Neural Network Satellite Retrievals of Nocturnal Stratocumulus Cloud Properties

by

Marc Rautenhaus

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

The Faculty of Graduate Studies

(Atmospheric Science)

The University Of British Columbia

October 2007

Abstract

I investigate the feasibility of retrieving cloud top droplet effective radius, optical thickness and cloud top temperature of nocturnal marine stratocumulus clouds by inverting infrared satellite measurements using an artificial neural network. For my study, I use the information contained in the three infrared channels centred at 3.7, 11.0 and 12.0 μm of the Moderate Resolution Imaging Spectroradiometer (MODIS) on-board NASA's Terra satellite, as well as sea surface temperature. A database of simulated top-of-atmosphere brightness temperatures of a range of cloud parameters is computed using a correlated-k parameterisation which I have embedded in the radiative transfer package libRadtran. The database is used to train feed-forward neural networks of different architecture to perform the inversion of the satellite measurements for the cloud properties. I investigate the application of Bayesian methods to estimate the retrieval uncertainties, and analyse the Jacobian of the networks in order to gain information about the functional dependence of the retrieved parameters on the inputs. A high variability in the Jacobian indicates that the nocturnal retrieval problem is ill-posed. My experiments show that because the problem is ill-conditioned, it is very difficult to find a network that approximates the database of simulated brightness temperatures well. Sea surface temperature proves to be a necessary input. I compare the retrievals of a selected network architecture with in-situ cloud measurements taken during the second Dynamics and Chemistry of Marine Stratocumulus experiment. The results show general agreement between retrievals and in-situ observations, although no collocated comparsions are possible because of a time lag of five hours between both measurements. I establish that the uncertainty estimates are prone to numerical problems and their results are questionable. I show that the Jacobian is a valuable tool in evaluating the retrieval networks.

Table of Contents

Abstract ii

Table of Contents iii

List of Tables vi

List of Figures vii

List of Algorithms xi

List of Abbreviations xii

Acknowledgements xv

1 Introduction 1

1.1 Clouds and Climate 2

1.2 Marine Stratocumulus Clouds and the Aerosol Indirect Effect 5

1.2.1 Bulk Properties 5

1.2.2 Aerosols 9

1.3 Radiative Transfer Terminology 12

1.4 Motivation 15

1.5 Overview and Theory of Existing Retrieval Algorithms 18

1.5.1 Daytime Methods 18

1.5.2 Nighttime Methods 21

1.5.3 Artificial Neural Networks 25

1.6 Objectives 29

2 Radiative Transfer and Development of the Forward Model . . . 31

2.1 DYCOMS-II Data . . . 32

2.2 Scattering and Droplet Spectra . . . 32

2.2.1 Radiative Transfer Equation . . . 32

2.2.2 Droplet Size Distribution and its Phase Function . . . 35

2.3 Adiabatic Cloud Model . . . 40

2.4 Choice of Channels and Correlated-k . . . 46

2.5 The Overlying Atmosphere . . . 51

2.6 Forward Model . . . 54

2.6.1 Radiative Transfer Model: libRadtran . . . 54

2.6.2 Design of the Forward Model . . . 55

3 Nonlinear Regression with Neural Networks and Uncertainty Estimation . . . 60

3.1 The Nonlinear Regression Problem . . . 61

3.1.1 Error Functions and Overfitting . . . 61

3.1.2 Controlling Model Complexity and Network Training . . . 64

3.1.3 Describing Uncertainty . . . 65

3.2 Distribution of the Neural Network Weights . . . 68

3.2.1 Derivation . . . 68

3.2.2 The Meaning of the Hyperparameters . . . 71

3.2.3 Gaussian Approximation to the Posterior Distribution . . . 72

3.3 Output Uncertainties . . . 73

3.4 Hyperparameter Re-Estimation . . . 76

3.4.1 Evidence Procedure for $\boldsymbol{A}_r$. . . . 79

3.5 The Jacobian Matrix . . . 82

3.6 Implementation Issues . . . 84

3.6.1 Normalisation of Input and Output Variables . . . 87

3.6.2 Regularisation of the Hessian . . . 87

3.7 Usefulness of the Aires et al. Method . . . 88

4 Retrieval, Results and Evaluation . . . 94

4.1 Retrieval Setup . . . 94

4.1.1 The Test Scene 94
4.1.2 Lookup Table Setup 95
4.2 Cloud Mask 97
4.2.1 MODIS Cloud Mask and Irretrievable Pixels 98
4.2.2 Possible Failure Mechanisms 100
4.3 Network Training 105
4.3.1 Network Architecture 105
4.3.2 Failure of the Aires et al. Hyperparameter Re-Estimation 106
4.3.3 Input Preprocessing 107
4.4 Network Architecture: Inputs and Hidden Neurons 107
4.4.1 Brightness Temperature Differences 107
4.4.2 Three Input Networks 110
4.4.3 Four Input Networks 111
4.5 Retrieval Evaluation, Jacobian and Uncertainty 120
4.5.1 Comparison with In-Situ Data 120
4.5.2 Jacobian 121
4.5.3 Uncertainty 129
4.6 Further Developments 130

5 Summary 135

Bibliography 139

Appendices

A Implementation of the Aires et al. Method in Matlab 147
A.1 Modified Netlab Functions 147
A.2 Main Loop: mlatrain 147
A.3 Gradient Computation with Backpropagation: mlagrad 149
A.4 The Hessian with the Pearlmutter $\mathcal{R}\{\cdot\}$-Algorithm: mlahess, mlahdotv 151
A.5 The Jacobian and its Distribution: mlajacob, mlajacobuncertainty 152
A.6 Normalisation of Input and Output Variables 153
A.7 Implementation Difficulties: Regularisation of the Hessian and Numerical Symmetry 154

List of Tables

2.1 Spectral intervals of the four MODIS channels implemented in the forward model and the atmospheric constituents whose absorptivity is accounted for. 50

3.1 Jacobian matrices and their uncertainty (one standard deviation) of the two networks trained to model the example function given by (3.9) and (3.10). 84

4.1 Point estimate of the Jacobian of a network containing 30 neurons in the hidden layer and using BT(3.7), BT(11), BT(12) and T_{sfc} as inputs. 109

4.2 The same as Table 4.1, but for a network containing 15 neurons in the hidden layer and using BT(3.7), BT(11), BTD(11-12) and T_{sfc} as inputs. 115

4.3 The same as Table 4.1, but for a network containing 30 neurons in the hidden layer and using BT(3.7), BT(11), BTD(11-12) and T_{sfc} as inputs. 115

4.4 Average Jacobian of the July 11, 2001, scene and its variability (15N network). 124

4.5 The same as in Table 4.4, but normalised by the standard deviation of the inputs. 125

A.1 List of NETLAB functions that were adapted or created in order to accommodate the matrix hyperparameters $\boldsymbol{A}_{in}$ and $\boldsymbol{A}_r$. 148

List of Figures

1.1 The role of clouds in the climate system. 2

1.2 Cloud feedback processes represent a large uncertainty in climate modelling. 4

1.3 The annual cycle of cloud amount, lower tropospheric stability and shortwave, longwave and net cloud forcings from a two-year Earth Radiation Budget Experiment climatology. . 5

1.4 Example of in-situ measurements of microphysical cloud properties taken during the First International Satellite Cloud Climatology Programme Regional Experiment in precipitating marine stratocumulus clouds off the coast of California. 7

1.5 Typical ranges of mean and effective particle diameter, cloud liquid water content and particle number concentration in low-level stratiform clouds. 8

1.6 Dependence of cloud radiative forcing on macroscopic and microscopic cloud properties. . 11

1.7 Different methods to observe clouds. 16

1.8 Dependence of radiative effects on microphysical properties of a cloud. 20

1.9 Physical basis for the remote sensing of nocturnal marine stratocumulus clouds. 23

1.10 Schematic diagram of a feed-forward network with two layers of adaptive weights. 26

2.1 Processes that influence radiation as it passes through the atmosphere. 33

2.2 In-situ measured spectra of cloud droplet sizes from the Jul 11th and Jul 13th flights, together with best-fit gamma distributions. 38

2.3 Phase functions of cloud distributions following a gamma distribution with a shape parameter of 26. 39

2.4 Vertical profile of cloud properties from the idealised adiabatic cloud model, together with in-situ measurements from the July 11 DYCOMS-II flight. 41

2.5 Measurements of k value versus height in marine stratocumulus clouds. 44

2.6 The same as Figure 2.4, but with a subadiabatic liquid water content of 75% of the adiabatic value. 45

2.7 Dependence of atmospheric transmittance on wavelength for a typical midlatitude summer atmosphere. 47

2.8 Illustration of the fundamental idea of correlated-k. 49

2.9 Atmospheric soundings from San Diego from July 11, 2001, and July 13, 2001. 52

2.10 Vertical brightness temperature profiles through a thin cloud and a thick cloud. 57

2.11 Dependence of brightness temperature (BT) and BT difference on varying effective radius and optical thickness for a cloud top temperature of 285 K and a cloud top pressure of 900 hPa. 58

2.12 The same in Figure 2.11, but for different cloud top pressures. 59

3.1 Illustration of overfitting I. 62

3.2 Illustration of overfitting II. 63

3.3 Distribution of the first weight of the neural network used for the example given by equations (3.9) and (3.10) after a short training run and after a longer training run. 73

3.4 Section through the functional surface of the example given by (3.9) and (3.10) with network prediction and error bars. 77

3.5 Scatterplots of the estimated error (one standard deviation) vs. the actual error for both output variables of the example given in (3.9) and (3.10) for the long training run. 78

3.6 The same as Figure 3.5, but for the short training run. 78

3.7 Development of the covariance matrices $\boldsymbol{C_0}$ and $\boldsymbol{C_{in}}$ and the hyperparameters $\boldsymbol{A_{in}}$ and $\boldsymbol{A_r}$ during the training run shown in Figure 3.8. 83

3.8 Development of the weight values and the mean-square-error of the training dataset and of a test dataset without added noise of the example given by (3.9) and (3.10) during a training run with 10 hyperparameter re-estimation iterations. 85

3.9 Scatterplot of the network predictions for both outputs of the example given in (3.9) and (3.10) vs. the target data. 86

3.10 Demonstration of the input dependence of the error estimation given in (3.43). 89

3.11 Same as Figure 3.10, but with a training dataset consisting of 10,000 datapoints. 90

3.12 The same example as in Figure 3.4, but the network has only been trained with data in an interval. 90

3.13 Limitations of the error estimation – a simple two-layer perceptron is not able to model a mapping that contains ambiguities. 91

3.14 The same as Figures 3.5 and 3.6, but using non-regularised Hessian. 91

4.1 Brightness temperature difference (BTD) between channel 20 (3.7 μm) and channel 31 (11 μm) and channel 31 brightness temperature at 6:25 UTC on July 11th, 2001. 96

4.2 k-value vs. height as inferred from the DYCOMS-II in-situ measurements of July 11 and July 13, 2001. 97

4.3 Histograms of ν_{gam} and k-value as inferred from the DYCOMS-II in-situ measurements of July 11, 2001. 97

4.4 MODIS cloud mask product for the July 11, 2001, scene and cloud mask derived from the BT/BTD range in the lookup table. 99

4.5 LUT and MODIS cloud mask filtered scene brightness temperatures. 101

4.6 The same as Figure 4.5, but for the brightness temperature difference between channels 29 and 31 (8.5 and 11 μm). 102

4.7 11 μm versus 12 μm radiance plots for the observations along the line shown in Figure 4.4. 103

4.8 BTD(3.7-11) and BTD(11-12) signals of the pixels between points A and B as shown in Figure 4.4. 104

4.9 Subset of the July 11 LUT. In addition to the cloud top pressure (939.5 hPa), the cloud top temperature is also held constant at 285 K. 108

4.10 Training performance of a three input network employing the BT(3.7), BT(11) and BTD(11-12) signals, compared to a four input network additionally making use of surface temperature. 112

4.11 Retrievals of cloud top temperature and visible cloud optical thickness of the three input network from which the scatter plots on the left side of Figure 4.10 were produced. 113

4.12 Mean-square-error of the validation dataset for the four input ANN (BT(3.7), BT(11), BTD(11-12), T_{sfc}) in dependence of the number of hidden neurons. 114

4.13 Sensitivity of 15N predictions if cloud top pressure (fixed in the LUT) is varied by ± 5 hPa. 117

4.14 The same as Figure 4.13, but for the 30N network. 118

4.15 Histograms of the 15N Jacobian of the LUT subset plotted in Figure 4.9. 119

4.16 Retrieved effective radii for the test scene and comparison of retrieved (15N network) and aircraft measured values. 122

4.17 The same as Figure 4.16, but for cloud top temperature. 123

4.18 The same as Figure 4.16, but for cloud optical thickness. 124

4.19 Histograms of the 15N Jacobian of the July 11, 2001, scene. 126

4.20 Spatial distribution of the sensitivity of the effective radius retrieval to changes in BT(3.7) and BTD(11-12) on July 11, 2001 (15N network). 127

4.21 The same as Figure 4.20, but for the sensitivity of cloud top temperature to changes in BT(3.7) and BT(11). 128

4.22 Spatial distribution of the uncertainty in the effective radius and optical thickness retrievals, as computed from the neural uncertainty term $\boldsymbol{G}^T\boldsymbol{H}^{-1}\boldsymbol{G}$. 131

List of Algorithms

3.1 Bayesian Neural Network Training with Matrix Hyperparameter Re-Estimation. 85

List of Abbreviations

15N	network with four inputs and 15 hidden neurons used in Chapter 4
30N	network with four inputs and 30 hidden neurons used in Chapter 4
ABL	atmospheric boundary layer
ACE-2	Second Aerosol Characterisation Experiment
AIE	aerosol indirect effect
AMSR-E	Advanced Microwave Scanning Radiometer for the Earth Observing System
ANN	artificial neural network
AR4	IPCC 4th Assessment Report
AS	adiabatic stratified cloud model
AVHRR	Advanced Very High Resolution Radiometer
BT	brightness temperature
BT(11)	11 μm brightness temperature
BT(12)	12 μm brightness temperature
BT(3.7)	3.7 μm brightness temperature
BT(8.5)	8.5 μm brightness temperature
BTD	brightness temperature difference
BTD(11-12)	BTD between the 11 μm and 12 μm channels
BTD(3.7-11)	BTD between the 3.7 μm and 11 μm channels

BTD(8.5-11)	BTD between the 8.5 μm and 11 μm channels
CCN	cloud condensation nuclei
CERES	Clouds and the Earth's Radiant Energy System
CFMIP	Cloud Feedback Model Intercomparison Project
CRF	cloud radiative forcing
DISORT	Discrete Ordinate Radiative Transfer
DYCOMS-II	Second Dynamics and Chemistry of Marine Stratocumulus field experiment
EPIC	East Pacific Investigation of Climate
ERBE	Earth Radiation Budget Experiment
FIRE	First ISCCP Regional Experiment
GCM	general circulation model
IPA	independent pixel approximation
IPCC	Intergovernmental Panel on Climate Change
ISCCP	International Satellite Cloud Climatology Programme
LUT	lookup table
LWC	liquid water content
LWP	liquid water path
MLP	multilayer perceptron
MODIS	Moderate Resolution Imaging Spectroradiometer
MSE	mean square error
NASA	National Aeronautics and Space Administration
OA	overlying atmosphere
PCA	principal component analysis

PDF	probability density function
PDT	Pacific Daylight Time
RF02	DYCOMS-II research flight II
RF03	DYCOMS-II research flight III
RTE	radiative transfer equation
Sc	stratocumulus
SHDOM	Spherical Harmonic Discrete Ordinate Method
SST	sea surface temperature
TOA	top of atmosphere
TRMM	Tropical Rainfall Measuring Mission
UTC	Coordinated Universal Time
VU	vertically uniform cloud model

Acknowledgements

I gratefully acknowledge the support of all of the people who made this thesis possible. First, I would like to thank Phil for giving me the opportunity to spend two years at the University of British Columbia, including the participation in the European Geosciences Union conference in Vienna. I am very grateful for his supervision, the numerous discussions, his patience, guidance and for his tremendous time commitment, especially while I was preparing my poster presentation for the conference and while I was writing up my thesis.

I am grateful to my committee members, William and especially Douw for his valuable advice. I thank Christian for supporting discussions when I needed them. I appreciate the help of Uli and Dr. Mayer with the radiative transfer model libRadtran.

I am enormously thankful to my parents for their continuing belief in me during my entire time at university and for accepting all of my decisions. Without their support my stay in Vancouver would not have been possible. I thank my family and my friends for support and for bringing joy into my life. Most of all, however, I thank Riki from whom I received all the love and support an individual person could have given during the past two years.

Chapter 1

Introduction

> Of paramount importance to a comprehensive understanding of the Earth's climate and its response to anthropogenic and natural variability is a knowledge, on a global sense, of cloud properties that may be achieved through remote sensing and retrieval algorithms.
> – King et al. (1997, Page 2)

Shallow boundary layer clouds play a critical role in the exchange of energy and water in our climate system. They do not contribute as much to the greenhouse effect as do upper level clouds, but due to their high albedo[1], they reflect a large portion of the incoming shortwave radiation back to space. The radiative forcing that these clouds exert on the atmosphere is modulated by their macrophysical, microphysical, and optical properties, and the way they interact with the climate system has been the subject of intense research activity in the last decades (Stephens, 2005, and references therein). However, despite this extensive research work, processes that are involved in cloud-atmosphere interaction and their representation in general circulation models remain some of the primary uncertainties in global climate modelling (Bony et al., 2006; Ringer et al., 2006; Williams and Tselioudis, 2007).

Accurate observations of clouds are a key component to solving the problems in this field. In this thesis I address the retrieval of marine stratocumulus cloud properties from satellite measurements, specifically during night time. These marine boundary layer clouds are common and cover large parts of the oceans (Klein and Hartmann, 1993; Norris and Leovy, 1994). Their horizontal homogeneity and usually low liquid water path put them amongst the simplest to treat. Nevertheless, in a recent retrieval comparison exercise, Turner et al. (2007) pointed out that such thin liquid water clouds are surprisingly difficult to observe. This is particularly true for nocturnal observations, where no operational retrieval data are available to date.

This introductory chapter will cover the fundamentals of the clouds and climate research field, present information on marine stratocumulus clouds, and review relevant previous research work in the field of satellite remote sensing. I conclude with an outline of the objectives for this work which will be addressed in the following chapters.

[1] A formal definition of the albedo will be given in section 1.3.

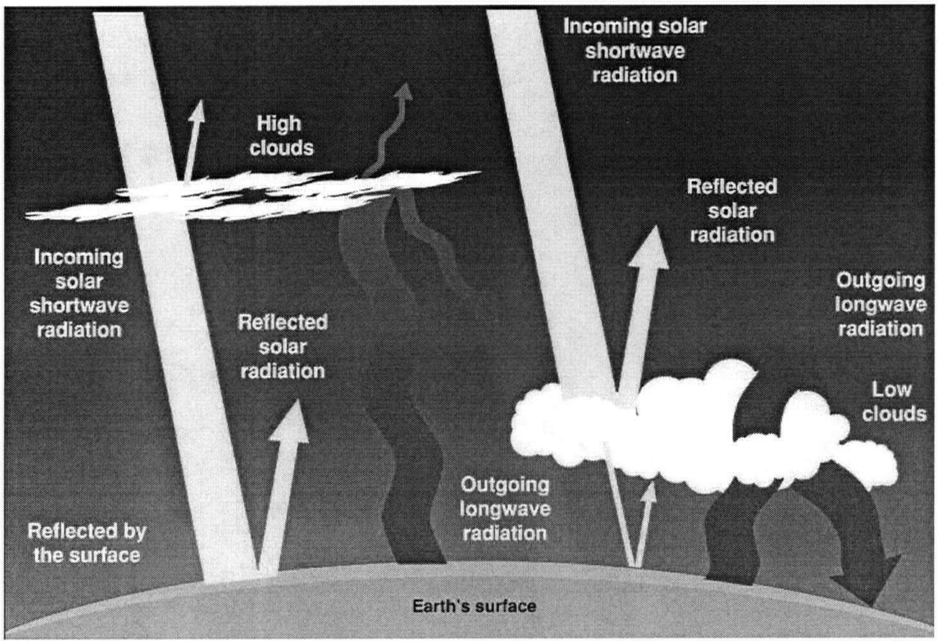

Incoming solar shortwave radiation
High clouds
Incoming solar shortwave radiation
Reflected solar radiation
Incoming solar shortwave radiation
Reflected solar radiation
Outgoing longwave radiation
Low clouds
Outgoing longwave radiation
Reflected by the surface
Earth's surface

Figure 1.1: Clouds play a complex role in the radiative energy balance of our planet. Reflecting incoming shortwave radiation from space and trapping outgoing longwave radiation from the earth's surface and atmosphere, they impact the climate with both cooling and warming effects. The magnitude of these radiative effects is dependent on macroscopic and microscopic cloud properties, giving rise to complex interactions within the climate system (from Phillips and Barry, 2002, courtesy of NASA Marshall Space Flight Center and Science@NASA).

1.1 Clouds and Climate

Cloud radiative forcing (CRF) is defined as the difference between the upwelling radiative flux at the top of the atmosphere and what it would be if the clouds were absent. Figure 1.1 illustrates the most important interactions between clouds and radiation. The largest contribution to the globally averaged long wave CRF is made by high clouds in the upper levels of the troposphere. They are cold and thus emit less radiation to space than the earth's surface and atmosphere under clear skies. Clouds that are optically thick, on the other hand, have the largest albedo, which makes them the largest contributor to the short wave CRF.

In the global annual average, clouds exert a net cooling effect on the climate system in its present state. The average top-of-atmosphere (TOA) short wave CRF is about -50 W/m^2 (the negative sign indicates cooling), while the TOA long wave CRF accounts for a gain of approximately 30 W/m^2 – a net cloud forcing of -20 W/m^2 (Wielicki et al., 1995). These values are large compared to the heating that would result if the current CO_2 concentration was doubled, which is estimated to be on the order of 4 W/m^2 (Wielicki et al., 1995).

Because of their small scale variability in time and space, clouds are difficult to simulate in numerical global climate and weather forecasting models. The processes governing cloud formation and dissipation occur on spatial scales smaller than a general circulation model grid cell, so that the clouds and their interaction with the atmosphere have to be parametrised. Since the phase change from the gaseous to the liquid phase produces a nonlinear transition from transparent vapour to opaque water droplets, parameterisations require detailed knowledge of the processes involved so as to capture the major effects of the governing physics. An additional complication is that knowledge of grid cell mean cloud properties does not uniquely determine knowledge of the CRF because of the nonlinear relationship between cloud properties and albedo (Harshvardhan, 1982).

Especially of interest to the climate science community is the radiative feedback between clouds and the evolving ocean and atmosphere. How do clouds react to a changing climate? Does a warmer atmosphere induce more shallow clouds, which in turn would cool the climate? Given the magnitude of the net CRF, small changes in cloud cover could already have a significant effect on the earth's energy balance. For instance, Hartmann et al. (1992), using one year of global satellite data, found the sensitivity of the radiation balance to low clouds to be on the order of -0.6 W/m^2 per percent fractional cloud cover in the annual average.

Cloud feedback processes depend on so many factors that they have long been identified as being the largest source of uncertainty in climate change predictions; as Bony et al. (2006) note, with a larger uncertainty than any other feedback. Despite intense research work in this area (see, for instance, the review articles of Stephens (2005) and Bony et al. (2006)), recent comparisons among climate models still show a large variability of the predicted change in global mean CRF if forcing factors such as a doubling of the atmospheric CO_2 content or an increase in sea surface temperature (SST) are prescribed (Ringer et al., 2006; Williams and Tselioudis, 2007). Figure 1.2 reproduces the results of a modelling study conducted by Ringer et al. (2006). The change in CRF, as computed by the current generation of climate models, varies not only in magnitude but also in sign – it is currently unclear whether clouds are changing so as to amplify global warming or to counteract it.

Williams and Tselioudis (2007) investigated how different cloud regimes contribute to the cloud feedback uncertainty. For the six general circulation models (GCMs) they compared, they found that differences in the radiative response of frontal clouds in the mid-latitudes and of stratocumulus clouds in low-latitude regions cause the largest proportion of the variance in the global cloud response. Bony and Dufresne (2005) also found low-latitude marine boundary layer clouds to be a major factor in the

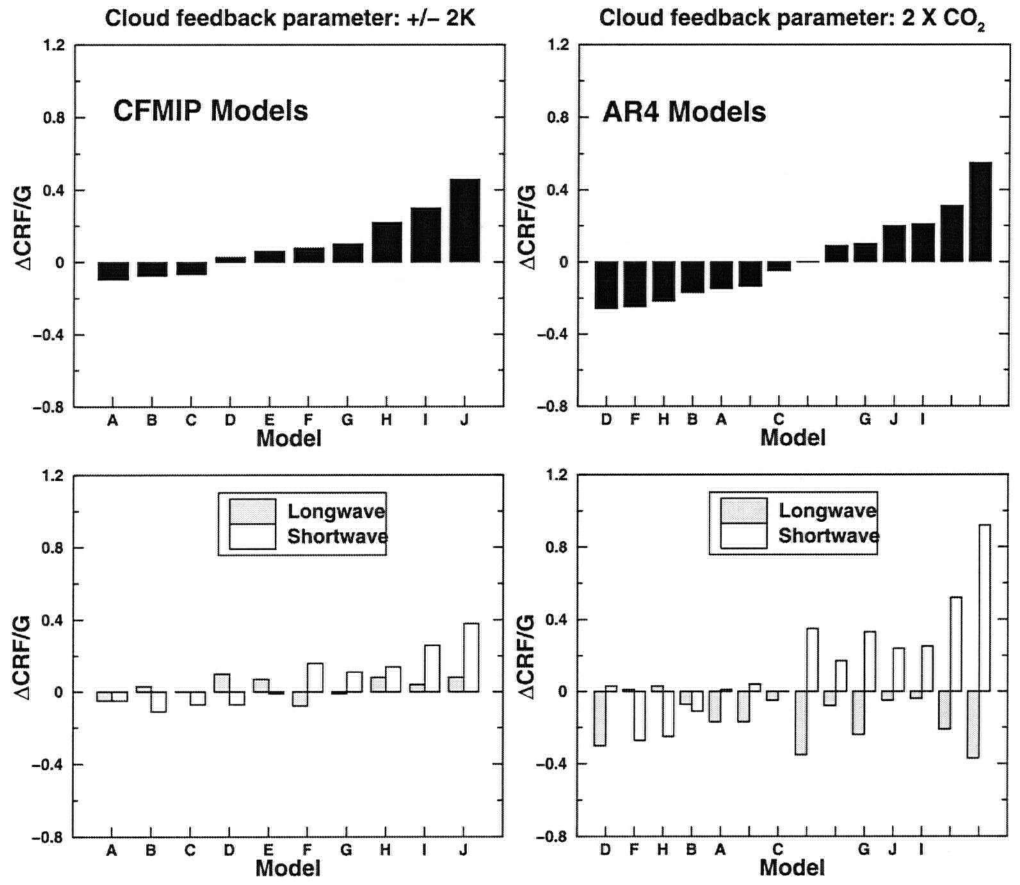


Figure 1.2: Cloud feedback processes represent a large uncertainty in climate modelling. Macroscopic and microscopic cloud properties cannot be resolved explicitly in the large grid cells of general circulation models, and different parameterisations lead to a large range of different feedbacks – from global mean cooling to warming of atmospheric temperature. Shown on the left is the global mean change in cloud radiative forcing (ΔCRF) in ten climate models participating in the Cloud Feedback Model Intercomparison Project (CFMIP), normalised by the radiative imbalance G resulting from a perturbation of the sea surface temperature in the simulations by $\pm 2K$ (top: total cloud feedback; bottom: longwave and shortwave components separated). On the right the forcing G is given by doubling the CO_2 content of the atmosphere, and the models used are the ones submitted to the IPCC 4th Assessment Report (AR4). $\Delta CRF/G = 1$ means that the radiative forcing associated with cloud feedback is the same as the original direct forcing G, i.e. the clouds double the forcing exerted by G. Note the large differences in cloud feedback amongst the models (reprinted from Ringer et al., 2006, © 2006, with permission from the American Geophysical Union).

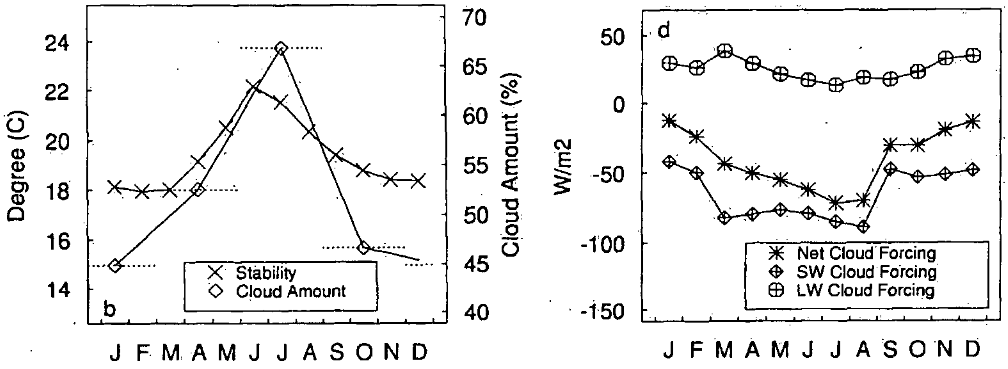


Figure 1.3: The annual cycle of (left) cloud amount and lower tropospheric stability (expressed as the potential temperature difference Θ(700 hPa) - Θ(sea level pressure)) and (right) shortwave, longwave and net cloud forcings from a two-year ERBE (Earth Radiation Budget Experiment) climatology in the region 20°-30°N and 120°-130°W off the coast of California (reprinted from Klein and Hartmann, 1993, © 1993, with permission from the American Meteorological Society).

disagreement of GCMs in cloud feedback predictions. Williams and Tselioudis (2007) note that the uncertainty is already present in model simulations of current climate, before the introduction of additional uncertainty from future climate forcing. This indicates that there is currently no consensus on how to represent shallow boundary layer cloud processes in climate models.

1.2 Marine Stratocumulus Clouds and the Aerosol Indirect Effect

1.2.1 Bulk Properties

The retrieval algorithm developed in this thesis focuses on the observation of marine stratocumulus (Sc) clouds. These clouds are so important to the climate problem because of their persistence and their high albedo (typically ~0.6-0.8 compared to the underlying sea surface value of typically ~0.05 for water (Driedonks and Duynkerke, 1989)). This causes a strong shortwave CRF especially in the low-latitude regions, where the incident solar radiation is very high. For example, off the coast of California, average cloud amount reaches more than 65% during the summer months, accounting for a net CRF of up to -70 W/m^2 (Klein and Hartmann, 1993, their Figure 5b,d is reproduced in Figure 1.3).

Marine Sc particularly favour the eastern subtropical oceans, where cold surface waters from the upwelling ocean currents produce boundary layer air temperatures that are cool compared to the warm, subsiding air aloft. The resulting strong temperature inversion caps the boundary layer and inhibits deep convection. Strong radiative cooling at cloud top helps to maintain the shallow convection and thus the cloud layer in the absence of strong sensible heat flux from the ocean surface (Klein and Hartmann, 1993). Recently, it has been recognised that drizzle also plays an important role in the dynamics of the cloud layer. However, how exactly precipitation processes interact with the circulation is still subject to ongoing research (e.g. Ackerman et al., 2004; Stevens et al., 2005; Savic-Jovcic and Stevens, 2007).

These complex physical processes, together with the clouds' small geometrical thickness (typically a few hundred metres) and the sharp capping inversion makes the representation of marine Sc in climate models particularly difficult (Bretherton et al., 2004). They are also still a problem for weather forecasting models, as a recent comparison of in-situ data taken during the second Dynamics and Chemistry of Marine Stratocumulus (DYCOMS-II) field experiment (Stevens et al., 2003a) with numerical weather prediction results shows (Stevens et al., 2007).

Several field experiments have been conducted during which in-situ and remotely sensed data of marine Sc were taken in order to better comprehend the mechanisms involved in sustaining and dissipating this type of cloud. Figures 1.4 and 1.5 give examples of typical vertical cloud profiles observed in marine Sc. The data in Figure 1.4 were taken during FIRE, the First ISCCP (International Satellite Cloud Climatology Programme) Regional Experiment (Albrecht et al., 1988), in precipitating Sc off the coast of California. Cloud liquid water content and average droplet radius typically follow their adiabatic values[2], with liquid water content often becoming subadiabatic at the top of the clouds by entrainment of dry air from aloft or by removal of droplets by precipitation (drizzle). The mean droplet concentration typically stays approximately constant with height, and the geometrical thickness of the clouds observed in this example is less than 300 m.

Figure 1.5 summarises the results compiled by Miles et al. (2000) from in-situ data reported in the literature. The magnitude of average cloud top droplet radii ranges from about 4 to about 12 μm.[3]. The effective radius, an area-weighted mean radius of the cloud droplets which is important in radiative transfer applications (see section 1.3 and Chapter 2), is 5 to 15 μm, a little larger than the mean droplet radius. Mean droplet number concentrations range from less than 20 cm^{-3} to about 200 cm^{-3} in marine clouds. As a comparison, number concentrations typical of continental clouds are also shown. Due to the higher aerosol content of continental air masses, which act as cloud condensation nuclei (CCN), significantly higher concentrations can be observed over land.

[2] *Adiabatic* refers to a well mixed cloud layer in which entropy and total water content (i.e. water vapour plus liquid water) are constant with height.

[3] Note that Miles et al. (2000) use diameter to describe droplet sizes in their work, whereas I will use radius in this thesis.

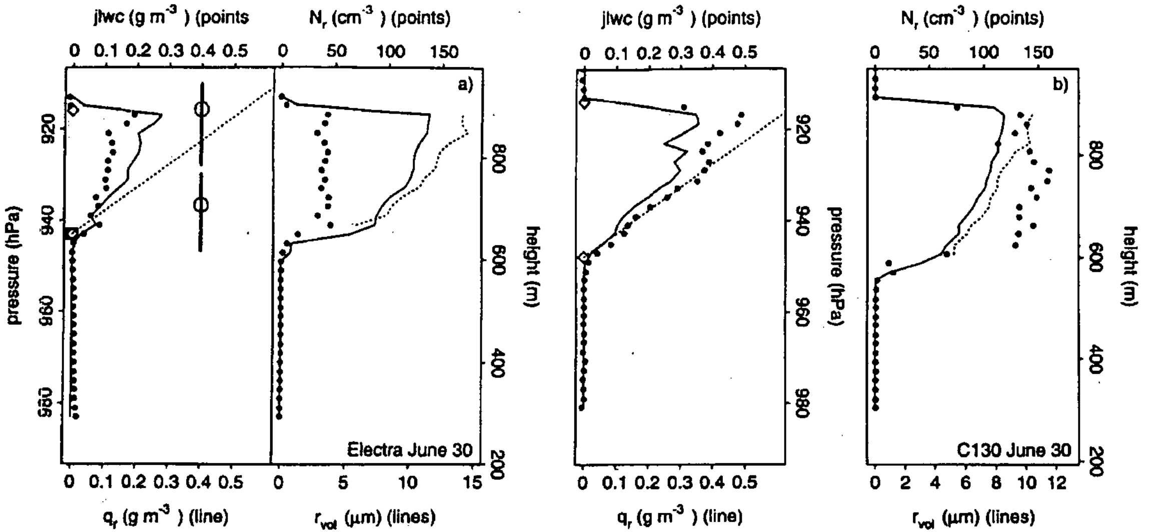


Figure 1.4: Example of in-situ measurements of microphysical cloud properties taken during the First ISCCP (International Satellite Cloud Climatology Programme) Regional Experiment (FIRE; Albrecht et al., 1988) in precipitating marine stratocumulus clouds off the coast of California. Shown are data from two aircraft flights (left and right): in-situ measured liquid water content (q_r, left side of each panel, solid line), adiabatic liquid water content (left side, straight dashed line), in-situ measured and adiabatic volume-mean radius (r_{vol}, right side, solid and dashed, respectively), and in-situ measured number concentration (N_r, right side, points). ◊ marks cloud-top and cloud-base (reprinted from Austin et al., 1995, © 1995, with permission from the American Meteorological Society).

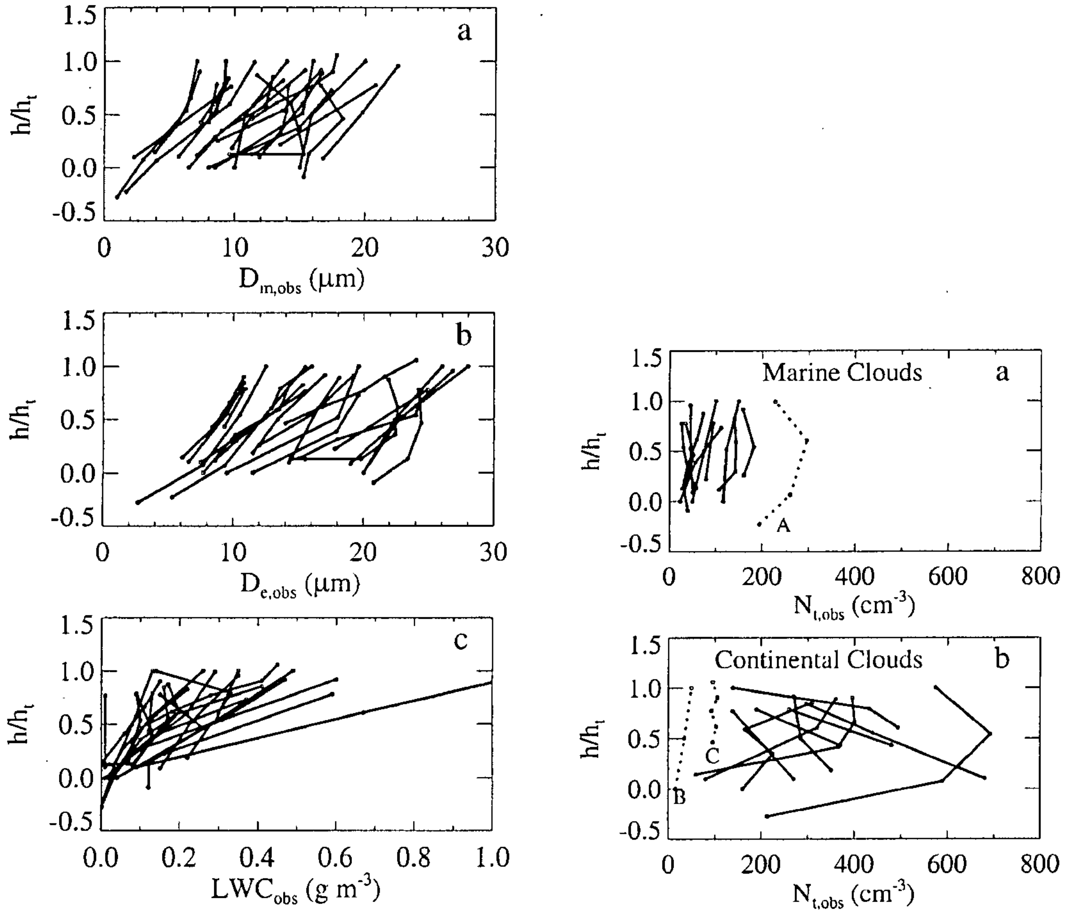


Figure 1.5: Miles et al. (2000) collected observations of low-level stratiform clouds from previously published studies in order to compare the results found by different authors. The figures shown here illustrate typical ranges of mean and effective particle diameter (D_m and D_e), cloud liquid water content (LWC) and particle number concentration (N_t). On the left profiles of both continental and marine clouds are plotted together versus the normalised cloud height ($h/h_t = 0$ marks cloud base, $h/h_t = 1$ cloud top), whereas in the plots of the number concentration on the right they are separated. Marine clouds typically have much smaller number concentrations than continental clouds. This is due to fewer aerosols acting as cloud condensation nuclei (CCN) over the oceans. Furthermore, N_t can be regarded as approximately constant with height within a cloud. Note that the figures on the left use diameter to describe cloud droplet size, whereas in this thesis radius is used (reprinted from Miles et al., 2000, © 2000, with permission from the American Meteorological Society).

Due to their occurrence in strong subsidence regions, marine Sc cloud tops are usually found at low altitudes of less than 1000 m (e.g. Driedonks and Duynkerke, 1989). Liquid water paths (LWP) are found to be on the order of up to a few hundred g/m^2, often well below 100 g/m^2 (e.g. Bretherton et al., 2004; Xu et al., 2005).

A distinctive feature of marine Sc decks is the occurrence of a pronounced diurnal cycle in cloud amount and LWP. For instance, during the East Pacific Investigation of Climate (EPIC), Bretherton et al. (2004) observed a rise of the inversion height (and thus cloud top height) of roughly 200 m each night from an early afternoon minimum value. The cloud base showed little diurnal cycle, so that the clouds were thinnest during the afternoon. Furthermore, cloud cover was almost 100% during nighttime, but during the afternoon clouds were often broken. Such diurnal cycles in cloud amount were also found by Rozendaal et al. (1995), who used low cloud fraction data inferred from infrared satellite channels in the ISCCP dataset. Wood et al. (2002) analysed the LWP derived from microwave measurements from the Tropical Rainfall Measuring Mission (TRMM) satellite. They found a strong diurnal cycle in LWP of subtropical low cloud regions, which with an early morning peak was in phase with the cloud amount cycle of Rozendaal et al. (1995). Similar results were found by Blaskovic et al. (1991).

The diurnal cycle of cloud thinning during the day and growth at night is mainly caused by solar forcing. During the day, the shortwave radiation that is absorbed in the cloud layer largely offsets the longwave radiative cooling from cloud top. This inhibits the turbulence in the layer, and the resulting less well-mixed structure is both less efficient in transporting moisture upwards from the surface and maintaining the cloud top by entrainment. Consequently, subsidence can advect the cloud top downwards, while it is also dried out. Typically, the cloud base is also lifted during the day. This leads to a commensurately thinner cloud (Bretherton et al., 2004; Stevens et al., 2007).

Daily amplitudes of the variations reach 5-10% for cloud amount (Rozendaal et al., 1995) and 15-35% for LWP (Wood et al., 2002), so that a correct representation of the diurnal cycle in GCMs is critical to the accurate representation of low clouds in large scale models.

1.2.2 Aerosols

Intimately linked to low boundary layer clouds are the aerosol indirect effects (AIEs), important for the estimation of the climate impact of anthropogenic aerosol emissions. Twomey (1974, 1977) proposed that an increased aerosol concentration shifts the droplet size distribution towards smaller values by providing more cloud condensation nuclei. If cloud liquid water content (LWC) is unchanged, more and commensurately smaller droplets are formed, which increase the albedo of the cloud (first AIE of Twomey effect).

A second key hypothesis was introduced by Albrecht (1989). He suggested that smaller cloud droplets suppress the formation of precipitation in the cloud.[4] This reduced cloud water sink would moisten the atmospheric boundary layer (ABL), leading to a lower cloud base and a thicker cloud that could live longer (second AIE or cloud lifetime effect). The effect on the radiation budget is similar to the Twomey effect – increasing cloud thickness increases the albedo, and clouds that persist over longer time spans also have an increased albedo if we consider the time average.

Both effects are important in the context of anthropogenic emissions. For instance, an idealised climate sensitivity study conducted by Hu and Stamnes (2000) investigated the sensitivity of the CRF to microphysical properties of clouds. As an illustration, Figure 1.6 shows the dependence of shortwave CRF on average cloud droplet radius for three different cloud thicknesses. If the average droplet size was decreased by just one micrometer, the effect on the radiation budget could already be significant. The figure also shows that thin clouds (given with a LWP of 50 g/m^2), such as marine Sc, exhibit the largest radiative sensitivity to microphysical changes.

However, the interactions between aerosols and clouds are complicated. Precipitation, including drizzle that does not reach the ground, occurs and can influence the aerosol concentration via scavenging, which in turn influences the cloud microphysics and cloud fraction (Stevens et al., 2005; Sharon et al., 2006). Also, precipitation is known to stabilise the cloud layer (by cooling the subcloud layer and increasing static stability), so less precipitation caused by the second AIE would increase turbulence and thus entrainment of warm and dry air from aloft – counteracting the cloud thickening process (Wood, 2007).

In total, the aerosol indirect effects make the problem of accurately representing boundary layer cloud processes in GCMs even more complicated, adding significantly to the already existing cloud feedback uncertainty (Lohmann and Feichter, 2005). In fact, whether higher aerosol concentrations actually increase the area-averaged albedo of a cloud deck and thus its CRF is discussed controversially in the literature. Recently, studies suggested that processes might exist that could cancel the indirect effect completely, leaving no effect of aerosols on the net radiative fluxes (Twohy et al., 2005; Wood, 2007; Xue et al., 2007).

A intriguing manifestation of the AIE are ship tracks (Hobbs et al., 2000; Durkee et al., 2000b,a). They occur when elevated aerosol levels in the area of a ship plume lead to an enhanced reflectivity of the cloud layer above the ship. The resulting "ship tracks" can often be observed on satellite imagery. Indeed,

[4] In ice-free clouds, large droplets are needed to form precipitation (via the collision/coalescence process). If the droplet size distribution is shifted towards smaller values, the propensity of the cloud to form precipitation decreases.

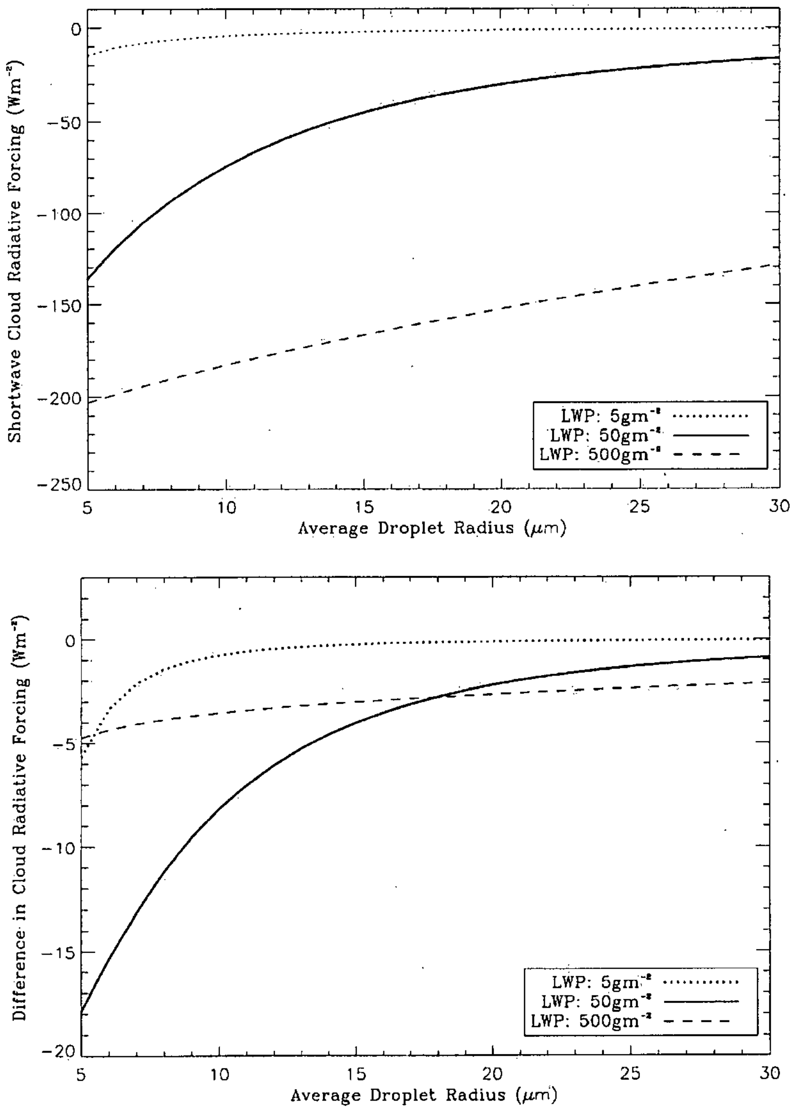


Figure 1.6: Example of the dependence of cloud radiative forcing on macroscopic and microscopic cloud properties: the dependence of shortwave CRF at the tropopause (global annual average, top) on average cloud droplet radius for three different cloud thicknesses, and its change if the average droplet radius is decreased by 1 μm (bottom) in an idealised climate sensitivity study. To put the numbers into context, Wielicki et al. (1995) estimate the globally averaged shortwave CRF to be on the order of -50 Wm^{-2}; an instantaneous doubling of CO_2 would result in a forcing of roughly 4 Wm^{-2} (reprinted from Hu and Stamnes, 2000, © 2000, with permission from Blackwell Publishing).

Durkee et al. (2000a) found that ships that emit more aerosols on average produce ship tracks that are brighter, wider, and longer-lived than ships with lower emissions. Here, I will use the phenomenon of ship tracks to check the physical consistency of the retrieval method (Chapter 4).

1.3 Radiative Transfer Terminology

Before I discuss the motivation for this work and give an overview of the relevant existing literature, it is useful to review some basic radiative transfer terminology. General references in which more detailed information can be found are, for instance, Petty (2006) or Bohren and Clothiaux (2006).

The *flux density*, or *irradiance* (sometimes abbreviated as *flux*), is a measure of the total energy per unit time and unit area transported by electromagnetic radiation through a flat surface:

$$F = \frac{\text{energy (Joules)}}{\text{time (seconds)} \times \text{area (m}^2\text{)}}. \tag{1.1}$$

It is measured in Wm^{-2}.

The *radiance*, or *intensity*, measures the directional energy transport:

$$I = \frac{\text{energy (Joules)}}{\text{time (seconds)} \times \text{area (m}^2\text{)} \times \text{field of view (sr}^{-1}\text{)}}. \tag{1.2}$$

It is measured in $\text{Wm}^{-2}\text{sr}^{-1}$. The direction is given by solid angle with units steradian (sr). A steradian is a "square radian"; solid angle is to "regular" angle as area is to length (Petty, 2006). An integration of solid angle over one hemisphere yields 2π, over an entire sphere 4π.

Both flux and intensity can be expressed in monochromatic form, i.e. per unit wavelength, λ (or, alternatively, wave number, $1/\lambda$):

$$I_\lambda = \frac{I}{\text{wavelength } (\mu\text{m})}, \tag{1.3}$$

with units $\text{Wm}^{-2}\text{sr}^{-1}\mu\text{m}^{-1}$.

When electromagnetic radiation is incident on a medium, part of it is absorbed, part reflected, and part transmitted. The *absorptivity* (also *absorptance*) a, *reflectivity* (also *reflectance*) r and *transmissivity* (also *transmittance*) t of a medium describe the corresponding fractions of the incident radiation and in general depend on wavelength and direction of the incident radiation. The shortwave reflectivity of a surface is also referred to as its (shortwave) *albedo*. Obviously, all three quantities range from zero to one, and

$$a_\lambda(\theta,\phi) + r_\lambda(\theta,\phi) + t_\lambda(\theta,\phi) = 1. \tag{1.4}$$

The transmissivity is described by *Beer's Law*:

$$t_\lambda = \frac{I_\lambda(s)}{I_{\lambda,0}} = \exp\left(-\beta_{\lambda,e} s\right), \tag{1.5}$$

where s denotes distance along the direction of propagation, $I_{\lambda,0}$ the intensity at position $s = 0$, $I_\lambda(s)$ the intensity after distance s, and $\beta_{\lambda,e}$ the *extinction coefficient* (m^{-1}). It describes the rate of energy attenuation per unit distance ($1/\beta_{\lambda,e}$ determines the distance for energy to be attenuated to e^{-1} of its original value).

Since all of the following equations correspond to the monochromatic case, I will drop the subscript λ for convenience. In equations that describe wavelength-integrated cases this will be explicitly stated.

Following Beer's Law, in a direct beam with no sources of radiation, the intensity I falls off exponentially with distance:

$$I(s) = I_0 \exp\left(-\beta_e s\right). \tag{1.6}$$

In this equation, the extinction coefficient describes the effects of two mechanisms for extinction; radiation can be either absorbed by a medium, or be scattered out of its original direction of propagation. The extinction coefficient is the sum of an absorption coefficient β_a and a scattering coefficient β_s:

$$\beta_e = \beta_a + \beta_s. \tag{1.7}$$

The *single scatter albedo* $\tilde{\omega}$ characterises the relative importance of scattering in the total extinction:

$$\tilde{\omega} = \frac{\beta_s}{\beta_e} = \frac{\beta_s}{\beta_a + \beta_s}. \tag{1.8}$$

In a purely absorbing medium, $\tilde{\omega}$ would be zero, whereas in a purely scattering one it would be unity.

The dimensionless *optical thickness* of a cloud between z_1 and z_2 (in an atmosphere in which z represents the vertical coordinate) indicates the opacity of the cloud for a given wavelength. It is defined as the path-integrated extinction coefficient and hence expresses how much extinction occurs over the geometrical thickness of the cloud:

$$\tau(z_1, z_2) = \int_{z_1}^{z_2} \beta_e(z)dz. \tag{1.9}$$

As a rule of thumb, Bohren et al. (1995) found that at an approximate visible optical thickness of 10 one can no longer see the sun through a cloud.

The water droplets found in a cloud are generally distributed over a range of sizes. The *droplet size distribution* is described by

$$n(r)dr = \text{number of droplets per unit volume with radii between } r \text{ and } r + dr, \tag{1.10}$$

and it is important for determining the *effective radius* r_{eff}, which is defined as

$$r_{eff} = \frac{\int n(r) r^3 dr}{\int n(r) r^2 dr}. \tag{1.11}$$

Two cloud parcels with identical liquid water content and r_{eff} have the same total droplet surface area, independent of the actual droplet size distribution.

The absorption and scattering coefficients β_a and β_s can be written in terms of the droplet size (and wavelength) dependent absorption and scattering *cross sections* $\sigma_a(r)$ and $\sigma_s(r)$ (m^2):

$$\beta_a = \int \sigma_a(r) n(r) dr \tag{1.12}$$

$$\beta_s = \int \sigma_s(r) n(r) dr. \tag{1.13}$$

$\sigma_a(r)$ and $\sigma_s(r)$ hence describe the absorption or scattering per particle. Water droplets larger than about 5 μm have a scattering cross section of $\sigma_s = 2\pi r^2$ and an absorption cross section of $\sigma_a \approx 0$ at visible wavelengths. It can be shown (Petty, 2006) that in this case the visible optical thickness is related to the effective radius by

$$\tau_{vis} \approx \frac{3\,\mathrm{LWP}}{2\,\rho_l\, r_{eff}}, \tag{1.14}$$

where ρ_l is the density of liquid water and the liquid water path (g m^{-2}) is given by

$$\mathrm{LWP} = \int_0^z q_l dz = \int_0^z \int_0^\infty n(r,z) \left[\rho_l \frac{4\pi}{3} r^3\right] dr\, dz, \tag{1.15}$$

with liquid water content q_l (g m^{-3}). Although (1.14) does not apply at absorbing wavelengths where σ_a and σ_s vary with r and λ, it is still the case that cloud reflectivity and emissivity (defined below) can be written as a unique function of r_{eff}, τ_{vis}, temperature and λ for all wavelengths.

Finally, the intensity emitted by a medium is given by the *Planck function* and the *emissivity* ε:

$$I_\lambda = \varepsilon_\lambda B_\lambda(T), \tag{1.16}$$

where T is the temperature of the emitting material. The emissivity describes how efficient the medium is emitting, and after *Kirchhoff's Law*, the emissivity of a material is equal to its absorptivity:

$$\varepsilon_\lambda(\theta, \phi) = a_\lambda(\theta, \phi). \tag{1.17}$$

The Planck function, dependent on the temperature of a medium, is given by

$$B_\lambda(T) = \frac{2hc^2}{\lambda^5 \left(\exp\left(\frac{hc}{k_B \lambda T}\right) - 1\right)}, \tag{1.18}$$

where $h = 6.626 \times 10^{-34}$ Js^{-1} is Planck's constant, $c = 2.998 \times 10^8$ ms^{-1} is the speed of light, and $k_B = 1.381 \times 10^{-23}$ J/K is Boltzmann's constant. It gives the emission of a *black body*, an object that absorbs all radiation that falls onto it ($a_\lambda = 1$) and consequently also emits the maximum possible radiation that a material with a given temperature can emit ($\epsilon_\lambda = 1$).

The *brightness temperature* T_B is often used as a substitute for describing the measured intensity in remote sensing. It is the temperature a black body must have in order to emit the observed radiance at a given wavelength:

$$I_\lambda = B_\lambda(T_B). \tag{1.19}$$

1.4 Motivation

Data that can shed light on the regional variations in cloud microphysical and optical properties, their diurnal cycle and the interaction of clouds, precipitation and aerosols is particularly important for further progress in the accurate representation of stratocumulus clouds in climate and weather models (Stevens et al., 2003a). Figure 1.7 summarises the options the scientific community has to observe clouds. In-situ instruments carried by aircraft can very accurately measure drop size distribution and liquid water content, as well as the standard meteorological variables. While field campaigns such as DYCOMS-II yield valuable data, the aircraft flights are expensive, and it is not practical to use in-situ sampling to conduct

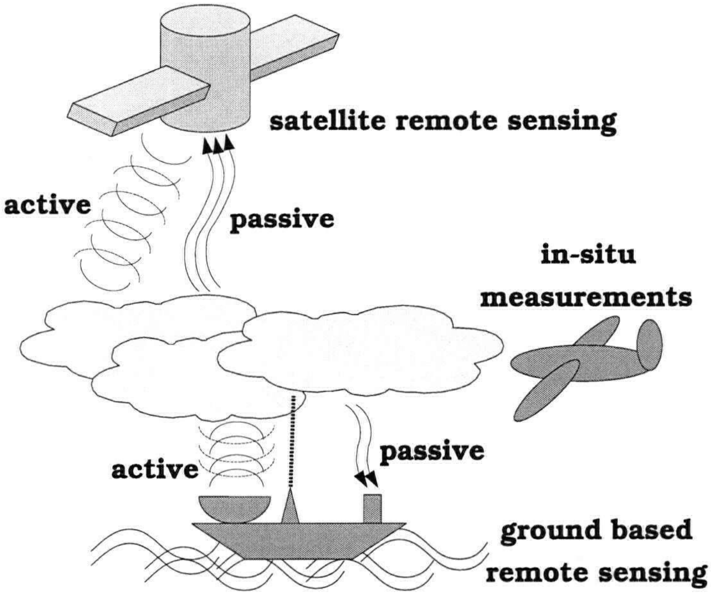


Figure 1.7: Observations of cloud microphysical and optical properties, their diurnal cycle and the interaction of clouds, precipitation and aerosols are important for further progress in the accurate representation of stratocumulus clouds in climate and weather models. Different methods can be employed; active (radar, lidar) and passive (radiometers) remote sensing can be used from the ground. Aircrafts or helicopter-based instrument-sondes are able to perform in-situ measurements and directly measure particle size, number and the standard meteorological parameters. However, for marine clouds, both ground based remote sensing (from ships) and in-situ measurements are cost and labour expensive. Satellites provide an ideal means to remotely sense data over large areas and long time spans. Most satellite-based approaches use passive multispectral radiometer data, but data from active satellite instruments will also be available.

the long-term observations that are needed to understand how the climatology of marine Sc responds to environmental change.

Remote sensing from surface and satellite based instruments can provide a long-term data record, with surface instruments typically being able to achieve a higher temporal resolution but with a limited spatial field of view. Also, it is difficult to deploy surface remote sensing instruments for long time spans over sea, leaving satellites to be the major data source for large-area and climatic studies of marine Sc.

Remote sensing instruments infer cloud properties from radiation that is reflected, transmitted or emitted by the cloud. While active instruments such as radar, sodar or lidar emit electromagnetic radiation and infer the property of interest from the backscatter signal, passive sensors record the radiation that originates from natural sources. Mathematical models of the radiative processes in the cloud and atmosphere are then employed to compute the intensity arriving at the sensor (i.e. for satellites, at the top of the atmosphere) as a function (referred to below as the *forward model*) of the relevant cloud parameters:

$$\boldsymbol{I}^{TOA} = \boldsymbol{f}(r_{eff},\ \tau,\ T_{cloud},\ T_{surface},\ \text{overlying atmosphere, ...}), \tag{1.20}$$

where the intensity has been written as a vector in order to account for observations at several wavelengths. Other parameters, indicated by "..." in the above equation, could include subadiabaticity, cloud inhomogeneity or partially filled pixels. The retrieval is then defined as the inversion of the function $\boldsymbol{f}$,

$$\boldsymbol{\varphi} = \boldsymbol{f}^{-1}(\boldsymbol{I}^{TOA},\ \text{overlying atmosphere, ...}), \tag{1.21}$$

where the vector $\boldsymbol{\varphi} = (r_{eff},\ \tau,\ T_{cloud},\ T_{surface},\ ...)$ contains the parameters to be determined.

Most satellites currently in orbit carry passive sensors, due to the high power consumption of active technology. For instance, the Moderate Resolution Imaging Spectroradiometer (MODIS) instruments, on-board the National Aeronautics and Space Administration's (NASA) near-polar orbiting Terra and Aqua satellites, measure in the visible and infrared wavelength range and provide four images daily for a given location on earth. While an operational cloud product including droplet size and optical thickness (a measure of how much radiation can pass through the cloud) is routinely retrieved from the daytime measurements (King et al., 1997; Platnick et al., 2003)[5], no such retrievals are available for nighttime images.[6][7] However, such retrievals would be useful for studies of marine Sc on the diurnal timescale.

One reason that no operational retrievals of nocturnal cloud properties are available is that solar radiation carries several orders or magnitude more energy than radiation emitted by terrestrial sources. The sun hence represents a source of very easily detectable photons, and the absence of emission of terrestrial sources in the visible wavelength range means that cloud temperature is not a confounding variable. The higher intensity allows for smaller pixel sizes, too. For instance, while near and thermal infrared signals are recorded at a 1-km resolution by the MODIS sensors, the visible channels have a 250-m resolution. Furthermore, most information on cloud microphysical and optical properties are carried by wavelengths in the visible and near infrared. While the reflected sunlight at visible channels is highly

[5] Available from http://ladsweb.nascom.nasa.gov/data/.

[6] This is also true for other instrument data, such as from the Advanced Very High Resolution Radiometer (AVHRR).

[7] In 2006, NASA launched the CloudSat and CALIPSO satellites within the so-called A-Train – a series of satellites equipped with special cloud remote sensing instrumentation, flying in formation with the Aqua satellite in order to provide near-simultaneous measurements. The satellites are equipped with an active radar and a lidar instrument, respectively, which operate independently of the available sunlight. However, the lidar instrument will mainly focus on upper level clouds (i.e. cirrus), and the radar on vertical profiles of LWC and precipitation – droplet size, for instance, will not be part of the retrieval product (Stephens et al., 2002; Vaughan et al., 2004)

sensitive to cloud optical thickness, near infrared channels provide information about cloud particle size (e.g. King et al., 1997).

However, a similar, although weaker, dependence can also be found at non-visible wavelengths. Radiative transfer calculations by Baum et al. (1994) showed that a combination of the three near infrared and thermal channels of the Advanced Very High Resolution Radiometer (AVHRR) could be used to infer cloud droplet size, optical thickness and cloud temperature. Attempts to construct a retrieval scheme from such measurements (Minnis et al., 1995; Heck et al., 1999; Pérez et al., 2000; González et al., 2002; Pérez et al., 2002) were partly successful, but suffered from high computational cost and included only simple sensitivity analyses in order to estimate the uncertainty on the retrievals. In particular, the computational time needed to retrieve an individual pixel in an image was on the order of seconds[8], which, again, is not suited to build a climatology build on a large number of satellite images. Also, ambiguities in the inverse mapping were a problem (i.e. different droplet size and optical thickness combinations can lead to the same set of emitted radiances), further complicating the uncertainty estimation and restricting the quantitative usefulness of the retrievals (Pérez et al., 2000; González et al., 2002).

Towards these ends, a retrieval algorithm both fast enough to process large datasets and able to provide reliable error estimates on the retrieved values could prove quite useful, and its non-existence motivates this study.

1.5 Overview and Theory of Existing Retrieval Algorithms

1.5.1 Daytime Methods

Retrieval techniques to extract boundary layer liquid water cloud properties from satellite measurements have been developed since the 1980s (Arking and Childs, 1985; Twomey and Cocks, 1989; Nakajima and King, 1990; Nakajima et al., 1991; Platnick and Twomey, 1994; Han et al., 1994; Platnick and Valero, 1995; Nakajima and Nakajma, 1995; Kawamoto et al., 2001), although interest in inferring cloud cover characteristics from satellite images can be traced back to Arking (1964). Since reflected solar radiation is observed by the sensors, and the reflectivity mainly depends on cloud optical thickness and effective radius, these two parameters are commonly retrieved.

The underlying principle is illustrated in Figure 1.8. Liquid water clouds are composed of spherical water droplets whose scattering properties can be described by Mie theory (Mie, 1908). Figure 1.8a shows

[8] on a 933 MHz Pentium III machine; P.H. Austin, 2005, per. comm.

the dependence of the single scatter albedo $\tilde{\omega}$ (1.8) on wavelength for three different droplet sizes. In the visible wavelength range, $\tilde{\omega}$ is very close to unity for all particle sizes, and no radiation is absorbed in the cloud. The reflectance of a cloud layer, i.e. its albedo, is determined by the amount of radiation that is scattered back into space, i.e. by the number of photons that leave the cloud at its top after various scattering events. Since water droplets tend to scatter visible radiation mainly in the forward direction[9], it needs strong and a large number of scattering events for a high albedo.

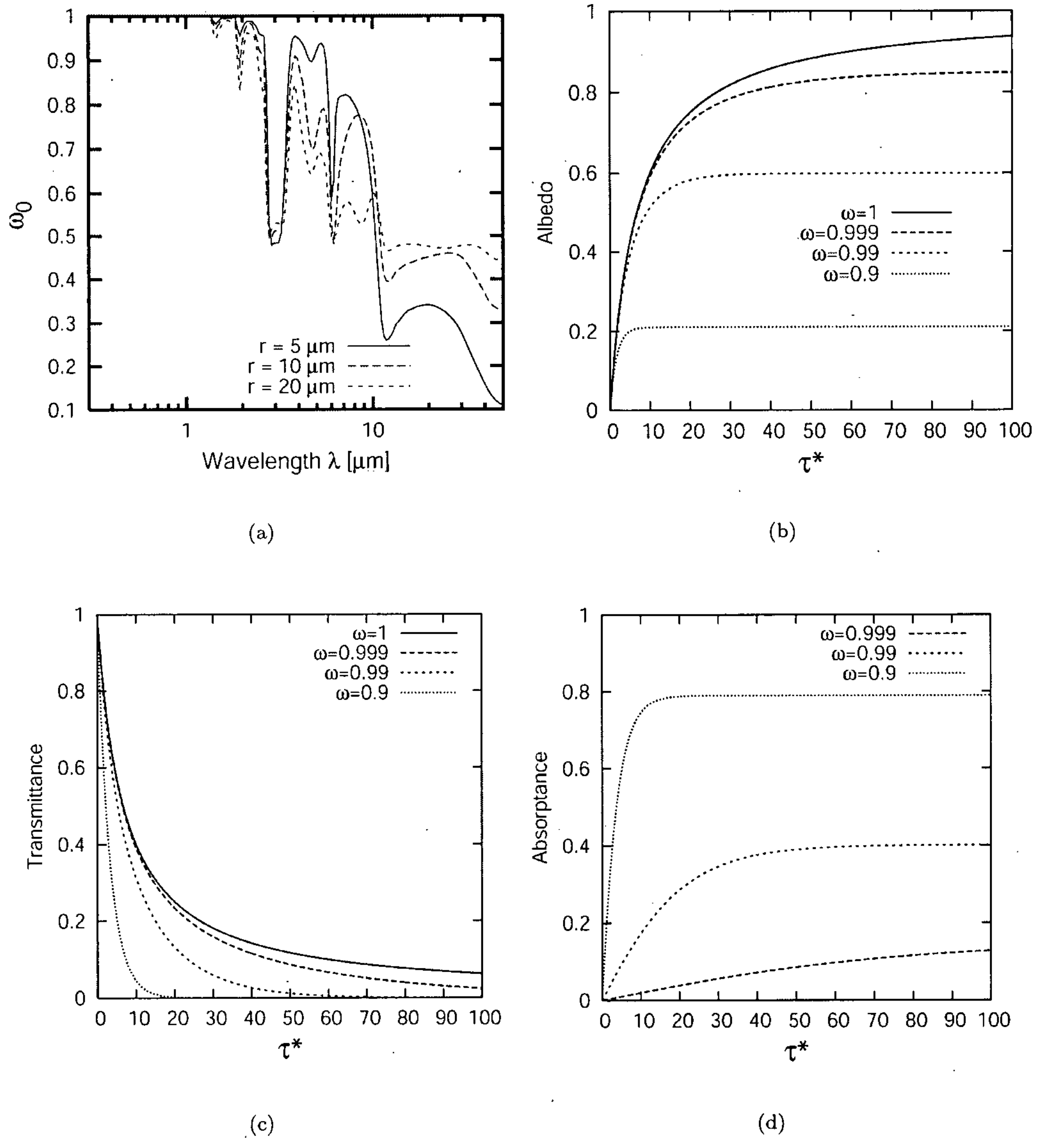
If all extinction is due to scattering, an optically thicker cloud will have a larger albedo. Since $\tilde{\omega}$ depends only weakly on particle size in the visible wavelength range, the optical thickness of a cloud can well be inferred from the measured reflectance (Figure 1.8b, the curve for $\tilde{\omega} = 1$).

Figure 1.8a shows that for wavelengths in the near and thermal infrared, the single scatter albedo becomes dependent on particle size. Furthermore, for sufficiently low $\tilde{\omega}$ and sufficiently thick clouds, the cloud albedo becomes independent of τ (illustrated in Figure 1.8b for $\tilde{\omega} = 0.9$), leaving the major dependence on the effective droplet radius r_{eff}. For thin clouds, the retrieved optical thickness from the visible wavelengths can be used to eliminate the optical depth dependence. Typically, near infrared channels centred at 2.1 μm (Nakajima and King, 1990) or 3.7 μm (Nakajima and Nakajma, 1995) are used to retrieve particle size information.

To infer the values of cloud optical depth and effective radius from the satellite measurements, a radiative transfer model is employed to compute a database of reflected intensities, the lookup table (LUT). Radiances that would be expected at the satellite sensor are computed using simplified models of the cloud layer and the atmosphere. For instance, the operational MODIS retrieval (King et al., 1997), based on the work of Nakajima and King (1990), uses a simple model of horizontally and vertically uniform clouds whose only parameters are r_{eff} and τ. For the atmosphere, absorption by water vapour or trace gases (mainly infrared) or scattering at aerosols (visible) has to be considered. Also, emittance of cloud droplets in the near infrared bands plays a role in the observed intensities. It is often removed by utilising further thermal bands for estimates of cloud temperature.

The components of cloud, atmosphere, and radiative transfer model together are called the forward model. Since the forward model is too complex to be inverted analytically, it is run for a variety of cloud parameters to build a LUT. The determination of the cloud properties is then done by "looking up" the measured intensities in the database – the difference between observed and computed radiances is

[9] More details will be discussed in Chapter 2.



(a)

(b)

(c)

(d)

Figure 1.8: Example of how radiative effects are dependent on microphysical properties of a cloud. (a) Dependence of the single scatter albedo $\tilde{\omega}$ (the fraction of radiation extinguished along a ray of radiation that is due to scattering effects) on wavelength for three different cloud droplet sizes (5, 10 and 20 μm). (b) Idealised albedo, (c) transmittance and (d) absorptance of a plane parallel layer cloud in dependence on cloud optical thickness τ for different values of $\tilde{\omega}$. Especially note the different $\tilde{\omega}$ for the satellite channels measuring at 3.7, 11 and 12 μm in (a). The resulting differing albedo, transmittance and absorptance can be exploited for remote sensing of droplet size and optical depth (reprinted from Petty, 2006, © 2006, with permission from Sundog Publishing).

represented by a cost function (also referred to as error surface)

$$\mathrm{COST} = (I_{TOA}^{observed} - I_{TOA}^{computed})^2, \tag{1.22}$$

which is minimised numerically.

The above mentioned studies all utilise these fundamental techniques, differing in the wavelength of the employed channels or the minimisation algorithm used. Also, forward models were refined in later publications, for instance the treatment of water vapour (Kawamoto et al., 2001), or the cloud model, which now is often assumed to be adiabatic rather than vertically uniform (Brenguier et al., 2000).

An important assumption in most retrieval schemes is the plane-parallel approximation, which treats the cloud layer within a pixel as horizontally homogeneous. While this might seem to be an oversimplification for many realistic clouds, it is often a good approximation for stratiform clouds at scales of about 1 km (the resolution of the MODIS channels) and considerably simplifies the radiative transfer calculations (Petty, 2006). The cloud parameters are then retrieved on a pixel-by-pixel basis, implying that the observed radiances are determined entirely by the properties of the clouds within the pixel of interest (independent pixel approximation, IPA).

Hence, care has to be exercised as to only apply the retrieval scheme to pixels that are fully covered by cloud – a requirement that can be problematic for regions of broken cloud. Unfortunately, precipitation produces inhomogeneities at scales of about 1 km (Stevens et al., 2005; Sharon et al., 2006). This poses obvious problems for remote sensing studies of the aerosol indirect effect; research showed a considerable error is introduced into the retrieval if inhomogeneities exist in the cloud layer, especially in the presence of broken clouds on the subpixel scale (Faure et al., 2001b,a,c, 2002; Kato et al., 2006; Iwabuchi, 2007).

1.5.2 Nighttime Methods

At night, the dependence of the radiative fields on cloud optical thickness and effective radius is slightly more complicated. Since radiation in the visible wavelengths is not available[10], the retrieval has to rely entirely on near and thermal infrared signals, resulting in the lack of the relatively independent optical thickness information in the visible range. Research work in this area dates back to Hunt (1973), whose work showed a difference in the sensitivity of cloud emissivity to changes in particle size between several wavelengths in the near and thermal infrared. Most follow-up studies in the 1970s and 1980,

[10] Although new highly sensitive radiometers are being developed that will be able to measure visible light reflected by the clouds from the moon (Lee et al., 2006).

however, were focused on cirrus clouds rather than shallow liquid water clouds (Minnis et al., 1995, and references therein). D'Entremont (1986) used the 3.7, 11 and 12 μm channels of the AVHRR instrument to determine fractional cloud amount and cloud top height for low and mid-level clouds, and Lin and Coakley (1993) and Luo et al. (1994) developed a method based on the 11 and 12 μm emissivities in order to determine fractional cloud cover and an effective particle radius for single layered clouds.

Baum et al. (1994) eventually found that the combination of the AVHRR channels at 3.7, 11 and 12 μm may be used for a simultaneous retrieval of cloud temperature, optical thickness and effective radius. At these wavelengths, the emitted radiation of a cloud depends on both effective radius and optical thickness, and additionally also on temperature. Figure 1.8a shows the strong dependence of $\tilde{\omega}$ on particle size for wavelengths larger than about 2 μm. Since, after Kirchhoff's Law, the emissivity of a droplet is equal to its absorptivity, the value of 1-$\tilde{\omega}$ gives an idea of how cloud emission as well as absorption depend on r_{eff}.[11] The cloud top radiance at a certain wavelength then becomes a complex function of cloud temperature (determining the black body emission), particle size, the thickness of the cloud (determining how many scattering or absorption/emission events occur), and, for clouds thin enough to allow for transmission, surface temperature (assuming a surface emissivity = 1). Of course, cloud inhomogeneities add further complexity.

Despite this complexity, the differences in the dependence of cloud droplet scattering, absorption and emission properties on wavelength can be exploited for the retrieval. While the objective of Baum et al. (1994) was to detect multilevel cloud situations, Minnis et al. (1995) and Heck et al. (1999) report on a nighttime retrieval algorithm within the Clouds and the Earth's Radiant Energy System (CERES) programme. They use a very simple radiative transfer approach, parametrising the forward model by an efficiently computable function that incorporates cloud optical depth and the emissivity dependence on effective radius, and minimise the sum-of-square error between the model and observations.

Pérez et al. (2000) also include surface temperature information, retrieved from clear sky pixels in the vicinity of the clouds, in their retrieval method to determine effective radius and cloud top temperature of nocturnal marine stratocumulus clouds. Their forward model consists of a vertically uniform, plane-parallel cloud over a sea surface. Emission and absorption effects of the atmosphere above the cloud are neglected. Following Baum et al. (1994), Pérez et al. (2000) employ brightness temperature differences (BTD) between the satellite channels to express the varying behaviour of the cloud radiative properties with wavelength, and extensively study the behaviour of the forward model when τ, r_{eff}, and temperature

[11] Indeed, although making the retrieval of particle size possible at nighttime, this dependence becomes a problem in the removal of the thermal contribution to the near infrared channels during daytime.

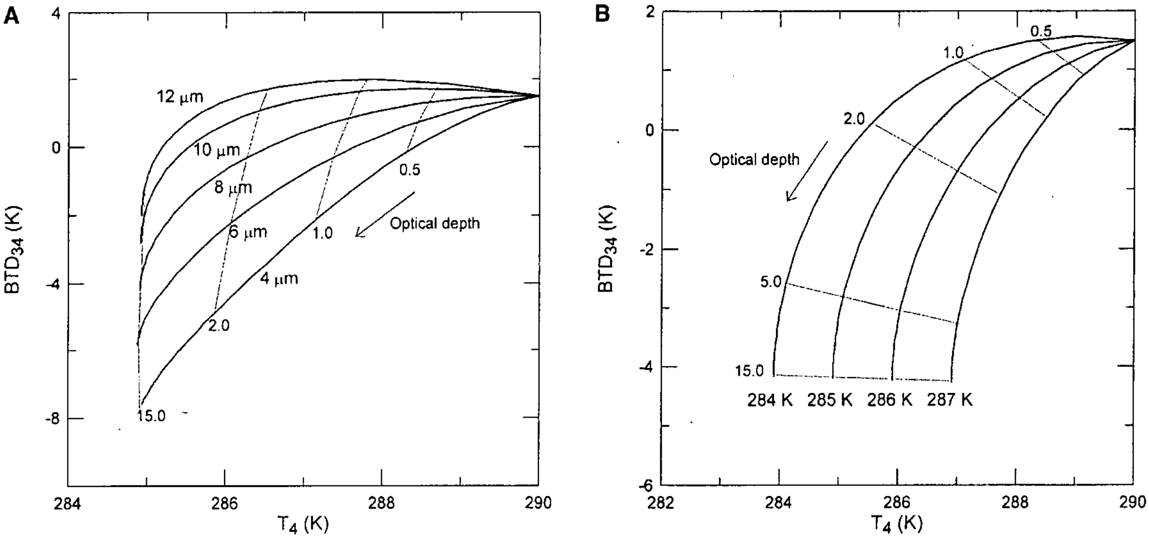


Figure 1.9: Physical basis for the remote sensing methods used by Heck et al. (1999), Pérez et al. (2000), Baum et al. (2003) and Cerdeña et al. (2007): plotted is the brightness temperature difference (BTD) between AVHRR channels 3 and 4 (3.7 and 11.0 μm), as measured at the top of the atmosphere, versus the brightness temperature for channel 4. The curves are based on radiative transfer computations for horizontally and vertically homogeneous clouds with varying particle radius and optical thickness (left) and temperature (right). By creating such curves with a radiative transfer model and comparing them to the brightness temperatures measured by the satellite, it is possible to infer information about the cloud properties that caused the satellite measurements (reprinted from Pérez et al., 2000, © 2000, with permission from Elsevier).

are varied.[12]

Figure 1.9 shows the BTD between the 3.7 μm and 11 μm channels of the AVHRR instrument (channels 3 and 4, respectively), henceforth abbreviated with BTD(3.7-11), versus the 11 μm brightness temperature (BT) for vertically uniform clouds with (a) a fixed cloud top temperature of 285 K and several effective radii and optical depths, and (b) a fixed radius of 8 μm and varying cloud temperature. The curves span an area of possible solutions of the forward model, and if some parameters can be fixed the remaining ones can readily be inferred from these diagrams. Similar diagrams exist for the BTD between 11 and 12 μm (BTD(11-12) in the following).

Yet two issues complicate the retrieval. One problem for nocturnal retrievals is the saturation of cloud emission with optical depth. For thin clouds a number of photons from the surface and all levels within the cloud layer are able to reach the satellite without absorption. For thicker clouds absorption will effectively remove all photons from lower levels in the cloud layer before they reach cloud top – leaving

[12] The intensities in the cost function (1.22) now become brightness temperatures and brightness temperature differences.

the cloud top radiance to be governed only by the upper cloud (of course, the absorption is wavelength dependent). Figure 1.8d illustrates the phenomenon. For an $\tilde{\omega}$ of 0.9, the emissivity of the cloud, equal to its absorptance, quickly reaches an asymptotic limit. A convergence for larger optical depths is also visible in the BTD diagrams of Figure 1.9. However, since most marine Sc have optical thicknesses of less that 10 (Xu et al., 2005; Rossow and Schiffer, 1999), the saturation problem does not pose a significant constraint.

Also problematic are ambiguities in the forward model. As noted by Pérez et al. (2000), some sets of measured brightness temperatures can be caused by several clouds having differing optical properties. This, of course, makes the unique retrieval of the cloud parameters impossible. Pérez et al. (2000) find that the effect is larger on BTD(3.7-11), but also occurs for BTD(11-12), especially for effective radii in the range of 4-7 μm.

The minimisation technique used by Pérez et al. (2000) is similar to the one used by Nakajima and King (1990) and King et al. (1997), who first retrieve optical depth from the visible wavelengths in order to transform the optimisation with respect to r_{eff} into a one-dimensional problem. Noting the complexity of the hypersurface of the cost function (with equivalently deep minima that resemble the ambiguities), Pérez et al. (2000) also split the retrieval into two parts. First, cloud top temperature is recovered from the BTD(11-12) signal. Fixing the atmospheric profile using the inferred surface and cloud top temperatures, they then run the forward model again for a range of effective radii. This produces a one-dimensional minimisation problem with respect to r_{eff}, which is easier to solve for multiple solutions.

The retrieval result then consists of a cloud top temperature, together with a list of possible effective radii. Pérez et al. (2000) compare the retrieval results to time-averaged in-situ measurements performed on the Canary Islands, and find that satellite and in-situ observed r_{eff} agree within 1.25 μm.

González et al. (2002) report on a modified version of Pérez et al. (2000). They eliminate the ambiguities due to effective radius by building a LUT in which, for situations where multiple solutions exist, they only store the median radius. The discarded solutions are used to compute a confidence interval around the central value, which they find to be as large as 7 μm for large r_{eff} in thin clouds. The cost function then possesses a unique global minimum, which they find with a genetic algorithm in order to avoid multiple local minima. In addition to cloud temperature and effective radius, they also retrieve optical depth values.

Pérez et al. (2002) adapt the retrieval to channels from the MODIS instrument. Instead of using BTs at 3.7, 11 and 12 μm, they employ the 3.7, 3.9, 8.5 and 11 μm channels (plus surface temperature).

Furthermore, atmospheric contributions to the top-of-atmosphere (TOA) BTs are included this time, assuming a known atmospheric profile from a different source. The error surface is minimised with a scatter search algorithm.

Baum et al. (2003) also extend the Baum et al. (1994) work to the MODIS instrument. Their objective is again to recognise situations in which high level cirrus overlays low boundary layer clouds. Including the 8.5 μm channel in their computations, they note that the BTD(8.5-11) (BTD between 8.5 and 11 μm) shows a strong dependence on cloud top temperature but not on particle radius.

Apart from the expensive minimisation techniques employed in the outlined nighttime retrieval methods, the procedures also suffer from the problem of computing a meaningful uncertainty estimate. Pérez et al. (2000), as well as González et al. (2002), perform a sensitivity analysis of their retrievals with respect to errors in the observed brightness temperatures. They find that a variation of ± 2 K in the observed BTs of cloudy pixels leads to variations in r_{eff} of more than 3 μm, variations of ± 0.5 K in the clear sky BTs lead to errors in r_{eff} of less than 0.5 um (Pérez et al., 2000). González et al. (2002) note that the sensitivities of retrieved cloud parameters to input BTs are largest in the case of thin clouds. They also discuss the effect of the ambiguities in effective radius and the error in the presence of broken clouds on the subpixel scale (neglecting nonlinear effects).

However, these sensitivity analyses cannot provide more than an idea of the general magnitude of the uncertainty. They also do not include the effects of assumptions made in the forward model or the accuracy of the numerical inversion. Indeed, Pincus et al. (1995), for the case of optical depth retrievals, report on the difficulty of obtaining a good uncertainty estimate. In order to obtain an accurate error estimate for an individual retrieval or for an average over a population of retrievals, additional inversions with perturbed input variables would have to be performed, which increases the computational cost.

1.5.3 Artificial Neural Networks

An interesting development to tackle the high computational cost of the inverse procedure is the use of artificial neural networks (ANNs). The theory of ANNs as a tool for statistical data modelling has been developed since the 1950s (e.g. Rosenblatt, 1958), but they were not used much in the atmospheric sciences until the 1990s (e.g. McCann, 1992; Hsieh and Tang, 1998). General references to ANNs are the books by Bishop (1995, 2006).

Inspired from the biological archetype of the human brain, an ANN imitates the concept of interconnected nodes, the *neurons*, that can "fire" in order to pass on information if certain input conditions are

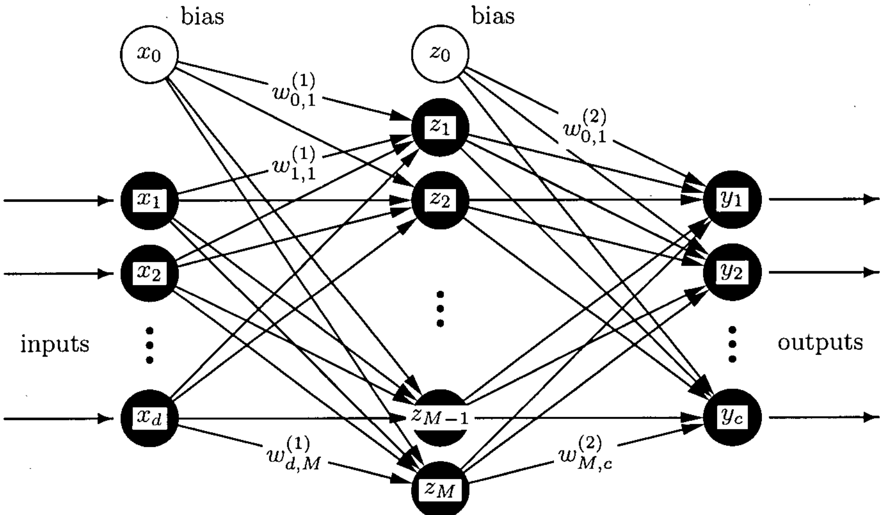


Figure 1.10: Schematic diagram of a feed-forward network with two layers of adaptive weights $w^{(1)}_{i,j}$ and $w^{(2)}_{j,k}$. The input neurons on the left are labelled with x_i. The information propagates in a forward direction through the hidden neurons z_j to the output neurons y_k. In addition to the input and hidden neurons, there are two bias parameters with a fixed input (activation) of $x_0 = 1$ and $z_0 = 1$, respectively.

met. In particular, a neuron in an ANN computes the sum over all of its input values and returns the value of an activation function of that sum. This activation function can be an arbitrary function, but usually a sigmoidal function is used that returns 1 if a certain threshold is reached and 0 otherwise. The connections between the individual neurons are weighted, and by adjusting these weights, the ANN can "learn" a certain behaviour.

Artificial neural networks have been widely used for both classification and regression problems (Bishop, 1995). There are many different types of ANNs, but the important one for this study and the works cited here is the multilayer perceptron (MLP).[13] Its architecture is schematically illustrated in Figure 1.10. Its neurons are grouped into several layers, one input layer, one output layer, and one to several hidden layers.[14] Each neuron in the input layer corresponds to an input variable, and each neuron in the output layer to an output. The number of hidden layers and neurons in them is variable and has to be determined individually for each case.

In order to learn a behaviour, the network has to be trained with a training dataset that includes input values as well as their corresponding target values. The weights are then modified until, for all

[13] I will not give a more detailed description of multilayer perceptrons in this thesis, a thorough introduction to many aspects concerning neural networks can be found in Bishop (1995), a shorter review in Bishop (1994).

[14] The numbering of the layers can be confusing. A "two-layer-perceptron" refers to a network with two layers of adaptive weights, that is only one layer of hidden neurons. "Two hidden layers", however, refer to two layers of hidden neurons and hence three layers of adaptive weights.

input values, the outputs of the ANN equal the target values to a sufficient accuracy.

An important property of such ANNs is that, given a large enough number of hidden neurons and weights, they are able to approximate any arbitrary smooth function (e.g. Bishop, 1995). This makes them ideally suited for nonlinear regression problems in which little is known about the shape of the cost function. Furthermore, once the training process is completed, the computation of the output values includes only a few summations and evaluations of the activation functions, which makes a trained ANN very fast. Of course, there are also disadvantages, as it can be difficult to determine the number of hidden neurons or to find a good set of weights (Bishop, 1995, more specific problems arising in the atmospheric sciences are outlined by Hsieh and Tang (1998)).

In the context of atmospheric satellite remote sensing, some work has been done that employs ANNs. For instance, Krasnopolsky et al. (1995) and Krasnopolsky et al. (2000) used ANNs to retrieve surface wind speeds over the ocean from microwave measurements. Aires et al. (2001) determined surface temperature, water vapour content and cloud liquid water path, also from microwave observations, and Aires et al. (2002) retrieved atmospheric and surface temperature from infrared measurements.

Faure et al. (2001c) applied ANNs to daytime retrievals of cloud parameters. In their work, they investigated the feasibility of simultaneously retrieving cloud optical thickness, effective radius, a relative cloud inhomogeneity and fractional cloud cover of inhomogeneous clouds from observations at 0.6, 1.6, 2.1 and 3.7 μm. They used a three dimensional Monte Carlo radiative transfer model to account for nonlinear effects due to the cloud inhomogeneities to build a database of 3000 clouds. An ANN with two hidden layers, 10 neurons in each layer, was trained from this database in order to model the inverse function. Faure et al. (2001c) concluded that a retrieval of inhomogeneous clouds using an ANN is feasible.

The study was extended by Cornet et al. (2004) to combine daytime observations available at different resolutions. In order to account for the increased complexity of the problem, they employed a combination of several ANNs and applied the method to MODIS data (Cornet et al., 2005). However, they did not compare their results to in-situ observations.

Schüller et al. (2003, 2005) also used ANNs for daytime retrievals of droplet number concentration, geometrical thickness and liquid water path of shallow convective clouds, but they did not give many details on the architecture they used.

Motivated by the high computational efficiency of an ANN-approximated inverse function, Cerdeña et al. (2004) continued the work of Pérez et al. (2000) and González et al. (2002), becoming the first to apply ANNs to nocturnal cloud property retrievals. In their study, they empirically explored several network architectures with differing numbers of hidden layers and neurons, and found that two layers with 100 neurons in the first and 20 neurons in the second layer yielded the best results. The ANNs were trained using a database of 20,000 data points, with an additional 10,000 independent points for validation. Cerdeña et al. (2004) compare two forwards models, one employing the vertically uniform cloud model, and the other an adiabatic profile with vertically increasing liquid water content (more details in Chapter 2). Analysing the same data as Pérez et al. (2000) and González et al. (2002), the ANN retrieved effective radii agreed within 2 μm with the in-situ observed values, with the adiabatic cloud model yielding slightly improved results.

Cerdeña et al. (2007) extended the retrieval method to the daytime case and also further investigated the network architecture. Utilising genetic algorithms, they were able to improve the network design to a network containing 20 neurons in the first and 5 neurons in the second hidden layer, yielding similar results as their first, more expensive architecture. They also presented a sensitivity analysis, similar to the one conducted by Pérez et al. (2000), by perturbing the input brightness temperatures by 0.5 K and noting the effect on the retrieved parameters for thin ($\tau < 2$) as well as thick ($\tau > 8$) clouds. For thin clouds, such perturbations can lead to errors of up to over 4 um in r_{eff}, 4 K in cloud temperature (T), and 0.6 in τ. For thick clouds, errors are smaller in r_{eff} and T, but larger for τ.

However, in the Cerdeña et al. (2007) work, the problem of estimating an accurate error interval for given inputs remains. Furthermore, they do not discuss the effect of an uncertainty in the neural network fit on the outputs (i.e. the uncertainty of having found the best set of weights during the training process), which can also be significant (Aires et al., 2004a).

Indeed, there exist both analytical methods to compute the Jacobian (i.e. the sensitivity) of a network for a given input and to compute the output uncertainty due to the network fit. MacKay (1992a,b, 1995) developed a Bayesian framework for neural network training that besides of making the training process more stable allows for an estimation of the uncertainties in the network predictions that are due to the network fit and to noise in the training data. Aires (2004); Aires et al. (2004a,b) generalised this concept to the multidimensional case and demonstrated its application in the context of their previous microwave remote sensing problem (Aires et al., 2001).

The Jacobian is very important for the validation of the network fit. Since for the inverse problem, we do not know the shape of the function to model, we can only verify the ANN by comparing its predictions to independent test data or by controlling the physical consistency of the Jacobian. This means that the output variables should be dependent on the expected input variables. For instance, during nighttime, we expect the effective radius output to be sensitive mainly to the BTD(3.7-11) signal (cf. Figure 1.9), while cloud top temperature should depend mainly on the thermal signals at 11 or 12 um. This can even be done in a quantitative way by numerically estimating sensitivities at certain points in the LUT and comparing them to the ANN predicted values.

However, inverse problems are often ill-conditioned. That means that there exist several differing functional mappings that approximate the training data equally well, but not necessarily represent the true function that generated the data – the training data are simply not precise enough to unambiguously specify the correct function. In the case a bad mapping is afterwards applied to input data that was not part of the training dataset, the prediction may be far from the actual value. The network is then said to generalise badly (Bishop, 1995). While this problem is difficult to avoid, additional care should be exercised when using ANNs for ill-conditioned problems. Aires et al. (2004b) describe an approach to estimate the variability of the Jacobian due to the uncertainty in the network fit. This variability is a good indicator of how unstable the training process was and thus how ill-conditioned the problem is.

1.6 Objectives

A neural network can potentially provide a fast retrieval algorithm for determining nocturnal cloud properties that might be able to overcome the performance problem encountered in classical retrieval approaches. Furthermore, the application of the methods developed by MacKay and Aires et al. could be very beneficial for understanding the "black-box nature" of ANNs and in order to improve the retrieval quality. I will extend the promising Cerdeña et al. (2007) results to a different network architecture, a different satellite sensor and different in-situ data. My focus lies on the uncertainty in the retrieval, and what we can learn from its sensitivities.

In particular, the objectives for this thesis can be outlined as follows. The goal is to set up a retrieval method capable of determining cloud effective radius, cloud top temperature and cloud optical thickness from nocturnal measurements of the MODIS instrument. In-situ data for evaluation purposes are available from the DYCOMS-II field campaign, and the method should be able to provide error bars on the results. The individual steps include:

- The Aires (2004); Aires et al. (2004a,b) method will be implemented and it will be investigated if it is suitable for the given problem. Apart from obtaining an uncertainty estimate of the network fit, the hope is that ambiguities in the training database will cause a larger uncertainty in the retrieval.

- A forward model will be developed that is capable of computing top-of-atmosphere brightness temperatures for the relevant near and thermal infrared MODIS channels for varying cloud parameters. A LUT (LUT will be used synonymously for database from here on) has to be computed using this model.

- The implemented methods will be applied to the retrieval problem. ANNs have to be trained from the computed LUT, different network architectures have to be explored and the results have to be evaluated. For this thesis, the goal is to retrieve one scene from the DYCOMS-II field campaign (so that several parameters including the atmospheric profile can be prescribed in order to simplify the problem) and to evaluate the results using the in-situ data.

- Finally, the uncertainties and sensitivities of the retrieval will be analysed and their usefulness investigated.

This thesis is meant to build a basis for further research. If the retrieval of the test case proves successful, future work can be performed on evaluating further scenes, with an eventual goal of creating a general method that could be used "operationally“ for any arbitrary scene.

The remainder of this document is structured as follows. In Chapter 2, the radiative transfer model will be described. The employed cloud model will be discussed, as well as radiative transfer techniques for computing the TOA brightness temperatures. A short section will be devoted to the effects of the overlying atmosphere. In Chapter 3 I discuss neural network techniques. The Aires method will be derived, and its implementation will be applied to simple test cases. The behaviour of the method for these cases will be analysed and its applicability to the remote sensing problem discussed. The training of ANNs from the LUT and the retrieval of the test scene is the topic of Chapter 4. Issues concerning the retrieval setup will be discussed, and the results of sensitivity and uncertainty estimates are presented. A discussion of the usefulness of the estimated variables is given, and the thesis concludes with a summary of the work in Chapter 5.

Chapter 2

Radiative Transfer and Development of the Forward Model

In this chapter I will describe the design of the forward model that is used for the case study. The scene of July 11, 2001, was chosen for the retrieval, and the in-situ aircraft measurements used in this chapter are taken from DYCOMS-II research flight II (RF02), which took place on that day. The DYCOMS-II campaign will briefly be introduced in section 2.1. Next, I will introduce the radiative transfer equation (RTE) that mathematically describes the propagation of radiation through the atmosphere and discuss how it can be solved. For the radiative transfer calculations, cloud model and droplet size distribution are needed. Using the RF02 data, I will demonstrate the adequacy of the adiabatic approximation for the cloud model and show that the size distribution can be described by a modified gamma distribution (sections 2.2 and 2.3).

The question of which MODIS channels are best suited for the retrieval is discussed in section 2.4. The radiances measured by the sensor at these channels are always average radiances over a wavelength interval, since the instrument cannot measure at individual wavelengths. Hence, the forward model must be able to compute such interval-averaged intensities. I will introduce the correlated-k approximation as an efficient way to calculate gaseous absorption over these intervals.

The forward model also requires specification of absorption and emission by gaseous atmospheric constituents including water vapour and carbon dioxide above the cloud. This is discussed in section 2.5.

The radiative transfer package libRadtran (Mayer and Kylling, 2005) is employed to compute cloud top radiances. In section 2.6, the setup of the forward model including libRadtran will be described, and I conclude the chapter by demonstrating that the forward model is able to reproduce the BTD relationships of Baum et al. (1994) and Pérez et al. (2000).

2.1 DYCOMS-II Data

The DYCOMS-II field campaign took place from July 7, 2001, to July 28, 2001. During nine nights, research flights collected extensive in-situ datasets in the nocturnal Sc cloud layer over the east Pacific ocean off the coast of California, approximately 350-400 km west southwest of San Diego. The campaign and the available data are described by Stevens et al. (2003a,b). While the major objective of the campaign was to perform measurements to advance understanding of entrainment and drizzle processes, the collected data are also very useful for this remote sensing study.

During seven nights, circles with an approximate diameter of 60 km (30 min) were flown at several heights in the boundary layer (subcloud and cloud layer). Additionally, frequent vertical profiles were taken. Data useful for this study includes the measurements taken during vertical profiling and in horizontally advected circles flown just below cloud top and above cloud base. Besides the standard meteorological measurements of temperature, humidity, pressure and wind speed, data of liquid water content, droplet concentration and droplet sizes were taken (a complete list can be found in Stevens et al. (2003b)).

The ground speed of the airplane was about 100 ms^{-1} and measurements relevant for this work were taken every second. To ensure comparability with the MODIS data, I have averaged the DYCOMS-II data over 10 s intervals in order to yield measurements on the 1 km scale. The measured droplet size distributions allowed for the computation of effective radius and mean radius.

For this thesis, the flights of July 11 and July 13 (RF02 and RF03 in the DYCOMS-II literature) were chosen. Satellite images of both scenes contain a number of ship tracks that provide contrasting droplet sizes which can be used to check the physical consistency of the retrieval. While the July 11 case will serve as the retrieval example in Chapter 4, data from both nights are used in this chapter for evaluating the cloud model. The actual satellite images from the MODIS sensor will be introduced in Chapter 4.

2.2 Scattering and Droplet Spectra

2.2.1 Radiative Transfer Equation

Figure 2.1 illustrates the processes that influence radiation propagating along a line-of-sight as it passes through the atmosphere. As discussed in section 1.3, energy can be lost by absorption and scattering out of the direction of propagation, while emission and scattering into the beam represent sources of energy.

The change in intensity across an infinitesimal volume hence is the sum of a sink term describing

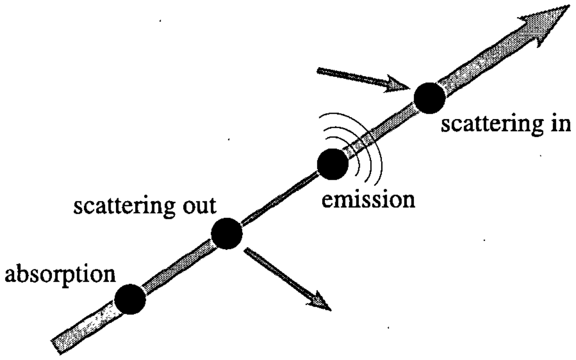


Figure 2.1: Processes that influence radiation as it passes through the atmosphere.

extinction by absorption and scattering, and source terms representing emission and scattering[15]:

$$dI = dI_{ext} + dI_{emit} + dI_{scat}. \tag{2.1}$$

The extinction term is described by Beer's Law (1.6):

$$dI_{ext} = -\beta_e I ds, \tag{2.2}$$

where ds represents an infinitesimal path length along the direction of propagation.

Emission is given by the emissivity of the medium times the Planck function (1.16):

$$dI_{emit} = \beta_a B(T) ds. \tag{2.3}$$

Here, Kirchhoff's Law (1.17) has been used to express the emissivity in term of the absorption coefficient. The scattering source term, however, is more complicated. The *scattering phase function* $p(\hat{\mathbf{\Omega}}', \hat{\mathbf{\Omega}})$ expresses the idea that radiation from any direction $\hat{\mathbf{\Omega}}'$ passing through an infinitesimal volume can contribute scattered radiation to our direction of interest $\hat{\mathbf{\Omega}}$. Furthermore, dI_{scat} must be proportional to the scattering coefficient β_s, which describes how much scattering occurs:

$$dI_{scat} = \frac{\beta_s}{4\pi} \int_{4\pi} p(\hat{\mathbf{\Omega}}', \hat{\mathbf{\Omega}}) I(\hat{\mathbf{\Omega}}') d\hat{\mathbf{\Omega}}' ds \tag{2.4}$$

The scattering phase function can be interpreted as a probability density. $p(\hat{\mathbf{\Omega}}', \hat{\mathbf{\Omega}})$ gives the probability that a photon from direction $\hat{\mathbf{\Omega}}'$ is scattered into direction $\hat{\mathbf{\Omega}}$. Hence, $p(\hat{\mathbf{\Omega}}', \hat{\mathbf{\Omega}}) I(\hat{\mathbf{\Omega}}')$ described the gain in

[15] Note that I am still omitting the subscript λ. All equations given here correspond to the monochromatic case.

intensity in direction $\hat{\mathbf{\Omega}}$ due to scattered radiation from direction $\hat{\mathbf{\Omega}}'$. In order to compute the total energy gain due to scattered radiation, contributions from all directions are summed in the integral, where the factor of 4π arises from the spherical integration to ensure that the integral over the phase function is one.

Putting the extinction and source terms into (2.1), the radiative transfer equation becomes

$$dI = -\beta_e I ds + \beta_a B ds + \frac{\beta_s}{4\pi} \int_{4\pi} p(\hat{\mathbf{\Omega}}', \hat{\mathbf{\Omega}}) I(\hat{\mathbf{\Omega}}') d\hat{\mathbf{\Omega}}' ds. \tag{2.5}$$

In this form, it has a general three-dimensional character, i.e. the direction vectors $\hat{\mathbf{\Omega}}$ can be expressed in any coordinate system, and the infinitesimal path length ds can be in any direction.

The most general approach to solve the radiative transfer problem numerically is the *Monte Carlo* method (e.g. Bohren and Clothiaux, 2006). It allows for arbitrary three-dimensional scenes by simulating the propagation of a large number of photons through the medium. The path of each photons is traced from its original source (e.g. the sun) until it leaves the defined scene (e.g. at the top of the atmosphere). While this method is very flexible and allows for inhomogeneities in the cloud layer, it is computationally very expensive and currently not suited to compute the TOA radiances for a large number of clouds, as needed for a LUT.

For the reasons discussed in Chapter 1, the plane parallel approximation is a good assumption for large parts of marine Sc on the scale of the MODIS pixels (1 km). In order to keep the forward problem as simple as possible and at the same time computationally tractable, I decided to construct the LUT for plane parallel clouds. In fact, most analytic solutions and approximations to the radiative transfer equation have been developed for the plane parallel case (Petty, 2006).

In a plane parallel atmosphere, all relevant parameters (i.e. β_a, β_s, T, r_{eff}; cf. section 1.3) are only dependent on height. Since the important aspect for radiation is how much absorbing, emitting and scattering atmosphere it must traverse, it is convenient to express the vertical coordinate in terms of the optical properties of the atmosphere rather than in geometrical units. The extinction *optical depth* is defined as the optical thickness of the atmosphere between the top of the atmosphere and a level at height z:[16]

$$\tau(z) = \int_z^{\infty} \beta_e(z') dz'. \tag{2.6}$$

At the top of the atmosphere, $\tau = 0$. Furthermore, directions are usually expressed in terms of zenith

[16] Some authors use *optical depth* synonymous for *optical thickness* when referring to the cloud optical thickness; in order to avoid confusion I will use *optical depth* for the vertical coordinate in the RTE and *optical thickness* for the cloud property.

angle θ (measured from directly overhead; $\theta = 0$ is overhead, and $\theta = \pi/2$ is the horizon), and azimuth angle ϕ (measured counterclockwise from a reference point on the horizon).

The infinitesimal path length ds in (2.5) can now be expressed as

$$ds = \frac{dz}{\mu}, \tag{2.7}$$

where $\mu = \cos\theta$. From (2.6) it follows that $d\tau = -\beta_e\, dz = -\beta_e\, \mu\, ds$. Dividing (2.5) by this new height increment, the radiative transfer equation becomes

$$\mu\frac{dI(\mu,\phi)}{d\tau} = I(\mu,\phi) - (1-\tilde{\omega})B - \frac{\tilde{\omega}}{4\pi}\int_0^{2\pi}\int_{-1}^{1} p(\mu',\phi';\mu,\phi)I(\mu',\phi')d\mu' d\phi'. \tag{2.8}$$

The problem of computing the TOA radiances "seen" by the satellite sensor now becomes the problem of solving this RTE.

A common simplification in order to solve the RTE is to divide the zenith angle into discrete intervals, so-called *streams*. The simplest of these methods is the two-stream method, which divides the intensity field into only two directions; upwelling and downwelling. Hence, it assumes that the intensity is approximately constant in each hemisphere. The multi-stream code DISORT (Stamnes et al., 1988, 2000) generalises this concept to an arbitrarily large number of discrete angles, so that it becomes possible to accurately compute radiances for pixels that are not directly underneath the satellite. The code is well documented and already part of the radiative transfer package libRadtran. It will be used for the forward model computations in this thesis.

2.2.2 Droplet Size Distribution and its Phase Function

A big simplification for liquid water clouds is that the water droplets can be treated as spherical droplets, which makes the computation of the phase function with Mie theory (Mie, 1908) possible. In contrast, the non-spherical character of ice particles makes the computation of the phase function for ice clouds much more difficult (Mayer and Kylling, 2005). Mie theory employs Maxwell's equations to derive a three-dimensional wave equation for electromagnetic radiation, which is solved for boundary conditions at the surface of a sphere (Petty, 2006).

For spherical particles, the phase function only depends on the angle Θ between the original direction $\hat{\Omega}'$ and the scattered direction $\hat{\Omega}$ of a photon. Since $\cos\Theta = \hat{\Omega}' \cdot \hat{\Omega}$, it is common to write the phase function as $p(\cos\Theta)$.

For remote sensing applications it is of interest to model the mean scattering effects of the entire cloud droplet size distribution. A common assumption is to model this distribution as a modified gamma distribution. It is given by (Miles et al., 2000)

$$n_{gam}(r) = \frac{N}{\Gamma(\nu_{gam})} \left(\frac{r}{r_N}\right)^{\nu_{gam}} \frac{1}{2r_N} \exp\left(\frac{-r}{r_N}\right), \tag{2.9}$$

where $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$ is the gamma function (e.g. Weisstein, 2007), ν_{gam} the *shape parameter*, and $r_N = r_{eff}/(\nu_{gam} + 2)$ a scaling parameter.

Location and width of the distribution are given by r_{eff} and ν_{gam}. The larger ν_{gam} becomes, the narrower the distribution will be. In many studies that employ Mie theory for computing the scattering properties of the cloud layer, the shape parameter is not discussed at all (for instance, in Pérez et al. (2000) or Cerdeña et al. (2007)). It is common, however, to assume ν_{gam} to be constant (Miles et al., 2000). For instance, the precalculated Mie tables for the AVHRR sensor, as available on the libRadtran homepage[17], use a value of 7.

Arduini et al. (2005) investigated the sensitivity of daytime retrieved continental cloud properties to droplet size distribution width. They found that the shape of the size distribution can have a significant effect on retrieved properties, therefore, the shape parameter must be representative of the cloud regime to be observed.

Miles et al. (2000) compared the droplet size distributions of both marine and continental stratocumulus clouds that were reported from different measurement campaigns in the literature. From more than 40 observations, they found a mean shape parameter of 8.6 for marine clouds, however, the observed values varied from less than 3 to over 40.

In order to avoid errors arising from an unrepresentative size distribution, I computed the best fit gamma distribution to the DYCOMS-II measured droplet size spectra. The shape parameter was determined from equations (3a) and (3b) in Miles et al. (2000):

$$\nu_{gam} = \frac{2r_{vol}}{r_{eff} - r_{vol}}. \tag{2.10}$$

Here, r_{eff} is the effective radius as computed from (1.11), and r_{vol} is the *volume mean radius*. It is

[17] http://www.libradtran.org/

computed from the measured spectra via (cf. section 1.3)

$$r_{vol} = \left(\frac{\displaystyle\int n(r) r^3 dr}{\displaystyle\int n(r) dr} \right)^{1/3} . \tag{2.11}$$

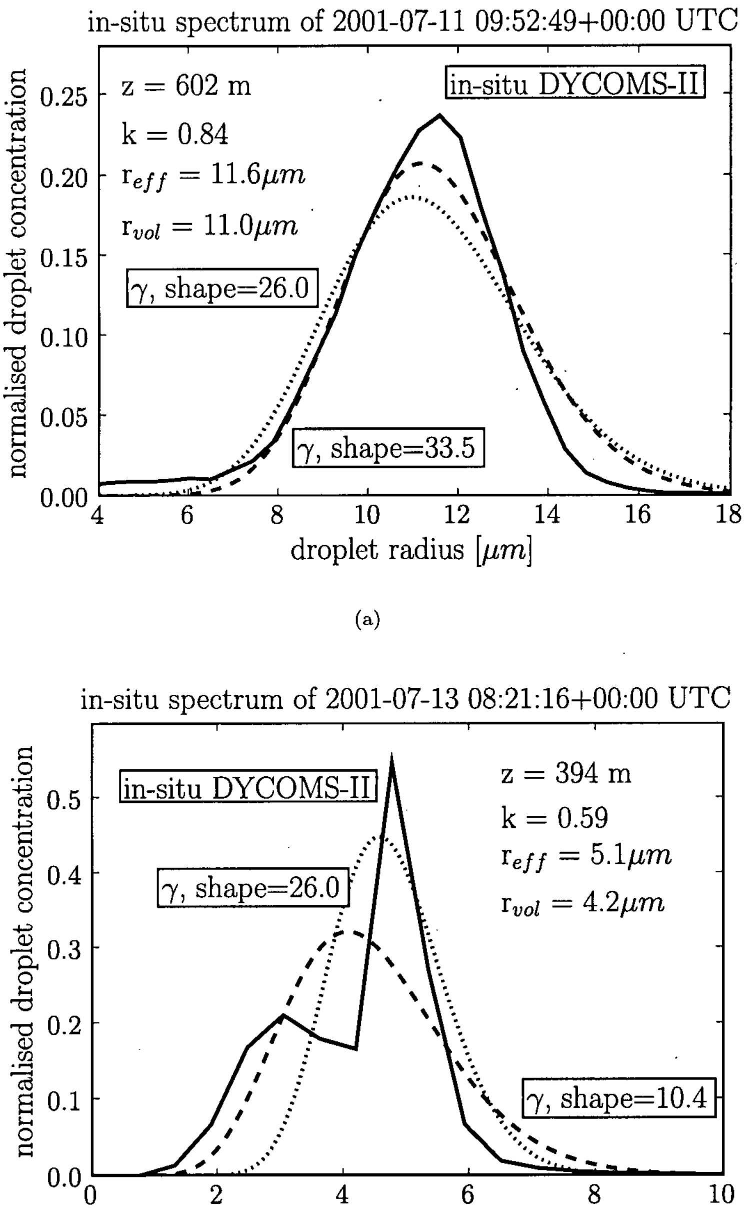
Figure 2.2a shows a size distribution measured at cloud top during the July 11 flight. The effective radius is 11.6 μm, and the best fit gamma distribution has a shape parameter of 33.5, significantly narrower than the average value of 8.6 found by Miles et al. (2000). Indeed, the average shape parameter encountered at this day was 26, the distribution of which is also plotted in the figure (see section 4.1 for more details).

Not all measured distributions were well represented by a modified gamma distribution. In fact, Miles et al. (2000) note that Sc droplet size distributions are frequently bimodal. An example of such a distribution, measured close to cloud base on July 13, is shown in Figure 2.2b. Here, the best fit distribution computed from (2.10) has a shape parameter of 10.4.

Since ν_{gam} is not a variable parameter in my retrieval algorithm, it has to be set to a fixed value as representative of the scene as possible. For the specific test case discussed in Chapter 4 (and also for the remaining DYCOMS-II cases), it is possible to analyse the in-situ data prior to building the LUT. Hence, a representative mean value can be found. For future applications of the method that require independence of the LUT of the scene to be retrieved, sensitivity tests should be conducted in order to investigate the impact of a fixed gamma distribution shape parameter on the retrieved values.

Figure 2.3a shows the phase functions of droplet size distributions with effective radii of 2, 11.5 and 25 μm and a shape parameter of 26 for radiation at $\lambda = 3.7$ μm. In Figure 2.3b, the phase function for the 11.5 μm distribution (as in Figure 2.2a) is reproduced in a polar plot (note the logarithmic scale in both figures). The functions were computed using Mie code included in K.F. Evans SHDOM (Spherical Harmonic Discrete Ordinate Method, another radiative transfer model) distribution (Evans, 1998).

A distinct feature worth mentioning is the strong forward scattering peak. It becomes more pronounced with increasing r_{eff}, which means that for this cloud a large amount of the scattered photons will be scattered into the forward direction (this is also true at visible wavelengths, the reason why sunlight penetrates even thick clouds). Also, more photons are scattered into the backwards direction than to the sides. The small peak at about $\cos\Theta = -0.8$ becomes more pronounced in the visible wavelength range, where it is seen as the rainbow.



(a)

(b)

Figure 2.2: In-situ measured spectra of cloud droplet sizes from the Jul 11th and Jul 13th flights, together with best-fit gamma distributions for (a) a unimodal distribution and (b) a bimodal distribution. Solid black lines mark the in-situ data, dashed lines are the best fit distributions. The average shape parameter on both days was approximately 26, this distribution is shown by the dotted line.

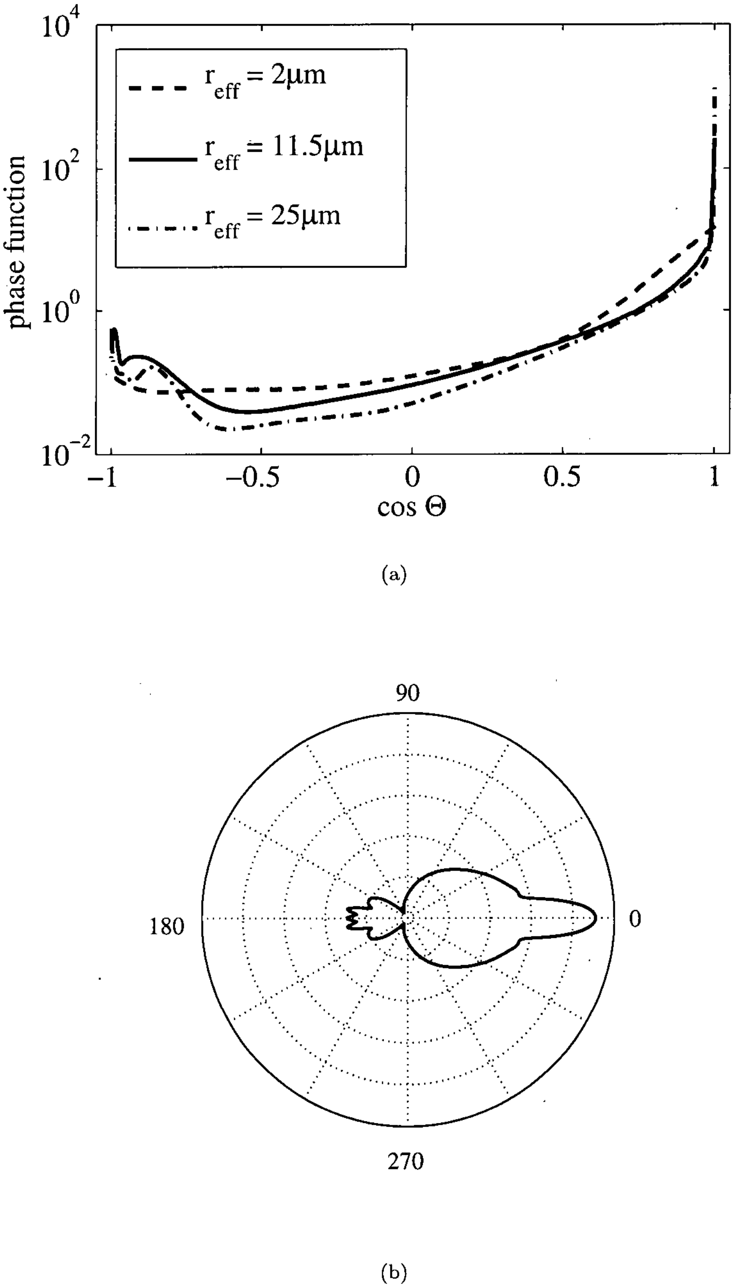


Figure 2.3: (a) Phase functions of cloud distributions following a gamma distribution with a shape parameter of 26. Shown are functions for effective radii of 2, 11.5 and 25 μm. Θ represents the scattering angle, $\cos(\Theta) = 1$ is forward scattering. (b) Polar view of the phase function for 11.5 μm, also with a logarithmic scale.

2.3 Adiabatic Cloud Model

In order to estimate the effect of the vertical stratification in the cloud layer on forward radiative transfer calculations, Brenguier et al. (2000) compared the vertically uniform (VU) cloud model (employed by Minnis et al. (1995); Heck et al. (1999); Pérez et al. (2000); González et al. (2002) or Pérez et al. (2002)) to an adiabatic stratified (AS) one. They found that the adiabatic stratified model in general is the more accurate choice. First, in-situ data analysed by Brenguier et al. (2000) of marine and continental Sc taken during the Second Aerosol Characterisation Experiment (ACE-2) showed a close-to-adiabatic stratification. Second, by comparing the radiative properties of clouds represented by a vertically uniform model and by the adiabatic stratified one, they found that in order to achieve radiative equivalence, the effective radius of the VU model had to be 80% - 100% of the cloud top r_{eff} of the AS model, the actual value depending strongly on cloud geometrical thickness and droplet concentration. These results mean that the effective radius retrieved by a scheme employing a VU model is not necessarily representative of the actual cloud top r_{eff}, since the VU model provides no way to retrieve cloud geometrical thickness or droplet concentration. By employing the AS model, on the other hand, it is possible to retrieve droplet concentration via the adiabatic relationships between the thermodynamic variables. This could be especially useful for monitoring of the aerosol indirect effect.

Indeed, the vertical profiles measured during the DYCOMS-II campaign closely resemble adiabatic profiles. Figure 2.4 shows in-situ data taken during RF02 (July 11), together with the vertical profile of an idealised adiabatic cloud.

When cloud droplet number concentration, humidity, temperature and pressure are prescribed at a reference level, it is straightforward to compute an adiabatic cloud profile using the thermodynamic equations. As thermodynamic variables I use the *moist static energy* h_m (e.g. Wallace and Hobbs, 2006) and the total specific humidity q_T (i.e. water vapour plus liquid water). In this thesis, "adiabatic profile" refers to a profile in which h_m and q_T are constant. If c_p denotes the specific heat of dry air at constant pressure (J K^{-1}), g the earth's gravitational acceleration (m s^{-2}), L_v the latent heat of evaporation (J kg^{-1}) and q_v the specific humidity (g kg^{-1}) (Curry and Webster, 1999; Wallace and Hobbs, 2006), h_m is given by

$$h_m = c_p T + gz + L_v q_v. \tag{2.12}$$

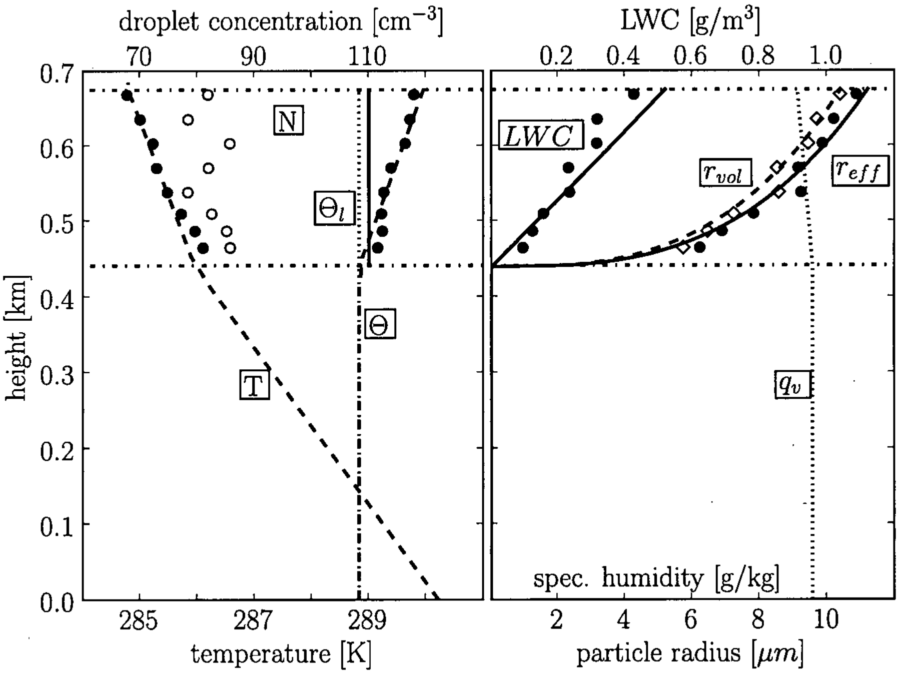


Figure 2.4: Vertical profile of cloud properties from the idealised adiabatic cloud model, together with in-situ measurements from the July 11 DYCOMS-II flight. The adiabatic cloud has a cloud top effective radius of 10.7 μm, a cloud top temperature of 284.8 K and a droplet concentration of N = 110 cm^{-3}. Volume mean radius is converted to effective radius with k = 0.8. Note that in order to fit the observed liquid water content the droplet concentration is overestimated in the adiabatic model (compare to Figure 2.6).

Following the findings of Miles et al. (2000), the total droplet concentration (cm^{-3})

$$N = \int_0^\infty n(r)dr \tag{2.13}$$

is also assumed to be constant in my model.

The above equations result in linear temperature profiles following

$$T(z) = T_{sfc} - \Gamma\, z, \tag{2.14}$$

where Γ represents the dry adiabatic lapse rate $\Gamma = \Gamma_d \approx 10$ K km^{-1} below cloud base and the saturated adiabatic lapse rate $\Gamma = \Gamma_s$ within the cloud. Γ_s varies with temperature and pressure, for marine Sc a value of $\Gamma_s \approx 6$ K km^{-1} is representative (Curry and Webster, 1999).

Below cloud base, no condensation occurs, so that the specific humidity remains constant with height:

$$q_v(z) = \text{const.} \tag{2.15}$$

Within the cloud, q_v follows the saturation specific humidity q_s:

$$q_v(z) = q_s(T, p), \tag{2.16}$$

where q_s is dependent on both temperature and pressure and can be obtained from the Clausius-Clapeyron equation (see Curry and Webster, 1999, for further details).

The liquid water content can be determined by subtracting the saturation water vapour content (the saturation specific humidity expressed in g m^{-3}) from the total water content (given by the specific humidity below cloud base). q_l also follows an almost linear profile within the cloud:

$$q_l(h) = C_w\, h, \tag{2.17}$$

where h denotes the height above cloud base, and the moist adiabatic condensate coefficient C_w varies from about 1 to 2.5×10^{-3} g m^{-3} m^{-1}, depending slightly on temperature (Brenguier et al., 2000).

In my model, cloud top and surface pressure, cloud top temperature, cloud top liquid water content and droplet concentration are prescribed. From the latter two, the volume mean radius r_{vol} is computed, given the relationship

$$q_l = \frac{4}{3}\pi \rho_l r_{vol}^3 N. \tag{2.18}$$

In order to compute the effective radius from the volume mean radius or vice versa, an assumption about the droplet size distribution has to be made. If the distribution is modelled as a modified gamma distribution as discussed in the previous section, (2.10) can be used for the conversion. More common in the literature, however, is to employ the k-value parameterisation suggested by Martin et al. (1994).

Martin et al. (1994) analysed in-situ measurements from marine and continental Sc of several field campaigns. In their data, the clouds were relatively homogeneous and entrainment played a minor role. For these cases, they proposed to parameterise the effective radius in terms of the volume mean radius and a dimensionless constant k following

$$r_{eff} = k^{-1/3} r_{vol}. \tag{2.19}$$

In maritime airmasses, Martin et al. (1994) found an average value of $k = 0.80 \pm 0.07$ (one standard deviation).

Measured values of Pawlowska and Brenguier (2000) during ACE-2 confirm the estimates made by Martin et al. (1994), however, they also show a strong variability of k with height, with values at cloud base ranging from as low as 0.4 to up to 1, while values at cloud top where more constant around 0.8 to 1 (Figure 2.5).

Combining (2.10) and (2.19), the relationship between k-value and ν_{gam} becomes

$$\nu_{gam} = 2\left(k^{-1/3} - 1\right)^{-1}. \tag{2.20}$$

A distinct problem in Figure 2.4 is that the droplet concentration is overestimated by about 30 cm^{-3} in order to fit the observed liquid water path and droplet sizes. Also, the slope of the liquid water content line seems slightly overestimated. This problem is due to subadiabatic regions in the cloud. For instance, as discussed in section 1.2, entrainment of dry air from above the inversion often causes subadiabaticity. The cloud column can hence be adiabatic in the lower part and subadiabatic in its upper part.

Brenguier et al. (2000) emphasise that the AS model still is an idealised representation of the actual cloud. They note that the adiabaticity should be taken as a maximum reference for the actual cloud microphysics at all levels. However, they also point out that the variety of possible profiles is too large for a simple parameterisation, and hence suggest the AS model as a simple and relatively accurate

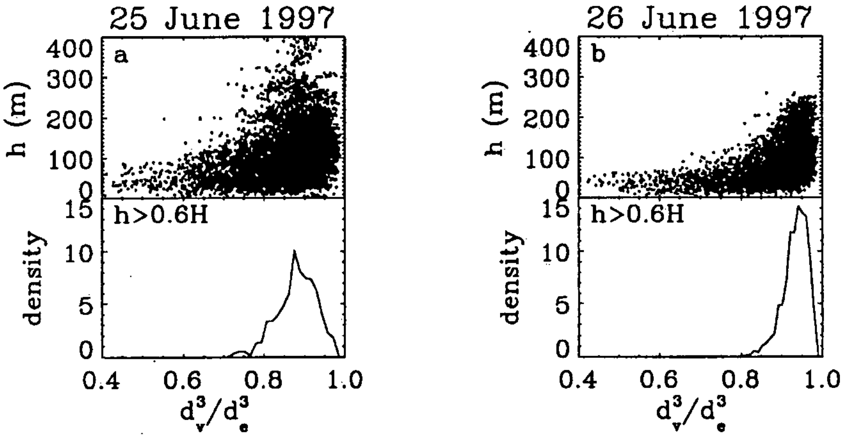


Figure 2.5: Measurements of k value versus height in marine stratocumulus clouds. The k value varies widely from values as low as 0.4 to almost 1, with a larger variability at cloud base (reprinted from Pawlowska and Brenguier, 2000, © 2000, with permission from Blackwell Publishing).

compromise solution.

For comparison, Figure 2.6 shows the same data as Figure 2.4, but this time with a subadiabatic profile using 75% of the adiabatic values. This time, both droplet concentration and liquid water content are well-fit.

In their investigation of radiative equivalence between the VU and AS models, Brenguier et al. (2000) found that a major difference between the two models is the dependence of optical thickness on cloud geometrical thickness. Equation (1.9) introduced the optical thickness as the integrated extinction coefficient between cloud base z_{bot} and cloud top z_{top}:

$$\tau = \int_{z_{bot}}^{z_{top}} \beta_e dz. \tag{2.21}$$

If the extinction cross section $\sigma_e = \sigma_a + \sigma_s$ (cf. 1.12 and 1.13) is written in terms of the *extinction efficiency* $Q_{ext} = \sigma_e/(2\pi r^2)$, the extinction coefficient of a droplet size distribution is given by

$$\beta_e = \int n(r) \left[Q_{ext}(r)\pi r^2\right] dr. \tag{2.22}$$

The extinction efficiency can be interpreted as how effective a particle can attenuate radiation with respect to its size, and is a complex function of the size parameter x, and hence particle size and wavelength.

In order to compute the optical thickness in the AS model, it is necessary to express (2.21) in terms

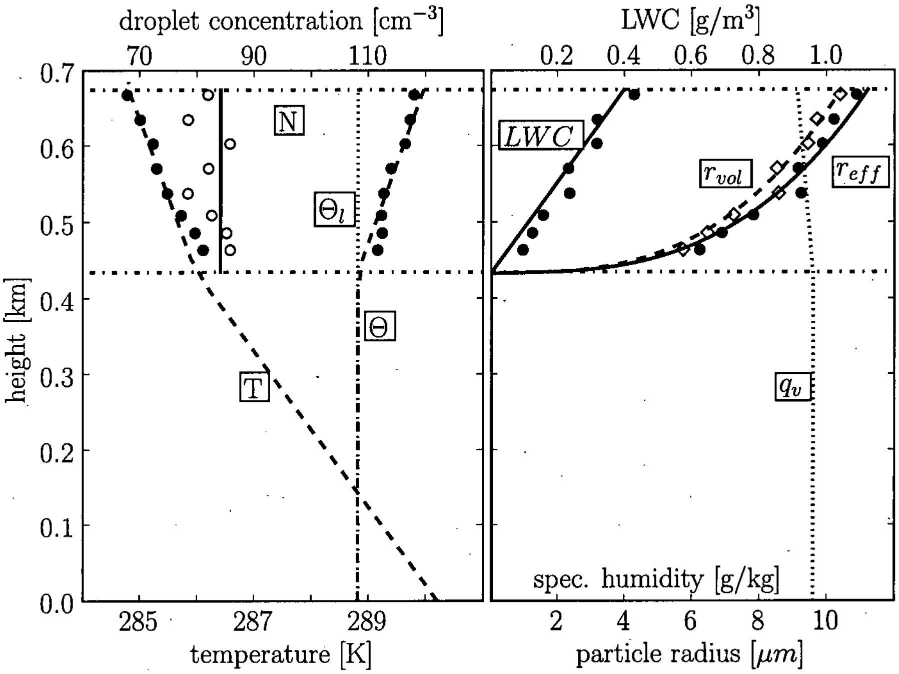


Figure 2.6: The same as Figure 2.4, but with a subadiabatic liquid water content of 75% of the adiabatic value.

of the available thermodynamic variables. By combining (2.11), (2.13) and (2.18), we can write the liquid water content as a function of the particle size distribution:

$$q_l = \int n(r) \left[\rho_l \frac{4\pi}{3} r^3 \right] dr. \tag{2.23}$$

This enables us to express τ in terms of the droplet radius and q_l:

$$\tau = \int_{z_{bot}}^{z_{top}} \frac{\beta_e}{q_l(z)} q_l(z) dz = \int_{z_{bot}}^{z_{top}} \left(\frac{\int n(r) \left[Q_{ext}(r) \pi r(z)^2 \right] dr}{\int n(r) \left[\rho_l \frac{4\pi}{3} r(z)^3 \right] dr} \right) q_l(z) dz. \tag{2.24}$$

For cloud droplet sizes in the visible wavelength range, the extinction efficiency Q_{ext} is approximately constant at a value of two (e.g. Petty, 2006, Figures 12.4 - 12.6), so that it can be pulled out of (2.24). The integrals can then be written as the effective radius:

$$\tau \approx \frac{3Q_{ext}}{4\rho_l} \int_{z_{bot}}^{z_{top}} \frac{q_l(z)}{r_{eff}(z)} dz \tag{2.25}$$

(e.g. Nakajima et al., 1991), which reduces to (1.14) for $Q_{ext} = 2$ and constant r_{eff}. For a VU model, the integral over height can be approximated by a multiplication of the integrand by cloud geometrical thickness ΔH, whereas in the AS case it has to be explicitly computed.

In the infrared wavelength range, however, Q_{ext} and thus the optical thickness of the cloud become a function of both cloud geometrical thickness and particle size distribution (Petty, 2006). Thus, there no longer is a simple relationship between τ, LWP and r_{eff}. For simplicity, I will use the visible optical thickness in my forward model. This has the advantage that the optical thickness retrieved by my algorithm can be compared with the preceding and subsequent operational daytime MODIS retrievals.

2.4 Choice of Channels and Correlated-k

The MODIS instrument offers a total of 36 channels in the visible and infrared spectrum (Barnes et al., 1998), 16 of which cover the infrared spectrum from 3.7 to 14.1 μm. Not all of these channels are suited for remote sensing of cloud properties. As discussed in Chapter 1, the radiative properties of the cloud droplets in the infrared strongly depend on particle size. Hence, the employed channels should maximise these differences. However, it is also desirable for the satellite sensor to measure photons arriving directly from the cloud. This means that the impact of the atmosphere above the cloud on the radiation arriving at the satellite should be as small as possible.

Figure 2.7 shows how the transmittance of a typical midlatitude summer atmosphere varies with wavelength. The bottom panel displays the total transmittance, while in the panels above, the effects of individual atmospheric components are shown. The four channels at approximately 3.7 (MODIS number 20), 8.5 (29), 11 (31) and 12 μm (32), as used in previous studies (cf. Chapter 1), are highlighted. They are located at wavelengths at which the total transmittance is large. Other MODIS channels are located, for instance, at approximately 4.5 or 9.7 μm. While these channels provide data for other applications, the atmosphere at these wavelengths is too opaque for remote sensing of cloud properties. Hence, channels 20, 29, 31 and 32 are indeed best suited for the remote sensing of cloud properties at night.

The second panel from the bottom shows that most of the absorption that occurs above cloud top is due to water vapour. It should thus be considered in the retrieval method. I will discuss in section 2.5 how the effect of water vapour can be accounted for.

The forward model must account for the band width of the chosen channels in the radiative transfer model. This requirement arises from the fact that the MODIS channels are not monochromatic, but cover finite wavelength intervals. Hence, the "monochromatic" radiances measured by the instrument are

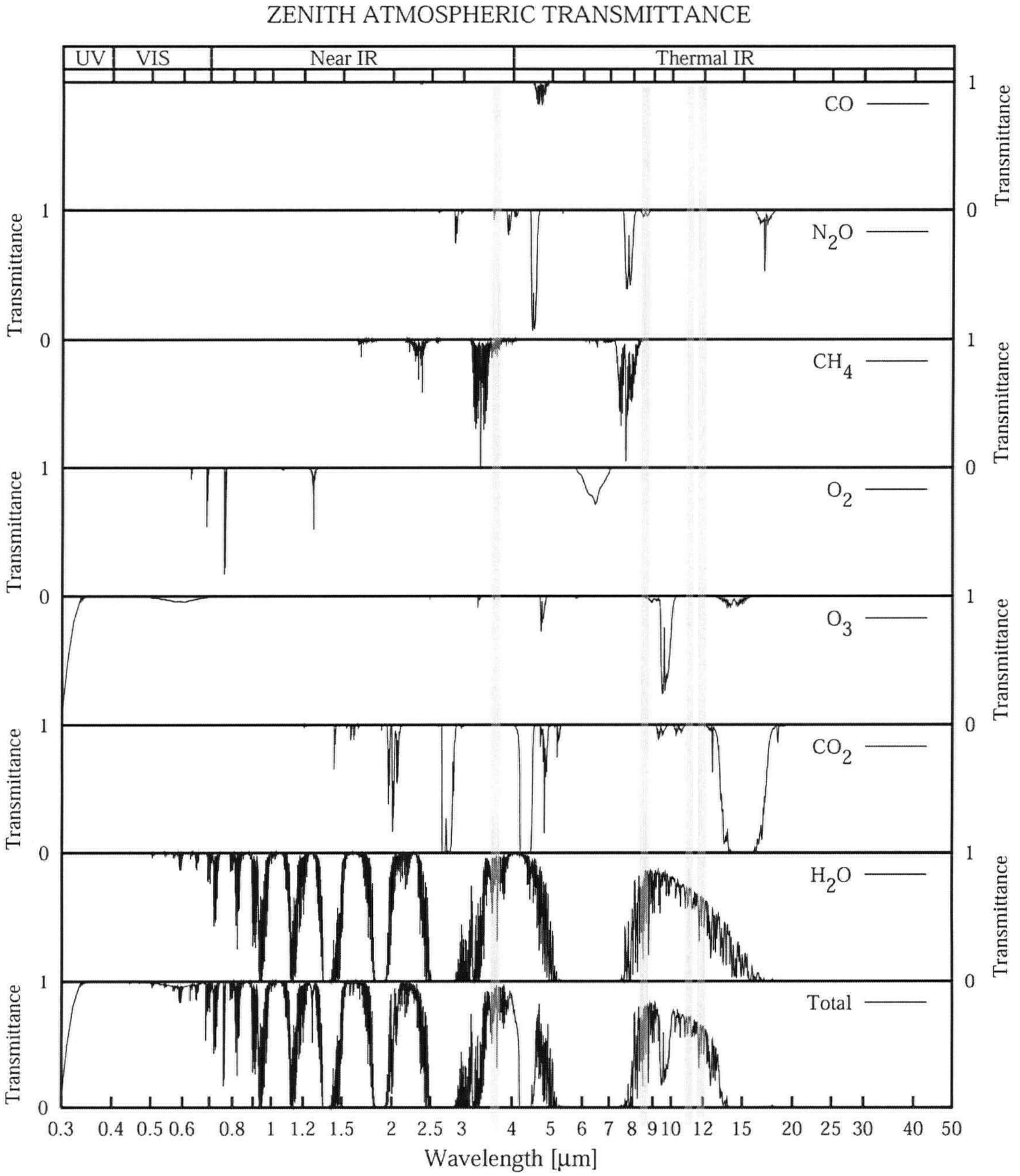


Figure 2.7: Dependence of atmospheric transmittance on wavelength for a typical midlatitude summer atmosphere. Highlighted in grey are the four MODIS channels relevant in this study (reprinted from Petty, 2006, © 2006, with permission from Sundog Publishing).

spectral-mean radiances.

In order to compute spectral-mean radiances with the radiative transfer model, the wavelength dependent absorption of the different molecular species listed in Figure 2.7 has to be taken into consideration. For an extact solution, an integration over many individually computed monochromatic radiances has to be performed, which can be computationally expensive. The *correlated-k procedure* (e.g. Fu and Liou, 1992; Kratz, 1995) provides a way to reduce the computational cost by several orders of magnitude while maintaining a high accuracy and is used in this thesis.

The concept of correlated-k can be explained by considering the simplified problem of computing the spectral-mean transmission of an atmospheric layer, due to only molecular absorption.[18] When the absorption coefficient of an atmospheric constituent is plotted against wavelength, the absorption spectrum in general is very complex. For example, Figure 2.8a shows a spectrum of CO_2 at a pressure of 507 hPa. The distinct peaks in the spectrum are called absorption lines, they can be explained by quantum theory (e.g. Petty, 2006) and their height and width are dependent on both temperature and pressure. In the spectral interval of a satellite channel, there can be thousands of such lines, especially if the effects of different molecular species are overlapping.

First consider a single absorbing species. Let the spectral interval width of a channel be $\Delta\lambda = \lambda_2 - \lambda_1$, and the *mass path* of the absorbing constituent be denoted by u. The mass path is the total mass of the species in a column, measured in kg m^{-2}. It is then convenient to introduce the mass absorption coefficient[19]

$$k_\lambda = \frac{\beta_{\lambda,a}}{\rho}, \tag{2.26}$$

where ρ is the density of the constituent. The spectral-mean transmittance follows from the integration of Beer's Law over wavelength:

$$t_{\Delta\lambda}(u, p, T) = \frac{1}{\Delta\lambda} \int_{\lambda_1}^{\lambda_2} \exp\left[-k_\lambda(p, T)\, u\right] d\lambda. \tag{2.27}$$

Here, p denotes pressure and T temperature.

The fundamental idea of correlated-k stems from the fact that in order to approximate the integral in (2.27) numerically, it is not important in which order the individual lines are arranged in the spectrum, as long as the area underneath the curve stays the same. Figure 2.8 illustrates this idea. If the complex

[18] The layer concept is relevant in as far as that DISORT and other RTE solvers divide the atmosphere into discrete layers.

[19] Do not confuse the mass absorption coefficient "k" used in correlated-k with the k-value used in the conversion of r_{vol} to r_{eff}.

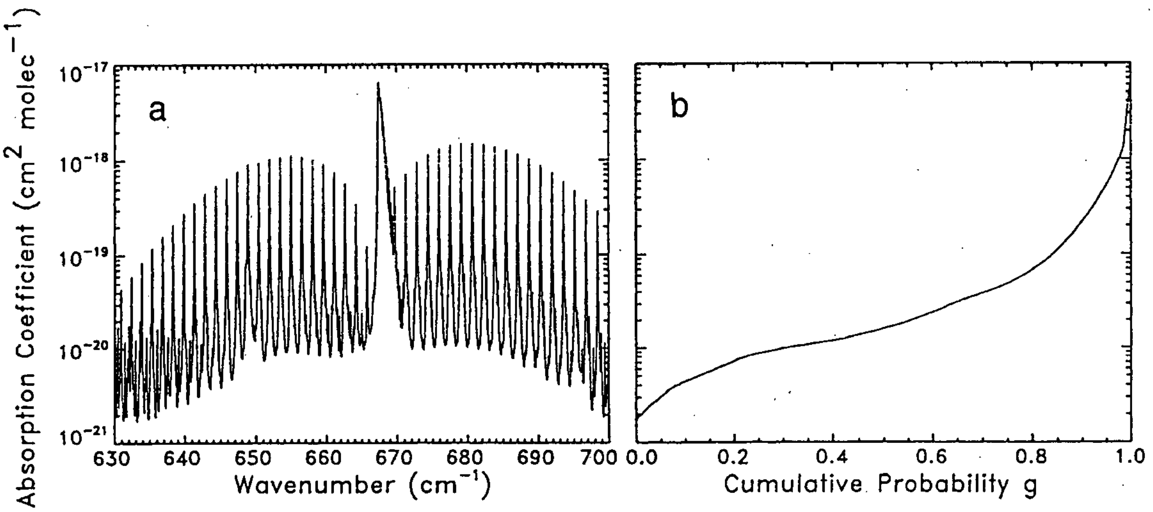


Figure 2.8: Illustration of the fundamental idea of correlated-k. (a) Absorption spectrum (k-coefficients) of CO_2 at a pressure of 507 hPa. (b) Inverse cumulative probability function $k(g)$, that represents the reordered k-coefficients from (a). For a numerical integration, the function in (b) requires much less quadrature points than the function in (a) (reprinted from Mlawer et al., 1997, © 1997, with permission from the American Geophysical Union).

spectrum of the absorption coefficient in Figure 2.8a is integrated numerically, a large number of quadrature points is needed in order to obtain a reasonable accuracy.[20] However, by reordering the absorption lines in ascending order (Figure 2.8b), we get a function that is much easier to integrate, yet covers the same area.

The reordering of the lines can be done by computing the inverse cumulative probability distribution $k(g)$ of the absorption coefficients. The cumulative probability function $g(k) = \int_0^k p(k')dk'$ is the integral of the probabilities $p(k')$ for all mass absorption coefficients k' between 0 and k, so that its inverse $k(g)$ gives the upper bound of the $(g \times 100)\%$ smallest ks. For instance, 60% of all k-values in the spectral interval of the channel are smaller than $k(g = 0.6)$. Equation (2.27) can then be written as

$$t_{\Delta\lambda}(u, p, T) = \int_0^1 \exp\left[-k_g(p, T)\, u\right] dg. \tag{2.28}$$

For the numerical integration, the integral is approximated by a finite sum:

$$t_{\Delta\lambda}(u, p, T) \approx \sum_{i=1}^{N} w_i \exp\left[-k_{g_i}(p, T)\, u\right], \tag{2.29}$$

with quadrature weights w_i and quadrature points g_i. For a function shaped as in Figure 2.8b, much

[20] Calculations of all individual lines are called *line-by-line* calculations.

less quadrature points are needed than for one shaped as in 2.8a, and it is hence computationally more efficient.

The method of reordering the absorption coefficients is called the *k-distribution* method. Since the ks are simply reordered and no approximation is made, it is exact. However, in a typical atmospheric layer pressure and temperature will vary with height, and since the ks are dependent on p and T (2.28) is only exact for vertically homogeneous layers (homogeneous mass paths). Nevertheless, the same equation is used to approximate the transmissivity of inhomogeneous mass paths. The assumption that is made in the correlated-k method is that for any p, T found in the atmosphere, any particular absorption coefficient $k(p, T)$ will always have the same cumulative probability. Thus, the absorption coefficients at different pressure and temperature levels are correlated with each other. In other words, the shape of $k(g)$ is assumed to be constant, but the magnitudes of the ks are scaled with p and T (e.g. Petty, 2006).

Frequently the absorption within a spectral interval arises from overlapping lines of two or more molecular species. The standard procedure in this situation (Kratz, 1995) is to assume the spectral features of the species to be uncorrelated. The mean transmittance for a combination of two constituents with mass paths u_1 and u_2 can then be expressed as a double summation:

$$t_{\Delta\lambda}(u_1, u_2, p, T) \approx \sum_{i=1}^{N} \sum_{j=1}^{M} \{w_i \exp\left[-k_i(p, T)\, u_1\right]\} \{w_j \exp\left[-k_j(p, T)\, u_2\right]\} . \tag{2.30}$$

Kratz (1995) computed optimised k-coefficients for the five AVHRR channels. He later applied the same technique to several of the MODIS channels (Kratz, 2001). Table 2.1 lists the MODIS channels centred at 3.7, 8.5, 11 and 12 μm, together with the spectral intervals taken into consideration in the correlated-k routines, and the atmospheric constituents that contribute to the absorption in each channel. The number of k-coefficients required for the integration is as low as one to five, depending on the gas.

Table 2.1: Spectral intervals (as considered in the Kratz (2001) correlated-k routines) of the four MODIS channels implemented in the forward model, the atmospheric constituents whose absorptivity is accounted for, and the number of k-coefficients required for the spectral integration of each species.

channel	centre wavelength (μm)	wavelength interval (μm)	considered gases (no. of ks)
20	3.748	3.656 - 3.839	H_2O (4), CO_2 (1), CH_4 (1)
29	8.553	8.333 - 8.772	H_2O (5), O_3 (1), CH_4 (1), N_2O (2)
31	11.010	10.526 - 11.494	H_2O (5), CO_2 (1)
32	11.920	11.494 - 12.346	H_2O (5), CO_2 (1)

The Kratz (2001) MODIS routines were not part of libRadtran at the time the work on this thesis was conducted. I thus implemented the corresponding functions into the model in order to compute the spectral-mean intensities for the desired channels.

2.5 The Overlying Atmosphere

Figure 2.7 showed that most of the absorption above cloud top is due to water vapour. As discussed in Chapter 1, marine Sc often appear under conditions of strong subsidence in subtropical regions. This generally implies a low water vapour content of the free atmosphere, nevertheless, it should be investigated how large the impact of the existing water vapour is and how it can be accounted for in the forward model.

The way the overlying atmosphere (referred to below as OA) is treated in the existing literature varies, but most studies (Pérez et al., 2000; González et al., 2002; Cerdeña et al., 2007) ignore absorption above cloud top, arguing that the subsiding air is mostly dry and that most of the atmospheric water vapour is contained in the boundary layer air – so that its effects are already contained in the observed clear sky brightness temperatures (which are used to fix parameters of the cloud model, cf. section 1.5.2).[21] The other possibility (Pérez et al., 2002) is to compute the bulk absorption effects from a given water vapour path observed from an independent source (e.g. microwave soundings or reanalyses of weather forecast models).

In order to estimate the impact of the OA for my retrieval scene, I computed absorption and emission effects from radiosonde water vapour and temperature profiles measured by the closest operational meteorological sounding station in San Diego.[22] Two soundings daily were available, the nighttime soundings of July 11 (RF02) and July 13 (RF03) are plotted in Figure 2.9.

The moisture profiles (middle panels) in fact show a very dry atmosphere above the subsidence inversion, especially for July 11. Using adapted versions of the Kratz (2001) correlated-k routines, I computed emissivity and transmissivity for each layer defined in the sounding data (left and middle panels), as well the profile of the upwelling radiances (right panels). Since the scattering effects of gaseous particles are negligible for infrared wavelengths (Petty, 2006), scattering was not considered. Trace gas concentrations of 350 ppm for CO_2, 1.75 ppm for CH_4 and 0.31 ppm for N_2O were assumed for the computations, and the O_3 profile was taken from the US standard atmosphere compiled by Anderson et al. (1986).

For all channels, the transmissivity of the dry atmosphere encountered on July 11 is close to 1 km^{-1}

[21] This, of course, assumes that the boundary layer water vapour path of the utilised cloud free pixels equals that of the observed cloudy pixels.

[22] The data are freely available at http://weather.uwyo.edu/upperair/sounding.html.

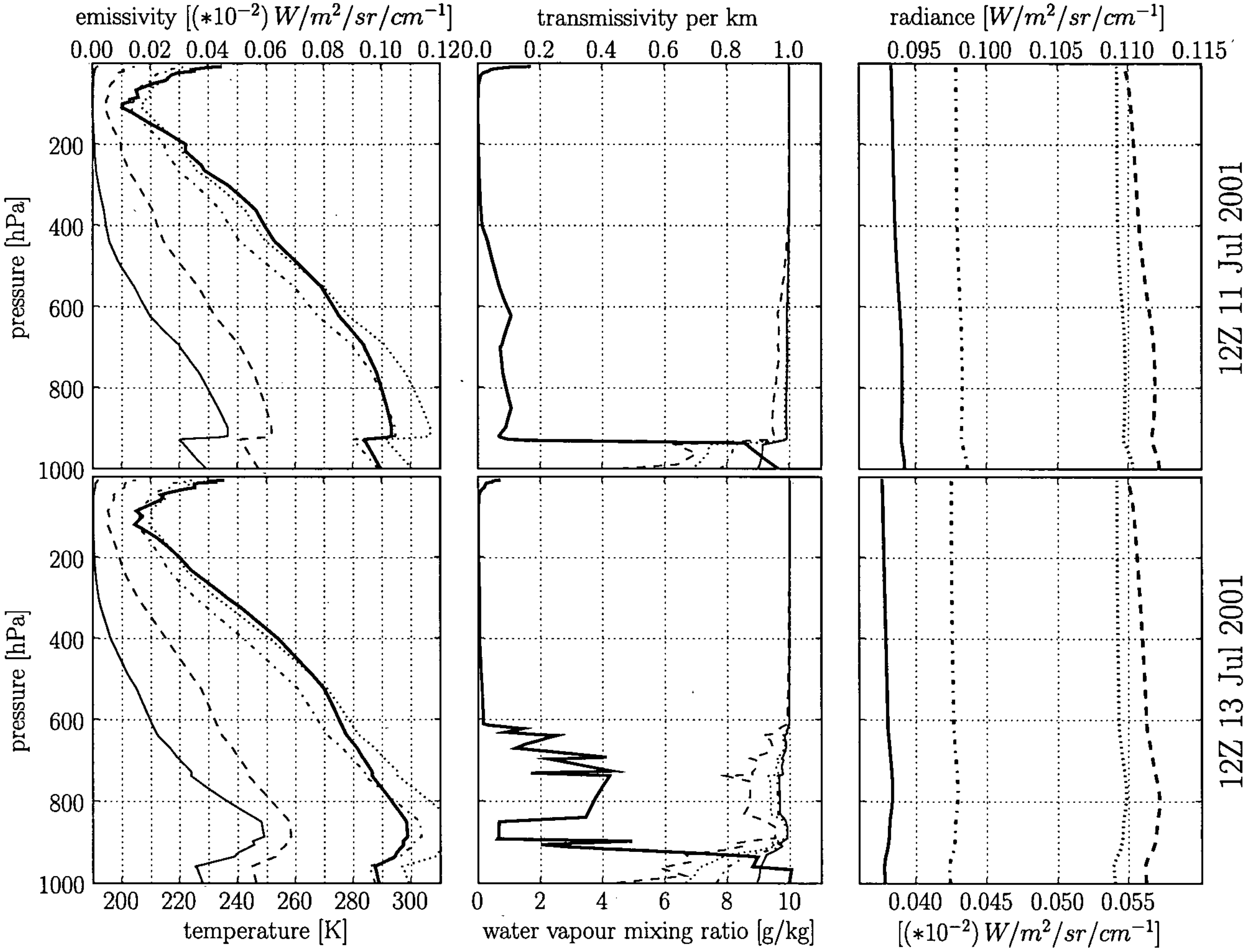


Figure 2.9: Atmospheric soundings from San Diego from (top) July 11, 2001, and (bottom) July 13, 2001. Shown are [left] temperature profile (thick) and corresponding emissivity of the atmosphere in the wavenumber ranges of the four MODIS channels 20, 29, 31 and 32; [middle] humidity (thick) and transmissivity for the channels; and [right] radiance profiles. No scattering has been considered for the computation of the radiances. Channels are marked as solid–channel 20, dashed–channel 29, dash-dotted–channel 31, dotted–channel 32. Channel 20 radiances and emissivities have been multiplied by 10^2 for better display (bottom scale; top scale for the remaining channels). See text for more details.

at all layers. Influenced most by the OA is channel 29. On July 13, there is a layer of increased humidity between 800 hPa and 600 hPa, where the layer transmissivity varies between 0.9 and 1 km^{-1} for channels 20, 31 and 32, and between 0.8 and 1 km^{-1} for channel 29. However, the attenuation of radiation by absorption is partly countered by emission, so that the total differences between the radiances at the inversion and the top of the atmosphere are small. On both days, the intensities changed by less that 2% for channel 20, less than 3.5% for channel 29 and less than 0.5% for channels 31 and 32. While these effects do not contribute much to the observed TOA radiances, they represent a potential and – if known – unnecessary source of uncertainty in the retrieval, hence, I decided to account for atmospheric transmission and emission above cloud top in the forward model.

The simplest, but also most computer time consuming approach to account for the OA in the forward model is to feed the entire sounding profile into the radiative transfer model. However, it is undesirable to make the LUT dependent on a specific atmosphere. Also, at most locations over the ocean, no radiosonde soundings are available in the vicinity of the retrieval scene. Rather, it is often possible to infer estimates of total LWP from an independent source as mentioned above. Alternatively, it might be feasible to add a total transmissivity t^* and average emitted intensity B^* of the OA as retrievable parameters to the algorithm. In order to keep my method flexible, I decided to account for OA effects by adding these two parameters to the forward model. Using this approach, the cloud top radiances can be computed independently from the OA, the impact of which can be added afterwards.

In my case, t^* and B^* can easily be computed from the available atmospheric soundings. In the absence of scattering, the RTE (2.8) can be written as *Schwarzschild's equation*:

$$I^{\uparrow}(\infty) = I^{\uparrow}(0)t^* + \int_0^{\infty} B(z)W^{\uparrow}(z)dz, \tag{2.31}$$

where $I^{\uparrow}(\infty)$ denotes the upwelling intensity at the top of the atmosphere, $I^{\uparrow}(0)$ the upwelling atmosphere at cloud top (the bottom of the OA), and $W^{\uparrow}(z)$ the *weighting function*, defined by

$$W^{\uparrow}(z) = \frac{dt(z,\infty)}{dz} = \frac{\beta_a(z)}{\mu}t(z,\infty). \tag{2.32}$$

By introducing the weighted average Planck function (Petty, 2006)

$$B^* = \frac{1}{1-t^*}\int_0^{\infty} B(z)W^{\uparrow}(z)dz, \tag{2.33}$$

and computing the total transmittance as the product of all individual layer transmissivities, the TOA intensity can be conveniently written as a function of cloud top radiance, t^* and B^*:

$$I^\uparrow(\infty) = I^\uparrow(0)t^* + B^*(1 - t^*). \tag{2.34}$$

In the above equations, the OA is effectively treated as a single homogeneous layer with transmittance t^* and emitted intensity B^*. Unfortunately, both parameters are wavelength dependent, so that they have to be computed for each channel. While this poses no constraint if the atmospheric profiles are known in advance, it adds additional parameters to the retrieval if they were to be inferred from the satellite observations. However, even in this case, it seems feasible to map t^* and B^* of one channel to all other channels employed in the algorithm. This idea is left for future work.

2.6 Forward Model

2.6.1 Radiative Transfer Model: libRadtran

The radiative transfer package libRadtran, described by Mayer and Kylling (2005), contains a number of tools for radiative transfer calculations in the earth's atmosphere. Particularly important for this thesis is its ability to read in arbitrary atmospheric and cloud profiles, as well as precomputed phase functions, and to solve the RTE using DISORT.[23] The package comes with the Kratz (1995) AVHRR correlated-k routines implemented, but, as mentioned above, I had to add the corresponding code for the MODIS channels (Kratz, 2001).

Clouds are included in the radiative transfer calculations by specifying vertical profiles of height, liquid water content and effective radius. Phase functions are read in from tables produced by the Mie code included in SHDOM (Evans, 1998), the same that was used to produce the examples in Figure 2.3. In Mie theory, the solution to the electromagnetic wave equation (cf. section 2.2) is expressed as a finite series of Legendre polynomials (e.g. Petty, 2006), so that the phase functions are listed in the Mie tables in terms of Legendre coefficients. These coefficients are directly used by DISORT for the numerical solution of (2.8).

The atmosphere is divided into discrete layers, for each of which the radiative transfer problem is solved. Absorption coefficients and single scatter albedos are computed from the Kratz (2001) routines, while the phase function is assumed to be constant over the spectral interval of the channel. DISORT is

[23] The complete software package, however, is capable of much more; see Mayer and Kylling (2005) for details.

called once for each k-coefficient, and the resulting intensities are summed using the weights w_i in (2.29) to compute the channel integrated intensities.

2.6.2 Design of the Forward Model

I implemented a system based on Python[24] scripts that automatically computes a number of adiabatic cloud profiles and calls libRadtran in order to generate a LUT. The cloud profiles can be computed from either randomly chosen cloud properties drawn from a given interval or from discrete values.

The forward model requires the four variable input parameters effective radius r_{eff}, total droplet number concentration N, cloud top pressure p_{ct} and cloud top temperature T_{ct} (either intervals or discrete values), as well as the fixed inputs surface pressure p_{sfc}, gamma distribution shape parameter ν_{gam} for computing the Mie properties of the cloud droplets, k-value for the conversion of r_{vol} to r_{eff}[25] and the number of datapoints that should be computed (if discrete values of the first four parameters are given, this input is not applicable). The system is able to run in parallel mode, so that the generation of large LUTs can be performed in reasonable time. For instance, the generation of 96,000 datapoints requires about 8 hours on a cluster of 32 2-GHz processors.

After the input parameters are read in, liquid water content q_l is computed from r_{eff} and N using (2.18) and (2.19). The specific humidity q_v is obtained from (2.16) by using T_{ct} and p_{ct}. Since libRadtran uses height as a vertical coordinate, the hydrostatic equation (e.g. Curry and Webster, 1999) is used to relate pressure and height in the cloud model, so that T_{sfc} can be computed from (2.12). The atmospheric and cloud variables are then integrated using a defined step-size (e.g. 0.2 hPa) from the surface to cloud top. Both cloud (z, r_{eff}, q_l) and atmospheric (z, p, T, specific humidity q_v and trace gases) profiles are passed to libRadtran, where for this study, the same trace gas concentrations were used as for the overlying atmosphere in section 2.5. libRadtran is then run to compute the cloud top radiances. The cloud visible optical thickness is computed using (2.25), with $Q_{ext} = 2$.

If desired, the effects of the OA can be accounted for using the procedure described in section 2.5. Precomputed t^* and B^* as well as radiosonde soundings can be used. As a last step, the TOA intensities are converted to brightness temperatures using the inverse of (1.18). Since the Planck function also varies over the spectral intervals of the channels, the channel-mean BT can either be approximated at the channel-centre wavelength, or a correction formula suggested by van Delst (2005) can be used in order to account for the polychromaticity of the channels.

[24] The free programming language Python can be obtained at http://www.python.org/.

[25] ν_{gam} and k are held separate in order to facilitate sensitivity studies and to avoid time consuming re-computations of the Mie properties for small changes of k.

For all computations, the satellite is assumed to be directly above the cloud (i.e. TOA BTs are computed for nadir view), no satellite zenith angle is considered.

Figure 2.10 shows two vertical profiles of channel 20 (3.7 μm), 31 (11 μm) and 32 (12 μm) brightness temperatures for an optically thin ($\tau \approx 3.8$) and an optically thick ($\tau \approx 43$) cloud. The plots illustrate how the attenuation of radiation within the cloud layer is simulated by the forward model. As discussed in Chapter 1, the surface signal still influences the cloud top radiances for the thin cloud case, while for the other case, the cloud emits as a black body in channels 31 and 32. The point at which the BTs of all three channels coincide with the actual temperature is clearly visible. At this point, all photons propagating upwards through the cloud that entered the cloud at its bottom are extinguished, and the cloud emits as a black body with its actual temperature T. The intensities observed at cloud top are only determined by emission in this upper part of the cloud (cf. section 1.5.2).

Figure 2.11 shows the BTD(3.7-11) and BTD(11-12) signals, each with the 11 μm BT as reference. As in Pérez et al. (2000), cloud top temperature is fixed at 285 K. Since in my forward model, cloud top pressure is a fixed parameter as well (in the given case at 900 hPa), the surface temperatures as computed from the adiabatic lapse rates depend on the cloud geometrical thickness and hence differ for the individual clouds. Thus, the curves do not converge in a single point on the right side, as they do in Figure 1.9.

My model is able to reproduce the relationships found by Baum et al. (1994) and Pérez et al. (2000). As expected and discussed in Chapter 1, the changes in BT with changing optical thickness become smaller for larger τ. In contrast to the VU cloud model, the maximum optical thickness in the AS model depends on r_{eff}. Consequently, clouds with small effective radii cannot become as optically thick as in the Baum et al. (1994) and Pérez et al. (2000) studies, where both parameters were independent (cf. Figure 1.9). As noted in Chapter 1, if T_{ct} and p_{ct} are known, r_{eff} and τ could in principle already be retrieved manually from the plots in Figure 2.11.

The characteristic of my AS model that surface temperature is obtained from the adiabatic lapse rates is particularly pronounced is the case of constant r_{eff}, N and T but varying p_{ct}. As shown in Figure 2.12, changes in cloud top pressure can cause large differences in the BTD signals. In Chapter 4, I will fix cloud top pressure in the LUT in order to simplify the retrieval problem. It will hence be important to evaluate the sensitivity of the retrieval to changes in cloud top pressure (or more accurately, to the pressure difference between cloud top and sea surface) in order to estimate the impact of slightly varying cloud top heights in the scene to the retrieval accuracy. I will come back to this topic in section 4.4.

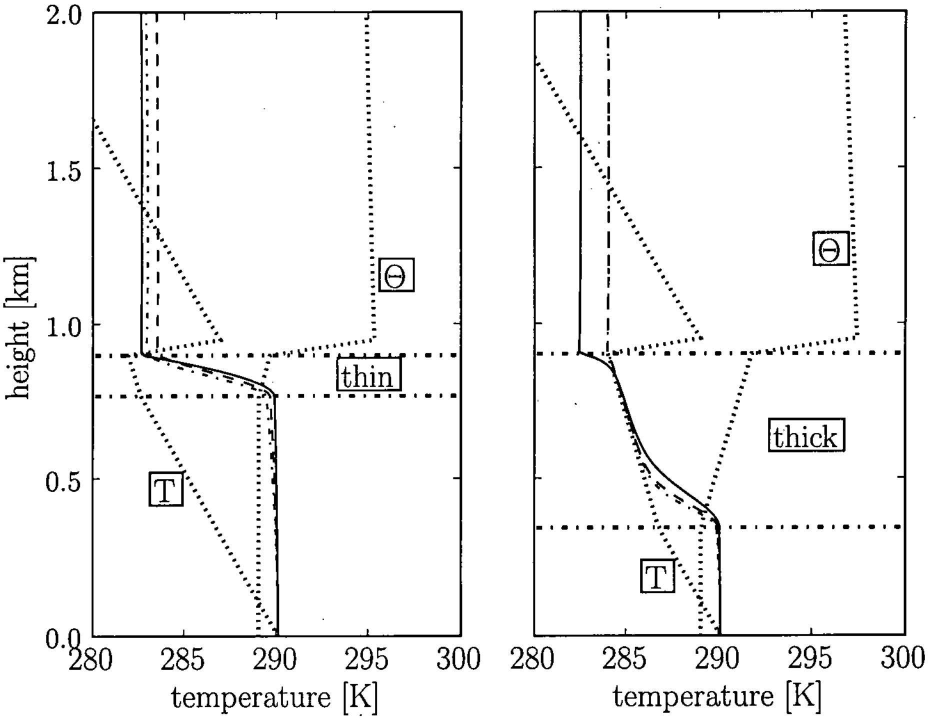


Figure 2.10: Vertical brightness temperature profiles through a thin cloud (left) and a thick cloud (right). Shown are the brightness temperatures of MODIS channel 20 (3.7 μm; thin solid line), channel 31 (11 μm; thin dashed line) and channel 32 (12 μm; thin dash-dotted line), as well as temperature (T) and potential temperature (Θ). For the thick cloud, the brightness temperatures of channels 31 and 32 are similar to the temperature of the cloud (i.e. the cloud emits as a black body in this wavelength range), whereas for the thin cloud, the surface temperature still influences the signal.

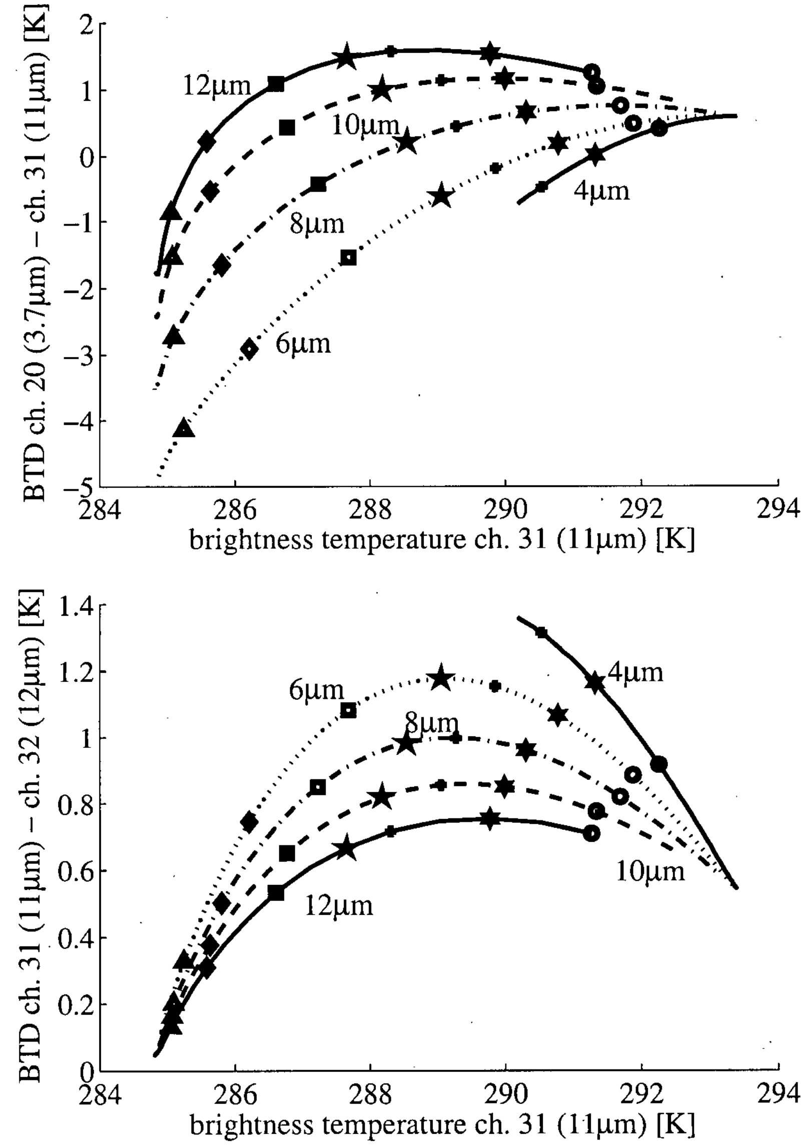


Figure 2.11: Droplet size and cloud optical thickness influence radiation at the different MODIS channels to a different extend. Shown is the dependence of brightness temperature (BT) and BT difference (BTD) on varying effective radius and optical thickness for a cloud top temperature of 285 K and a cloud top pressure of 900 hPa. Symbols from left to right: optical thickness of 0.5, 1, 1.5, 2, 3, 5, 8. Note that clouds with a small effective radius cannot become as thick as clouds with a larger r_{eff} in the adiabatic cloud model (compare to the vertically uniform model of Pérez et al. (2000)).

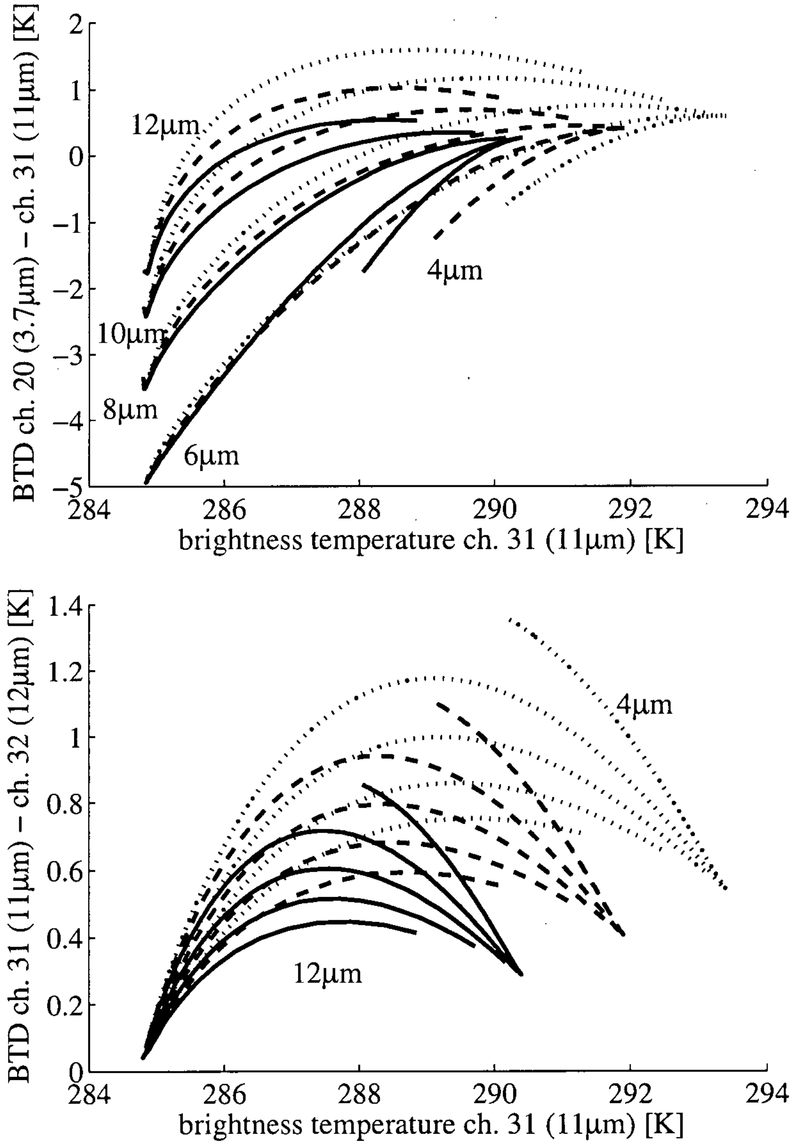


Figure 2.12: The same in Figure 2.11, but for different cloud top pressures (solid–940 hPa, dashed–920 hPa, dotted–900 hPa). In the adiabatic model, a different cloud top pressure with a fixed cloud top temperature results in a different surface temperature. Hence, the left convergence points in the figures (surface temperature) are warmer for higher cloud tops.

Chapter 3

Nonlinear Regression with Neural Networks and Uncertainty Estimation

The forward model I developed in the previous chapter maps given cloud properties to satellite observations computed by the radiative transfer model. From a set of such forward computations, the unknown inverse function has to be inferred in order to determine the cloud parameters from the observed satellite data. This poses a nonlinear multiple regression problem, which, as proposed in Chapter 1, I will tackle with artificial neural networks and the Bayesian framework suggested by MacKay (1992a,b) and Aires (2004); Aires et al. (2004a,b).

My objective for this chapter is to describe this approach and to discuss its applicability to my remote sensing problem. For simplicity, the original MacKay (1992a,b) publications, as well as the book by Bishop (1995) and the review paper by MacKay (1995), only discuss networks with one output variable and hence only one output uncertainty. For multidimensional mappings, MacKay (1995) suggests the use of full covariance matrices to describe the output uncertainties, but does not elaborate on this idea. Aires (2004); Aires et al. (2004a,b) point out the importance of uncertainty estimates for multidimensional mappings that arise in atmospheric inverse problems, and expand the original MacKay (1992a,b) method to networks having multiple input and output variables.

In this chapter, I will first formulate the regression problem in a general context (section 3.1). This introduction (mainly based on the book by Bishop (1995)) is useful in order to understand the problems that arise in neural network training for which the Bayesian framework provides some remedy. I will then derive the theoretical foundation of how the uncertainty in the network prediction and the Jacobian can be estimated (sections 3.2 - 3.5). The derivation will be illustrated by an application of the method to a simple problem, which will demonstrate important benefits but also highlight problems.

I implemented the Aires method by extending the NETLAB toolbox by Nabney (2002), a set of routines implemented in Matlab[26]. Some information about the implementation is given in section 3.6; in section 3.7 I discuss the usefulness of the method for my remote sensing problem.

3.1 The Nonlinear Regression Problem

Consider the problem of nonlinear regression on noisy data, for which MacKay and Aires et al. derive their methods. The data consist of a dataset $\mathcal{S} = \{X, D\}$ with *input* data $X = \{\boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^N\}$ and *target* (or *observed*) data $D = \{\boldsymbol{t}^1, \boldsymbol{t}^2, ..., \boldsymbol{t}^N\}$. $\boldsymbol{x}^n$ and $\boldsymbol{t}^n$ denote vectors that contain all input and target variables, respectively. We assume the inputs X to be exact, while the target data D are noisy.

Strictly speaking, for the inverse problem, we deal with the opposite case; the input data – the satellite measurements – are noisy, whereas the target data – the inputs to the forward model – are known exactly at the time the dataset $\mathcal{S}$ is generated. I will come back to this issue shortly.

3.1.1 Error Functions and Overfitting

Once the dataset $\mathcal{S}$ has been observed, we wish to gain information about the function or relation which generated the targets from the inputs. Since the target data are noisy, they are composed of two parts – the underlying generator, representing the physical relationship we wish to capture, and the noise.

Let $\mathcal{M}$ denote a statistical model for $\mathcal{S}$. $\mathcal{M}$ will be a neural network shortly, but first consider a polynomial of order K:

$$y(x) = w_0 + w_1 x + ... + w_K x^K = \sum_{j=1}^{K} w_j x^j. \tag{3.1}$$

This model maps an input variable x to an output variable y, the *prediction.* The exact form of the model function depends on the order of the polynomial and the values of the parameters w_j. Hence, following Bishop (1995), I write $y = y(x; \boldsymbol{w})$ (where all parameters w_j are grouped into one parameter vector $\boldsymbol{w}$).

A standard way to find the best possible model architecture for a given $\mathcal{S}$ is to consider the errors between the model predictions y^n and the desired values, the targets, t^n for the N data pairs in $\mathcal{S}$. In order to obtain a good fit, the differences $y^n - t^n$ should be as small as possible, which can be achieved by minimising the sum-of-squares error (cf. the cost function (1.22))

$$E = \frac{1}{2} \sum_{n=1}^{N} \{y(x^n; \boldsymbol{w}) - t^n\}^2 . \tag{3.2}$$

[26] http://www.mathworks.com/

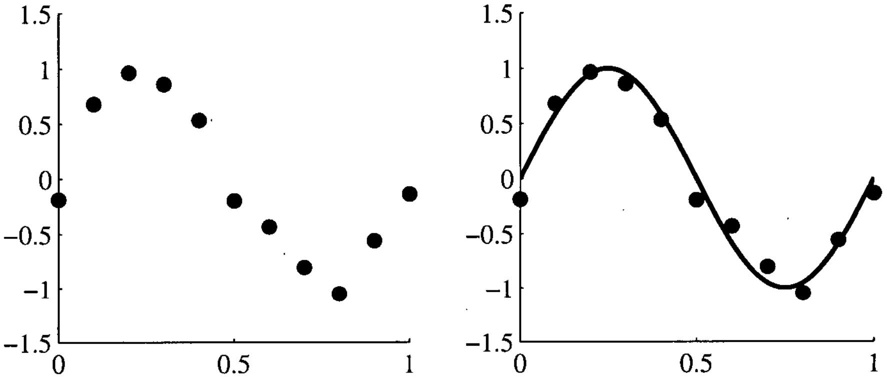


Figure 3.1: Left: a dataset S, from which we would like to infer information about the *underlying generator*, i.e. the function from which the values were generated. Right: the sine function from which the data were produced. However, if the shape of the generating function is not known, it is difficult to judge different models based only on their sum-of-squares error (compare Figure 3.2).

However, will the model with the smallest error E best represent the underlying generator of the data? In fact, this often is not the case. The phenomenon is known as *overfitting* in the literature, and describes the problem of fitting the noise rather than the generating function. Its major effect is that models that fit the training data S too well do not *generalise* well. In this case the model, being presented with an input value which is not part of the training dataset, predicts a value far from the desired one.

Figures 3.1 and 3.2 illustrate the problem. In the left panel of Figure 3.1, eleven datapoints are plotted, which are generated by adding random noise to the sine function depicted in the right panel. Figure 3.2 shows two polynomials fitted to the data, in the left panel a cubic polynomial, in the right panel a polynomial of order ten. Since the underlying generator of the data is known, we easily judge that the cubic polynomial is a better representation of the original sine function. The sum-of-squares error, however, is much smaller for the higher order polynomial, since it perfectly fits the training data. If we knew nothing about the original function, how could we judge which model is best?[27]

The same problem arises if neural networks are used instead of polynomials. For simplicity of the derivation and implementation, I restrict myself to a two-layer perceptron (i.e. one layer of hidden units) as used by Aires et al. As discussed by, for instance, Bishop (1995), this network is already capable of approximating arbitrary functions.

Mathematically, an ANN such as the one depicted in Figure 1.10 computes the output values from the following equations. The inputs x_i are weighted by the parameters $w_{i,j}^{(1)}$ of the first layer, then summed

[27] Concluding that less complex models (i.e. lower degree) are better is also misleading – consider, for example, a linear polynomial for the given case.

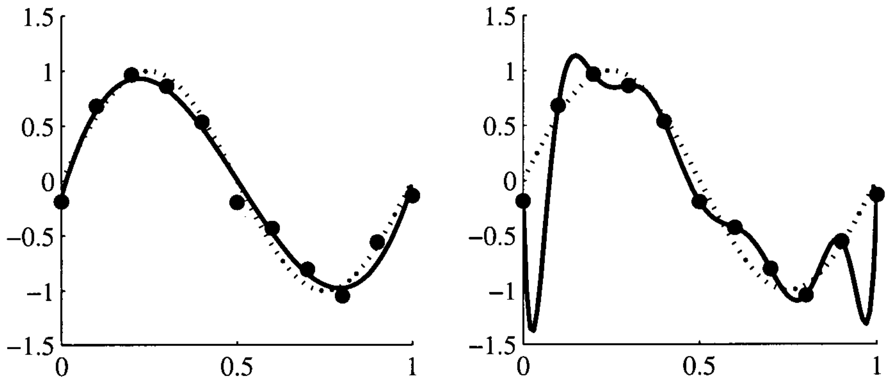


Figure 3.2: Left: a cubic polynomial (solid line) fitted to the data from Figure 3.1, together with the generating sine function (dashed). Right: a polynomial of order ten, also plotted with the generating sine function. It is easily seen by eye that the cubic polynomial represents the sine function better, however, the sum-of-squares error is smaller for the right polynomial. This phenomenon is known as *overfitting.*

for each hidden neuron and transformed by an activation function $g(\cdot)$ to give the hidden values

$$z_j = g\left(\sum_{i=0}^{d} w_{i,j}^{(1)} x_i\right). \tag{3.3}$$

The same equation is applied to the hidden values, which yields the output values

$$y_k = \tilde{g}\left(\sum_{j=0}^{M} w_{j,k}^{(2)} z_j\right), \tag{3.4}$$

where $\tilde{g}(\cdot)$ denotes the activation of the output units. In the architecture considered here, $g(\cdot)$ is a sigmoidal function, whereas $\tilde{g}(\cdot)$ is a simple linear function (necessary to obtain output values other than zero and one). Combining (3.3) and (3.4) gives the complete network function

$$y_k = \tilde{g}\left(\sum_{j=0}^{M} w_{j,k}^{(2)} \times g\left(\sum_{i=0}^{d} w_{i,j}^{(1)} x_i\right)\right). \tag{3.5}$$

As for the polynomial in (3.1), the goal is to find a set of weights $\boldsymbol{w}$ so that (3.5) becomes the best possible representation of the generator of the dataset $\mathcal{S}$. Also, as in the polynomial example, a trade-off between model complexity and a small error between predicted outputs and target data has to be found in order to find a model that generalises the data well.

The problem of finding a model that is neither too complex nor too simple is known as *Occam's* *razor* (after William of Occam, 1285-1349; see Bishop (1995)). Assessing the representativeness of the model and controlling its complexity is an important part of assessing the uncertainty inherent in the predictions.

3.1.2 Controlling Model Complexity and Network Training

Finding the optimal set of weights in (3.5) involves finding the minimum of an error surface in weight space, such as defined by (3.2). Obviously, the number of weights in an ANN architecture rapidly increases with the number of hidden units. Hence, the error surface is of a high dimensionality and typically has local minima.

The neural network error function is similarly to the cost function (1.22) between computed and observed radiances in the retrieval methods described in section 1.5 too complex to be minimised analytically. Instead, several numerical algorithms exist to perform the minimisation. Overviews of commonly used algorithms and their functionality are given by, for instance, Bishop (1995) and Press et al. (2007). In this work, I used the scaled conjugate gradients algorithm, as implemented by Nabney (2002) in the NETLAB toolbox.

Obviously, both the number of adaptive parameters (i.e, weights) in a network architecture and the type of the error function determine the complexity of the error surface. Common problems arise in specifying the error surface in a way so that the global minimum does not correspond to a state of overfitting and in actually finding this global minimum – numerical minimisation algorithms often get stuck in local minima, depending on the initial point where the search for the global minimum started.

Bishop (1995) points out two principal approaches to controlling the complexity of the model. The first, called *structural stabilisation*, involves controlling the number of adaptive parameters in the network. The second, *regularisation*, adds a penalty term to the error function (3.2) to counteract overfitting by avoiding strongly fluctuating mappings such as the example in the right panel of Figure 3.2.

The simplest approach to network structure optimisation (and, as Bishop (1995) points out, still the most widely adopted approach in practise) is to perform an exhaustive search through a restricted class of architectures (i.e. varying number of hidden units). Obviously, such a search requires a large computational effort. Nevertheless, due to the lack of easy-to-use alternatives (cf. Bishop, 1995, Chapter 9), I will follow this approach as well.

The optimisation of model complexity is performed with respect to a given training dataset. The simplest approach to comparing the generalisation performance of different network architectures is evaluate the error function for an independent validation dataset. A common approach is to split the original dataset S into two parts, one used for training and one for validation (typically two-thirds are used for training and one-third for validation). After the models have been compared, the one with the smallest validation error can be selected.

As for controlling the complexity of the model by regularisation, I will show in sections 3.2 and 3.3 that the Bayesian approach provides a natural framework for both regularising the training process and estimating the uncertainty of the network predictions.

3.1.3 Describing Uncertainty

The idea of the Bayesian approach is that instead of finding the single weights vector $\boldsymbol{w}^*$ that minimises an error function E and represents the most probable solution to the regression problem, we compute an entire *distribution* of weights. By examining this distribution, we can see how "certain" it is that the training process has found the generator of the data; if the distribution is wide, it is uncertain, if it is narrow, it is certain. Since the distribution of the weights is inferred from the training data, it is usually written as

$$p(\boldsymbol{w}|D). \tag{3.6}$$

In order to simplify the notation and to follow the notation of Bishop, the dataset is only denoted by D. Implicitly, however, the distribution depends on the entire dataset, including the input values (written as $p(\boldsymbol{w}|D, X)$).[28]

The certainty in the network outputs is also restricted by the noise on the data. If we were, for example, to model the long-term generator of hourly-averaged windspeed given a dataset S composed of minute-by-minute measurements there would be *intrinsic noise* caused by turbulence on the measurements that would not be part of the generator.

Obviously, apart from describing this noise with a probability distribution, we cannot make statements about exactly where the value of the generator lies within a noise interval. The noise distribution

$$p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}) \tag{3.7}$$

[28] In section 3.2 I will show how so-called *prior information* about the weights distribution can be inserted into the training process that leads to the weights distribution (3.6). Research showed that weight values close to zero lead to smoother network mappings than large weights (Bishop, 1995, Section 10.1.2). This can be achieved by favouring small weights by using a special prior, a process is known as *regularisation* in the literature. Regularisation is an important tool to control the complexity of the model. However, it does not solve the problem of how many weights and hidden nodes the architecture should contain. In the *evidence framework* (MacKay, 1992a; Bishop, 1995, Section 10.6), the Bayesian approach provides the tools for comparing different architectures (e.g. different numbers of weights) of networks based on the training data. This topic, however, is outside the scope of my thesis and will not be discussed further.

gives the probability of observing a specific target $\boldsymbol{t}$, given an input $\boldsymbol{x}$ and the most likely generator function represented by a network with weights $\boldsymbol{w}$.

To come back to the issue that the target data in the forward model are not noisy, this problem in fact turns out to be a flaw in the application of the method to the retrieval problem. However, from the studies of Pérez et al. (2000) and González et al. (2002), we expect the forward model to contain ambiguities. One can reason that these ambiguities could be interpreted as noise on the target data. Furthermore, the noise in the input data (the satellite observations) can be accounted for with the network Jacobian. This issue will be further discussed in the remainder of this thesis.

Given the two sources of uncertainty described by the weights distribution and the noise distribution, the total uncertainty of the network predictions can be calculated from

$$p(\boldsymbol{t}|\boldsymbol{x}, D) = \int p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}) p(\boldsymbol{w}|D) d\boldsymbol{w}, \tag{3.8}$$

where the noise distribution has been integrated over all likely generators $\boldsymbol{w}$. This distribution describes the probability that a target $\boldsymbol{t}$ will be observed if an input vector $\boldsymbol{x}$ is given and the general behaviour of the target data is specified by the training dataset D.

When I first studied the derivation of the network outputs distribution (3.8), it appeared confusing to me that the uncertainty in the network outputs, denoted by $\boldsymbol{y}$, is expressed as a distribution of the targets $\boldsymbol{t}$. However, since we have no information about the generator of the data except the dataset $\mathcal{S}$, it is best to make predictions about new data that would likely be observed for a new input. The network prediction $\boldsymbol{y}$ describes the mean of the posterior target distribution, and hence the location of the most probable data value to be observed. The shape and width of $p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w})$, however, describe the certainty that a data value we would observe in reality would actually be the most probable value – hence, the distribution width can be interpreted directly as error bars on the network prediction $\boldsymbol{y}$. The weights distribution $p(\boldsymbol{w}|D)$ additionally describes our degree of belief in the ability of the network $\boldsymbol{w}$ to model the generator, thereby further widening the distribution of possibly observed data values.

The network Jacobian represents another important tool to analyse the model. It is defined as the derivative of the network outputs with respect to its inputs, $\partial y_k / \partial x_i$. For instance, the Jacobian provides a mechanism to estimate the impact of errors associated with the inputs on outputs errors. Furthermore, the Jacobian provides a good tool to answer questions concerning the complexity and adequateness of the mapping – to what extend does the output change when the input is changed? Is this the behaviour we expect the solution to have? Is the model sensitive to the inputs we expect it to be sensitive to given our knowledge of the problem? This information can make the ANN much more transparent than a "black-box network" would be.

Using the weights distribution, it is also possible to estimate a Jacobian distribution. As mentioned in Chapter 1, such an estimate of the variability in the Jacobian is a useful indicator of how ill-conditioned the regression problem in question is. Aires et al. (2004b) point out that inverse problems are often ill-posed; similar output statistics of the network can be obtained by a variety of different network parameters (i.e. weights). A possible reason is redundant information in the input variables, caused, for instance, by correlated inputs. Imagine, for example, a network with one output and two highly correlated inputs. The output might physically be dependent on the magnitude of one of the two inputs as well as their difference. In this case, the training algorithm cannot decide which input carries the information, and separate training runs could yield different results in terms of the ANN weights. The Jacobian would be fundamentally different between the networks. In one case the output variable could be mainly sensitive to the first input and to a lesser extend to the second, while in the other case it could get most of its information from the second input and represent the dependence on the input difference with an opposite sign dependence on the other input. The output statistics for the training data might be similarly good for all networks, but the ability to generalise to new inputs could vary considerably. In particular, if we know little about the problem and are interested in interpreting the network Jacobians as actual physical Jacobians (so that we can learn on which inputs the output depends), such a high variability in the Jacobian is problematic and we should be careful about assuming that the network represented by the most probable weights vector models the real generator. If, however, the variability in the Jacobian is small, we might expect that a "good" mapping has been found.

In the following sections, I will show how the weights distribution, output uncertainties and variability in the Jacobian can be obtained. To illustrate some important aspects, I will use a simple example function – a mapping from two input to two output variables, given by

$$y_1(x_1, x_2) = 4x_1^2 + \sin(2\pi x_2) \tag{3.9}$$

$$y_2(x_1, x_2) = \cos(2\pi x_1) + x_2. \tag{3.10}$$

A two-layer network with six hidden units has been trained from a dataset generated from these equations by adding normally distributed noise with a standard deviation of 0.1 to output 1 and 0.05 to output 2. I will show examples that originate from two training runs – a long run and a short run –, resulting in different posterior weights distributions and consequently different output uncertainties and Jacobian variabilities.

3.2 Distribution of the Neural Network Weights

In this section I will present and in places elaborate on the Aires (2004) derivation of the posterior network weights distribution, $p(\boldsymbol{w}|D)$, and demonstrate how neural network training can be formulated in the Bayesian framework.

3.2.1 Derivation

Using Bayes' theorem, $p(\boldsymbol{w}|D)$ can be written as

$$p(\boldsymbol{w}|D) = \frac{p(D|\boldsymbol{w})\,p(\boldsymbol{w})}{p(D)}. \tag{3.11}$$

The density $p(D|\boldsymbol{w})$ is called the *likelihood* of the data and indicates how likely the observation of the dataset D is in the light of a given model (here represented by a set of weights). $p(\boldsymbol{w})$ is called the *weights prior.* This density represents all information about the weights that is available before the data is observed. Finally, the denominator $p(D)$ is a normalisation factor to ensure that the integral over $p(\boldsymbol{w}|D)$ is one:[29]

$$p(D) = \int p(D|\boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}. \tag{3.12}$$

In order to find the mapping that best represents the generator of the data, the maximum of the posterior distribution $p(\boldsymbol{w}|D)$ has to be found (following Aires, I will write $\boldsymbol{w}^*$ for this maximum). Therefore, expressions for $p(D|\boldsymbol{w})$ and $p(\boldsymbol{w})$ are needed.

The first assumption that is made is that the target data are composed of a smooth function $\boldsymbol{h}$ and an additive noise component $\boldsymbol{\epsilon}$ (Bishop, 1995, Chapter 6):

$$\boldsymbol{t} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{\epsilon}. \tag{3.13}$$

The noise $\boldsymbol{\epsilon}$ is assumed to follow a Gaussian distribution, which I write as a multivariate Gaussian with zero mean (zero mean because it is the noise around the generator value):

$$p(\boldsymbol{\epsilon}) = \frac{1}{(2\pi)^{c/2}\,|\boldsymbol{C_{in}}|^{1/2}} \exp\left(-\frac{1}{2}\,\boldsymbol{\epsilon}^T \cdot \boldsymbol{C_{in}^{-1}} \cdot \boldsymbol{\epsilon}\right). \tag{3.14}$$

[29] Note that again, the explicit dependence on the input data X has been omitted in the notation.

$\boldsymbol{C_{in}}$ denotes the covariance matrix of this *intrinsic* noise on the data; c denotes the number of network outputs (cmp. Figure 1.10). The generator function $\boldsymbol{h}(\boldsymbol{x})$ is the function that we would like to model with the neural network, therefore I replace $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{y}(\boldsymbol{x};\boldsymbol{w})$. Combining (3.13) and (3.14) yields the distribution of the target variables:

$$p(\boldsymbol{t}|\boldsymbol{x},\boldsymbol{w}) = \frac{1}{(2\pi)^{c/2}\,|\boldsymbol{C_{in}}|^{1/2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{y}(\boldsymbol{x};\boldsymbol{w}) - \boldsymbol{t}\right)^T \cdot \boldsymbol{C_{in}}^{-1} \cdot \left(\boldsymbol{y}(\boldsymbol{x};\boldsymbol{w}) - \boldsymbol{t}\right)\right). \tag{3.15}$$

Assuming that all datapoints of the dataset are drawn independently from this distribution, the probability density of the entire dataset becomes

$$\begin{aligned} p(D|\boldsymbol{w}) &= \prod_{n=1}^{N} p(\boldsymbol{t}^n|\boldsymbol{x}^n,\boldsymbol{w}) & (3.16)\\ &= \frac{1}{(2\pi)^{c/2}\,|\boldsymbol{C_{in}}|^{1/2}} \exp\left(-\frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{\epsilon}^n)^T \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{\epsilon}^n\right), & (3.17)\end{aligned}$$

where I have used $\boldsymbol{\epsilon}^n = \boldsymbol{t}^n - \boldsymbol{y}^n$ and replaced $\boldsymbol{A_{in}} = \boldsymbol{C_{in}}^{-1}$. The matrix $\boldsymbol{A_{in}}$ is called a *hyperparameter* in the context of Bayesian learning, since it is a parameter that controls the distribution of other parameters (the network weights). I will explain its function in the next paragraph. In analogy to the conventional maximum likelihood approach (Bishop, 1995, Chapter 6), the sum in the exponential is called the *data error function*

$$E_D(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{\epsilon}^n)^T \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{\epsilon}^n. \tag{3.18}$$

In order to find an expression for the weights prior $p(\boldsymbol{w})$, Aires (2004) assumes that the weights follow a Gaussian distribution as well:

$$p(\boldsymbol{w}) = \frac{1}{Z_W} \exp\left(-\frac{1}{2}\boldsymbol{w}^T \cdot \boldsymbol{A_r} \cdot \boldsymbol{w}\right) = \frac{1}{Z_W} \exp\left(-E_W(\boldsymbol{w})\right). \tag{3.19}$$

For convenience and following Aires (2004), I have grouped all normalisation factors into a single constant Z_W. The parameter $\boldsymbol{A_r} = \boldsymbol{C_r}^{-1}$ is the inverse of the covariance matrix $\boldsymbol{C_r}$ of the weights and the second hyperparameter that occurs in the context of Bayesian neural network learning.

Analogous to the data error function, the weights error function is defined as

$$E_W(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^T \cdot \boldsymbol{A_r} \cdot \boldsymbol{w}. \tag{3.20}$$

In the form given by (3.19), the prior distribution has zero mean, thereby expressing that the weights are expected to be centred around zero. The use of such a prior weights distribution regularises the training process (cf. subsection 3.1.2); smooth network mappings, which usually generalise better than strongly fluctuating functions, can be achieved by favouring small weights (Bishop, 1995, Section 10.1.2). If the weights are large, then E_W will be large and $p(\boldsymbol{w})$ will be small; for small weights $p(\boldsymbol{w})$ will be large. Hence, by setting $\boldsymbol{A_r}$ correspondingly, we can prefer small weights over large ones. Bishop points out that priors other than Gaussian can be considered as well. In this work, however, I will only consider the Aires (2004) approach.

Given the expressions for $p(D|\boldsymbol{w})$ and $p(\boldsymbol{w})$, the posterior distribution of the weights (3.11) becomes

$$p(\boldsymbol{w}|D) = \frac{1}{Z} \exp\left(-\frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{\epsilon}^n)^T \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{\epsilon}^n\right) \exp\left(-\frac{1}{2}\boldsymbol{w}^T \cdot \boldsymbol{A_r} \cdot \boldsymbol{w}\right) = \frac{1}{Z}\exp\left(-E_D - E_W\right), \qquad (3.21)$$

where I have again used the shorthand notation Z for the normalisation factors. This expression could be maximised, however since many standard algorithms exist to minimise a function rather than to maximise it (Bishop, 1995, Chapter 7) it is more convenient to minimise the negative logarithm of (3.21). The logarithm removes the exponential, and because of its monotonicity the location of the minimum remains unchanged. Since the normalisation constant Z also does not affect the position of the minimum, we are left with minimising the total error function[30]

$$E(\boldsymbol{w}) = E_D(\boldsymbol{w}) + E_W(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{\epsilon}^n)^T \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{\epsilon}^n + \frac{1}{2}\boldsymbol{w}^T \cdot \boldsymbol{A_r} \cdot \boldsymbol{w}. \qquad (3.22)$$

Conventional (i.e. non-Bayesian) network learning can be regarded as a special case of this Bayesian framework; if we have no information about the weights prior and assume it to be a uniform distribution ($p(\boldsymbol{w}) = const.$), then $E_W = 0$. If we furthermore assume independent output variables (all off-diagonal elements of $\boldsymbol{A_{in}}$ are zero) which have the same variance (all diagonal elements of $\boldsymbol{A_{in}}$ set to σ^2), then σ^2 becomes a constant factor in (3.22) and can be omitted, leaving E to be a sum-of-squares error function as in the example of polynomial curve fitting:

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{c}\{y_k(\boldsymbol{x}^n;\boldsymbol{w}) - t_k^n\}^2 . \qquad (3.23)$$

A problem with maximising the posterior weights distribution is that the hyperparameters $\boldsymbol{A_{in}}$ and $\boldsymbol{A_r}$

[30] This is analogous to the principle of maximum likelihood; Bishop (1995, Chapter 2).

are usually not known beforehand. I will discuss how they can be estimated in section 3.4.

3.2.2 The Meaning of the Hyperparameters

MacKay (1992a, 1995) and Bishop (1995) introduce the Bayesian framework with only scalars β and α as hyperparameters instead of the matrices $\boldsymbol{A_{in}}$ and $\boldsymbol{A_r}$. For simplicity, they only use one output, thereby reducing $\boldsymbol{A_{in}}$ to a single element β. The weights are assumed to be independent, making $\boldsymbol{A_r}$ a diagonal matrix. In the simplest form, the variance of all weights is the same $(1/\alpha)$[31]. The error function (3.22) then becomes

$$E^{\alpha\beta}(\boldsymbol{w}) = \frac{\beta}{2}\sum_{n=1}^{N}\{y(\boldsymbol{x}^n;\boldsymbol{w}) - t^n\}^2 + \frac{\alpha}{2}\sum_{i=1}^{W} w_i^2. \tag{3.24}$$

If the intrinsic noise on the target data is small, then β will be large, and small deviations of the network predictions from the target will result in a large "penalty". If the noise is large, β will be small and larger differences will be tolerated. Setting the ratio α/β becomes important under the aspect of the size of the training dataset. While the first term in (3.24) grows with increasing numbers N of datapoints in the dataset, the second term does not. Hence, the ratio of both hyperparameters controls the importance of the weights prior and the size of the dataset form which it will become insignificant.

The matrix hyperparameters $\boldsymbol{A_{in}}$ and $\boldsymbol{A_r}$ used by Aires (2004) generalise this concept to interdependent weights and outputs, respectively. Using $\boldsymbol{A_{in}}$ instead of β allows for more outputs and also incorporates the correlation structure of the errors in the individual variables. (Remember that $\boldsymbol{C_{in}} = \boldsymbol{A_{in}}^{-1}$ does not represent the covariance matrix of the outputs, but of the errors in the outputs (the noise).) This way, mapping errors for variables with small noise are penalised stronger than those for targets with larger intrinsic noise.

In order to understand $\boldsymbol{A_r}$, it is important to keep in mind that the prior $p(\boldsymbol{w})$ is a Gaussian distribution with zero mean. This means that, similar to the scalar hyperparameter case, we expect the weights to be small. The major difference to the scalar case is that we assign different inverse variances (diagonal elements of $\boldsymbol{A_r}$ if the weights are independent) to different weights, hence controlling individually how much the weights are penalised for being large. This may be useful since the magnitudes of weights in different layers of a network can have fundamentally different ranges (MacKay, 1995; Aires, 2004).

[31] MacKay (1995, Section 3.2) notes that the weights of a two-layer perceptron will usually fall into three or more distinct classes, depending on the structure of the inputs. For a good regularisation performance, he suggests the use of different hyperparameters α for these different classes.

3.2.3 Gaussian Approximation to the Posterior Distribution

Although (3.21) is an exact equation for a given noise model and prior, it is useful to approximate the posterior with a Gaussian distribution in order to make it analytically tracktable when used in integrals such as (3.8) (Bishop, 1995, Section 10.1.7). This can be obtained by performing a second-order Taylor expansion of the total error function (3.22) around its minimum $\boldsymbol{w}^*$ (i.e. the maximum of $p(\boldsymbol{w}|D)$):

$$E(\boldsymbol{w}) = E(\boldsymbol{w}^*) + \boldsymbol{b}^T \cdot \Delta\boldsymbol{w} + \frac{1}{2}\Delta\boldsymbol{w}^T \cdot \tilde{\boldsymbol{H}} \cdot \Delta\boldsymbol{w}, \qquad (3.25)$$

where $\Delta\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{w}^*$. $\boldsymbol{b}$ denotes the gradient of E at $\boldsymbol{w}^*$,

$$\boldsymbol{b} = \nabla E(\boldsymbol{w})|_{\boldsymbol{w}=\boldsymbol{w}^*} = \boldsymbol{0}, \qquad (3.26)$$

which vanishes because $\boldsymbol{w}^*$ marks the minimum of E. $\tilde{\boldsymbol{H}}$ is the Hessian matrix of E (second derivative) with respect to the weights, and it will play an important role in the remaining parts of this thesis. Looking at (3.22), we can see that $\tilde{\boldsymbol{H}}$ is composed of two parts, the *data Hessian* $\boldsymbol{H_D}$ and the weights hyperparameter $\boldsymbol{A_r}$[32]:

$$\begin{aligned} \tilde{\boldsymbol{H}} &= \nabla\nabla E(\boldsymbol{w})|_{\boldsymbol{w}=\boldsymbol{w}^*} = \nabla\nabla E_D(\boldsymbol{w})|_{\boldsymbol{w}=\boldsymbol{w}^*} + \boldsymbol{A_r} && (3.27)\\ &= \boldsymbol{H_D} + \boldsymbol{A_r}. && (3.28) \end{aligned}$$

Using the approximated error function (3.25), (3.21) becomes

$$p(\boldsymbol{w}|D) = \frac{1}{Z}\exp\left(-E(\boldsymbol{w}*) - \frac{1}{2}\Delta\boldsymbol{w}^T \cdot \tilde{\boldsymbol{H}} \cdot \Delta\boldsymbol{w}\right), \qquad (3.29)$$

and, including the constant $\exp\left(-E(\boldsymbol{w}*)\right)$ in the normalisation factor Z,

$$p(\boldsymbol{w}|D) = \frac{1}{Z}\exp\left(-\frac{1}{2}\Delta\boldsymbol{w}^T \cdot \tilde{\boldsymbol{H}} \cdot \Delta\boldsymbol{w}\right). \qquad (3.30)$$

The posterior weights distribution thus becomes a Gaussian with mean $\boldsymbol{w}^*$ and covariance matrix $\tilde{\boldsymbol{H}}^{-1}$. The information contained in this covariance matrix can be immediately used to give error bars on the most probable weights vector $\boldsymbol{w}^*$, an example of which is shown in Figure 3.3.

[32] Note that if we use no prior information about the weights (i.e. $p(\boldsymbol{w}) = const.$), $E \equiv E_D$ and $\tilde{\boldsymbol{H}} = \boldsymbol{H_D}$. This case corresponds to ANN training without regularisation.

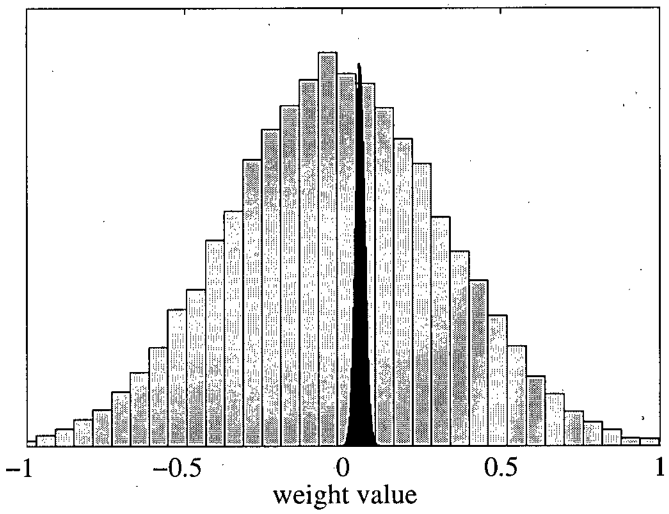


Figure 3.3: Distribution of the first weight w_{11} of the neural network used for the example given by equations (3.9) and (3.10) after a short training run (grey histogram, the network has not gained much certainty about the weight value yet) and after a longer training run (black histogram, the distribution has narrowed considerably).

3.3 Output Uncertainties

In the previous section, both terms under the integral in (3.8) – the noise model of the target variables (3.15) and the Gaussian approximation to the posterior weights distribution (3.30) – have been derived. Using these results, the derivation of the distribution of the network outputs is straightforward (Aires et al., 2004a):

$$p(\boldsymbol{t}|\boldsymbol{x}, D) = \int p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w})p(\boldsymbol{w}|D)d\boldsymbol{w} \tag{3.31}$$

$$\propto \int \exp\left(-\frac{1}{2}(\boldsymbol{t}-\boldsymbol{y})^T \cdot \boldsymbol{A_{in}} \cdot (\boldsymbol{t}-\boldsymbol{y})\right) \cdot \exp\left(-\frac{1}{2}\Delta\boldsymbol{w}^T \cdot \tilde{\boldsymbol{H}} \cdot \Delta\boldsymbol{w}\right) d\boldsymbol{w}. \tag{3.32}$$

All normalisation factors have been omitted in the notation in (3.32), instead, the $\cdot \propto \cdot$ sign has been used to indicate the missing normalisation. This integral can be evaluated by assuming that the posterior distribution of the weights, (3.30), is narrow enough to approximate the network function $\boldsymbol{y}(\boldsymbol{x};\boldsymbol{w})$ by its linear expansion around the optimal weights value $\boldsymbol{w}^*$:

$$\boldsymbol{y}(\boldsymbol{x};\boldsymbol{w}) = \boldsymbol{y}(\boldsymbol{x};\boldsymbol{w}^*) + \boldsymbol{G}^T\Delta\boldsymbol{w}, \tag{3.33}$$

where the $W \times c$ matrix $\boldsymbol{G}$ represents the gradient of the network function (c is the number of network outputs):

$$\boldsymbol{G} = \nabla \boldsymbol{y}(\boldsymbol{x}; \boldsymbol{w})|_{\boldsymbol{w}=\boldsymbol{w}^*}. \tag{3.34}$$

Hence, (3.32) becomes

$$p(\boldsymbol{t}|\boldsymbol{x}, D) \propto \int \exp\left(-\frac{1}{2}\left(\boldsymbol{t} - \boldsymbol{y}(\boldsymbol{x}; \boldsymbol{w}^*) - \boldsymbol{G}^T\Delta\boldsymbol{w}\right)^T \cdot \boldsymbol{A_{in}} \cdot \left(\boldsymbol{t} - \boldsymbol{y}(\boldsymbol{x}; \boldsymbol{w}^*) - \boldsymbol{G}^T\Delta\boldsymbol{w}\right)\right) \cdot \\ \exp\left(-\frac{1}{2}\Delta\boldsymbol{w}^T \cdot \tilde{\boldsymbol{H}} \cdot \Delta\boldsymbol{w}\right) d\boldsymbol{w}. \tag{3.35}$$

Writing $\boldsymbol{\epsilon}^* = \boldsymbol{t} - \boldsymbol{y}(\boldsymbol{x}; \boldsymbol{w}^*)$ for simplicity, expanding the product and rearranging yields

$$p(\boldsymbol{t}|\boldsymbol{x}, D) \propto \exp\left(-\frac{1}{2}\boldsymbol{\epsilon}^{*T} \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{\epsilon}^*\right) \cdot \\ \int \exp\left(\boldsymbol{\epsilon}^{*T} \cdot \boldsymbol{A_{in}} \cdot (\boldsymbol{G}^T\Delta\boldsymbol{w}) - \frac{1}{2}(\boldsymbol{G}^T\Delta\boldsymbol{w})^T \cdot \boldsymbol{A_{in}} \cdot (\boldsymbol{G}^T\Delta\boldsymbol{w}) - \frac{1}{2}\Delta\boldsymbol{w}^T \cdot \tilde{\boldsymbol{H}} \cdot \Delta\boldsymbol{w}\right) d\boldsymbol{w}, \tag{3.36}$$

where the first factor is independent of $\boldsymbol{w}$ and has hence been pulled out of the integral. Using the matrix identity $(\boldsymbol{AB})^T = \boldsymbol{B}^T\boldsymbol{A}^T$ and further rearranging leads to

$$p(\boldsymbol{t}|\boldsymbol{x}, D) \propto \exp\left(-\frac{1}{2}\boldsymbol{\epsilon}^{*T} \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{\epsilon}^*\right) \cdot \\ \int \exp\left(\left(\boldsymbol{\epsilon}^{*T} \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{G}^T\right)\Delta\boldsymbol{w} - \frac{1}{2}\Delta\boldsymbol{w}^T\left(\boldsymbol{G} \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{G}^T + \tilde{\boldsymbol{H}}\right)\Delta\boldsymbol{w}\right) d\boldsymbol{w}. \tag{3.37}$$

Bishop (1995, Appendix B) shows that Gaussian integrals with a linear term evaluate to

$$\int \exp\left(\boldsymbol{L}^T\boldsymbol{w} - \frac{1}{2}\boldsymbol{w}^T \cdot \boldsymbol{A} \cdot \boldsymbol{w}\right) d\boldsymbol{w} = (2\pi)^{W/2}\,|\boldsymbol{A}|^{-1/2} \exp\left(\frac{1}{2}\boldsymbol{L}^T \cdot \boldsymbol{A} \cdot \boldsymbol{L}\right). \tag{3.38}$$

Setting $\boldsymbol{L} = \boldsymbol{\epsilon}^{*T} \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{G}^T$ and $\boldsymbol{A} = \boldsymbol{G} \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{G}^T + \tilde{\boldsymbol{H}}$, the integral in (3.37) becomes

$$\int \exp(\ldots)\, d\boldsymbol{w} = (2\pi)^{W/2}\left|\boldsymbol{G} \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{G}^T + \tilde{\boldsymbol{H}}\right|^{-1/2} \cdot \\ \exp\left(\frac{1}{2}\left(\boldsymbol{\epsilon}^{*T} \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{G}^T\right)^T \cdot \left(\boldsymbol{G} \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{G}^T + \tilde{\boldsymbol{H}}\right) \cdot \left(\boldsymbol{\epsilon}^{*T} \cdot \boldsymbol{A_{in}} \cdot \boldsymbol{G}^T\right)\right). \tag{3.39}$$

Omitting the constant factor and rearranging (3.37) again gives

$$p(\boldsymbol{t}|\boldsymbol{x}, D) \propto \exp\left(-\frac{1}{2}\boldsymbol{\epsilon}^{*T} \cdot \boldsymbol{A}_{in} \cdot \boldsymbol{\epsilon}^{*}\right) \exp\left(-\frac{1}{2}\boldsymbol{\epsilon}^{*T}\left[-\boldsymbol{A}_{in}\boldsymbol{G}^{T} \cdot \left(\boldsymbol{G} \cdot \boldsymbol{A}_{in} \cdot \boldsymbol{G}^{T} + \tilde{\boldsymbol{H}}\right)^{-1} \cdot \boldsymbol{G}\boldsymbol{A}_{in}\right]\boldsymbol{\epsilon}^{*}\right) \tag{3.40}$$

and further

$$p(\boldsymbol{t}|\boldsymbol{x}, D) \propto$$
$$\exp\left(-\frac{1}{2}\left(\boldsymbol{t} - \boldsymbol{y}(\boldsymbol{x}; \boldsymbol{w}^{*})\right)^{T}\left[\boldsymbol{A}_{in} - \boldsymbol{A}_{in}\boldsymbol{G}^{T} \cdot \left(\boldsymbol{G} \cdot \boldsymbol{A}_{in} \cdot \boldsymbol{G}^{T} + \tilde{\boldsymbol{H}}\right)^{-1} \cdot \boldsymbol{G}\boldsymbol{A}_{in}\right]\left(\boldsymbol{t} - \boldsymbol{y}(\boldsymbol{x}; \boldsymbol{w}^{*})\right)\right). \tag{3.41}$$

Thus, the target variables follow a multivariate Gaussian distribution with mean $\boldsymbol{y}(\boldsymbol{x}; \boldsymbol{w}^{*})$ and covariance matrix

$$\boldsymbol{C}_{0} = \left[\boldsymbol{A}_{in} - \boldsymbol{A}_{in}\boldsymbol{G}^{T} \cdot \left(\boldsymbol{G} \cdot \boldsymbol{A}_{in} \cdot \boldsymbol{G}^{T} + \tilde{\boldsymbol{H}}\right)^{-1} \cdot \boldsymbol{G}\boldsymbol{A}_{in}\right]^{-1}. \tag{3.42}$$

The expression for the covariance matrix can be simplified by multiplying by $\left[\boldsymbol{G}\left(\boldsymbol{I} + \tilde{\boldsymbol{H}}^{-1}\boldsymbol{G}\boldsymbol{A}_{in}\boldsymbol{G}^{T}\right)\boldsymbol{G}\right] \times$ $\ldots \times \left[\boldsymbol{G}\left(\boldsymbol{I} + \tilde{\boldsymbol{H}}^{-1}\boldsymbol{G}\boldsymbol{A}_{in}\boldsymbol{G}^{T}\right)\boldsymbol{G}\right]^{-1}$ to give

$$\boldsymbol{C}_{0} = \boldsymbol{C}_{in} + \boldsymbol{G}^{T}\tilde{\boldsymbol{H}}^{-1}\boldsymbol{G}, \tag{3.43}$$

where $\boldsymbol{C}_{in} = \boldsymbol{A}_{in}{}^{-1}$ has been used.

Equation (3.43) is the main result of this derivation (Aires et al., 2004a). It shows that the uncertainty in the neural network predictions is composed of two parts; the intrinsic noise contained in the training data, represented by its covariance matrix $\boldsymbol{C}_{in}$, and a term $\boldsymbol{G}^{T}\tilde{\boldsymbol{H}}^{-1}\boldsymbol{G}$ that represents the impact of the uncertainty of the posterior weights distribution on the predictions – a result that is expected from (3.8). Unless we have situation dependent (= input dependent) information about intrinsic noise on the data, $\boldsymbol{C}_{in}$ will be constant. The neural prediction term, however, is situation dependent through the gradient $\boldsymbol{G}$ (the Hessian is not dependent on the input data).

We can determine the error bars on a network prediction by taking the standard deviation from the covariance matrix $\boldsymbol{C}_{0}$. Figures 3.4, 3.5 and 3.6 illustrate the results that can be obtained for the example defined at the end of section 3.1. Williams et al. (1995) show that the weights uncertainty term $\boldsymbol{G}^{T}\tilde{\boldsymbol{H}}^{-1}\boldsymbol{G}$ is approximately proportional to the inverse training data density[33], and note that consequently in high-

[33] They can prove the result for generalised linear regression models (models of the form $y(\boldsymbol{x}) = \sum_{j=1}^{m} w_j \phi_j(\boldsymbol{x})$, where ϕ_j are basis functions), and note that empirical studies provide evidence that the result also holds for multi-layer networks – especially for networks with linear output activations trained with a least-squares error function (as in our case), which "is effectively a generalised linear regression model with adaptive basis functions" (Williams et al., 1995, Section 5).

data-density regions, the contribution of this term will become insignificant compared to the noise term C_0 – a result that I will discuss in section 3.7.

3.4 Hyperparameter Re-Estimation

With the results from the previous sections, we have to know the values of the hyperparameters $\boldsymbol{A_{in}}$ and $\boldsymbol{A_r}$ in advance. However, using the complete structure of $\boldsymbol{A_r}$ in the training process requires a good knowledge of the network mapping, which usually is not available a priori. Similarly, information about the noise distribution in the target data will also not be available in most cases. Therefore, how can the hyperparameters be estimated? Aires et al. (2004a) suggest the following – omit both hyperparameters in an initial learning stage, then estimate them from the trained network and re-train the network with the new values.

In the case of $\boldsymbol{A_{in}}$, Aires et al. suggest use of (3.43). The approximation they make is to assume the intrinsic noise to be constant throughout the training data. Then an average covariance matrix $\boldsymbol{C_{in}}$ can be computed from the training dataset D by determining the covariance matrix of the output errors C_0 from the $(\epsilon^*)^n$ of the training datapoints, and by using an average of the neural prediction term $\boldsymbol{G}^T\tilde{\boldsymbol{H}}^{-1}\boldsymbol{G}$ over the training dataset. From these two matrices, an average $\boldsymbol{C_{in}}$ can be computed, and $\boldsymbol{A_{in}}$ is obtained by inversion:

$$\boldsymbol{C_{in}} = \langle C_0 \rangle - \langle \boldsymbol{G}^T\tilde{\boldsymbol{H}}^{-1}\boldsymbol{G} \rangle, \tag{3.44}$$

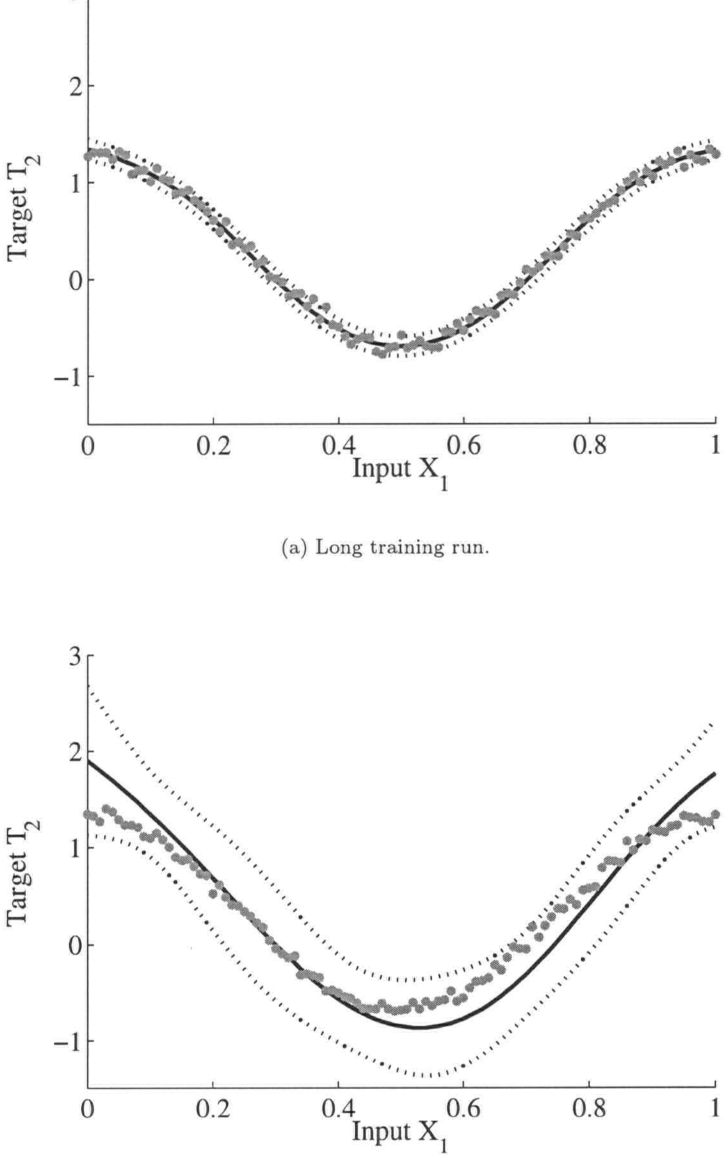
where the averages have been denoted by $\langle\cdot\rangle$.

Unfortunately, Aires et al. do not give details on how to re-estimate $\boldsymbol{A_r}$. Their papers suggest use of the posterior weights distribution of one training run as the prior distribution in the consecutive run. This, however, would destroy the intended regularisation mechanism – the prior (3.19) is intentionally chosen to be a Gaussian with zero mean in order to keep the weight values small. The posterior (3.30), in contrast, is a Gaussian with mean $\boldsymbol{w}^* \neq 0$. Thus, using this distribution as a prior for the consecutive training run would pull the weights towards the already found minimum, not encourage them to be small[34]. This also becomes clear when considering the following:

In order to accommodate the mean $\boldsymbol{w}^*$, (3.20) would have to be changed to

$$E_W(\boldsymbol{w})^{(i)} = \frac{1}{2}\left(\boldsymbol{w} - (\boldsymbol{w}^*)^{(i-1)}\right)^T \cdot \boldsymbol{A_r}^{(i-1)} \cdot \left(\boldsymbol{w} - (\boldsymbol{w}^*)^{(i-1)}\right), \tag{3.45}$$

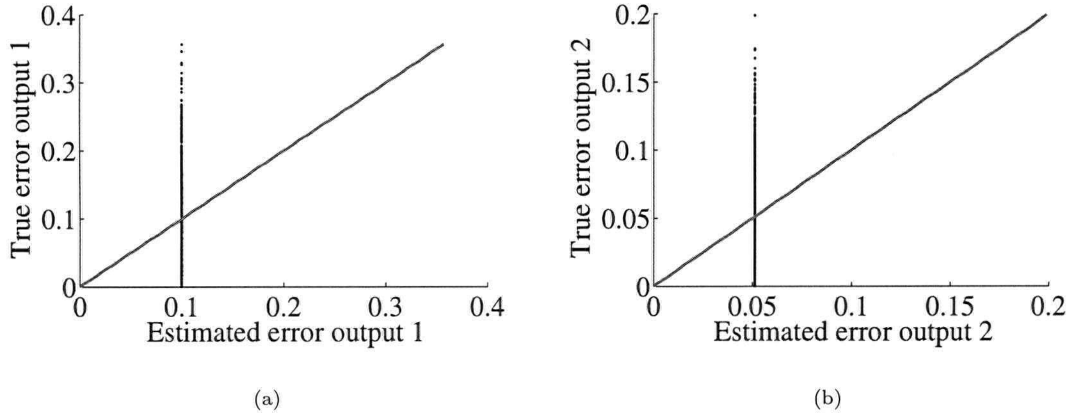
[34] Although MacKay (1992b) points out that using weight decays with non-zero means would just correspond to using a different model – whose performance could be compared to a zero-mean regulariser within the evidence framework (MacKay, 1992a).



(a) Long training run.

(b) Short training run.

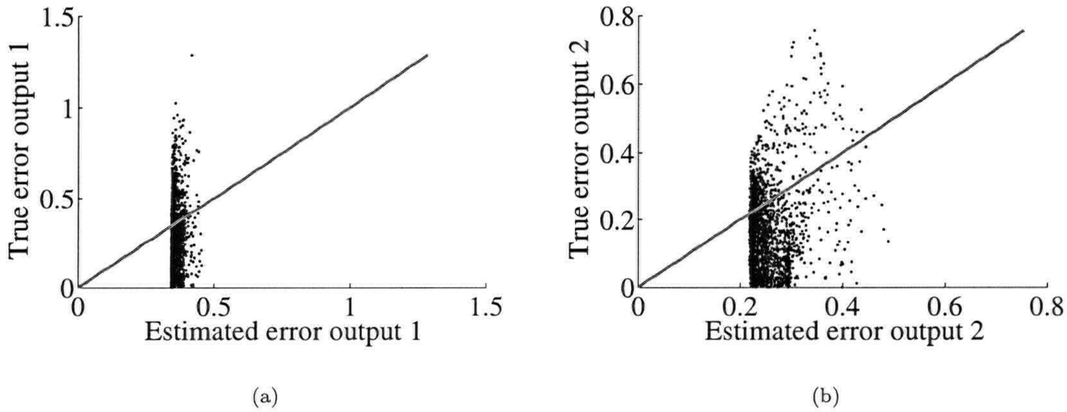
Figure 3.4: Section through the functional surface of the example given by (3.9) and (3.10) at $X_2 = 0.3$. Plotted are datapoints from the training dataset (grey), the network prediction (solid line) and error bars (dotted lines). Panel (a) shows the error prediction after the long training run (the narrow weight distribution in Figure 3.3), while panel (b) displays the same prediction for the network trained only a short time (the broad distribution in Figure 3.3). The broader weights distribution is noticeably reflected in larger error bars.



(a)

(b)

Figure 3.5: Scatterplots of the estimated error (one standard deviation) vs. the actual error for both output variables of the example given in (3.9) and (3.10), for the long training run shown in Figure 3.8 (also see Figure 3.4a). The narrow weights distribution (cmp. Figure 3.3) causes the network term in (3.43) to be much smaller than the intrinsic noise C_{in}. Consequently, the estimated error is almost identical to the noise on the data, which is correctly estimated to have a standard deviation of 0.1 and 0.05, respectively. Hence, the predicted error does not show significant input dependence, it can only be stated that the true error will be smaller than the estimated error in at least 68.2% of all cases. In fact, for the given example, 68.5% of the true errors are smaller than the predicted ones for output variable 1 (68.0% for variable 2).



(a)

(b)

Figure 3.6: The same as Figure 3.5, but for the short training run leading to the broad weights distribution in Figure 3.3 and the larger error bars in Figure 3.4b. Note that this time, the network dependent term in (3.43) is much larger, due to the broader weights distribution, resulting in a more input dependent error (in panel (a) 67.5% of the predicted errors are smaller than the actual error; in panel (b) 75.4%).

where $\cdot^{(i)}$ refers to the i-th re-estimation iteration. Since $\nabla\nabla E_W(\boldsymbol{w})|_{\boldsymbol{w}=(\boldsymbol{w}^*)^{(i)}} = \boldsymbol{A_r}^{(i-1)}$ does not change when (3.45) is used instead of (3.20), $\boldsymbol{A_r}$ is re-estimated following

$$\boldsymbol{A_r}^{(i)} = \tilde{\boldsymbol{H}}^{(i)} = \boldsymbol{H_D}^{(i)} + \boldsymbol{A_r}^{(i-1)}. \qquad (3.46)$$

Since $\tilde{\boldsymbol{H}}$, $\boldsymbol{H_D}$ and $\boldsymbol{A_r}$ are the inverses of covariance matrices, which are positive definite (Aires, 2004), they are also positive definite. This means that at least the diagonal elements of these matrices are positive (Weisstein, 2007). Hence, the diagonal elements of $\boldsymbol{A_r}$ would become larger and larger with each training iteration.

This makes sense; larger diagonal elements of $\boldsymbol{A_r}$ approximately[35] mean a smaller variance of the weights, hence the weights distribution is more sharply peaked. If the optimisation algorithm has found a minimum $\boldsymbol{w}^*$ in a first training run, and in a second training run is "told" that this minimum is very likely (even if it was not a very good minimum), then it will further increase the certainty in that minimum by decreasing the variance. Furthermore, by increasing the diagonal elements in $\boldsymbol{A_r}$, the importance of the data is decreased, until they eventually are insignificant compared to the weights prior. This, however, will only increase the chance of remaining stuck in a local minimum found in the first training run.

However, since we are using a zero-mean regulariser, we wish to get a new estimation of how strongly the weights should be pulled towards zero, which eventually should lead to a compromise between a mapping that fits the data well and a guard against overfitting. A method to re-estimate an $\boldsymbol{A_r}$ for a zero-mean Gaussian is needed, something that Aires et al. do not derive.

The approach I use is to adapt the re-estimation technique suggested by Aires et al. (2004a) for $\boldsymbol{A_{in}}$, but to use the so-called *evidence procedure* (MacKay, 1992a) in order to re-estimate a modified version of $\boldsymbol{A_r}$. However, in order to use the evidence procedure in the form derived by MacKay, $\boldsymbol{A_r}$ can only contain diagonal elements. Hence, correlations between the weights will be ignored below.

3.4.1 Evidence Procedure for $\boldsymbol{A_r}$.

The evidence procedure has been derived by MacKay (1992a) in order to re-estimate the scalar hyperparameters α and β from the data during the network learning stage. Nabney (2002) shows how the approach can be generalised to multiple α for different groups of weights, up to an individual α for every weight – which corresponds to having a diagonal $\boldsymbol{A_r}$ in (3.20).

When in addition to the network weights the values of the hyperparameters are inferred from the

[35] if no covariances exist exactly

data, the result can be expressed by the joint probability distribution $p(\boldsymbol{w}, \alpha, \beta|D)$. The correct Bayesian treatment (Bishop, 1995, Section 10.4) to get the posterior distribution of the weights $p(\boldsymbol{w}|D)$ from this joint distribution would be to integrate over all possible values of α and β:

$$\begin{aligned} p(\boldsymbol{w}|D) &= \int\int p(\boldsymbol{w}, \alpha, \beta|D)\, d\alpha d\beta && (3.47) \\ &= \int\int p(\boldsymbol{w}|\alpha, \beta, D)\, p(\alpha, \beta|D)\, d\alpha d\beta. && (3.48) \end{aligned}$$

The evidence procedure, however, makes the approximation that the density $p(\alpha, \beta|D)$ is sharply peaked around the most probable values α^{MP} and β^{MP}, reducing (3.48) to

$$p(\boldsymbol{w}|D) = p(\boldsymbol{w}|\alpha^{MP}, \beta^{MP}, D) \underbrace{\int\int p(\alpha, \beta|D)\, d\alpha d\beta}_{=1}. \quad (3.49)$$

In order to find α^{MP} and β^{MP}, the posterior distribution

$$p(\alpha, \beta|D) = \frac{p(D|\alpha, \beta)\, p(\alpha, \beta)}{p(D)} \quad (3.50)$$

has to be maximised. The prior $p(\alpha, \beta)$ is assumed to be uniform in the evidence procedure, so that it does not affect the maximum of $p(\alpha, \beta|D)$[36]. The normalisation factor $p(D)$ (the integral of the numerator over α and β) also does not affect the maximum, hence only $p(D|\alpha, \beta)$ – known as the *evidence* of the hyperparameters – has to be maximised.

This term can be written as an integral of the data likelihood over all possible weights w:

$$p(D|\alpha, \beta) = \int p(D|\boldsymbol{w}, \alpha, \beta)\, p(\boldsymbol{w}|\alpha, \beta)\, d\boldsymbol{w}. \quad (3.51)$$

Since the hyperparameters are given, the factors $p(D|\boldsymbol{w}, \alpha, \beta)$ and $p(\boldsymbol{w}|\alpha, \beta)$ of the integrand are given by (3.17) and (3.19) (exchange the matrix hyperparameters with the scalar ones), leading to

$$p(D|\alpha, \beta) = \frac{1}{Z} \int \exp\left(-E^{\alpha\beta}(\boldsymbol{w})\right)\, d\boldsymbol{w}, \quad (3.52)$$

where (3.24) has been used because of the scalar hyperparameters. The Gaussian integral in (3.52) is

[36] Such a prior is said to be an *improper* prior, since it does not have a finite integral and cannot be normalised.

given by

$$\int \exp\left(-E^{\alpha\beta}(\boldsymbol{w})\right)\, d\boldsymbol{w} = \exp\left(-E^{\alpha\beta}(\boldsymbol{w}^*)\right)\, (2\pi)^{W/2} \left|\tilde{\boldsymbol{H}}\right|^{-1/2} \tag{3.53}$$

(see Bishop (1995, Appendix B) for the evaluation of Gaussian integrals). Together with terms from the normalisation factor Z, the logarithm of the evidence (3.52) is then given by

$$\log p(D|\alpha,\beta) = -\frac{\alpha}{2}\sum_{i=1}^{W}(w_i^*)^2 - \frac{\beta}{2}\sum_{n=1}^{N}\{y(\boldsymbol{x}^n;\boldsymbol{w}) - t^n\}^2 - \frac{1}{2}\log\left|\tilde{\boldsymbol{H}}\right| + \frac{W}{2}\log\alpha + \frac{N}{2}\log\beta - \frac{N}{2}\log(2\pi). \tag{3.54}$$

In order to optimise this log evidence with respect to α (the corresponding approach for β will not be considered here), the partial derivative $\frac{\partial}{\partial\alpha}(\log p(D|\alpha,\beta))$ has to be computed. Bishop (1995, p. 410) shows that

$$\frac{\partial}{\partial\alpha}\log\left|\tilde{\boldsymbol{H}}\right| = \operatorname{tr}\left(\tilde{\boldsymbol{H}}^{-1}\right), \tag{3.55}$$

where the approximation has been made that the eigenvalues of $\tilde{\boldsymbol{H}}$ do not depend on α. (I have skipped the eigenvalue step here, see Bishop (1995, Section 10.4) for further details.) Hence,

$$\frac{\partial}{\partial\alpha}\left(\log p(D|\alpha,\beta)\right) = -\frac{1}{2}\sum_{i=1}^{W}(w_i^*)^2 - \frac{1}{2}\operatorname{tr}\left(\tilde{\boldsymbol{H}}^{-1}\right) + \frac{W}{2\alpha}. \tag{3.56}$$

Equating (3.56) to zero yields the expression

$$\alpha = \frac{W - \alpha\operatorname{tr}\left(\tilde{\boldsymbol{H}}^{-1}\right)}{\sum_{i=1}^{W}(w_i^*)^2}. \tag{3.57}$$

If groups of weights are assigned different hyperparameters, this equation can be adjusted correspondingly, down to an individual α for each weight:

$$\alpha_i = \frac{1 - \alpha_i\left(\tilde{\boldsymbol{H}}^{-1}\right)_{ii}}{(w_i^*)^2}. \tag{3.58}$$

In order to optimise α, the ANN training is first started with an initial (random) value of α. Once the training algorithm has found a minimum, α is re-estimated from (3.58):

$$\alpha_i^{new} = \frac{1 - \alpha_i^{old}\left(\tilde{\boldsymbol{H}}^{-1}\right)_{ii}}{(w_i^*)^2}. \tag{3.59}$$

Of course, the optimisation of the hyperparameters by iterative re-estimation – valid for both re-estimating $\boldsymbol{A_{in}}$ with the Aires et al. (2004a) approach and $\boldsymbol{A_r}$ with the evidence procedure – is computationally expensive, since the network has to be re-trained several times, ideally until the hyperparameters stabilise. Figure 3.7 illustrates how they develop during the long training run of the example used in this chapter.

3.5 The Jacobian Matrix

The Jacobian matrix is not a part of the Bayesian framework described in the previous sections. It represents the first derivative of the network outputs $\boldsymbol{y}$ with respect to the inputs $\boldsymbol{x}$ and is defined as

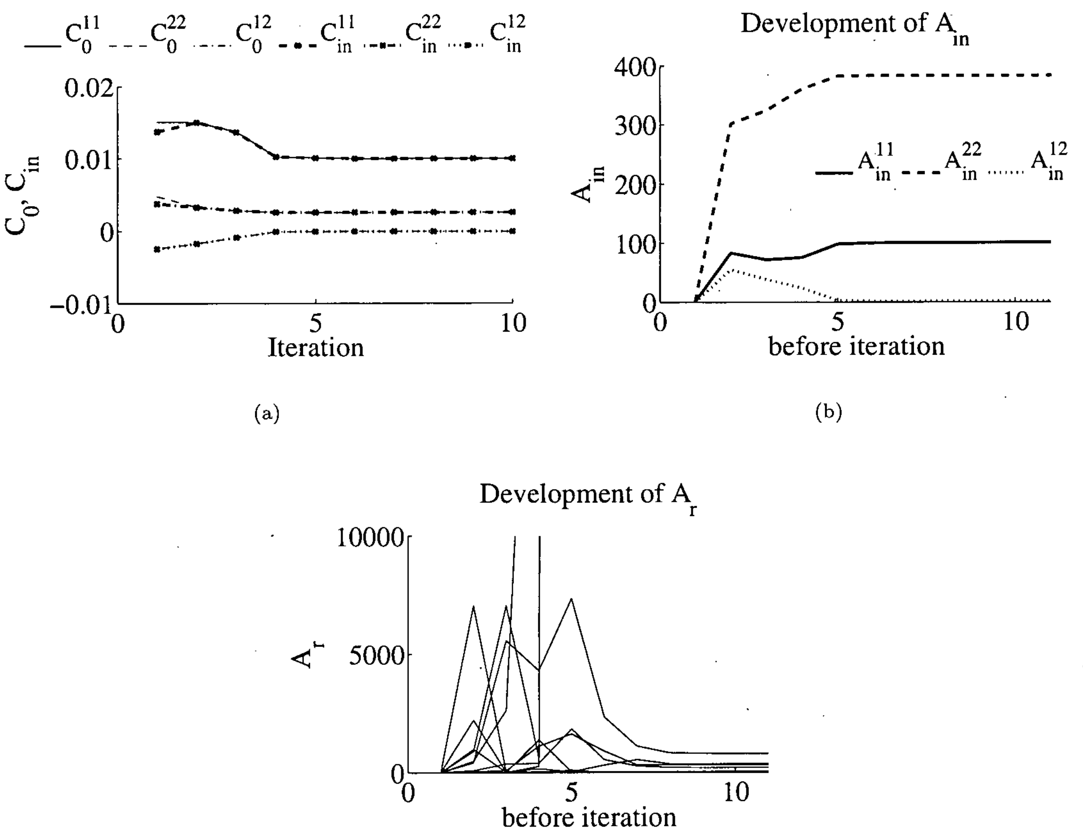
$$J_{ik} = \frac{\partial y_k}{\partial x_i}. \tag{3.60}$$

Aires et al. (2004b) suggest that the variability in the Jacobian can be determined by computing a distribution of Jacobians from the posterior weights distribution given by (3.30). Their approach, and the one adopted in this study, is to use Monte Carlo techniques to sample from the weights distribution, then to construct a Jacobian distribution from these samples. Aires et al. use $R = 1000$ samples from (3.30), compute the Jacobian for each weights sample, and compute mean and variance from all these Jacobians. This, of course, assumes a Gaussian distribution for the Jacobian as well, although the histogram of the Jacobians itself could be used as a PDF (probability density function) representation.

The Jacobian, as defined in (3.60), is input dependent, hence its distribution can be computed for individual inputs.[37] For the purpose of identifying ill-posed problems Aires et al. (2004b) propose to compute an average Jacobian over the training dataset and to interpret its variability. This approach applied to the example given by (3.9) and (3.10) yields the results listed in Table 3.1. Indeed, the wider weights distribution results in a larger variability in the Jacobian.

If a problem has been identified as ill-posed, they suggest a principal component decomposition of the input and output data. They apply this approach to a remote sensing problem with many correlated inputs and use it to reduce the number of inputs and to exploit the decorrelated structure of the principal components. However, if the number of inputs is small from the beginning, it is usually not desired to further decrease their number. Bishop (1995, Section 8.2) discusses a similar approach, which decorrelates the inputs and is known as *whitening* in the literature. It will become important in chapter 4.

[37] A property that will prove useful in chapter 4, where the Jacobian provides information about whether a network is modelling the "right" function.



(a)

(b)

(c)

Figure 3.7: Development of the covariance matrices $\boldsymbol{C_0}$ and $\boldsymbol{C_{in}}$ (panel (a)) and the hyperparameters $\boldsymbol{A_{in}}$ (panel (b)) and $\boldsymbol{A_r}$ (panel (c)) during the training run shown in Figure 3.8. After a few re-estimation iterations, the estimated intrinsic noise stabilises at the correct variance values of 0.01 and 0.0025, respectively, leading to a stabilisation in the data hyperparameter $\boldsymbol{A_{in}}$. Notable is that the uncertainty in the neural prediction, given by the difference between $\boldsymbol{C_0}$ and $\boldsymbol{C_{in}}$, quickly becomes smaller. Most elements of $\boldsymbol{A_r}$ stay in the range between 0 and 100, however, some weights are more strongly penalised for getting large. After approximately eight iterations, $\boldsymbol{A_r}$ has stabilised as well.

Table 3.1: Jacobian matrices and their uncertainty (one standard deviation) of the two networks trained to model the example function given by (3.9) and (3.10). Left: the Jacobian of the network with the narrow weights distribution (long training run) shown in Figure 3.3; right: the Jacobian of the network with the broad weights distribution (short training run) in Figure 3.3. Note how the width of the weights distribution is reflected in the uncertainty of the Jacobian, and how the only shortly trained network models a significantly different dependence of output 1 to input 2 – a dependence that is also most uncertain in the long trained network and indicates a problematic input.

	$\partial y_1/$	$\partial y_2/$
∂x_1	4.00 ± 0.02	-0.04 ± 0.01
∂x_2	-0.02 ± 0.43	1.00 ± 0.05

	$\partial y_1/$	$\partial y_2/$
∂x_1	3.83 ± 0.50	0.01 ± 0.87
∂x_2	-1.52 ± 0.37	0.77 ± 0.34

3.6 Implementation Issues

The NETLAB toolbox by Nabney (2002) is implemented in MATLAB and provides many functions for neural networks. The toolbox comes with the ability to handle two-layer MLP with a Bayesian module that implements the scalar hyperparameter approach of MacKay (1992b). Since NETLAB is very well documented and already provides many functions needed for the Aires et al. algorithm, I chose it as a basis for my implementation.

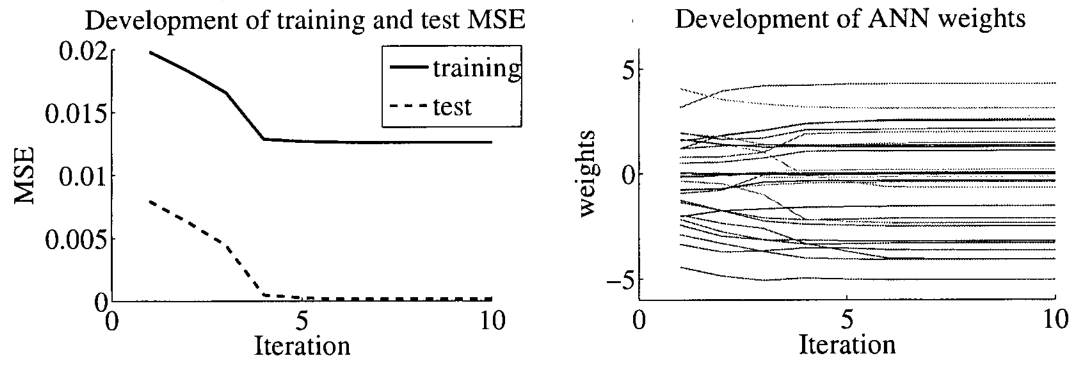
Algorithm 3.1 below summarises the algorithm discussed in the previous sections in a schematic outline. The initial hyperparameters $\boldsymbol{A_{in}}^{(0)}$ and $\boldsymbol{A_r}^{(0)}$ will usually be set to the identity matrix $\boldsymbol{I}$ and $\alpha^0\boldsymbol{I}$, respectively, where α^0 is some constant that controls the degree of regularisation in the first training run. If no regularisation is desired, $\boldsymbol{A_r}$ can be set to zero. The loop in lines 2 to 9 trains the network and re-estimates the hyperparameters. It is repeated until some termination criterion has been reached, which can be either a stabilisation of the hyperparameters or simply a maximum number of iterations.

In line 3, the error surface of equation (3.22) is minimised with one of the optimisation algorithms provided by NETLAB, for instance, conjugate gradients. Afterwards all input data is propagated through the network, and the covariance matrix $\boldsymbol{C_0}$ is computed from the errors of the predictions compared to the targets. In the following step, the gradient of the outputs with respect to the weights $\boldsymbol{G}$ and the Hessian $\tilde{\boldsymbol{H}}$ is computed, which together with $\boldsymbol{C_0}$ are used to estimate $\boldsymbol{C_{in}}$ and $\boldsymbol{A_{in}}$. Eventually, $\boldsymbol{A_r}$ is re-estimated with the evidence procedure described in section 3.4.

While details about the implementation are given in Appendix A, a few important issues are mentioned in subsections 3.6.1 and 3.6.2.

Algorithm 3.1: Bayesian Neural Network Training with Matrix Hyperparameter Re-Estimation.

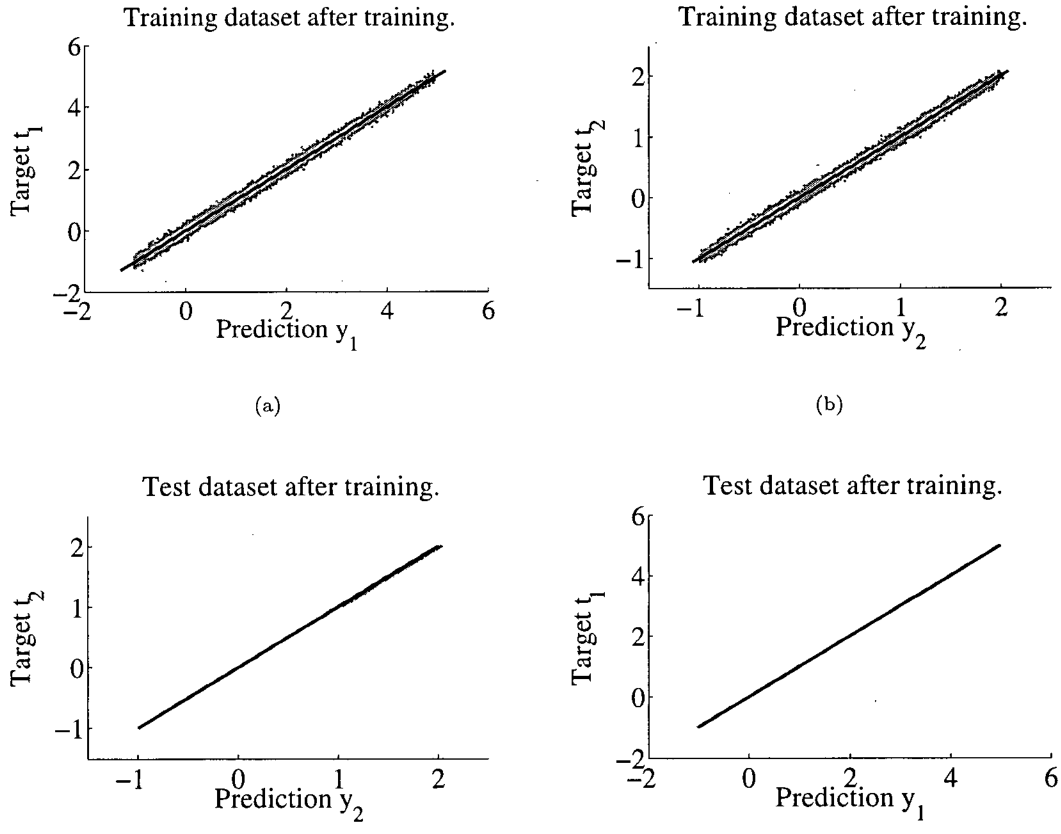
```
Input: Training dataset $\mathcal{S} = \{X, D\}$, initial hyperparameters ${\boldsymbol{A}_{in}}^{(0)}$, ${\boldsymbol{A}_r}^{(0)}$.
Result: Maximum of posterior weights distribution $\boldsymbol{w}^*$, hyperparamaters $\boldsymbol{A}_{in}$ and $\boldsymbol{A}_r$, covariance
        matrix of intrinsic noise $\boldsymbol{C}_{in}$ and covariance matrix of posterior weights distribution $\tilde{\boldsymbol{H}}^{-1}$.
$\boldsymbol{A}_r \leftarrow {\boldsymbol{A}_r}^{(0)}$; $\boldsymbol{A}_{in} \leftarrow {\boldsymbol{A}_{in}}^{(0)}$;
repeat
    Train network (minimise error function (3.22)) with $\boldsymbol{A}_{in}$ and $\boldsymbol{A}_r$ in order to find $\boldsymbol{w}^*$;
    Estimate covariance matrix $\boldsymbol{C}_0 = covariance(\boldsymbol{t}(\boldsymbol{x}) - \boldsymbol{y}(\boldsymbol{x}; \boldsymbol{w}^*))$ from target data and network
    predictions;
    Compute gradient $\boldsymbol{G} = \nabla \boldsymbol{y}(\boldsymbol{x}; \boldsymbol{w})|_{w=w^*}$ (3.34) and Hessian $\tilde{\boldsymbol{H}} = \nabla\nabla E(\boldsymbol{w})|_{w=w^*}$ (3.27);
    Compute the approximate covariance matrix of the intrinsic noise $\boldsymbol{C}_{in} = \langle \boldsymbol{C}_0 \rangle - \langle \boldsymbol{G}^T \tilde{\boldsymbol{H}}^{-1} \boldsymbol{G} \rangle$
    (3.44);
    Set $\boldsymbol{A}_{in} = {\boldsymbol{C}_{in}}^{-1}$;
    Re-estimate $\boldsymbol{A}_r$ with the evidence procedure, (3.59);
until hyperparameters have stabilised or maximum number of iterations has been reached ;
```



(a)

(b)

Figure 3.8: (a) Development of the mean-square-error (MSE) of the training dataset and of a test dataset without added noise of the example given by (3.9) and (3.10) during a training run with 10 hyperparameter re-estimation iterations (corresponding to the narrow weight distribution in Figure 3.3). The stabilisation of the test MSE close to zero after iteration 4 is a sign that no overfitting is taking place; in the case of overfitting, the training MSE would further decrease while the test MSE would increase (the network prediction would not model the true function anymore). (b) Development of the weight values during the training. After iteration 6 the values hardly change anymore. The regularisation effect of $\boldsymbol{A}_r$ causes all weights to stay of order unity.



(a)

(b)

(c)

(d)

Figure 3.9: Scatterplot of the network predictions for both outputs of the example given in (3.9) and (3.10) vs. the target data. Panels (a) and (b) show plots of the training data (with noise), panels (c) and (d) show plots of the test data (no noise). Whereas panels (a) and (b) reflect the intrinsic noise in the training data, panels (c) and (d) illustrate the good generalisation performance of the network; if overfitting had occurred during the training process, predictions of the test data would unlikely match the targets.

3.6.1 Normalisation of Input and Output Variables

As Bishop (1995, Section 8.2) points out, rescaling input and output variables is useful if different variables have typical values that differ significantly. In atmospheric remote sensing, for instance, particle size and temperature would have very different value ranges. In such cases, pre-processing can have a significant effect on the generalisation performance of the network; after normalising input and output variables to be of order unity, it is expected that the weights will also be of order unity and hence be small (Bishop, 1995, Chapter 8).

A simple method to achieve similar values of order unity for all variables is to apply a linear rescaling by subtracting the mean of the variable and normalising by its standard deviation:

$$\tilde{v}_i^n = \frac{v_i^n - \overline{v}_i}{\sigma_i}, \tag{3.61}$$

where v can be either input or output variable.

Whitening, the more sophisticated linear rescaling method mentioned in section 3.5, decorrelates the variables in addition to normalising them. With whitening, the rescaled variables $\tilde{\boldsymbol{v}}$ are computed using

$$\tilde{\boldsymbol{v}}^n = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{U}^T\left(\boldsymbol{x}^n - \overline{\boldsymbol{x}}\right) \tag{3.62}$$

(Bishop, 1995, Section 8.2). Here, $\boldsymbol{\Lambda}$ denotes a diagonal matrix containing the eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$ of the variables v_i, and $\boldsymbol{U}$ contains the eigenvectors of $\boldsymbol{\Sigma}$.

3.6.2 Regularisation of the Hessian

While working with the implementation of the Aires et al. algorithm, I encountered a severe difficulty, which has already been discussed by Aires (2004) – the positive definite character of the Hessian. As the covariance matrix of a Gaussian distribution, the Hessian has to be positive definite. This means that all its eigenvalues have to be strictly positive, and that $\boldsymbol{v}^T\boldsymbol{H}\boldsymbol{v} > 0$ for any non-zero vector $\boldsymbol{v}$ (Bishop, 2006). Aires (2004) points out that for the local quadratic approximation (3.30) to be valid, the optimal weights vector $\boldsymbol{w}^*$ must be at a real minimum of the error surface, otherwise the positive definite character is not guaranteed. This is obviously a problem, since every training algorithm can only approximate this minimum. Furthermore, as Aires states, the possibly large size of the Hessian ($W \times W$, with W being the total number of weights in the network) has the consequence that its estimation needs to be done with a large enough dataset, otherwise the eigenvalues could become very small or even negative, also violating the positive definiteness of $\boldsymbol{H}$.

A solution to this problem suggested by Aires (2004) is to add a diagonal regularisation matrix[38] $\lambda \boldsymbol{I}$ to the Hessian, where λ is a small scalar and $\boldsymbol{I}$ is the identity matrix. If λ is large enough, this approach will result in a positive definite matrix. However, the addition will also result in a bias in quantities estimated from the regularised matrix (Rigdon, 1997). Using a combination of criteria to measure condition number and positive definiteness, Aires (2004) obtains $\lambda = 12$ for his example. My own experiments, however, showed that λ needs to assume values larger than $10,000$ in certain cases, so that the original character of the Hessian is considerably altered.

Another possibility, mentioned in passing by Nabney (2002), is to use an eigenvalue decomposition, which, for the given case, is closely related to the *truncated singular value decomposition* (e.g. Hansen, 1994). If the matrix $\tilde{\boldsymbol{H}}$ is not positive definite, then one or more of its eigenvalues will be negative. Nabney (2002, Section 9.4.2) decomposes the data Hessian $\boldsymbol{H_D}$ into its eigenvalues and eigenvectors, sets all negative eigenvalues to zero and reconstructs the matrix:

$$\boldsymbol{H_D} = \boldsymbol{V \Lambda V}^T, \tag{3.63}$$

where $\boldsymbol{V}$ contains the eigenvectors of $\boldsymbol{H_D}$ and $\boldsymbol{\Lambda}$ the modified eigenvalues. If the matrix $\boldsymbol{H_D}$ is at least positive semi-definite (i.e. eigenvalues can also be zero) and Hermitian (which all real symmetric matrices are), the eigenvalue decomposition is equivalent to the singular value decomposition and the eigenvalues coincide with the singular values (Abdi, 2007, Section 2). Since the reconstructed $\boldsymbol{H_D}$ still has zero eigenvalues, Nabney further adds the weights hyperparameter $\boldsymbol{A_r}$ in order to reconstruct the full Hessian $\tilde{\boldsymbol{H}}$. Since in Nabney's case $\boldsymbol{A_r}$ is a diagonal matrix, his procedure is a combination of an eigenvalue decomposition and the method proposed by Aires.

3.7 Usefulness of the Aires et al. Method

The figures given in this chapter so far show that the Aires et al. method yields the expected results for the example given by equations (3.9) and (3.10); the hyperparameters stabilise as expected after several re-estimation iterations (Figure 3.7), the network converges (Figures 3.8 and 3.9), and the network with

[38] Do not confuse the meaning of *regularisation* used here with the meaning of the word used for the weights term in the error function. Regularisation for neural network training denotes adding a weight penalty term to the error function in order to encourage small network weights. Regularisation of the Hessian here describes methods to make the Hessian (and its inverse) positive definite. Note that the term regularisation of matrices is also often used in the context of matrix inversion or the solution of linear systems, in this case a matrix is singular or close to singular and has to be regularised in order to make it invertible with the available numerical precision.

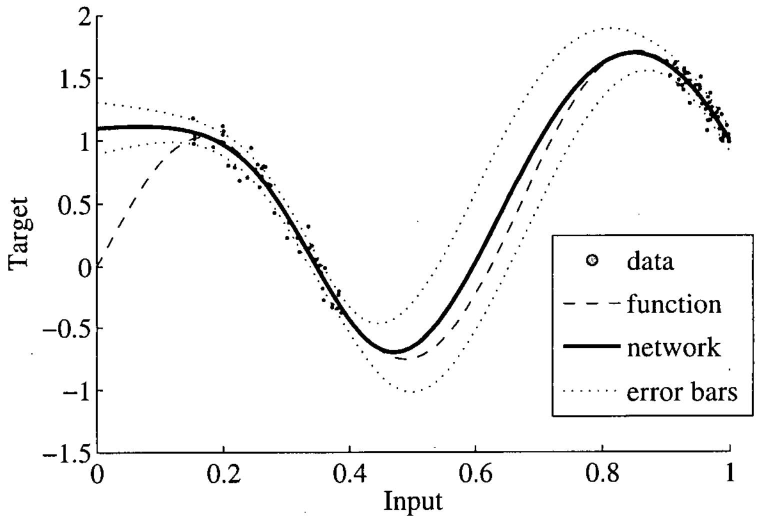


Figure 3.10: Demonstration of the input dependence of the error estimation given in (3.43). Shown is the one-dimensional generator function $y(x) = \sin(3\pi x) + x^2$ (dashed line), from which 100 datapoints were created by adding Gaussian noise with a standard deviation of 0.1 in the intervals [0.15..0.4] and [0.9..1.0]. The network prediction (solid line) is shown together with error bars at one standard deviation. Note the larger error bars in the regions where no training data existed, reflecting the inverse dependence of the estimated error on the training data density. (Example generated with the original NETLAB implementation with scalar hyperparameters.)

the broader weights distribution (Figure 3.3) produces larger error bars (Figure 3.4). However, in many examples, I found the performance of the error estimation to be unsatisfactory, and several problems and open questions require further investigation. Figures (3.10) to (3.14) illustrate some of the problems and failures I encountered while testing my implementation.

A serious problem is the actual size of the error bars. As mentioned in sections 3.3 and 3.4, Aires et al. assume the intrinsic noise on the data to be constant throughout the dataset, and Williams et al. (1995) show that the uncertainty due to the neural network weights is approximately proportional to the inverse data density. Especially in the examples given by Bishop (1995, Figure 10.9) and Nabney (2002, Figure 9.4), the error bars increase significantly in regions where no training data is given. However, I found that for many other functions such good results seem to be difficult to obtain. Furthermore, Williams et al. (1995) found that the neural uncertainty contribution of a network trained from a very dense dataset will become insignificant compared to the noise term. For instance, Figure 3.10 shows a simple example of a network trained from a dataset including 100 datapoints in two distinct intervals. As expected, the error bars in the middle section, where no training data was given, are larger than in the sections where data was available. Consequently, the true generator function is still within the error interval of

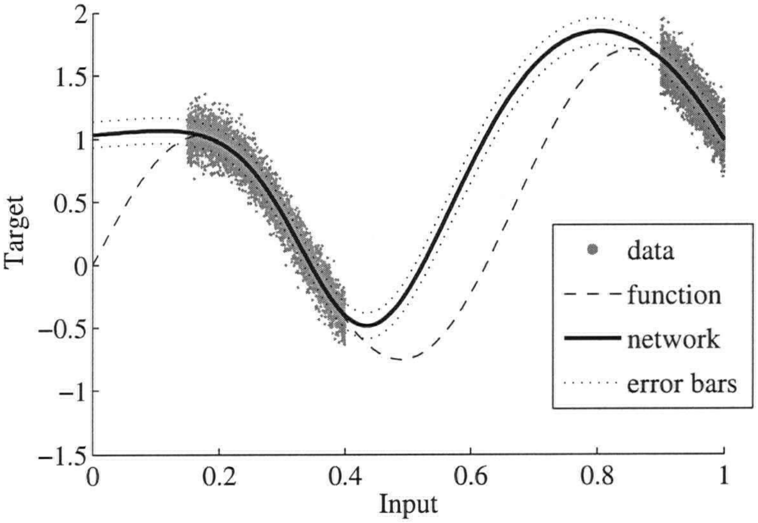


Figure 3.11: Same as Figure 3.10, but with a training dataset consisting of 10,000 datapoints. This example demonstrates how the estimated error bars depend on the factors such as the size of the training dataset, an effect that is not desired in the given case.

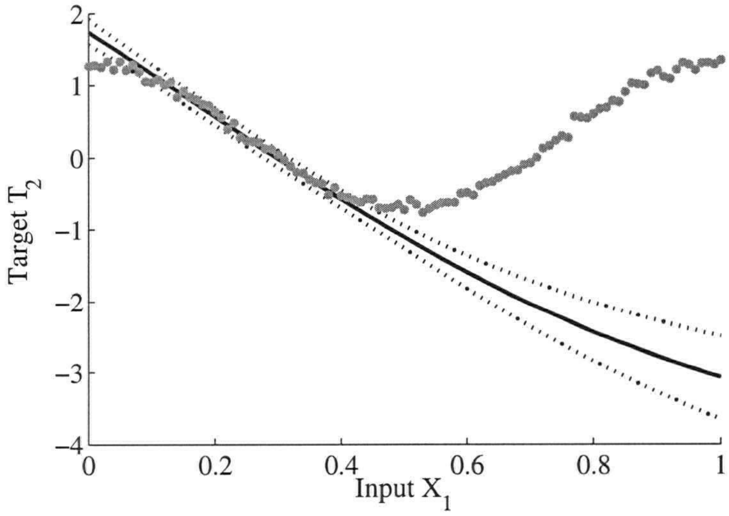


Figure 3.12: The same example as in Figure 3.4, but the network has only been trained with data in the interval $X_1 = [0.15..0.4]$ (plotted in grey, however, are the entire training data). Again, the error bars diverge where no training data was present, however, the divergence is much too weak to indicate the actual error – a problem that I encountered in many cases and which emphasises how difficult it can be to interpret the error bars.

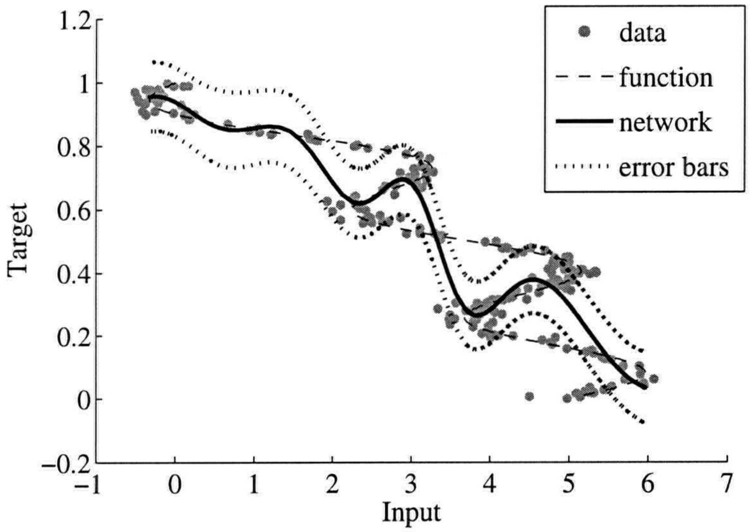
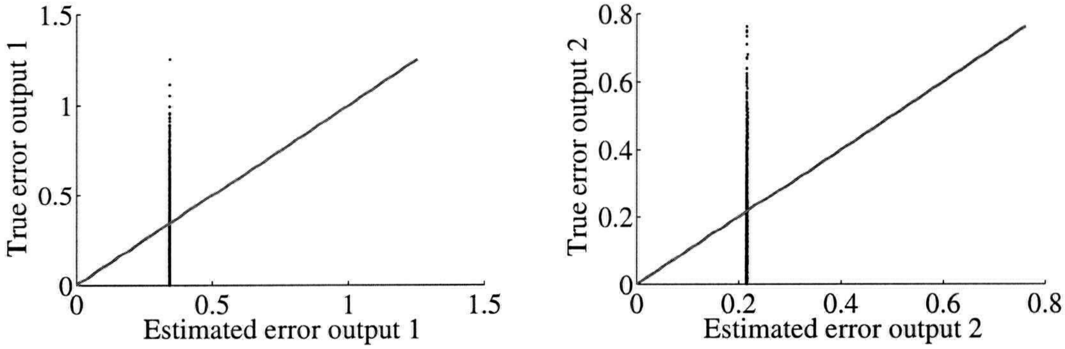


Figure 3.13: Limitations of the error estimation – a simple two-layer perceptron is not able to model a mapping that contains ambiguities, such as the mapping shown here (dashed line). We would hope that at least the error bars could reflect the problematic areas by becoming larger in the ambiguous regions, however, this is not the case. Furthermore, the lower data density in the non-ambiguous regions acts as a counter-productive effect. (One-dimensional example generated with the original NETLAB implementation with scalar hyperparameters.)



(a)

(b)

Figure 3.14: The errors in Figures 3.5 and 3.6 were predicted using a Hessian regularised with the eigenvalue decomposition described in section 3.6. Shown here is the error predicted by the same network as in Figure 3.6, but using non-regularised Hessian. Note the striking difference between computations performed with a regularised and a non-regularised Hessian.

the network prediction. However, if the same problem is repeated with 10,000 datapoints (Figure 3.11), then the network uncertainty term suddenly becomes very small – including the interval where no data were present. In this case, the true function is outside of the error bars of the prediction, and we are confronted with the counter-intuitive result that more observations lead to a worse result.

Figure 3.12 shows a variation of the two-dimensional example used in the previous sections. It displays the same section through the functional surface as Figure 3.4, but this time the network has been trained only with data in a narrow interval. As expected, the error bars diverge in the part where no training data was present, however, they are much too small to indicate the true error to the actual function.

Another problem is the presence of ambiguities in the training data. A simple two-layer network architecture is not able to model such ambiguities (Bishop, 1995), but as I pointed out in Chapter 1, the hope is that the error bars reflect these regions through a larger uncertainty. However, I was not able to achieve this result. On the contrary, regions where ambiguities exist actually have a higher data density than regions where only one functional value is present, leading to the possibility that the error bars are even smaller in the ambiguous parts (Figure 3.13).

This investigation also raises several questions concerning the regularisation of the Hessian – how large is the influence of the regularisation on the training algorithm and on the error bars? Which is the better regularisation method, and how much information is destroyed by performing the regularisation? Figure 3.14 shows the errors from Figure 3.6, but computed using the non-regularised Hessian – the neural uncertainty term almost vanishes. In other examples, I also encountered negative errors due to a non positive definite inverse Hessian. The problem is made worse by the fact that the Hessian is estimated from a large dataset. Since it represents the second derivative of the error with respect to the weights, estimating it from different datasets should not significantly change the Hessian. Usually it will be estimated from the training dataset. However, if the size of the dataset is changed, or the Hessian is estimated from only a part of it, the negative eigenvalues can slightly change, resulting in a different regularisation and hence different error bars. Also unclear is the effect of the regularised Hessian on the Jacobian variability estimates.

The numerical accuracy of the implementation also becomes a problem in the light of the issue that the method has been devised for noisy target data. Hence, if the intrinsic noise variance of the training data becomes small and the network prediction is good enough so that the covariance matrix of the errors $\boldsymbol{C}_0$ has very small elements, the average neural uncertainty term $\boldsymbol{G}^T \tilde{\boldsymbol{H}}^{-1} \boldsymbol{G}$ has to be even smaller in order to re-estimate $\boldsymbol{C}_{in}$ with the help of (3.44). Given the regularisation issue, however, I often encountered the case that neural uncertainty term was larger than $\boldsymbol{C_0}$, leading to negative variances in $\boldsymbol{C_{in}}$, which obviously does not make sense. As a simple workaround, the $\boldsymbol{G}^T\tilde{\boldsymbol{H}}^{-1}\boldsymbol{G}$ term can be omitted in such cases, re-estimating $\boldsymbol{C_{in}}$ by setting it to the $\boldsymbol{C_0}$ values, however, such an approach is likely to generate further uncertainties and problems.

Unfortunately, the noise on the target data in my inverse problem is small, and consequently, I was confronted with this problem during the application of the method to the actual retrieval problem. I will come back to this topic in chapter 4.

For future investigations, it would be interesting to follow the suggestions of MacKay (1995) and compare the performance of the training algorithm using a full diagonal $\boldsymbol{A_r}$ with an individual hyperparameter for each weight to its performance when only a few scalar hyperparameters for distinct groups of weights are used. For some test runs I observed large fluctuations of some elements of $\boldsymbol{A_r}$, and it might be worth testing whether suppressing such fluctuations can influence the training performance.

In conclusion, it seems to require a lot of skill and experience with neural networks in order to train a network well and to interpret the results of the error estimation correctly. Given the training data density dependent magnitude of the error bars and the inability of the method to recognise ambiguities, the usefulness of the uncertainty estimates for the retrieval problem is limited. This is especially true when considering the numerical problems in the implementation. Nevertheless, the obtained uncertainties will give a general idea of the certainty in the neural network fit.

The Jacobian and its variability, on the other hand, provide powerful tools to analyse a network. Hence, at least this part of the uncertainty estimation framework should provide useful results for the retrieval ANNs.

Chapter 4

Retrieval, Results and Evaluation

In the previous chapters, I presented the theory and design of the forward model and the neural network techniques to be used in the retrieval algorithm. In this chapter, I will report on the application of these methods to the actual retrieval. The satellite images of the July 11, 2001, scene as well as the corresponding LUT setup will be described in section 4.1.

Not all pixels observed by the satellite were overcast. In section 4.2, I discuss the operational MODIS cloud mask product and report on problems I encountered with a number of pixels classified as overcast, but exhibiting BTD values outside the expected range.

The design of the neural network is the topic of sections 4.3 and 4.4. I also discuss how the Jacobian can be used to analyse the retrieval performance of a given network architecture, present the sensitivity of the retrievals to cloud top pressure (cf. section 2.6) and check on their physical plausibility.

Good results were obtained with a network architecture containing 15 neurons in the hidden layer that used the brightness temperatures of channels 20 (3.7 μm) and 31 (11 μm), the BTD of channel 31 to 32 (12 μm) and surface temperature as inputs. This network's results will be evaluated in section 4.5 by comparison with the DYCOMS-II in-situ data and an analysis of its Jacobian. I conclude this chapter with a discussion of my findings in section 4.6.

4.1 Retrieval Setup

4.1.1 The Test Scene

The satellite image of the test scene was taken by the MODIS instrument aboard NASA's Terra satellite at 6:25 UTC on July 11th, 2001,[39] corresponding to a local time of 11:25pm PDT (Pacific Daylight Time). Figure 4.1 shows the BTD(3.7-11), as well as the 11 μm brightness temperature (hereafter BT(11)) images. The black circles in the BTD(3.7-11) image mark the flight track during this night. The satellite

[39] In 2001, only the Terra satellite was operational. Hence, no image from Aqua, which would have been approximately six hours later, was available.

overpass did not coincide with the in-situ measurements, which were taken approximately five hours after the MODIS images were recorded. A mean wind speed of about 8 ms^{-1} from the north-west (310°N) was observed during the flight (Stevens et al., 2003b), so that the clouds within the dashed rectangle shown in Figure 4.1 (top) have likely been advected into the flight area. They will serve for comparison with the in-situ data.

The scene contains two ship tracks, discernible in the BTD(3.7-11), but not in the BT(11) image. This can be explained by noting the much smaller single scatter albedo in the thermal infrared in Figure 1.8. This means that the thermal cloud top radiances are mainly a function of cloud emission, and the impact of the scattering term in (2.8) becomes small. Since the LWP of the cloud stays constant across the ship tracks, the emission does not change. As noted in Chapter 2, the ship tracks will be used to check on the physical consistency of the retrieval.

4.1.2 Lookup Table Setup

As discussed in Chapter 2, the forward model requires the input of four variable (r_{eff}, N, p_{ct}, T_{ct}) and three fixed (p_{sfc}, ν_{gam}, k-value) parameters. These parameters were obtained from the in-situ measurements. During RF02, cloud top effective radii ranged from approximately 8 to 16 μm and droplet number concentrations varied between 25 and 115 cm^{-3}. Cloud top temperature in the flight area varied only slightly between 284 and 285 K, and a fairly constant cloud top pressure of 939.5 hPa was observed (not shown). In order to account for a potentially larger variability in the entire satellite scene, the LUT was generated with effective radii ranging from 3 to 23 μm, cloud top temperatures varying between 280 to 288.5 K, and droplet concentrations ranging from 20 to 200 cm^{-3}. Cloud top pressure was fixed at the observed value of 939.5 hPa, as was the surface pressure at 1016.8 hPa.

In order to find representative values for k-value and ν_{gam} (cf. section 2.3), I created statistics of the droplet size distributions that were encountered during the flight. Figure 4.2 shows scatterplots of k-value (obtained from equation (2.19)) versus height. As noted in section 2.1, the data obtained during RF02 cover only cloud top and bottom. The k-values that are found cover ranges similar to those found by Pawlowska and Brenguier (2000) during ACE-2 (Figure 2.5). A similar picture was obtained from the RF03 data (right panel).

Figure 4.3 displays histograms of both k-value and ν_{gam} inferred from the RF02 measurements. Since the radiative transfer through the entire cloud is simulated, I decided to take k as the average of cloud top and bottom values. From the distributions shown, I chose a k of 0.8. This corresponds roughly to

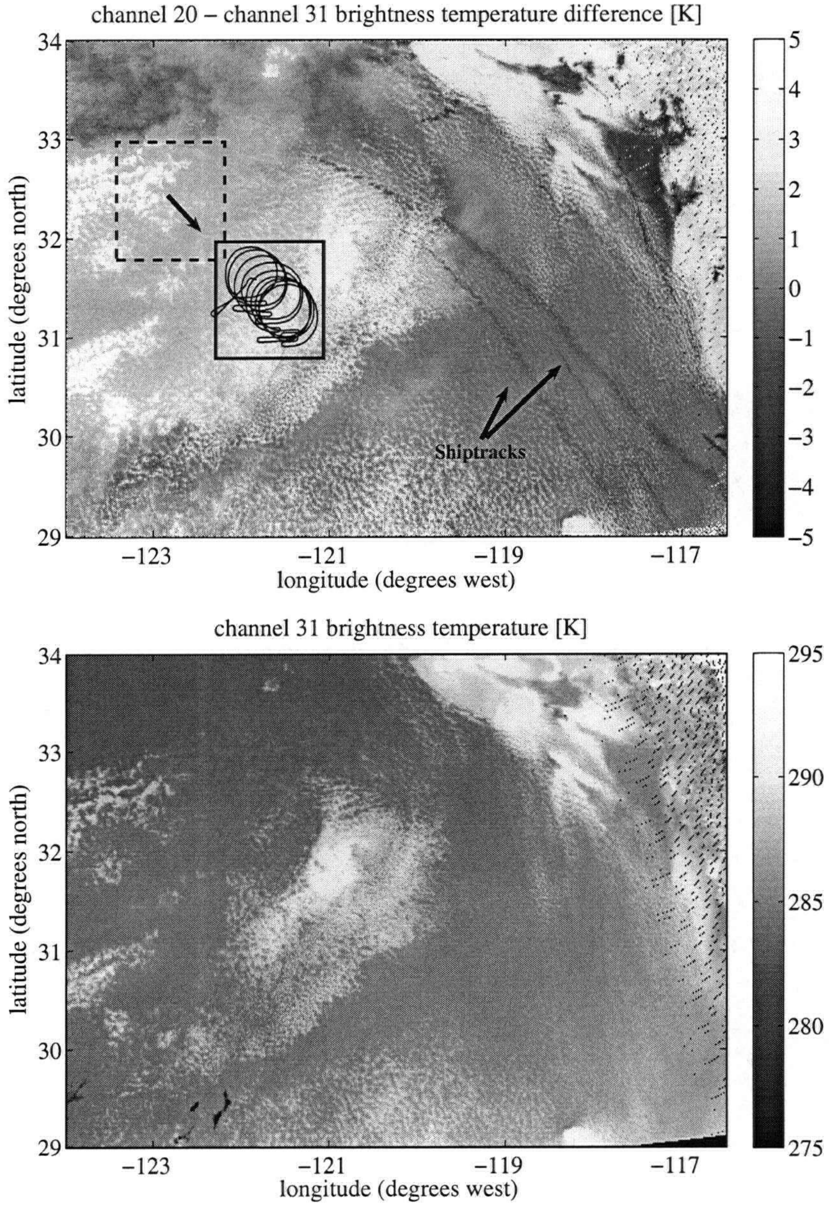


Figure 4.1: (Top) Brightness temperature difference (BTD) between channel 20 (3.7 μm) and channel 31 (11 μm) at 6:25 UTC on July 11th, 2001. The ship tracks have a smaller BTD than their environment, as expected from Figure 2.11. The black circles mark the flight track during this night. The in-situ measurements were taken approximately five hours after the satellite overpass, so that the clouds within the dashed rectangle have likely been advected into the flight area and will serve for comparison. (Bottom) The same for channel 31 brightness temperature. Note that the shiptracks are not discernible in this channel.

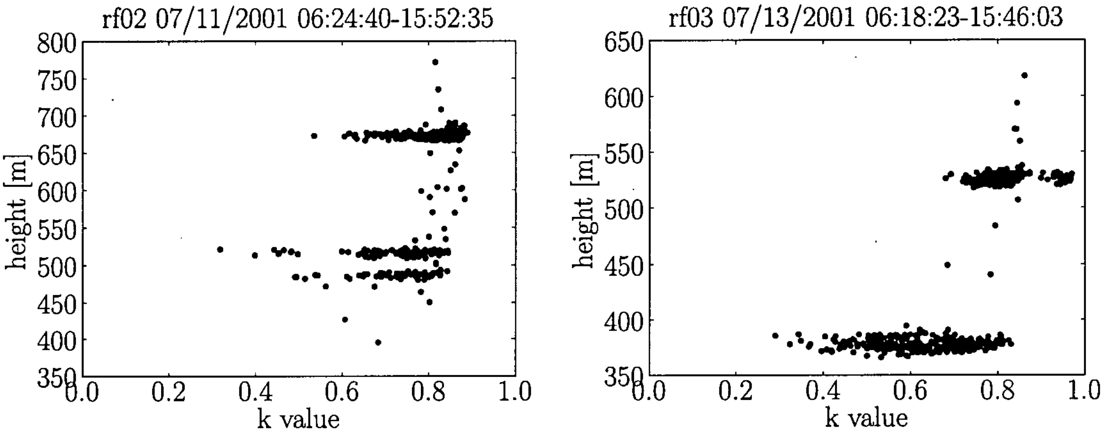


Figure 4.2: k-value vs. height as inferred from the DYCOMS-II in-situ measurements of (left) July 11 and (right) July 13, 2001.

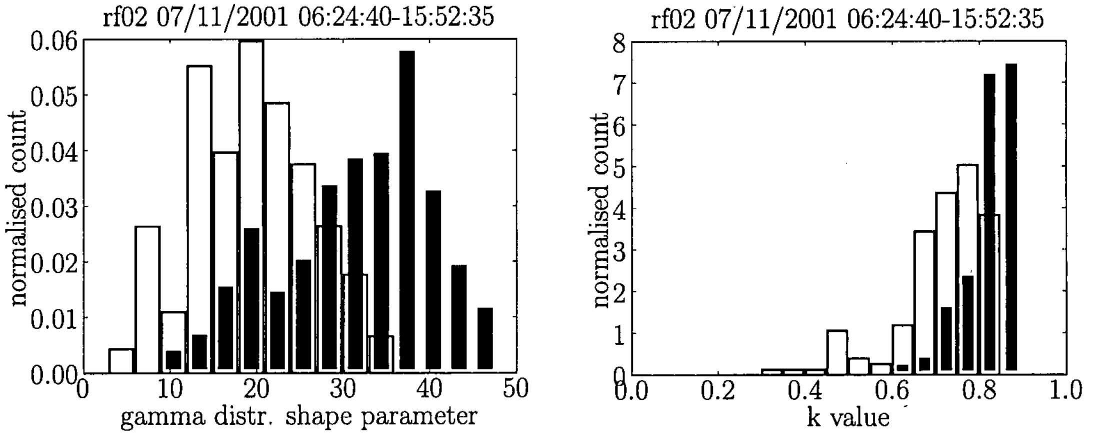


Figure 4.3: Histograms of (left) ν_{gam} and (right) k-value as inferred from the DYCOMS-II in-situ measurements of July 11, 2001. The black histograms represent cloud top values, the white ones cloud base values.

$\nu_{gam} \approx 26$, which I used to compute the Mie tables. Note that these values are considerably larger than the average values found by Miles et al. (2000).

The optical properties of the overlying atmosphere were computed from the San Diego sounding displayed in Figure 2.9. The sounding was recorded at 12:00 UTC, also about five hours after the satellite overpass.

4.2 Cloud Mask

In order to confirm the accurateness of the forward computations, it is important to test how the computed brightness temperatures compare to the observations. If the forward model represents a reasonable approximation to the actual clouds, the observed BT/BTD values should lie well inside the ranges defined by the LUT. However, if BT/BTD values outside the defined ranges are encountered, the corresponding pixels have to be flagged as "irretrievable", since the inverse function is not defined for such cases.[40]

It is likely that some pixels in the scene will contain clear sky or broken clouds. Multiple scattering from cloud sides and inhomogeneous mixtures of cloud and clear sky mean that I expect some pixels in these broken regimes to exhibit radiances not reproducible with the plane parallel forward model. A scatterplot of the forward model computations, overlain with the MODIS observations indeed showed a large number of outliers (not shown). Hence, the operational MODIS cloud mask product (Ackerman et al., 1998) was employed to filter the satellite image for fully cloud covered pixels.

4.2.1 MODIS Cloud Mask and Irretrievable Pixels

A simple way to discriminate cloudy from clear sky pixels and the approach used in the studies of Pérez et al. (2000), González et al. (2002) and Cerdeña et al. (2007) is the spatial coherence method proposed by Coakley and Bretherton (1982). The algorithm uses information from neighbouring pixels to assess the spatial homogeneity in the observations. By computing mean BT and standard deviation from small clusters of pixels (e.g. 2-by-2), homogeneous (i.e. low standard deviation) areas with cold temperatures are classified as cloudy and homogeneous areas with warm temperatures as clear.

Since Coakley and Bretherton (1982), several other methods have been proposed to recognise cloudy pixels in satellite images (Ackerman et al., 1998, and references therein). The operational MODIS cloud mask product combines several of these methods into one product, selecting amongst a variety of tests optimised for different underlying surfaces (i.e. water, different land surfaces, ice) and employing several of the available visible (daytime) and infrared (at day and night) channels. A description can be found in Ackerman et al. (1998).

The top panel of Figure 4.4 shows a map of the cloud mask for the July 11 scene. The image pixels are classified into four categories: cloudy, uncertain clear, probably clear, and confident clear.

Application of the cloud mask to the scene significantly reduced the number of points outside the LUT-defined BT/BTD range. However, after I eliminated all pixels that were not classified as cloudy, the brightness temperature difference diagrams still contained many outliers (Figure 4.5). As noted in section 1.5, Baum et al. (2003) also employed the 8.5 μm signal (MODIS channel 29) in their work. As an unfortunate result, the observed brightness temperatures for channel 29 were entirely outside of the range

[40] Note that the neural network will still retrieve *some* values for such undefined inputs, so that it is important to remove the corresponding pixels from the scene in order to avoid unphysical retrievals.

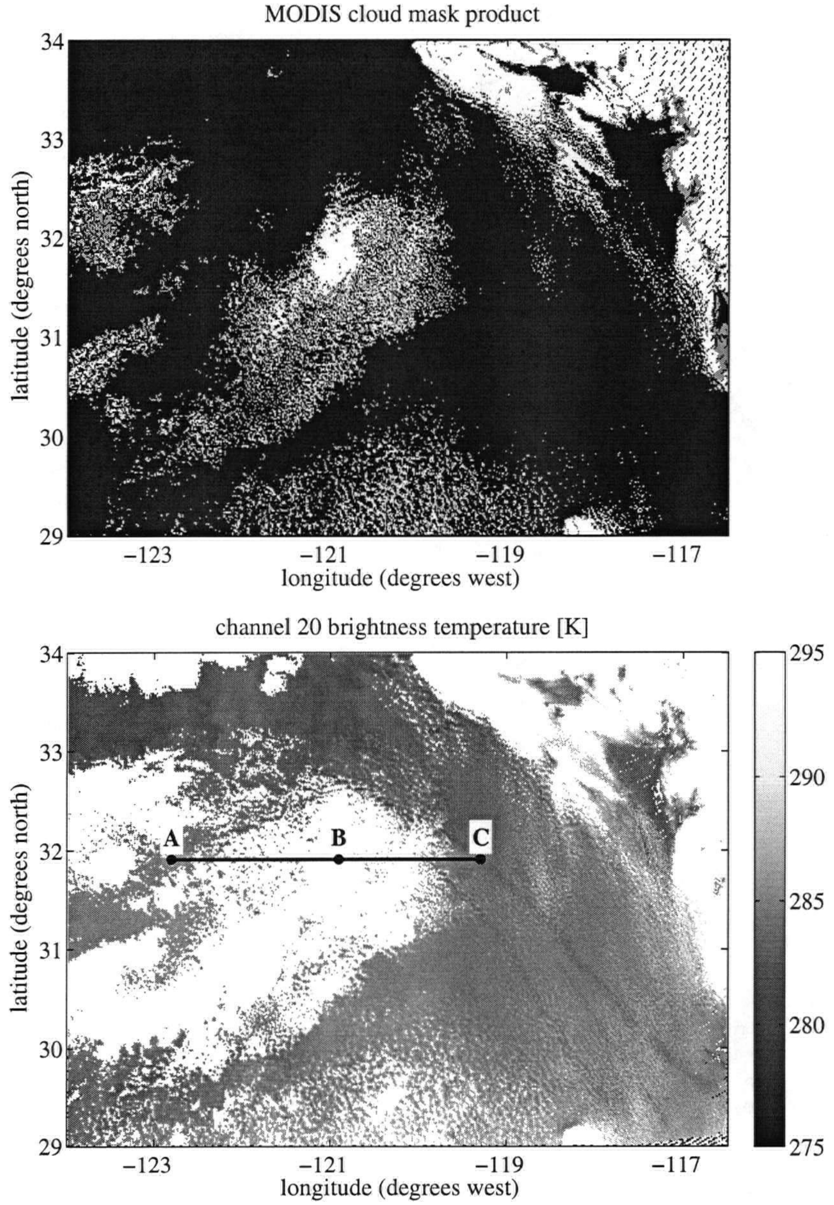


Figure 4.4: (Top) MODIS cloud mask product for the scene. All black pixels are classified as cloudy, the orange (grey in black and white) pixels are uncertain clear, and the white pixels are probably clear. (Bottom) Cloud mask derived from the BT/BTD range in the LUT (all observations that are outside the LUT-defined BT/BTD region in Figure 4.5 are marked as dark red). Note that the MODIS cloud mask has "grown"; are the additional pixels broken clouds?

computed by my forward model (Figure 4.6). Currently, I can only conclude that there is an error in the forward model, and consequently I will drop channel 29 from the retrieval scheme. It would, however, be desirable to investigate the cause of the failure and to include the 8.5 μm information in future retrieval designs.

Yet what causes the large number of outliers in the remaining channels 20 (3.7 μm), 31 (11 μm) and 32 (12 μm) brightness temperatures? The bottom panel of Figure 4.4 shows the scene with all pixels outside the LUT-defined BT/BTD range removed. The cloud mask basically grows – this could be an indication of more broken or otherwise inhomogeneous clouds at the edges of the clear areas.

4.2.2 Possible Failure Mechanisms

There are several possibilities for the failure of the forward model to match the observed brightness temperatures. Besides inhomogeneous clouds, it is possible that other assumptions in the forward model do not match the real clouds accurately enough. For instance, the anomalous pixels could contain subadiabatic clouds or clouds with a significantly different cloud top height. Effective radii, droplet number concentrations and temperatures outside the in the LUT ranges seem unlikely. The majority of the "irretrievable" pixels in Figure 4.5 exhibit both a larger BTD(3.7-11) and BTD(11-12) signal than defined by the LUT, while the BT(11) signal is well inside the computed range. Warmer or colder cloud temperatures would cause the location of the failing points on the BT(11) axis to fall outside the computed range. Larger r_{eff} than defined would cause a larger BTD(3.7-11) signal, but at the same time a smaller BTD(11-12) observation (cf. Figure 2.11). Since droplet concentration is not a direct input into libRadtran (cf. section 2.6), changing droplet concentrations would influence r_{eff} and τ, the latter of which would merely displace the BT/BTD points within the defined ranges (cf. Figure 2.11).

Another possibility is that the structure of the overlying atmosphere changed with time and location, so that the sounding recorded at San Diego is not representative for the entire scene. This could also include the presence of thin high level clouds (cirrus).

In order to investigate the problem of the irretrievable pixels, I analysed the pixels along a test line in an area that included clouds within the LUT-defined BT/BTD range, pixels classified as cloudy by the MODIS cloud mask, but outside the defined BT/BTD range, and pixels classified as clear sky by the MODIS cloud mask. The line is shown in the bottom panel of Figure 4.4. The capital letters A and C mark pixels within the LUT-defined BT/BTD range, and B marks a clear sky pixel.

I first tested for broken clouds. Luo et al. (1994) showed that broken clouds exhibit a characteristic

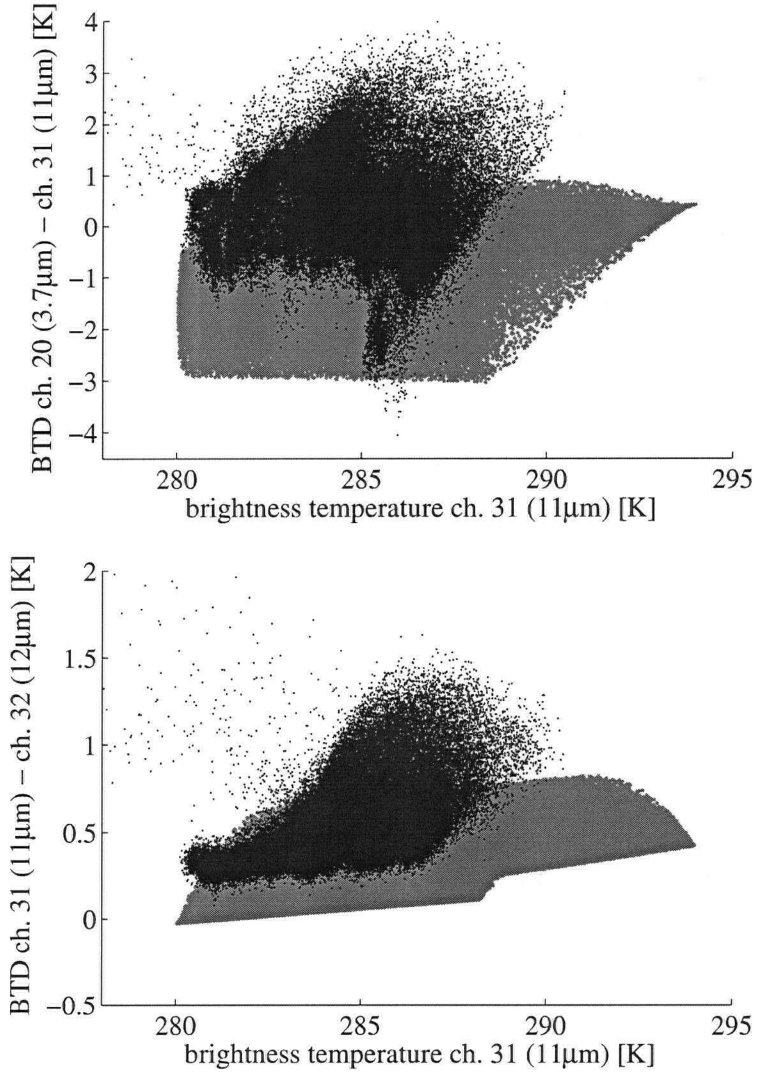


Figure 4.5: LUT (grey) and MODIS cloud mask filtered scene brightness temperatures. Many pixels are still outside of the BT range contained in the LUT. These pixels are not defined if a retrieval network is trained with the LUT data, hence they have to be flagged as "irretrievable".

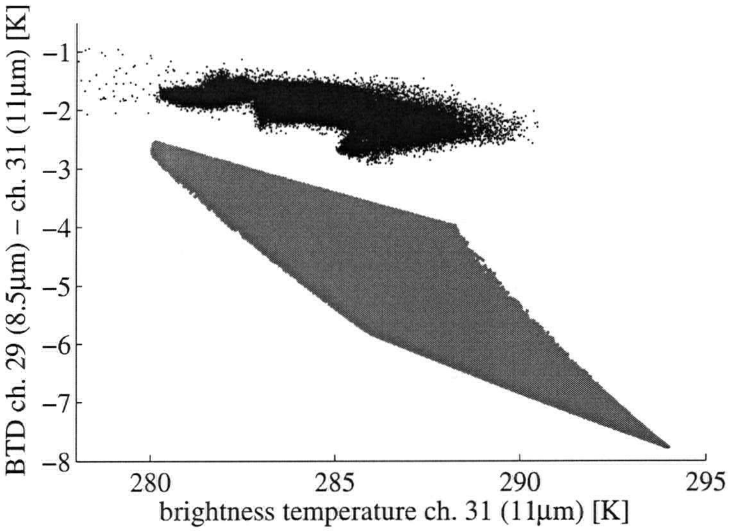
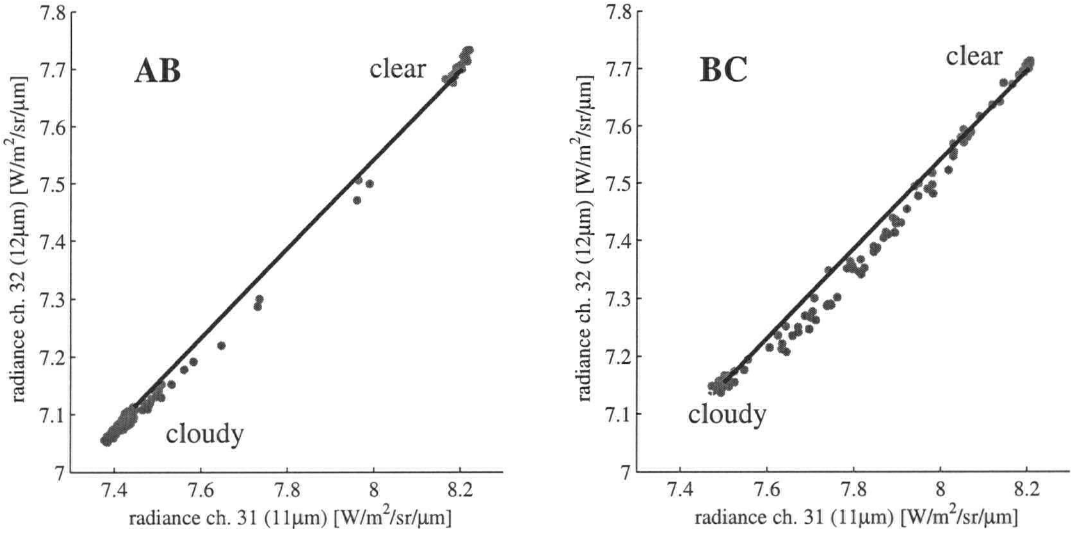


Figure 4.6: The same as Figure 4.5, but for the brightness temperature difference between channels 29 and 31 (8.5 and 11 μm). Channel 29 BTs are 1-5 K cooler than the MODIS measurements. Channel 29 hence could not be used for the retrievals.

pattern when the observed 11 μm radiances (not brightness temperatures) are plotted against the 12 μm radiances. In particular, if a given area contains clear sky pixels, broken clouds and fully overcast pixels, the entire set of observations resembles a continuous curve in the diagram between cloudy and clear pixels. Figure 4.7 shows an 11 versus 12 μm radiance plot for the sections from A to B (left panel) and from B to C (right panel). The section between B and C clearly contains broken clouds. This area already contains many pixels classified as "uncertain clear" by the MODIS cloud mask product, so that I conclude that inhomogeneities on the sub-pixel scale are likely to be responsible for the failures.

The other section between A and B, however, does not show the characteristic broken cloud signature in the radiance plot. Yet a larger number of the pixels along this line are irretrievable. In order to extend the analysis, I examined the observed BTD(3.7-11) and BTD(11-12) signals (Figure 4.8). Similar to the radiance plot in Figure 4.7, the observed values cluster around the cloudy and clear foot. The elevated BTD(3.7-11) signal places almost all points outside of the LUT-defined BT/BTD range (with point A just at the edge), however, the pixels clustering around the cloudy BTD(11-12) signal are all well inside the computed range.

A higher cloud top seems to be an unlikely cause of such a behaviour. As noted in Chapter 2, the differences in the BTD signals due to varying cloud top pressure in the forward model are mainly due to the change in sea surface temperature produced by the new adiabatic profile (cf. Figure 2.12). However, the existence of a much warmer T_{sfc} between A and B compared to B and C seems unlikely. Also, both



(a)

(b)

Figure 4.7: 11 μm versus 12 μm radiance plots after Luo et al. (1994) for the observations along the line shown in Figure 4.4. Left panel: the section from A to B, right panel: from B to C. The solid lines connect the observed radiances at A and B (left) and B and C (right), respectively, and are only shown as references to the point clouds. The observations between points B and C follow a typical signature of broken clouds.

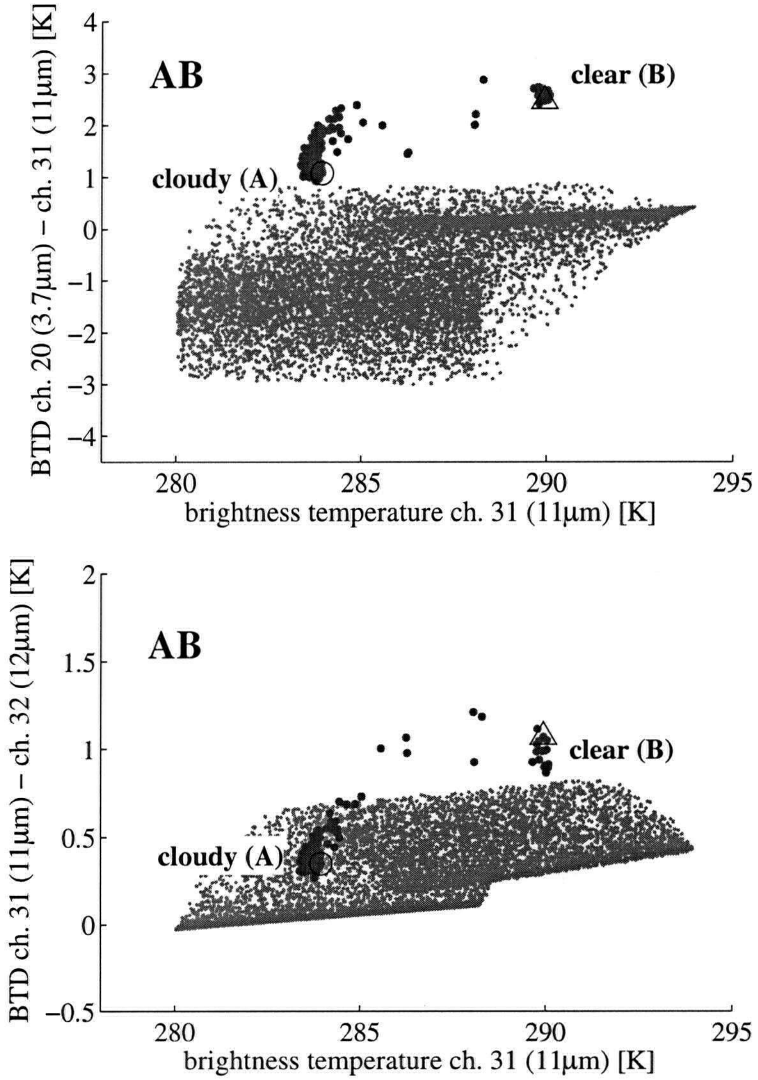


Figure 4.8: BTD(3.7-11) and BTD(11-12) signals of the pixels between points A and B as shown in Figure 4.4. As in the radiance plot in Figure 4.7, the observed values cluster around the cloudy and clear foot. It is possible that this behaviour is caused by thin overlying cirrus clouds.

BTD(3.7-11) and BTD(11-12) would be impacted by such a change.[41] Other causes for an increased p_{ct} could be equally thick, but elevated clouds, or geometrically thicker clouds. The first possibility would not change the optical properties of the cloud, and the second mechanism would merely lead to increased τ and r_{eff}.

Baum et al. (1994) and Baum et al. (2003) showed that overlying cirrus clouds lead to an increase in both the BTD(3.7-11) and BTD(11-12) signals. They also showed that thin cirrus has relatively small impact on the BTD(11-12) signal, making thin cirrus a likely cause of the lookup table failure along AB. However, in order to give this presumption more confidence, further investigations would have to be conducted. Also, possible effects of subadiabatic clouds should be examined in the future. For this thesis, I will continue to work with those pixels that fall within the LUT-defined BT/BTD range.

4.3 Network Training

4.3.1 Network Architecture

As described in Chapter 3, I restricted myself to the two-layer perceptron design. Cerdeña et al. (2007) found that a three-layer architecture exhibited a better generalisation performance in their study, however, since a two-layer network should already be able to model arbitrary functions (cf. Chapter 3), the generalisation of the software to three-layer architectures was not a priority of my work.

With respect to the inputs, decisions to be made included whether to provide the satellite observations as radiances or as brightness temperatures to the network, the inclusion of sea surface temperature as input, and how the inputs should be preprocessed. Another difficult problem was selecting a good number of hidden neurons. Aires (2004) used a two-layer perceptron with 30 neurons in the hidden layer in their example of microwave remote sensing, and as noted in Chapter 1, Cerdeña et al. (2007) employed 20 neurons in the first and five neurons in the second layer. Motivated by these values, the number of hidden neurons in my architectures will be of the same order of magnitude.

It did not seem feasible within the timeframe of this Master's thesis to systematically train, analyse and compare a large number of different network architectures in order to find the best possible one. Instead, after some preliminary try-outs, I selected some architectures that I will present in more detail in this chapter. These networks yielded the most interesting results. The methods applied to analyse these

[41] The data in Figure 2.12, computed with a similar cloud top temperature as encountered in the July 11 scene, also showed that a relatively large increase in p_{ct} is necessary in order to increase the BTD(3.7-11) signal (40 hPa between the solid and dotted curve in Figure 2.12). Also, the resulting relative change in the BTD(11-12) signal is stronger than in the BTD(3.7-11) signal.

networks, however, can readily be used for any other network architectures to be employed in future work.

The authors of all works presented in Chapter 1 that investigated the nocturnal retrieval case employed satellite observations in the form of brightness temperatures, while the daytime retrieval techniques in general use radiances. In fact, my tests showed that networks using radiances as inputs were unable to approximate the inverse function (possibly because channel differences provide the most information about cloud optical properties – I did not investigate the use of radiance differences as inputs). Therefore, and in order to stay consistent with the published literature, I chose to use brightness temperatures instead of radiances as inputs (this also facilitates the analysis of the networks in sections 4.4 and 4.5).

Unfortunately, the training process generally was unstable, as I will discuss in section 4.4. Training of networks of a given architecture led to very different results when the weights were initialised differently at the beginning of the training process – a property that I attribute to the inverse problem being ill-posed (cf. chapter 3). This is also reflected in a large variability in the Jacobian (see the following section). Consequently, my goal for this thesis is not to find the best possible architecture, but to show that the method is working in principle and to demonstrate the use of the Jacobian and other tools to compare architectures and to evaluate the performance of a given network.

4.3.2 Failure of the Aires et al. Hyperparameter Re-Estimation

The estimation of the matrix hyperparameter $\boldsymbol{A_{in}}$ also often failed during network training. As discussed in section 3.7, the neural uncertainty term $\boldsymbol{G}^T\boldsymbol{H}^{-1}\boldsymbol{G}$ in general was larger than the error covariance matrix $\boldsymbol{C_0}$, leading to negative variances in $\boldsymbol{C_{in}}$. The problem was encountered with both regularisation methods described in section 3.6.

Aires (2004) pointed out that the estimation of the Hessian $\boldsymbol{H}$ from the training dataset has to be done with a large enough number of datapoints in order to avoid numerical problems. However, increasing the number of samples in the LUT from initially 30,000 (20,000 for training and 10,000 for validation) to 96,000 (64,000 and 32,000, respectively) did not improve the situation (for comparison, Aires (2004) used 15,000 samples for training and 5,000 for testing, and Cerdeña et al. (2007) 20,000 and 10,000, respectively).

The true cause of the failures currently remains unclear. Further research is needed to clarify whether the modification of the Hessian due to the regularisation is the important factor, or whether the noise level on the target data in the LUT is too small to be treated with the Aires et al. method. As noted in section 3.1, there is no noise on the target data except for the expected ambiguities. However, as I did not investigate the actual magnitude of such ambiguities, I cannot rule out the possibility that they are small.

Due to these problems I eventually decided to train the networks with the original scalar hyperparameter approach implemented in NETLAB (cf. section 3.2). Using the LUT containing 96,000 datapoints and the Nabney (2002) eigenvalue regularisation (cf. section 3.6), I was able to estimate a $\boldsymbol{C_{in}}$ from the trained network in some cases, although often this strategy failed as well. However, the estimation of the Hessian and the network gradient were always possible, so that the variability of the Jacobian as well as the neural uncertainty term $\boldsymbol{G}^T\boldsymbol{H}^{-1}\boldsymbol{G}$ could be computed.

4.3.3 Input Preprocessing

As discussed in Chapter 3 and expected from the findings of Aires et al. (2004b), correlations among the input variables were a problem. The observed brightness temperatures of the three employed channels were all correlated amongst each other with (linear) correlation coefficients larger than 0.8. This lead to widely varying Jacobians, and it was practically impossible to obtain a network fit that approximated the lookup table well. The input data were thus decorrelated and normalised with the whitening procedure described in Chapter 3, leading to better results.

4.4 Network Architecture: Inputs and Hidden Neurons

4.4.1 Brightness Temperature Differences

Despite the input preprocessing, it appeared to be difficult for the ANNs to infer the brightness temperature differences from the BT inputs. Especially the close correspondence between the 11 and 12 μm channels seemed to have a negative effect on the stability of the training process. In fact, I was not able to produce a reasonable fit to the LUT with networks that used BT(11) and BT(12) (12 μm BT) as inputs. To analyse the problem, I computed the Jacobians of individual points in the LUT in order to compare the sensitivities to expected values. Figure 4.9 shows the BT/BTD diagrams of a subset of the July 11 LUT. In addition to the cloud top pressure, cloud top temperature is also held constant at 285 K. The effective radius varies in steps of two microns from 4 μm to 12 μm, as in Figure 2.11.

Since all variables are connected with each other in a nonlinear way, it is difficult to assess Jacobian values visually from the plots in Figure 4.9. Of course, it is possible to compute finite difference derivatives from the LUT. However, finding the required datapoints for a sensitivity estimation constitutes a

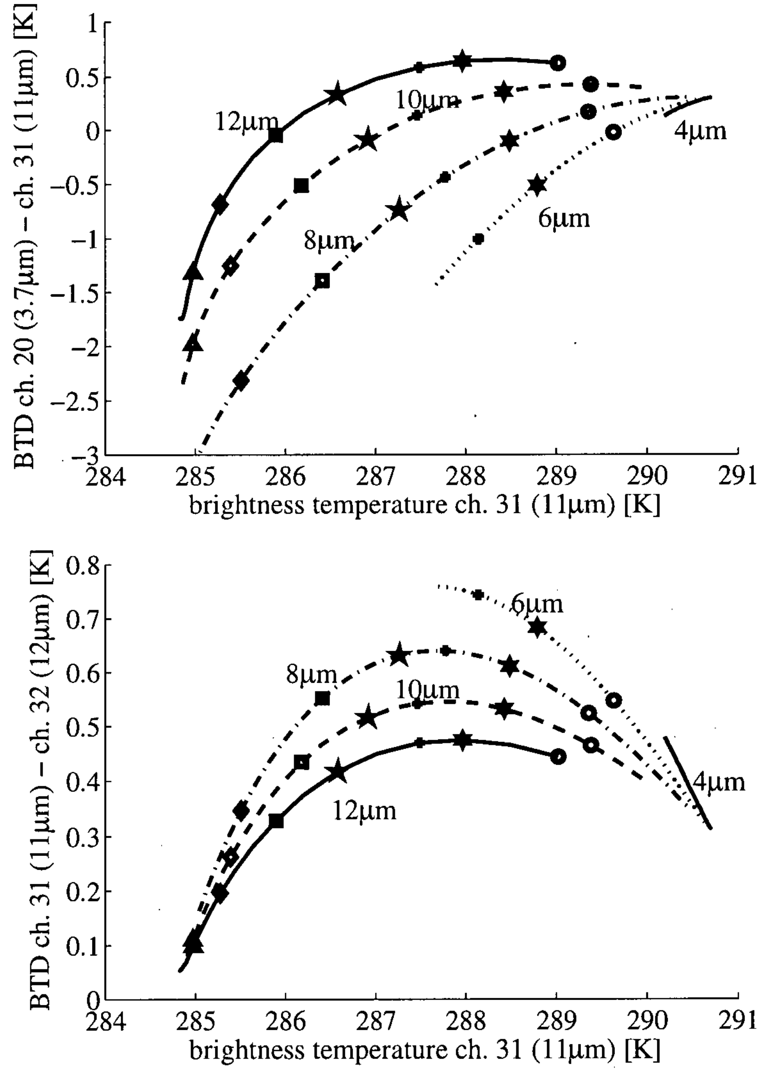


Figure 4.9: Subset of the July 11 LUT. In addition to the cloud top pressure (939.5 hPa), the cloud top temperature is also held constant at 285 K. Compare visually obtained estimates of the Jacobian for $r_{eff} \approx 8\mu m$ and BT(11) $\approx$ 288 K to the computed values in Tables 4.1, 4.2 and 4.3. The symbols correspond to the optical thicknesses defined in Figure 2.11.

multidimensional optimisation problem itself. Computing the derivative of an output to a given input, at least two points have to be found for which the other three inputs are constant. Even with 96,000 datapoints, this was not possible to a good enough accuracy (solving this problem, for instance, with larger LUTs and interpolation could constitute an important part of future research in this area).

I thus restricted myself to order-of-magnitude estimates from Figure 4.9 where possible. For instance, for an effective radius of 8 μm, if BT(11) is held constant at 288 K, $\partial r_{eff}/\partial\text{BT}(3.7) \approx 2.5\mu\text{m/K}$. Similarly, $\partial r_{eff}/\partial\text{BT}(12) \approx 18\mu\text{m/K}$. Cloud optical thickness decreases slightly with increasing BT(3.7), and increases with increasing BT(12) ($\partial\tau/\partial\text{BT}(3.7) \approx -0.5\text{K}^{-1}$ and $\partial\tau/\partial\text{BT}(12)$ is positive). If all cloud parameters are held constant, an increase in surface temperature will lead to a similar increase in cloud top temperature (cf. Figure 2.9). Since the cloud in question is thin ($\tau \approx 1.3$), the surface temperature signal is expected to be "visible" through the cloud, so that $\partial T_{ct}/\partial T_{sfc} \approx 1$. Likewise, T_{ct} should depend on BT(11) and BT(12); for thicker clouds, these two inputs almost entirely determine cloud top temperature (cf. section 4.1 and Figure 2.9).

Table 4.1 shows the Jacobian estimated with the entire LUT trained network at the discussed coordinates of $r_{eff} \approx 8\mu$m and BT(11) $\approx$ 288 K. The ANN contained 30 neurons in the hidden layer, and in addition to BT(3.7), BT(11) and BT(12) also used surface temperature as an input (the T_{sfc} input will be discussed shortly). The variability was obtained by computing the Jacobians of 10,000 samples from the weights distribution, as described in section 3.5.

Table 4.1: Point estimate of the Jacobian of a network containing 30 neurons in the hidden layer and using BT(3.7), BT(11), BT(12) and T_{sfc} as inputs. Note the large variability of the Jacobian, indicating an ill-posed problem. Furthermore, the dependences on BT(11) and BT(12) are large and of opposite sign.

	∂r_{eff} / — $[\mu m/K]$	∂T / — $[K/K]$	$\partial\tau$ / — $[K^{-1}]$
∂BT(3.7)	0.22 ± 4.72	-0.07 ± 1.58	-1.10 ± 9.83
∂BT(11)	-20.16 ± 20.92	-0.78 ± 7.38	-4.83 ± 34.53
∂BT(12)	19.11 ± 21.67	0.75 ± 8.30	5.56 ± 37.12
∂T_{sfc}	1.18 ± 3.37	1.09 ± 1.27	0.47 ± 6.05

Two features are particularly noticeable. First, there is a large variability in the computed values (i.e. uncertain network fit), especially in the dependences on BT(11) and BT(12). Second, the dependences on these two channels are large and of opposite sign. Since BT(11) and BT(12) are so highly correlated, this behaviour models the dependence on the differences between the two channels (since BT(11) and BT(12) will always change by approximately the same value). However, the large values make the retrieval very sensitive to noise in the inputs, and there is no reason why they could not be much smaller or of the same sign but slightly different magnitude.

Large and opposite sign sensitivities for at least one output could be observed for all ANNs trained with BT(11) and BT(12) inputs (although the magnitude of the dependences and variability varied, depending on the minimum that was found in the weights error surface), and the performance of the networks was very sensitive to the initialisation of the weights and the number of hidden neurons. In an attempt to make the training process more stable, I replaced the BT(12) input with the BTD(11-12) signal, which indeed improved the training performance. It is worth noting that a similar replacement of the BT(3.7) input by the BTD(3.7-11) signal did not lead to comparable improvements.

The empirical Jacobian reported by Cerdeña et al. (2007) (cf. Chapter 1) shows the same behaviour of opposite signs, but with smaller magnitudes (although they compute a scene-averaged Jacobian, whose values should not be compared with the point estimate given in Table 4.1). Their network seems to perform well (although no results are given in the paper); hence I conclude that in general it is possible for the ANN to model the correct dependences from the "raw" BT inputs, but that using the BTD(11-12) input improves the stability in the training process.

4.4.2 Three Input Networks

A different question was whether surface temperature, as used in all previous studies, is a necessary input. Pérez et al. (2000) argued that T_{sfc} is necessary in order to compute the upwelling cloud base radiation, and Cerdeña et al. (2007) added that the effects of water vapour in the atmosphere can be accounted for by using clear sky BTs of all input channels. I prescribe the optical properties of the overlying atmosphere and the effects of subcloud water vapour are connected with the cloud parameters through the adiabatic model. However, it is true that three variables are not enough to uniquely specify an adiabatic cloud in my forward model. If cloud top pressure is held constant, four more parameters are needed to compute a profile (for instance, cloud top T, r_{eff}, liquid water content and either surface pressure or temperature). Nevertheless, since surface temperature does not belong to the direct satellite measurements, it was worth testing the behaviour of the networks if no T_{sfc} input is used.

The retrieval attempts using three inputs were not successful. Figure 4.10 shows scatter plots of the target variables in the LUT (cloud top r_{eff}, T and τ) versus ANN predictions, for both a three input network and a four input input network that made use of T_{sfc}. Such scatter plots provide a good way for visualising whether the network is able to approximate the target data in the training or validation database. If the network predictions are perfect, the plots will take the shape of a straight line. The wider the scatter around this line, the worse is the fit. Obviously, intrinsic noise on the target data also creates scatter.

The plots in Figure 4.10 lead to the conclusion that the three input network is not able to produce good predictions of r_{eff} and T, only τ is predicted reasonably well. The ANN that produced the displayed results had 30 neurons in the hidden layer, but varying this number did not improve the performance.

Nevertheless, in order to ensure that the wide scatter is not caused by ambiguities in the LUT and that the four input network possibly overfits the data, I applied the three input ANN to the actual scene. The retrieval results of cloud top temperature and cloud optical thickness are shown in Figure 4.11. While the optical thickness retrieval shows the expected signature of optically thicker clouds along the ship tracks, the tracks are also discernible in the temperature retrieval, exhibiting a higher temperature than their environment. This is not the expected physical behaviour – the aerosol particles contained in the ship exhaust should not influence the cloud temperature. Furthermore, many of the retrieved droplet sizes were negative (not shown).

A curious feature of all trained networks is that they are able to retrieve optically very thick clouds from the LUT. Due to the saturation effect discussed in section 1.5 I had expected that target optical thicknesses above a certain threshold would be retrieved as the threshold value. It is currently unclear if there actually is enough information in the input data to infer the large optical thicknesses or if the networks are overfitting in this case. Since the scene of July 11, 2001, does not contain clouds with (retrieved) τ larger than about 6, no anomalies can be found in the retrieval (cf. section 4.5). However, this behaviour should be further investigated in the future.

4.4.3 Four Input Networks

The scatter plots of the four input network in Figure 4.10 showed that including T_{sfc} as an input significantly improves the ability of the network to fit the lookup table well. In contrast to Cerdeña et al. (2007), I used the actual surface temperature as a single input, not the clear sky brightness temperatures of all three employed channels. Using T_{sfc} as an input at first seems to be a significant restriction to the usefulness of the retrieval, since clear sky pixels are not always available in the near vicinity of the clouds whose properties are to be retrieved. However, sea surface temperature retrievals are also possible from microwave imagers such as the Advanced Microwave Scanning Radiometer for the Earth Observing System (AMSR-E; Kawanishi et al., 2003) on-board the Aqua satellite (which also carries the

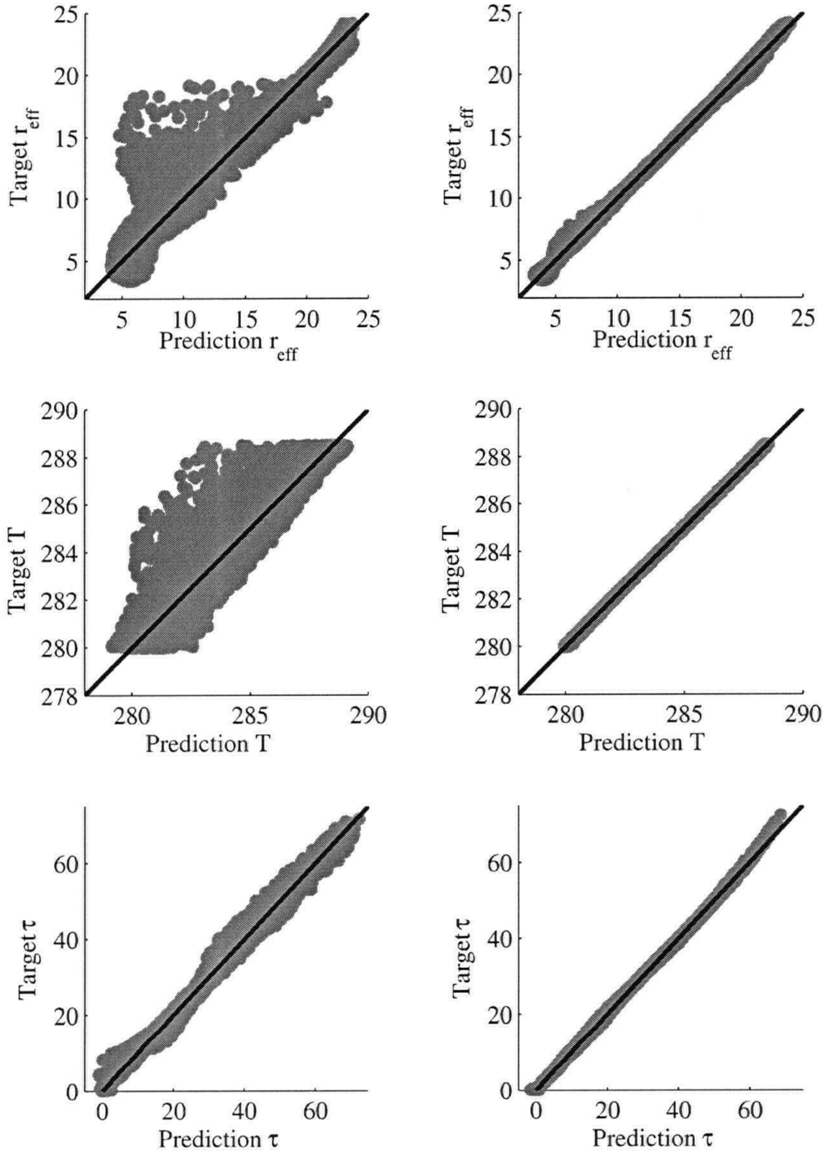


Figure 4.10: Training performance of a three input network (left column) employing the BT(3.7), BT(11) and BTD(11-12) signals, compared to a four input network (right column) additionally making use of surface temperature. The scatter plots show the target data, as contained in the validation database, plotted against the corresponding network predictions. It is clearly visible that three inputs are not enough to approximate the inverse function.

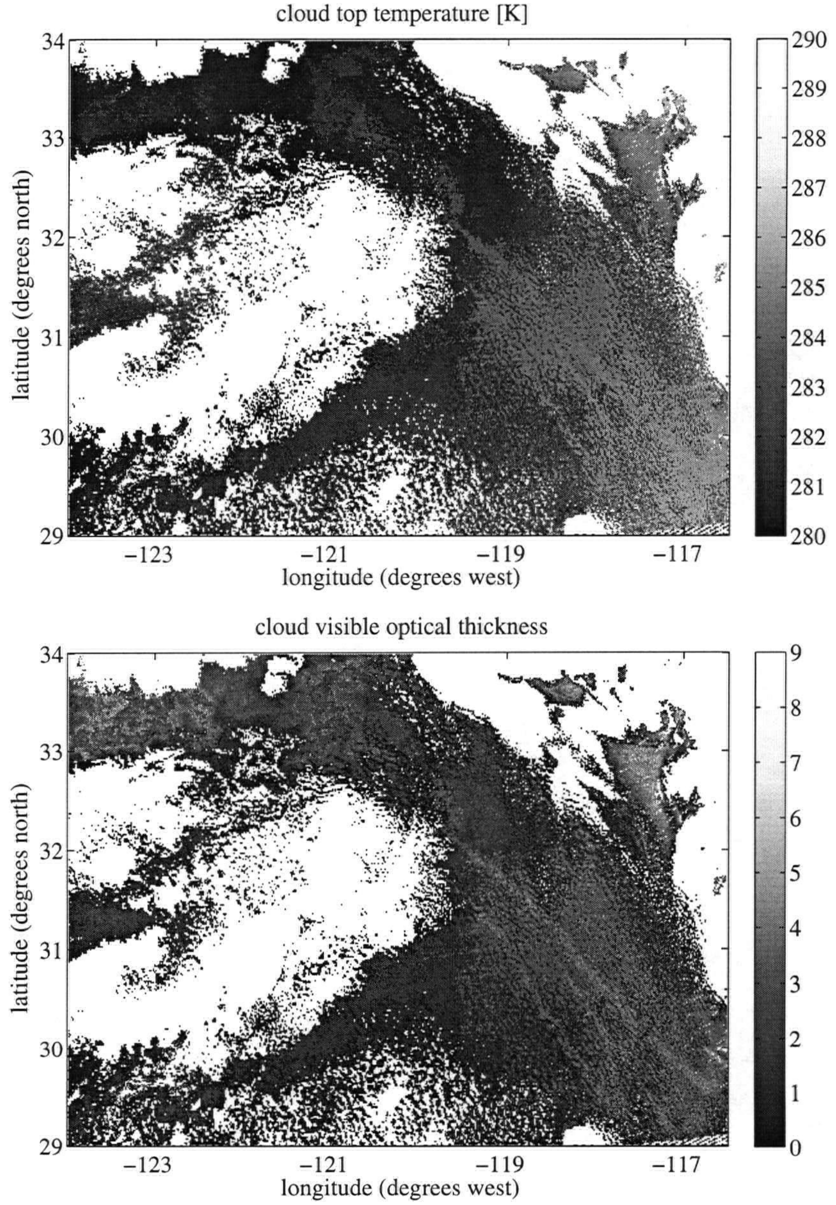


Figure 4.11: Retrievals of cloud top temperature (top) and visible cloud optical thickness (bottom) of the three input network from which the scatter plots on the left side of Figure 4.10 were produced. The ship tracks are discernible in the T_{ct} retrieval, which is not the expected physical behaviour.

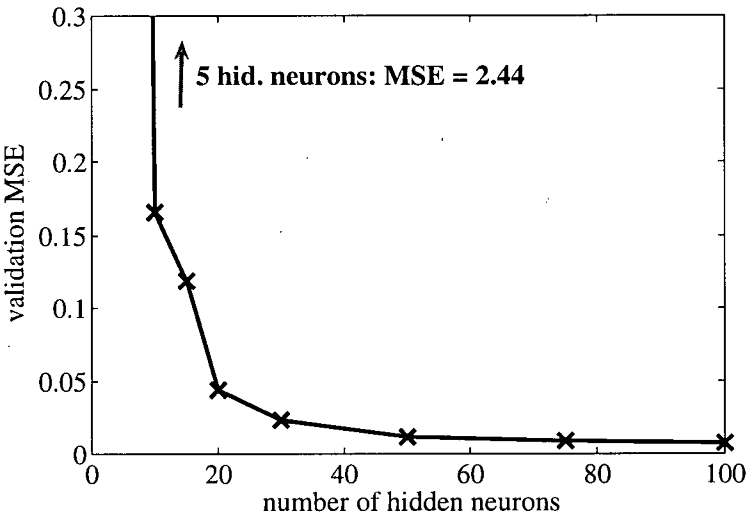


Figure 4.12: Mean-square-error of the validation dataset for the four input ANN (BT(3.7), BT(11), BTD(11-12), T_{sfc}) in dependence of the number of hidden neurons. The networks were trained with four hyperparameter re-estimation cycles, each with 1500 optimisation steps.

second MODIS instrument). Such retrievals are largely independent of cloud cover, since clouds are semi-transparent in the microwave. For this study, I used the T_{sfc} measurements taken during the DYCOMS-II research flight. On July 11, 2001, the surface temperature was approximately 19°C.

The next open question was how many hidden neurons are needed to approximate the inverse function well without adding too many degrees of freedom to the training process. As mentioned above, the work by Aires (2004) and Cerdeña et al. (2007) suggested that a number on the order of 30 neurons should be sufficient. I thus trained a number of ANNs with different numbers of hidden neurons ranging from five to 100, and compared the MSE of the validation data. Figure 4.12 shows the decrease of the error with an increasing number of hidden neurons.

However, the curve is deceptive – the conclusion that a 100 hidden neurons network performs better in the retrieval than a 15 hidden neurons network proved wrong. In fact, from all networks that were trained the most physical plausible results were obtained from the one containing 15 neurons in the hidden layer, although its MSE was higher than that of other networks. As noted in section 3.1, models of a higher complexity (i.e. a larger number of hidden neurons) are more susceptible to overfitting. However, the Bayesian regularisation in the training process should effectively prevent overfitting to noise in the target data (cf. section 3.2), and as noted above, there seems to be little noise in the LUT. Instead, I assume that the described behaviour, too, has to be attributed to the ill-conditioning of the inverse problem – if a network contains more hidden neurons, there exist more possibilities to map the inputs to the target data. More possible mappings also mean a larger number of possible dependences, hence it becomes more likely that a physically incorrect dependence is modelled. It is thus especially in this case important to find a network with a good degree of complexity (cf. section 3.1).

In order to investigate the problem, I selected two networks – 15 and 30 neurons in the hidden layer, referred to as 15N and 30N hereafter – and analysed some of their properties.[42] Although 30N produced a better fit to the LUT (lower MSE), the retrieval of the July 11 scene yielded many unphysical results (for instance, negative optical thicknesses).

Tables 4.2 and 4.3 show point estimates of the Jacobian at $r_{eff} \approx 8\mu$m and BT(11) $\approx$ 288 K, as in Table 4.1, for 15N and 30N. The most striking difference is the much larger variability in the 30N Jacobian, indicating that a less well-defined minimum in the weights error surface has been found. This increases the probability of an incorrect mapping. Furthermore, while the sensitivities of r_{eff} to the four inputs are similar for both networks (and at least $\partial r_{eff}/\partial$BT(3.7) is in the expected range), those of T_{ct} and τ are different. For instance, as expected, 15N retrieves a good part of the T_{ct} output from the BT(11) signal. 30N, on the other hand, infers cloud top temperature almost exclusively from the surface temperature input (since the variation in BTD(11-12) is so small).

Table 4.2: The same as Table 4.1, but for a network containing 15 neurons in the hidden layer and using BT(3.7), BT(11), BTD(11-12) and T_{sfc} as inputs (referred to as 15N in the text).

	∂r_{eff} / — [$\mu m/K$]	∂T / — [K/K]	$\partial \tau$ / — [K^{-1}]
∂BT(3.7)	2.04 ± 1.03	-0.42 ± 0.75	-0.78 ± 1.26
∂BT(11)	-3.36 ± 1.22	0.45 ± 1.34	0.37 ± 1.50
∂BTD(11-12)	-6.47 ± 3.30	-3.32 ± 2.34	-5.31 ± 4.17
∂T_{sfc}	1.35 ± 0.36	1.04 ± 0.61	0.49 ± 0.41

Table 4.3: The same as Table 4.1, but for a network containing 30 neurons in the hidden layer and using BT(3.7), BT(11), BTD(11-12) and T_{sfc} as inputs (referred to as 30N in the text).

	∂r_{eff} / — [$\mu m/K$]	∂T / — [K/K]	$\partial \tau$ / — [K^{-1}]
∂BT(3.7)	1.93 ± 4.32	-0.15 ± 1.46	-2.05 ± 10.33
∂BT(11)	-3.19 ± 5.90	0.07 ± 2.53	1.80 ± 16.08
∂BTD(11-12)	-7.11 ± 15.01	-1.20 ± 7.21	-11.96 ± 38.32
∂T_{sfc}	1.34 ± 2.60	1.08 ± 1.20	0.50 ± 7.71

Since, as noted above, I do not have further independent estimates available for comparison, I cannot

[42] The networks discussed here were trained with three hyperparameter re-estimation cycles with 1500 optimisation steps each, unlike the networks that were used to produce Figure 4.12.

judge how reasonable the remaining sensitivities are. In order to gain more insight into the response of the two networks, I analysed the sensitivity of the retrieval to changes in cloud top pressure (or to the pressure difference between cloud top and sea surface, cf. section 2.6). This is important because of the sensitivity of the computed brightness temperature differences to changes in cloud top pressure in the forward model, as discussed in section 2.6. Since p_{ct} is fixed in the LUT, it is necessary to know about the retrieval response to changes in this parameter. Of course, in the forward model, changes in p_{ct} will cause changes in the other cloud properties through the adiabatic assumption. It is hence quite possible that for a cloud with a different p_{ct} encountered in the scene a cloud of similar thickness, particle size and temperature, but different cloud top pressure is contained in the LUT and the desired parameters can be retrieved correctly. Nevertheless, a sensitivity analysis can provide further insight.

This sensitivity of the retrieval to changes in p_{ct} cannot be determined with the Jacobian, since cloud top pressure is not an input variable. I thus repeated the forward model runs that produced the subset of the training LUT plotted in Figure 4.9, but this time the cloud top pressure was changed by ± 5 hPa. Figures 4.13 and 4.14 show scatter plots of the forward propagation of these datasets through 15N and 30N, in both cases compared to the results obtained with the original LUT, fixed at a cloud top pressure of 939.5 hPa.

While the sensitivity of effective radius and cloud top temperature is relatively small and of similar magnitude for both networks ($\pm$ 2 μm for r_{eff} and $\pm$ 0.5 K for T_{ct}), the retrieved optical thickness of the 30N network changes drastically with changes in cloud top pressure (up to ± 4) – in contrast to 15N, where the τ retrieval is much less sensitive. Indeed, for higher cloud tops (decreased p_{ct}), 30N infers negative optical thicknesses for thin clouds. This behaviour might be a possible mechanism for the unphysical 30N retrievals.

Of course, the point estimates of the Jacobian presented in Tables 4.1, 4.2 and 4.3 can only be used to evaluate the sensitivity of the retrieval in the selected area of the LUT. Histograms of the Jacobian, on the other hand, can provide information on the range in which the sensitivity varies over the entire LUT or a given scene. Figure 4.15 shows such histograms of the 15N Jacobian over the database from which Figure 4.9 was created. As expected from Figure 4.9, the sensitivities vary significantly, with the larger magnitudes likely corresponding to the thick and thin clouds.

A possible application of the Jacobian histograms could be to identify pixels for which the retrieving network exhibits an unreasonably large sensitivity. The retrieval could thus be further constrained to pixels for which the sensitivity is in the expected range.

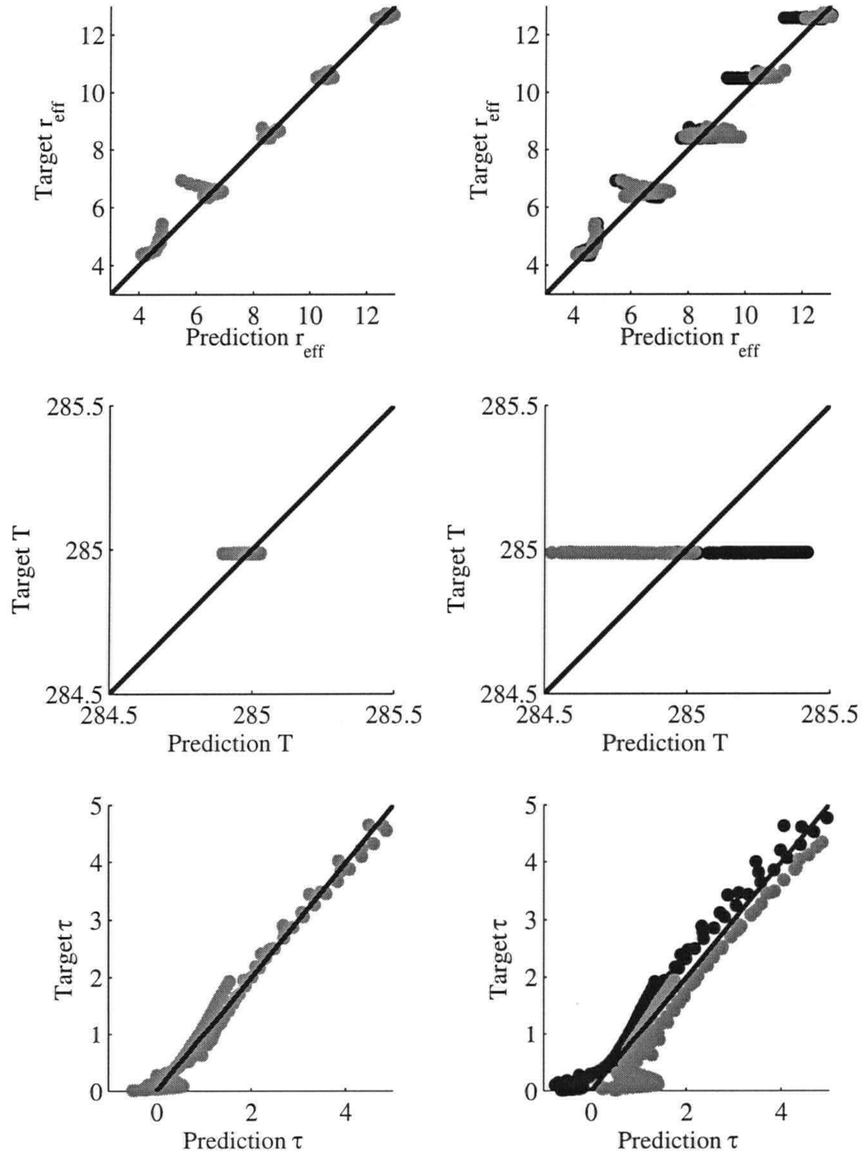


Figure 4.13: Sensitivity of 15N predictions if cloud top pressure (fixed in the LUT) is varied by ± 5 hPa. In the right column, predictions of the unperturbed subset of the training LUT plotted in Figure 4.9 are shown. The left column shows the same network-predicted cloud parameters from LUT subsets in which p_{ct} was perturbed by +5 hPa (grey) and -5 hPa (black). See text for more details.

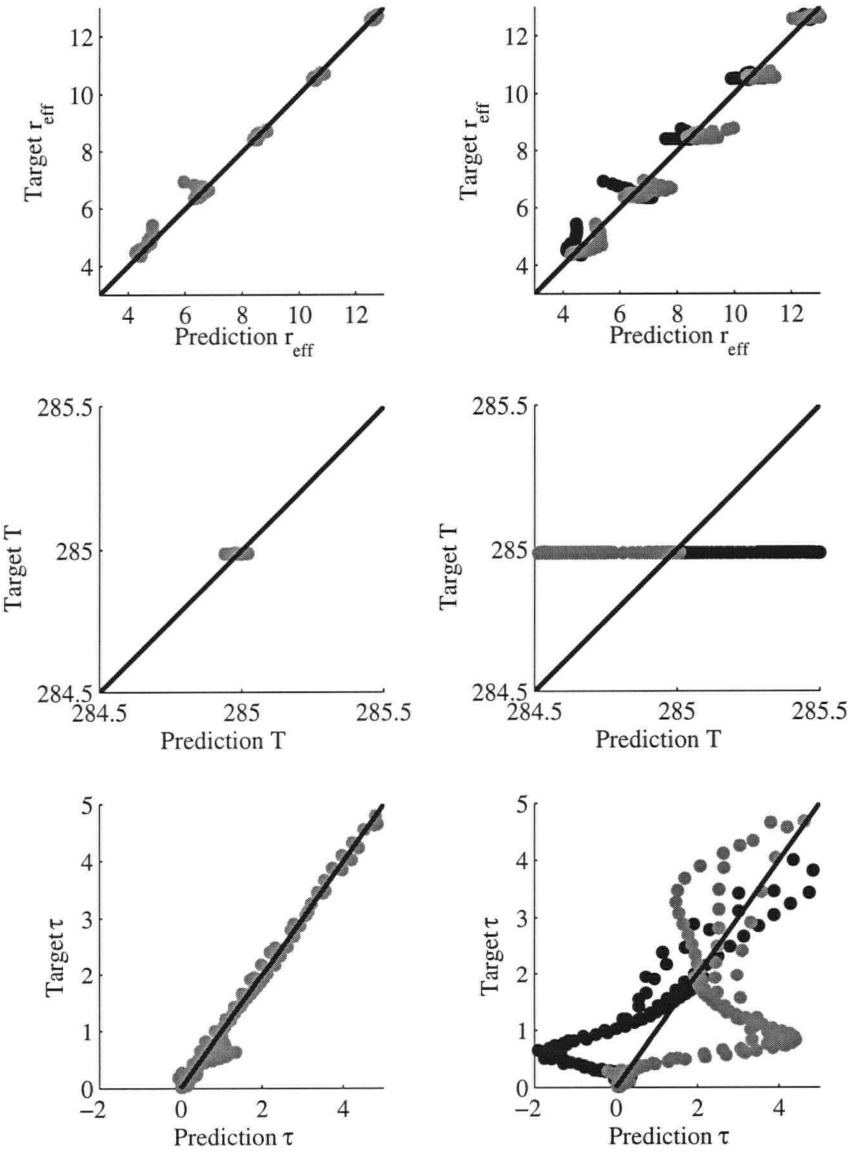


Figure 4.14: The same as Figure 4.13, but for the 30N network.

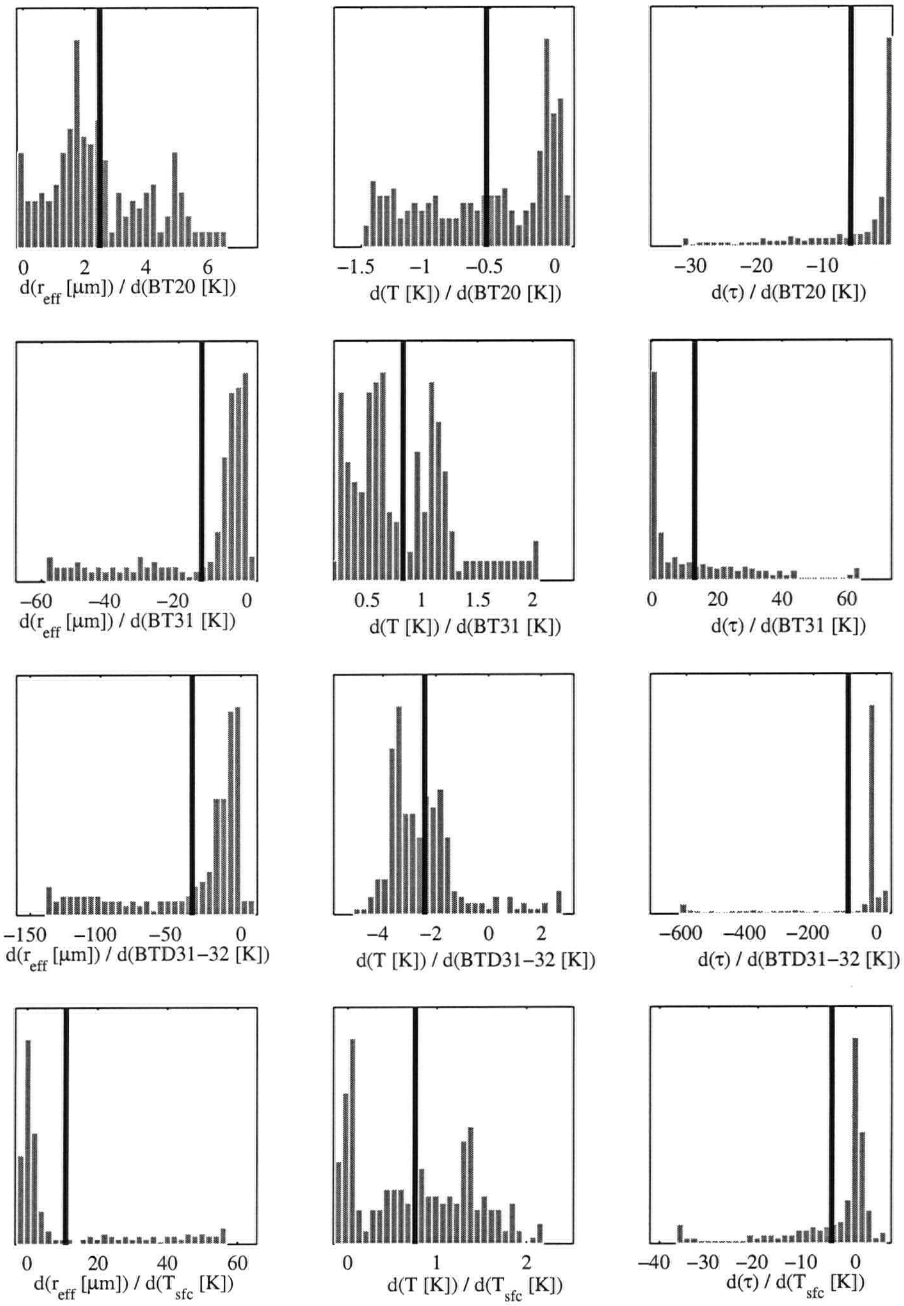


Figure 4.15: Histograms of the 15N Jacobian of the LUT subset plotted in Figure 4.9. Average values are indicated by the black lines.

4.5 Retrieval Evaluation, Jacobian and Uncertainty

4.5.1 Comparison with In-Situ Data

As discussed in the previous section, the 15N network yielded the best retrieval performance. In this section, I will analyse the retrieval by comparing the results to the RF02 in-situ data (cf. section 2.1) and by checking the physical plausibility. Figure 4.16 shows a map of the retrieved r_{eff} values, along with a histogram of the cloud top measurements from RF02 and a histogram of the retrieved values in the area. Since the in-situ measurements were taken about five hours after the satellite overpass, the measured values have been advected and are plotted over the clouds that likely have been observed from the aircraft.

By comparing against advected data I assumed that the clouds in question did not change over the five hours. Also, I assumed that the wind speed and direction at cloud level were constant at the values measured from the aircraft at the time the in-situ measurements were taken. While the cloud layer is both extensive and horizontally homogeneous (so that changes in wind speed and direction still advect clouds with similar characteristics), changes due to the diurnal cycle and precipitation are possible. The cloud top was likely lower earlier in the diurnal cycle when the MODIS image was taken (cf. section 1.2), however, the in the LUT employed cloud top pressure is probably underestimated (I used the aircraft altitude just below cloud top), thereby offsetting this change to some extend. Precipitation, on the other hand, was significant during RF02 (Stevens et al., 2003a, Figure 6), hence, it represents a possible source of uncertainty.

As noted in section 2.1, the DYCOMS-II data were averaged over 10 s intervals in order to yield measurements on the 1 km scale of the MODIS pixels. Due to the time lag between satellite and in-situ observations and the uncertainty in the advection, no collocated observations were possible. I hence chose distributions of the retrieved parameters within the box shown in Figure 4.16 and in-situ measured values from the cloud top flight tracks of RF02 as the means for comparison. Unfortunately, the selected area contains a large number of “irretrievable” pixels.

The range of the retrieved effective radii corresponds well to the range of the in-situ measured values, both ranging from about 8 μm to about 16 μm (Figure 4.16b,c). The shape of the distribution, however, does not agree as well. Whereas the retrieved effective radius peaks at both about 9.5 and 13 μm, the aircraft measured values are more uniformly distributed, with several small peaks and one larger peak located at 11 μm. A possible cause for these discrepancies are changes in the structure of the cloud layer between the satellite and in-situ observations as described above. Also, pixels in the vicinity of "irretrievable" pixels might still be influenced by the mechanisms that cause the anomalous pixels, e.g. overlying cirrus. Furthermore, sea surface temperature was assumed to be constant in the scene. As I will show at the end of this section, the uncertainty in the neural inversion ($G^T \tilde{H}^{-1} G$, cf. section 3.3) is on the order of 2 μm in the selected area; however, as discussed below, the usefulness of this value is questionable.

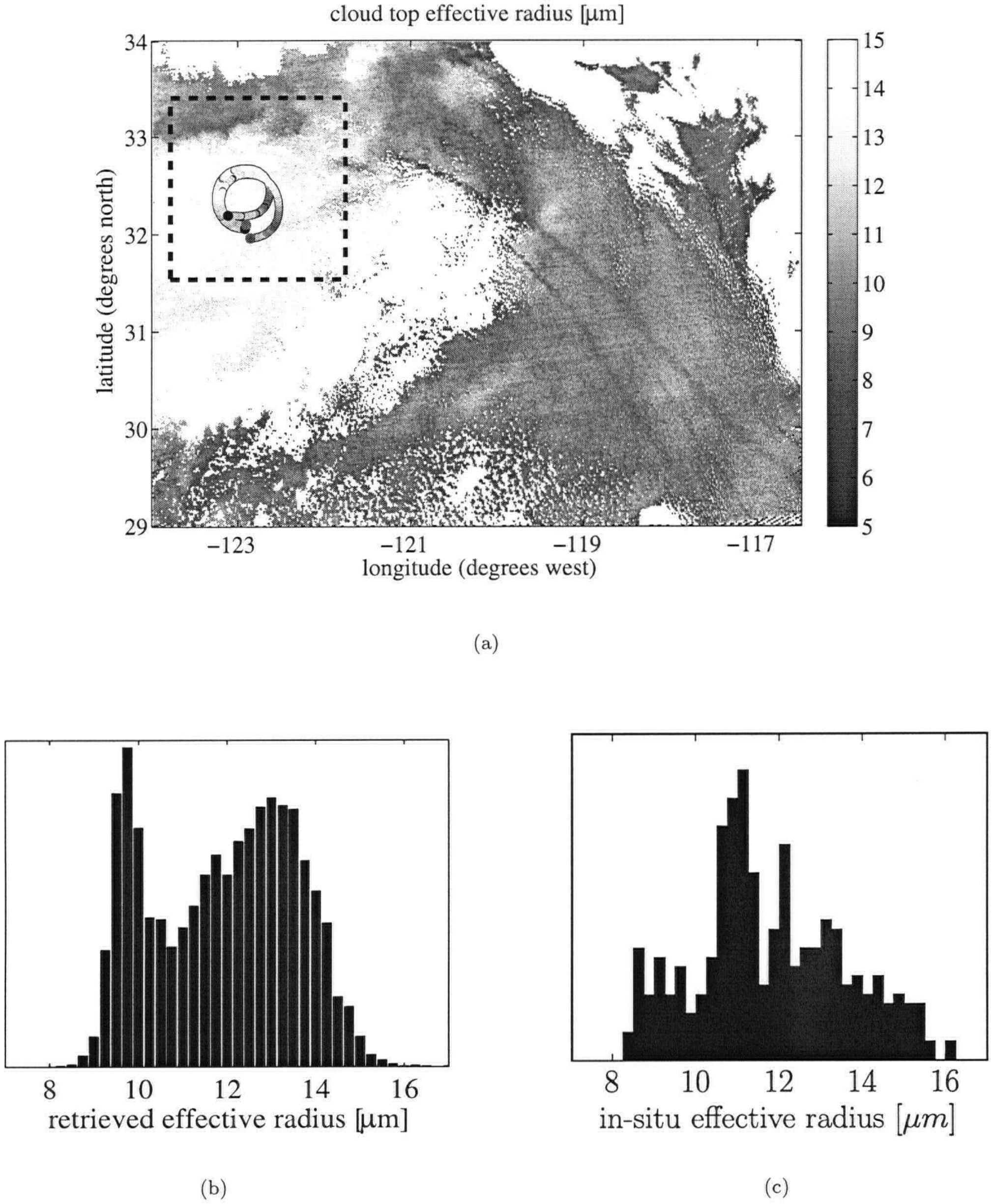
Nevertheless, the retrieved values look physically plausible. The ship tracks are clearly discernible with a decreased particle radius of about 2 μm smaller than particle sizes in the environment of the tracks, which is in the expected range (Schreier et al., 2006). The radii retrieved for the remaining scene also look reasonable for marine stratocumulus (values ranging from 5 to 15 μm, some structure present, but no abrupt changes).

Similar results are obtained for cloud top temperature, shown in Figure 4.17. As expected, the ship tracks are not discernible in the temperature retrieval. Instead, the cloud deck has temperatures varying only slightly between values of about 284 K and 286 K, with smooth transitions and no abrupt changes. The range of the retrieved temperature agrees well with the in-situ measurements, and for this variable the shape of the distribution also agrees well.

The optical thickness of the clouds can only be judged by physical plausibility, since values of τ cannot be inferred from the in-situ measurements along the horizontal flight legs. As Figure 4.18 shows, values of τ vary from less than 1 to about 5, all reasonable values for marine Sc. The ship tracks are thicker by about one to two compared to their environment, and the majority of the clouds are thin ($\tau < 2$).

4.5.2 Jacobian

In section 4.3, I introduced the application of point estimates of the Jacobian to compare the ANN sensitivities with independent estimates. However, the information in the Jacobian can also be used in different ways to analyse the retrieval network. As noted in section 1.5, Aires et al. (2004b) investigated whether a PCA (principal component analysis) preprocessing of the input data can reduce the variability in the Jacobian. To obtain a variability representative of the entire retrieval scene, they computed an average Jacobian and its variability from all pixels contained in the scene. Furthermore, they normalised this mean Jacobian by the standard deviations of the input data (over the retrieval scene) to gain information about the relative importance of the individual inputs. They argue that the normalised Jacobian can be used to refine the inversion procedure by identifying inputs that do not contribute

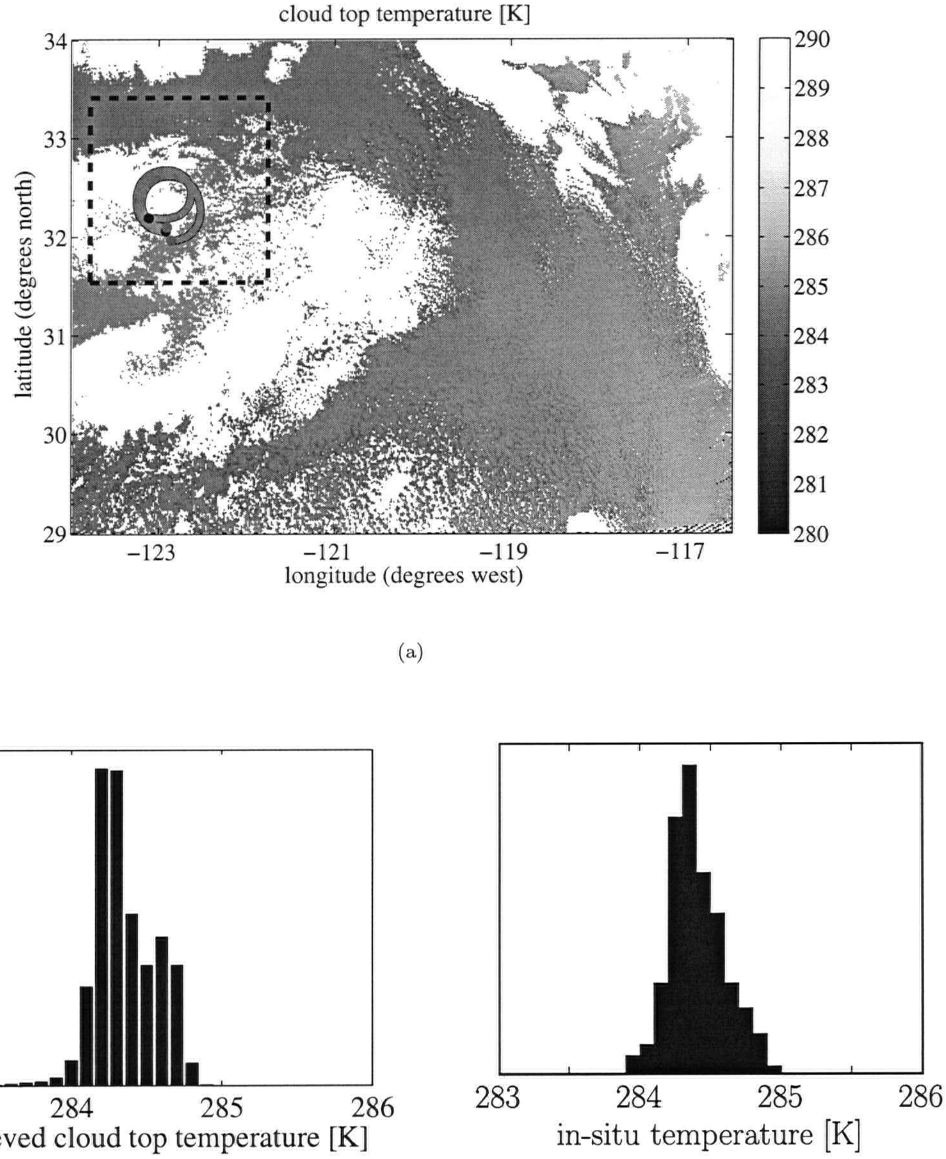


(a)

(b)

(c)

Figure 4.16: Retrieved effective radii for the test scene and comparison of retrieved (15N network) and aircraft measured values. The ship tracks are discernible with smaller droplet sizes than their environment. In-situ measurements of two cloud top flights have been advected and overlain on the retrieved data. Due to the uncertainty in the advection and the large number of irretrievable pixels in the flight area the histogram of in-situ data is compared to a histogram of the surrounding retrieved values.



(a)

(b)

(c)

Figure 4.17: The same as Figure 4.16, but for cloud top temperature.

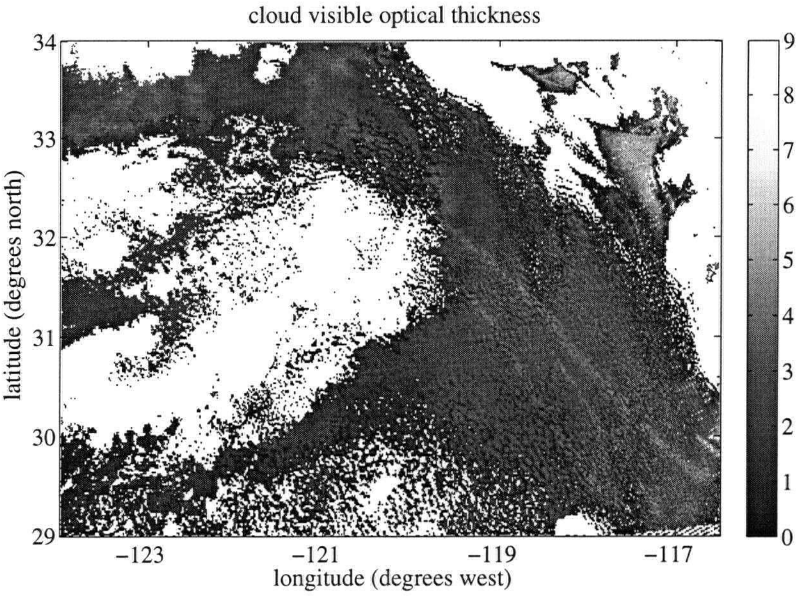


Figure 4.18: The same as Figure 4.16, but for cloud optical thickness. Note that for the optical thickness no in-situ measurements were available.

significantly to the outputs (which could consequently be eliminated).

Table 4.4 shows the average Jacobian of the July 11, 2001, scene. The values are similar to those found for the point estimate in Table 4.2. On average, the dependence of r_{eff} to BT(11) is smaller than in Table 4.2, $\partial\tau/\partial$BT(11) is larger and $\partial r_{eff}/\partial T_{sfc}$ and $\partial\tau/\partial T_{sfc}$ are very small. Cloud top temperature is similarly dependent on both T_{sfc} and BT(11).

Table 4.4: Average Jacobian of the July 11, 2001, scene and its variability (15N network).

	∂r_{eff} / — $[\mu m/K]$	∂T / — $[K/K]$	$\partial\tau$ / — $[K^{-1}]$
∂BT(3.7)	1.47 ± 1.32	-0.22 ± 0.56	-0.94 ± 1.53
∂BT(11)	-1.23 ± 1.37	0.56 ± 0.80	1.07 ± 1.69
∂BTD(11-12)	-7.75 ± 6.11	-2.39 ± 2.32	-5.24 ± 7.86
∂T_{sfc}	-0.16 ± 0.54	0.62 ± 0.43	-0.04 ± 0.82

The normalised mean Jacobian is listed in Table 4.5. The standard deviations of BT(3.7), BT(11) and BTD(11-12) were obtained from the satellite data, while that of T_{sfc} was estimated from the RF02 measurements. The normalisation leads to some interesting results. Effective radius is indeed mainly determined by the BT(3.7) signal, while its dependence on BTD(11-12) becomes more relative. Cloud top temperature still is equally dependent on both BT(11) and T_{sfc}, which also contribute most to this output (BTD(11-12) is less significant). T_{sfc} contributes very little to both r_{eff} and τ, so that this input likely is mainly needed for a correct T_{ct} retrieval (cf. Figure 4.11).

Table 4.5: The same as in Table 4.4, but normalised by the standard deviation of the inputs. This allows for a better judgement of the importance of the individual inputs. The standard deviation of the surface temperature input has been estimated from the DYCOMS-II measurements.

	∂r_{eff} / — [μm]	∂T / — [K]	$\partial \tau$ / — [1]
∂BT(3.7)	2.36	−0.35	−1.51
∂BT(11)	−1.82	0.91	1.74
∂BTD(11-12)	−1.34	−0.42	−0.92
∂T_{sfc}	−0.20	0.78	−0.04

When inferring information about the importance of individual inputs from an averaged Jacobian, it is important that the average indeed is representative of the scene. To verify the representativeness of the Jacobian given in Tables 4.4 and 4.5, I computed histograms of the individual sensitivities of all pixels in the scene, as was done for the LUT subset in Figure 4.15. The histograms for the July 11 scene are displayed in Figure 4.19. The average Jacobian values correspond well with the most often occurring sensitivities, so that the relative Jacobian in Table 4.5 indeed gives a good idea of the information content of the individual inputs.

The last useful representation of the Jacobian that shall be discussed in this thesis is its spatial distribution, as shown in Figures 4.20 and 4.21 for effective radius and cloud top temperature, respectively. The spatial distributions of $\partial r_{eff}/\partial$BT(3.7) and $\partial r_{eff}/\partial$BTD(11-12) in Figure 4.20 show a coherent picture. The absolute magnitudes of both sensitivities decrease across the ship tracks, which is expected from Figure 4.9 due to the increased space between the curves for clouds with effective radii of approximately 10 μ and optical thicknesses of about 3. Similarly, the sensitivities for the thin clouds in the area that served for comparison with the in-situ data exhibit a much larger absolute magnitude – consistent with the converging lines in Figure 4.9 for thin clouds.

A curious feature is that the negative dependence of T_{ct} on BT(3.7) decreases in magnitude with increasing τ (Figure 4.21, cf. Figure 4.18). At $\tau \approx 3.5$, the dependence becomes positive. At the same time, the positive $\partial T_{ct}/\partial$BT(11) decreases with increasing τ. This sensitivity is largest for thin clouds. This behaviour can likely be attributed to the transition between thin clouds that influence the surface-emitted radiation very little and thick clouds that emit approximately as black bodies in the thermal infrared.

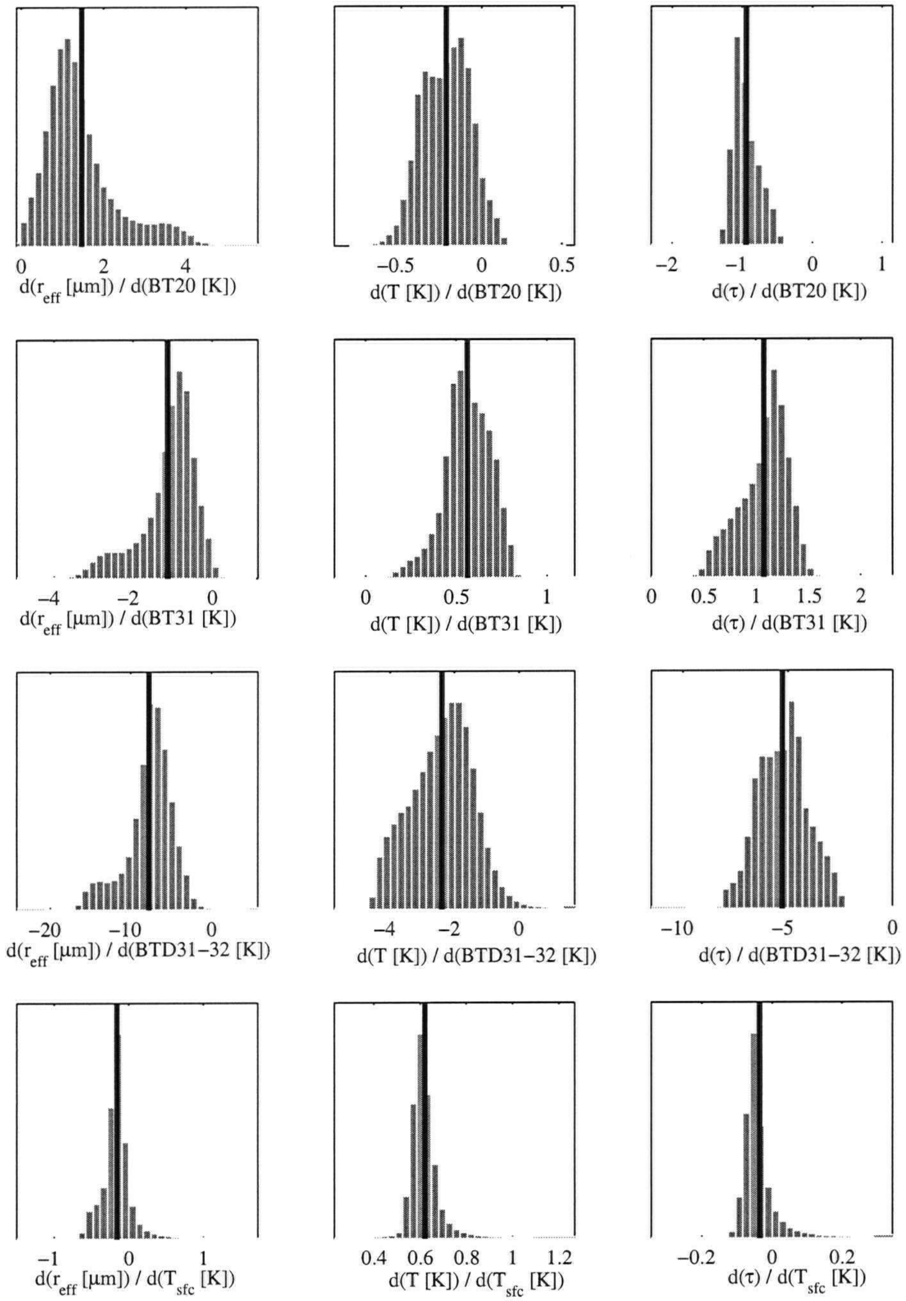


Figure 4.19: Histograms of the 15N Jacobian of the July 11, 2001, scene. The average values as listed in Table 4.4 are highlighted by the black lines.

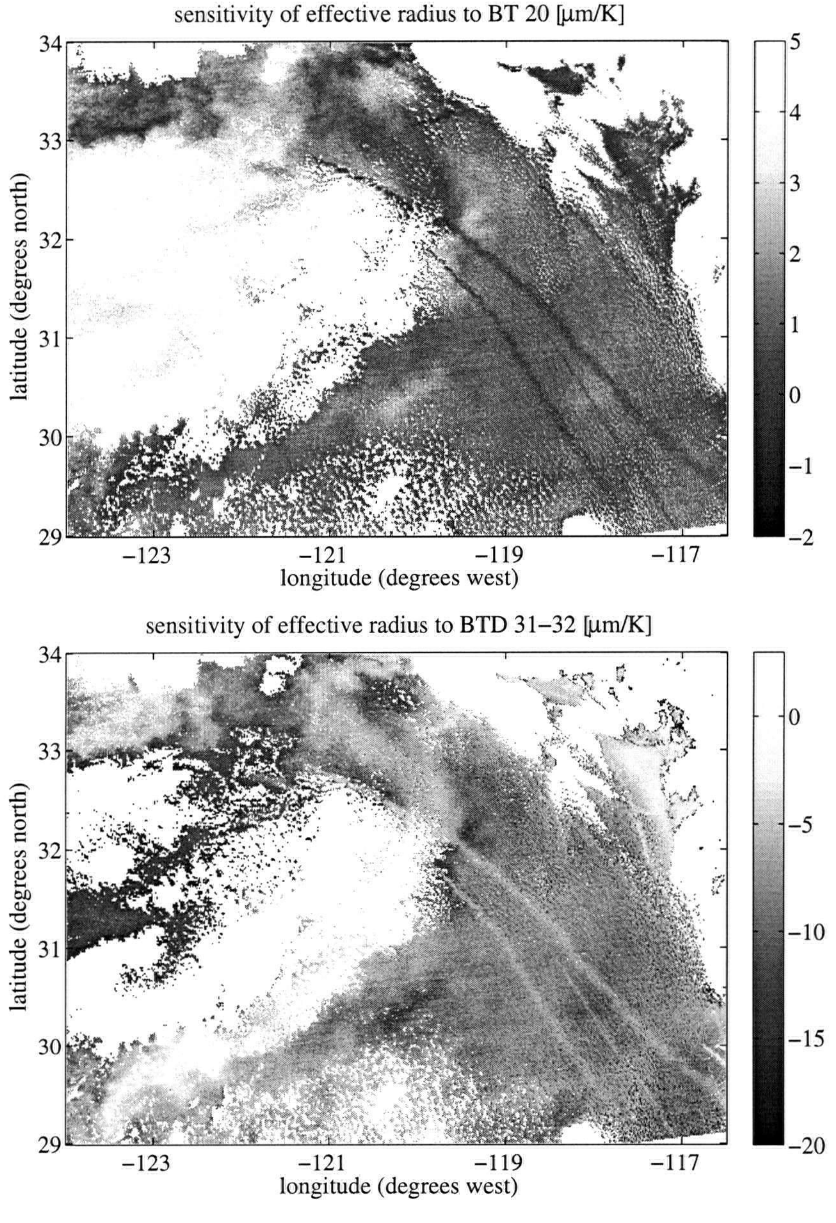
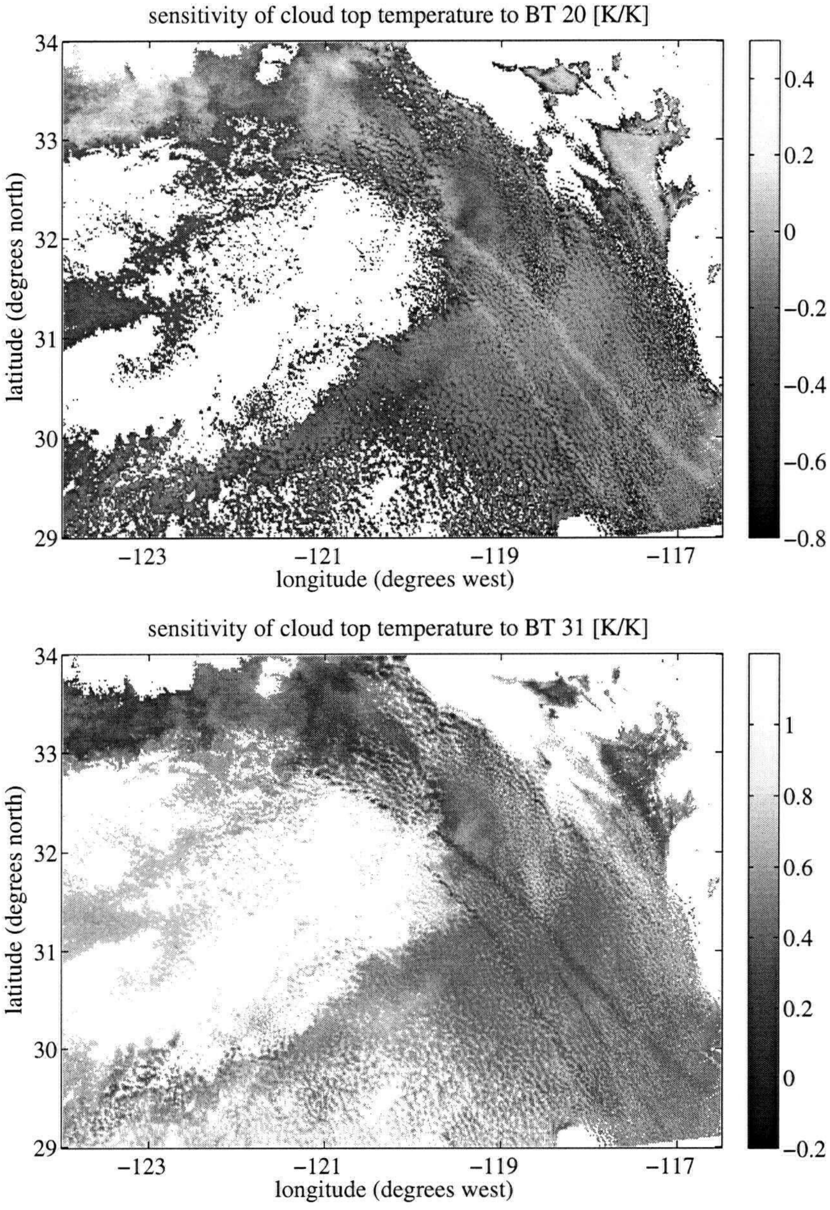


Figure 4.20: Spatial distribution of the sensitivity of the effective radius retrieval to changes in BT(3.7) (top panel) and BTD(11-12) (bottom panel) on July 11, 2001 (15N network).



sensitivity of cloud top temperature to BT 20 [K/K]

sensitivity of cloud top temperature to BT 31 [K/K]

Figure 4.21: The same as Figure 4.20, but for the sensitivity of cloud top temperature to changes in BT(3.7) (top panel) and BT(11) (bottom panel).

Since p_{ct} was held constant in the LUT, the dependence of T_{ct} on T_{sfc} should be approximately 1 for very thin clouds ($\tau < 1$) – the cloud layer has little impact on the radiances, and due to the linear temperature decrease with height changes in T_{sfc} lead to immediate changes in T_{ct}. The sea surface also approximately emits as a black body in the infrared (cf. the subcloud layer in Figure 2.10), thus the dependence of T_{ct} on BT(11) is also expected to be close to 1. Indeed, the sensitivity of T_{ct} on T_{sfc} is also largest for thin clouds (not shown).

If the clouds have reached an optical thickness large enough to emit approximately as black bodies at 11 μm (e.g. $\tau \approx 43$ in the right panel of Figure 2.10), $\partial T_{ct}/\partial \mathrm{BT}(11)$ is again expected to be about 1. In between, however, BT(11) is influenced through absorption in the cloud. Independently of T_{ct}, the optically thicker the cloud, the smaller BT(11) as the cold cloud top becomes more opaque (cf. the left panel of Figure 2.10, with $\tau \approx 4$). Similar effects are expected to impact the BT(3.7) signal.

While I cannot verify this argumentation from the BT/BTD diagrams computed from my LUT (cloud top temperature is fixed in all plots), the corresponding plot by Pérez et al. (2000) in Figure 1.9b confirms the dependence of T_{ct} on BT(11) for thick clouds (note that the surface temperature is constant in this plot). This example shows how valuable the Jacobian is in interpreting the behaviour of the retrieval ANN. Since the creation of diagrams similar to Figure 4.9 but for varying T_{ct} is straightforward, I recommend a more detailed analysis of the spatial distribution of the Jacobian in the future.

4.5.3 Uncertainty

Although the estimation of the intrinsic noise covariance matrix $\boldsymbol{C_{in}}$ failed during the training process, the neural uncertainty term $\boldsymbol{G}^T\boldsymbol{H}^{-1}\boldsymbol{G}$ could still be computed and provides an estimate of the uncertainty in the retrieval due to the weights distribution. Figure 4.22 shows maps of the uncertainty (standard deviation computed from the variance in the diagonal elements of the matrix) for effective radius and cloud optical thickness. The uncertainty in r_{eff} (τ) ranges from less than 1 μm (1) in the area of the ship tracks to more than 2 μm (4) in the upper left corner of the scene.

However, based on my findings in Chapter 3, I question the usefulness of these uncertainty estimates. The larger uncertainty in the upper left corner of the scene is correlated with neither of the retrievals – there are no significantly larger or smaller effective radii, warmer or colder or optically thicker or thinner clouds in this area than in the remainder of the scene. As discussed in section 3.7, ambiguities in the LUT cannot be recognised by the neural network. Hence, the only reason for the increased uncertainty would be a lower data density of the type of clouds occurring in the area in question in the LUT – possibly the setup of my forward model caused the cloud type in the upper left corner of the scene to be more sparsely represented in the LUT compared to the other clouds in the scene. Further research is needed to clarify the meaning of the uncertainties.

4.6 Further Developments

In this chapter, I analysed the retrieval results of the 15N network because it yielded the most plausible retrieval performance. As noted, it did not produce the smallest MSE. Given the instabilities in the training process, it is likely that the same network architecture, trained from a different weights initialisation would not perform as well. I hence attribute the relatively good results to "luck" rather than a well functioning method. It is likely that a network containing a larger number of hidden units could also approximate the inverse function correctly. Possibly such a network would produce a smaller MSE and exhibit an improved retrieval performance. However, given that the inverse problem is very ill-posed, more hidden neurons imply that it becomes more difficult to find the correct mapping.

Due to the various problems I encountered, I was not able to give ultimate answers to the questions that arose in this thesis. Instead, the work that I presented should be taken as a first step to find a good and reliable network architecture suited for the given inverse problem. I demonstrated the high potential of neural networks for application in remote sensing, and explored several techniques that can be used to construct a stable retrieval scheme. However, my work also showed the need for further investigations.

Given the results presented in this chapter, I propose, in roughly descending order of importance, to continue work in the following areas:

Stabilisation of the Jacobian In order to make the training process less dependent on the network architecture and the initialisation of the weight values, it is necessary to decrease the variability in the Jacobian and make the problem less ill-conditioned. The PCA approach suggested by Aires et al. (2004b) is not useful for this problem, since it involves reducing the number of inputs – feasible only if a large number of inputs are involved.

Krasnopolsky (2007) notes alternative methods that aim at obtaining a Jacobian with a low uncertainty for data assimilation purposes, such as training a separate ANN to represent the Jacobian. However, while such methods might be able to provide a good estimate of the physical Jacobian, they say nothing about the Jacobian of the retrieval network. In his work, Krasnopolsky (2007) suggests an approach based on ensembles of ANNs. Several networks of the same architecture are

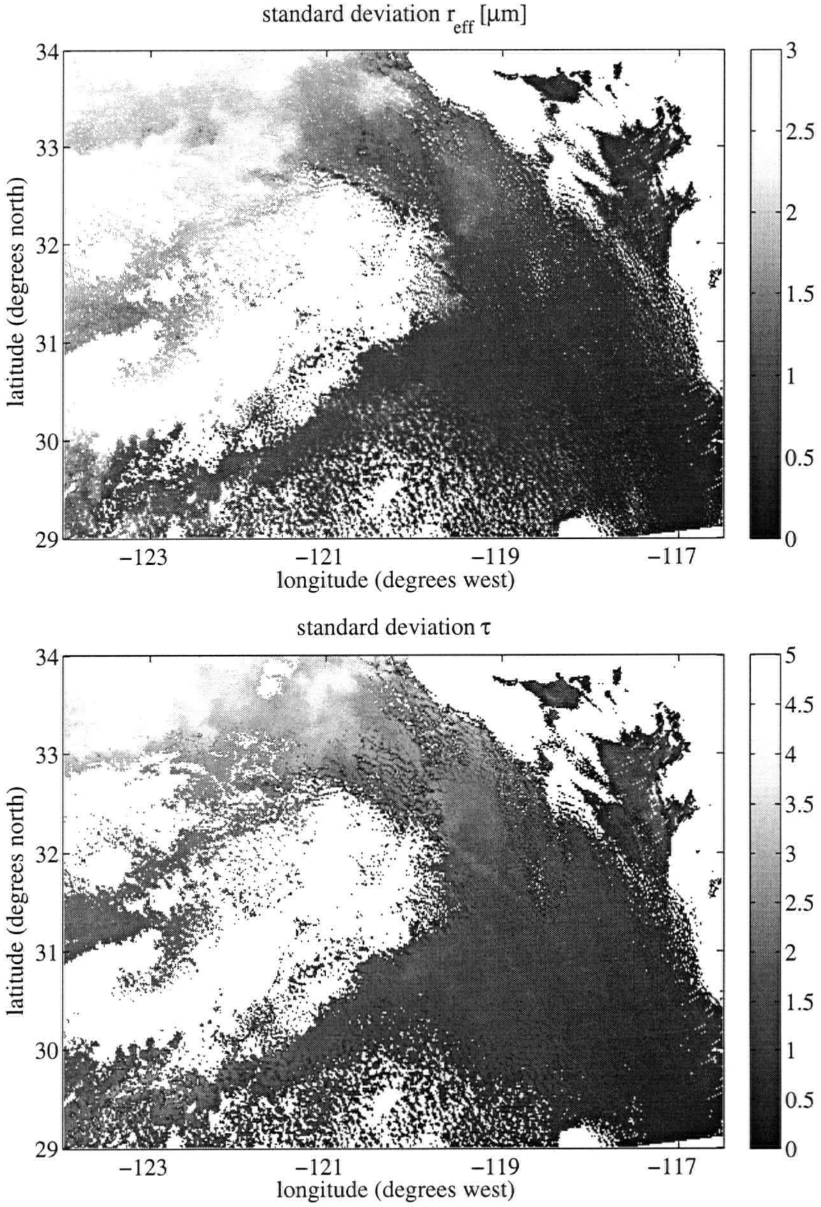


Figure 4.22: Spatial distribution of the uncertainty in the effective radius (top panel) and optical thickness (bottom panel) retrievals, as computed from the neural uncertainty term $\boldsymbol{G}^T\boldsymbol{H}^{-1}\boldsymbol{G}$.

trained using differently perturbed initial conditions for the ANN weights, so that each network will likely exhibit different weight values after the training. The average of all networks is then used to predict the outputs and estimate the Jacobians.

Of course, such an approach implies increased computational effort for the network training. However, since the number of inputs cannot be reduced in my retrieval problem, the method could lead to improvements.

Optimisation of the Network Architecture Once the Jacobian is more stable, the network architecture can be optimised. This can be continued to be done by training a restricted class of architectures and comparing both their validation MSE and retrieval performance. For comparison, it might be worthwhile to test whether three-layer networks, such as employed by Cerdeña et al. (2007), perform better than the two-layer architectures used in this thesis. An alternative for finding the optimal number of hidden units would be to adopt the genetic algorithm approach proposed by Cerdeña et al. (2007). Also, Bishop (1995, Chapters 9 and 10) discusses further model selection tools.

In my work, I restricted the number of training cycles during network training due to the high computational cost. However, if a stabilised Jacobian allows for a more structured exploration of different network architectures it should be ensured that the training process is long enough to find the global minimum in the weights error surface.

Retrieval Evaluation The retrieval evaluation can be extended and improved in several ways. First, it would be desirable to compute accurate point estimates of the physical Jacobian from the LUT, as discussed in section 4.4. This would significantly increase the usefulness of ANN-estimated point Jacobians.

Next, the comparison of retrieved and in-situ observed values by means of histograms is not very satisfactory. More work should be invested into applying the retrieval to other scenes – possibly in which satellite overpass and in-situ measurements are closer together. The DYCOMS-II campaign provides further scenes, as does the EPIC campaign (Bretherton et al., 2004). In addition to the in-situ data, the ANN retrievals could also be compared to precedent and subsequent daytime retrievals from independent sources (e.g. the operational MODIS product).

Also, it would be useful to know precisely what caused the irretrievable pixels. If computed correctly in the forward model, the 8.5 μm channel could help to identify high clouds (Baum et al., 2003), and the operational MODIS algorithms also provide information about cloud height (Ackerman et al., 1998).

For comparing the obtained retrieval results with those of other nighttime retrieval schemes (e.g. Cerdeña et al., 2007) and for estimating the true impact of the cloud layer on the upwelling thermal flux, it would be desirable to implement the computation of the 11 μm optical thickness into the forward model.

Sensitivities to Assumptions in the Forward Model It is also important to explore the sensitivities of the retrieval to the assumptions made in the forward model. A sensitivity study to k-value and ν_{gam} (cf. sections 2.2 and 2.3) is difficult, since in order to determine the precise effects, for each new value a new LUT would have to be computed in the current setup, and a new network would have to be trained. However, it would be possible to compute the changes in the computed TOA BTs due to changing k and ν_{gam}, and to estimate the impact on the retrieval via the Jacobian. The same is valid for the sensitivity of the retrieval to subadiabatic clouds, and the impact on TOA BTs caused by broken cloud regimes should also be investigated.

In my forward model, I assumed that the satellite is directly above the cloud (cf. section 2.6). In many cases, this might not be a good assumption (in fact, I did not check whether it was fulfilled in the July 11 case). Hence, if a sensitivity analysis shows a large impact of a slanted radiation path on the BTs observed by the satellite, the satellite zenith angle should be considered in the forward model.

Overlying Atmosphere In order to make the retrieval independent of a prescribed overlying atmosphere, the optical properties of the OA could be introduced as variable parameters t^* and B^*, such as suggested in section 2.5. Given the ease of including additional inputs into the ANN architecture, an alternative could be to obtain independent estimates of the water vapour path in the atmosphere (cf. section 2.5) and use it as an additional input. The network could then automatically infer the bulk absorption effects. Also, it could be investigated how much information is gained if the brightness temperatures of the clear sky pixels are obtained for all channels, such as done by Pérez et al. (2000) and Cerdeña et al. (2007).

Uncertainties and Ambiguities It would be interesting to check whether the type of cloud in the upper left region of the scene is actually underrepresented in the LUT, as discussed in section 4.5. This information could help to refine the forward model so that the data density is similar for all types of clouds.

It is unsatisfying that the Aires et al. (2004a) method is not able to recognise ambiguous situations. Another analysis of the LUT should be performed to estimate the actual magnitude of the occurring ambiguities and to determine the regimes in which they occur (for instance, thin clouds vs. thick clouds). In order to incorporate the uncertainty due to ambiguities into the retrieval, I propose the following. First, if we consider the ambiguities as being a part of the intrinsic noise, we could discretise the input space and compute localised noise matrices $\boldsymbol{C}'_{in}$ (where $'$ indicates the localisation). By assuming that the network mapping is correct, the covariance matrix of the errors $\boldsymbol{C}'_0$ in the selected input area could be used to approximate $\boldsymbol{C}'_{in}$ (thereby either ignoring the neural uncertainty term, or, if stable enough, considering localised versions of $\boldsymbol{G}^T\tilde{\boldsymbol{H}}^{-1}\boldsymbol{G}$ as well).

An alternative could be to employ a different network architecture known as *mixture models* (Bishop, 1995, Chapter 6). Mixture models are able to compute multimodel output distributions; they are therefore not restricted by the Gaussian assumption as the Aires et al. (2004a) method (cf. section 3.3). While this could represent an approach with a high potential to improve some of the difficulties encountered in my work, the Bayesian methods applied in this thesis could not be directly applied to mixture models. Hence, this approach would require an increased effort.

A comprehensive uncertainty estimate requires the combination of the errors of all contributing sources. If the above problems are solved, such an estimate can be obtained by combining network uncertainty, ambiguities, uncertainties due to assumptions in the forward model and instrument noise of the MODIS instrument, which can be converted to output error with the Jacobian.

Aires et al. Implementation Finally, Chapter 3 raised many questions concerning the implementation and application of the Aires et al. (2004a) method. Whether we can improve the unsatisfying need to regularise the Hessian and whether the problems with the hyperparameter re-estimation can be solved are issues that should also be addressed in the future.

Chapter 5

Summary

In this thesis I have investigated the feasibility of retrieving cloud top effective radius, cloud optical thickness and cloud top temperature of nocturnal marine stratocumulus clouds by inverting infrared satellite measurements using an artificial neural network.

In Chapter 1, I described the scientific context of my work. Marine Sc play a critical role in the exchange of energy and water in our climate system. However, processes that are involved in cloud-atmosphere interaction and their representation in general circulation models remain some of the primary uncertainties in global climate modelling. Observational data that can shed light on regional variations in cloud microphysical and optical properties, their diurnal cycle and the interaction of clouds, precipitation and aerosols is important for further progress in the accurate representation of marine Sc in climate and weather models.

A few studies have investigated the nocturnal retrieval problem and shown that it is possible to infer cloud properties from the information contained in the infrared channels centred at 3.7, 8.5, 11.0 and 12.0 μm. However, problems with the standard optimisation approach to these retrievals include high computational cost and a lack of consistent error estimates. The aim of this study was to use neural networks to design a retrieval method that is both fast and able to give such uncertainty estimates. The fundamental idea behind this approach is to approximate the inverse of the forward function that describes the dependence of top-of-atmosphere radiances on cloud parameters with a neural network architecture that is capable of solving nonlinear regression problems.

Cerdeña et al. (2007) were the first to use neural networks to invert nocturnal measurements of the AVHRR instrument. The objective of this study was to extend their approach by investigating the applicability of methods proposed by Aires (2004) and Aires et al. (2004a,b) – allowing for the estimation of uncertainties and sensitivities – to the problem, to apply the retrieval scheme to nocturnal measurements of the higher resolution MODIS instrument, and to compare the retrieved values to in-situ measurements obtained during the DYCOMS-II field campaign off the coast of California. My focus lay on the uncertainty in the retrieval, and on what we can learn from retrieval sensitivities. My initial work included the construction of a forward model capable of computing top-of-atmosphere brightness temperatures for varying cloud parameters and the implementation of the Aires et al. method.

The topic of Chapter 2 was the theory and implementation of the forward model. The scene of July 11, 2001, was chosen for the retrieval, and the in-situ aircraft measurements obtained on that day were used for the development of the forward model and the evaluation of the retrieval performance. The scene contains a number of ship tracks that provide contrasting droplet sizes which were used to check the physical consistency of the retrieval.

The clouds were modelled as adiabatic plane parallel cloud layers in the forward model and the droplet size distribution was described by a modified gamma distribution. Comparisons of idealised adiabatic profiles and size distributions with in-situ measured data showed good agreement. The radiative transfer model libRadtran (Mayer and Kylling, 2005), which incorporates the multiple scattering code DISORT (Stamnes et al., 1988) to solve the radiative transfer equation, was used to compute cloud top radiances. To account for the spectral intervals of the MODIS instrument channels, I implemented correlated-k code developed by Kratz (2001) into libRadtran. Gaseous absorption and emission above cloud top were accounted for by incorporating average transmission and emission properties of the atmosphere obtained from radiosonde soundings. The forward model proved capable of reproducing relationships between cloud top brightness temperature differences and cloud parameters that were previously described by Baum et al. (1994) and Pérez et al. (2000).

In Chapter 3, I further explored theory and implementation of the method proposed by Aires (2004) and Aires et al. (2004a,b). The method ideally provides estimates of the uncertainty arising from both the neural network fit to the inverse function and the intrinsic noise inherent in the data, as well as the variability in the Jacobian that is due to the network fit. The Jacobian, describing the sensitivities of the outputs with respect to the inputs, is particularly important for analysing the dependences of the network fit in order to ensure that the network models the "correct" function. Its variability indicates how ill-conditioned the regression problem is, a common problem with inverse problems that makes it difficult to find a good approximation to the inverse function.

Unfortunately, I encountered several difficulties with the Aires et al. method that limited its usefulness for the retrieval problem. Initial tests with simple examples showed questionable results, with uncertainty intervals often not including the true value. For instance, the method was not able to recognise ambiguities that were expected to occur in the lookup table. Furthermore, numerical problems involved in the estimation of the Hessian matrix of the network in some cases led to a failure in the estimation of the uncertainty due to the intrinsic noise in the data and questionable results for the uncertainty due to the network fit. The estimation of the Jacobian and its variability, however, proved promising.

In Chapter 4, I applied the methods described in Chapters 2 and 3 to the retrieval scene. A lookup table consisting of 96,000 cloud profiles with varying effective radius, droplet number concentration and temperature was computed. Parameter ranges were obtained from the in-situ measurements, and, for simplicity, cloud top and surface pressure were held constant at the observed values. An average droplet size distribution width representative of the night was determined from the measurements. The brightness temperatures and cloud parameters in the lookup table were used to train different architectures of neural networks. I explored several configurations with different inputs and numbers of hidden units in the network and compared their results based on training error, sensitivities and physical plausibility. I restricted myself to network architectures containing one layer of hidden neurons.

The major results of the retrieval investigations can be summarised as follows:

- Comparison of the computed brightness temperatures to those observed by MODIS showed that a large number of observations were outside the computed range. Many of these "irretrievable pixels" could be attributed to broken clouds and clear sky pixels; for the remaining pixels I found indications of overlying cirrus clouds.

- Unfortunately, the computations of the 8.5 μm brightness temperatures did not agree with the observations. An error in the forward model seemed to be the likely cause, hence, the corresponding channel could not be used in the retrieval.

- The uncertainty estimation of Aires et al. (2004a) failed for all networks trained from the lookup table. It is possible that there was only little intrinsic noise in the lookup table, which could have amplified the numerical problems noted above. While I was able to obtain uncertainty estimates due to the network fit with a regularised Hessian, I question the usefulness of the obtained values.

- The brightness temperatures at 3.7, 11 and 12 μm alone were not sufficient for retrieving effective radius, cloud optical thickness and cloud temperature. All networks employing only these inputs showed unphysical results in at least one output variable.

- The Jacobian and its variability proved to be a valuable tool for analysing the networks. Point estimates of the network Jacobian were compared with estimations of the physical Jacobian obtained from the lookup table, the average Jacobian of the satellite scene gave information about the average information content of the network inputs and the ill-conditioning of the problem, and maps of the Jacobian showed the spatial distribution of the sensitivities and were interpreted for physical plausibility.

- Obtaining a good network fit to the inverse function proved to be a highly ill-conditioned problem. This led to a very unstable learning process and the high variability in the Jacobian made it difficult to find a network modelling the expected dependences.
- Almost all network architectures exhibited unphysical behaviour in at least one output variable. This included networks employing sea surface temperature as an additional input to the satellite observations and networks employing brightness temperature differences instead of brightness temperatures as inputs.
- A network that predicted reasonable results employed the brightness temperatures at 3.7 and 11 μm, the brightness temperature difference between 11 and 12 μm and sea surface temperature as inputs and included 15 units in the hidden layer. Satellite image and in-situ measurements were recorded with a five hour difference; histograms of advected cloud properties with histograms of the in-situ measurements showed good agreement of cloud top temperature, ranges of observed and retrieved effective radii also agreed well. Retrievals of all three retrieved parameters were physically plausible, as was the Jacobian.

I have demonstrated that it is feasible to use artificial neural networks for the retrieval of nocturnal marine stratocumulus properties. My results are promising in as far as that despite the difficulties I encountered a good agreement between retrieved and observed data was obtained. In particular the Jacobian proved to be a very valuable tool that has not been employed in the investigations of other authors. However, further refinements and analysis of the method are required. I discussed open questions and suggested areas for continuing work in the conclusions of Chapter 4.

In conclusion, I believe that it is essential to improve the ill-conditioning of the inverse problem in order to achieve a more stable training process and improved retrieval results. My work provides a foundation for future work, but further research is required to provide a reliable, stable method that can provide accurate uncertainty estimates.

Bibliography

Abdi, H., 2007: *Encyclopedia of Measurement and Statistics.* chapter Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition (GSVD). Sage, Thousand Oaks (CA).

Ackerman, A. S., M. P. Kirkpatrick, D. E. Stevens and O. B. Toon, 2004: The impact of humidity above stratiform clouds on indirect aerosol climate forcing. *Nature*, **432**(7020), 1014–1017.

Ackerman, S. A., K. I. Strabala, P. W. Menzel, R. A. Frey, C. C. Moeller and L. E. Gumley, 1998: Discriminating clear sky from clouds with MODIS. *Journal of Geophysical Research*, **103**(D24), 32141–32158.

Aires, F., 2004: Neural network uncertainty assessments using Bayesian statistics with application to remote sensing: 1. Network weights. *Journal of Geophysical Research*, **109**, D10303.

Aires, F., A. Chédin, N. A. Scott and W. B. Rossow, 2002: A regularized neural net approach for retrieval of atmospheric and surface temperatures with the iasi instrument. *Journal of Applied Meteorology*, **41**, 144–159.

Aires, F., C. Prigent and W. Rossow, 2004a: Neural network uncertainty assessments using Bayesian statistics with application to remote sensing: 2. Output errors. *Journal of Geophysical Research*, **109**, D10304.

Aires, F., C. Prigent and W. Rossow, 2004b: Neural network uncertainty assessments using Bayesian statistics with application to remote sensing: 3. Network Jacobians. *Journal of Geophysical Research*, **109**, D10305.

Aires, F., C. Prigent, W. B. Rossow and M. Rothstein, 2001: A new neural network approach including first guess for retrieval of atmospheric water vapor, cloud liquid water path, surface temperature, and emissivities over land from satellite microwave observations. *Journal of Geophysical Research*, **106**, 14887–14908.

Albrecht, B., 1989: Aerosols, cloud microphysics, and fractional cloudiness. *Science*, **245**, 1227–1230.

Albrecht, B. A., D. A. Randall and S. Nicholls, 1988: Observations of marine stratocumulus clouds during FIRE. *Bulletin of the American Meteorological Society*, **69**(6), 618–626.

Anderson, G. P., S. A. Clough, F. X. Kneizys, J. H. Chetwynd and E. P. Shettle, 1986: AFGL atmospheric constituent profiles (0-120km). AFGL-TR-86-0110, Air Force Geophys. Lab, Hanscom Air FOrce Base, Bedford, Mass.

Arduini, R. F., P. Minnis, Smith, J. K. Ayers, M. M. Khaiyer and P. Heck, 2005: Sensitivity of satellite-retrieved cloud properties to the effective variance of cloud droplet size distribution. in *Fifteenth Atmospheric Radiation Measurement (ARM) Science Team Meeting, Daytona Beach, FL (US), 03/14/2005-03/18/2005.*

Arking, A. and J. D. Childs, 1985: Retrieval of cloud cover parameters from multispectral satellite images. *Journal of Applied Meteorology*, **24**, 322–334.

Austin, P. H., Y. Wang, R. Pincus and V. Kujala, 1995: Precipitation in stratocumulus clouds: Observational and modeling results. *Journal of Atmospheric Sciences*, **52**, 2329–2352.

Barnes, W. L., T. S. Pagano and V. V. Salomonson, 1998: Prelaunch characteristics of the moderate resolution imaging spectroradiometer (MODIS) on EOS-AM1. *Geoscience and Remote Sensing, IEEE Transactions on*, **36**(4), 1088–1100.

Baum, B. A., R. F. Arduini, B. A. Wielicki, P. Minnis and S. C. Tsay, 1994: Multilevel cloud retrieval using multispectral hirs and avhrr data: Nighttime oceanic analysis. *Journal of Geophysical Research*, **99**, 5499–5514.

Baum, B. A., R. A. Frey, G. G. Mace, M. K. Harkey and P. Yang, 2003: Nighttime multilayered cloud detection using MODIS and ARM data. *Journal of Applied Meteorology*, **42**, 905–919.

Bishop, C. M., 1994: Neural networks and their applications. *Review of Scientific Instruments*, **65**, 1803–1832.

Bishop, C. M., 1995: *Neural Networks for Pattern Recognition.* Oxford Univ. Press.

Bishop, C. M., 2006: *Pattern Recognition and Machine Learning.* Springer.

Blaskovic, M., R. Davies and J. B. Snider, 1991: Diurnal variation of marine stratocumulus over San Nicolas island during July 1987. *Monthly Weather Review*, **119**(6), 1469–1478.

Bohren, C. F. and E. Clothiaux, 2006: *Fundamentals of Atmospheric Radiation: An Introduction with 400 Problems.* Wiley-VCH.

Bohren, C. F., J. R. Linskens and M. E. Churma, 1995: At what optical thickness does a cloud completely obscure the sun? *Journal of Atmospheric Sciences*, **52**, 1257–1259.

Bony, S., R. Colman, V. M. Kattsov, R. P. Allan, C. S. Bretherton, J. L. Dufresne, A. Hall, S. Hallegatte, M. M. Holland, V. Ingram, D. A. Randall, B. J. Soden, G. Tselioudis and M. J. Webb, 2006: How well do we understand and evaluate climate change feedback processes? *Journal of Climate*, **19**(15), 3445–3482.

Bony, S. and J. L. Dufresne, 2005: Marine boundary layer clouds at the heart of tropical cloud feedback uncertainties in climate models. *Geophysical Research Letters*, **32**, 20806+.

Brenguier, J. L., H. Pawlowska, L. Schüller, R. Preusker, J. Fischer and Y. Fouquart, 2000: Radiative properties of boundary layer clouds: Droplet effective radius versus number concentration. *Journal of Atmospheric Sciences*, **57**, 803–821.

Bretherton, C. S., T. Uttal, C. W. Fairall, S. E. Yuter, R. A. Weller, D. Baumgardner, K. Comstock, R. Wood and G. B. Raga, 2004: The EPIC 2001 stratocumulus study. *Bulletin of the American Meteorological Society*, **85**, 967–977.

Cerdeña, A., A. González and J. C. Pérez, 2007: Remote sensing of water cloud parameters using neural networks. *Journal of Atmospheric and Oceanic Technology*, **24**(1), 52–63.

Cerdeña, A., J. C. Pérez and A. González, 2004: Cloud properties retrieval using neural networks. in K. P. Schäfer, A. Comerón, M. R. Carleer, R. H. Picard and N. I. Sifakis, editors, *Remote Sensing of Clouds and the Atmosphere IX.*, volume 5571 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pp. 11–19.

Coakley, J. A. and F. P. Bretherton, 1982: Cloud cover from high-resolution scanner data: Detecting and allowing for partially filled fields of view. *Journal of Geophysical Research*, **87**, 4917–4932.

Cornet, C., J. C. Buriez, J. Riédi, H. Isaka and B. Guillemet, 2005: Case study of inhomogeneous cloud parameter retrieval from modis data. *Geophysical Research Letters*, **32**, 13807+.

Cornet, C., H. Isaka, B. Guillemet and F. Szczap, 2004: Neural network retrieval of cloud parameters of inhomogeneous clouds from multispectral and multiscale radiance data: Feasibility study. *Journal of Geophysical Research*, **109**, 12203+.

Curry, J. A. and P. J. Webster, 1999: *Thermodynamics of Atmospheres and Oceans (International Geophysics)*. Academic Press.

D'Entremont, R. P., 1986: Low- and midlevel cloud analysis using nighttime multispectral imagery. *Journal of Applied Meteorology*, **25**, 1853–1869.

Driedonks, A. G. M. and P. G. Duynkerke, 1989: Current problems in the stratocumulus-topped atmospheric boundary layer. *Boundary-Layer Meteorology*, **46**, 275–303.

Durkee, P. A., R. E. Chartier, A. Brown, E. J. Trehubenko, S. D. Rogerson, C. Skupniewicz, K. E. Nielsen, S. Platnick and M. D. King, 2000a: Composite ship track characteristics. *Journal of Atmospheric Sciences*, **57**, 2542–2553.

Durkee, P. A., K. J. Noone and R. T. Bluth, 2000b: The Monterey area ship track experiment. *Journal of Atmospheric Sciences*, **57**(16), 2523–2541.

Evans, K. F., 1998: The spherical harmonics discrete ordinate method for three-dimensional atmospheric radiative transfer. *Journal of Atmospheric Sciences*, **55**, 429–446.

Faure, T., H. Isaka and B. Guillemet, 2001a: Mapping neural network computation of high-resolution radiant fluxes of inhomogeneous clouds. *Journal of Geophysical Research*, **106**, 14961–14974.

Faure, T., H. Isaka and B. Guillemet, 2001b: Neural network analysis of the radiative interaction between neighboring pixels in inhomogeneous clouds. *Journal of Geophysical Research*, **106**, 14465–14484.

Faure, T., H. Isaka and B. Guillemet, 2001c: Neural network retrieval of cloud parameters of inhomogeneous and fractional clouds - feasibility study. *Remote Sensing of Environment*, **77**(2), 123–138.

Faure, T., H. Isaka and B. Guillemet, 2002: Neural network retrieval of cloud parameters from high-resolution multispectral radiometric data - a feasibility study. *Remote Sensing of Environment*, **80**(2), 285–296.

Fu, Q. and K. N. Liou, 1992: On the correlated k-distribution method for radiative transfer in nonhomogeneous atmospheres. *Journal of Atmospheric Sciences*, **49**, 2139–2156.

Gonzälez, A., J. C. Pérez, F. Herrera, F. Rosa, M. A. Wetzel, R. D. Borys and D. H. Lowenthal, 2002: Stratocumulus properties retrieval method from noaa-avhrr data based on the discretization of cloud parameters. *International Journal of Remote Sensing*, **23**(4), 627–645.

Han, Q., W. B. Rossow and A. A. Lacis, 1994: Near-global survey of effective droplet radii in liquid water clouds using isccp data. *Journal of Climate*, **7**, 465–497.

Hansen, P., 1994: Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems. *Numerical Algorithms*, **6**(1), 1–35.

Harshvardhan, 1982: The effect of brokenness on cloud-climate sensitivity. *Journal of Atmospheric Sciences*, **39**(8), 1853–1861.

Hartmann, D. L., M. E. Ockert-Bell and M. L. Michelsen, 1992: The effect of cloud type on earth's energy balance: Global analysis. *Journal of Climate*, **5**, 1281–1304.

Heck, P. W., W. L. Smith, P. Minnis and D. F. Young, 1999: Multispectral retrieval of nighttime cloud properties for CERES, ARM, and FIRE. in *Proceedings of ALPS 99 Symposium*, Meribel, France.

Hobbs, P. V., T. J. Garrett, R. J. Ferek, S. R. Strader, D. A. Hegg, G. M. Frick, W. A. Hoppel, R. F. Gasparovic, L. M. Russell, D. W. Johnson, C. O'Dowd, P. A. Durkee, K. E. Nielsen and G. Innis, 2000: Emissions from ships with respect to their effects on clouds. *Journal of Atmospheric Sciences*, **57**, 2570–2590.

Hsieh, W. W. and B. Tang, 1998: Applying neural network models to prediction and data analysis in meteorology and oceanography. *Bulletin of the American Meteorological Society*, **79**, 1855–1870.

Hu, Y. and K. Stamnes, 2000: Climate sensitivity to cloud optical properties. *Tellus B*, **52**(1), 81–93.

Hunt, G. E., 1973: Radiative properties of terrestial clouds at visible and infra-red thermal window wavelengths. *Quarterly Journal of the Royal Meteorological Society*, **99**, 346–369.

Iwabuchi, H., 2007: Retrieval of cloud optical thickness and effective radius using multispectral remote sensing and accounting for 3D effects. in A. A. Kokhanovsky, editor, *Light Scattering Reviews 2*, pp. 97–124. Springer.

Kato, S., L. M. Hinkelman and A. Cheng, 2006: Estimate of satellite-derived cloud optical thickness and effective radius errors and their effect on computed domain-averaged irradiances. *Journal of Geophysical Research*, **111**, 17201+.

Kawamoto, K., T. Nakajima and T. Y. Nakajima, 2001: A global determination of cloud microphysics with avhrr remote sensing. *Journal of Climate*, **14**, 2054–2068.

Kawanishi, T., T. Sezai, Y. Ito, K. Imaoka, T. Takeshima, Y. Ishido, A. Shibata, M. Miura, H. Inahata and R. W. Spencer, 2003: The Advanced Microwave Scanning Radiometer for the Earth Observing System (AMSR-E), NASDA's contribution to the EOS for global energy and water cycle studies. *Geoscience and Remote Sensing, IEEE Transactions on*, **41**(2), 184–194.

King, M., S. Tsay, S. Platnick, M. Wang and K. Liou, 1997: Cloud retrieval algorithms for MODIS: optical thickness, effective particle radius, and thermodynamic phase. *MODIS Algorithm Theoretical Basis Document No. ATBD-MOD-05*, NASA.

Klein, S. A. and D. L. Hartmann, 1993: The seasonal cycle of low stratiform clouds. *Journal of Climate*, **6**, 1587–1606.

Krasnopolsky, V. M., 2007: Reducing uncertainties in neural network jacobians and improving accuracy of neural network emulations with nn ensemble approaches. *Neural Networks*, **20**(4), 454–461.

Krasnopolsky, V. M., L. C. Breaker and W. H. Gemmill, 1995: A neural network as a nonlinear transfer function model for retrieving surface wind speeds from the special sensor microwave imager. *Journal of Geophysical Research*, **100**, 11033–11046.

Krasnopolsky, V. M., W. H. Gemmill and L. C. Breaker, 2000: A neural network multiparameter algorithm for SSM/I ocean retrievals - comparisons and validations. *Remote Sensing of Environment*, **73**(2), 133–142.

Kratz, D. P., 1995: The correlated k-distribution technique as applied to the AVHRR channels. *Journal of Quantitative Spectroscopy and Radiative Transfer*, **53**, 501–517.

Kratz, D. P., 2001: MODIS correlated k-distributions. *http://asd-www.larc.nasa.gov/~kratz/modis.html*, accessed April 9th, 2007.

Lee, T. E., S. D. Miller, F. J. Turk, C. Schueler, R. Julian, S. Deyo, P. Dills and S. Wang, 2006: The NPOESS VIIRS day/night visible sensor. *Bulletin of the American Meteorological Society*, **87**, 191–199.

Lin, X. and J. A. Coakley, 1993: Retrieval of properties for semitransparent clouds from multispectral infrared imagery data. *Journal of Geophysical Research*, **98**, 18501–18514.

Lohmann, U. and J. Feichter, 2005: Global indirect aerosol effects: a review. *Atmospheric Chemistry & Physics*, **5**, 715–737.

Luo, G., X. Lin and J. A. Coakley, 1994: 11-μm emissivities and droplet radii for marine stratocumulus. *Journal of Geophysical Research*, **99**, 3685–3698.

MacKay, D. J. C., 1992a: Bayesian interpolation. *Neural Computation*, **4**(3), 415–447.

MacKay, D. J. C., 1992b: A practical Bayesian framework for backpropagation networks. *Neural Computation*, **4**(3), 448–472.

MacKay, D. J. C., 1995: Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, **6**, 469–505.

Martin, G. M., D. W. Johnson and A. Spice, 1994: The measurement and parameterization of effective radius of droplets in warm stratocumulus clouds. *Journal of Atmospheric Sciences*, **51**, 1823–1842.

Mayer, B. and A. Kylling, 2005: Technical note: The libRadtran software package for radiative transfer calculations - description and examples of use. *Atmospheric Chemistry & Physics*, **5**, 1855–1877.

McCann, D. W., 1992: A neural network short-term forecast of significant thunderstorms. *Weather and Forecasting*, **7**(3), 525–534.

Mie, G., 1908: Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen. *Annalen der Physik*, **330**, 377–445.

Miles, N. L., J. Verlinde and E. E. Clothiaux, 2000: Cloud droplet size distributions in low-level stratiform clouds. *Journal of Atmospheric Sciences*, **57**, 295–311.

Minnis, P., D. P. Kratz, J. A. Coakley, M. D. King, D. Garber, P. Heck, S. Mayor, D. F. Young and R. Arduini, 1995: *Cloud Optical Property Retrieval (Subsystem 4.3)*. volume 3, pp. 135–176. NASA RP 1376.

Mlawer, E. J., S. J. Taubman, P. D. Brown, M. J. Iacono and S. A. Clough, 1997: Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-k model for the longwave. *Journal of Geophysical Research*, **102**, 16663–16682.

Nabney, I. T., 2002: *NETLAB – Algorithms for Pattern Recognition*. Springer.

Nakajima, T. and M. D. King, 1990: Determination of the optical thickness and effective particle radius of clouds from reflected solar radiation measurements. Part I: Theory. *Journal of Atmospheric Sciences*, **47**(15), 1878–1893.

Nakajima, T., M. D. King, J. D. Spinhirne and L. F. Radke, 1991: Determination of the optical thickness and effective particle radius of clouds from reflected solar radiation measurements. Part II: Marine stratocumulus observations. *Journal of Atmospheric Sciences*, **48**, 728–751.

Nakajima, T. Y. and T. Nakajma, 1995: Wide-area determination of cloud microphysical properties from noaa avhrr measurements for fire and astex regions. *Journal of Atmospheric Sciences*, **52**(23), 4043–4059.

Norris, J. R. and C. B. Leovy, 1994: Interannual variability in stratiform cloudiness and sea surface temperature. *Journal of Climate*, **7**, 1915–1925.

Pawlowska, H. and J. Brenguier, 2000: Microphysical properties of stratocumulus clouds during ACE-2. *Tellus B*, **52**, 868+.

Pearlmutter, B. A., 1994: Fast exact multiplication by the Hessian. *Neural Computation*, **6**(1), 147–160.

Pérez, J. C., P. H. Austin and A. González, 2002: Retrieval of boundary layer cloud properties using infrared satellite data during the dycoms-ii field experiment. in *Proceedings 15th Symposium Boundary Layer and Turbulence*, Wageningen, Netherlands.

Pérez, J. C., F. Herrera, F. Rosa, A. González, M. A. Wetzel, R. D. Borys and D. H. Lowenthal, 2000: Retrieval of marine stratus cloud droplet size from NOAA-AVHRR nighttime imagery. *Remote Sensing of Environment*, **73**(1), 31–45.

Petty, G. W., 2006: *A First Course in Atmospheric Radiation (2nd Ed.)*. Sundog Publishing.

Phillips, T. and P. L. Barry, 2002: Clouds in the greenhouse. *http://science.nasa.gov/headlines/y2002/22apr_ceres.htm*, accessed June 30th, 2007.

Pincus, R., M. Szczodrak, J. Gu and P. H. Austin, 1995: Uncertainty in cloud optical depth estimates made from satellite radiance measurements. *Journal of Climate*, **8**, 1453–1462.

Platnick, S., M. D. King, S. A. Ackerman, W. P. Menzel, B. A. Baum, J. C. Riedi and R. A. Frey, 2003: The MODIS cloud products: algorithms and examples from Terra. *Geoscience and Remote Sensing, IEEE Transactions on*, **41**(2), 459–473.

Platnick, S. and S. Twomey, 1994: Determining the susceptibility of cloud albedo to changes in droplet concentration with the Advanced Very High Resolution Radiometer. *Journal of Applied Meteorology*, **33**, 334–347.

Platnick, S. and F. P. J. Valero, 1995: A validation of a satellite cloud retrieval during ASTEX. *Journal of Atmospheric Sciences*, **52**, 2985–3001.

Press, W. H., S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, 2007: *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press.

Rigdon, E. E., 1997: Not positive definite matrices – causes and cures. *http://www2.gsu.edu/~mkteer/npdmatri.html*, accessed June 18th, 2007.

Ringer, M. A., B. J. Mcavaney, N. Andronova, L. E. Buja, M. Esch, W. J. Ingram, B. Li, J. Quaas, E. Roeckner, C. A. Senior, B. J. Soden, E. M. Volodin, M. J. Webb and K. D. Williams, 2006: Global mean cloud feedbacks in idealized climate change experiments. *Geophysical Research Letters*, **33**, 7718+.

Rosenblatt, F., 1958: The perception: a probabilistic model for information storage and organization in the brain. *Psychological Review*, **65**(6), 386–408.

Rossow, W. B. and R. A. Schiffer, 1999: Advances in understanding clouds from ISCCP. *Bulletin of the American Meteorological Society*, **80**, 2261–2288.

Rozendaal, M. A., C. B. Leovy and S. A. Klein, 1995: An observational study of diurnal variations of marine stratiform cloud. *Journal of Climate*, **8**, 1795–1809.

Savic-Jovcic, V. and B. Stevens, 2007: The structure and mesoscale organization of precipitating stratocumulus. *Journal of Atmospheric Sciences*, **in press**.

Schreier, M., A. A. Kokhanovsky, V. Eyring, L. Bugliaro, H. Mannstein, B. Mayer, H. Bovensmann and J. P. Burrows, 2006: Impact of ship emissions on the microphysical, optical and radiative properties of marine stratus: a case study. *Atmospheric Chemistry & Physics*, **6**, 4925–4942.

Schüller, L., R. Bennartz, J. Fischer and J. L. Brenguier, 2005: An algorithm for the retrieval of droplet number concentration and geometrical thickness of stratiform marine boundary layer clouds applied to MODIS radiometric observations. *Journal of Applied Meteorology*, **44**, 28–38.

Schüller, L., J. L. Brenguier and H. Pawlowska, 2003: Retrieval of microphysical, geometrical, and radiative properties of marine stratocumulus from remote sensing. *Journal of Geophysical Research*, **108**, 5-1.

Sharon, T. M., B. A. Albrecht, H. H. Jonsson, P. Minnis, M. M. Khaiyer, T. M. van Reken, J. Seinfeld and R. Flagan, 2006: Aerosol and cloud microphysical characteristics of rifts and gradients in maritime stratocumulus clouds. *Journal of Atmospheric Sciences*, **63**, 983–997.

Stamnes, K., S. C. Tsay, K. Jayaweera and W. Wiscombe, 1988: Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media. *Applied Optics*, **27**, 2502–2509.

Stamnes, K., S. C. Tsay, W. Wiscombe and I. Laszlo, 2000: DISORT, a general-purpose Fortran program for discrete-ordinate-method radiative transfer in scattering and emitting layered media: Documentation of methodology. Technical report, Dept. of Physics and Engineering Physics, Stevens Institute of Technology, Hoboken, NJ 07030.

Stephens, G. L., 2005: Cloud feedbacks in the climate system: A critical review. *Journal of Climate*, **18**, 237–273.

Stephens, G. L., D. G. Vane, R. J. Boain, G. G. Mace, K. Sassen, Z. Wang, A. J. Illingworth, E. J. O'Connor, W. B. Rossow, S. L. Durden, S. D. Miller, R. T. Austin, A. Benedetti, C. Mitrescu and The, 2002: The Cloudsat mission and the A-Train. *Bulletin of the American Meteorological Society*, **83**, 1771–1790.

Stevens, B., A. Beljaars, S. Bordoni, C. Holloway, M. Köhler, S. Krueger, V. Savic-Jovcic and Y. Zhang, 2007: On the structure of the lower troposphere in the summertime stratocumulus regime of the northeast pacific. *Monthly Weather Review*, **135**(3), 985–1005.

Stevens, B., D. H. Lenschow, G. Vali, H. Gerber, A. Bandy, B. Blomquist, J. L. Brenguier, C. S. Bretherton, F. Burnet, T. Campos, S. Chai, I. Faloona, D. Friesen, S. Haimov, K. Laursen, D. K. Lilly, S. M. Loehrer, S. P. Malinowski, B. Morley, M. D. Petters, D. C. Rogers, L. Russell, V. Savic-Jovcic, J. R. Snider, D. Straub, M. J. Szumowski, H. Takagi, D. C. Thornton, M. Tschudi, C. Twohy, M. Wetzel and M. C. van Zanten, 2003a: Dynamics and Chemistry of Marine Stratocumulus – DYCOMS-II. *Bulletin of the American Meteorological Society*, **84**(5), 579–593.

Stevens, B., D. H. Lenschow, G. Vali, H. Gerber, A. Bandy, B. Blomquist, J. L. Brenguier, C. S. Bretherton, F. Burnet, T. Campos, S. Chai, I. Faloona, D. Friesen, S. Haimov, K. Laursen, D. K. Lilly, S. M. Loehrer, S. P. Malinowski, B. Morley, M. D. Petters, D. C. Rogers, L. Russell, V. Savic-Jovcic, J. R. Snider, D. Straub, M. J. Szumowski, H. Takagi, D. C. Thornton, M. Tschudi, C. Twohy, M. Wetzel and M. C. van Zanten, 2003b: Supplement to Dynamics and Chemistry of Marine Stratocumulus – DYCOMS-II (flight summaries). *Bulletin of the American Meteorological Society*, **84**(5), S12–S25.

Stevens, B., G. Vali, K. Comstock, R. Wood, M. C. van Zanten, P. H. Austin, C. S. Bretherton and D. H. Lenschow, 2005: Pockets of open cells and drizzle in marine stratocumulus. *Bulletin of the American Meteorological Society*, **86**, 51–57.

Turner, D. D., A. M. Vogelmann, R. T. Austin, J. C. Barnard, K. Cady-Pereira, J. C. Chiu, S. A. Clough, C. Flynn, M. M. Khaiyer, J. Liljegren, K. Johnson, B. Lin, C. Long, A. Marshak, S. Y. Matrosov, S. A. McFarlane, M. Miller, Q. Min, P. Minnis, W. O'Hirok, Z. Wang and W. Wiscombe, 2007: Thin liquid water clouds: Their importance and our challenge. *Bulletin of the American Meteorological Society*, **88**(2), 177–190.

Twohy, C. H., M. D. Petters, J. R. Snider, B. Stevens, W. Tahnk, M. Wetzel, L. Russell and F. Burnet, 2005: Evaluation of the aerosol indirect effect in marine stratocumulus clouds: Droplet number, size, liquid water path, and radiative impact. *Journal of Geophysical Research*, **110**, 8203+.

Twomey, S., 1974: Pollution and the planetary albedo. *Atmospheric Environment*, **8**(12), 1251–1256.

Twomey, S., 1977: The influence of pollution on the shortwave albedo of clouds. *Journal of Atmospheric Sciences*, **34**, 1149–1154.

Twomey, S. and T. Cocks, 1989: Remote sensing of cloud parameters from spectral reflectance in the near-infrared. *Beiträge zur Physik der Atmosphäre*, **62**, 172–179.

van Delst, P., 2005: Sensor SpcCoeff data. *http://cimss.ssec.wisc.edu/~paulv/*, accessed September 6th, 2007.

Vaughan, M. A., S. A. Young, D. M. Winker, K. A. Powell, A. H. Omar, Z. Liu, Y. Hu and C. A. Hostetler, 2004: Fully automated analysis of space-based lidar data: an overview of the CALIPSO retrieval algorithms and data products. in U. N. Singh, editor, *Laser Radar Techniques for Atmospheric Sensing.*, volume 5575 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pp. 16–30.

Wallace, J. M. and P. V. Hobbs, 2006: *Atmospheric Science, Volume 92, Second Edition: An Introductory Survey (International Geophysics).* Academic Press.

Weisstein, E. W., 2007: MathWorld – a Wolfram web resource. *http://mathworld.wolfram.com*, accessed June 5th, 2007.

Wielicki, B. A., E. F. Harrison, R. D. Cess, M. D. King and D. A. Randall, 1995: Mission to planet earth: Role of clouds and radiation in climate. *Bulletin of the American Meteorological Society*, **76**(11), 2125–2154.

Williams, C. K. I., C. Qazaz, C. M. Bishop and H. Zhu, 1995: On the relationship between Bayesian error bars and the input data density. in *Artificial Neural Networks, 1995., Fourth International Conference on*, pp. 160–165.

Williams, K. and G. Tselioudis, 2007: GCM intercomparison of global cloud regimes: present-day evaluation and climate change response. *Climate Dynamics*, **29**(2), 231–250.

Wood, R., 2007: Cancellation of aerosol indirect effects in marine stratocumulus through cloud thinning. *Journal of Atmospheric Sciences*, **64**(7), 2657–2669.

Wood, R., C. S. Bretherton and D. L. Hartmann, 2002: Diurnal cycle of liquid water path over the subtropical and tropical oceans. *Geophysical Research Letters*, **29**, 7–1.

Xu, K. M., T. Wong, B. A. Wielicki, L. Parker and Z. A. Eitzen, 2005: Statistical analyses of satellite cloud object data from CERES. part I: Methodology and preliminary results of the 1998 El Niño/2000 La Niña. *Journal of Climate*, **18**, 2497–2514.

Xue, H., G. Feingold and B. Stevens, 2007: The role of precipitating cells in organizing shallow cumulus convection. *Journal of Atmospheric Sciences*, **in press**.

Appendix A

Implementation of the Aires et al. Method in Matlab

A.1 Modified Netlab Functions

Table A.1 lists all NETLAB functions that were modified for this study. About half of the functions implementing MLPs in NETLAB had to be modified in order to accommodate the matrix hyperparameters $\boldsymbol{A}_{in}$ and $\boldsymbol{A}_r$, and some additional functions were implemented as well. However, only the most important changes and implementation details will be discussed in this appendix.

A.2 Main Loop: mlatrain

The implementation of Algorithm 3.1 in `mlatrain` is straightforward. Following the NETLAB design, the network is stored in a structure `net` (cf. Nabney, 2002). It is trained using the generic NETLAB function `netopt`:

```
net = netopt(net, options, x, t, 'scg');
```

The covariance matrix $\boldsymbol{C}_0$ is computed from the network predictions (evaluated with `mlafwd`) and the target variables by using the MATLAB function `cov`:

```
y = mlafwd(net, x);
epsilon = y - t;
C0 = cov(epsilon)
```

Next, $\boldsymbol{G}$ and $\tilde{\boldsymbol{H}}$ are computed using `mladeriv` and `mlahess`, and $\langle \boldsymbol{G}^T \tilde{\boldsymbol{H}}^{-1} \boldsymbol{G} \rangle$ is evaluated:

```
G = mladeriv(net, x);
[hess, hdata] = mlahess(net, x, t);
invhess = inv(hess);
GHGavg = zeros(net.nout);
for n = 1 : ndata
```

Table A.1: List of NETLAB functions that were adapted or created in order to accommodate the matrix hyperparameters $\boldsymbol{A}_{in}$ and $\boldsymbol{A}_r$ (cf. Nabney, 2002, Table 5.1).

original function	new function	function	changes
`mlp`	`mla`	create two-layer MLP	scalar hyperparameters have been replaced by the matrix ones
`mlpbkp`	`mlabkp`	backpropagate error gradient through network	none
`mlpderiv`	`mladeriv`	evaluate derivatives of network outputs with respect to weights	none
`mlperr`	`mlaerr`	evaluate error function	implementation of (3.22) instead of (3.24)
`mlpfwd`	`mlafwd`	forward propagation	none
`mlpgrad`	`mlagrad`	evaluate error gradient	full backpropagation with matrix hyperparameters
`mlphdotv`	`mlahdotv`	evaluate the product of the Hessian with a vector	$\mathcal{R}\{\cdot\}$-algorithm has been adapted to the new error function
`mlphess`	`mlahess`	evaluate the Hessian matrix	use full covariance matrix $\boldsymbol{A}_r$ instead of α
`mlppak`	`mlapak`	combine weights and biases into one parameter vector	none
`mlpunpak`	`mlaunpak`	separate parameter vector into weight and bias matrices	none
—	`mlatrain`	implementation of Algorithm 3.1	—
—	`mlarescale`	normalise input or target variables	—
—	`mlareverserescale`	undo the rescaling of `mlarescale`	—
—	`mlajacob`	evaluate the Jacobian matrix	—
—	`mlajacobcheck`	verify the Jacobian matrix with finite differences	—
—	`mlajacobuncertainty`	compute the uncertainty in the Jacobian matrix	—

```
    Gn = squeeze(G(n, :, :));
    GHG = Gn' * invhess * Gn;
    GHGavg = GHGavg + GHG;
end
GHGavg = GHGavg / ndata;
```

Finally, the hyperparameters are re-estimated following equations (3.44) and (3.59):

```
Cin = C0 - GHGavg;
net.Ain = inv(Cin);
w = netpak(net);
newalpha = (ones(net.nwts, 1) - diag(net.Ar).*diag(invhess)) ./ w'.^2;
net.Ar = diag(newalpha);
```

A.3 Gradient Computation with Backpropagation: mlagrad

The derivative of the error function with respect to the weights is needed by the optimisation algorithm and is computed in `mlagrad`. The function is based on the method of *error back-propagation* (Bishop, 1995, Section 4.8). In short, the total error function is decomposed into error terms of the individual patterns of the training dataset, so that

$$\frac{\partial E}{\partial w_{ji}} = \sum_n \frac{\partial E^n}{\partial w_{ji}}. \tag{A.1}$$

The error E^n depends on the weight w_{ji} through the summed input $a_j = \sum_i w_{ji} z_i$ to unit j, hence we can write

$$\frac{\partial E^n}{\partial w_{ji}} = \frac{\partial E^n}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}} = \delta_j z_i, \tag{A.2}$$

where the notation

$$\delta_j = \frac{\partial E^n}{\partial a_j} \tag{A.3}$$

has been introduced and $a_j = \sum_i w_{ji} z_i$ has been differentiated to give $\partial a_j / \partial w_{ji} = z_i$. Since the inputs z_i to a particular unit are known, the δs have to be computed. For the output units,

$$\delta_k = \tilde{g}'(a_k)\frac{\partial E^n}{\partial y_k} \tag{A.4}$$

(since $y_k = \tilde{g}(a_k)$), whereas Bishop (1995) shows that for the hidden units

$$\delta_j = g'(a_j)\sum_k w_{kj}\delta_k. \tag{A.5}$$

Since the error derivatives of the hidden units depend on the δs of the output units, the algorithm is called back-propagation.

The derivative of the output activation function in our case is $\tilde{g}'(a_k) \equiv 1$ for the output units, since $\tilde{g}$ is a linear function. For the in NETLAB implemented sum-of-squares error function (3.23),

$$E^n = \frac{1}{2}\sum_{k=1}^{c}\left(y_k^n - t_k^n\right)^2, \tag{A.6}$$

the output δs evaluate to

$$\delta_k = \frac{\partial E^n}{\partial y_k} = \frac{1}{2}\cdot 2\cdot\left(y_k^n - t_k^n\right) = y_k^n - t_k^n. \tag{A.7}$$

However, for the Aires error function (3.22),

$$E^n = \frac{1}{2}\left(\boldsymbol{y}^n - \boldsymbol{t}^n\right)^T\cdot\boldsymbol{A_{in}}\cdot\left(\boldsymbol{y}^n - \boldsymbol{t}^n\right) + \frac{1}{2N}\boldsymbol{w}^T\cdot\boldsymbol{A_r}\cdot\boldsymbol{w} \tag{A.8}$$

(where the weights term has been divided by N to express its contribution to the error of a single pattern n), the derivative with respect to the weights becomes

$$\frac{\partial E^n}{\partial w_{ji}} = \frac{\partial E_D^n}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}} + \frac{1}{N}\frac{\partial E_W}{\partial w_{ji}}. \tag{A.9}$$

The weights term can be evaluated directly to give

$$\frac{\partial E_W}{\partial w_r} = \frac{\partial}{\partial w_r}\left\{w_r A_W^{rs} w_s\right\} = A_W^{rs} w_s, \tag{A.10}$$

where all weights are considered to be a vector with a single index. The data term is computed with back-propagation, leading to the output δs

$$\begin{aligned}\delta_k = \frac{\partial E_D^n}{\partial y_k} &= \frac{\partial}{\partial y_k}\left\{\frac{1}{2}\left(y_k^n - t_k^n\right)A_{in}^{kl}\left(y_l^n - t_l^n\right)\right\} &\text{(A.11)}\\ &= A_{in}^{kl}\left(y_l^n - t_l^n\right). &\text{(A.12)}\end{aligned}$$

Here, the property that $\boldsymbol{A_{in}}$ is symmetric has been used.

The above equations can be implemented in a few lines in MATLAB:

```
delout = (y - t) * net.Ain;
gdata = mlabkp(net , x, z, delout);
w = mlapak(net);
```

```
gprior = w * net.Ar;
g = gdata + gprior;
```

First, all δ_k are computed with (A.12). Then the data term of the error gradient is computed with back-propagation (`mlabkp`) and the weights term (A.10) is added.

A.4 The Hessian with the Pearlmutter $\mathcal{R}\{\cdot\}$-Algorithm: mlahess, mlahdotv

Nabney (2002) uses an algorithm derived by Pearlmutter (1994) to compute the Hessian matrix of the network. The idea is to use an efficient algorithm to compute the product $\boldsymbol{v}^T\boldsymbol{H}$ of a vector $\boldsymbol{v}$ with the Hessian, and to evaluate the full Hessian by using a sequence of unit vectors that each pick out one column of $\boldsymbol{H}$. Pearlmutter uses the notation $\mathcal{R}\{\cdot\}$ for the operator $\boldsymbol{v}^T\nabla$, so that

$$\boldsymbol{v}^T\boldsymbol{H} \equiv \boldsymbol{v}^T\nabla\left(\nabla E\right) = \mathcal{R}\{\nabla E\}. \tag{A.13}$$

His derivation, a summary of which can be found in Bishop (1995, Section 4.10.7), leads to two expressions for the derivative of the error function with respect to the first and second layer weights, respectively:

$$\begin{aligned}
\mathcal{R}\left\{\frac{\partial E}{\partial w_{ji}}\right\} &= \mathcal{R}\left\{\delta_j\right\} x_i && \text{(A.14)}\\
\mathcal{R}\left\{\frac{\partial E}{\partial w_{kj}}\right\} &= \mathcal{R}\left\{\delta_k\right\} z_j + \delta_k \mathcal{R}\left\{z_j\right\}, && \text{(A.15)}
\end{aligned}$$

where the δs are the standard back-propagation expressions given by (A.4) and (A.5). As was the case for `mlagrad`, the NETLAB implementation had to be adapted to the new δ_k. Using (A.12),

$$\begin{aligned}
\mathcal{R}\left\{\delta_k\right\} &= \mathcal{R}\left\{A_{in}^{kl}\left(y_l - t_l\right)\right\} && \text{(A.16)}\\
&= \mathcal{R}\left\{A_{in}^{kl} y_l\right\} - \underbrace{\mathcal{R}\left\{A_{in}^{kl} t_l\right\}}_{=0} && \text{(A.17)}\\
&= \underbrace{\mathcal{R}\left\{A_{in}^{kl}\right\}}_{=0} y_l + A_{in}^{kl}\mathcal{R}\left\{y_l\right\} && \text{(A.18)}\\
&= A_{in}^{kl}\mathcal{R}\left\{y_l\right\}. && \text{(A.19)}
\end{aligned}$$

Together with the results

$$\mathcal{R}\{a_j\} = \sum_i v_{ji} x_i \tag{A.20}$$

$$\mathcal{R}\{z_j\} = g'(a_j)\,\mathcal{R}\{a_j\} \tag{A.21}$$

$$\mathcal{R}\{y_k\} = \sum_j w_{kj} \mathcal{R}\{z_j\} + \sum_j v_{kj} z_j \tag{A.22}$$

$$\mathcal{R}\{\delta_j\} = g''(a_j)\,\mathcal{R}\{a_j\} \sum_k w_{kj}\delta_k + g'(a_j) \sum_k v_{kj}\delta_k + g'(a_j) \sum_k w_{kj} \mathcal{R}\{\delta_k\} \tag{A.23}$$

(see Bishop, 1995, Section 4.10.7), the algorithm is implemented in **mlahdotv**. Again, all δ_ks in (A.19) can be evaluated in one vector:

```
rdel = ry * net.Ain;
```

In **mlahess**, the data Hessian part of the total Hessian given by (3.27) is evaluated using **mlahdotv**, then the weights part $\boldsymbol{A}_r$ is added:

```
for v = eye(net.nwts);
  hdata(find(v),:) = mlahdotv(net, x, t, v);
end
h = hdata + net.Ar;
```

A.5 The Jacobian and its Distribution: mlajacob, mlajacobuncertainty

As Nabney (2002, Chapter 5) notes, the Jacobian matrix for a two layer MLP can be computed with

$$J_{ki} \equiv \frac{\partial y_k}{\partial x_i} = \sum_{j=1}^{M} w_{kj} \left(1 - z_j^2\right) w_{ji}, \tag{A.24}$$

which is based on a tanh hidden unit activation function. The implementation of (A.24) in **mlajacob** is straightforward. Following Nabney's radial basis functions implementation (Nabney, 2002, Section 6.4.2), I used the shortcut $\Psi_{ji} = \left(1 - z_j^2\right) w_{ji}$, leading to

```
[y, z, a] = mlafwd(net, x);
for n = 1:ndata
    Psi = (ones(net.nin, 1)*(1-z(n, :).^2)).*net.w1;
    jac(n, :, :) = Psi * net.w2;
end
```

Note that the function returns a three dimensional array which contains the Jacobian for each individual input pattern.

The Aires et al. (2004b) technique to estimate the variability in the Jacobian (see section 3.5) is implemented in `mlajacobuncertainty`. Samples from the weights distribution are generated with the MATLAB STATISTICS TOOLBOX function `mvnrnd`, which returns random samples from a multivariate Gaussian distribution (here, according to (3.30), the most probable weights vector $\boldsymbol{w}^*$ is used as the mean of the distribution and the inverse Hessian $\tilde{\boldsymbol{H}}^{-1}$ is the covariance matrix):

```
invH = inv(H)
wmp = mlapak(net);
wsamples = mvnrnd(wmp, invH, nsamples);
```

For each of these weights samples, the Jacobian is computed and averaged over all input patterns $\boldsymbol{x}^n$:

```
mjac = zeros(nsamples, net.nin, net.nout);
for i = 1:nsamples
    neti = mlaunpak(net, wsamples(i,:));
    jac = mlajacob(neti, x);
    mjac(i, :, :) = squeeze(mean(jac));
end
```

Eventually, the mean and standard deviation of all mean Jacobians are computed and returned:

```
meanjac = squeeze(mean(mjac));
stdjac = squeeze(std(mjac));
```

A.6 Normalisation of Input and Output Variables

The functions `mlarescale` and `mlareverserescale` implement the normalisation methods mentioned in section 3.6.1. `mlarescale` implements the simple linear rescaling (3.61):

```
nrm.mean = mean(D);
nrm.norm = std(D);
RD = D - (ones(ndata, 1)*nrm.mean);
RD = RD ./ (ones(ndata, 1)*nrm.norm);
```

All variables v_i^n are passed as a matrix D and processed all at once.

Whitening, the more sophisticated linear rescaling method (3.62), is also implemented in `mlarescale`:

```
nrm.mean = mean(D);
nrm.sig = cov(D);
[evU, evL] = eig(nrm.sig);
nrm.evU = evU;
nrm.evL = evL;
```

```
RD = D - (ones(ndata, 1)*nrm.mean);
RD = (evL^(-.5) * evU' * RD')';
```

The corresponding code to reverse both these rescaling methods is implemented in `mlareverserescale`.

A.7 Implementation Difficulties: Regularisation of the Hessian and Numerical Symmetry

While working with the implementation of the Aires et al. algorithm, I encountered two difficulties. One of them has been discussed in section 3.6.2, the positive definite character of the Hessian. The Nabney (2002) method is implemented at the corresponding places in the `mla*` routines:

```
[hess, hdata] = mlahess(net, x, t);
[evec, evl] = eig(hdata);
evl = evl .* (evl > 0);
hdata = evec * evl * evec';
hess = hdata + net.Ar;
```

Another problem I encountered while working with my implementation is that the Hessian matrices that are computed (and consequently their inverses) are not completely numerically symmetric, as would be expected. This means that due to numerical inaccuracies in MATLAB, the elements h_{ij} of the Hessian follow $h_{ij} = h_{ji} + \epsilon$, with ϵ being a small perturbation. Such small errors can amplify considerably when the matrices are multiplied with other matrices. Furthermore, the function `mvnrnd` used in `mlajacobuncertainty` expects a numerically symmetric covariance matrix. However, the issue is easily resolved by copying one half of the matrix into the other half where needed.