

Bayesian ocean colour modelling

RÉMI PIERRE SERRA

MSc by Research in Pattern Analysis and Neural Networks



ASTON UNIVERSITY

September 2002

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without proper acknowledgement.

ASTON UNIVERSITY

Bayesian ocean colour modelling

RÉMI PIERRE SERRA

MSc by Research in Pattern Analysis and Neural Networks, 2002

Thesis Summary

Current atmospheric correction algorithms are based on physical models and pixel wise retrieval. The goal of this thesis is to build a Bayesian framework using a probabilistic approach to enable the use of priors for the joint retrieval of ocean and aerosol parameters on case I waters. Simulated data containing ocean and aerosol parameters as well as the corresponding top of atmosphere information and its components will be used to train neural networks able to output the top of atmosphere components given the ocean and aerosol parameters. A Bayesian framework will be built to enable the retrieval of the ocean and aerosol parameters from the top of atmosphere information using the neural networks previously trained and priors which will be designed according to biological and physical knowledge. The Bayesian framework will then be tested on small problems, parts of the global retrieval problem, and then on the global retrieval problem, on simulated data at first, and secondly on real data.

Keywords: Bayesian ocean colour modelling, neural networks, priors.

Acknowledgements

I would like to thank my supervisors, Doctor Dan Cornford and Doctor Ian Nabney, for their help and their availability.

I would also thank Cedric Jamet and Cyril Moulin, LODYC, for the data and help they provided, and Anton Lyaskovskiy and Lehel Csato, NCRG, for their help.

I am grateful to all my lecturers Professor D. Lowe, Doctor I. Nabney, Doctor M. Opper, Professor D. Saad, Doctor I. Stainvas and Doctor P. Tino for the quality of their lectures. Many thanks also to Vicky Bond.

I would also like to thank all the people from the Neural Computing Research Group department for their constant support and friendly presence through all this year and especially the Msc students in PANN, Boremi, Mani, Manu, Nicolas, Oliver and Stephane.

I also want to thank my friends from Aston, Alexandre, Amine, Amos, Anna, Katherine, Chris, Clement, Eric, Jenni, Jimmy, Liz, Raphael, Rita, Sebastien, Sergei, Shumsa and Sophie with whom I've spent a wonderful year as well as Bruno A, Bruno H, Gilles, Laure, Matthias, Vitali and Vivek for their visits.

To finish with my acknowledgements, I would like to thank my parents and my family for their support.

Contents

1	Introduction	12
1.1	Context	12
1.2	Overview of the research project	13
1.3	Research steps	14
2	Theoretical background	17
2.1	Neural Networks	17
2.1.1	Multi-Layer Perceptron	17
2.1.2	Radial Basis Function	19
2.1.3	Mixture Density Network	20
2.2	Gaussian processes	23
3	Observation models	24
3.1	Oceanic model	25
3.1.1	Data	25
3.1.2	Model selection	25
3.1.3	Chosen model	25
3.2	Atmospheric models	27
3.2.1	Atmospheric reflectance	27
3.2.2	Atmospheric transmittance	31
4	Building the Bayesian framework	35
4.1	Principle of the Bayesian framework	35
4.2	Atmospheric priors	36
4.3	Oceanic priors	37
4.4	Implementation of the Bayesian framework	37
5	Testing of Bayesian frameworks	38
5.1	Test on NIR bands	38
5.1.1	Retrieval of a single variable	39
5.1.2	Joint retrieval of two variables	41

CONTENTS

5.2 Test on oceanic model 45
5.2.1 Retrieval of a single variable 45
5.2.2 Joint retrieval of two variables 48
5.3 Conclusion of the test of the Bayesian framework 49

6 Tests on simulated data 50
6.1 First retrieval framework 50
6.2 Second retrieval framework 51
6.3 Physically wrong results 53
6.3.1 MLP vs RBF 54
6.3.2 Gaussian mixture model priors over the data space 54
6.3.3 Log parameter space 54
6.3.4 Improved first guess 55
6.4 Third retrieval framework 59
6.5 Fourth retrieval framework 65
6.6 Conclusion of the tests on simulated data 69

7 Tests on real data 71
7.1 Standard SeaWiFS atmospheric correction 71
7.2 Bayesian retrieval 72
7.3 Conclusion of the tests on real data 73

8 Conclusion 76

9 Appendix 78
9.1 Classifiers for m_r and m_i 78
9.1.1 Pyramidal Classifiers for m_i 78
9.2 RBFs 82
9.3 Log parameter space 84
9.3.1 MLPs 84
9.3.2 RBFs 86
9.4 First guess models 88
9.5 Test on simulated data 90

List of Figures

1.1	The physical problem: decomposition of top of atmosphere radiance . . .	13
1.2	Diagram of the decomposition of ρ_{toa} in the three neural networks of $\rho_{atmo}(\lambda), t(\lambda), \rho_w(\lambda)$	15
2.1	Multi-Layer Perceptron	17
2.2	RBF network	19
2.3	The Mixture Density Network	21
3.1	Oceanic reflectance model, inputs are logs of b^0 and [chl], λ , output is ρ_w	26
3.2	Atmospheric reflectance model, $\nu \leq 3$, inputs are $m_r, m_i, \tau, \nu, \theta_s, \theta_v, \phi, \lambda$, output is ρ_a	29
3.3	Atmospheric reflectance model, $\nu > 3$, inputs are $m_r, m_i, \tau, \nu, \theta_s, \theta_v, \phi, \lambda$, output is ρ_a	30
3.4	Atmospheric reflectance model, all ν , inputs are $m_r, m_i, \tau, \nu, \theta_s, \theta_v, \phi, \lambda$, output is ρ_a	30
3.5	Atmospheric transmittance model, inputs are $m_r, m_i, \tau, \nu, \theta_s, \theta_v, \lambda$, output is t	32
3.6	MLP Classifier for ν	33
3.7	MLP Regression for ϵ , inputs are geometry(3 angles), ν and τ (normalised), output is $\epsilon = \rho(s)/\rho(l)$ (unnormalised).	34
5.1	Diagram of the Bayesian framework test.	38
5.2	Retrieval of a τ field given the NIR bands observations using a Bayesian framework and an MLP, without noise.	39
5.3	Retrieval of a τ field given the NIR bands observations using a Bayesian framework, a noise of 1% of $mean(\epsilon = \rho(s)/\rho(l))$ has been applied to the observations.	40
5.4	Joint retrieval of ν and τ fields given $\epsilon = \rho(s)/\rho(l)$ using a Bayesian framework.	42
5.5	Joint retrieval of ν and τ fields given NIR bands observations (instead of $\epsilon = \rho(s)/\rho(l)$) using a Bayesian framework.	43

LIST OF FIGURES

5.6	Joint retrieval of ν and τ fields given NIR bands observations and $\epsilon = \rho(s)/\rho(l)$ using a Bayesian framework.	43
5.7	Retrieval of ν and τ fields iterating on two networks.	44
5.8	Retrieval of a $\log([chl])$ field given the observations using a Bayesian framework, without noise.	45
5.9	Retrieval of a $\log([chl])$ field given the observations using a Bayesian framework. A noise of 20% of $mean(\rho)$ has been applied on the observations, the noise assumption is not consistent with the noise added. . .	46
5.10	Retrieval of a $\log([chl])$ field given the observations using a Bayesian framework. A noise of 20% of $mean(\rho)$ has been applied on the observations, the noise assumption is consistent with the noise added. . . .	46
5.11	Comparison between priors with noise assumption consistent or inconsistent with the added noise. Average on 100 experiments.	47
5.12	Retrieval of $\log(b^0)$ and $\log([chl])$ fields given the observations using a Bayesian framework, without noise.	48
5.13	Retrieval of $\log(b^0)$ and $\log([chl])$ fields given the observations using a Bayesian framework. A noise of 5% of $mean(\rho)$ has been applied on the observations, the noise assumption is consistent with the noise added. .	48
6.1	Testing process.	50
6.2	Pyramidal classifiers to retrieve m_i from ρ_{toa}	52
6.3	Second global retrieval framework, using the NIR bands to retrieve ν and τ , and the classifiers to first guess m_τ and m_i	52
6.4	Global retrieval using MLPs - retrieved parameters.	53
6.5	Global retrieval using MLPs - observations at the SeaWiFS visible wavelengths.	53
6.6	Gaussian mixture priors on $\nu, \tau, m_\tau, m_i, b^0$ and $[chl]$. Vertical blue lines are the limits of the data range in the training set, red curve is the Gaussian mixture prior.	55
6.7	Second framework - Retrieval using the complex forward model in logspace (MLPs): parameters.	56
6.8	Second framework - Retrieval using the complex forward model in logspace (MLPs): observations.	56
6.9	Network for first guess of b^0 and $[chl]$, inputs are $\tau, \nu, \theta_s, \theta_v, \phi, \lambda$, outputs are b^0 and $[chl]$. Plot of target versus output for $[chl]$	57
6.10	Network for first guess of b^0 and $[chl]$, inputs are $\tau, \nu, \theta_s, \theta_v, \phi, \lambda$, outputs are b^0 and $[chl]$. Plot of target versus output for b^0	57
6.11	Network for first guess of $[chl]$, inputs are $\tau, \nu, \theta_s, \theta_v, \phi, \lambda$, output is $[chl]$. Plot of target versus output for $[chl]$	57

LIST OF FIGURES

6.12	Third Global retrieval framework.	59
6.13	Third framework - First guess (MLPs): parameters.	60
6.14	Third framework - First guess (MLPs): observations.	60
6.15	Third framework - Simple forward model (MLPs): parameters.	61
6.16	Third framework - Simple forward model (MLPs): observations.	61
6.17	Third framework - First guess and simple forward model (MLPs): parameters.	62
6.18	Third framework - First guess and simple forward model (MLPs): observations.	62
6.19	Third framework - Complex model (MLPs): parameters.	63
6.20	Third framework - Simple forward model and complex model (MLPs): parameters.	63
6.21	Third framework - Simple forward model and complex model (RBFs): parameters.	64
6.22	Third framework - Simple forward model and complex model (RBFs): observations.	64
6.23	Fourth Global retrieval framework.	66
6.24	Fourth framework - Simple forward model and complex model (RBFs): ρ_a	67
6.25	Fourth framework - Simple forward model and complex model (RBFs): t	67
6.26	Fourth framework - Simple forward model and complex model (RBFs): ρ_w . The errors on ρ_a and t lead to a important distortion on ρ_w	68
6.27	Final Global retrieval framework.	70
7.1	Retrieval using the simple forward model and complex model (MLPs): observations.	72
7.2	Retrieval using the simple forward model and complex model (MLPs): parameters.	73
7.3	Retrieval using the complex model (MLP): observations.	73
7.4	Retrieval using the complex model (MLP): parameters.	74
7.5	Retrieval using the complex model (MLP): ρ_w	74
7.6	Retrieval using the complex model (MLP): ρ_a	75
9.1	Results of the classifier MLP to retrieve m_r from ρ_{toa}	78
9.2	Results of the classifier MLP to retrieve m_i from ρ_{toa}	79
9.3	Pyramidal classifiers to retrieve m_i from ρ_{toa} - First classifier.	79
9.4	Pyramidal classifiers to retrieve m_i from ρ_{toa} - Classifier for low values.	80
9.5	Pyramidal classifiers to retrieve m_i from ρ_{toa} - Classifier for high values.	81
9.6	RBF for atmospheric reflectance.	82
9.7	RBF for atmospheric transmittance.	83

LIST OF FIGURES

9.8	RBF for water-leaving reflectance.	83
9.9	MLP for atmospheric reflectance in logspace.	84
9.10	MLP for atmospheric transmittance in logspace.	85
9.11	MLP for oceanic reflectance in logspace.	85
9.12	RBF for atmospheric reflectance in logspace.	86
9.13	RBF for atmospheric transmittance in logspace.	86
9.14	RBF for water-leaving reflectance.	87
9.15	Network for first guess of m_r and m_i , inputs are $m_r, m_i, \tau, \nu, \theta_s, \theta_v, \phi, \lambda$, output is $\rho_{toa}(\lambda)$	88
9.16	Network for first guess of b^0 and [chl], inputs are $\tau, \nu, \theta_s, \theta_v, \phi, b^0, [chl], \lambda$, output is $\rho_{toa}(\lambda)$	88
9.17	Network for first retrieval of m_r, m_i, b^0 and [chl], inputs are $m_r, m_i, \tau,$ $\nu, \theta_s, \theta_v, \phi, b^0, [chl], \lambda$, output is $\rho_{toa}(\lambda)$	89

List of Tables

1.1	Description of model parameters.	15
1.2	Characteristics of model parameters.	16
1.3	Model parameter values. The λ are taken at SeaWiFS wavelengths.	16
3.1	Oceanic variables, dataset size: 1224.	25
3.2	Oceanic model errors, using 1000 training samples and 1000 epochs.	26
3.3	Oceanic final model errors.	26
3.4	Atmospheric reflectance dataset.	27
3.5	Atmospheric reflectance variables (training set).	27
3.6	Errors on atmospheric reflectance without weight decay.	28
3.7	Errors on atmospheric reflectance with weight decay.	28
3.8	Atmospheric reflectance, final model errors.	29
3.9	Atmospheric transmittance dataset.	31
3.10	Atmospheric transmittance variables (training set).	31
3.11	Atmospheric transmittance model errors.	31
3.12	Choice of angles inputs for atmospheric transmittance.	32
3.13	Atmospheric transmittance final model errors.	32
3.14	NIR bands model variables.	33
3.15	Table of errors on ϵ , NIR final model.	34
4.1	Priors for the NIR bands parameters	37
4.2	Priors for the oceanic parameters	37
5.1	Table of errors, ν and τ retrieval from NIR bands and ϵ , plotted on Figure 5.6. λ_1 , λ_2 and λ_3 are SeaWiFS three last wavelengths (Near Infra Red bands).	41
5.2	Table of errors, ν and τ retrieval iterating on two networks, plotted on Figure 5.7.	42
6.1	Table of errors, global retrieval in two steps.	51
6.2	Errors on retrieval using only first guess networks (MLPs).	60
6.3	Characteristics of ρ_w derived from the retrieved parameters and ρ_w in the training set of the neural network.	66

LIST OF TABLES

9.1	RBFs errors.	82
9.2	Log MLPs and RBFs errors.	84
9.3	Errors on retrieval using only the simple forward model (MLPs).	90
9.4	Errors on retrieval using first guess networks and simple forward model (MLPs).	90
9.5	Errors on retrieval using the complex forward model (MLPs).	90
9.6	Errors on retrieval using the simple forward model and complex forward model (MLPs).	91
9.7	Errors on retrieval using the simple forward model and complex forward model (RBFs).	91
9.8	Errors on retrieval using the complex forward model in logpsace (MLPs).	91
9.9	Errors on retrieval using the simple forward model and complex forward model in logpsace (MLPs).	92
9.10	Errors on the observation derived from the retrieved parameters in the fourth framework.	92

Chapter 1

Introduction

1.1 Context

Phytoplankton is a plant which floats at the surface of the oceans. It is the beginning of the food chain in the ocean. Thus, information on the location of phytoplankton can be useful for the fishing industry. Moreover, as phytoplankton depend upon specific conditions for growth, it frequently becomes the first indicator of a change in the environment.

Satellites provide images of the colour of the ocean from which the presence of chlorophyll and thus phytoplankton can be inferred. The retrieval of the chlorophyll concentration is complicated by the presence of the atmosphere which alters the signal received by the satellite; the atmosphere can be responsible of more than 80% of the signal received by the satellite. Thus, atmospheric correction must be done to deduce the chlorophyll concentration.

Current atmospheric correction algorithms use near infrared wavelength to assess aerosol optical properties together with lookup tables built from physical models to remove the aerosol contribution and obtain the water contribution. Chlorophyll concentration is then deduced from the water-leaving contribution.

Algorithms to improve the calculation of atmospheric properties have been developed [7] and methods to jointly retrieve ocean and aerosol optical properties have been proposed [4].

Current methods process each pixel at a time, thus the knowledge of spatial structure of the ocean and atmosphere is not used.

The purpose of this project is to build a retrieval process using a probabilistic approach to retrieve the ocean and aerosol properties from SeaWiFS images, exploiting prior knowledge of the structure of ocean and aerosol data in case I waters.

These probabilistic methods have been successfully used for the calculation of wind-fields from satellite images.

1.2 Overview of the research project

The satellite coverage is regular and covers wide ocean areas; thus knowledge of the spatial structure of the parameters can improve the retrieval, especially to correct the noise induced by the use of sensors. This will be done by building a Bayesian framework which will enable the use of spatial and spatio-temporal priors on the different parameters of the system.

The physical model used in this thesis was designed by Chomko & Gordon [4].

The signal received by the satellite can be divided in two parts: the reflection of the incoming solar radiance, before being modified by the sea surface, and the radiance that has been modified by the sea surface and by the atmosphere.

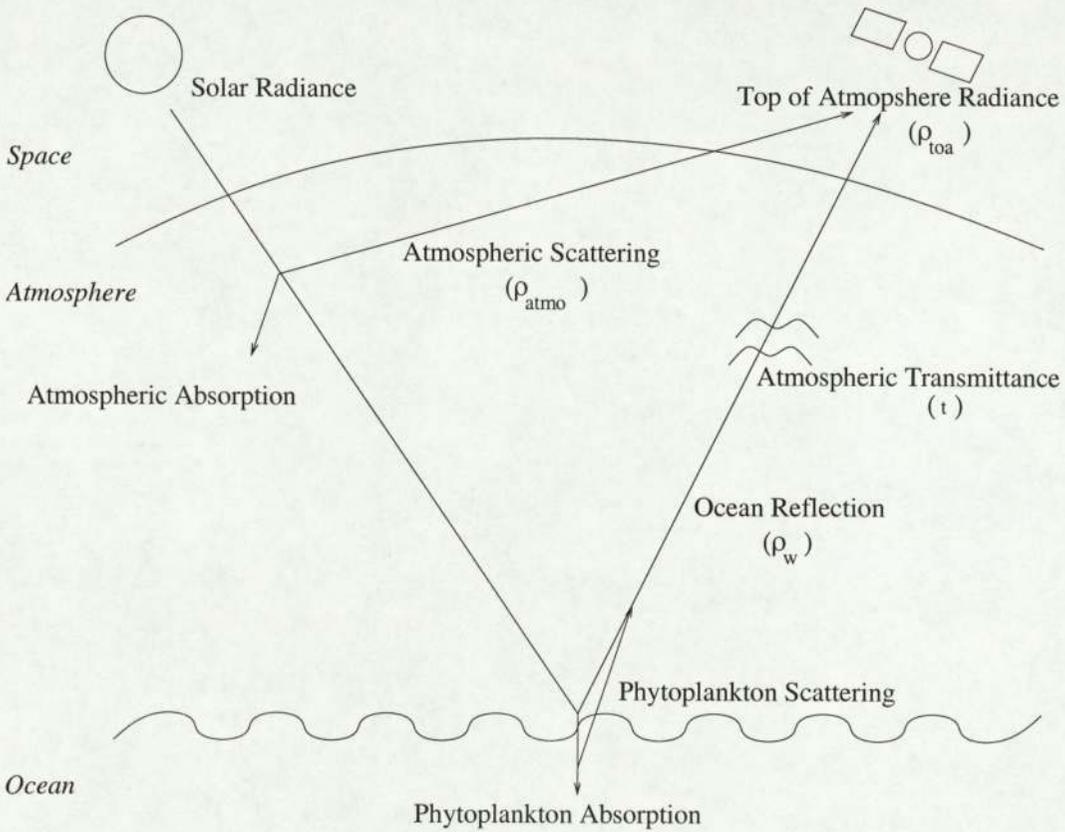


Figure 1.1: The physical problem: decomposition of top of atmosphere radiance

First of all, we will work on the reflectances ρ in place of radiance L , as the measurement on the reflectance ought to be more accurate on next generation satellites [6]. Radiance and reflectance are related by:

$$\rho(\lambda) = \frac{\pi L(\lambda) \cos(\theta_0)}{F_0(\lambda)} \quad (1.1)$$

Where $F_0(\lambda)$ is the extraterrestrial irradiance and θ_0 is the solar zenith angle.

The top of atmosphere (TOA) radiance (ρ_{toa}), which is measured by the satellite, is composed of the pure aerosol-scattering contribution (ρ_a), the pure Rayleigh-scattering contribution (ρ_r), the contribution due to the interaction effect between air molecules and aerosols (ρ_{ra}), the water-leaving contribution (multiplied by the atmospheric transitivity) $t(\lambda) \cdot \rho_w(\lambda)$ and the contribution of whitecaps (ρ_{wc}) and sun glitter (ρ_g).

$$\rho_{toa}(\lambda) = \rho_a(\lambda) + \rho_{ra}(\lambda) + \rho_r(\lambda) + t(\lambda) \cdot \rho_w(\lambda) + t(\lambda) \cdot \rho_{wc}(\lambda) + T(\lambda) \cdot \rho_g(\lambda) \quad (1.2)$$

The contribution of whitecaps (ρ_{wc}) and sun glitter (ρ_g) are removed in SeaWiFS preprocessing, and the pure Rayleigh-scattering contribution (ρ_r) can be computed precisely and then be subtracted from ρ_{toa} [4].

$$\rho'_{toa}(\lambda) = \rho_a(\lambda) + \rho_{ra}(\lambda) + t(\lambda) \cdot \rho_w(\lambda) \quad (1.3)$$

Then the atmospheric contributions (ρ_a and ρ_{ra}) can be considered as an unique variable (ρ_{atmo}). This simplified interaction is shown in Figure 1.1.

$$\rho'_{toa}(\lambda) = \rho_{atmo}(\lambda) + t(\lambda) \cdot \rho_w(\lambda) \quad (1.4)$$

The aim is to compute ρ'_{toa} from atmospheric and oceanic parameters, thus three models will be built: atmospheric reflectance (light reflected before reaching the sea); oceanic reflectance (light reflected by the ocean, carrying colour information about chlorophyll concentration); and atmospheric transmittance (light modified by the atmosphere), each model having atmospheric or oceanic parameters as inputs.

The parameters of the models, shown in Table 1.1, are both atmospheric (ν , τ , m_r , m_i) and oceanic (b^0 , [chl]), the more important, that is the more interesting, is [chl], chlorophyll concentration. ν and b^0 represent the size distribution of particles in the atmosphere and the ocean respectively; that is the aerosols and dust particles for the atmosphere, and detritus for the ocean. Their properties in the datasets provided by Cedric Jamet and Cyril Moulin are shown in Tables 1.2 and 1.3.

Only ν , τ , m_r , m_i , b^0 and [chl] need to be retrieved. The other parameters, θ_s , θ_v , ϕ_v , which represent the position of the satellite and the sun, and λ , which is the wavelength at which the radiance is measured, are known.

1.3 Research steps

The research will follow the following steps: first the data provided by Cedric Jamet and Cyril Moulin, LODYC, will be learned using forward models; those networks will output the different reflectances given the oceanic or atmospheric parameters. One model will be built for each of the three components.

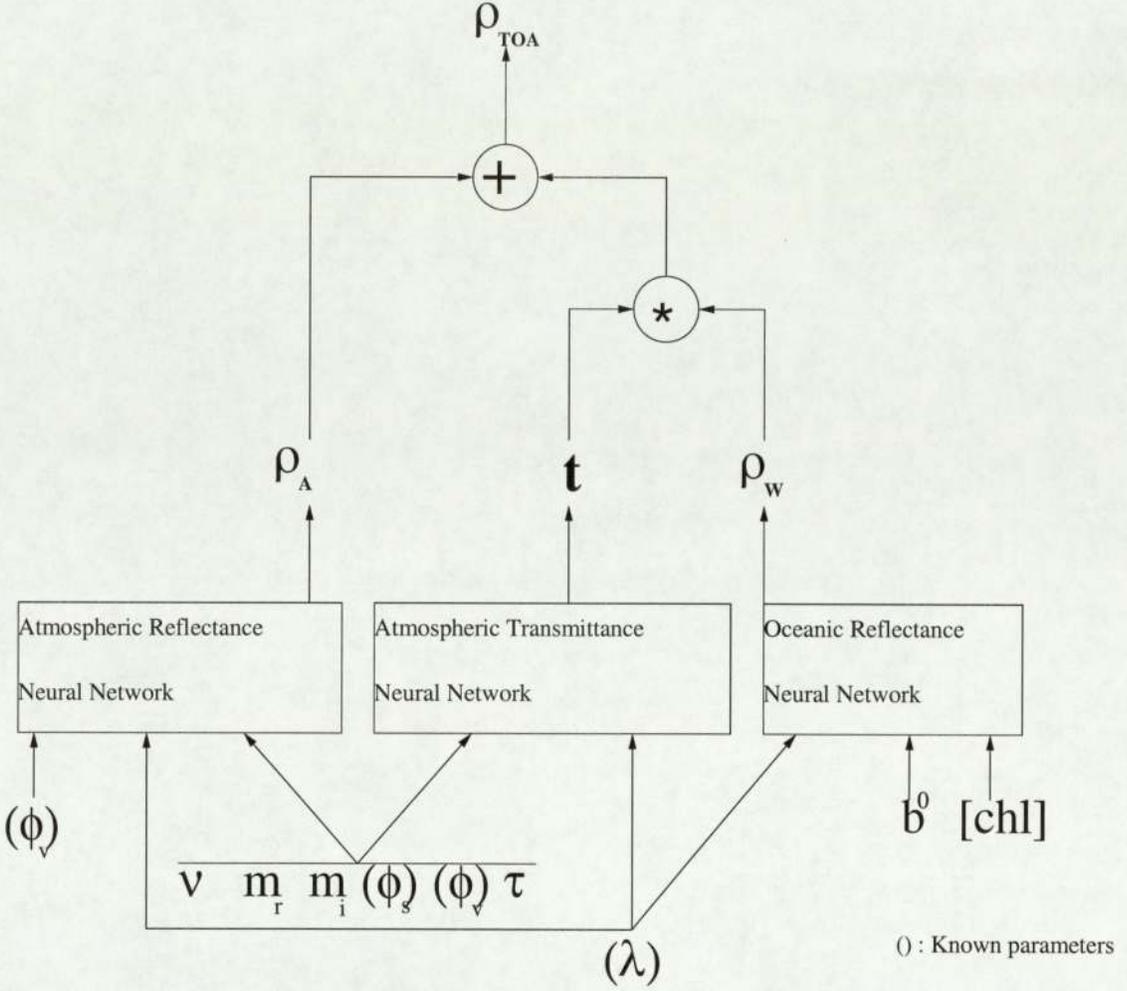


Figure 1.2: Diagram of the decomposition of ρ_{toa} in the three neural networks of $\rho_{atmo}(\lambda)$, $t(\lambda)$, $\rho_w(\lambda)$.

Parameter	Description	Type	Unit
ν	Particle size distribution parameter	Atmospheric	NA
τ	Aerosol optical depth	Atmospheric	NA
m_r	Refractive index, real part	Atmospheric	NA
m_i	Refractive index, imaginary part	Atmospheric	NA
θ_s	Solar zenith angle	Geometric	Degrees
θ_v	Sensor zenith angle	Geometric	Degrees
ϕ_v	Sun to sensor angle	Geometric	Degrees
λ	Wavelength	Optical	nm
b^0	Particle size distribution parameter	Oceanic	m^{-1}
[chl]	Chlorophyll concentration	Oceanic	$mg.cm^{-2}$

Table 1.1: Description of model parameters.

A Bayesian framework will then be built to retrieve all of the atmospheric and

	ν	τ	m_r	m_i	θ_s	θ_v	ϕ_v	λ	b^0	[chl]
mean	3.25	0.20	1.42	0.01	34.95	29.96	179.60	513.33	0.28	0.49
std	0.85	0.11	0.08	0.02	9.47	18.12	113.50	83.75	0.10	0.74
max	4.5	0.35	1.5	0.04	50	60	360	670	0.45	3
min	2	0.05	1.33	0	20	0	0	412	0.12	0.03

Table 1.2: Characteristics of model parameters.

ν	2 - 4.5, step 0.5
τ	0.05 - 0.35, step 0.1
m_r	1.33 - 1.5
m_i	0 - 0.001 - 0.003 - 0.01 - 0.03 - 0.04
θ_s	20 - 50, step 3
θ_v	0 - 60, step 3
ϕ_v	0 - 360, step 36
λ	412 - 443 - 490 - 510 - 555 - 670 - 765 - 865
b^0	0.12 - 0.45, step 0.03
[chl]	0.03 - 0.05 - 0.062 - 0.078 - 0.09 - 0.107 - 0.125 - 0.145 ...
...	0.17 - 0.2 - 0.25 - 0.31 - 0.41 - 0.62 - 1.15 - 1.5 - 3

Table 1.3: Model parameter values. The λ are taken at SeaWiFS wavelengths.

oceanic parameters, the main goal being to retrieve chlorophyll concentrations via the oceanic model. The individual models will be tested separately on their specialised retrieval problems before they are combined.

The Bayesian framework uses Bayes' rule to compute the posterior probability $p(\tau|\rho)$ of a parameter, τ , conditioned on the observations, ρ , using priors on each parameter, $p(\tau)$:

$$p(\tau|\rho) = \frac{p(\rho|\tau).p(\tau)}{p(\rho)}. \quad (1.5)$$

The priors over the parameters will be constructed using physical knowledge of the different components.

Once the retrieval on the three models is working, two different things will be needed, the first is the ability to handle large datasets. As the covariance matrices used to define the priors are calculated over the whole dataset, the computational limitation will require the use of sparse Gaussian processes, although in this thesis standard GPs will be used. Secondly, the three different models need to be combined to perform the actual retrieval of the chlorophyll concentration given TOA observations.

Finally, tests on simulated data and on real data will be performed to measure the real efficiency of the framework.

Chapter 2

Theoretical background

In this thesis, neural networks will be used, thus a basic background is provided here. More information on neural networks and optimisation methods can be found in [2], [1] and [9].

2.1 Neural Networks

2.1.1 Multi-Layer Perceptron

The Multi-layer Perceptron (MLP) [2] is a feed-forward neural network composed of several hidden layers (two in our case) of adaptive weights (See Figure 2.1). The

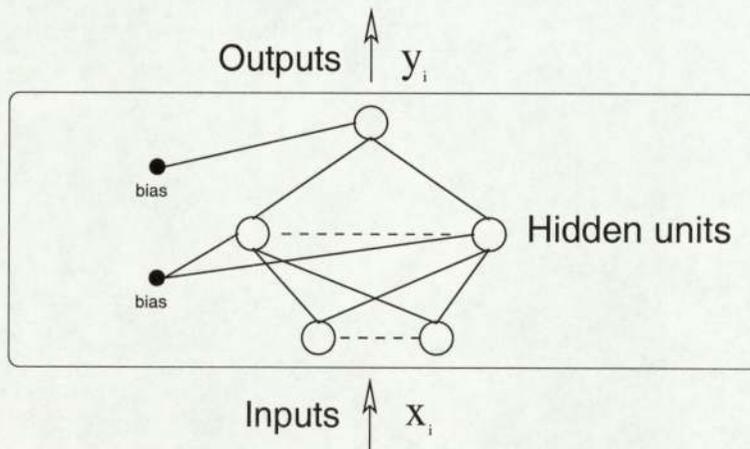


Figure 2.1: Multi-Layer Perceptron

outputs are computed from the inputs through each layer, according to the weights of the layer and the activation function used in the network. The value z_i^{h+1} of the

neuron i of layer $h + 1$ is

$$z_i^{h+1} = f \left(\sum_{j=1}^{N_h} w_{ij}^{h,h+1} z_j^h + b_i^h \right), \quad (2.1)$$

where h is the layer index, N_h is the number of neurons in layer h , b_i^h is the threshold of layer h for the i^{th} neuron, $w^{h,h+1}$ is the weight matrix between layers h and $h + 1$, and f is the activation function.

To build a network which from a given set of inputs leads to a corresponding output, it is necessary to adapt the weights and biases. This procedure is called learning.

The likelihood of the dataset is defined as

$$L = \prod_{q=1}^n p(t^q, x^q) = \prod_q p(t^q|x^q)p(x^q), \quad (2.2)$$

where n is the number of samples in the training set, and q is the training sample index. The error function E is defined as the negative log-likelihood

$$E = -\ln L. \quad (2.3)$$

Considering the noise model to be Gaussian, of variance σ , it becomes

$$p(t_k|x) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp \left\{ -\frac{(F_k(x) - t_k)^2}{2\sigma^2} \right\}, \quad (2.4)$$

where $F_k(x)$ is the underlying generator function to be learnt by the neural network. Using 2.2, 2.3 and 2.4, it becomes:

$$E = nc \ln \sigma + \frac{nc}{2} \ln(2\pi) + \frac{1}{2\sigma^2} \sum_{q=1}^n \sum_{k=1}^c (f_k(x^q; w) - t_k^q)^2 + \sum_{q=1}^n \ln p(x^q), \quad (2.5)$$

where n is the number of samples, c the dimension of the output, $f_k(x^q; w)$ is the modelisation by the neural networks of $F_k(x)$ and w are the weights of the network. In this equation, only the middle part is a function of the networks, and thus the cost function to be minimised is:

$$E = \frac{1}{2} \sum_q \sum_k (v_k^{out,q} - t_k^q)^2, \quad (2.6)$$

where q is the training sample index, k is the output vector index, and t^q is the desired output vector for the q^{th} input.

Given that the network is feed-forward and the activation function is differentiable, the derivative of this error function with respect to the weights and biases of the network can be found. This enables the training of the network using optimisation algorithms in order to adapt the weights and biases so a given set of inputs leads to a corresponding output [2].

2.1.2 Radial Basis Function

Radial Basis Function (RBF) networks [9] are related to kernel methods for density estimation and regression and to normal mixture models. The idea of an RBF model is to expand a given function f using a set of basis function of the form $\Phi(\|x - x_j\|)$, where Φ is a non-linear function to be chosen. The output is then taken to be a linear combination of the basis functions:

$$f(x) = \sum_j w_j \Phi_j(\|x - x_j\|) + w_0 \tag{2.7}$$

Where w_j is the weight given to the j^{th} basis function and w_0 is the bias. Several forms of basis function can be used, the most commonly used are the Gaussian and the thin-plate spline. The Gaussian basis function is:

$$\Phi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right), \tag{2.8}$$

where σ controls the smoothness properties of the interpolating function. The Gaussian is a localised basis function with the property that $\lim_{x \rightarrow \infty} \Phi(x) = 0$.

The thin-plate spline basis function is

$$\Phi(x) = x^2 \ln(x), \tag{2.9}$$

which is the best solution for curve fitting according to [9].

A radial basis function network uses several RBFs as hidden units (See Figure 2.2). The number M of basis functions needs to be less than the number N of data points,

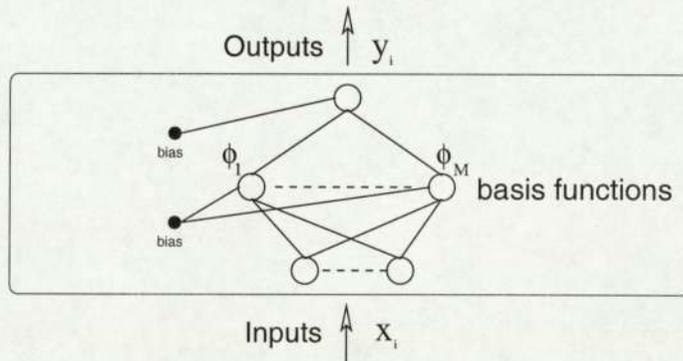


Figure 2.2: RBF network

each basis function has its own width σ_j . The interpolation formula 2.7 is then:

$$y_k(x) = \sum_{j=1}^M w_{kj} \Phi_j(x) + w_{k0} . \quad (2.10)$$

The Gaussian basis function can be expressed by:

$$\Phi_j(x) = \exp \left(-\frac{\|x - \mu_j\|^2}{2\sigma_j^2} \right) \quad (2.11)$$

and the thin-plate spline basis function by:

$$\Phi_j(x) = \|x - \mu_j\|^2 \ln(\|x - \mu_j\|) , \quad (2.12)$$

where x is the input vector and μ_j is the vector determining the centres of the basis function Φ_j . Once the basis function has been chosen, we have a simple model whose parameters can be found by a least squares procedure, or any other optimisation procedure.

For a large class of basis functions, RBF networks are universal approximators [9]. Besides, they possess the property of best approximation, which means that the set of functions corresponding to all possible choices of the adjustable parameters includes the optimal approximation. The advantage of this network family is that RBF models are very fast to train in comparison to networks with sigmoidal units such as MLPs. The main drawback of this type of network is that it becomes impractical with input vectors of large dimension.

2.1.3 Mixture Density Network

Mixture Density Networks (MDN) [1] combine a standard neural network with a mixture density model to provide a conditional distribution rather than a single output.

Introduction

The probability density of the target is a combination of kernel functions in the form

$$p(t|x) = \sum_{i=1}^m \alpha_i(x) \phi_i(t|x) , \quad (2.13)$$

where m is the number of components in the mixture, $\alpha_i(x)$ are the mixing coefficients (priors probabilities) and $\phi_i(t|x)$ is the conditional density of the target vector t for the i^{th} kernel. The implementation of such a model is shown in Figure 2.3:

- The first part of the MDN is a neural network with input vector \mathbf{x} and output vector \mathbf{z} . The output corresponds to the parameters for the kernel functions: priors, centres and variance. If c is dimension of the target, the dimension of \mathbf{z} is $(c+2)m$: m priors, $c m$ centres or origins of the kernel functions and m variances.
- The second part of the MDN is a mixture model with the parameter vector \mathbf{z} as input and $p(\mathbf{t}|\mathbf{x})$ as output.

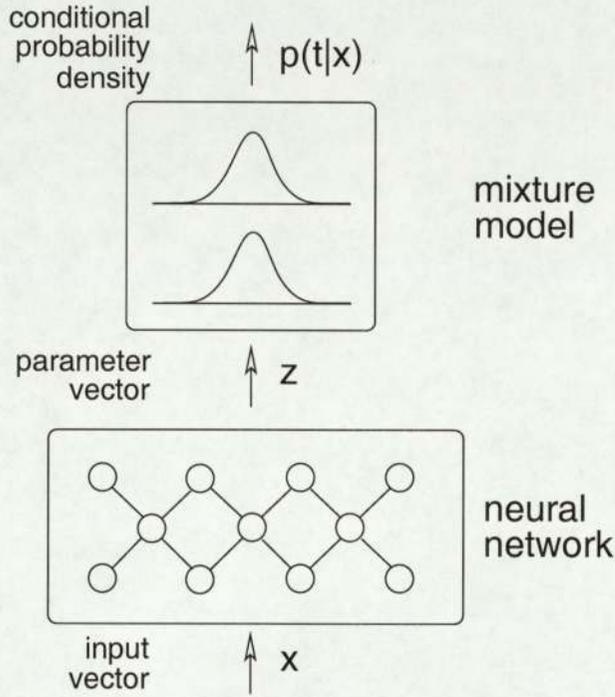


Figure 2.3: The Mixture Density Network

As this is a mixture model, there is the constraint

$$\sum_{i=1}^m \alpha_i(x) = 1 . \quad (2.14)$$

This can be obtained by considering $\alpha_i(x)$ as the *softmax* function of the output of the neural network

$$\alpha_i = \frac{\exp(z_i^\alpha)}{\sum_{j=1}^m \exp(z_j^\alpha)} . \quad (2.15)$$

The centres/origins are simply the neural network outputs

$$\mu_{ik} = z_{ik}^\mu . \quad (2.16)$$

The variances are the exponential of neural network outputs

$$\sigma_{ik} = \exp(z_{ik}^\mu) . \quad (2.17)$$

The error function for the MDN is defined as

$$E = \sum_q E^q, \quad (2.18)$$

where the error from pattern q is the negative logarithm of the likelihood (see equation 2.2) without the terms $p(x^q)$ as they are independent of the parameters of the mixture model:

$$E^q = -\ln \left\{ \sum_{i=1}^m \alpha_i(x^q) \phi_i(t^q|x^q) \right\}. \quad (2.19)$$

To optimise the network, this function is minimised with respect to the outputs of the neural network and then the modifications are back-propagated to optimise the weights of the networks.

The *posterior* probabilities, obtained using Bayes' rule are defined as

$$\pi_i(x, t) = \frac{\alpha_i \phi_i}{\sum_{j=1}^m \alpha_j \phi_j}. \quad (2.20)$$

Using 2.20 and 2.19 it becomes

$$\frac{\partial E^q}{\partial \alpha_i} = -\frac{\pi_i}{\alpha_i}, \quad (2.21)$$

and using 2.15

$$\frac{\partial \alpha_i}{\partial z_k^\alpha} = \delta_{ik} \alpha_i - \alpha_i \alpha_k. \quad (2.22)$$

Using the chain rule:

$$\frac{\partial E^q}{\partial z_k^\alpha} = \sum_i \frac{\partial E^q}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial z_k^\alpha}. \quad (2.23)$$

So finally using 2.21, 2.22 and 2.23 it follows that

$$\frac{\partial E^q}{\partial z_k^\alpha} = \alpha_k - \pi_k. \quad (2.24)$$

The other gradients are dependent on the distribution used.

Gaussian kernels functions

First of all, the functions used as kernels are Gaussian kernels, as Gaussians are universal estimators [9]. In this case, the function ϕ_i is defined by

$$\phi_i(t|x) = \frac{1}{(2\pi)^{c/2} \sigma_i(x)^c} \exp -\frac{\|t - \mu_i(x)\|^2}{2\sigma_i(x)^2}, \quad (2.25)$$

where c is the dimension of the output, $\mu_i(x)$ and $\sigma_i(x)$ are the centre and variance of the i^{th} kernel respectively.

The variance parameter has to be always positive. It can be viewed as an exponential

$$\sigma_i = \exp(z_i^\sigma). \quad (2.26)$$

Using 2.19, 2.20 and 2.25 it follows that

$$\frac{\partial E^q}{\partial \sigma_i} = -\pi_i \left\{ \frac{\|t - \mu_i(x)\|^2}{\sigma_i^3} - \frac{c}{\sigma_i} \right\}, \quad (2.27)$$

and as

$$\frac{\partial \sigma_i}{\partial z_i^\sigma} = \sigma_i, \quad (2.28)$$

it becomes

$$\frac{\partial E^q}{\partial z_i^\sigma} = -\pi_i \left\{ \frac{\|t - \mu_i(x)\|^2}{\sigma_i^2} - c \right\}. \quad (2.29)$$

And finally, using 2.20 and 2.25 the derivative of the cost-function with respect to the outputs of the network is:

$$\frac{\partial E^q}{\partial z_{ik}^\mu} = \pi_i \left\{ \frac{(\mu_{ik} - t_k)}{\sigma_i^2} \right\}. \quad (2.30)$$

2.2 Gaussian processes

A Gaussian process is a family of random variables $y(x)$, $x \in D$, such that for any finite collection the joint distribution of $y(x_1), \dots, y(x_n)$ is Gaussian. In the case of ocean colour data, the spatial structure is known through semivariograms of the different parameters, defined by

$$\gamma(v) = \frac{1}{2N(v)} \sum_{N(v)} (Z(x_i) - Z(x_j))^2, \quad (2.31)$$

where $N(v)$ are the numbers of data pairs $Z(x_i)$ and $Z(x_j)$ separated by v , which can be a vector or a simple distance [5]. A covariance function can be defined from the semivariogram as

$$C(v) = C(0) - \gamma(v), \quad (2.32)$$

then a Gaussian process can be derived from the covariance function as to say from the prior physical knowledge embedded in the semivariogram.

Chapter 3

Observation models

The forward models are based on data generated from the model designed by Gordon & Chomko [4], provided by Cedric Jamet and Cyril Moulin, LODYC. This data correspond to case I waters [4]. This dataset is built by constructing a basis covering all the possible combinations of the parameters shown in Table 1.3; the atmospheric reflectance, atmospheric transmittance and oceanic reflectance are then computed using Gordon & Chomko model. Thus the dataset is discrete by construction and covers a large range of the different parameters as shown in Table 1.3.

As the dataset is made of variables of different types, and different scales, some variables are likely to be overweighted compared to others. This can be corrected by normalizing the dataset. For each variable \mathbf{X}_i , the mean $\overline{\mathbf{X}_i}$ and variance σ_i are computed to obtain $\widetilde{\mathbf{X}_i}$, the normalized variable:

$$\widetilde{\mathbf{X}_i} = \frac{\mathbf{X}_i - \overline{\mathbf{X}_i}}{\sigma_i}. \quad (3.1)$$

The choices of the networks for the three forward models, atmospheric reflectance, atmospheric transmittance and oceanic reflectance, have been made using cross comparison on several MLPs, changing the number of hidden units, type of inputs (noised or not), weight decay and the number of iterations. The choice was done according to the efficiency of each network and the computational cost of the training. As the available dataset is large and the data to be learnt is complex, the network size required to overfit the training set is generally greater than what can be handled in terms of computational cost.

The models were trained and tested using two distinct datasets. The datasets used for the model selection are not the ones used for the final training and testing of the chosen models.

The final network for each of the three models has been trained on datasets bigger than the one used for cross comparison so as to produce precise models to be used in next steps (See Figures 3.1, 3.2, 3.3 and 3.5).

The errors of the different models which have been tested and the description of the final network for each model follow.

The results of the training of RBFs are in Appendix 9.2.

3.1 Oceanic model

3.1.1 Data

The oceanic database contains 1224 samples corresponding to a basis built on the parameters λ , b^0 and $[\text{chl}]$. λ is restricted to SeaWiFS visible wavelengths, which are the first six wavelengths. The measurements are shown in Table 3.1.

	λ	b^0	$[\text{chl}]$	ρ_w
mean	513.3333	0.285	0.4881	0.0122
std	83.7513	0.1036	0.7428	0.0108

Table 3.1: Oceanic variables, dataset size: 1224.

3.1.2 Model selection

Cross validation has been performed for networks having 5 to 30 hidden units, with three inputs and one output (Table 3.2).

The inputs and the outputs were normalised before training, the means and standard deviations used for normalisation were recorded together with the network. As the chlorophyll concentration is likely to be log distributed [5], it was then chosen to use the logs of b^0 and $[\text{chl}]$ as inputs instead of their true values.

The error globally decreases when using more hidden units (Table 3.2). The variations observed in the efficiency of each optimisation algorithm are due to random initialisation. Averaging the measurements on several random initialisation would have given more accurate results, but it would have been time consuming. The final network chosen will reach a small level of error, keeping in mind the time needed to train big networks.

A complete description of the optimisation algorithms used can be found in [2].

3.1.3 Chosen model

The network chosen for the oceanic reflectance model has 30 hidden units, with quasi-Newton optimisation, the target versus predicted output for this network is shown in Figure 3.1.

The inputs are the logs of b^0 and $[\text{chl}]$.

Hidden units	Algo	Training Error(RMS)	Test Error(RMS)	Relative RMS (% of target)
5	conjgrad	4.5989	0.0947	9.3045
5	quasinew	7.4352	0.1183	11.6222
5	scg	5.5243	0.1093	10.7394
10	conjgrad	0.6993	0.0344	3.3826
10	quasinew	1.2728	0.0501	4.9208
10	scg	1.1891	0.0536	5.2653
20	conjgrad	0.3303	0.0279	2.7376
20	quasinew	0.1005	0.0162	1.5966
20	scg	0.3894	0.0277	2.7226
30	conjgrad	0.2976	0.0258	2.5352
30	quasinew	0.0147	0.0064	0.6299
30	scg	0.3382	0.0277	2.7171

Table 3.2: Oceanic model errors, using 1000 training samples and 1000 epochs.

The errors calculated on the test set are in Table 3.3, the standard deviation is in target unit, which is the oceanic reflectance.

Relative RMS (% of target)	Bias	Std
0.1362	3.6909e-07	1.3900e-05

Table 3.3: Oceanic final model errors.

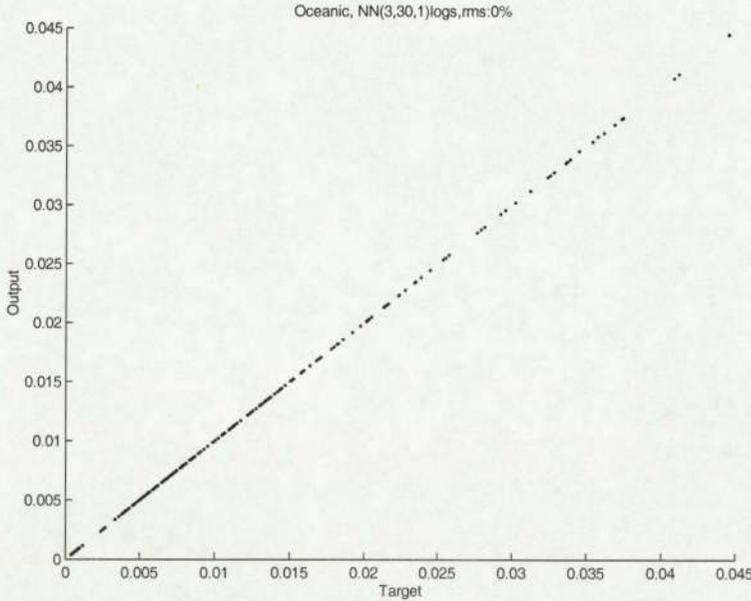


Figure 3.1: Oceanic reflectance model, inputs are logs of b^0 and $[chl]$, λ , output is ρ_w

3.2 Atmospheric models

3.2.1 Atmospheric reflectance

According to Cedric Jamet and Cyril Moulin, the physical model used to generate the data which has been used to train the network is wrong when $\nu \leq 3$ and $m_i > 0.01$. These samples were then removed from the training set and test set. It was first tried to cut the dataset into two parts, for $\nu \leq 3$ and $\nu > 3$. Two networks were used, one for the inputs having $\nu \leq 3$ (See Figure 3.2), another for the inputs having $\nu > 3$ (See Figure 3.3).

Afterwards, a solution using a single network was tested (See Figure 3.4).

Data

The atmospheric database for reflectance contains 120000 samples corresponding to a basis built on the parameters ν , m_r , m_i , θ_s , θ_v , ϕ_v , τ and λ (Table 3.4). λ is restricted to SeaWiFS visible wavelengths, which are the first six wavelengths. The measurements are shown in Table 3.5.

	Training set	Test set	Total
$\nu \leq 3$	40000	20000	60000
$\nu > 3$	40000	20000	60000
Total	80000	40000	120000

Table 3.4: Atmospheric reflectance dataset.

Variable	mean ($\nu \leq 3$)	std ($\nu \leq 3$)	mean ($\nu > 3$)	std ($\nu > 3$)	mean global	std global
ν	2.5	0.4092	4	0.4071	3.3980	0.8416
m_r	1.415	0.0850	1.415	0.0850	1.4152	0.0850
m_i	0.0035	0.0039	0.014	0.0155	0.0098	0.0133
θ_s	35	9.4490	35	9.4826	34.9625	9.4653
θ_v	30	18.1116	30	18.1504	29.9439	18.1252
ϕ_v	180	113.4547	180	113.7952	179.5898	113.4967
τ	0.2	0.1118	0.2	0.1115	0.2001	0.1117
λ	482	50.3632	482	50.2561	482.3115	50.2576
reflectance	0.0278	0.0245	0.0575	0.0452	0.0455	0.0422

Table 3.5: Atmospheric reflectance variables (training set).

Model selection

Cross validation has been performed for networks having 10 to 50 hidden units, with 8 inputs and 1 output.

The training was done either without weight decay (Table 3.6) or using different values for the weight decay rate (Table 3.7).

Hidden units	Relative RMS (% of target)	Relative RMS (% of target)
	Model for $\nu \leq 3$	Model for $\nu > 3$
10	26.4588	10.7269
20	22.5948	9.3610
35	18.9186	7.7926
50	18.1106	8.7530

Table 3.6: Errors on atmospheric reflectance without weight decay.

Hidden units	Weight decay variance	Relative RMS (% of target)	Relative RMS (% of target)
		Model for $\nu \leq 3$	Model for $\nu > 3$
20	0.02	23.4833	10.0182
20	2	25.0877	13.2244
20	200	61.6651	46.2164
30	0.02	20.6649	9.0194
30	2	22.9202	13.2258
30	200	61.6642	46.2064

Table 3.7: Errors on atmospheric reflectance with weight decay.

The best models are:

- For $\nu \leq 3$: 50 hidden units, no weight decay.
- For $\nu > 3$: 35 hidden units, no weight decay.

Chosen models

The best networks for the atmospheric reflectance model were two networks having 35 and 50 hidden units, but due to the computational cost of such large networks, it was chosen to use networks having only 30 hidden units, with scg optimisation and without weight decay (see Figures 3.2 and 3.3). The weight decay is usually used for regularisation, i.e. to avoid overfitting the data of the training set [2]. The fact that weight decay is not useful can be explained by the large amount of data provided, the regularisation is then done by the dataset itself.

However, having a single neural network to model the atmospheric reflectance was more convenient, a single network having 30 hidden units was then chosen (See Figure 3.4).

The errors are in Tables 3.8, the standard deviation is in target unit.

ν	Relative RMS (%)	Bias	Std
$\nu \leq 3$	14.8578	1.0690e-05	0.0038
$\nu > 3$	5.5822	9.0312e-06	0.0025
All ν	9.2339	1.6601e-05	0.0039

Table 3.8: Atmospheric reflectance, final model errors.

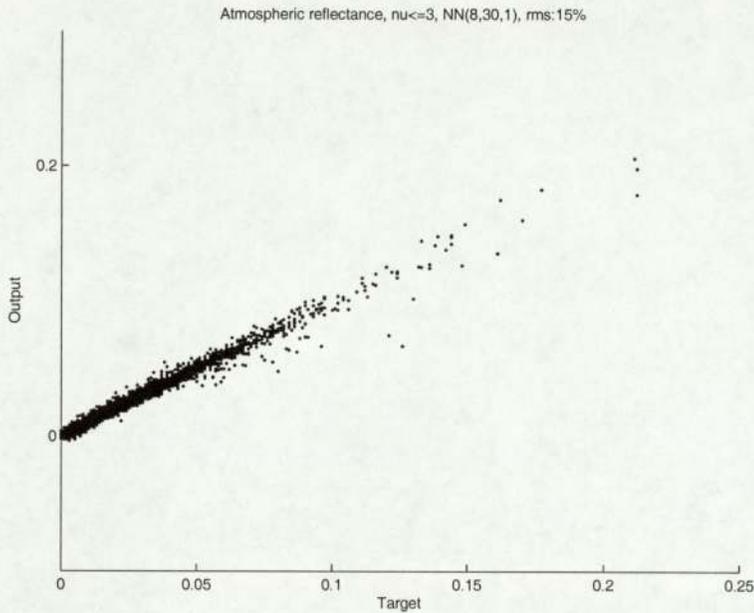


Figure 3.2: Atmospheric reflectance model, $\nu \leq 3$, inputs are $m_r, m_i, \tau, \nu, \theta_s, \theta_v, \phi, \lambda$, output is ρ_a

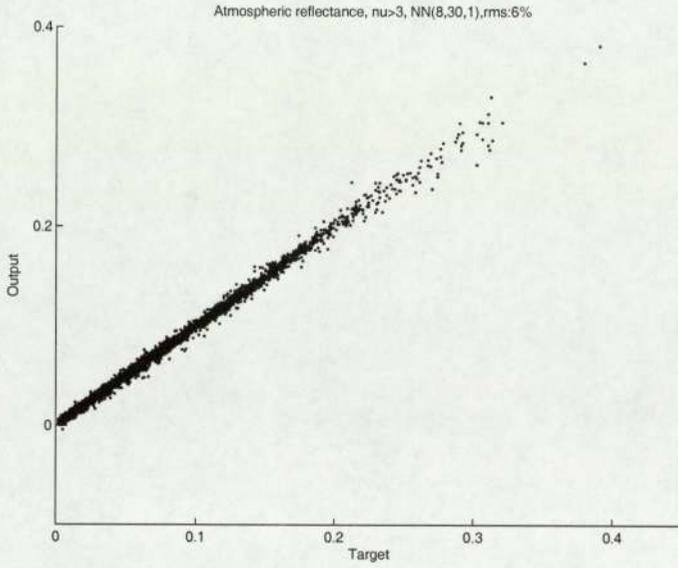


Figure 3.3: Atmospheric reflectance model, $\nu > 3$, inputs are $m_r, m_i, \tau, \nu, \theta_s, \theta_v, \phi, \lambda$, output is ρ_a

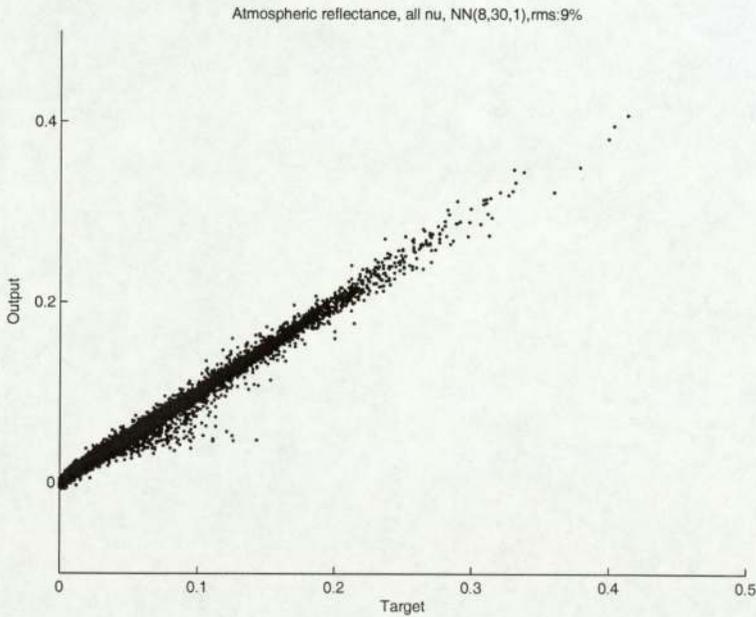


Figure 3.4: Atmospheric reflectance model, all ν , inputs are $m_r, m_i, \tau, \nu, \theta_s, \theta_v, \phi, \lambda$, output is ρ_a

3.2.2 Atmospheric transmittance

Data

The atmospheric database for transmittance contains 100000 samples corresponding to a basis built on the parameters ν , m_r , m_i , θ_s , θ_v , τ and λ (Table 3.9). λ is restricted to SeaWiFS visible wavelengths, which are the first six wavelengths. The measurements are shown in Table 3.10.

Training set	Test set	Total
70000	30000	100000

Table 3.9: Atmospheric transmittance dataset.

	ν	m_r	m_i	θ_s	θ_v	τ	λ	transmittance
mean	3.2507	1.4152	0.0139	34.9463	29.9548	0.1999	482.3415	0.6406
std	0.8539	0.0850	0.0155	9.4702	18.1245	0.1117	50.2553	0.1381

Table 3.10: Atmospheric transmittance variables (training set).

Model selection

Cross validation has been performed for networks having 10 to 50 hidden units, with 7 inputs and 1 output (Table 3.11).

Hidden units	Relative RMS (% of target)
10	4.2111
20	2.8712
35	2.3701
50	2.3066

Table 3.11: Atmospheric transmittance model errors.

As the transmittance is related to the path of light through the atmosphere, and more precisely to the depth of the atmosphere, using the cosines of the angles instead of the angles themselves could be useful. Thus, another comparison was done to determine if it was beneficial to use the cosines of the angles as inputs (Table 3.12). The results of this test shows that the network using the cosines of the angles is much worse than the one which is not using the cosines. The reason of such a difference is not clear, however as the interest of using cosines is not certain, the network directly using the angles will be preferred.

30 hidden units	Relative RMS (%)	Bias	Std
With cosines	19.0345	2.6715e-05	0.0262
Without cosines	1.8308	1.4343e-05	0.0025

Table 3.12: Choice of angles inputs for atmospheric transmittance.

Chosen model

The network chosen for the atmospheric transmittance model has 30 hidden units, with scg optimisation (Figure 3.5).

The errors are in Table 3.13, the standard deviation is in target units.

Relative RMS (%)	Bias	Std
1.8308	1.4343e-05	0.0025

Table 3.13: Atmospheric transmittance final model errors.

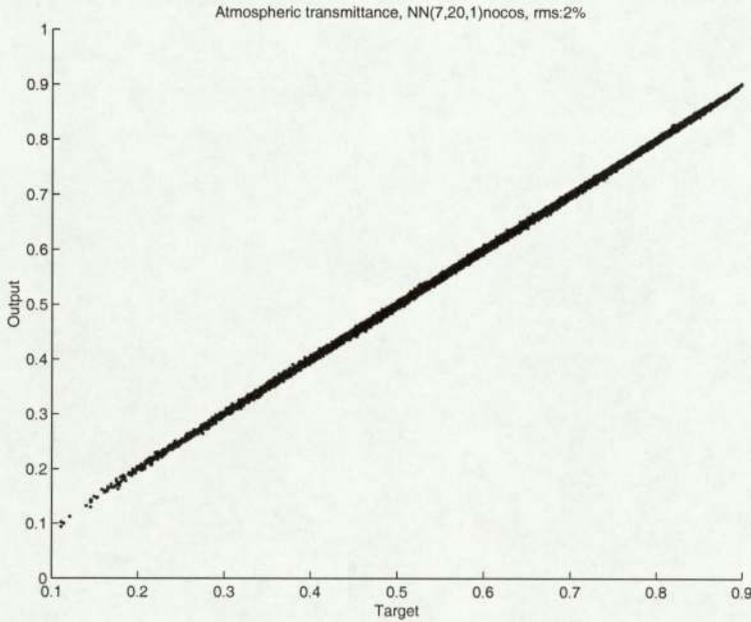


Figure 3.5: Atmospheric transmittance model, inputs are $m_r, m_i, \tau, \nu, \theta_s, \theta_v, \lambda$, output is t .

Use of Near Infra-Red bands

According to Chomko & Gordon [4], it is possible to retrieve ν (size distribution) and τ (optical thickness) from the reflectances in Near Infra-Red (NIR) SeaWiFS bands.

Thus an inverse model and a forward model learning those atmospheric parameters (ν : size distribution and τ : optical thickness) from the reflectances in NIR SeaWiFS bands have been setup and tested.

As the targets are discrete (by construction), the first attempt was using an MLP as classifier; this method was quite effective (Figure 3.6), but relies on an ‘artificial’ property of the data, thus a continuous model was set up.

The atmospheric database for transmittance contains 100000 samples; the measurements are in Table 3.14.

	ν	τ	θ_s	θ_v	ϕ_v	ρ_s/ρ_l
mean	3.2489	0.2	35	30	180	1.1316
std	0.8539	0.1118	9.4851	18.1628	113.8450	0.1086

Table 3.14: NIR bands model variables.

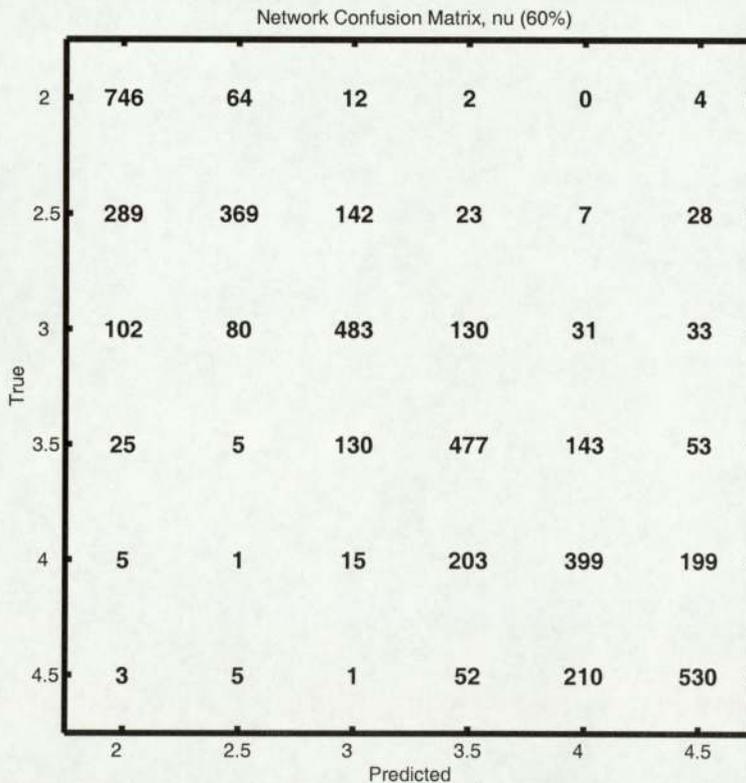


Figure 3.6: MLP Classifier for ν

A continuous model was built. According to Chomko & Gordon [4], a linear function would be able to retrieve ν from $\epsilon = \rho(s)/\rho(l)$: the ratio of the short and long NIR

band reflectances for a given instrument geometry. Therefore a forward model giving the ratio ϵ given ν , τ and the three observation angles has been trained.

As the data used for the training are discrete, the error of the network is difficult to calculate for the values where no training or test data is given, but the results of the forward model seem good. This will be confirmed in the tests of the Bayesian framework (Chapter 5).

The output of the network for different values of ν at a fixed given observation point are shown in Figure 3.7. As the observation point is fixed, this figure can be considered as a restriction of the global result which covers a basis of all the observation points.

This figure shows that for a known ϵ there is several possible values of ν in the training set, then it is impossible to retrieve ν from ϵ . However, the neural network is able to build a function to interpolate the data provided in the dataset, this neural network will then be inverted using the Bayesian framework so as to try to retrieve ν from ϵ in chapter 5.

The errors on $\epsilon = \rho(s)/\rho(l)$ are shown in Table 3.15.

Relative RMS (%)	Bias	Std
57.4606	-0.0019	0.5758

Table 3.15: Table of errors on ϵ , NIR final model.

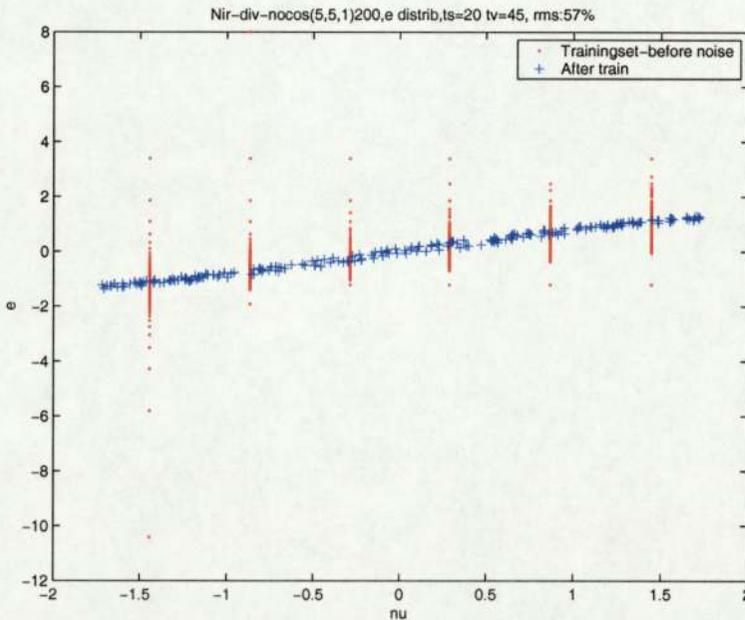


Figure 3.7: MLP Regression for ϵ , inputs are geometry(3 angles), ν and τ (normalised), output is $\epsilon = \rho(s)/\rho(l)$ (unnormalised).

Chapter 4

Building the Bayesian framework

Once the individual models were built, the next step was to build spatial priors and thus a spatial Bayesian framework on each one of the three parts of the model.

4.1 Principle of the Bayesian framework

The principle of the Bayesian framework is to use Bayes' rule to compute the probability of (a field of) parameter values given the observations. For example, the probability of a parameter $\boldsymbol{\tau}$ given the observations $\boldsymbol{\rho}$ is:

$$p(\boldsymbol{\tau}|\boldsymbol{\rho}) = \frac{p(\boldsymbol{\rho}|\boldsymbol{\tau})}{p(\boldsymbol{\rho})} \cdot p(\boldsymbol{\tau}) . \quad (4.1)$$

$p(\boldsymbol{\rho})$ is constant as the observations don't change, so

$$p(\boldsymbol{\tau}|\boldsymbol{\rho}) \propto p(\boldsymbol{\rho}|\boldsymbol{\tau}) \cdot p(\boldsymbol{\tau}) . \quad (4.2)$$

Then, assuming that the noise on the components τ_i of $\boldsymbol{\tau}$ is iid,

$$p(\boldsymbol{\tau}|\boldsymbol{\rho}) \propto \left[\prod_{i=1}^n p(\rho_i|\tau_i) \right] \cdot p(\boldsymbol{\tau}) . \quad (4.3)$$

Then the error on the parameter, which is the negative log likelihood of the parameter is

$$\begin{aligned} E &= -\log(p(\boldsymbol{\tau}|\boldsymbol{\rho})) \\ &= -\log \left(\prod_{i=1}^n p(\rho_i|\tau_i) \cdot p(\boldsymbol{\tau}) \right) + constant \\ &= -\sum_{i=1}^n \log(p(\rho_i|\tau_i)) - \log(p(\boldsymbol{\tau})) + constant . \end{aligned} \quad (4.4)$$

As Gaussian noise is assumed on the data, the expressions of $p(\rho_i|\tau_i)$ and $p(\boldsymbol{\tau})$ are as follows:

$$p(\rho_i|\tau_i) = \frac{1}{\sqrt{2\pi\sigma_\rho^2}} \exp \left(-\frac{1}{2} \frac{\delta_i^2}{\sigma_\rho^2} \right) , \quad (4.5)$$

where σ_ρ represents the confidence we have when doing the comparison. If no noise is added on the observation (or if the observations are assumed to be noise-free), σ_ρ is the sum of the error due to the neural network and the uncertainty due to the physical model used to simulate the data in first place. If some noise is added to the observations, or if the observations are not assumed to be noise-free, σ_ρ is the sum of the noise due to the neural network errors, the noise in the observations and the physical model uncertainty.

δ is the difference between the real observations and the values predicted by the model:

$$\delta_i = f(\tau_i) - \rho_i . \quad (4.6)$$

The prior $p(\boldsymbol{\tau})$ is given by

$$p(\boldsymbol{\tau}) = \frac{1}{\sqrt{2\pi}|\Sigma|} \exp \left(-\frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{\tau})'\Sigma^{-1}(\boldsymbol{\mu} - \boldsymbol{\tau}) \right) , \quad (4.7)$$

where $\boldsymbol{\mu}$ is the prior mean of $\boldsymbol{\tau}$. The covariance matrix Σ encapsulates our prior knowledge about the parameter $\boldsymbol{\tau}$. Each parameter has its own prior hyperparameters which define the Gaussian process; the covariance matrix is computed using the hyperparameters of this Gaussian process. The parameters used for each prior are discussed below.

To apply the Bayesian framework it was necessary to build priors for the different parameters. The priors on the different parameters have been chosen according to physical and biological knowledge.

4.2 Atmospheric priors

For the NIR bands model, Gaussian processes priors were defined (see Table 4.1).

The basis function and length scale were chosen according to prior knowledge. The basis function used is the polyexponential function:

$$C(r) = v.(1 + \frac{r}{l}) \exp \left(\frac{-r}{l} \right) + n , \quad (4.8)$$

where v is the variance of the Gaussian process, l is the length scale and n is the nugget. The nugget variance is the variance of the semivariogram, i.e. of the Gaussian process, for a zero distance.

The prior function and the scale length were arbitrarily chosen as no physical value was available; however it can easily be modified according to real measurements. The mean and variance were calculated from the training data.

At first, no prior was defined for m_r and m_i , then, priors were built according to the characteristics of the training data.

Parameter	Prior Type	Function	Variance	Scale length	Nugget	Mean
τ	Gaussian process	polyexp	0.1	100km	0.01	0.2
ν	Gaussian process	polyexp	0.6	100km	0.01	3.2

Table 4.1: Priors for the NIR bands parameters

4.3 Oceanic priors

For the oceanic model, log Gaussian processes priors were built (see Table 4.2).

The basis function, length scale, variance and nugget were chosen according to the semivariograms build by [5] from real observations.

The basis function used is the exponential function:

$$C(r) = v. \exp\left(\frac{-r}{l}\right) + n, \quad (4.9)$$

Where v is the variance of the Gaussian process, l is the length scale and n is the nugget.

The mean was calculated from the training data.

Parameter	Prior Type	Function	Variance	Scale length	Nugget	Mean
b^0	Gaussian process	exp	0.1	20km	0.01	$\log(0.3)$
$[chl]$	Gaussian process	exp	0.42	21km	0.03	$\log(0.5)$

Table 4.2: Priors for the oceanic parameters

4.4 Implementation of the Bayesian framework

The goal is to retrieve the parameters using the MAP (Maximum A Posteriori) method, for example the τ which gives the maximum $p(\tau|\rho)$. The Bayesian process starts from an uniform field at the expected mean, to facilitate the convergence. However, it can also start from a random point, or from a point previously computed as a first guess. The parameter field is optimised using a scalar gradient based optimisation [2], using the error function described in equation 4.4 and its gradient. The implementation of the final framework has been done together with the testing and is described in the next chapter.

Chapter 5

Testing of Bayesian frameworks

A test of the Bayesian frameworks was done using simulated data. A field of parameters was created using the Gaussian process (a realisation from the prior), then the forward model was used to simulate the observations. Noise was added to the simulated observation and then the retrieval of the original parameter from the noisy observations was performed. The error between the retrieved parameter and the original parameter was then computed. Results on [chl] were compared with SeaWiFS specifications, which require to have an error less than 35% of the signal.

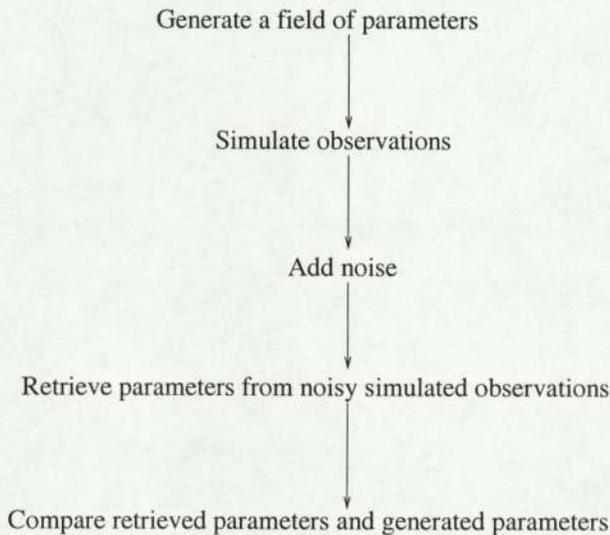


Figure 5.1: Diagram of the Bayesian framework test.

5.1 Test on NIR bands

The Bayesian framework was first built for the NIR bands model, the goal was to retrieve either ν or τ given the observations ($\epsilon = \rho(s)/\rho(l)$) at a given satellite position,

characterised by three geometry angles.

5.1.1 Retrieval of a single variable

At first the framework was built to retrieve τ , assuming ν to be known.

This retrieval has been done without and with noise added on the NIR observations (Figures 5.2 and 5.3). The noise added was Gaussian noise with zero mean and standard deviation of 1% of the standard deviation of the data (σ_ϵ). Adding more noise on $\epsilon = \rho(s)/\rho(l)$ was leading to much worse results. The optimisation, as well as all the other optimisations in this chapter, was done with a limit of 1000 epochs. The relative weight of the prior is related to the amount of noise through equations 4.5 and 4.4.

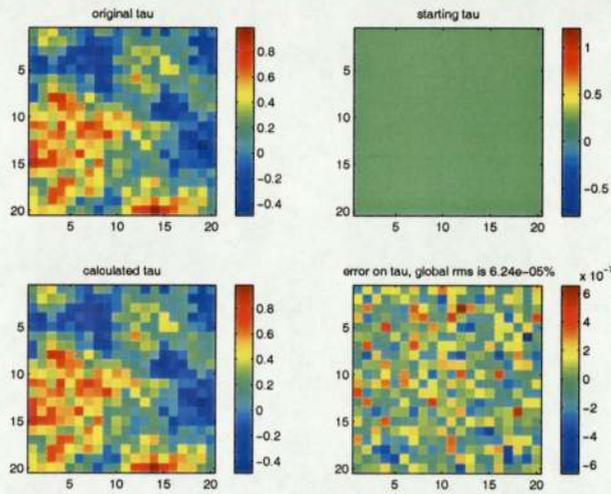


Figure 5.2: Retrieval of a τ field given the NIR bands observations using a Bayesian framework and an MLP, without noise.

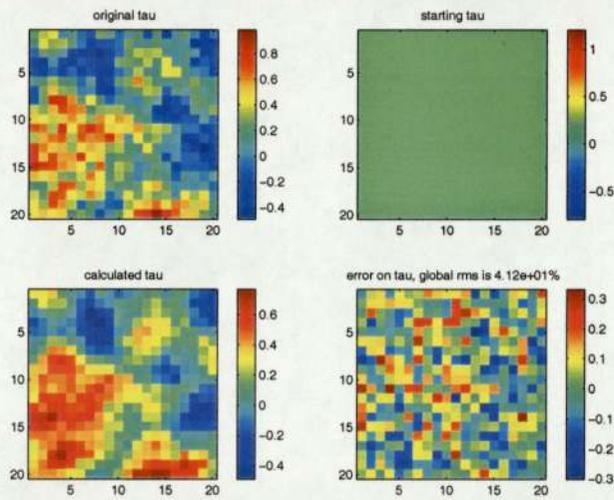


Figure 5.3: Retrieval of a τ field given the NIR bands observations using a Bayesian framework, a noise of 1% of $mean(\epsilon = \rho(s)/\rho(l))$ has been applied to the observations.

5.1.2 Joint retrieval of two variables

The second step was to retrieve jointly ν and τ . Different methods were explored, based on different networks.

- The first method used a network having ν , τ and the viewing angles as inputs and ϵ as output. This method generally found a local minimum, which gave a perfect ϵ for a good ν but a wrong τ as shown in Figure 5.4.
- Next a network having ν , τ , the viewing angles and λ as inputs and $\rho(\lambda)$ as output was setup, this method was able to retrieve τ but not ν (see Figure 5.5).

These tests tended to show that τ can be retrieved from NIR bands reflectances whereas ν can be retrieved from ϵ . A network should theoretically be able to retrieve ν and τ from the NIR bands reflectances, but it might be better to also include the information about ϵ explicitly. The two different solutions were then tested.

- The first one was done using a network having ν , τ and the viewing angles as inputs and the reflectances in NIR bands and ϵ as outputs. See Figure 5.6 and Table 5.1.
- The second one was done using the two first networks and iterating, guessing τ with a fixed ν , then ν with a fixed τ . See Figure 5.7 and Table 5.2.

These two methods gave different results, both were correct. The retrieval using NIR bands and ϵ was more direct, more elegant and slightly better, thus it was used in the global framework.

	Relative RMS (% of target)	Bias	RMS
$\rho(\lambda_1)$	0.0042	-0.0000	0.0000
$\rho(\lambda_2)$	0.0036	-0.0000	0.0000
$\rho(\lambda_3)$	0.0050	-0.0000	0.0000
$\rho(\lambda_3)/\rho(\lambda_2)$	0.0355	-0.0000	0.0000
ν	0.0443	-0.0000	0.0003
τ	0.0163	0.0000	0.0000

Table 5.1: Table of errors, ν and τ retrieval from NIR bands and ϵ , plotted on Figure 5.6. λ_1 , λ_2 and λ_3 are SeaWiFS three last wavelengths (Near Infra Red bands).

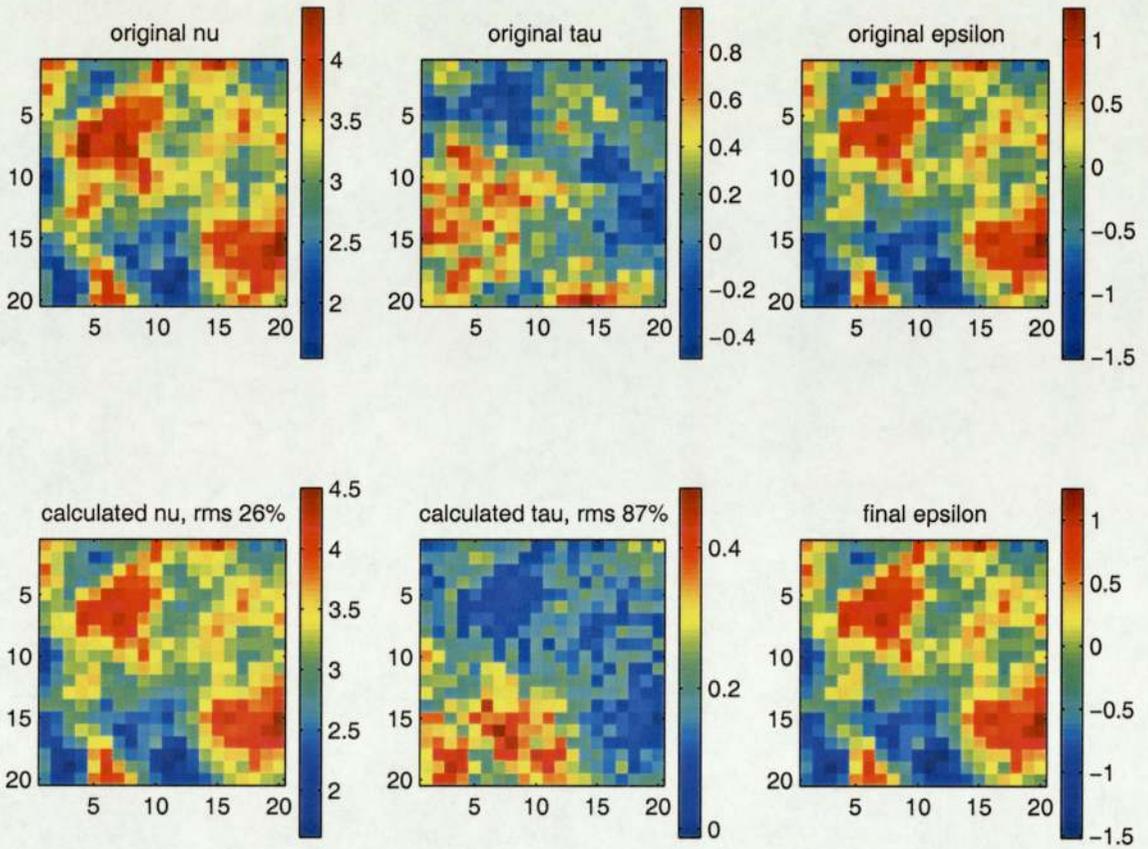


Figure 5.4: Joint retrieval of ν and τ fields given $\epsilon = \rho(s)/\rho(l)$ using a Bayesian framework.

	Relative Error (percentage of target)	Bias	RMS
$\rho(\lambda_1)$	0.7859	-0.0000	0.0003
$\rho(\lambda_2)$	0.8697	0.0000	0.0004
$\rho(\lambda_3)$	0.1638	0.0000	0.0001
$\rho(\lambda_3)/\rho(\lambda_2)$	0.2296	0.0000	0.0001
ν	2.4794	0.0028	0.0151
τ	9.7368	0.0042	0.0254

Table 5.2: Table of errors, ν and τ retrieval iterating on two networks, plotted on Figure 5.7.

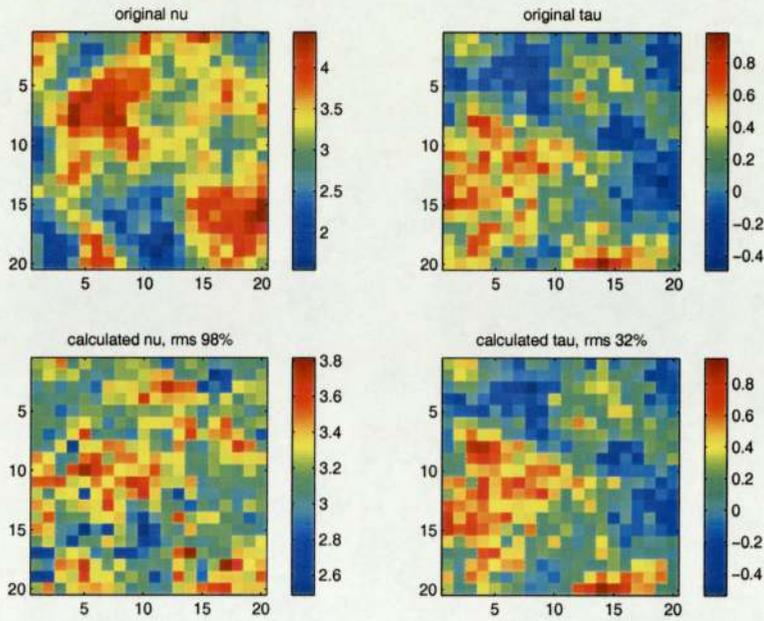


Figure 5.5: Joint retrieval of ν and τ fields given NIR bands observations (instead of $\epsilon = \rho(s)/\rho(l)$) using a Bayesian framework.

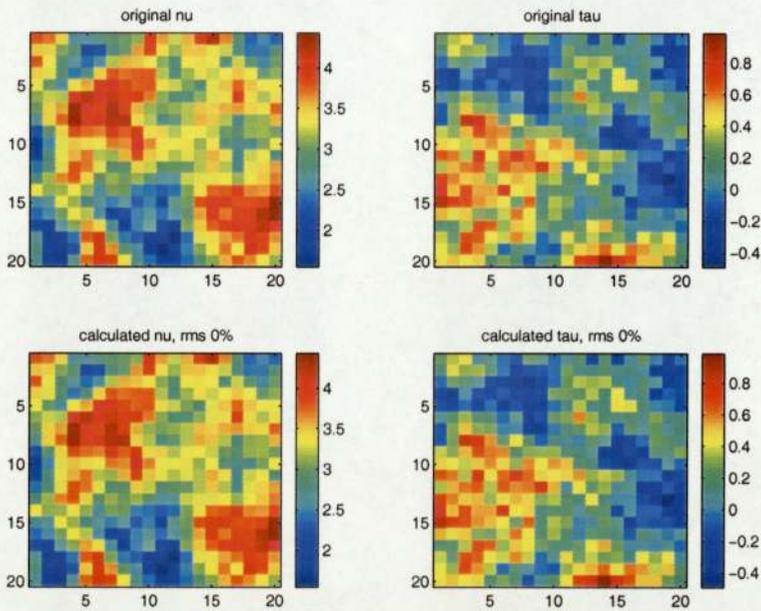


Figure 5.6: Joint retrieval of ν and τ fields given NIR bands observations and $\epsilon = \rho(s)/\rho(l)$ using a Bayesian framework.

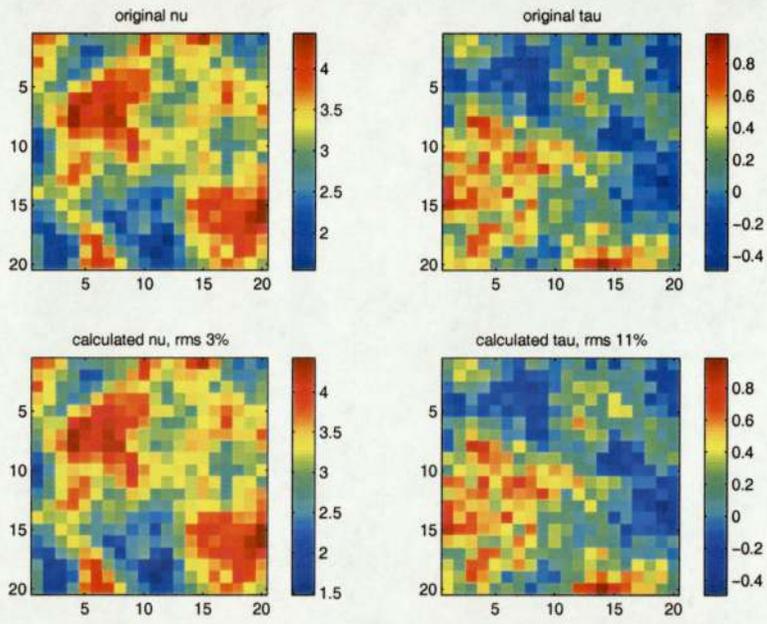


Figure 5.7: Retrieval of ν and τ fields iterating on two networks.

5.2 Test on oceanic model

5.2.1 Retrieval of a single variable

The Bayesian framework was then adapted to the oceanic model to retrieve the chlorophyll concentration given b^0 and the observations at the different SeaWiFS visible wavelengths, at a given satellite position (see Figures 5.8, 5.9 and 5.10).

The first tests were done using simulated data with added noise, the relative weight of the prior is related to the amount of noise on the observations through equations 4.5 and 4.4.

A comparison of consistent and inconsistent priors has been made (see Figure 5.11). The prior is said to be consistent with the noise when its weight is adapted according to the amount of noise added. The prior is deemed inconsistent with the noise when its weight is not adapted.

The noise added on the observations is Gaussian and has a standard deviation of

$$std = noise\ factor * mean(observations) . \quad (5.1)$$

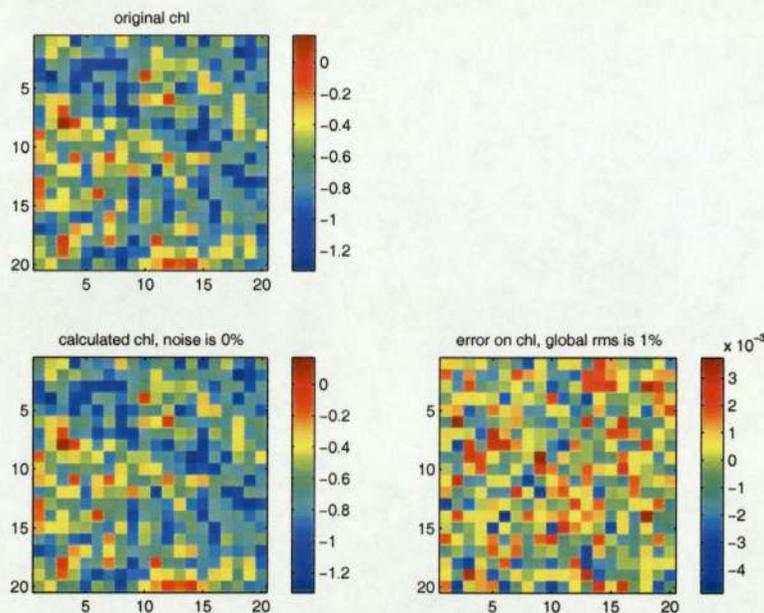


Figure 5.8: Retrieval of a $\log([chl])$ field given the observations using a Bayesian framework, without noise.

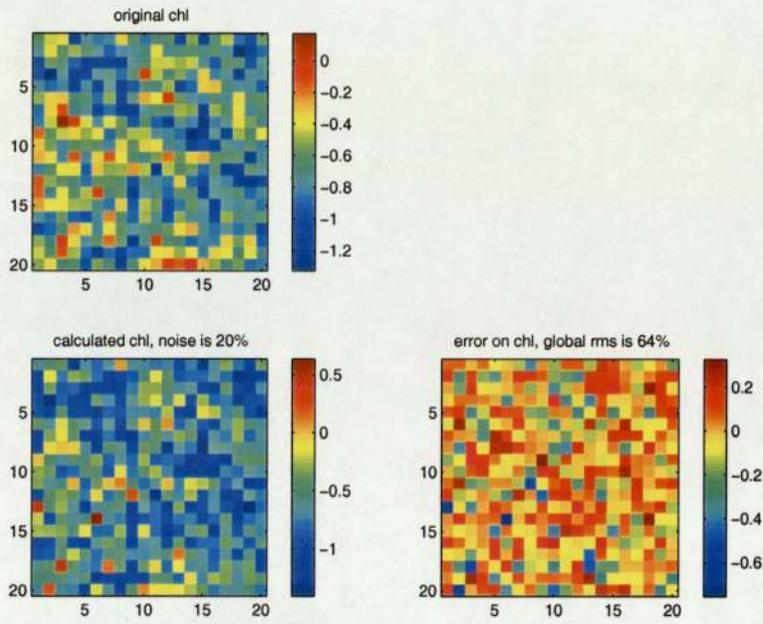


Figure 5.9: Retrieval of a $\log([chl])$ field given the observations using a Bayesian framework. A noise of 20% of $mean(\rho)$ has been applied on the observations, the noise assumption is not consistent with the noise added.

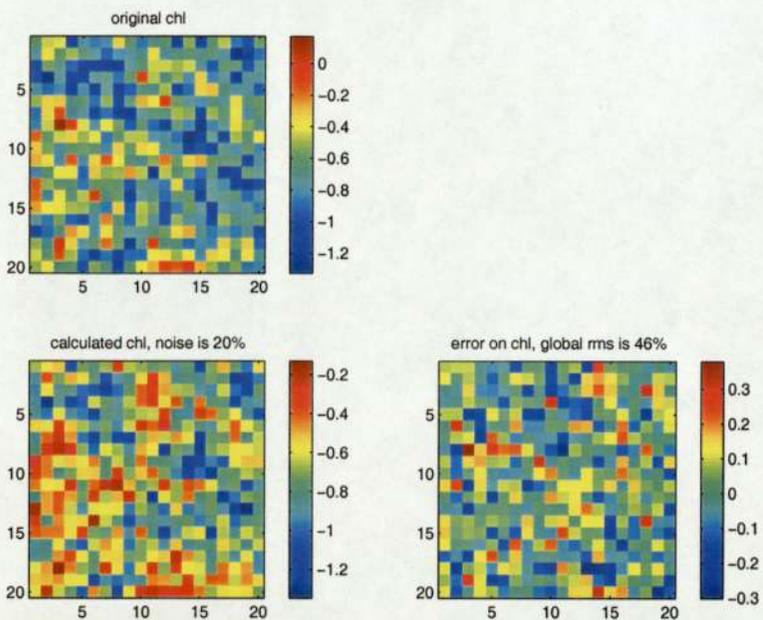


Figure 5.10: Retrieval of a $\log([chl])$ field given the observations using a Bayesian framework. A noise of 20% of $mean(\rho)$ has been applied on the observations, the noise assumption is consistent with the noise added.

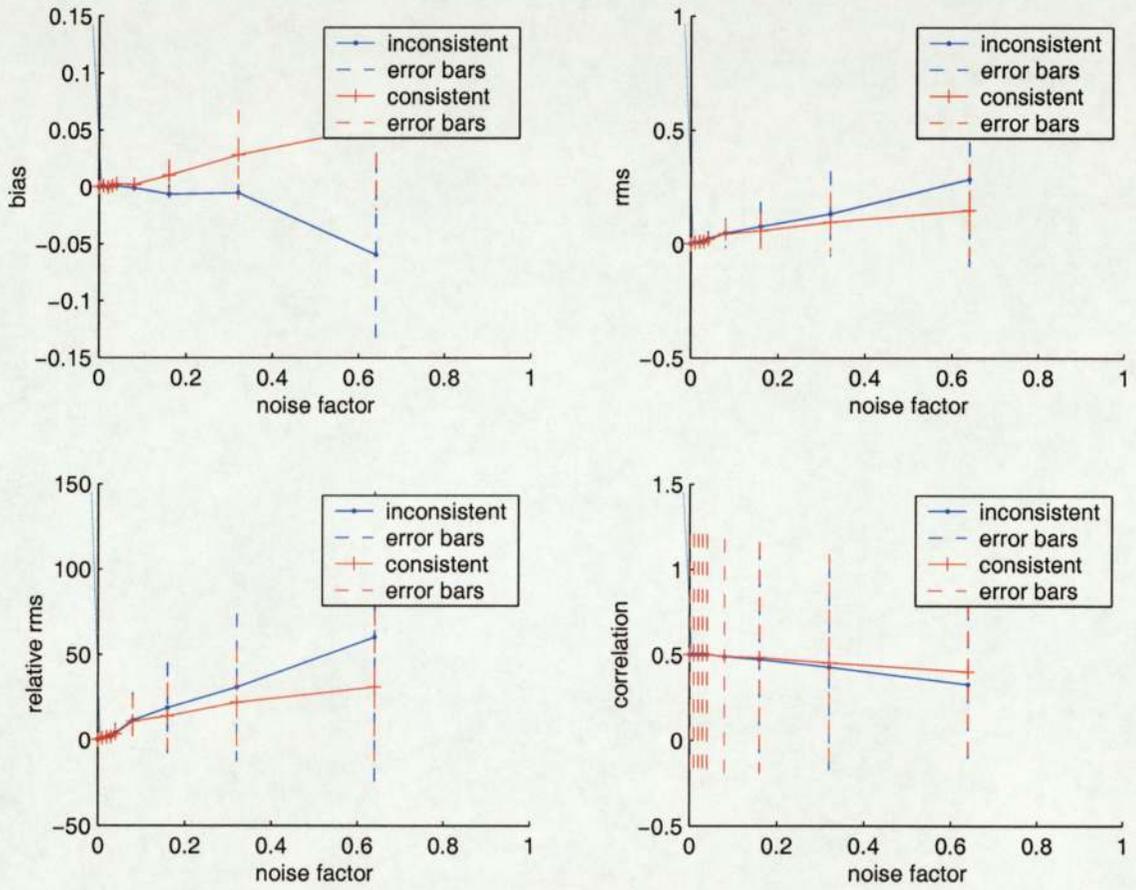


Figure 5.11: Comparison between priors with noise assumption consistent or inconsistent with the added noise. Average on 100 experiments.

5.2.2 Joint retrieval of two variables

Then the Bayesian framework was tested to retrieve jointly two variables, b^0 and $[chl]$. (See Figures 5.12 and 5.13)

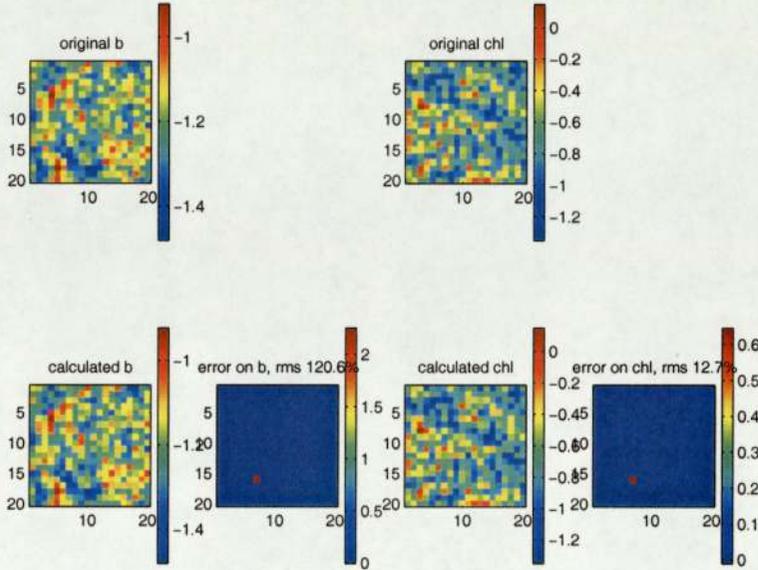


Figure 5.12: Retrieval of $\log(b^0)$ and $\log([chl])$ fields given the observations using a Bayesian framework, without noise.

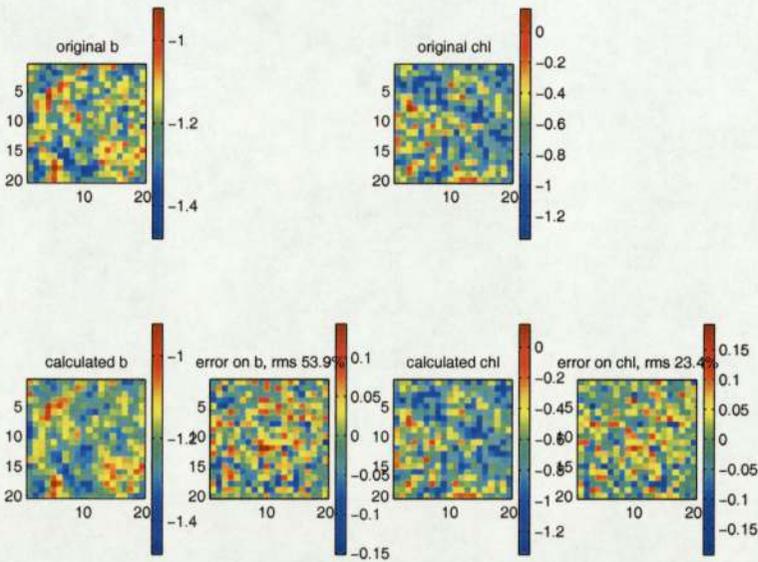


Figure 5.13: Retrieval of $\log(b^0)$ and $\log([chl])$ fields given the observations using a Bayesian framework. A noise of 5% of $mean(\rho)$ has been applied on the observations, the noise assumption is consistent with the noise added.

5.3 Conclusion of the test of the Bayesian framework

The tests of the Bayesian framework on small problems, using only one neural network at once, were successful. The error on the retrieval of [chl] was less than 13%, whereas SeaWiFS requirement is only to have an error under 35% of the signal. However this was done on a sub-problem of the real global problem. The use of priors was verified and the behaviour of the retrieval process when adding noise was tested. These tests have shown that the Bayesian framework was reliable and useful on the sub-problems, then the framework was tested on the global problem.

Chapter 6

Tests on simulated data

Once the Bayesian retrieval process had been tested on the sub-problems described previously, the global Bayesian framework was built for the whole problem.

The tests on the global model were first done using simulated data. The test was done by generating fields of atmospheric and oceanic parameters, simulating the observations using the neural networks, noising these observations, retrieving the parameters from the noisy observation and comparing the retrieved parameters and the original simulated parameters (see Figure 6.1).

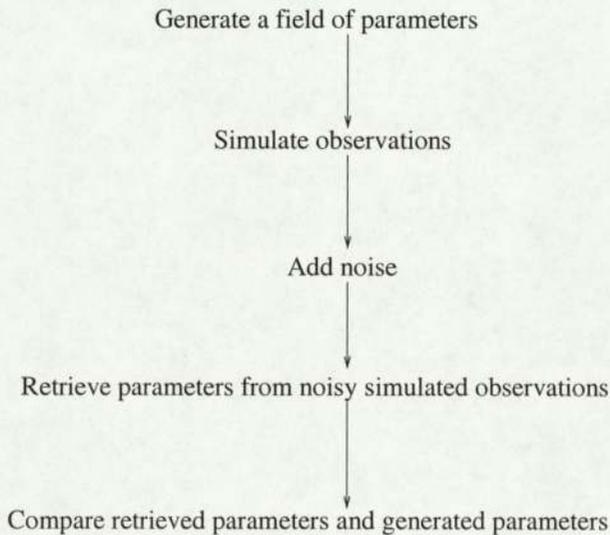


Figure 6.1: Testing process.

6.1 First retrieval framework

The retrieval was first attempted in two steps. The first step was to retrieve ν and τ from the NIR bands. This was done by inverting the NIR model presented before, as

only one neural network is involved this part was fairly accurate. This first optimisation was done with a limit of 500 epochs.

The second step was to retrieve m_r , m_i , b^0 and [chl] from the visible bands. This second optimisation was done with a limit of 5000 epochs. This method gave poor results, the model staying in local minima, ρ_{toa} being close to the real observations but m_r , m_i , b^0 and [chl] being wrong (Table 6.1).

	Relative Error (percentage of target)	Bias	RMS
ρ_{toa}	7.654508e+00	4.663121e-05	2.785888e-03
ν	2.209596e-05	-9.396992e-10	2.447876e-08
τ	3.574356e-06	4.758629e-10	2.682411e-09
m_r	3.606464e+02	-1.315479e-01	3.816374e-01
m_i	1.765500e+03	-6.874906e-02	2.344982e-01
b^0	1.438938e+03	-2.546753e-01	1.185927e+00
[chl]	2.461670e+03	4.343113e-01	1.881375e+00

Table 6.1: Table of errors, global retrieval in two steps.

6.2 Second retrieval framework

To correct the problem that the algorithm found local minima, another method was tried. The second method was done in three steps.

- The first step was to retrieve ν and τ from the near infra red bands.
- The second step was to perform a first guess to retrieve m_r and m_i from ρ_{toa} in visible wavelenths. It was first attempted to use a direct inverse model learning both m_r and m_i from the TOA reflectance and the geometry, but as the training data is discrete (see Table 1.3) this method was inappropriate. Thus two classifiers were built, one classifying m_r from the TOA reflectance and the geometry and the other classifying m_i (see Figures 9.1 and 9.2 in appendix). As m_i can take 6 different values in the dataset used, a single classifier wasn't effective, a pyramidal structure of three classifiers was then built, the first classifier choosing between low and high values (see Figures 6.2 and Figures 9.3, 9.4, 9.5 in Appendix).
- The third step was to retrieve m_r , m_i , b^0 and [chl], using the previous results as starting points for m_r and m_i (see Figure 6.3).

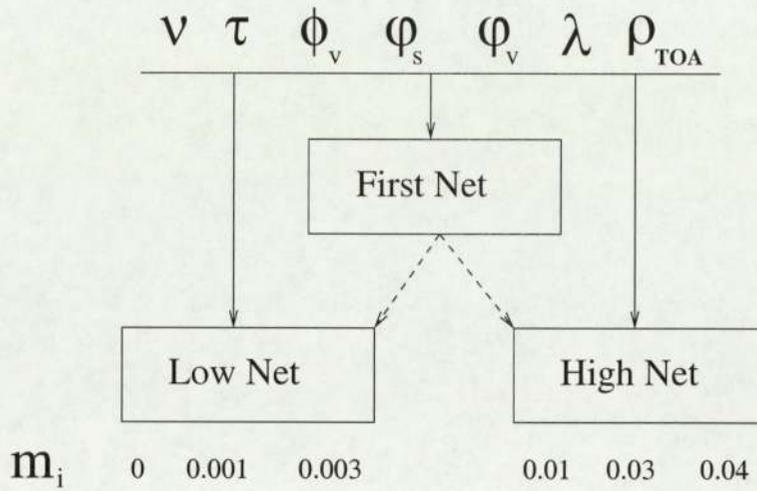


Figure 6.2: Pyramidal classifiers to retrieve m_i from ρ_{toa} .

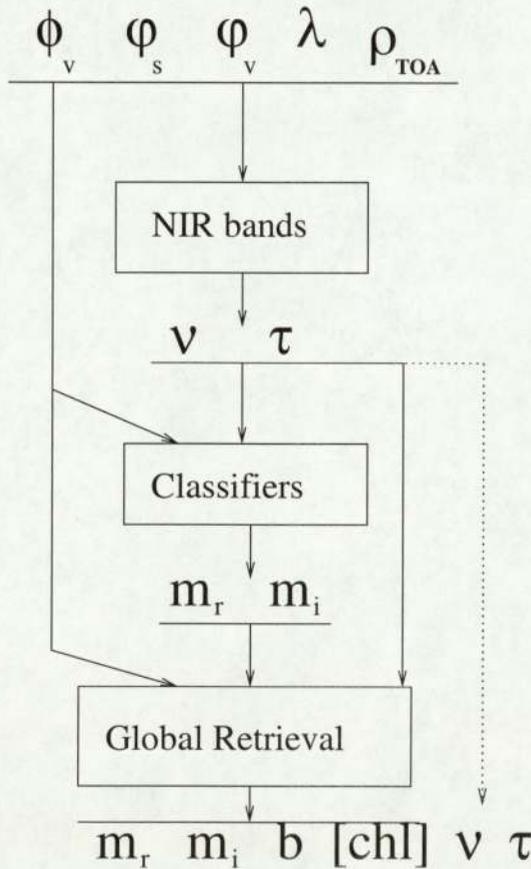


Figure 6.3: Second global retrieval framework, using the NIR bands to retrieve ν and τ , and the classifiers to first guess m_r and m_i .

6.3 Physically wrong results

The first tests on simulated data gave good results for ν and τ , but very poor results for m_r , m_i , b^0 and $[chl]$, giving results outside the physical range of the parameters, with the corresponding observations near the real ones, (see Figures 6.4 and 6.5).

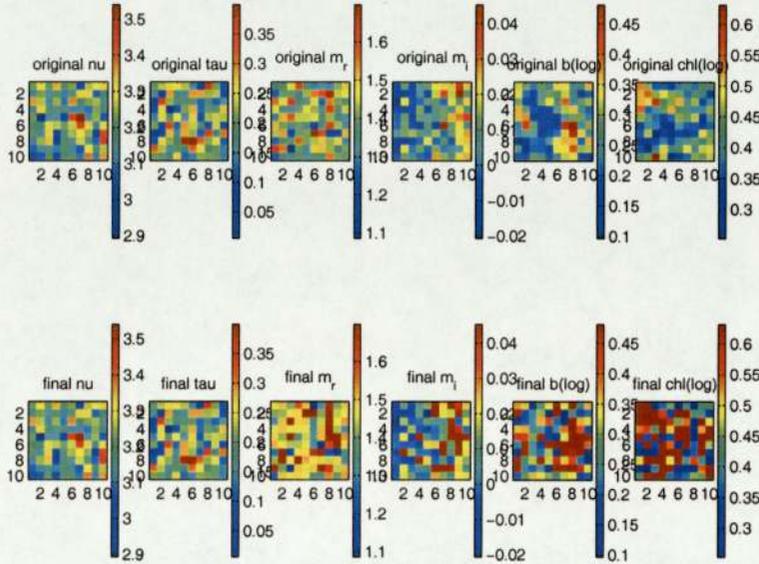


Figure 6.4: Global retrieval using MLPs - retrieved parameters.

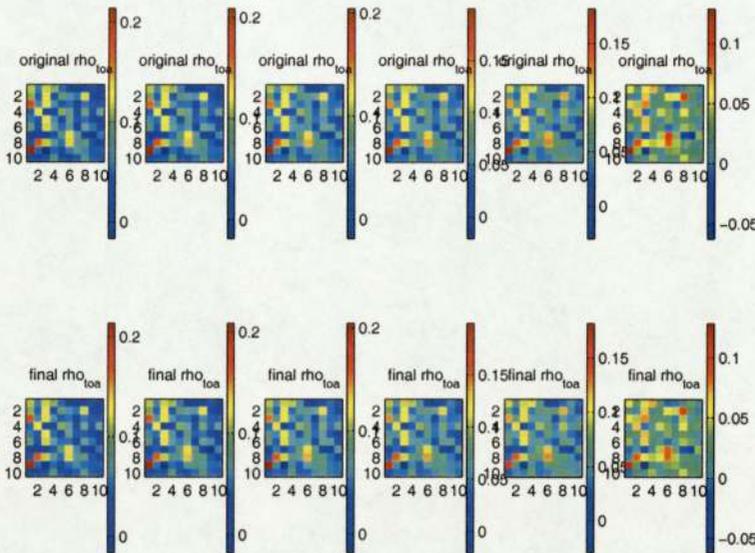


Figure 6.5: Global retrieval using MLPs - observations at the SeaWiFS visible wavelengths.

Four solutions were then considered.

6.3.1 MLP vs RBF

The first tests were showing that wrong parameters were found, leading to observations corresponding to the real ones. This behaviour can be attributed to the use of MLPs, which are trained to give results corresponding to the training set in the data range of the inputs given in the training set, but are not constrained outside this data range. Thus MLPs can lead to a minimum outside the physical range of the different parameters which gives observations close to the real ones.

The use of RBF was then considered, as the RBF are using Gaussian basis functions, the output given for an input outside the training set range tends to zero. The figures and errors of the trained RBFs are in the appendix (Table 9.1, Figures 9.6, 9.7 and 9.8). The RBF for atmospheric transmittance was not as efficient as the corresponding MLP, but increasing the size of the network (the number of basis functions) would have increased too much the computational cost of the training. Therefore, the maximum size of the RBF was kept at 64 basis functions.

6.3.2 Gaussian mixture model priors over the data space

The second solution was to fit Gaussian mixtures to each variable in the training set so as to set simple priors on each of them (See equation 4.3). These priors would constrain the parameters in the physically real data range (see Figure 6.6). As the data samples are not uniformly distributed in the data range, the Gaussian mixtures can fit artefacts due to the conception of the basis of parameters, as shown for the parameter m_r in Figure 6.6. These priors cannot be used where other priors are already in use, thus they will be used where no other prior is available.

6.3.3 Log parameter space

The third solution was to work in log parameter space. This solution would prevent the parameters becoming negative when they can't physically be. Moreover, the oceanic parameters, b^0 and [chl], and the atmospheric optical parameters, ν and τ are more likely to be Gaussian distributed in log space [3]. The figures and errors of the trained MLPs and RBFs are in the appendix (Table 9.2, Figures 9.12, 9.13, 9.14 and 9.9, 9.10, 9.11). The basic framework was then tested in logspace, (Figures 6.7 and 6.8 and Table 9.8 in appendix), the results were not better than not using the logspace. Moreover, the transformation from logspace to real space induced more error: If x is the retrieved parameter in logspace, the real value of the parameter is

$$y = \exp(x) , \quad (6.1)$$

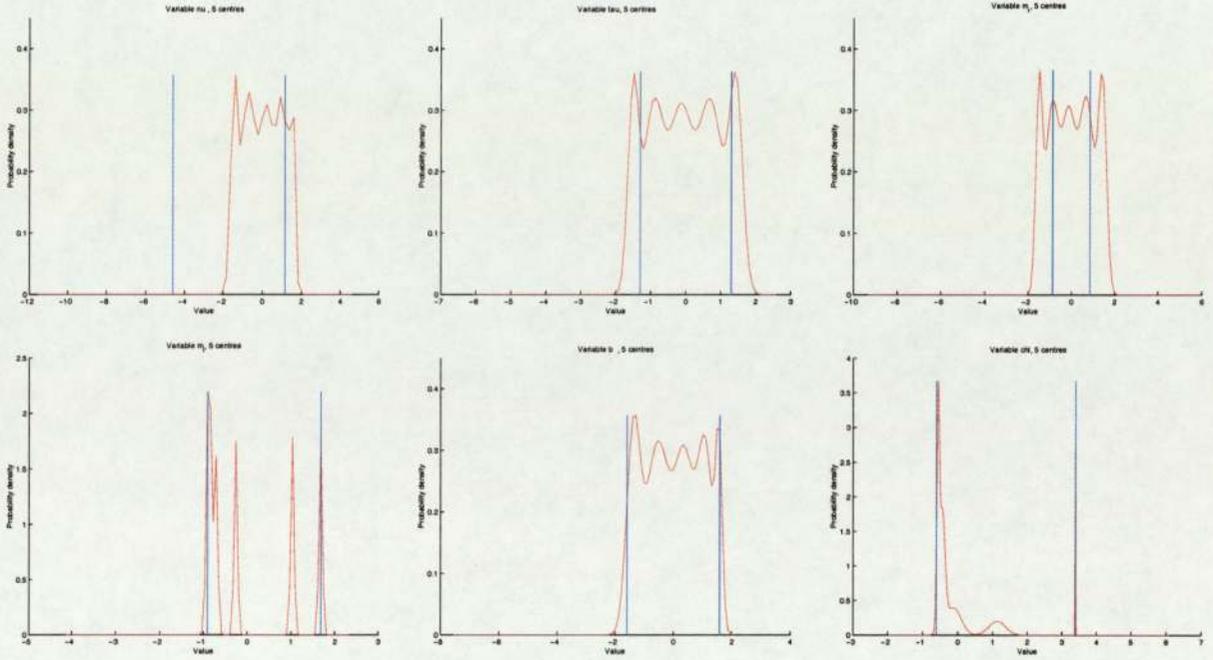


Figure 6.6: Gaussian mixture priors on ν, τ, m_r, m_i, b^0 and $[chl]$. Vertical blue lines are the limits of the data range in the training set, red curve is the Gaussian mixture prior.

and then the error on the real parameter is

$$\begin{aligned} \Delta y &= \frac{\partial y}{\partial x} \cdot \Delta x \\ &= \exp(x) \cdot \Delta x . \end{aligned} \tag{6.2}$$

As the retrieval using logspace was not as good as the normal retrieval, this solution was not kept.

6.3.4 Improved first guess

The fourth solution was to try to improve the first guess of the different parameters. In the previous framework, ν and τ were retrieved first, then m_r and m_i were first guessed and at last m_r, m_i, b^0 and $[chl]$ were retrieved. Several improvements were examined:

- In the first framework, no first guess was done for b^0 and $[chl]$. However having a first estimation of these two parameters would be useful. Then a direct inverse model giving b^0 and $[chl]$ from $\nu, \tau, \theta_s, \theta_v, \phi, \lambda$ was built, as this problem is multi-modal, an MDN was used. Unfortunately this problem seemed to be too difficult to be directly learnt by an MDN (Figures 6.9, 6.10, 6.11). This solution was then abandoned.
- The first guess of m_r and m_i was done using classifiers, but was not very efficient, a better first guess for m_r and m_i would improve the global efficiency. A more

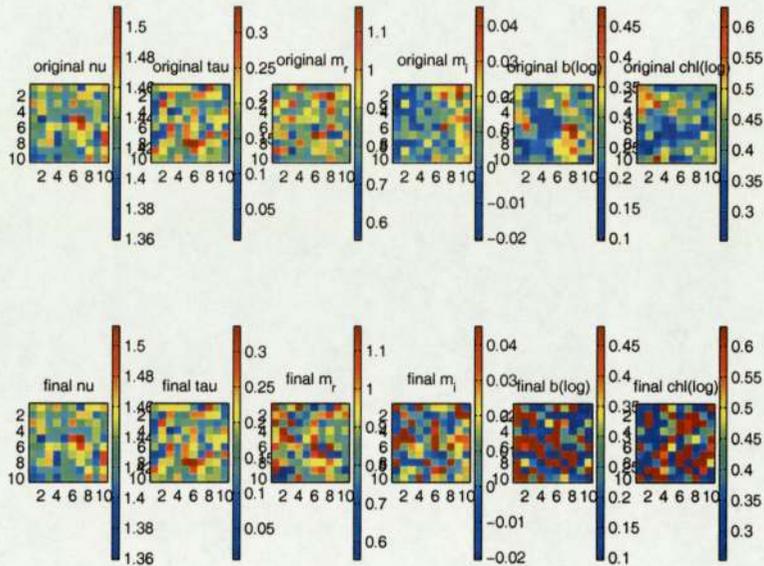


Figure 6.7: Second framework - Retrieval using the complex forward model in logspace (MLPs): parameters.

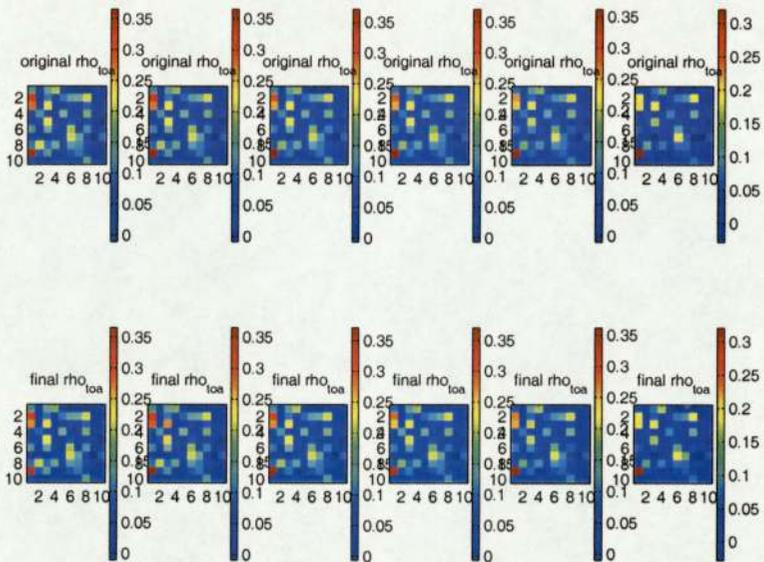


Figure 6.8: Second framework - Retrieval using the complex forward model in logspace (MLPs): observations.

efficient first guess was tried by inverting a network which was able to compute ρ_{toa} from ν , τ , m_r and m_i . As ν and τ are retrieved first, inverting this network gave m_r and m_i (See Figure 9.15 in appendix). This network will be referred to as *first guess model* for m_r and m_i .

- A solution to first guess b^0 and [chl] as well was examined and implemented using a single neural network able to compute ρ_{toa} from ν , τ , b^0 and [chl] (Figure 9.16

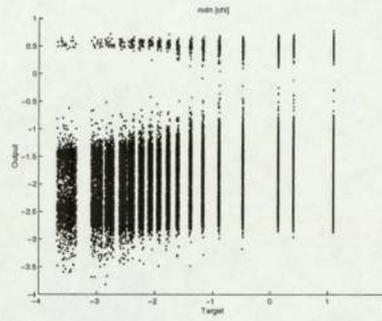


Figure 6.9: Network for first guess of b^0 and [chl], inputs are $\tau, \nu, \theta_s, \theta_v, \phi, \lambda$, outputs are b^0 and [chl]. Plot of target versus output for [chl].

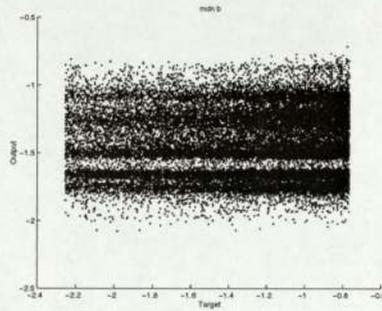


Figure 6.10: Network for first guess of b^0 and [chl], inputs are $\tau, \nu, \theta_s, \theta_v, \phi, \lambda$, outputs are b^0 and [chl]. Plot of target versus output for b^0 .

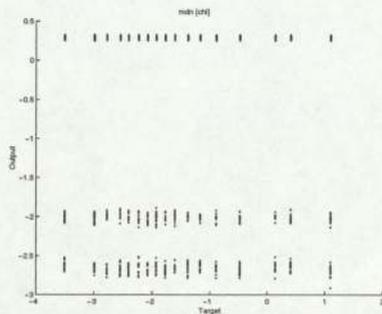


Figure 6.11: Network for first guess of [chl], inputs are $\tau, \nu, \theta_s, \theta_v, \phi, \lambda$, output is [chl]. Plot of target versus output for [chl].

in appendix). This solution, as well as the previous one, is based on a restriction of the problem, as the networks are predicting ρ_{toa} without all the parameters. This network was not as good as the previous one, however, it was tested in the global framework. This network will be referred to as *first guess model for b^0 and [chl]*.

- Finally, a network computing ρ_{toa} from ν, τ, m_r, m_i, b^0 and [chl] was built as a simpler and less accurate modelisation to the global problem, to perform a first

retrieval for m_r, m_i, b^0 and [chl] (see Figure 9.17 in appendix). This network will be referred to as *simple forward model*.

Those three solutions can be applied successively before the last retrieval to start closer to the true parameters and avoid local minima.

6.4 Third retrieval framework

The third framework (Figure 6.12) used the first guess model for m_r and m_i , for which two different solutions, the classifiers and the inversion of a neural network were tested.

The first guess model for b^0 and [chl] was also implemented as well as the simple forward model for the first retrieval of m_r, m_i, b^0 and [chl].

The previously described Gaussian mixture models were used where no prior knowledge was available.

The tests were first done using MLPs.

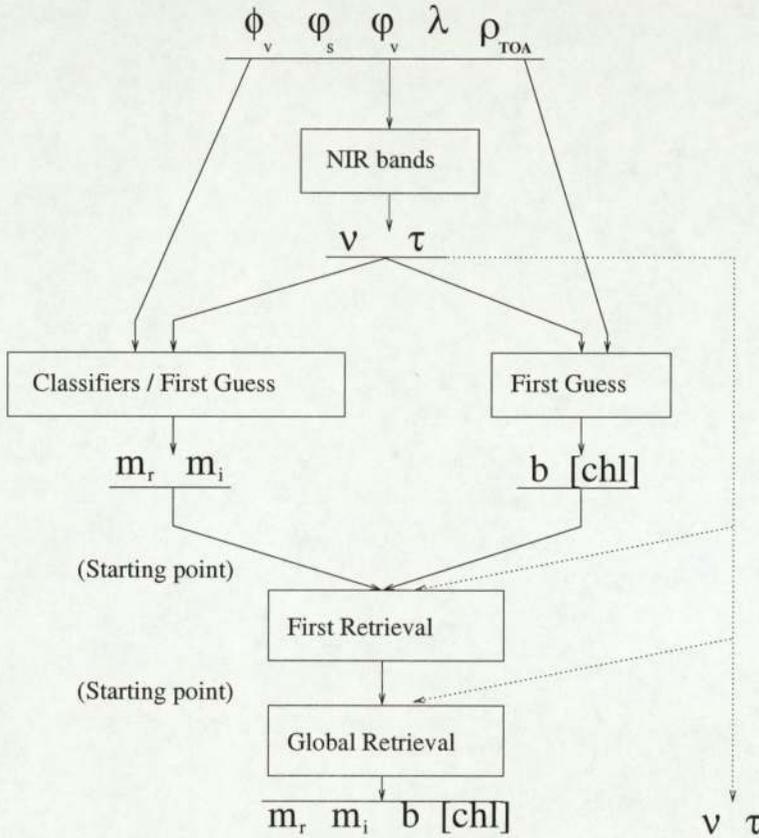


Figure 6.12: Third Global retrieval framework.

This new framework was more complicated; the additional networks were designed to give good starting points for the more complicated networks. The first test was done using only the first networks, performing the first guess on m_r, m_i, b^0 and [chl], to check that this first guess was efficient (see Table 6.2 and Figures 6.13 and 6.14). This first guess was not better than the one achieved using classifiers, as the retrieved parameters were still outside of the data range. Moreover, as m_r and m_i were retrieved separately from b^0 and [chl], the corresponding TOA observations were far from the real ones.

	Relative RMS (% of target)	Bias	Std
ρ_{toa}	4.327416e+02	-1.188342e-01	1.574980e-01
ν	2.209596e-05	-9.396992e-10	2.447876e-08
τ	3.574356e-06	4.758629e-10	2.682411e-09
m_r	6.782622e+03	-1.304376e+00	7.177396e+00
m_i	4.488820e+05	1.088222e+01	5.962163e+01
b^0	2.722770e+03	2.738618e-01	2.244020e+00
[chl]	3.282592e+03	3.001337e+00	2.508779e+00

Table 6.2: Errors on retrieval using only first guess networks (MLPs).

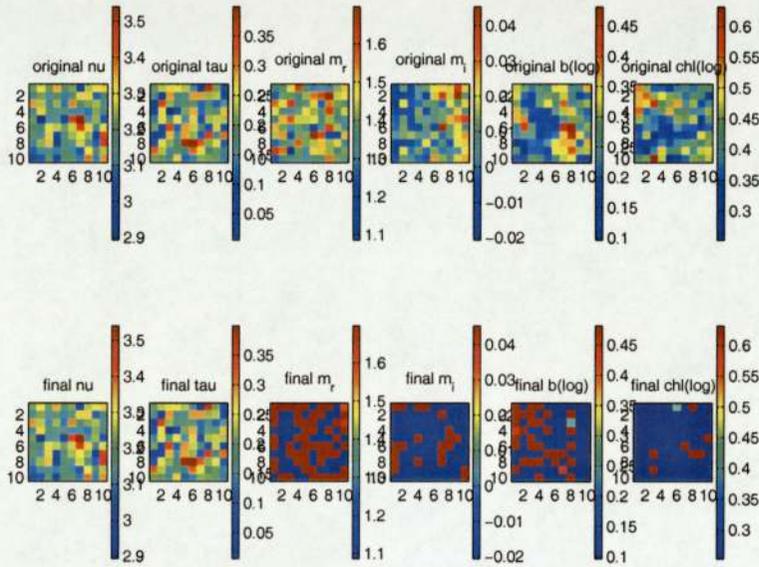


Figure 6.13: Third framework - First guess (MLPs): parameters.

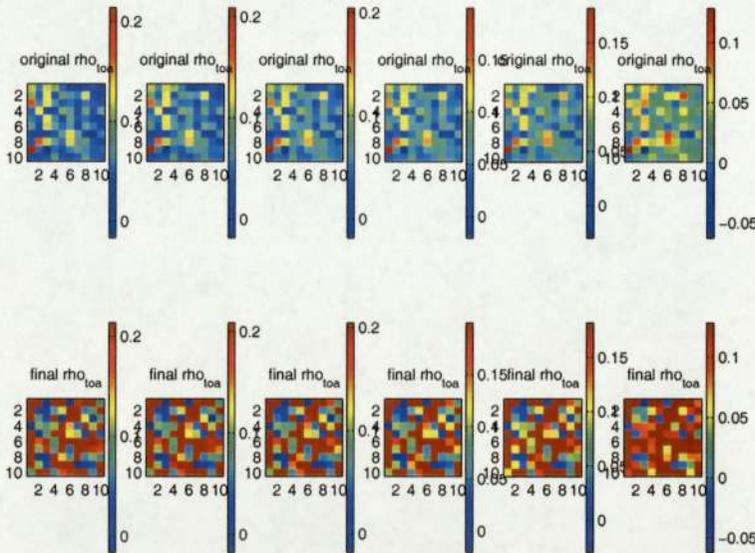


Figure 6.14: Third framework - First guess (MLPs): observations.

Then the network doing the first retrieval of all the four parameters m_r, m_i, b^0 and [chl] was tested, alone at first (Figures 6.15 and 6.16, and Table 9.3 in appendix), and then with the other two first guess networks (Figures 6.17 and 6.18 and Table 9.4 in appendix). As shown in Figures 6.15 and 6.16, the algorithm not using the first guess found a minimum which had wrong parameters even if the corresponding observations were close to the real ones. The algorithm using the first guess didn't even find a local minimum. This test showed that the first guess was wrong. (Figures 6.17 and 6.18) This test has shown that the simple forward model was useful only for the parameter m_i when it was used without the other first guess.

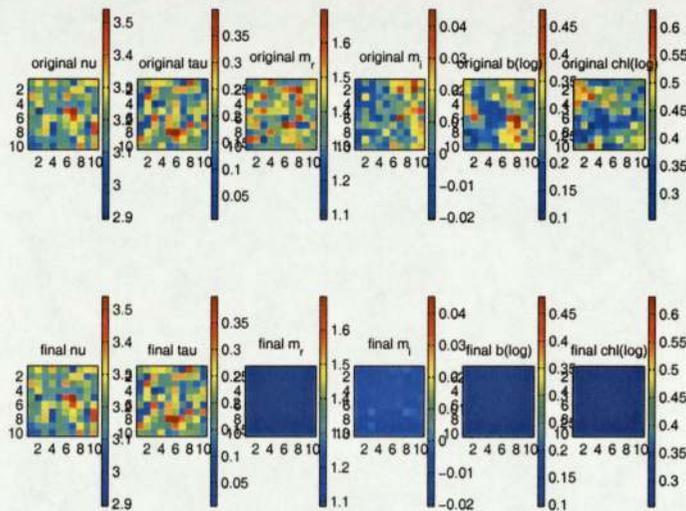


Figure 6.15: Third framework - Simple forward model (MLPs): parameters.

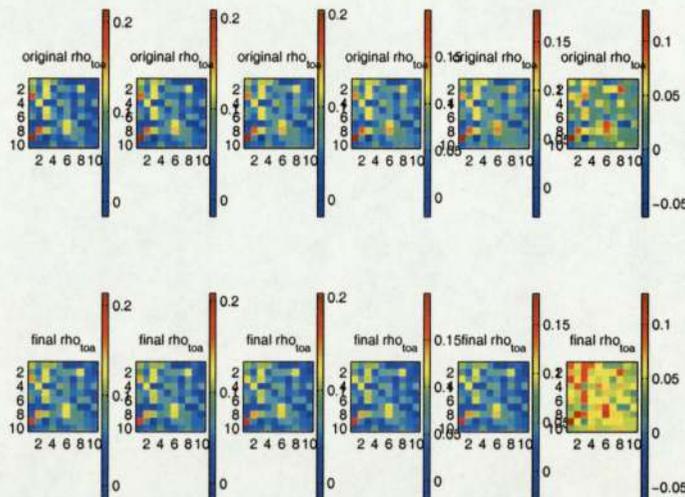


Figure 6.16: Third framework - Simple forward model (MLPs): observations.

CHAPTER 6. TESTS ON SIMULATED DATA

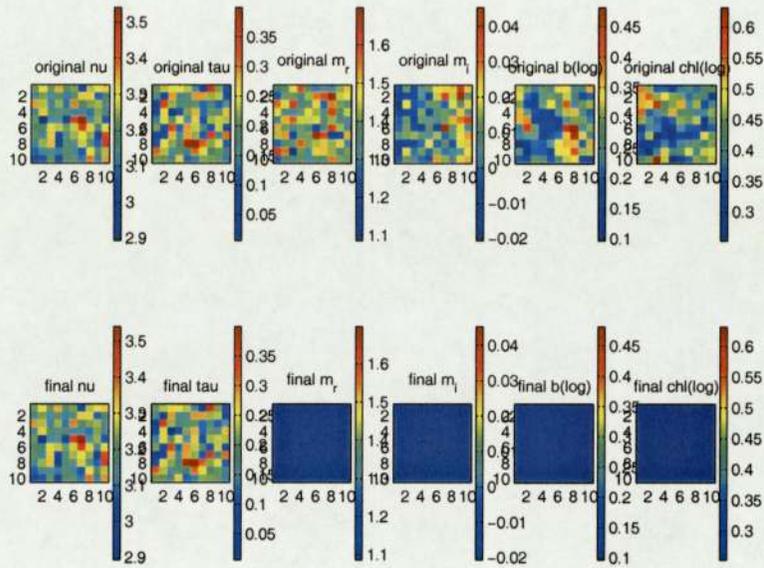


Figure 6.17: Third framework - First guess and simple forward model (MLPs): parameters.

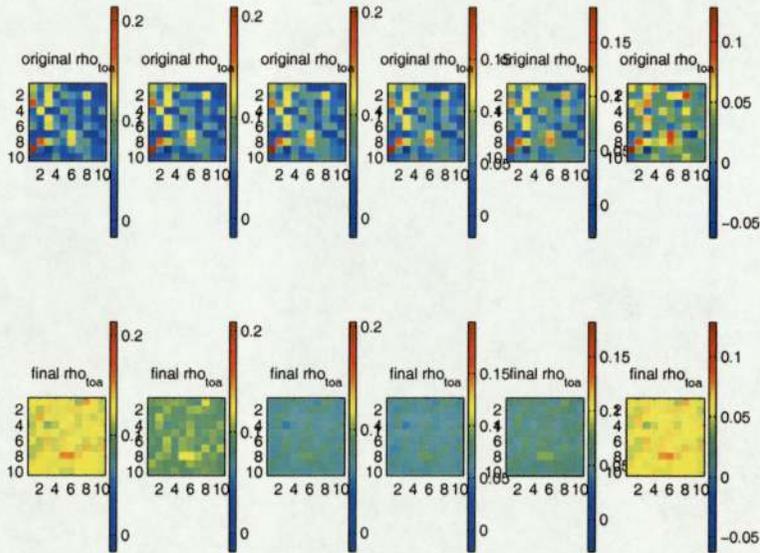


Figure 6.18: Third framework - First guess and simple forward model (MLPs): observations.

It was hypothesised that using this simple forward model for a first retrieval to fix the starting point of the complex model would be useful for m_i only. This was tested by using the simple forward model together with the complex model (see Figure 6.20 and Table 9.6 in appendix). This retrieval was compared with the results using the complex model alone (Figure 6.19 and Table 9.5 in appendix). This comparison showed that the parameter m_i was retrieved with better accuracy when the simple forward model was used. However, the oceanic parameters, b^0 and [chl] were still poorly retrieved.

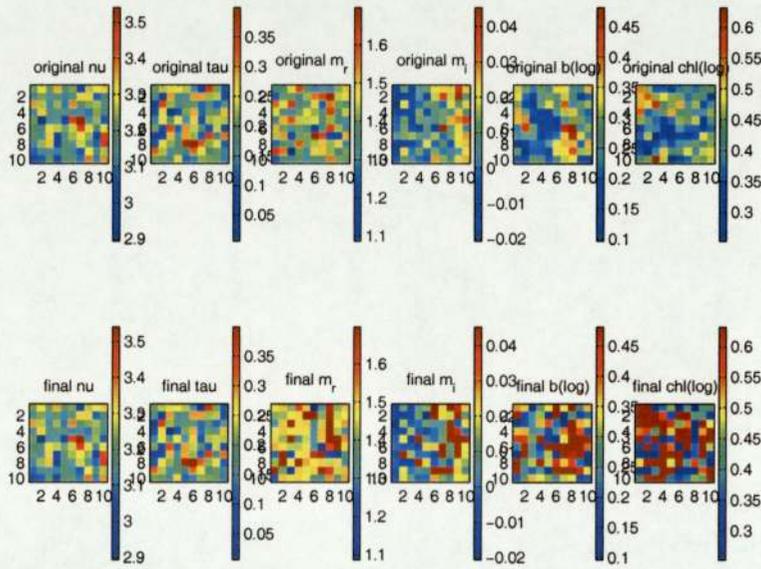


Figure 6.19: Third framework - Complex model (MLPs): parameters.

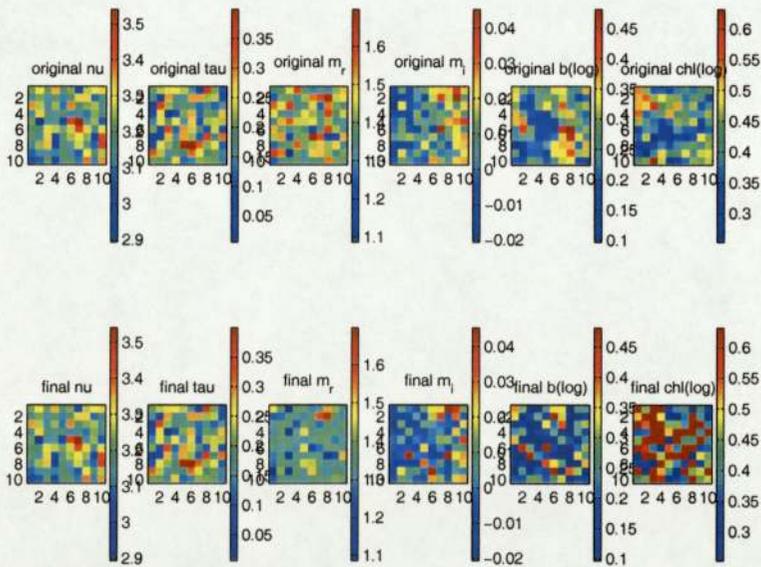


Figure 6.20: Third framework - Simple forward model and complex model (MLPs): parameters.

This framework, using the simple forward model for a first guess and the complex model to refine the retrieval was the best one using MLPs. It was then tested using RBFs instead of MLPs. Using RBFs, the retrieval of m_r and m_i was better, even if the retrieval of b^0 and [chl] were still poor (See Figures 6.21 and 6.22).

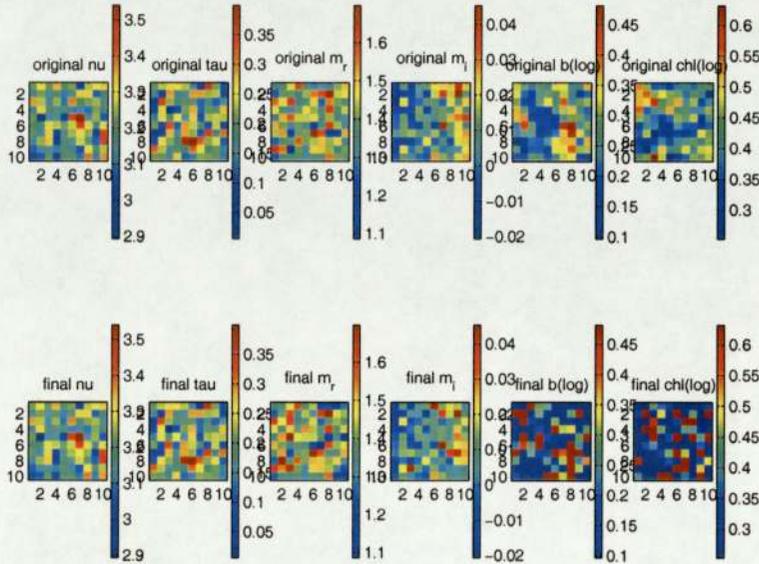


Figure 6.21: Third framework - Simple forward model and complex model (RBFs): parameters.

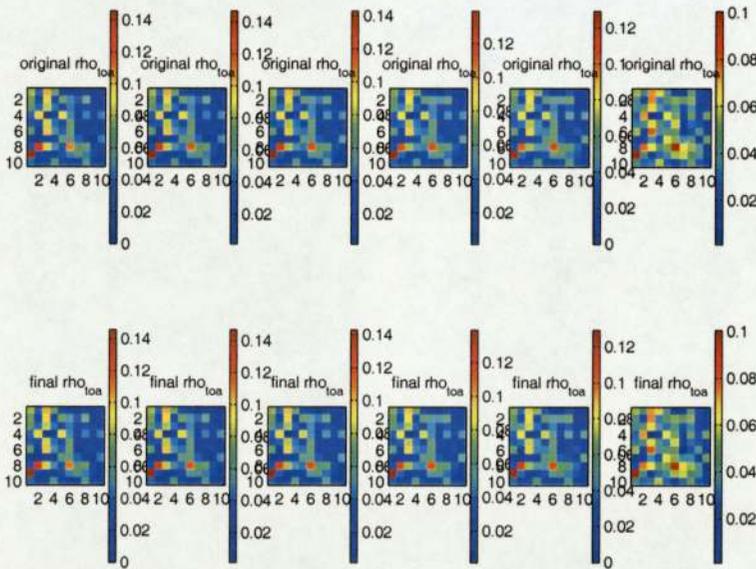


Figure 6.22: Third framework - Simple forward model and complex model (RBFs): observations.

6.5 Fourth retrieval framework

At this point the retrieval of ν, τ, m_r, m_i was done, but b^0 and [chl] were still to be retrieved. As all the atmospheric parameters were then known $\rho_{atmo}(\lambda)$ and $t(\lambda)$ could be computed, and as

$$\rho'_{toa}(\lambda) = \rho_{atmo}(\lambda) + t(\lambda) \cdot \rho_w(\lambda) \quad (6.3)$$

$\rho_w(\lambda)$ could be deduced

$$\rho_w(\lambda) = \frac{\rho'_{toa}(\lambda) - \rho_{atmo}(\lambda)}{t(\lambda)} \quad (6.4)$$

Then b^0 and [chl] were retrieved from $\rho_w(\lambda)$ as in section 5.2.

The fourth retrieval framework was then performed using (Figure 6.23)

- The NIR bands model to retrieve ν and τ ,
- The classifier for a first guess of m_r and m_i ,
- The simple forward model to do a first retrieval of m_r, m_i, b^0 and [chl],
- The complex forward model to refine the retrieval of m_r, m_i, b^0 and [chl],
- The oceanic model to refine the retrieval of b^0 and [chl].

This framework relied on $\rho_{atmo}(\lambda)$ and $t(\lambda)$, which means that errors on ν, τ, m_r, m_i were transmitted to $\rho_w(\lambda)$ and then to b^0 and [chl]. Moreover, as in 6.4, the error on $\rho_w(\lambda)$ was

$$\begin{aligned} \Delta\rho_w &= \left| \frac{\partial\rho_w}{\partial\rho'_{toa}} \right| \cdot \Delta\rho'_{toa} + \left| \frac{\partial\rho_w}{\partial t} \right| \cdot \Delta t + \left| \frac{\partial\rho_w}{\partial\rho_{atmo}} \right| \cdot \Delta\rho_{atmo} \\ &= \left| \frac{1}{t} \right| \cdot \Delta\rho'_{toa} + \left| -\frac{\rho'_{toa} - \rho_{atmo}}{t^2} \right| \cdot \Delta t + \left| -\frac{1}{t} \right| \cdot \Delta\rho_{atmo} \\ &= \left| \frac{1}{t} \right| \cdot (\Delta\rho'_{toa} + \Delta\rho_{atmo} + \rho_w \cdot \Delta t) \end{aligned} \quad (6.5)$$

The λ are not written as each variable is dependent of λ .

The error on ρ_w was then too big to enable any retrieval (See Figures 6.24, 6.25 and 6.26 and Table 9.10 in appendix). As shown in Table 6.3, the ρ_w computed by the algorithm was then completely different from the ρ_w in the dataset used to train the neural networks. The neural network was then unable to find any parameter which would lead to this ρ_w . This framework was then useless.

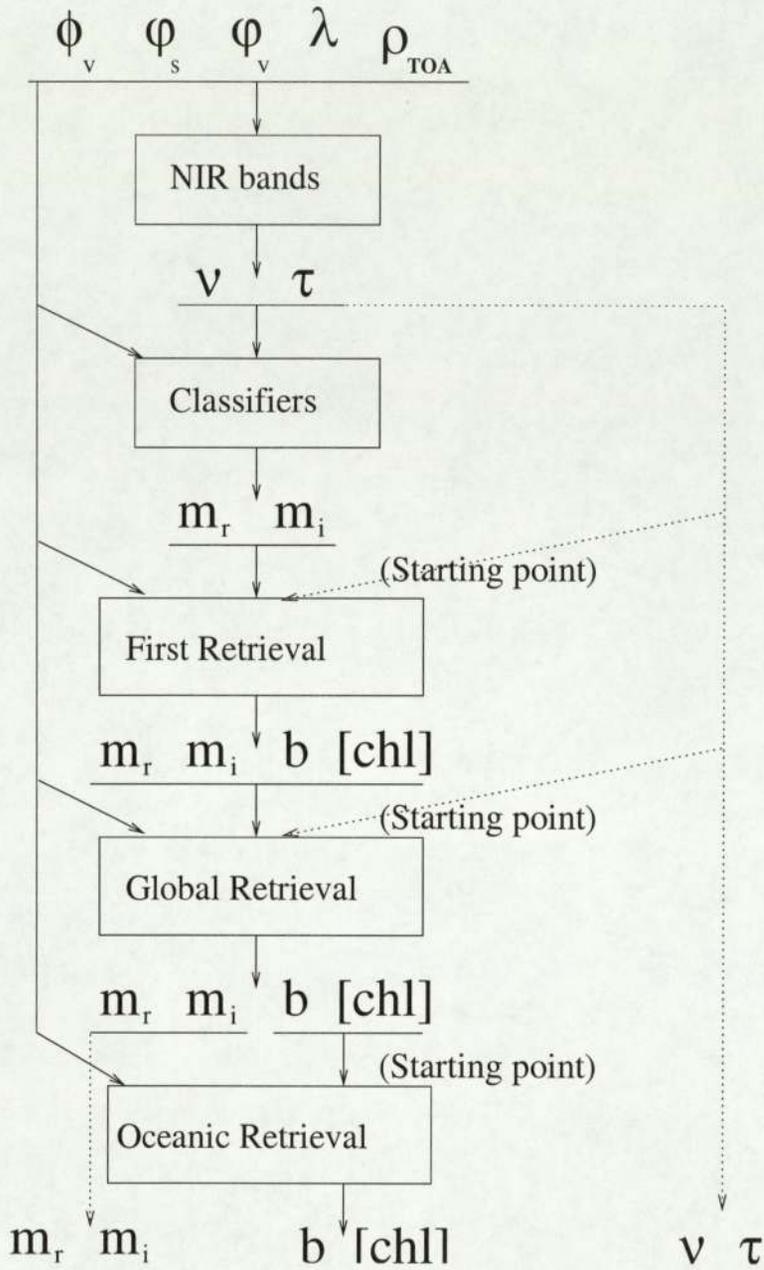


Figure 6.23: Fourth Global retrieval framework.

Dataset	Max	Mean	Min
ρ_w derived from the retrieved parameters	0.2859	0.0100	-1.2795
ρ_w in training data	0.0485	0.0122	0.0003

Table 6.3: Characteristics of ρ_w derived from the retrieved parameters and ρ_w in the training set of the neural network.

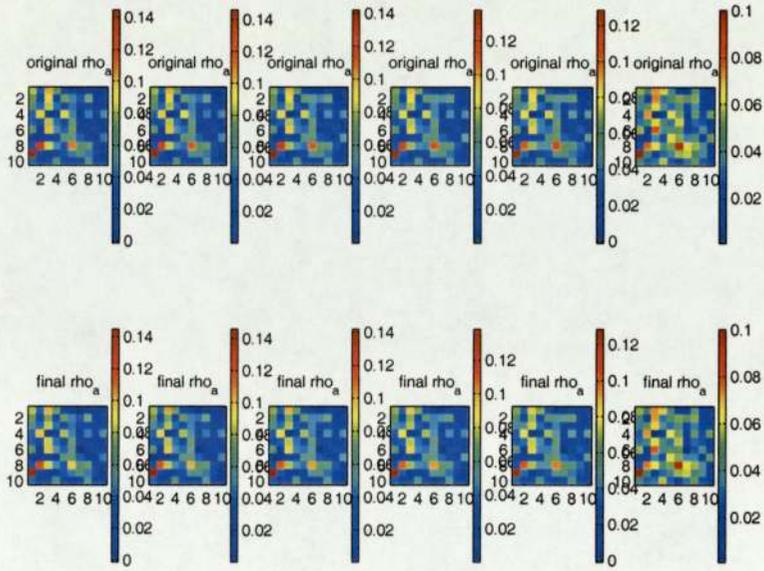


Figure 6.24: Fourth framework - Simple forward model and complex model (RBFs): ρ_a .

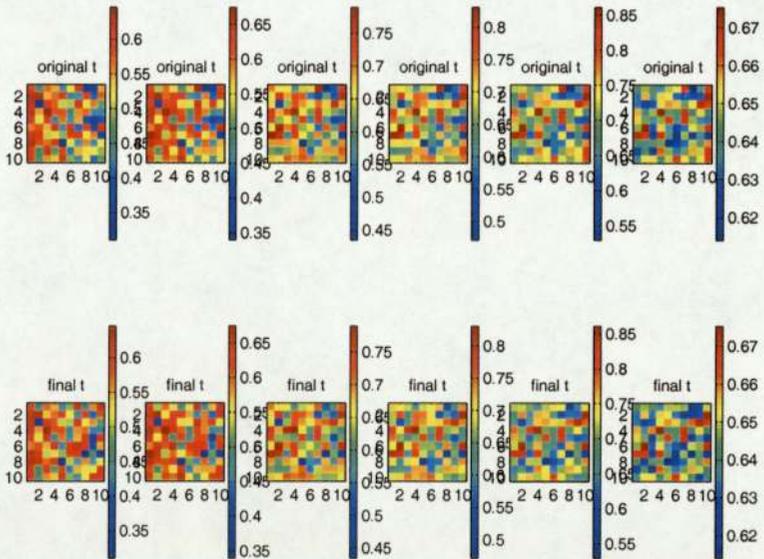


Figure 6.25: Fourth framework - Simple forward model and complex model (RBFs): t .

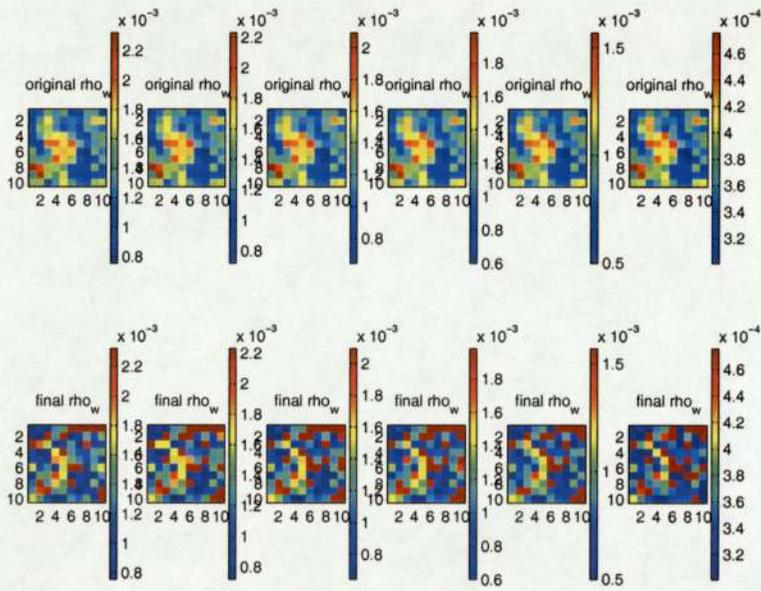


Figure 6.26: Fourth framework - Simple forward model and complex model (RBFs): ρ_w . The errors on ρ_a and t lead to a important distortion on ρ_w .

6.6 Conclusion of the tests on simulated data

The tests on simulated data have shown that the atmospheric parameters, ν , τ , m_r , m_i could be retrieved with acceptable precision, even if only the retrieval of ν and τ were accurate to the level required by SeaWiFS specifications, that is with an error less than 5% of the signal. The reason why b^0 and [chl] were not retrieved with acceptable confidence is that the architecture of the model enables a solution giving good observations (ρ_{toa}) but wrong components (ρ_a , t , ρ_w) and thus wrong parameters. Any small error on the atmospheric parameters leads to an error on the atmospheric transmittance (t), and thus on the water-leaving reflectance (ρ_w). This compromises the retrieval of the oceanic parameters. To enable a better retrieval of b^0 and [chl], a much more accurate model would be needed for the atmospheric transmittance (t) as the error on ρ_w comes mainly from the error on t .

The final and most reliable framework according to the tests done on simulated data was achieved using (See Figure 6.27):

- The NIR band model to retrieve ν and τ ,
- The classifiers to perform a first guess of m_r and m_i ,
- The simple forward model to perform a first retrieval of m_r , m_i , b^0 and [chl], using the previous m_r and m_i as starting points,
- The complex forward model to refine the retrieval of m_r , m_i , b^0 and [chl], using the previous m_r , m_i , b^0 and [chl] as starting points.

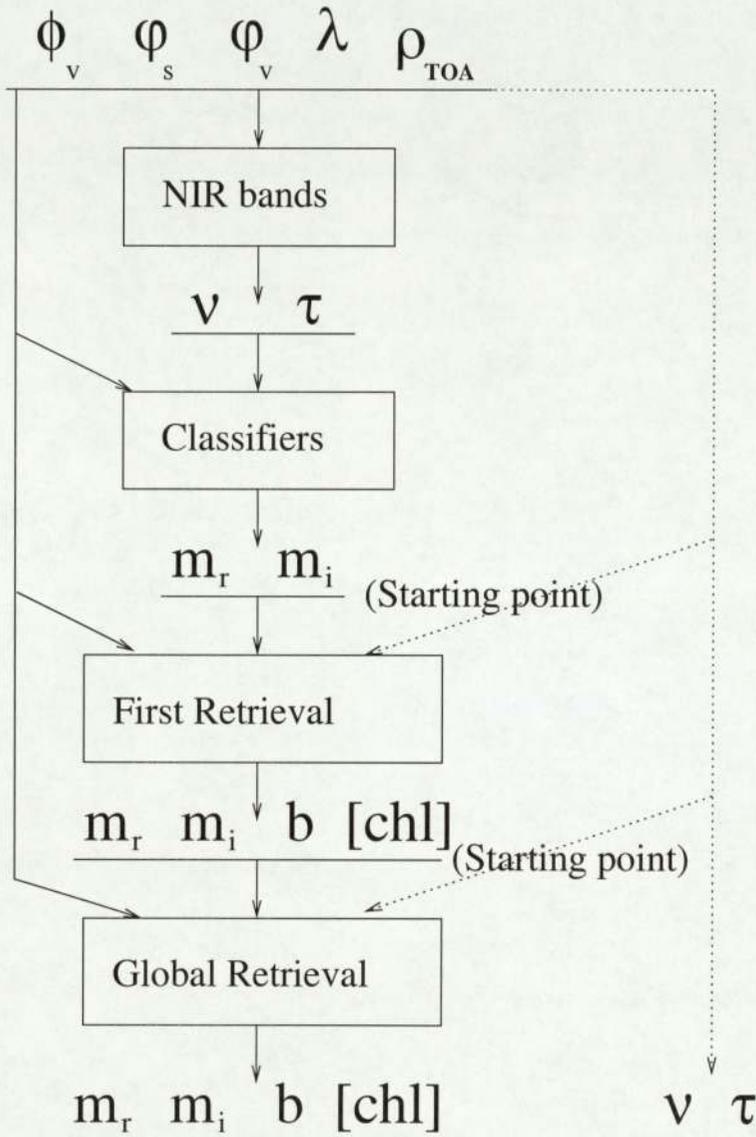


Figure 6.27: Final Global retrieval framework.

Chapter 7

Tests on real data

The tests on the global model were first done using simulated data, and then using real data instead.

7.1 Standard SeaWiFS atmospheric correction

The real data was provided by Anton Lyaskovskiy, NCRG. The raw top of atmosphere information from SeaWiFS was provided together with basic correction information for Rayleigh scattering, sun glitter, whitecaps and oxygen and ozone absorption.

SeaWiFS provides the radiances; the conversion from radiances to reflectances is:

$$\rho(\lambda) = \frac{\pi L(\lambda) \cos(\theta_0)}{F_0(\lambda)} .$$

Then, several corrections were performed in order to compute the corrected reflectances which can be handled by the trained neural network.

First, the sun glitter is removed

$$\rho_{c_1}(\lambda) = \rho_{toa}(\lambda) - T(\lambda) \cdot \rho_g(\lambda) .$$

Secondly, the Rayleigh-corrected reflectance is defined by

$$\rho_{c_2}(\lambda) = \rho_{c_1}(\lambda) - \rho_r(\lambda) - t(\lambda) \cdot \rho_{wc}(\lambda) .$$

Thirdly, it is assumed [8] that the effects of ozone and oxygen can be removed

$$\rho_{c_3}(\lambda) = \frac{\rho_{c_2}(\lambda)}{t_{O_2} \cdot t_{oz}} .$$

These corrections are adding a certain amount of noise to the data, this noise must be taken in account in the assumption made in the retrieval. However, as this noise is negligible compared to the errors of the neural networks, the assumptions will not be modified. As the standard SeaWiFS atmospheric correction were performed, the corrected TOA reflectances could be used in the Bayesian retrieval framework.

7.2 Bayesian retrieval

The main problem when testing on real data was that only the TOA reflectance and its three components, the atmospheric reflectance, the atmospheric transmittance and the water-leaving radiance were known. Thus it was difficult to estimate the errors on the retrieved parameters as the real values for most of them were unknown. However, the results found were compared with SeaWiFS results.

The first test was done using the framework designed during the tests on simulated data, using MLPs (Figures 7.1 and 7.2). The MLPs were chosen because they were more reliable than the RBFs according to the training and test errors, even if the RBFs were slightly better in the tests on simulated data.

The retrieved parameters were wrong, as well as the corresponding observations (Figures 7.2 and 7.1). This behaviour came from the use of a first guess model. These were efficient using simulated data because the data was simulated with the same neural networks as the ones used for the retrieval. But when using real data, the first guess models were more likely to find local minima as they are more simple than the complex model.

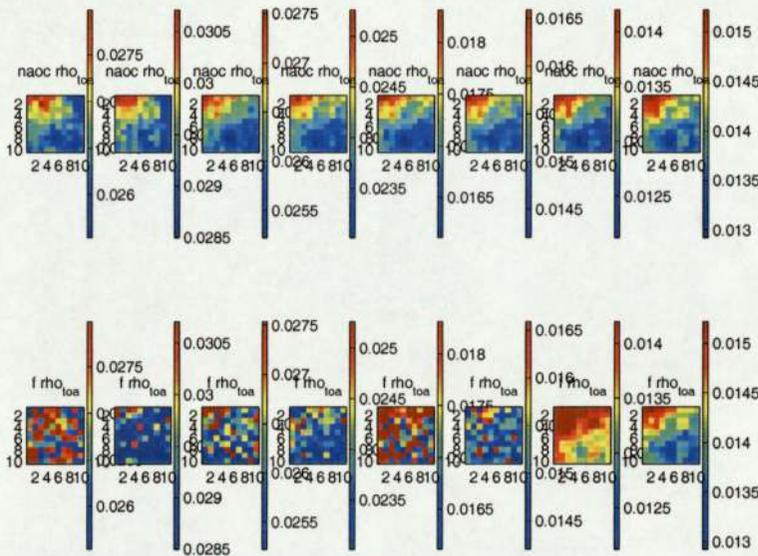


Figure 7.1: Retrieval using the simple forward model and complex model (MLPs): observations.

Then the retrieval was tested with a framework using only the NIR bands model and the complex forward model (Figure 7.4). The results were better as the observations corresponding to the retrieved parameters were closer to the real values (Figure 7.3). However, the components of ρ_{toa} : ρ_a (Figure 7.6) and ρ_w (Figure 7.5) were still far from the real ones, even if the parameters were closer to the physical range than in the previous test.

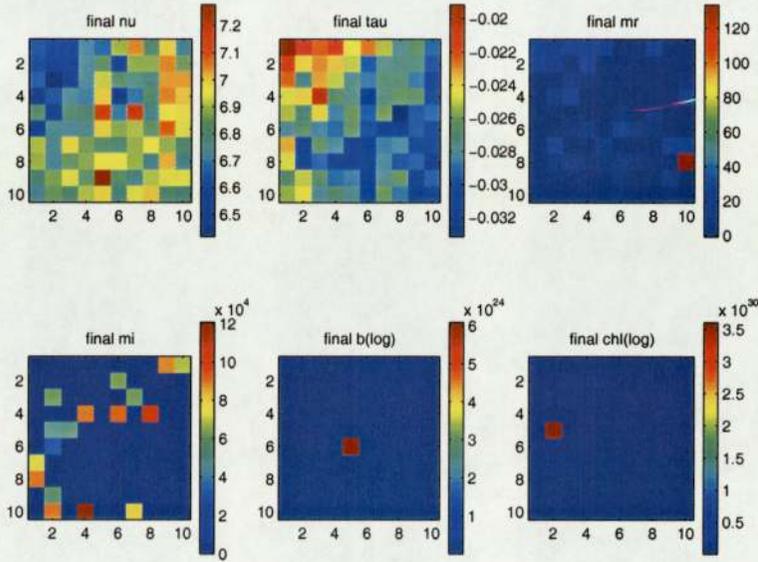


Figure 7.2: Retrieval using the simple forward model and complex model (MLPs): parameters.

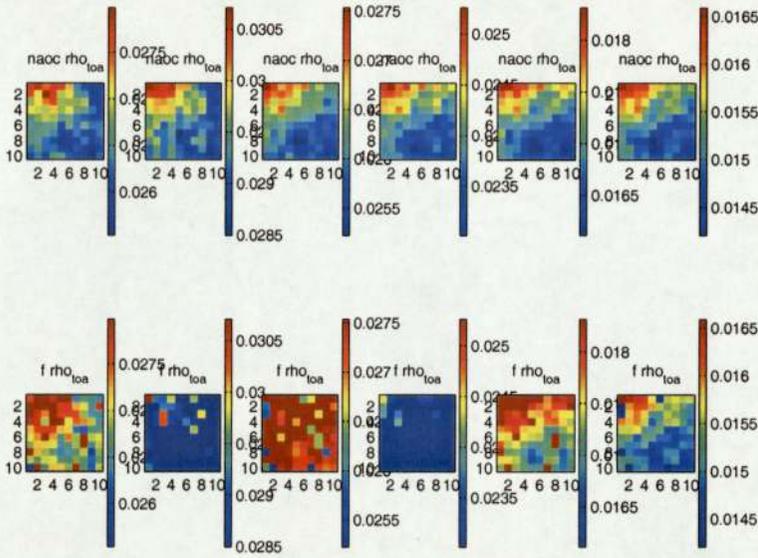


Figure 7.3: Retrieval using the complex model (MLP): observations.

7.3 Conclusion of the tests on real data

The tests on real data have shown that the framework built to be efficient on simulated data gave poor results on real data. This is mainly because the neural networks used in the retrieval have been trained on the same datasets as the ones used to simulate the data. Thus, their first guess was more accurate on simulated data than on real data, and so consequently was the global retrieval.

The reason why the algorithm is doing worse on real data than on simulated data

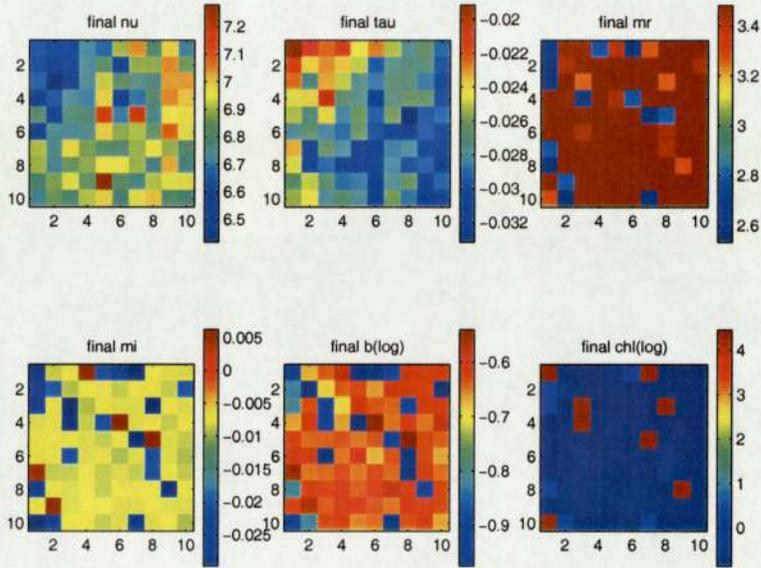


Figure 7.4: Retrieval using the complex model (MLP): parameters.

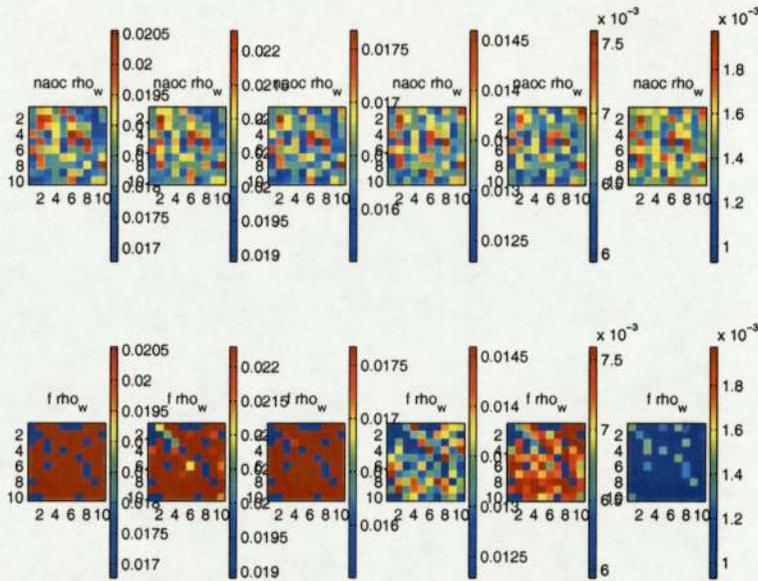


Figure 7.5: Retrieval using the complex model (MLP): ρ_w .

is that the errors induced by the physical model used to simulate the data, and thus to train the networks, is added to the error directly coming from the networks and the noise on the observations. The error at each level is then greater than when using simulated data, thus the retrieval of ν and τ is worse, then this error is transmitted through to the retrieval of the other parameters.

The use of priors enables us to direct the parameters to the physically correct values, but as no prior is given for the values of ρ_a , t and ρ_w , parameters giving a nearly true

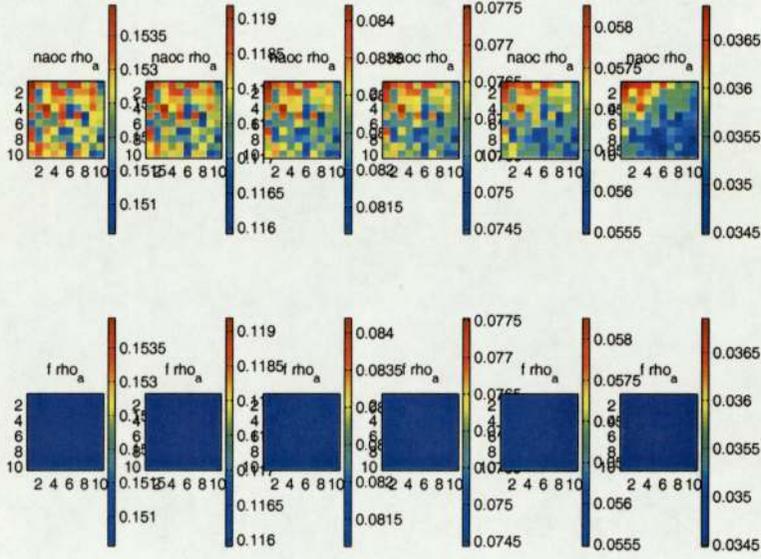


Figure 7.6: Retrieval using the complex model (MLP): ρ_a .

ρ_{toa} but wrong components ρ_a, t and ρ_w are found.

Some propositions to improve the retrieval on simulated as well as on real data are explained in the next chapter.

Chapter 8

Conclusion

First of all, the forward neural networks to learn the data simulated from the Chomko & Gordon model have been designed and trained. Additional networks have been built to perform the first guess of the different parameters.

Secondly, the Bayesian framework has been set up, implemented and tested on small problems corresponding to the retrieval of some of the parameters (ν and τ in section 5.1), and the retrieval of the parameters of a single model (oceanic model, in section 5.2). Other tests were done to demonstrate the use of priors (in section 5.2). These tests have shown that the Bayesian framework was effective on the inversion of a single neural network, and that the use of priors enabled a more accurate retrieval when using noisy observations.

Thirdly, tests on simulated and real data have been performed. The framework has been modified according to the results of the tests on simulated data, and according to the results of the tests on real data so as to improve the accuracy of the retrieval in both cases. These tests have shown that four out of the six parameters were retrieved in the case of simulated data, whereas none of the parameters was retrieved in the case of real data. The main problem observed during the tests on real data was that small errors on the atmospheric parameters led to an important distortion on the oceanic information, and thus on the oceanic parameters. Thus more accurate models would be needed to improve these results, as well as a first guess for b^0 and [chl]; however none of the attempts to build a first guess for the oceanic parameters was successful, the problem being too complex to be solved by a single neural network, and thus all the attempts to build a model to perform a first guess were leading to wrong local minima. The tests on real data have shown that the retrieval led to parameters giving TOA observations close to the real ones, but with wrong components (ρ_a, t, ρ_w). The results of the tests on simulated and real data have shown that the use of Bayesian methods is useful. However the global retrieval framework performance is still far below SeaWiFS requirements. Hence, several improvements can be suggested.

The first possible improvement comes from current retrieval algorithms which are

using lookup tables to infer aerosols properties, m_r and m_i , from ν and τ . As the goal of this thesis was to build a Bayesian approach to the problem of aerosol correction this solution wasn't used. However, this issue can be useful to give a first estimation of the parameters m_r and m_i which are still difficult to retrieve with enough precision in the presented framework. Secondly, the tests on real data have shown that the retrieved parameters were corresponding to wrong ρ_a, ρ_w and t , thus it could be useful to embed prior knowledge on these components in the Bayesian framework. Thirdly, the efficiency of priors is better when the processing is done over large datasets, however the mathematical tools used in the framework makes the use of large datasets difficult. Therefore sparse Gaussian processes might prove useful as they would enable us to process larger datasets. Finally, the model used to simulate the data which is used to train the neural networks is a simple and convenient model, however a more precise model would enable to build a framework more efficient on real data. The problem of retrieving wrong parameters leading to the good ρ_{toa} shows that the model is not directly invertible; either because the problem itself is not invertible, or because the model is too far from the real problem.

Chapter 9

Appendix

9.1 Classifiers for m_r and m_i

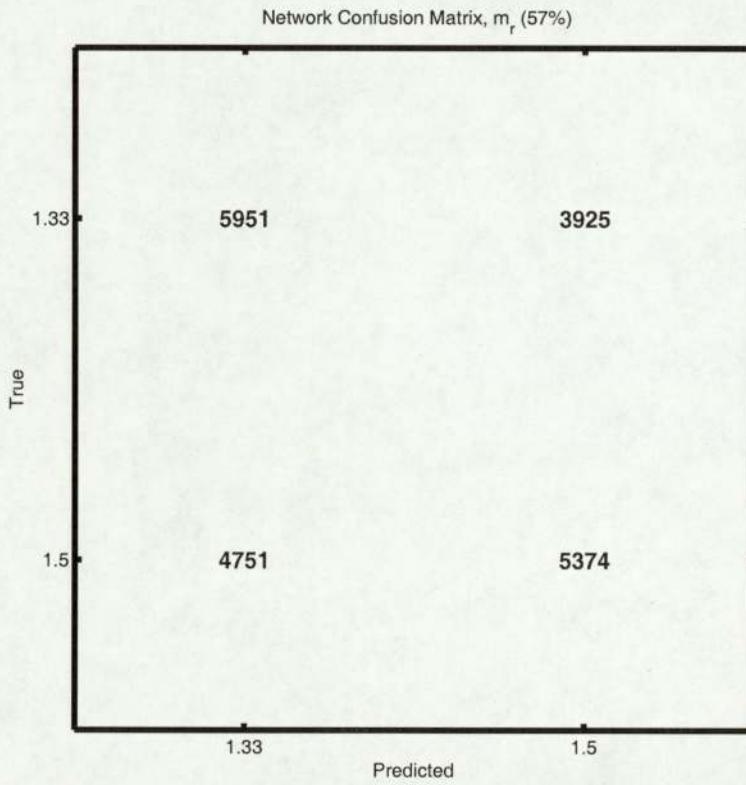


Figure 9.1: Results of the classifier MLP to retrieve m_r from ρ_{toa} .

9.1.1 Pyramidal Classifiers for m_i

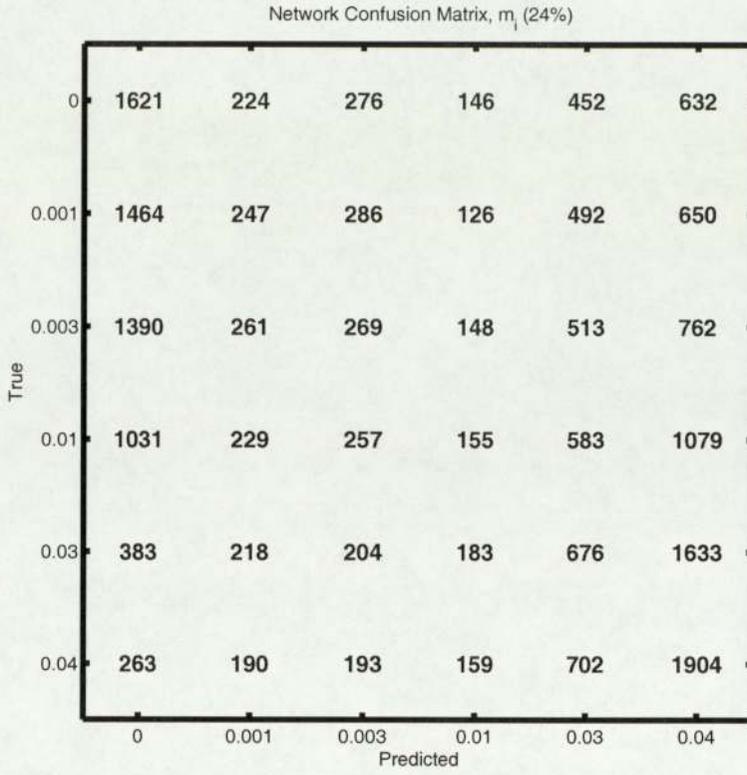


Figure 9.2: Results of the classifier MLP to retrieve m_i from ρ_{toa} .

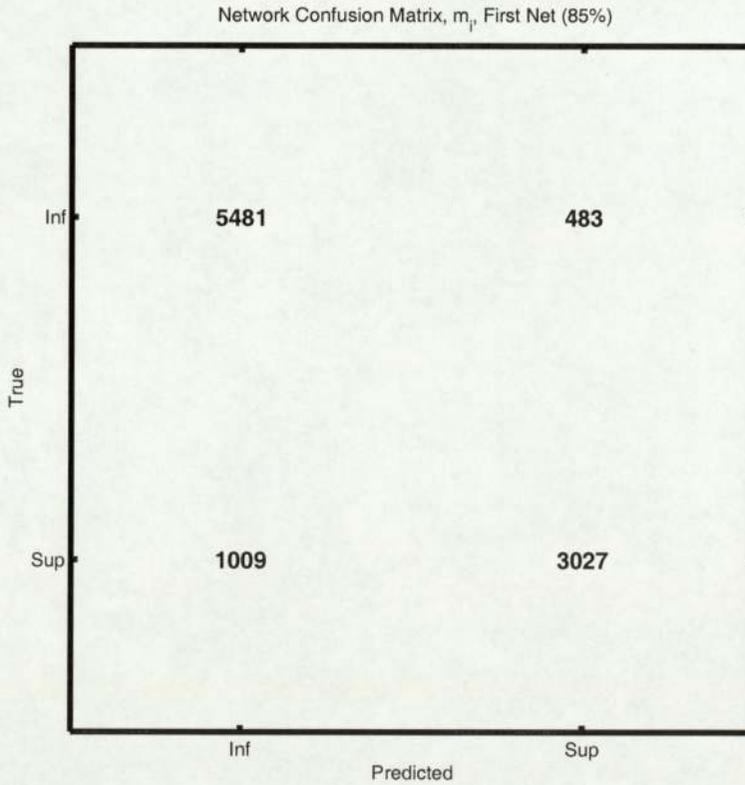


Figure 9.3: Pyramidal classifiers to retrieve m_i from ρ_{toa} - First classifier.

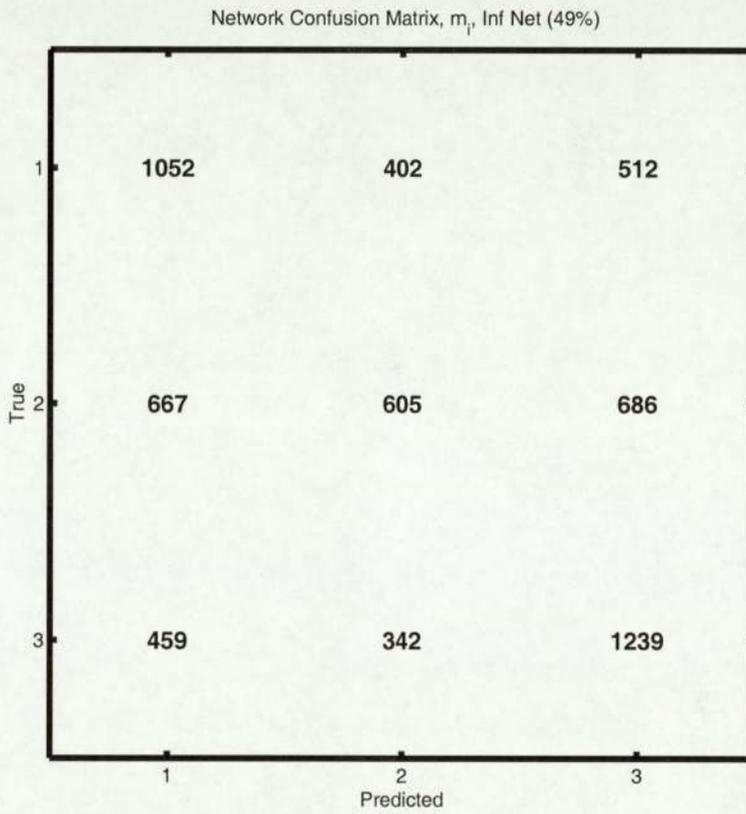


Figure 9.4: Pyramidal classifiers to retrieve m_i from ρ_{toa} - Classifier for low values.

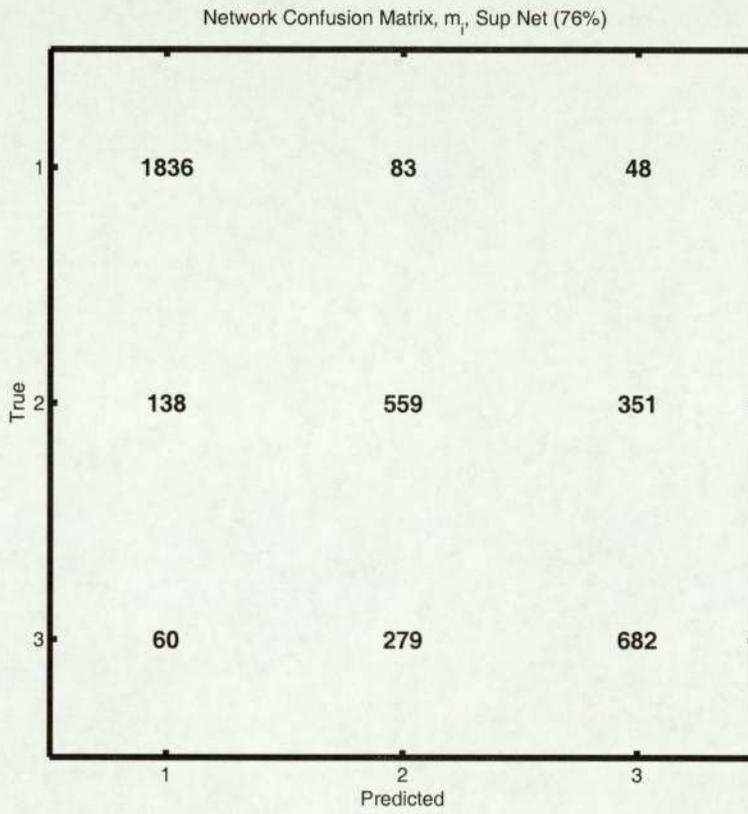


Figure 9.5: Pyramidal classifiers to retrieve m_i from ρ_{toa} - Classifier for high values.

9.2 RBFs

The three components of the global model were learnt using RBFs as an alternative to MLPs

Model	Relative RMS (% of target)	Bias	Std
Atmospheric reflectance	9.2649	1.1823e-05	0.0039
Atmospheric transmittance	29.6081	0.0025	0.0407
Oceanic reflectance	1.6533	1.3768e-05	1.8114e-04

Table 9.1: RBFs errors.

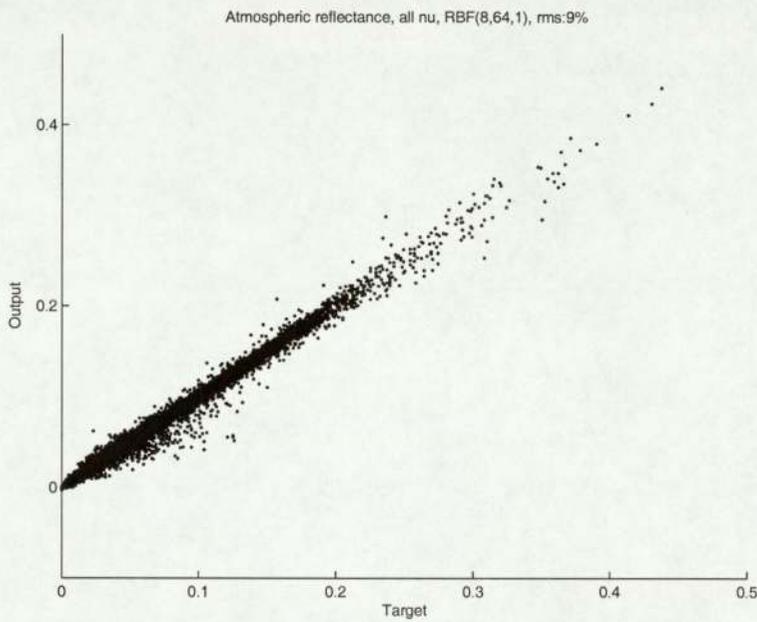


Figure 9.6: RBF for atmospheric reflectance.

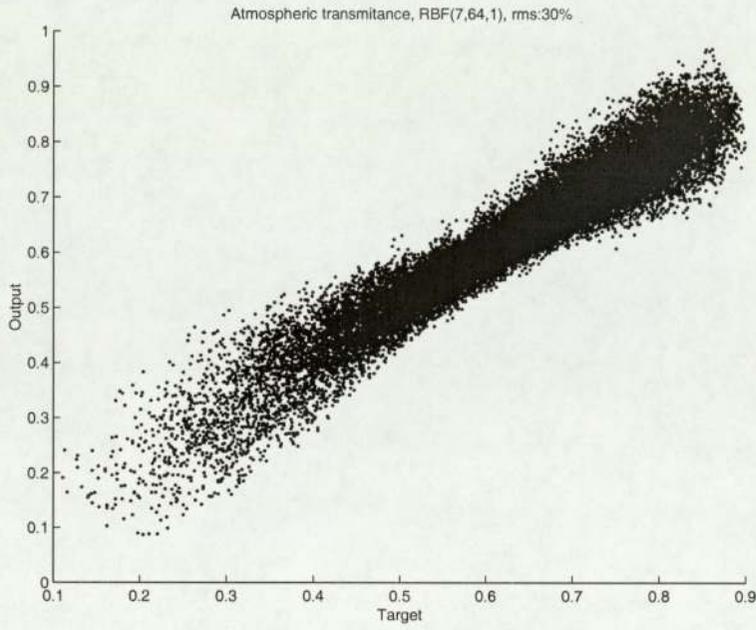


Figure 9.7: RBF for atmospheric transmittance.

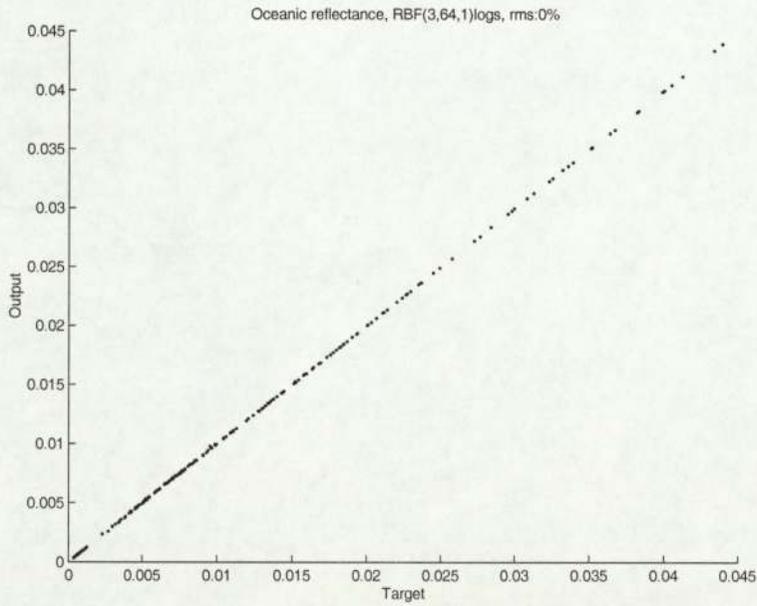


Figure 9.8: RBF for water-leaving reflectance.

9.3 Log parameter space

Model	Relative RMS (% of target)	Bias	Std
MLPs			
Atmospheric reflectance	9.3470	7.4263e-05	0.0039
Atmospheric transmittance	1.8016	-2.0401e-05	0.0025
Oceanic reflectance	0.1469	8.1074e-07	1.4997e-05
RBFs			
Atmospheric reflectance	9.6287	2.5543e-05	0.0040
Atmospheric transmittance	2.1135	-1.1196e-05	0.0029
Oceanic reflectance	1.6533	1.3768e-05	1.8114e-04

Table 9.2: Log MLPs and RBFs errors.

9.3.1 MLPs

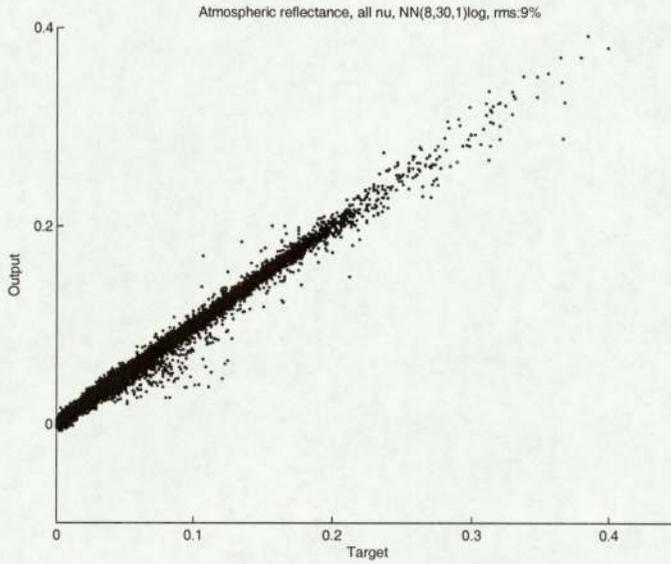


Figure 9.9: MLP for atmospheric reflectance in logspace.

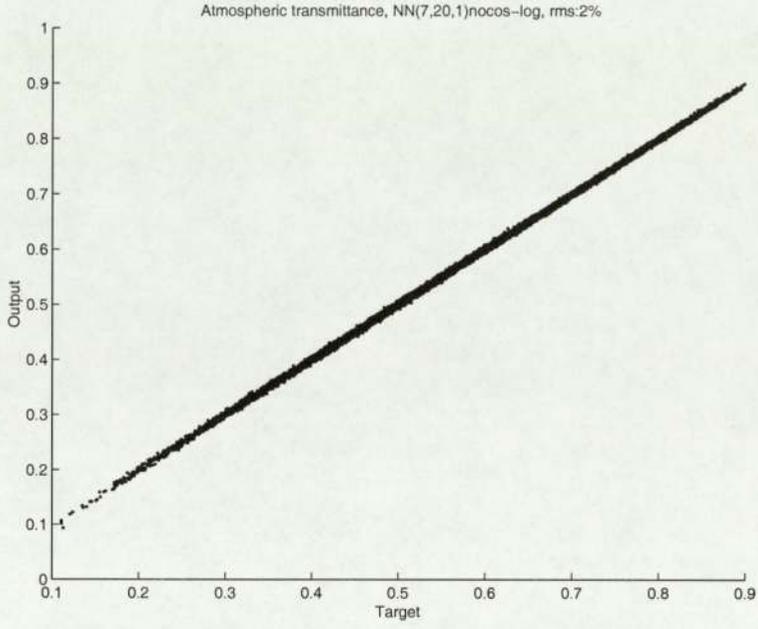


Figure 9.10: MLP for atmospheric transmittance in logspace.

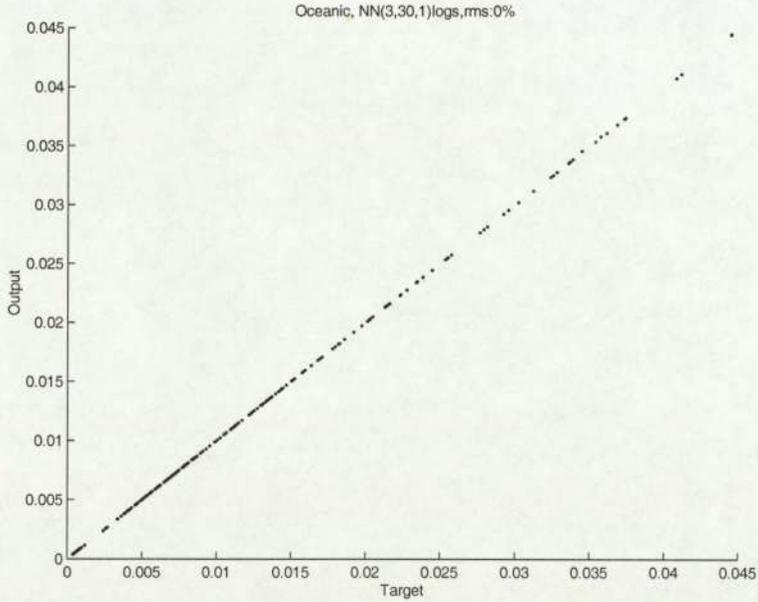


Figure 9.11: MLP for oceanic reflectance in logspace.

9.3.2 RBFs

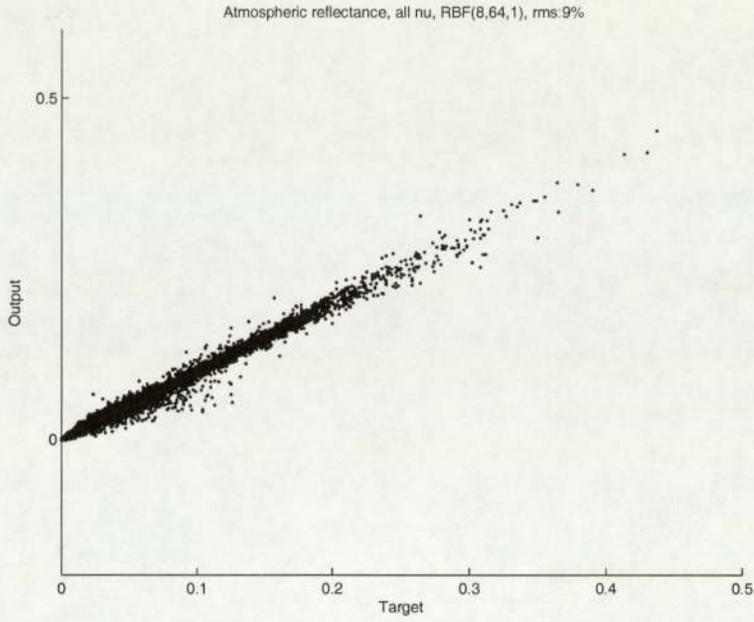


Figure 9.12: RBF for atmospheric reflectance in logspace.

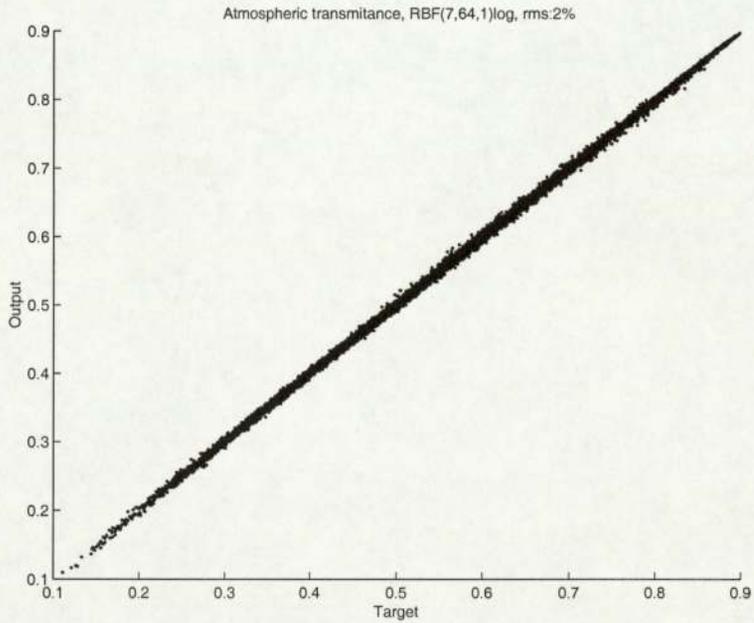


Figure 9.13: RBF for atmospheric transmittance in logspace.

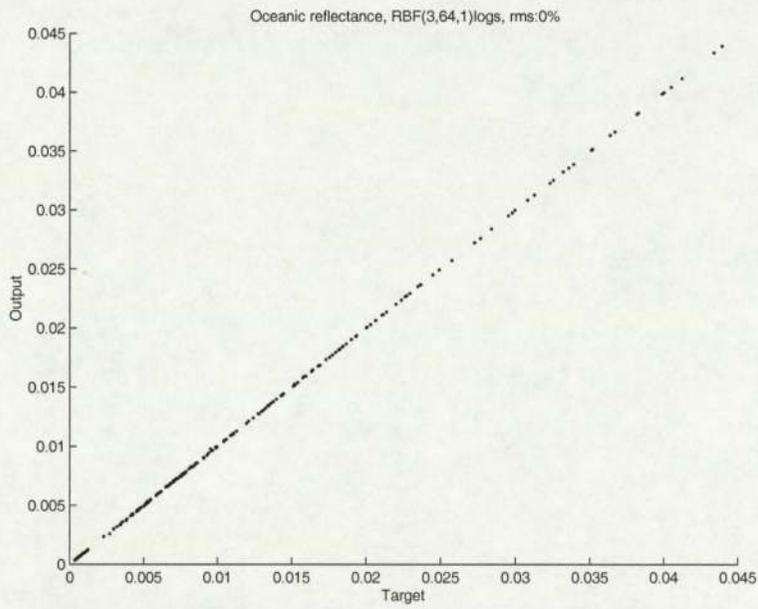


Figure 9.14: RBF for water-leaving reflectance.

9.4 First guess models

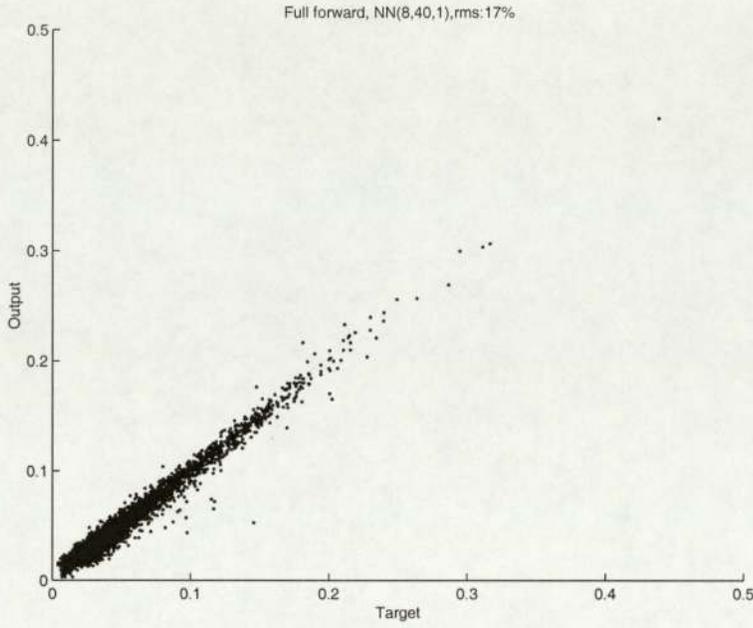


Figure 9.15: Network for first guess of m_r and m_i , inputs are $m_r, m_i, \tau, \nu, \theta_s, \theta_v, \phi, \lambda$, output is $\rho_{toa}(\lambda)$.

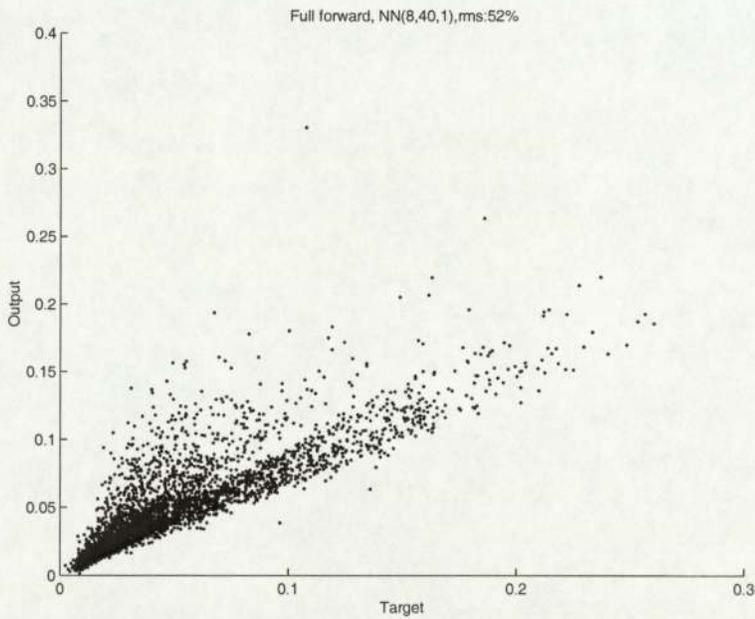


Figure 9.16: Network for first guess of b^0 and $[chl]$, inputs are $\tau, \nu, \theta_s, \theta_v, \phi, b^0, [chl], \lambda$, output is $\rho_{toa}(\lambda)$.

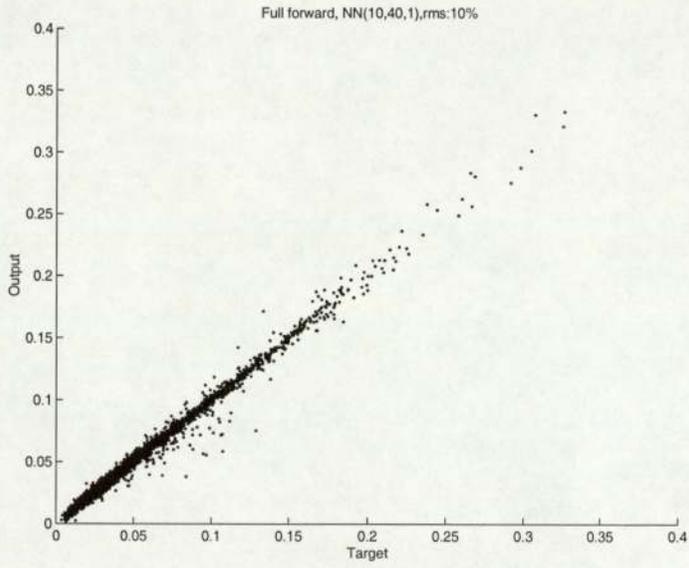


Figure 9.17: Network for first retrieval of m_r, m_i, b^0 and [chl], inputs are $m_r, m_i, \tau, \nu, \theta_s, \theta_v, \phi, b^0, [chl], \lambda$, output is $\rho_{toa}(\lambda)$.

9.5 Test on simulated data

	Relative RMS (% of target)	Bias	Std
ρ_{toa}	4.218870e+01	-2.053998e-03	1.535474e-02
ν	2.209596e-05	-9.396992e-10	2.447876e-08
τ	3.574356e-06	4.758629e-10	2.682411e-09
m_r	9.884582e+01	4.994772e-01	1.045990e-01
m_i	8.959418e+01	1.722522e-02	1.190012e-02
b^0	9.997610e+01	1.466172e+00	8.239712e-02
[chl]	1.000221e+02	1.098480e+00	7.644363e-02

Table 9.3: Errors on retrieval using only the simple forward model (MLPs).

	Relative RMS (% of target)	Bias	Std
ρ_{toa}	1.080382e+02	-2.404762e-02	3.932093e-02
ν	2.209596e-05	-9.396992e-10	2.447876e-08
τ	3.574356e-06	4.758629e-10	2.682411e-09
m_r	9.761647e+01	2.961668e+00	1.032981e-01
m_i	1.074468e+02	4.526660e+00	1.427135e-02
b^0	9.987612e+01	1.464737e+00	8.231472e-02
[chl]	9.969616e+01	1.102580e+00	7.619455e-02

Table 9.4: Errors on retrieval using first guess networks and simple forward model (MLPs).

	Relative RMS (% of target)	Bias	Std
ρ_{toa}	7.616414e+00	4.639166e-05	2.772023e-03
ν	2.209596e-05	-9.396992e-10	2.447876e-08
τ	3.574356e-06	4.758629e-10	2.682411e-09
m_r	3.740673e+02	-1.405530e-01	3.958395e-01
m_i	1.847542e+03	-7.385514e-02	2.453952e-01
b^0	1.450901e+03	-2.607126e-01	1.195786e+00
[chl]	2.467827e+03	4.484771e-01	1.886080e+00

Table 9.5: Errors on retrieval using the complex forward model (MLPs).

	Relative RMS (% of target)	Bias	Std
ρ_{toa}	8.974800e+00	2.169341e-04	3.266413e-03
ν	2.209596e-05	-9.396992e-10	2.447876e-08
τ	3.574356e-06	4.758629e-10	2.682411e-09
m_r	6.934279e+02	2.376368e-01	7.337880e-01
m_i	2.289981e+03	-3.887063e-02	3.041611e-01
b^0	4.404384e+03	5.435456e-01	3.629953e+00
[chl]	1.918609e+03	-2.406318e-01	1.466331e+00

Table 9.6: Errors on retrieval using the simple forward model and complex forward model (MLPs).

	Relative RMS (% of target)	Bias	Std
ρ_{toa}	4.090293e+00	3.752404e-05	1.012735e-03
ν	2.209596e-05	-9.396992e-10	2.447876e-08
τ	3.574356e-06	4.758629e-10	2.682411e-09
m_r	3.802717e+02	3.727865e-02	4.024050e-01
m_i	1.399237e+02	-1.621316e-03	1.858502e-02
b^0	1.834568e+03	9.411496e-01	1.511992e+00
[chl]	2.905553e+03	7.499058e-01	2.220620e+00

Table 9.7: Errors on retrieval using the simple forward model and complex forward model (RBFs).

	Relative RMS (% of target)	Bias	Std
ρ_{toa}	8.083655e+00	-2.167141e-04	5.404841e-03
ν	2.199892e-05	-3.695148e-10	5.791470e-09
τ	3.378364e-06	3.440438e-10	2.110911e-09
m_r	2.668438e+02	-3.936850e-02	2.823751e-01
m_i	9.414218e+02	-1.922374e-02	1.250420e-01
b^0	9.359567e+03	1.296091e+00	7.713857e+00
[chl]	5.974722e+03	1.253112e+00	4.566287e+00

Table 9.8: Errors on retrieval using the complex forward model in logspace (MLPs).

	Relative RMS (% of target)	Bias	Std
ρ_{toa}	4.883307e+01	-2.460933e-02	3.265045e-02
ν	2.199892e-05	-3.695148e-10	5.791470e-09
τ	3.378364e-06	3.440438e-10	2.110911e-09
m_r	1.280907e+03	-2.203115e-01	1.355461e+00
m_i	1.670517e+04	-1.015184e+00	2.218823e+00
b^0	9.327908e+03	4.517126e+00	7.687765e+00
[chl]	4.289991e+03	-1.086711e+00	3.278702e+00

Table 9.9: Errors on retrieval using the simple forward model and complex forward model in logspace (MLPs).

	Relative RMS (% of target)	Bias	Std
ρ_a	1.264569e+01	9.147767e-04	3.127103e-03
t	4.017066e+01	-7.661576e-04	3.939003e-02
ρ_w	1.189311e+03	-1.487785e-03	5.258583e-03

Table 9.10: Errors on the observation derived from the retrieved parameters in the fourth framework.

Bibliography

- [1] C. M. Bishop. Mixture Density Networks. Technical Report NCRG/94/004, 1994.
- [2] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [3] J. W. Campbell. The lognormal distribution as a model for bio-optical variability in the sea. *Journal of geophysical research*, 100(C7):13,237–13,254, July 1995.
- [4] R. M. Chomko and H. R. Gordon. Atmospheric correction of ocean color imagery: use of the junge power-law distribution with variable refractive index to handle aerosol absorption. *Applied optics*, 37(24):5560–5572, 1998.
- [5] M. Fuentes, S. C. Doney, D. M. Glover, and S. J. McCue. Spatial structure of the SeaWiFS ocean color data for the North Atlantic ocean.
- [6] H. R. Gordon. Atmospheric correction of ocean color imagery in the earth observing system era. *Journal of geophysical research*, 102(D14):17,081–17,106, July 1997.
- [7] H. R. Gordon, T. Du, and T. Zhang. Remote sensing of ocean color and aerosol properties: resolving the issue of aerosol absorption. *Applied optics*, 36(33):8670–8684, November 1997.
- [8] K. G. Ruddick, F. B. Ovidio, and M. Rijkeboer. Atmospheric correction of SeaWiFS imagery for turbid coastal and inland waters. *Applied optics*, 39(6):897–912, 2000.
- [9] A. Webb. *Statistical Pattern Recognition*. Arnold, 1999.

Index

- ϵ , 41–43
- ν , 15, 16, 27, 31, 33, 34, 38
- ν and τ retrieval, 41–44
- ν classifier, 33
- ρ_{toa} decomposition, 14, 15
- τ , 15, 16, 27, 31, 33, 34, 38
- b^0 , 15, 16, 25, 45
- m_i , 15, 16, 27, 31, 78
- m_i retrieval, 52, 79
- m_i retrieval, first classifier, 79
- m_i retrieval, high classifier, 81
- m_i retrieval, low classifier, 80
- m_r , 15, 16, 27, 31, 78
- m_r retrieval, 78
- [chl], 15, 16, 25, 45
- [chl] priors comparison, 47
- [chl] retrieval, 45, 46, 48
- τ retrieval, 39, 40

- Abstract, 2
- Atmospheric models, 27
- Atmospheric priors, 36
- Atmospheric reflectance, 27
- Atmospheric reflectance $\nu > 3$, 30
- Atmospheric reflectance $\nu \leq 3$, 29
- Atmospheric reflectance all ν , 30
- Atmospheric transmittance, 31, 32

- Bayes rule, 16, 35
- Bayesian framework, 35, 37, 38

- Chosen model, 25, 32
- Chosen models, 28
- Context, 12

- Data, 2, 24, 25, 27, 31

- Epsilon regression, 34

- Gaussian basis function, 19, 20
- Gaussian mixture model, 54
- Gaussian mixtures, 55
- Gaussian processes, 23
- Global retrieval, 53
- Global retrieval framework, 52, 59, 66, 70
- Global retrieval in two steps, 51

- Introduction, 12

- Likelihood, 18
- Logspace, 54, 84

- MDN, 20
- Mixture Density Network, 20
- MLP, 17, 53, 54
- Model parameters, 15, 16
- Model selection, 25, 28, 31
- Multi-Layer Perceptron, 17

- Neural networks, 2, 15, 17, 24
- NIR bands, 33, 34, 38–41, 43
- Normalisation, 24

- Observation models, 24
- Oceanic model, 25
- Oceanic priors, 37
- Oceanic reflectance, 26

- Priors, 2, 23, 36, 37, 55
- Pyramidal classifiers, 78

INDEX

Radial Basis Function, 19

Radiance, 13

RBF, 19, 54, 82

Reflectance, 13

Research overview, 13

Research steps, 14

Semivariograms, 23

Tests, 38, 50

The physical problem, 13

Theoretical background, 17

Thin-plate spline basis function, 19, 20