OR. T->MULT_ACC = Y .OR. T->OTHER_INJ = 1 .OR. T-
>OTHER_INJ = 2 .OR. T->OTHER_INJ = 3 .OR. T->SUBSQ_INJ = 1 .OR. T-
>SUBSQ_INJ = 2 .OR. T->SUBSQ_INJ = 3   .OR. T->SUBSQ_INJ = 4 .OR. T-
>SUBSQ_INJ = 5 .OR. T->THINSKULL = 1
  RET1= T->THN_SKULL = YC .OR. T->THN_SKULL = YG .OR. T->PRE_EXIST =
1 .OR. T->PRE_EXIST = 2 .OR. T->PRE_EXIST = 3 .OR. T->PRE_EXIST = 4 .OR.
T->PRE_EXIST = 5 .OR. T->PRE_EXIST = 6
  RET2= T->PRE_EXIST = 7 .OR. T->PRE_EXIST = 8 .OR. T->PRE_EXIST = 9 .OR.
T->PRE_EXIST = 10 .OR. T->PRE_EXIST = 11 .OR. T->PRE_EXIST = 12 .OR. T-
>PRE_EXIST = 13 .OR. T->PRE_EXIST = 14 .OR. T->PRE_EXIST = 15


 IF .NOT. RET .AND. .NOT. RET1 .AND. .NOT. RET2
   SELECT 2
   APPEND BLANK
   REPLACE B->NAME WITH T->NAME
   REPLACE B->JUDGE WITH T->JUDGE
   REPLACE B->DATE WITH YEAR (T->DATE)-1984
   REPLACE B->INJDATE WITH YEAR (T->INJURYDATE)-1984

   IF T->FTWORKLOSS >0
     REPLACE B->FTWOLO WITH T->FTWORKLOSS
```

```
ELSE
  IF T->FTWORKLOSS = -2
    REPLACE B->FTWOLO WITH 99
  ELSE
    IF T->FTWORKLOSS = -1
      REPLACE B->FTWOLO WITH (B->DATE-B->INJDATE)*12
    ELSE
      IF T->FTWORKLOSS = -3
        REPLACE B->FTWOLO WITH 0
      ELSE
        IF T->FTWORKLOSS = -4
          REPLACE B->FTWOLO WITH 0
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF

IF T->PTWORKLOSS >= 0
  REPLACE B->PTWOLO WITH T->PTWORKLOSS
ELSE
  IF T->PTWORKLOSS = -2
    REPLACE B->PTWOLO WITH 99
  ELSE
    IF T->PTWORKLOSS = -1
      REPLACE B->PTWOLO WITH (B->DATE-B->INJDATE)*12
    ELSE
      IF T->PTWORKLOSS = -3
        REPLACE B->PTWOLO WITH 0
      ELSE
        IF T->PTWORKLOSS = -4
          REPLACE B->PTWOLO WITH 0
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF

IF T->DISABILITY >0
  REPLACE B->DISAB WITH T->DISABILITY
ELSE
  IF T->DISABILITY = 0.00
    REPLACE B->DISAB WITH 0
  ELSE
    IF T->DISABILITY = -1
```

```
      REPLACE B->DISAB WITH (B->DATE-B->INJDATE)/4
    ELSE
      IF T->DISABILITY = -2
        REPLACE B->DISAB WITH 99
      ELSE
        IF T->DISABILITY = -3
          REPLACE B->DISAB WITH 0
        ELSE
          IF T->DISABILITY = -4
            REPLACE B->DISAB WITH 0
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF

IF T->RN = ""
   REPLACE B->RECOV WITH 0
ELSE
  IF T->RN = -1
    REPLACE B->RECOV WITH (B->DATE-B->INJDATE)*12
  ELSE
    IF T->RN = -2
      REPLACE B->RECOV WITH 99
    ELSE
      IF T->RN = -3
        REPLACE B->RECOV WITH 0
      ELSE
        IF T->RN = -4
          REPLACE B->RECOV WITH 0
        ELSE
          IF T->RN >0
            REPLACE B->RECOV WITH T->RN
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF

REPLACE B->SEVER WITH T->SN
REPLACE B->QUANT WITH T->QU

IF T->ATHLETIC >0
```

```
  REPLACE B->ATHLETIC WITH 1
ELSE
  REPLACE B->ATHLETIC WITH 0
ENDIF

IF T->RECREATION >0
  REPLACE B->RECREAT WITH 1
ELSE
  REPLACE B->RECREAT WITH 0
ENDIF

IF T->HOBBY >0
  REPLACE B->HOBBY WITH 1
ELSE
  REPLACE B->HOBBY WITH 0
ENDIF

IF T->SOCIAL >0
  REPLACE B->SOCIAL WITH 1
ELSE
  REPLACE B->SOCIAL WITH 0
ENDIF

IF T->SEXUAL >0
  REPLACE B->SEXUAL WITH 1
ELSE
  REPLACE B->SEXUAL WITH 0
ENDIF

IF T->LIFESTYLE >0
  REPLACE B->LIFESTYLE WITH 1
ELSE
  REPLACE B->LIFESTYLE WITH 0
ENDIF

IF T->HOUSEWORK >0
  REPLACE B->HOUSE WITH 1
ELSE
  REPLACE B->HOUSE WITH 0
ENDIF

IF T->PSYCH >0
  REPLACE B->PSYCH WITH 1
ELSE
  REPLACE B->PSYCH WITH 0
```

```
ENDIF

IF T->DEPRESS >0
  REPLACE B->DEPRESS WITH 1
ELSE
  REPLACE B->DEPRESS WITH 0
ENDIF

IF T->PERSONLTY >0
  REPLACE B->PERSON WITH 1
ELSE
  REPLACE B->PERSON WITH 0
ENDIF

IF T->JOB_CHANGE >0
  REPLACE B->JBCHANG WITH 1
ELSE
  REPLACE B->JBCHANG WITH  0
ENDIF

REPLACE B->NECKP WITH T->NECKPAIN
REPLACE B->CPS WITH T->CHRONPAIN
REPLACE B->LBP WITH T->LOW_BACK
REPLACE B->HEADAC WITH T->HEADACHES
REPLACE B->MYOFAS WITH T->MYOFASCIAL
REPLACE B->TMJ WITH T->TMJ
REPLACE B->SPASMS WITH T->SPASMS
REPLACE B->SHOULD WITH T->SHOULDER
REPLACE B->FIBROS WITH T->FIBROSITIS
REPLACE B->NUMBT WITH T->NUMBTINGLE
REPLACE B->ANKYL WITH T->ANKYLOSING
REPLACE B->FATIGUE WITH T->FATIGUE
REPLACE B->SLEEP WITH T->SLEEPLOSS
REPLACE B->THOE WITH T->THORACIC
REPLACE B->DIZZY WITH T->DIZZINESS
REPLACE B->MIDBACK WITH T->MIDBACK
REPLACE B->DRUGD WITH T->DRUG_DEP

IF T->CREDIBLE
   REPLACE B->CREDIB WITH 1
ELSE
   REPLACE B->CREDIB WITH 0
ENDIF
```

```
REPLACE B->SURGERY WITH T->SURGERY
REPLACE B->HOSP WITH T->HOSPITAL

IF T->RX
  REPLACE B->RX WITH 1
ELSE
  REPLACE B->RX WITH 0
ENDIF

IF (T->PHYSIO="")
  REPLACE B->PHYSIO WITH 0
ELSE
  REPLACE B->PHYSIO WITH 1
ENDIF


IF (T->MASSAGE="")
  REPLACE B->MASSAGE WITH 0
ELSE
  REPLACE B->MASSAGE WITH 1
ENDIF

IF T->CHIRO
  REPLACE B->CHIRO WITH 1
ELSE
  REPLACE B->CHIRO WITH 0
ENDIF

IF T->PSYCHOL
  REPLACE B->PSYCHO WITH 1
ELSE
  REPLACE B->PSYCHO WITH 0
ENDIF

IF T->PAINCLINIC
  REPLACE B->PAINCL WITH 1
ELSE
  REPLACE B->PAINCL WITH 0
ENDIF

IF T->ACUPUNCTUR
  REPLACE B->ACUPUN WITH 1
ELSE
  REPLACE B->ACUPUN WITH 0
ENDIF
```

```
IF T->COLLAR >0
  REPLACE B->COLLAR WITH 1
ELSE
  REPLACE B->COLLAR WITH 0
ENDIF

IF AT("A",T->FIN)>0
  REPLACE B->INFAMB WITH 1
ELSE
  REPLACE B->INFAMB WITH 0
  IF AT ("B1",T->FIN)>0
    REPLACE B->PENSION WITH 1
  ELSE
    REPLACE B->PENSION WITH 0
    IF AT ("B2",T->FIN)>0
      REPLACE B->FINOTH WITH 1
    ELSE
      REPLACE B->FINOTH WITH 0
      IF AT ("C",T->FIN)>0
        REPLACE B->LOSTJOB WITH 1
      ELSE
        REPLACE B->LOSTJOB WITH 0
        IF AT ("C2",T->FIN)>0
          REPLACE B->RETIRE WITH 1
        ELSE
          REPLACE B->RETIRE WITH 0
          IF AT ("C3",T->FIN)>0
            REPLACE B->STUDENT WITH 1
          ELSE
            REPLACE B->STUDENT WITH 0
            IF AT ("Q",T->FIN)>0
              REPLACE B->WAGES WITH 1
            ELSE
              REPLACE B->WAGES WITH 0
            ENDIF
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF


REPLACE B->GRMONTH WITH 0
REPLACE B->GRONG WITH 0
REPLACE B->GRPERM WITH 0

REPLACE B->CD1 WITH 0
REPLACE B->CD2 WITH 0
REPLACE B->CD3 WITH 0
REPLACE B->CD4 WITH 0
REPLACE B->CD5 WITH 0
REPLACE B->CD6 WITH 0




IF T->RN >=0
  REPLACE B->GRMONTH WITH T->RN
ELSE
  IF T->RN = -1.00
    REPLACE B->GRONG WITH 1
  ELSE
     IF T->RN = -2.00
       REPLACE B->GRPERM WITH 1
     ENDIF
  ENDIF
ENDIF

REPLACE B->JBTEMP WITH 0
REPLACE B->JBPERM WITH 0
IF T->JOB_PERF = 1
   REPLACE B->JBTEMP WITH 1
ELSE
   IF T->JOB_PERF = 2
       REPLACE B->JBPERM WITH 1
   ENDIF
ENDIF

IF T->CAR_DAMAGE = 1
   REPLACE B->CD1 WITH 1
ELSE
   IF T->CAR_DAMAGE = 2
      REPLACE B->CD2 WITH 1
   ELSE
      IF T->CAR_DAMAGE = 3
         REPLACE B->CD3 WITH 1
      ELSE
         IF T->CAR_DAMAGE = 4
```

```
           REPLACE B->CD4 WITH 1
         ELSE
           IF T->CAR_DAMAGE = 5
             REPLACE B->CD5 WITH 1
           ELSE
             IF T->CAR_DAMAGE = 6
               REPLACE B->CD6 WITH 1.
             ENDIF
           ENDIF
         ENDIF
       ENDIF
     ENDIF
   ENDIF
 ENDIF
 SELECT 1
 SKIP
ENDDO
```

APPENDIX B

LIST OF FACTS USED AS INPUTS

NAME Name of the case, usually only includes name of the plaintiff. Used to annote the case, rather than as an input. This fact was not used as an input but was retained in the database so that I could keep track of an individual case i.e. an annotation

JUDGE Code used to protect the identity of the judge.Also used for annotation.

DATE The date of the trial.

INJDATE The date of the injury.

FTWOLO Number of months lost from full-time work.

PTWOLO Number of months lost from part-time work.

DISAB Time of any total disability

RECOV The Recovery Period (if expressed in numerical form.)

GRMONTH General recovery period. This is largely the same as RECOV above. The original database uses two fields.

GRONG If the recovery is expressed in the case as being "ongoing." A logical value (i.e. yes or no.)

GRPERM If the general recovery is described as "permanent." Logical value.

ATHLETIC Athletic impairment? Logical value.

RECREAT Recreational impairment? Logical value.

HOBBY Significant/Favorite hobby impaired? Logical value.

SOCIAL Social life impaired? Logical value.

SEXUAL Sexual impairment?Logical value.

LIFEST Lifestyle impairment? Logical value.

HOUSE	Housework impairment?Logical value.
PSYCHO	Psychological impairment? Logical value.
DEP	Depression? Logical value.
PER	Personality affected, changed? Logical value.
JBTEMP	Job performance temporarily impaired? Logical value.
JBPERM	Job performance permananently impaired? Logical value.
JBCHAN	Job change necessitated? Logical value.
NECK	Persistent neckpain or stiffness? Logical value.
CPS	Chronic Pain Syndrome diagnosed? Logical value.
LBP	Lower Back Pain? Logical value.
HEADAC	Persistent headaches? Logical value.
MYOFAS	Myofascial pain syndrome diagnosed? Logical value.
TMJ	Jaw problems diagnosed? Logical value.
SPASMS	Muscle spasms diagnosed? Logical value.
MIDBACK	Midback pain diagnosed? Logical value.
DRUGD	Drug dependency? Logical value.
CREDIB	Credibility problem? Logical value.
SURGERY	Surgery required? Logical value.
HOSP	Hopitalization required? Logical value.
RX	Prescription medicine required? Logical value.
PHYSIO	Physiotherapist treatment required? Logical value.
MASSAGE	Massage treatment required? Logical value.
CHIRO	Chiropractor appointments? Logical value.

PSYCHO Psychologist/Psychiatrist appointments required? Logical value.

PAINCLINIC Pain Clinic attendance required? Logical value.

ACUPUNCTUR Accupuncture appointements/treatment required.

COLLAR Collar required? Logical value.

INFAMB Future income loss - Infant/ambitions? Logical value.

PENSION Future income loss - Pension? Logical value.

FINOTH Future income loss - other loss? Logical value.

LOSTJOB Future income loss - loss of job? Logical value.

RETIRE Future income loss - early retirement? Logical value.

STUDENT Future income loss - Lost opportunity to be a student? Logical value.

WAGES Lost wages? Logical value.

CD1-6 Car damage (1 = $0; 2 = <$500; 3 = $500-1,000; 4 = $1,000-2,500; 5 = $2,500-$5,000; 6 = >$5,000.)

APPENDIX C

(Segment of TEST3.PRG changing 1 output to 10. All other code remains unchanged.)

```
IF T->QU <5000
  REPLACE B->QU100 WITH 0
  REPLACE B->QU80 WITH 0
  REPLACE B->QU60 WITH 0
  REPLACE B->QU40 WITH 0
  REPLACE B->QU30 WITH 0
  REPLACE B->QU20 WITH 0
  REPLACE B->QU15 WITH 0
  REPLACE B->QU10 WITH 0
  REPLACE B->QU5 WITH 1
ELSE
  IF T->QU <10000
    REPLACE B->QU100 WITH 0
    REPLACE B->QU80 WITH 0
    REPLACE B->QU60 WITH 0
    REPLACE B->QU40 WITH 0
    REPLACE B->QU30 WITH 0
    REPLACE B->QU20 WITH 0
    REPLACE B->QU15 WITH 0
    REPLACE B->QU10 WITH 1
    REPLACE B->QU5 WITH 0
  ELSE
    IF T->QU <15000
      REPLACE B->QU100 WITH 0
      REPLACE B->QU80 WITH 0
      REPLACE B->QU60 WITH 0
      REPLACE B->QU40 WITH 0
      REPLACE B->QU30 WITH 0
      REPLACE B->QU20 WITH 0
      REPLACE B->QU15 WITH 1
      REPLACE B->QU10 WITH 0
      REPLACE B->QU5 WITH 0
    ELSE
      IF T->QU <20000
        REPLACE B->QU100 WITH 0
        REPLACE B->QU80 WITH 0
        REPLACE B->QU60 WITH 0
        REPLACE B->QU40 WITH 0
  REPLACE B->QU30 WITH 0
  REPLACE B->QU20 WITH 1
  REPLACE B->QU15 WITH 0
  REPLACE B->QU10 WITH 0
  REPLACE B->QU5 WITH 0
ELSE
  IF T->QU <30,000
    REPLACE B->QU100 WITH 0
    REPLACE B->QU80 WITH 0
    REPLACE B->QU60 WITH 0
    REPLACE B->QU40 WITH 0
    REPLACE B->QU30 WITH 1
    REPLACE B->QU20 WITH 0
    REPLACE B->QU15 WITH 0
    REPLACE B->QU10 WITH 0
    REPLACE B->QU5 WITH 0
  ELSE
    IF T->QU <40,000
      REPLACE B->QU100 WITH 0
      REPLACE B->QU80 WITH 0
      REPLACE B->QU60 WITH 0
      REPLACE B->QU40 WITH 1
      REPLACE B->QU30 WITH 0
      REPLACE B->QU20 WITH 0
      REPLACE B->QU15 WITH 0
      REPLACE B->QU10 WITH 0
      REPLACE B->QU5 WITH 0
    ELSE
      IF T->QU <60,000
        REPLACE QU100 WITH 0
        REPLACE QU80 WITH 0
        REPLACE QU60 WITH 1
        REPLACE QU40 WITH 0
        REPLACE QU30 WITH 0
        REPLACE QU20 WITH 0
        REPLACE QU15 WITH 0
        REPLACE QU10 WITH 0
        REPLACE QU5 WITH 0
      ELSE
        IF T->QU <80,000
          REPLACE QU100 WITH 0
          REPLACE QU80 WITH 1
          REPLACE QU60 WITH 0
          REPLACE QU40 WITH 0
          REPLACE QU30 WITH 0
```

```
                REPLACE QU20 WITH 0
                REPLACE QU15 WITH 0
                REPLACE QU10 WITH 0
                REPLACE QU5 WITH 0
              ELSE
                IF T->QU <100,000
                  REPLACE B->QU100 WITH 1
                  REPLACE B->QU80 WITH 0
                  REPLACE B->QU60 WITH 0
                  REPLACE B->QU40 WITH 0
                  REPLACE B->QU30 WITH 0
                  REPLACE B->QU20 WITH 0
                  REPLACE B->QU15 WITH 0
                  REPLACE B->QU10 WITH 0
                  REPLACE B->QU5 WITH 0
                ELSE
                  IF T->QU<150,000
                    REPLACE B->QU150 WITH 1
                    REPLACE B->QU100 WITH 0
                    REPLACE B->QU80 WITH 0
                    REPLACE B->QU60 WITH 0
                    REPLACE B->QU40 WITH 0
                    REPLACE B->QU30 WITH 0
                    REPLACE B->QU20 WITH 0
                    REPLACE B->QU15 WITH 0
                    REPLACE B->QU10 WITH 0
                    REPLACE B->QU5 WITH 0
                  ENDIF
                ENDIF
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF
```

APPENDIX D

(Segment of Test4.Prg changing 1 output to 6 outputs. All other outputs remained the same.)

```
IF T->QU <120,001
        REPLACE B->QU120 WITH 1
        REPLACE B->QU60 WITH 0
        REPLACE B->QU40 WITH 0
        REPLACE B->QU25 WITH 0
        REPLACE B->QU15 WITH 0
        REPLACE B->QU7 WITH 0
ELSE
        IF T->QU <60,001
                REPLACE B->QU120 WITH 0
                REPLACE B->QU60 WITH 1
                REPLACE B->QU40 WITH 0
                REPLACE B->QU25 WITH 0
                REPLACE B->QU15 WITH 0
                REPLACE B->QU7 WITH 0
        ELSE
                IF T->QU <40,001
                        REPLACE B->QU120 WITH 0
                        REPLACE B->QU60 WITH 0
                        REPLACE B->QU40 WITH 1
                        REPLACE B->QU25 WITH 0
                        REPLACE B->QU15 WITH 0
                        REPLACE B->QU7 WITH 0
                ELSE
                        IF T->QU <25,001
                                REPLACE B->QU120 WITH 0
                                REPLACE B->QU60 WITH 0
                                REPLACE B->QU40 WITH 0
                                REPLACE B->QU25 WITH 1
                                REPLACE B->QU15 WITH 0
                                REPLACE B->QU7 WITH 0
                        ELSE
                                IF T->QU <15,001
                                        REPLACE B->QU120 WITH 0
                                        REPLACE B->QU60 WITH 0
                                        REPLACE B->QU40 WITH 0
                                        REPLACE B->QU25 WITH 0
                                        REPLACE B->QU15 WITH 1
```

```
                    REPLACE B->QU7 WITH 0
                ELSE
                    IF T->QU <7,501
                        REPLACE B->QU120 WITH 0
                        REPLACE B->QU60 WITH 0
                        REPLACE B->QU40 WITH 0
                        REPLACE B->QU25 WITH 0
                        REPLACE B->QU15 WITH 0
                        REPLACE B->QU7 WITH 1
                    ENDIF
                ENDIF
            ENDIF
        ENDIF
    ENDIF
ENDIF
```

APPENDIX E

dBASE Coding for the "fuzzification of Quantum. (All other code remains unchanged.)

```
BAND = 0
   DO WHILE BAND < 5
     NODEVALUE = 0
     BAND = BAND + 1

     DO CASE
         CASE BAND = 1           &&  all 5 band widths are
           MIN = 0             &&  an identical 108 units wide
           MID = 54
           MAX = 108
         CASE BAND = 2
           MIN = 54
           MID = 108
           MAX = 164
         CASE BAND = 3
           MIN = 108
           MID = 164
           MAX = 216
         CASE BAND = 4
           MIN = 164
           MID = 216
           MAX = 270
         CASE BAND = 5
           MIN = 216
           MID = 270
           MAX = 324
     ENDCASE

     IF B->QU > MIN .AND. B->QU <= MID
       NODEVALUE = (B->QU - MIN)/(MID - MIN)
     ELSE
       IF B->QU > MID .AND. B->QU <= MAX
         NODEVALUE = 1-(B->QU - MID)/(MAX - MID)
       ENDIF
     ENDIF

     DO CASE
         CASE BAND = 1
           REPLACE B->BAND1 WITH NODEVALUE
```

```
      CASE BAND = 2
        REPLACE B->BAND2 WITH NODEVALUE
      CASE BAND = 3
        REPLACE B->BAND3 WITH NODEVALUE
      CASE BAND = 4
        REPLACE B->BAND4 WITH NODEVALUE
      CASE BAND = 5
        REPLACE B->BAND5 WITH NODEVALUE
   ENDCASE
  ENDDO       &&  WHILE band < 5
```

APPENDIX F

GLOSSARY OF NEURAL NETWORK TERMS

Artificial Neural Network

A model made to simulate a biological neural systems; it may model brain processes or brain capabilities.

Back-Propogation

A supervised learning method in which the error signal is fed back through the network altering weights in order to prevent the same error from recurring.

Back-Propogation Rule

Also called the Generalized Delta Rule, training consists of running patterns forward through the network layers, then propogating the errors backwards and updating the weights in order to reduce the errors.

Biological Neural Networks

The combination of neurons that go to make up the brain of any living species.

Connectionists

Those researchers who accept the new electronic approach to computation using highly interconnected neuron-like components that take certain aspects of neurology and apply these to create "intelligent" computer programs, in the sense that they have the ability to improve their performance.

Connection

The path between two neurons across which information passes. These are the equivalent of synapses in biological neural networks. The weights on the connections determines the strength of the signals.

Epoch

One complete presentation of the training set to the network during training.

Error

The difference between the network response and the required network response.

Fault Tolerance

The degree to which the network can deal with noisy data.

Feedback Network

A network in which the inputs to any given neuron can be taken from outputs either of its own layer or the following layer(s).

Hidden Layer

The layer of neurons that are hidden from the user - neither its inputs or outputs are fed to the outside world. There may be one or more layers of neurons in a given network. If there is only one hidden layer of neurons.

Layer

A group of neurons that have a specific function i.e. inputs, outputs, or hidden layers.

Learning Rule

The algorithm used for modifying the connection strengths/ weights in response to training patterns while training is being carried out.

Multi-Layer Perceptron

The most common of neural network architectures in which perceptrons are connected in layers. The most common of these architectures is the three layer network.

Neuron

The smallest processing element in the network. The neuron receives inputs from outputs of other preceding neurons. Using the inputs the neuron produces an output using simple vector matrix multiplication, (see Transfer Function for more on this point.) This output is either fed to other neurons or to directly to the outside world.

Output Neuron

The neuron in the network that gives the output, or the results.

Over training

Training a neural network involves the presentation of problems facts into the network. Over training is the result of training the network to respond very accurately to the training set only which means that it is less able to generalize.

Perceptron

A one layer network in which inputs are summed to provide an output. See Section 1.17.3 of Chapter 1.

Pre-Processing

The process prior to processing by the neural network in which the raw data is manipulated into a network-readable form. For an example of a pre-processing programming see the Test2.prg program in Appendix A.

Post-Processing

Modification of the output of the network before applying them to the real world.

Supervised Learning

The process by which examples of presented to the network together with examples of the expected outputs. If the outputs from the network differs from the output specified the network weights are modified. Section 1.11.3 of Chapter 1 for a description of both supervised and unsupervised learning.

Test Set

Those examples that are set aside, that have not been used in the training process. Once training has been completed, the network can be tested using the Test set to see how accurate it is.

Topology

Another name for the architecture of the network or the way in which the neurons are arranged.

Training

The process during which the neuron weights are adjusted so that the network performs the function for which it was designed.

Transfer Function

The function that translates the internal value calculated by the neuron to a value suitable to represent its output. There are a variety of possible functions. See Section 1-10 in Chapter 1.

Unsupervised Learning

The training set has no output response associated with the input stimuli. The network makes associations between the input data.

Weight

The value associated with a connection between neurons in the network. The value of the weights determines how much of the output of one neuron is fed into the input of the next.

APPENDIX G

GLOSSARY OF BRAINMAKER TERMS

Bad Facts

The total number of facts that the training network has got "wrong" in that data run.

Data Run

The process by which all facts/cases in the training data set are presented to the neural network program once. The training of a neural network will usually involve many, possibly hundreds or thousands of data runs. Also known as "epochs."

Good Facts

The total number of facts that the training network has got "right" in that data run.

Learning Rate

This determines how large or small an adjustment is made by the network during training.

Output

This is what the network produces during training by using the back-propogation rule.

Pattern Output

This is the set of number or symbols that the network should be producing, or very close to it.

Running

This function puts input data through a trained neural network and provides the user with the neural network output usually on screen. When users want to put the neural network to work they "run " the network.

Testing

This function puts input-output pairs of data that were not used during training through the neural network program and allows the user to see how well (or how badly) the network has trained.

Tolerance

The percentage of the range of output. No corrections are made to the network if the output is within the tolerance

Total

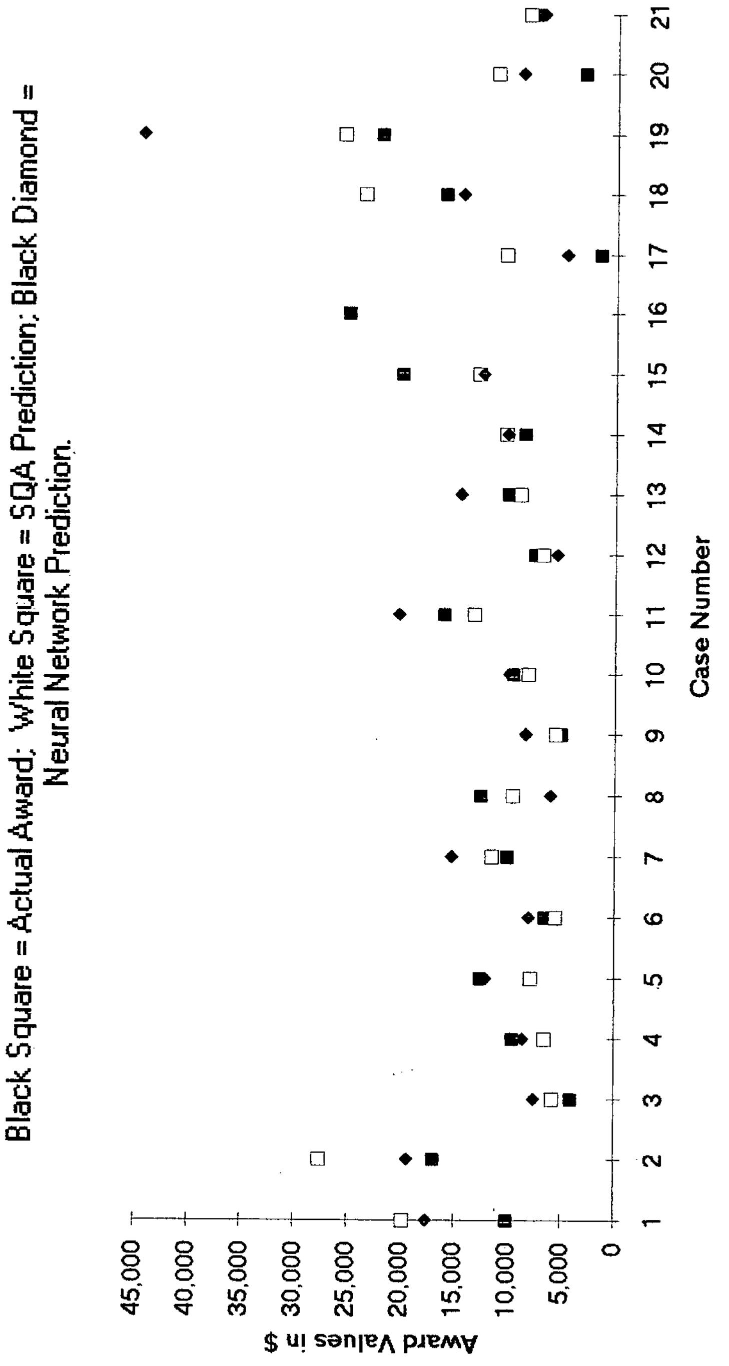
The total cumulative number of facts evaluated by the training network.

Training

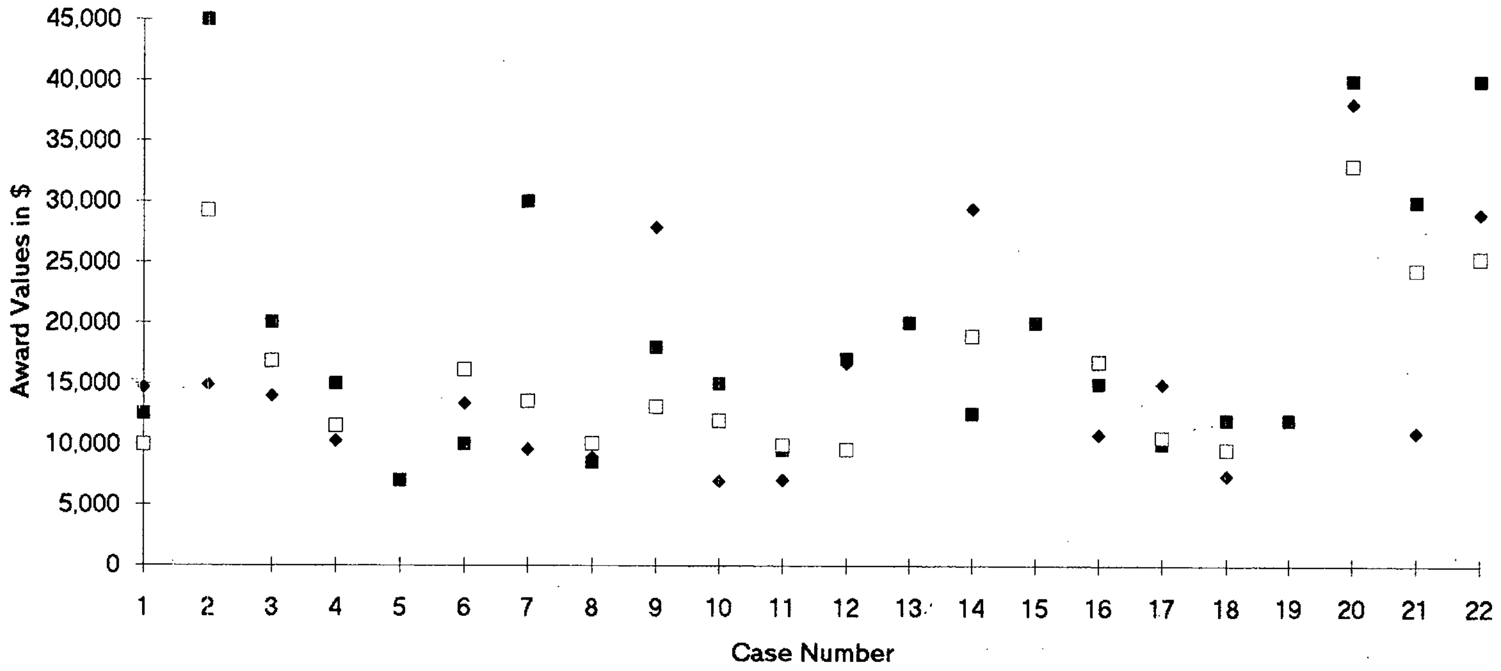
This function puts the input-output pairs of data through the neural network program repeatedly and corrects the network as it learns.

APPENDIX H

Graphical Representation of Figure 4.13



Black Square = Actual Award; White Square = SQA Prediction; Black Diamond = Neural Network Prediction


Award Values in $
Case Number