UCGE Reports
Number 20151

Department of Geomatics Engineering

Land Cover Mapping by Combinations of Multiple Artificial Neural Networks

(URL: http://www.geomatics.ucalgary.ca/GradTheses.html)

by

Donnalouise C. Watts

September 2001

THE UNIVERSITY OF CALGARY

Land Cover Mapping by Combinations of Multiple Artificial Neural Networks

by

Donnalouise Cécilia Watts

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF GEOMATICS ENGINEERING

CALGARY, ALBERTA

SEPTEMBER, 2001

Abstract

Experimenting with various artificial neural network structures and feature sets is common practice when trying to improve classification accuracies. Combining multiple networks which employ different architectures and features is a popular alternative to choosing a single "best" network for the classification problem at hand.

In this thesis, combinations of multiple networks were applied to remotely sensed data with the objective of improving canopy type classifications in subalpine ecosystems. Networks were designed, trained and tested for the following three feature sets: spectral reflectance, geomorphology and process model variables. Various combinations of the networks' outputs were then analysed using algorithms based on the following five theories: Dempster's orthogonal sum, fuzzy logic, Bayesian principles, voting techniques and a linear combination by reduction of mean squared error. The classification accuracies were increased and training times decreased for many combination methods when compared to the single "best" network.

Acknowledgments

I would first like to thank my supervisor Dr. Michael J. Collins for providing the opportunity to work in a new field of study. Without his knowledge and teachings I would not have been able to complete this work. Appreciations also go out to Dr. David Coleman of the University of New Brunswick's Geodesy and Geomatics Engineering Department for all his help and encouragement.

Without the funding provided by the Natural Sciences and Engineering Council of Canada (NSERC), I would not have been able to even consider undertaking a graduate degree. Extra funding provided by the University of Calgary's School of Graduate Studies and Geomatics Engineering Department was also gratefully appreciated.

I would like to thank Dr. Edward Johnson and Ms. Caren Dymond of the University of Calgary's Biological Sciences Department for supplying me with the data for this research. I am also very grateful to Mr. Kevin Croft of the University of Calgary's Geomatics Engineering Department for developing the artificial neural network software.

The continual support of my friends and family, especially my mother and sisters, helped me complete this dream. To all my friends in the department, thanks for the laughs, support and memories. To my former colleagues in New Brunswick, you will never be forgotten. To all the students I have helped with their Calculus, it was an experience that shall never be forgotten. Thanks for letting me learn with you.

I must mention four very important people who have taught me so much: Mr. John McCarthy and Ms. Elizabeth Walker from Miramichi Valley High School; and

Dr. Nancy Mathis and Dr. Pradeep Banerjee from the University of New Brunswick. All teachers are a very special gift, but you all stand out above the rest. Without your instruction and guidance I would have never reached this goal. Thank you for sharing your knowledge with me.

A heartfelt note of gratitude goes out the Dr. J. Douglas Williams and Ms. Sandy Harris of the Canadian Research Institute for Social Policy, University of New Brunswick. You opened your arms to me and were always there when I needed help. I will cherish the memories forever...especially France, right Sandy?? "C'est dangereux?!"

To the Kilkenny crew: Brad, Donald, France, Heidi, Jaime, Mark, Misha and Ray. I have these final words for you...Ben! Ben! Ben!!!, Non! Non! Non!!! Enjoy the wings, beer, and the waitresses for me! Will miss you all...(Notice the lack of words beginning with the letter 's'??)

For all the rides to and from school; for all the late night coffees, beer and heart-to-hearts; for putting up with my emotions and quirks; for so many unfair snowball, water and paper fights; for kicking my butt at Pinball repeatedly; for always having a shoulder to cry on when needed; and for just being such a big (but cute) goonie-head all the time...thank you Bernie.

Darren, your music plays forever in my heart. We may be an ocean apart but you are a true friend. I wish you and Belinda the best in love and life.

Karrie Shalala Woman!! What more can I say - "You Rock!!". I'll never forget Orientation Week, sambuka, Campobello, junk food, pumpkins, shopping sprees and human sexuality class!! Thanks for the memories and your friendship.

Claire-Élise, no words can express how dear you and your family are to me. Thank

you for always being there. For the tears and the laughs, through the best and the worst of times, we grew together from Barbie dolls to push-up bras, from first dates to break-ups, from Néguac to the big city...what would I have done without you?! I love you...

Anita and Brian, so many times I wanted to throw my arms up in the air and just give up. You were always there to calm me down. It's no secret that I would have never gotten this far in my studies if it weren't for your support. Anita, having you as a role-model helped me overcome many obstacles in my life. I learned to never give up when times got tough. You taught me how to hold my head high and to be proud of who I am...and I'm most proud of having you as my sister.

Dedications

This work is dedicated to my nephews Philippe, Kevin and Joey; and my nieces Ashley and Amanda. May you fulfill all your dreams and overcome any obstacles with ease. I love you all dearly and thank you for the joy you have brought into my life. Always know that I am with you even when it may seem that I am so far away.

I would also like to dedicate this work to my father, Ronald Bryan Watts. Although you were taken from me so unexpectedly, my love for you will never die.

To mi Habibi guapo...Bradley

Table of Contents

Approval Page ii

Abstract iii

Acknowledgments iv

Dedications vii

Table of Contents ix

List of Tables xii

List of Figures xiii

List of Abbreviations and Symbols xviii

1 Introduction 1

2 Theoretical Background 6
- 2.1 Introduction . . . 6
- 2.2 Artificial neural networks in remote sensing . . . 6
- 2.3 Output types and transformations . . . 11
- 2.4 Combination methods . . . 13
  - 2.4.1 Voting principles . . . 14
  - 2.4.2 Bayesian approaches . . . 17
  - 2.4.3 Dempster's orthogonal sum . . . 20
  - 2.4.4 Fuzzy logic . . . 23
  - 2.4.5 Optimal linear combination - Mean squared error reduction . . . 26
- 2.5 Accuracy assessment . . . 31

3 Study Area and Feature Sets 34
- 3.1 Introduction . . . 34
- 3.2 Study area . . . 35
- 3.3 Remotely sensed feature set . . . 37
- 3.4 Geomorphology feature set . . . 42
- 3.5 Process model feature set . . . 43

4 Experimental Design and Methodology **44**
4.1 Introduction 44
4.2 Classification by artificial neural networks 44
4.2.1 Single versus multiple feature set networks 46
4.2.2 Approach to problematic pixels 48
4.2.3 Design of network architecture 51
4.2.4 Network training 54
4.3 Combination of classification results 58
4.4 Analysing results 58
4.5 Research experiments 59
4.6 Methodology justification 60

5 Results **61**
5.1 Introduction 61
5.2 Class-conditional kappas and training 61
5.3 Confusion matrix 72
5.4 Kappa values 77
5.5 Further results 88

6 Discussion of Results **92**

7 Conclusion **97**

References **100**

A Bayesian Combination Methods **110**
A.1 Product rule 110
A.2 Summation rule 111
A.3 Maximum rule 113
A.4 Minimum rule 114
A.5 Average rule 115
A.6 Median rule 115

B Atmospheric Correction Parameters **116**

C Back-Propagation Training Algorithm with Momentum **120**

List of Tables

2.1 Confusion matrix layout 32

3.1 Characteristics of Landsat-TM's spectral bands. 39

4.1 Number of training and testing pixels per output class. 55

5.1 Number of training iterations processed per artificial neural network. 72
5.2 Confusion matrix depicting the LTM network's classification results. . 73
5.3 Confusion matrix depicting the GEO network's classification results. . 73
5.4 Confusion matrix depicting the PM network's classification results. . 74
5.5 Confusion matrix depicting the LTM/GEO network's classification results. 74
5.6 Confusion matrix depicting the LTM/PM network's classification results. 75
5.7 Confusion matrix depicting the LTM/GEO/PM network's classification results. 75
5.8 Combination method abbreviations and their definitions. 78
5.9 Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) and LTM networks; and for the voting, DOS, fuzzy logic and OLC-MSE combination methods of the three individual networks - LTM, GEO and PM. 80
5.10 Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) and LTM networks, and for the Bayesian combination methods of the three individual networks - LTM, GEO and PM. 81
5.11 Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) and LTM/GEO (LG) networks; and for the voting, DOS, fuzzy logic and OLC-MSE combination methods of following two networks - LTM/GEO (LG) and PM. 82
5.12 Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) and LTM/GEO (LG) networks, and for the Bayesian combination methods of the following two networks - LTM/GEO (LG) and PM. 83
5.13 Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) and LTM/PM (LP) networks; and for the voting, DOS, fuzzy logic and OLC-MSE combination methods of the following two networks - LTM/PM (LP) and GEO. 84

5.14 Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) and LTM/PM (LP) networks, and for the Bayesian combination methods of the following two networks - LTM/PM (LP) and GEO. 85
5.15 Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) network; the DOS combination of the LTM, GEO and PM (LTM + GEO + PM) networks; the LTM/GEO (LG) network; the DOS combination of the LTM and GEO networks (LTM + GEO); the LTM/PM (LP) network; and the DOS combination of the LTM and PM (LTM + PM) networks 89

B.1 Parameters used for determining ground visibility using the Atmospheric Correction's ATCOR0 task. 117
B.2 Parameters used for creating a reflectance image using the Atmospheric Correction's ATCOR1 task without taking into account the adjacency effect. 118
B.3 Parameters used for average filtering using the Image Processing's FAV task. 118
B.4 Parameters used for creating a reflectance image using the Atmospheric Correction's ATCOR2 task taking the adjacency effect into account. 119

List of Figures

2.1 A two-layered artificial neural network model. 8
2.2 A unit square representative of the various evidence areas supporting three outputs classes (c_1, c_2, c_3) and uncertainty (θ) for two artificial neural networks. 22

3.1 Location of study area in Alberta's Kananaskis Valley; black circles represent individual stands, source: Dymond (1998), no scale provided. 35
3.2 Landsat-TM image of study area using the blue, green and red electro-optical channels. 38
3.3 Possible solar radiance paths from the sun, atmosphere, Earth and background to the sensor. 40
3.4 Reflectance overlap of vegetative (aspen (AS), pine (PI), pine/spruce (PS) and spruce/fir (SF)) and non-vegetative (alpine (AL), deep water (DP), rock (RK) and shallow water (SH)) classes, source: Collins *et al.* (2001). 41

5.1 Class-conditional kappa (kappa) and mean squared error (MSE) values for the LTM network after every 25 iterations. 63
5.2 Class-conditional kappa (kappa) and mean squared error (MSE) values for the GEO network after every 25 iterations. 64
5.3 Class-conditional kappa (kappa) and mean squared error (MSE) values for the PM network after every 25 iterations. 65
5.4 Class-conditional kappa (kappa) and mean squared error (MSE) values for the LTM/GEO network after every 25 iterations. 66
5.5 Class-conditional kappa (kappa) and mean squared error (MSE) values for the LTM/PM network after every 25 iterations. 67
5.6 Class-conditional kappa (kappa) and mean squared error (MSE) values for the LTM/GEO/PM network after every 25 iterations. 68

List of Symbols and Abbreviations

NN_i	Artificial neural network i
i	Artificial neural network index
X_i	Input vector/feature set for network i, $X_i = [x_1, x_2, ..., x_{t_0}]^T$
x_I	Input I of feature set X_i
t_0	Number of feature set inputs
$w_{IJ}^{(l)}$	Layer l weight associated with input I and node J
l	Layer index
I	Input index
J	Node index
$x_{0J}^{(l)}$	Bias input corresponding to node J in layer l
$w_{0J}^{(l)}$	Bias weight corresponding to node J in layer l
$\Sigma_J^{(l)}$	Node J in layer l
t_l	Number of nodes in hidden layer l
$s_J^{(l)}$	Weighted sum outputted by layer l's J^{th} node
$f()$	Transfer function
a_J^l	Node J's input to layer $l+1$
m	Number of output nodes/classes
c_j	Output class j
j	Output class index
$y_i(c_j)$	A posteriori probability of output class j computed by network i
Y_i	Vector of a posteriori probabilities computed by network i, $Y_i = [y_i(c_1), y_i(c_2), ..., y_i(c_m)]^T$
I	Index
κ_j	Class j conditional kappa value
$e(c_j)$	Number of winning outcomes for class j
n	Number of artificial neural networks
a	WMV weight variable
$B_i(c_j)$	Network i's Borda count for output class j
$P(c_j \| X_1, ..., X_n)$	A posteriori class j probability conditioned on the feature sets X_i for $i = 1, ..., n$
$p(X_1, ..., X_n \| c_j)$	Joint probability distribution of feature sets X_i for $i = 1, ..., n$ conditioned on output class j
$p(c_j)$	A priori class j probability
$p(X_1, ..., X_n)$	Unconditional joint probability density of feature sets X_i for $i = 1, ..., n$
C	Set of output classes, $C = [c_1, c_2, ..., c_m]$

θ_i	Network i's a posteriori uncertainty class probability
θ	Uncertainty class
$M_i(c_j)$	Evidential mass for class j computed by network i
$M_i(\theta$	Evidential mass for the uncertainty class computed by network i
$M(c_j)$	Orthogonal sum of evidential masses for class j
$M(\theta)$	Orthogonal sum of evidential masses for the uncertainty class
$g(A)$	Fuzzy measure for set A
$g_\lambda(A)$	g_λ-fuzzy measure for set A
λ	g_λ-fuzzy measure multiplier
g^i	Network i's fuzzy measure
$g_j(A_i)$	Class j's fuzzy measure for set A_i
A_i	Set of first i ordered networks for a specific output class
Λ	OLC-MSE combination weight vector, $\Lambda = (\alpha_1, \alpha_2, ..., \alpha_n)^T$
ρ_ι	Image pixel ι
ι	Image pixel index
$X_i(\rho_\iota)$	Pixel ι's input vector/feature set for network i
$v_i(\rho_\iota)$	Network i's scalar class output for pixel ι
$\tau(\rho_\iota)$	Pixel ι's true class
$\delta_i(\rho_\iota)$	Network i's scalar approximation error for pixel ι
$v(\rho_\iota, \Lambda)$	Pixel ι's linearly combined scalar class output for n networks
$\overline{v}(\rho_\iota)$	Pixel ι's scalar class output vector for n networks $\overline{v}(\rho_\iota) = (v_1(\rho_\iota), v_2(\rho_\iota), ..., v_n(\rho_\iota))^T$
α_i	Combination weight value for network i
$\delta(\rho_\iota; \Lambda)$	n combined networks' scalar approximation error pixel ι
E	Mean or expected value
Θ	Simplified notation for the vector of $E(\tau(\rho_\iota)v_i(\rho_\iota))$ values, for $i = 1, ..., n$ networks
Ψ	Simplified notation for the matrix of $E(v_i(\rho_\iota)v_I(\rho_\iota))$ values, for $i = 1, ..., n$ and $I = 1, ..., n$ networks
$\overline{\delta}(\rho_\iota)$	Vector of approximation errors for n networks regarding pixel ι
Ω	Simplified notation for the matrix of $E(\delta_i(\rho_\iota)\delta_I(\rho_\iota))$, for $i = 1, ..., n$ and $I = 1, ..., n$ networks
Ł	Lagrangian function applied in OLC-MSE
η	Lagrangian multiplier
O	Set of input vectors/feature sets
$\hat{\Theta}$	Approximation to Θ
$\hat{\Psi}$	Approximation to Ψ
$\hat{\Omega}$	Approximation to Ω
d_{ji}	Deviation of class j's a posteriori probability from its a priori probability for network i

k	Training iteration counter
$\Delta w_{IJ}^{(l)}$	Weight change accumulator
$Y_i(\rho_\iota)$	Network i's output vector for pixel ι
$e_j(\rho_\iota)$	Pixel ι's class j classification error
$e(\rho_\iota)$	Pixel ι's classification error vector, $e(\rho_\iota) = [e_1(\rho_\iota), ..., e_m(\rho_\iota)]^T$
cm_{jk}	Confusion matrix variable denoting the number of class k pixels categorised to class j
sum_{j+}	Confusion matrix variable denoting the number of pixels categorised to class j
sum_{+j}	Confusion matrix variable denoting the number of pixels belonging to class j
sum_{++}	Confusion matrix variable denoting the number of pixels being classified
K	An artificial neural network's overall kappa value
Z	Z-score test statistic
$d_I^{(l)}$	Back-propagating variable
f'	Derivative of the transfer function
L	Output layer index of an artificial neural network
AELEV	Ground elevation
AFILE	Atmospheric correction file
AFLSZ	Adjacency effect filter size
AL	Alpine output class
ANN	Artificial neural network
AS	Aspen output class
ASAS	Advanced Solid-State Array Spectroradiometer
ASP	Computation of a pixel's aspect
ATCOR0	Determination of ground visibility package
ATCOR1	Creation of reflectance image without adjacency effect package
ATCOR2	Creation of reflectance image with adjacency effect package
AVE	Bayesian average combination
AVHRR	Advanced Very High Resolution Radiometer
BP	Back-propagation learning/training algorithm
BPNN	Back-propagation artificial neural network
CALIB1	Calibration coefficient for channel 1
CALIB2	Calibration coefficient for channel 2
CALIB3	Calibration coefficient for channel 3
CALIB4	Calibration coefficient for channel 4
CALIB5	Calibration coefficient for channel 5
CALIB6	Calibration coefficient for channel 7

CNN	Consensual artificial neural network
DATE	Date of scene
DBFC	Database filtered channel
DBIC	Database input channel
DBOC	Database output channel
DBIW	Database input window
DEM	Digital elevation model
DN	Digital number
DOS	Dempster's orthogonal sum combination
DP	Deep water output class
FAV	Average filtering
FILE	Input file to be atmospherically corrected
FLSZ	Filter size
FUZ	Fuzzy integral combination
GEO	Geomorphology artificial neural network
k-NN	k nearest neighbour classifier
Landsat-MSS	Landsat Multispectral Scanner
Landsat-TM	Landsat Thematic Mapper
LG	Landsat Thematic Mapper and geomorphology artificial neural network
LGP	Landsat Thematic Mapper, geomorphology and process model artificial neural network
LP	Landsat Thematic Mapper and process model artificial neural network
LTM	Landsat Thematic Mapper artificial neural network
Mal	OLC-MSE combination using weights equalling testing overall kappa values
MAX	Bayesian maximum combination
Mcon	OLC-MSE combination using constrained weights
MDC	Minimum distance classifier
MED	Bayesian median combination
MIN	Bayesian minimum combination
MLC	Maximum-likelihood classifier
MSE	Mean squared error
Mun	OLC-MSE combination using unconstrained weights
OLC-MSE	Optimal linear combination by reduction of mean squared error
PI	Pine output class
PM	Process model artificial neural network
PMV	Parliamentary/Plurality majority vote

PNN	Probabilistic artificial neural network
PROD	Bayesian product combination
PS	Pine/spruce output class
PVMS	Predictive vegetation mapping systems
REFLECT	Surface reflectance
REPORT	Location to which atmospheric correction report is saved
RK	Rock output class
RS	Remote Sensing
SAR	Synthetic aperture radar
SBAND	Satellite channels
SENSOR	Spacecraft's data to be atmospherically corrected
SF	Spruce/fir output class
SH	Shallow water output class
SLP	Computation of a pixel's slope
SMV	Simple majority vote
SNN	Structured artificial neural network
SONN	Self-organising artificial neural network
SPOT	Système pour l'observation de la Terre
SUM	Bayesian summation combination
TFLSZ	Target area filter size
VISIBIL	Ground visibility
WMV	Weighted majority vote
ZNANGLE	Solar zenith angle

Chapter 1

Introduction

It is no secret that satellite remote sensing (RS) has become a valuable tool for gathering information about our planet. RS is the field of study which derives information regarding the Earth's land and water surfaces using images acquired from an overhead perspective, using electromagnetic radiation from one or more areas of the electromagnetic spectrum, emitted and/or reflected from the Earth's surface (Campbell, 1996). RS identifies land surface features using measured spectral differences. By employing spectral signatures which characterise the spectral response of a feature over an observed range of wavelengths, land surface features are identifiable (Parker & Wolff, 1965). Remotely sensed data is a an excellent basis for mapping land cover and land use because of the "high correlation between variation in remotely sensed data and variation across the Earth's surface" (Congalton & Green, 1999). As well as monitoring physical changes in the environment, RS has been widely applied in many other areas: military operations, ocean mapping, agriculture, etc...

Supervised classification is one of the most extensively applied procedures in the analysis of remotely sensed data. Pixels of unknown classes are allocated to a predefined class on the basis of their spectral reflectance (Foody, McCulloch, & Yates, 1995). For example, a supervised classification of deep water pixels would require a network to be trained with representative pixels having reflectance values of 4, 5, 3, 1 and 1 percent for Landsat Thematic Mapper's (Landsat-TM) blue, green, red, near infrared and middle infrared channels. After the network is trained, it can then classify pixels not employed during training as deep water pixels or not depending on their reflectance values. Bayesian classifiers are commonly used for such classifications but recently, there has been an increase in the usage of artificial neural networks (ANNs) for RS classification due to their many advantages such as no statistical assumptions are made. An ANN is a system of interconnected processing nodes modelled after the human brain. In this research, supervised classifications were performed on remotely sensed, geomorphology and process model feature sets. ANNs were designed, trained and tested for each of these feature sets. The various network results were then combined using algorithms based on the following five theories: Dempster's orthogonal sum, fuzzy logic, mean squared error (MSE) reduction, voting techniques and Bayesian principles. These final combined results are the desired classification results analysed in this research.

Why the use of ANNs to classify remotely sensed data? Why not use conventional statistical classifiers? Why the combination of multiple networks? Why not use one large network that classifies all feature sets at once? Why the addition of differently scaled/distributed auxiliary feature sets? Why not use additional spectral reflectance feature sets? The answers to these questions provide the basis for and purpose of this research.

ANNs were chosen to perform the classifications of this research's feature sets because of their many advantages. They do not make statistical assumptions regarding the distribution of the feature set to be classified - it is not necessary to know whether a feature set is of a Gaussian or Chi square distribution in order for it to be classified by an ANN. A maximum-likelihood classifier (MLC), however, assumes that each output class's data displays a Gaussian distribution which is "often untenable with remotely sensed data where classes may display a range of distributions" (Foody et al., 1995). Not only can they classify data having an unknown distribution, ANNs have the ability to integrate multiple sources of data having different or unknown distributions. They are tolerant of outliers within the data to be classified, their testing phases are quite rapid once training is completed, they can describe patterns separable by disconnected or non-convex regions and past studies have demonstrated their ability to accurately train on small training sets (Ardö, Pilesjö, & Skidmore, 1997; Foody & Arora, 1997; Hepner, Logan, Ritter, & Bryant, 1990; Paola & Schowengerdt, 1995).

As explained in §3, the feature sets involved in this research are being partitioned primarily because they provide fundamentally different types of information regarding the Earth's surface. This partitioning is achieved by combining the results of multiple ANNs, each classifying one of the above mentioned feature sets or a subset thereof. These results are then compared to the alternative method of classifying all feature sets at once using a large network. The following outlines the many reasons for wanting to partition feature sets through combinations of multiple ANNs.

- In order to achieve the best possible classification accuracy for the problem at hand, it is common practice to investigate the construction of several ANNs each employing different parameters. This entails experimenting with various architecture sizes, connection schemes and feature sets. The results from these various networks are then compared and it is often of habit to choose the "best" network for the classification - perhaps the network with the best classification accuracy - while all other results are disregarded (Kittler, Hatef, Duin, & Matas, 1998). One major problem with this method is the risk of losing valid information contained within the other networks' results (Tumer & Ghosh, 1996).

- Larger networks tend to be more time-consuming and difficult to train than smaller less complex networks. If a large network is trained on several feature sets at once, it must learn and classify data of various magnitudes and natures. It is expected that by training smaller networks independently on differing feature sets, each will learn that feature set better than such a large network. If one individual network has problems classifying a certain type of pixel, perhaps the other two networks' results will overcome the third's mistakes once all results are combined, leading to a better overall classification.

- It is easier to evaluate the usefulness of a feature set when it is being classified independently by a network. For example, if all deep water pixels are continuously being misclassified as tree pixels, this error would be apparent in the network's final classification results. Instead of redesigning the network, the source of the error would be pinpointed to the feature set itself. If several feature sets were to be classified by a large network, the error in that specific feature set's deep water pixel measurements might be misinterpreted as errors in network design, or training time.

The remotely sensed feature set consists of six inputs from Landsat-TM's blue, green, red, near infrared and two middle infrared channels. This feature set provides measurements of the Earth's spectral properties. The two auxiliary feature sets added in the land cover classifications performed are geomorphology and process model feature sets. The geomorphology feature set consists of elevation, slope and aspect measurements which provide important topographical and spatial information of the land cover to be classified. The process model feature set consists of available energy and available moisture. Such variables contribute in achieving a better understanding of patterns present in species composition and diversity, as well as spatial heterogeneity (Dymond, 1998). These additional feature sets provide a link between the terrain and its various species' spectral properties.

The purpose of this research is to improve the accuracy of land cover classifications of remotely sensed data. Concentration is placed on improving the classification of canopy types in subalpine ecosystems. It is believed that by using combinations of smaller ANNs' results, classification accuracies will be increased and the effort required to construct and train smaller networks will be decreased. Also, it provides an opportunity to evaluate each feature set individually as it is trained and tested separately with its own network. Furthermore, the addition of auxiliary feature sets to remotely sensed data will also improve classification accuracy particularly of vegetative classes which are difficult to distinguish from one another.

This thesis is organised as follows. §2 introduces the theoretical background relevant to this research. §3 describes the study area and various feature sets employed in this research, while §4 outlines all methods and experiments implemented. Complete experimental results are detailed in §5, a thorough discussion of all results in context of other published research is presented in §6 and finally, conclusions are drawn with regards to the author's initial hypothesis in §7.

Chapter 2

Theoretical Background

2.1 Introduction

This research incorporates several areas of study - neural networks, remote sensing, classifier combinations, etc... This chapter presents the theoretical backgrounds of these areas and their significance to the research at hand. First, an introduction to ANNs and a summary of their use in RS is presented in §2.2. Next, a description of the three types of classifier outputs is addressed in §2.3. A review of various classifier combination methods follows in §2.4 while §2.5 introduces the methods of assessing accuracy which will be employed.

2.2 Artificial neural networks in remote sensing

The human brain is far superior to digital computers at many recognition tasks, such as interpreting sensory inputs e.g., face and speech recognition, and is therefore a major motivation behind the theory of ANNs. An ANN is a system of interconnected processing nodes, as shown in figure 2.1. NN_i's feature set is denoted by the input vector X_i, where i is the network index. The feature set's inputs are $X_i = [x_1, ..., x_{t_0}]^T$, where t_0 denotes the total number of inputs. This level is termed the input layer. The network weights are denoted by $w_{IJ}^{(l)}$, where I is the input index, J is the node index and l is the layer index. The bias input and bias weight are represented by $x_{0J}^{(l)}$ and $w_{0J}^{(l)}$ respectively. Layer l's nodes are denoted by $\Sigma_J^{(l)}$. These nodes compute a weighted sum of all their inputs. All layers in between the input and output layers are termed hidden layers. A multi-layered ANN can have several hidden layers each having several interconnecting nodes. The total number of nodes in hidden layer l is denoted by t_l. The weighted sums outputted by the network's nodes are represented by $s_J^{(l)}$. $f()$ performs some non-linear transformation on $s_J^{(l)}$, resulting in the inputs to the $l+1$ layer - a_J^l. The final layer of an ANN is termed the output layer. The number of output nodes/classes is represented by m, while c_j denotes the j^{th} output class. The non-linear output computed by network i for class j is denoted by $y_i(c_j)$. All computed values are assembled in the output vector $Y_i = [y_i(c_1), y_i(c_2), ..., y_i(c_m)]^T$, where m denotes the total number of output classes (Bishop, 1995). An ANN's outputs are estimates of a posteriori class probabilities (Cho & Kim, 1995; Kittler et al., 1998; Rogova, 1994).

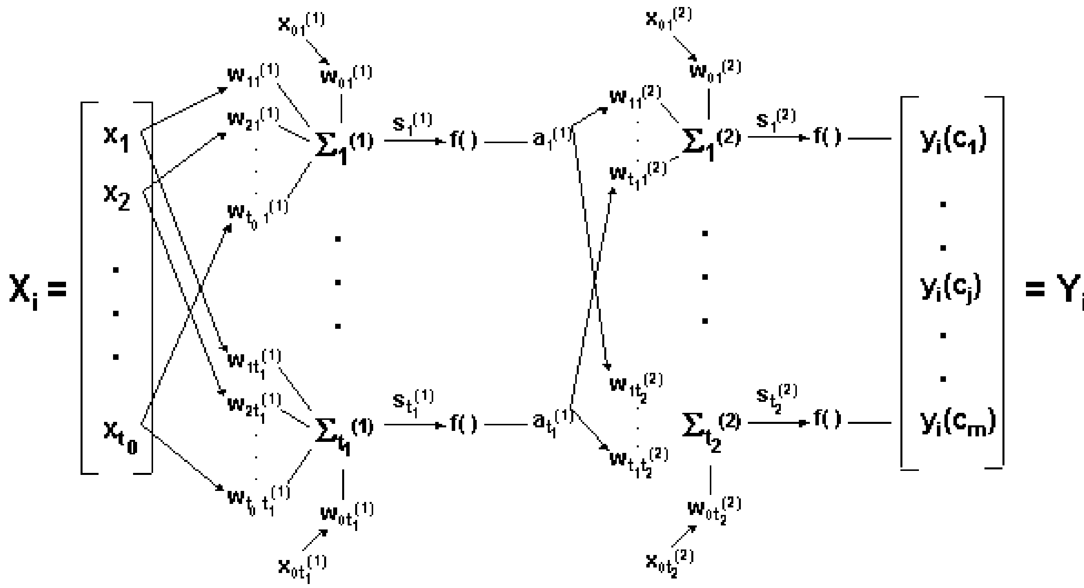

Figure 2.1: A two-layered artificial neural network model.

ANNs have been applied in many fields of study: computer science, psychology, biology and organic chemistry, to name but a few. As of the late 1980s, ANNs have been used to analyse remotely sensed data in various applications e.g., land cover classification (Gopal, Woodcock, & Strahler, 1999), cloud detection (Azimi-Sadjadi, Shaikh, Tian, Eis, & Reinke, 1996) and ocean color mapping (Ainsworth & Jones, 1999). ANNs have been most often applied to Landsat-TM data (Hepner et al., 1990; Kumar, Basu, & Majumdar, 1997; Li, Liu, & Sun, 1993) however, other forms of remotely sensed data analysed using ANNs are: Landsat Multispectral Scanner (Landsat-MSS) (Benediktsson, Ersoy, & Swain, 1991), synthetic aperture radar (SAR) (Foody et al., 1995), Advanced Very High Resolution Radiometer (AVHRR) (Moody, Gopal, Strahler, Borak, & Fisher, 1994), Système pour l'observation de la

Terre (SPOT) (Tzeng, Chen, Kao, & Fung, 1994), and Advanced Solid-State Array Spectroradiometer (ASAS) (Abuelgasim, Gopal, Irons, & Strahler, 1996). Recently, there has been an increase in ANN classifications of geomorphology features such as elevation, slope and aspect, in combination with remotely sensed information (Tenorio, Safavian, & Kassebaum, 1990). Combining multiple sources of remotely sensed data is also of interest to researchers. In Moody *et al.* (1994), AVHRR and Landsat-TM data were combined with the three above mentioned topographic variables to categorise vegetation classes. In Li *et al.* (1993), the spatial coordinates of each pixel were used as additional network inputs to Landsat-TM data, resulting in an overall accuracy increase from 66.19% to 77.21%.

In RS research, ANNs are often compared to conventional statistical methods such as MLC and k-nearest neighbour (k-NN). In Omatu & Yoshida (1993), a smaller MSE was achieved by an ANN compared to a MLC for the classification of Landsat-TM data - 0.00956 to 0.119 respectively. Even after performing an elimination of erroneous training pixels as described in §4.2.2, the ANN maintained a smaller MSE classification value. Another study compared a multi-layered ANN (Rumelhart, Hinton, & Williams, 1986) to a k-NN classifier on SAR and airborne thematic mapper feature sets, the multi-layered ANN achieving higher testing accuracies (Serpico, Bruzzone, & Roli, 1996). Foody *et al.* (1995) compared a discriminant analysis to an ANN for the classification of SAR data. In general, their results showed that markedly higher classification accuracies may be obtained from the ANN. As previously mentioned, Li *et al.*'s (1993) ANN containing spatial information performed better than the network without the additional information. However, both networks resulted in higher classification accuracies than a minimum distance classifier

(MDC). ANNs were chosen over other classifiers for this research due to their superior results achieved in the above mentioned RS studies.

Other RS studies have concentrated on comparing different types of ANNs to one another. The back-propagation training algorithm (BP) is one of the most commonly applied ANN training processes in RS classification. Various back-propagation artificial neural networks (BPNNs) (Rumelhart et al., 1986) were compared to a self-organising artificial neural network (SONN) in Tenorio *et al.* (1990) for the classification of Landsat-MSS and geomorphology feature sets. The SONN classifier resulted in a considerably shorter training time however, three of the four BPNNs achieved higher training and testing accuracies. Due to the many studies publishing the strengths of using BPNNs for the classification of remotely sensed data (Decatur, 1989; McClellan, DeWitt, Hemmer, Matheson, & Moe, 1989), BP was chosen as the network training algorithm to be employed in this research.

There are two general methods for adding extra feature sets to a RS classification problem. First, the additional feature sets can be added to the input space of the classifier, increasing its size and complexity. The second method consists of additional classifiers each classifying a specific new feature set. The results of the various classifiers are then combined using various algorithms, as presented in §2.4. There is an increase in the use of the later method. Researchers are moving away from using a single "best" classifier towards using consensus decisions. In Benediktsson, Swain & Ersoy (1990), Landsat-MSS was divided into visible and infrared feature sets and classified separately. The likelihoods generated by the different classifiers were then combined using each classifier's accuracy. Another study performed the same partitioning of Landsat-MSS data but used Dempster's orthogonal sum to combine the classifiers' likelihoods (Lee, Richards, & Swain, 1987). Similarly to this research, Collins, Dymond & Johnson (2001) partitioned Landsat-TM, geomorphology and process model feature sets using combinations of k-NN classifiers. Dempster's orthogonal sum was then employed to combine the classifiers' likelihoods. Results compared the consensus classification to a single large k-NN classifying all feature sets at once, the combination method resulting in a higher classification accuracy . In this research, combinations of multiple BPNNs were analysed because consensus decisions have shown to provide better classification accuracies in previous studies.

Although Paola & Schowengerdt (1995) say that ANNs need to be faster, easier to implement and more predictable in order to be considered a useful tool in RS, Foody *et al.* (1995) and Omatu & Yoshida (1993) agree that ANNs have considerable potential in classifying satellite images. Kanellopoulos & Wilkinson (1997) even boast that ANNs can be used with much "confidence for routine operational requirements in remote sensing". Given enough time to train, many researchers believe that any ANN can learn and accurately classify remotely sensed data.

2.3 Output types and transformations

There are three types of outputs that classifiers in general can produce. The first and simplest kind of output results when the classifier produces a "winner takes all" response; only one class is chosen as the winning class. This is a *type I* response. A *type II* response is obtained when a classifier outputs a ranking of classes. And finally, a *type III* response is outputted when probabilities or other such confidence ratings are assigned to each class by the classifier. As mentioned in the previous section, an ANN's outputs are estimates of a posteriori class probabilities and are therefore *type III* responses (Huang & Suen, 1995; Parker, 1997; Xu, Krzyżak, & Suen, 1992).

The reader is referred to Parker (1997) for full mathematical details regarding the six possible response type transformations. Seeing that ANNs are the only classifier employed in this research, only transformations from *type III* responses to either *type I* or *type II* responses are necessary. The following outlines these two transformations (Parker, 1997).

- *Type III* to *Type I*: Selecting the class with the highest estimated a posteriori probability value will produce a *type I* response.
- *Type III* to *Type II*: Arranging the estimated a posteriori probability values in descending order will result in a *type II* response.

As previously noted, *type III* responses can be probability estimates and in the case of ANNs, a posteriori class probabilities. It is sometimes necessary to normalise these values so that they sum to 1.0, for instance, when employing Bayesian combination techniques. As demonstrated by equation (2.1), normalisation is achieved by dividing each ANN's a posteriori class probabilities estimates by their sum.

$$\mathrm{norm}(y_i(c_j)) = \frac{y_i(c_j)}{\sum_{j=1}^{m} y_i(c_j)}, \tag{2.1}$$

is the normalised a posteriori class j probability for network i (Kittler et al., 1998). The *type I* and *type II* responses do not change after normalisation has occurred.

2.4 Combination methods

Multiple classifier results can be combined using either a combination method, such as voting or bagging (Breiman, 1996), or a separate classifier, such as an ANN or Quilan's C4.5 (Quilan, 1993). The classifier used to combine others' results can be of the same or of a different type of classifier then the ones being combined. It is also possible to combine results of different types of classifiers. Here is a brief summary of some combination methods applied to pattern recognition, and more particularly to RS. The reader is referred to the cited works as only the combination methods investigated in this research will be fully described.

The combination of multiple classifier results has been implemented in many areas of pattern recognition, such as personal identification using Bayesian based combinations (Kittler et al., 1998) and on-line handwritten character recognition using fuzzy integral combinations (Cho & Kim, 1995). Lipreading is a difficult speech recognition problem to solve because not all words can be uniquely visualised (Yu, Jiang, & Bunke, 1997). Three different classifiers based on hidden Markov models, time-delay ANNs and a one-dimensional Fourier transform were combined using a Borda count and other combination algorithms involving addition, subtraction, median and network weights in Yu *et al.* (1997). Results showed how these simple combination methods can lead to significant improvements in classification accuracies when compared to the individual classifiers' results. Boosting, the method of combining classifiers by weighing outputs using terms related to each classifier's accuracy on the training data, was applied to problems involving image analysis and chemical weapon detection in Prokop & Marchette (2000). The results demonstrated how very simple classifiers can be combined to form a reliable classifier, even when the individual classifiers performed poorly (Prokop & Marchette, 2000).

Classification of agricultural land use is just one RS problem to which combination methods have been applied (Giacinto, Roli, & Bruzzone, 2000; Vieira & Mather, 2000). Using voting, Bayesian and belief function combinations of a k-NN classifier and two ANNs, Giacinto *et al.* (2000) demonstrated how the combination of statistical and neural classifiers was an efficient method of obtaining higher agricultural classification accuracies when compared to the various individual classifiers investigated - Bayes, k-NN, BPNN, radial basis function ANN and probabilistic ANN (PNN). Vieira & Mather (2000) investigated combining classifiers using voting and Bayesian methods as well as an ANN. The highest crop classification accuracies were achieved using the ANN to combine the results of the individual classifiers - a MLC, two ANNs and a MDC. Collins *et al.* (2001) improved the classification of canopy types using Dempster's combination rule to join multiple k-NN classifiers.

This research investigates the different algorithms outlined from §2.4.1 to §2.4.5 for the combination of multiple ANNs. They were chosen due to their published use in RS and in combining ANNs, their levels of complexity - voting very simple to implement while OLC-MSE much more computationally complex, and their differing natures - certain techniques employ a priori knowledge while others incorporate a network uncertainty term.

2.4.1 Voting principles

Combination schemes based on voting principles are perhaps the easiest algorithms to understand and implement. Four of such combination methods are presented in the following sections.

Simple majority voting

The simple majority voting (SMV) algorithm employs *type I* responses. This combination scheme produces as output the class which has been chosen the most often by all networks. There is a specific restriction associated with this combination method, the winning class has to have been chosen by more than half of the networks (Cho & Kim, 1995). An error is assumed to have occurred if no one class was chosen by more than half of the networks (Hansen & Salamon, 1990). One plausible explanation for this minimum vote restriction is perhaps the avoidance of assigning pixels to classes for which they are truly not representative. The SMV combination assigns the pixel to class k if the following mathematical expression is satisfied,

$$k = \arg max_{j=1}^{m} e(c_j) \quad \text{and} \quad e(c_j) > n/2, \tag{2.2}$$

where n is the number of different networks, m is the number of output classes, c_j and c_k are output classes j and k respectively, and $e(c_k)$ represents the number of times class k was chosen by the various networks (Parker, 1997; Xu et al., 1992).

A simpler version of the SMV is the parliamentary or plurality majority voting (PMV) algorithm which removes the "more than half" restriction. PMV selects the class with the greatest amount of votes as the combination result (Hansen & Salamon, 1990; Parker, 1997).

Weighted majority voting

The weighted majority voting (WMV) method also combines *type I* responses. It uses a less restrictive expression by replacing $n/2$ with n/a in equation (2.2) (Xu et al., 1992). The analyst is allowed to decide to what extent a majority vote is needed in order to produce reliable results. The WMV is expressed as (Parker, 1997)

$$k = \arg max_{j=1}^{m} e(c_j) \quad \text{and} \quad e(c_j) > n/a, \tag{2.3}$$

where a is the weight placed on the classification.

Before using WMV to classify image pixels, the researcher must decide what value of a will be satisfactory for his/her purposes. By setting a to a value greater than 2, the restriction encountered by the SMV method is lessened. During training, various different values may be tested and their results compared. It is then the decision of the researcher to explain why a certain value was chosen over others.

Obviously, SMV becomes a special case of WMV when $a = 2$. Equation (2.3) will result in the same classification as (2.2). Seeing that increasing a is less restrictive, any value greater than 2 will generate the same result. By decreasing a, the classification becomes more restrictive and different results can now possibly be achieved.

Borda count

The Borda count is a value $B_i(c_k)$ which represents the number of classes network i ranked below a particular class k. The image pixel is assigned a value corresponding to the class with the largest Borda count summed over all networks (Cho & Kim, 1995). This method is unlike SMV and WMV as it utilises *type II* responses. Mathematically, the pixel to be classified is assigned to class k if,

$$k = \arg max_{j=1}^{m} \sum_{i=1}^{n} B_i(c_j), \tag{2.4}$$

where i is the ANN index for $i = 1, ..., n$.

2.4.2 Bayesian approaches

The following Bayesian approaches combine normalised *type III* ANN responses. The expression X_i represents the input vector X used by network i for $i = 1, ..., n$, where n indicates the total number of networks. A specific input vector represents a specific feature set.

Bayes theory of classification states that the image pixel to be classified should be assigned to class k if

$$k = \arg max_{j=1}^{m} P(c_j|X_1, ..., X_n), \tag{2.5}$$

where $P(c_j|X_1, ..., X_n)$ is the a posteriori probability for class j computed by considering all network input vectors simultaneously (Kittler et al., 1998; Xu et al., 1992). Equation (2.5) is a correct but impractical proposition of the classification problem and therefore, Kittler *et al.* (1998) simplify it "in terms of decision support computations performed by individual classifiers, each exploiting only the information conveyed by vector X_i."

First the a posteriori class probability $P(c_j|X_1, ..., X_n)$ is rewritten in the following form using Bayes theorem,

$$P(c_j|X_1, ..., X_n) = \frac{p(X_1, ..., X_n|c_j)P(c_j)}{p(X_1, ..., X_n)}, \tag{2.6}$$

where $p(X_1, ..., X_n|c_j)$ is the joint probability distribution of the input vectors conditioned on the output classes, $P(c_j)$ is the a priori class probability and $p(X_1, ..., X_n)$ is the unconditional input vector joint probability density. This in turn can be expressed in terms of the conditional input vector distributions by (Kittler, Hojjatoleslami, & Windeatt, 1997)

$$p(X_1, ..., X_n) = \sum_{k=1}^{m} p(X_1, ..., X_n|c_j)P(c_k). \tag{2.7}$$

Next, equations (2.5), (2.6) and (2.7) are used in formulating the product and summation classifier combination rules, which in turn are employed in forming other Bayesian combination schemes. The complete mathematical derivations of sections §2.4.2 to §2.4.2 can be found in Appendix A.

Product rule

The product rule (PROD),

$$k = \arg max_{j=1}^{m} P^{-(n-1)}(c_j) \prod_{i=1}^{n} P(c_j|X_i). \tag{2.8}$$

assigns the image pixel to the class k with the largest product value of a posteriori probabilities over all classifiers. The product of $P(c_j|X_i)$ must be computed for each class j over all networks i. These products are then multiplied by their a priori class probability $P(c_j)$ raised to the power of $-(n-1)$. Complete mathematical derivations of PROD are outlined in Appendix A.1.

Summation rule

The summation rule (SUM),

$$k = \arg max_{j=1}^{m}[(1-n)P(c_j) + \sum_{i=1}^{n} P(c_j|X_i)]. \tag{2.9}$$

assigns the image pixel to the class k with the largest summation of a posteriori probabilities over all classifiers. SUM is similar to PROD in that it also uses a priori class probabilities. Instead of multiplying the values of $P(c_j|X_i)$, they are summed for each class j over all networks i. Next, these values are added to their respective a priori class probability $P(c_j)$ multiplied by a factor of $1-n$. Complete mathematical derivations of SUM are outlined in Appendix A.2.

Maximum rule

The maximum rule (MAX),

$$k = \arg max_{j=1}^{m}\{max_{i=1}^{n}P(x_j|X_i)\}. \tag{2.10}$$

assigns the image pixel to the class with the largest maximum value of a posteriori probabilities computed over all classifiers. The maximum values of $P(c_j|X_i)$ are chosen for each class j over all networks i. The largest of these values is then selected and the pixel is assigned to its associated class. Complete mathematical derivations of MAX are outlined in Appendix A.3.

Minimum rule

The minimum rule (MIN),

$$k = \arg max_{j=1}^{m}\{min_{i=1}^{n}P(c_j|X_i)\}. \tag{2.11}$$

assigns the image pixel to the class with the largest minimum value of a posteriori probabilities computed over all classifiers. The minimum values of $P(j|x_i)$ are chosen for each class j over all networks i. And similarly to MAX, the pixel is assigned to the class with the largest of these values. Complete mathematical derivations of MIN are outlined in Appendix A.4.

Average rule

The average rule (AVE),

$$k = \arg max_{j=1}^{m} \frac{1}{n} \sum_{i=1}^{n} P(c_j|X_i). \tag{2.12}$$

assigns the image pixel to the class with the largest average value of a posteriori probabilities computed over all classifiers. The average value of $P(j|x_i)$ is computed for each class j over all networks i. The pixel is then assigned to the class with the largest of these average values. Complete mathematical derivations of AVE are outlined in Appendix A.5.

Median rule

The median rule (MED),

$$k = \arg max_{j=1}^{m} \{med_{i=1}^{n} P(c_j|X_i)\}. \tag{2.13}$$

assigns the image pixel to the class with the largest median value of a posteriori probabilities computed over all classifiers. The median must first be selected for each class j over all networks i. The class with the largest median value is the one to which the pixel is assigned. Complete mathematical derivations of MED are outlined in Appendix A.6.

2.4.3 Dempster's orthogonal sum

Applying Dempster's orthogonal sum to an ANN problem is quite simple. Let NN_i denote the i^{th} network. As previously mentioned, each network computes an output vector of a posteriori class probabilities denoted by Y_i. Possible output classes are represented by the set $C = (c_1, c_2, ...c_m)$, where m denotes the total number of output classes. The computed estimates of a posteriori class probabilities for NN_i can then be represented by the vector

$$Y_i = \left\{ \begin{array}{c} y_i(c_1) \\ y_i(c_2) \\ \dots \\ y_i(c_m) \end{array} \right\}, \tag{2.14}$$

where $y_i(c_j)$ is the estimated a posteriori class j probability computed by network i.

Dempster's combination rule models the uncertainty associated with each network. As well as producing a posteriori probabilities for each output class, each network must have an associated value θ_i, network i's uncertainty value. Network uncertainty can be determined in various ways. One method is to use the network's performance during training. If NN_i classified 90% of the training pixels correctly, then its uncertainty could equal 0.10. It is the responsibility of the researcher to explain his or her reasons for choosing a specific value for each network's uncertainty. In this research, each network's uncertainty equalled $1 - K_i$, where K_i was network i's overall testing kappa value.

The following computed product, M_i, is the evidential mass of network i for all output classes and its associated uncertainty, (Shafer, 1976)

$$M_i(\langle c_1, c_2, ...c_m, \theta\rangle) = \langle y_i(c_1)(1-\theta_i), y_i(c_2)(1-\theta_i), ..., y_i(c_m)(1-\theta_i), \theta_i\rangle\,. \tag{2.15}$$

The various network masses are then combined using Dempster's orthogonal sum (DOS) resulting in a consensus classification decision. DOS is equal to the total support for a particular class divided by the total non-contradictory evidence for all classes (Shafer, 1976). The notation $M(c_j)$ denotes the orthogonal sum for class j.

Before computing each class's DOS, it is best to visualise the problem as applied to ANNs. Assume that two networks (NN_1 and NN_2) are classifying an image pixel to one of three possible output classes (c_j, for $j = 1, 2, 3$). The uncertainty associated with each network is then denoted by θ_1 for the first network and θ_2 for the second. Seeing that this example employs two networks, one can imagine the evidential masses as subdividing a unit square. A unit cube would be visualised when three networks are used, and so forth. From figure 2.2, it is easy to extract the evidential areas of the square supporting each output class as applied to a RS classification problem.

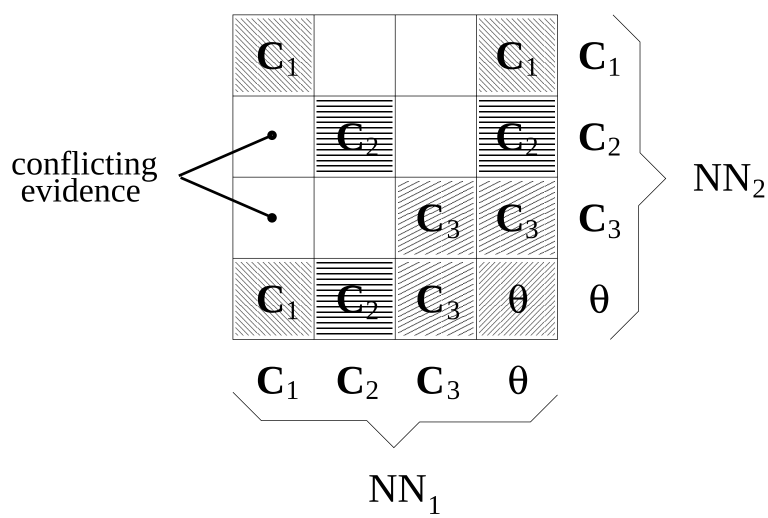

Figure 2.2: A unit square representative of the various evidence areas supporting three outputs classes (c_1, c_2, c_3) and uncertainty (θ) for two artificial neural networks.

The DOS for class j is equal to

$$M(c_j) = \frac{M_1(c_j)M_2(c_j) + M_1(c_j)M_2(\theta) + M_1(\theta)M_2(c_j)}{\sum_{j=1}^{3}(M_1(c_j)M_2(c_j) + M_1(c_j)M_2(\theta) + M_1(\theta)M_2(c_j)) + M_1(\theta)M_2(\theta)}, \tag{2.16}$$

while

$$M(\theta) = \frac{M_1(\theta)M_2(\theta)}{\sum_{j=1}^{3}(M_1(c_j)M_2(c_j) + M_1(c_j)M_2(\theta) + M_1(\theta)M_2(c_j)) + M_1(\theta)M_2(\theta)} \tag{2.17}$$

is the uncertainty orthogonal sum. It should be noted that this expression can be extended to include any number of networks and output classes. Once all DOS are computed for every output class j and uncertainty θ, the pixel is assigned to the class with the maximum of these combined evidential masses or remains unclassified if the uncertainty value is the maximum.

2.4.4 Fuzzy logic

Randomness and fuzziness are two kinds of uncertainties which can be treated mathematically. The concept of probabilities is known with respect to randomness and that of fuzzy sets with respect to fuzziness. No direct comparison between probabilities and fuzzy sets can be made (Sugeno, 1977). The following is a multiple ANN combination scheme based on the concept of fuzzy logic (FUZ), specifically the fuzzy integral. A fuzzy integral non-linearly combines networks' outputs with subjective evidence of each individual network's importance (Cho & Kim, 1995).

As noted in Cho & Kim (1995), the fuzzy integral introduced by Sugeno (1977) and the fuzzy measures from Yager (1993) and Leszeynski *et al.* (1995) are very useful in combining information. The following definitions and concepts descend from the mathematics presented in these aforementioned works.

A fuzzy measure, g, is a set function such that,

- the fuzzy measure of an empty set is equal to zero - $g(\emptyset) = 0$,

- the fuzzy measure of an entire set Q is equal to one - $g(Q) = 1$, and
- the fuzzy measure of set A is less than or equal to that of set B if A is a subset of B - $g(A) \leq g(B)$ if $A \subset B$.

If the following additional property is also satisfied, the fuzzy measure is referred to as a g_λ-fuzzy measure.

$$\forall\ A, B \subset Q \ \text{ and } \ A \cap B = \emptyset, \quad g(A \cup B) = g(A) + g(B) + \lambda g(A)g(B), \qquad (2.18)$$

for some $\lambda > -1$ (Cho & Kim, 1995; Leszeynski, Penczek, & Grochulskki, 1985; Yager, 1993).

When combining multiple ANNs, let g^i denote the fuzzy measure of network i. These measures can be interpreted as quantifying how well a network properly classified the image pixels. They must be known and can be determined in different ways i.e., the fuzzy measure of a network could equal the ratio of correctly classified pixels during supervised training over the total number of pixels being classified. In this research, each network's fuzzy measure equalled $1 - K_i$, where K_i was network i's overall testing kappa value.

A fuzzy integral is a non-linear function defined with respect to fuzzy measures, particularly g_λ-fuzzy measures. This function can be interpreted as finding the maximal grade of agreement between networks' outputs and their fuzzy measures for a particular class (Cho & Kim, 1995; Sugeno, 1977). In order to better comprehend the mathematics of a fuzzy integral, an ANN combination algorithm employing such a function is presented next.

A pixel is being classified to one of m possible output classes, c_j for $j = 1, ..., m$. The outputs of n different networks are being combined, where NN_i denotes the i^{th}

network. First, these networks must be renumbered/rearranged such that their a posteriori class probabilities are in descending order of magnitude for each output class j,

$$y_1(c_j) \geq y_2(c_j) \geq ... \geq y_n(c_j), \tag{2.19}$$

where $y_i(c_j)$ is the i^{th} network's a posteriori class j probability. Next, each network/set of networks' g_λ-fuzzy measure is computed for every output class j and is denoted by $g_j(A_i)$. $A_i = \{NN_1, NN_2, ..., NN_i\}$ is the set of the first i networks ordered correspondingly to class j's associated a posteriori probabilities. These values can then be computed using the following recursive method,

$$g_j(A_1) = g_j(\{NN_1\}) = g^1, \tag{2.20}$$

$$g_j(A_i) = g_j(\{NN_1, ..., NN_i\}) = g^i + g(A_{i-1}) + \lambda g^i g(A_{i-1}) \text{ for } 1 < i < n, \tag{2.21}$$

$$g_j(A_n) = g_j(\{NN_1, ..., NN_n\}) = 1, \tag{2.22}$$

where $\lambda > -1$, $\lambda \neq 0$ and is determined by the following equation (Cho & Kim, 1995),

$$\lambda + 1 = \prod_{i=1}^{n}(1 + \lambda g^i). \tag{2.23}$$

Finally, the fuzzy integral for each class j is defined as,

$$max_{i=1}^{n}[min[y_i(c_j), g_j(A_i)]]. \tag{2.24}$$

The class with the largest fuzzy integral value is then chosen as the output class to which the pixel is classified. Equation (2.25) summarises combining multiple ANNs using a fuzzy integral approach.

$$max_{\text{class}} \begin{bmatrix} max_{\text{network}} & \begin{bmatrix} min[y_1(c_1), g_1(A_1)] \\ min[y_2(c_1), g_1(A_2)] \\ ... \\ min[y_n(c_1), g_1(A_n)] \end{bmatrix} \\ max_{\text{network}} & \begin{bmatrix} min[y_1(c_2), g_2(A_1)] \\ min[y_2(c_2), g_2(A_2)] \\ ... \\ min[y_n(c_2), g_2(A_n)] \end{bmatrix} \\ & ... \\ max_{\text{network}} & \begin{bmatrix} min[y_1(c_m), g_m(A_1)] \\ min[y_2(c_m), g_m(A_2)] \\ ... \\ min[y_n(c_m), g_m(A_n)] \end{bmatrix} \end{bmatrix} . \tag{2.25}$$

2.4.5 Optimal linear combination - Mean squared error reduction

The name of this method explains its objective quite clearly: to find an optimal linear combination of the ANNs' results by reducing each network's mean squared error (OLC-MSE). During supervised training, the true classifications of the pixels are used to improve the combination weight vector Λ by reducing network mean squared error (Hashem & Schmeiser, 1995). *Type I* responses are combined using this scheme. Network i employs the input vector $X_i(\rho_\iota)$ to represent a particular feature set for pixel ρ_ι, where ι is the pixel index. A scalar response $v_i(\rho_\iota)$ is outputted, the class to which the pixel is classified. Pixel ι's true class is denoted by the scalar value $\tau(\rho_\iota)$.

The example values below show that network i classifies pixel ι to output class 4 when it truly belongs to output class 2:

$$v_i(\rho_\iota) = 4, \quad \tau(\rho_\iota) = 2. \tag{2.26}$$

Network i's error for pixel ι, δ_i, can be computed from its true and outputted classes using

$$\delta_i(\rho_\iota) = \tau(\rho_\iota) - v_i(\rho_\iota). \tag{2.27}$$

Using our values from (2.26), the error would equal $\delta_i(\rho_\iota) = 2 - 4 = -2$. If we extend this idea to n ANNs, their linearly combined scalar output for pixel ι would result in

$$v(\rho_\iota, \Lambda) = \Lambda^T \overline{v}(\rho_\iota) = \sum_{i=1}^{n} \alpha_i v_i(\rho_\iota), \tag{2.28}$$

where $\overline{v}(\rho_\iota) = (v_1(\rho_\iota), v_2(\rho_\iota), ..., v_n(\rho_\iota))^T$ is the n by 1 scalar output vector of the n networks. $\Lambda = (\alpha_1, \alpha_2, ..., \alpha_n)^T$ is the n by 1 combination weight vector with elements α_i, individual network combination weights. The approximation error of pixel ι's classification under the combination of these n ANNs is defined as the scalar value (Hashem & Schmeiser, 1995)

$$\delta(\rho_\iota; \Lambda) = \tau(\rho_\iota) - v(\rho_\iota; \Lambda). \tag{2.29}$$

The real task at hand is finding values for the combination weight vector Λ. These values should not be randomly assigned; they should be appropriate and "good". Remember, it is the reduction of the classification errors through combination which is sought. One simple answer would be to choose a "best" network, say i, and setting $\alpha_i = 1$ and $\alpha_I = 0, \forall \;\; I \neq i$ (Hashem & Schmeiser, 1995). As mentioned in §1, choosing the single "best" network risks loosing valid information contained within the results of the others. Another approach, known as simple averaging, uses equal combination weights i.e., $\alpha_1 = \alpha_2 = ... = \alpha_n = 1/n$. However, this method assumes that all network outputs are equally good (Clemen, 1989; Hashem & Schmeiser, 1995).

As stated in the title of this method, the MSE is the basis for this network combination scheme. The desire is to minimise MSE defined by

$$\mathrm{MSE}(v(\rho_\iota, \Lambda)) = \mathrm{E}((\delta(\rho_\iota; \Lambda))^2), \tag{2.30}$$

where E is the mean or expected value of the squared error, $(\delta(\rho_\iota; \Lambda))^2$. The value of Λ that minimises equation (2.30) can be computed by taking the derivative of the MSE with respect to Λ and then equating this derivative to zero. There are two cases for which this can be done: 1) when the α_i's are constrained to sum to one (Mcon), $\sum_{i=1}^{n} \alpha_i = 1$, and 2) when they are not constrained (Mun) (Hashem & Schmeiser, 1995). The unconstrained case is examined first.

Assuming that the combination weights are not constrained to sum to one, the MSE can be expanded as follows (Hashem & Schmeiser, 1995)

$$\begin{aligned}
\mathrm{MSE}(v(\rho_\iota, \Lambda)) &= \mathrm{E}((\delta(\rho_\iota; \Lambda))^2) = E((\tau(\rho_\iota) - \Lambda^T \overline{v}(\rho_\iota))^2) \\
&= E(\tau^2(\rho_\iota) - 2\tau(\rho_\iota)\Lambda^T \overline{v}(\rho_\iota) + (\Lambda^T \overline{v}(\rho_\iota))^2) \\
&= E(\tau^2(\rho_\iota)) - E(2\tau(\rho_\iota)\Lambda^T \overline{v}(\rho_\iota)) + E(\Lambda^T \overline{v}(\rho_\iota))^2). \qquad (2.31)
\end{aligned}$$

To simplify terminology a little, let Θ be the n by 1 vector

$$\Theta = \left\{ \begin{array}{c} E(\tau(\rho_\iota)v_1(\rho_\iota)) \\ E(\tau(\rho_\iota)v_2(\rho_\iota)) \\ \dots \\ E(\tau(\rho_\iota)v_n(\rho_\iota)) \end{array} \right\}, \tag{2.32}$$

and Ψ be the n by n matrix (Hashem & Schmeiser, 1995)

$$\Psi = \left\{ \begin{array}{cccc} E(v_1(\rho_\iota)v_1(\rho_\iota)) & E(v_1(\rho_\iota)v_2(\rho_\iota)) & \dots & E(v_1(\rho_\iota)v_n(\rho_\iota)) \\ E(v_2(\rho_\iota)v_1(\rho_\iota)) & E(v_2(\rho_\iota)v_2(\rho_\iota)) & \dots & E(v_2(\rho_\iota)v_n(\rho_\iota)) \\ \dots & \dots & \dots & \dots \\ E(v_n(\rho_\iota)v_1(\rho_\iota)) & E(v_n(\rho_\iota)v_2(\rho_\iota)) & \dots & E(v_n(\rho_\iota)v_n(\rho_\iota)) \end{array} \right\}. \tag{2.33}$$

Substituting equations (2.32) and (2.33) into equation (2.31), the MSE becomes

$$\mathrm{MSE}(v(\rho_\iota, \Lambda)) = E(\tau^2(\rho_\iota)) - 2\Lambda^T\Theta + \Lambda^T\Psi\Lambda. \tag{2.34}$$

By differentiating equation (2.34) with respect to Λ and equating to it zero, the n by 1 vector of combination weights which minimises the MSE is then

$$\Lambda^T = \Psi^{-1}\Theta, \tag{2.35}$$

where the exponent -1 denotes the inverse of the matrix Ψ (Hashem & Schmeiser, 1995). Next, the constrained situation is outlined.

Another format for expressing $\sum_{i=1}^{n} \alpha_i = 1$ is $\Lambda^T\overline{1} = 1$ where $\overline{1}$ is a n by 1 unit vector. The expression for the approximation error can then be rewritten as (Hashem & Schmeiser, 1995)

$$\begin{aligned} \delta(\rho_\iota; \Lambda) &= \tau(\rho_\iota) - v(\rho_\iota; \Lambda) = \tau(\rho_\iota)\Lambda^T\overline{1} - \Lambda^T\overline{v}(\rho_\iota) \\ &= \Lambda^T(\tau(\rho_\iota)\overline{1} - \overline{v}(\rho_\iota)) = \Lambda^T\overline{\delta}(\rho_\iota), \end{aligned} \tag{2.36}$$

where $\overline{\delta}(\rho_\iota)$ is the n by 1 approximation error vector

$$\overline{\delta}(\rho_\iota) = \left\{ \begin{array}{c} \tau(\rho_\iota) - v_1(\rho_\iota) \\ \tau(\rho_\iota) - v_2(\rho_\iota) \\ \dots \\ \tau(\rho_\iota) - v_n(\rho_\iota) \end{array} \right\}. \tag{2.37}$$

From equation (2.36), the MSE becomes

$$\begin{aligned} \mathrm{MSE}(v(\rho_\iota;\Lambda)) &= E((\delta(\rho_\iota;\Lambda))^2) = E((\Lambda^T\overline{\delta}(\rho_\iota))^2) \\ &= E((\Lambda^T\overline{\delta}(\rho_\iota))(\Lambda^T\overline{\delta}(\rho_\iota))^T) \\ &= E(\Lambda^T\overline{\delta}(\rho_\iota)\overline{\delta}(\rho_\iota)^T\Lambda) = \Lambda^T\Omega\Lambda, \end{aligned} \tag{2.38}$$

where Ω is the following n by n matrix

$$\Omega = \left\{ \begin{array}{cccc} E(\delta_1(\rho_\iota)\delta_1(\rho_\iota)) & E(\delta_1(\rho_\iota)\delta_2(\rho_\iota)) & \dots & E(\delta_1(\rho_\iota)\delta_n(\rho_\iota)) \\ E(\delta_2(\rho_\iota)\delta_1(\rho_\iota)) & E(\delta_2(\rho_\iota)\delta_2(\rho_\iota)) & \dots & E(\delta_2(\rho_\iota)\delta_n(\rho_\iota)) \\ \dots & \dots & \dots & \dots \\ E(\delta_n(\rho_\iota)\delta_1(\rho_\iota)) & E(\delta_n(\rho_\iota)\delta_2(\rho_\iota)) & \dots & E(\delta_n(\rho_\iota)\delta_n(\rho_\iota)) \end{array} \right\}. \tag{2.39}$$

and the approximation error of the i^{th} ANN is denoted by $\delta_i(\rho_\iota)$ for pixel ι (Hashem & Schmeiser, 1995).

Solving Λ^T for the constrained case is more difficult than the unconstrained situation. The Lagrangian function Ł$= \Lambda^T\Omega\Lambda + 2\eta(\Lambda^T\overline{1} - 1)$ must be differentiated, producing

$$\partial Ł/\partial\Lambda = 2\Omega\Lambda + 2\eta\overline{1}. \tag{2.40}$$

Setting equation (2.40) equal to zero and solving for Λ yields

$$\Lambda^T = -\eta\Omega^{-1}\overline{1}. \tag{2.41}$$

By pre-multiplying equation (2.41) by $\overline{1}^T$ and using the fact that $\Lambda^T\overline{1} = \overline{1}^T\Lambda = 1$, the values of η and Λ^T are (Hashem & Schmeiser, 1995)

$$\eta = \frac{-1}{\overline{1}^t\Omega^{-1}\overline{1}} \quad \text{and} \quad \Lambda^T = \frac{\Omega^{-1}\overline{1}}{\overline{1}^T\Omega^{-1}\overline{1}}. \tag{2.42}$$

Seeing that the distribution function of the input vector X_i is usually unknown, the values of Θ, Ψ and Ω in equations (2.35) and (2.42) need to be estimated (Hashem & Schmeiser, 1995). Let the input vector belong to some set O, $X_i \in O$. So far, this discussion has been limited to one single input vector X_i. However, network i will input many vectors during training, each representing the feature values of the different image pixels to be classified. For example, if NN_i is being trained with 5 image pixels, the network's input vectors would be denoted by $X_i(\rho_1), X_i(\rho_2), ..., X_i(\rho_5)$ for pixels $\rho_1, \rho_2, ..., \rho_5$. The components of Θ, Ψ and Ω can therefore be approximated by

$$\hat{\Theta}_i = \sum_{\iota=1}^{|O|} \frac{(\tau(\rho_\iota)v_i(\rho_\iota))}{|O|}, \forall\ i; \tag{2.43}$$

$$\hat{\Psi}_{iI} = \sum_{\iota=1}^{|O|} \frac{(v_i(\rho_\iota)v_I(\rho_\iota))}{|O|}, \forall\ i, I; \tag{2.44}$$

$$\hat{\Omega}_{iI} = \sum_{\iota=1}^{|O|} \frac{(\delta_i(\rho_\iota)\delta_I(\rho_\iota))}{|O|}, \forall\ i, I; \tag{2.45}$$

where i and I are network indices, ι is the pixel index, and $|O|$ represents the cardinality of O (Hashem & Schmeiser, 1995).

2.5 Accuracy assessment

The performance of an ANN can be analysed using its confusion matrix from which, it is possible to read off the number of pixels belonging to a certain class which have been correctly classified and misclassified. A confusion matrix's columns represent known class membership whereas its rows represent the class to which the pixel is assigned. The column sums represent the total "true" pixels while the row sums represent the total "classified" pixels (Congalton & Green, 1999). For example, table 5.2 of §5.3 denotes that there are 12 actual aspen pixels. However, from the first row's sum one reads that 13 pixels have been classified as aspen pixels. Of the 12 actual aspen pixels, 7 are properly classified and 5 are misclassified as pine pixels. Of the 13 pixels classified as aspen, once again 7 are actual aspen pixels, 2 are actually pine pixels, 1 is a pine/spruce pixel and 3 are really spruce/fir pixels. The values on the diagonal are therefore the number of correctly categorised pixels in each class.

The general format for a confusion matrix is depicted in table 2.1, where output class j is denoted by c_j, cm_{kj} is the number of pixels belonging to class j that were classified to class k and cm_{jj} is the number of properly categorised pixels to class j. The number of pixels classified to class j is denoted by sum_{j+} while the actual number of pixels belonging to class j is represented by sum_{+j}. The total number of pixels being classified is denoted by sum_{++}.

Classifier Categories	**Known Categories**						
	c_1	c_2	...	c_j	...	c_m	*Sum*
c_1	cm_{11}	cm_{12}	...	cm_{1j}	...	cm_{1m}	sum_{1+}
c_2	cm_{21}	cm_{22}	...	cm_{2j}	...	cm_{2m}	sum_{2+}
...			...		...		...
c_j	cm_{j1}	cm_{j2}	...	cm_{jj}	...	cm_{jm}	sum_{j+}
...			...		...		...
c_m	cm_{m1}	cm_{m2}	...	cm_{mj}	...	cm_{mm}	sum_{m+}
Sum	sum_{+1}	sum_{+2}	...	sum_{+j}	...	sum_{+m}	sum_{++}

Table 2.1: Confusion matrix layout

Overall and class-conditional kappas are two accuracy values which can be computed from a confusion matrix. Overall kappas (K) were computed for all ANNs. K is used to determine whether a confusion matrix is statistically different from another (Bishop, Fienberg, & Holland, 1975). It is a measure of agreement between the actual and chance agreement of the classification and is similar to the Chi square analysis. For example, if a an overall kappa value of 0.87 is achieved, this can be interpreted as the classification performing 87% better than randomly assigning pixels to output classes. The actual agreement is represented by the major diagonal in the classification's confusion matrix while chance agreement is indicated by its row and column totals. The closer a K value is to one, the higher its respective ANN's classification accuracy will be. Network's i's overall kappa value, K_i, is computed using the following equation

$$K_i = \frac{\mathrm{sum}_{++} \sum_{j=1}^{m} \mathrm{cm}_{jj} - \sum_{j=1}^{m} \mathrm{cm}_{j+}\mathrm{cm}_{+j}}{\mathrm{sum}_{++}^2 - \sum_{j=1}^{m} \mathrm{cm}_{j+}\mathrm{cm}_{+j}}, \tag{2.46}$$

where all notation is as previously defined (Congalton & Green, 1999).

Class-conditional kappas (κ) were also computed for each individual class. κ combines a class's user's accuracy and producer's accuracy in such a way as to provide a better understanding of the its overall accuracy. It is also a measure of agreement between the actual and chance agreement of the classification. The closer a κ value is to one, the higher its respective class's classification accuracy will be. Class j's conditional kappa value, κ_j, is computed using the following equation

$$\kappa_j = \frac{\mathrm{sum}_{++}\mathrm{cm}_{jj} - \mathrm{cm}_{j+}\mathrm{cm}_{+j}}{\mathrm{sum}_{++}\mathrm{cm}_{j+} - \mathrm{cm}_{j+}\mathrm{cm}_{+j}}, \tag{2.47}$$

where all notation is as previously defined (Congalton & Green, 1999).

Chapter 3

Study Area and Feature Sets

3.1 Introduction

Ecologists can use predictive vegetation mapping systems (PVMS) to map and comprehend vegetative spatial patterns across a landscape (Franklin, 1995). PVMS are used to estimate a number of vegetation properties such as diversity and abundance, and can map existing and potential vegetation growth. Environmental gradient maps and remote sensing images are two broad categories covering possible PVMS data types (Collins, Dymond, & Johnson, 2001). Environmental gradients are subdivided into indirect environmental, direct environmental and resource gradients. Indirect gradients, such as elevation, do not have a direct physiological influence on plant growth; direct gradients, such as temperature, have a direct physiological influence on vegetation but are not used directly for growth; while resource gradients, such as soil moisture, are directly used by plants (Austin, Cunningham, & Fleming, 1984). Two fundamentally different types of data, remote sensing images are observations of the vegetation under study whereas environmental gradient maps provide information regarding the environment in which the vegetation grows.

3.2 Study area

The data employed in this research was collected from 58 field sites in the Kananaskis River and Spray Reservoir Watersheds covering 1184 km^2 of high relief mountains. Located in the front ranges of the Southern Canadian Rockies at approximately 51^o North and 115^o West, see figure 3.1, the outflow from the valley is at 1350 m above sea level while the mountain peaks are from 2400 and 3000 m above sea level (Dymond, 1998).

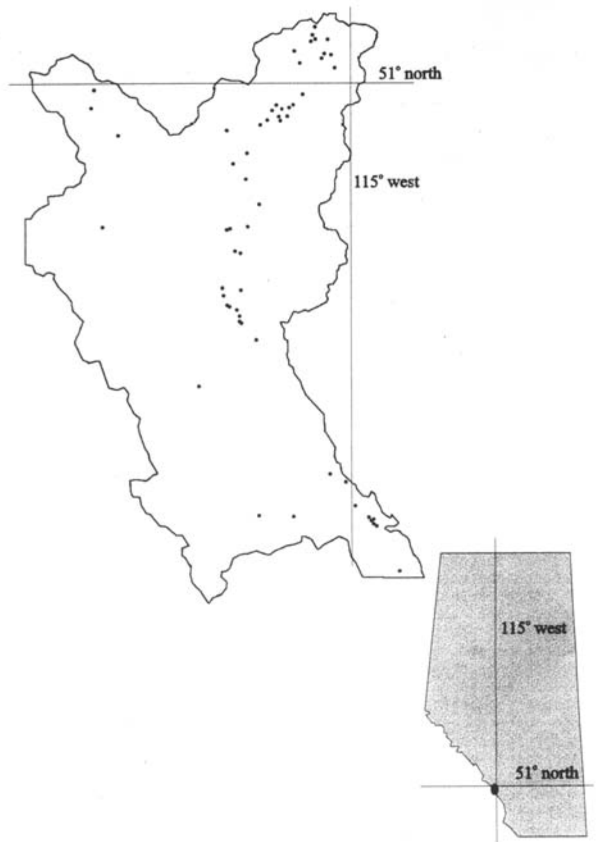

Figure 3.1: Location of study area in Alberta's Kananaskis Valley; black circles represent individual stands, source: Dymond (1998), no scale provided.

The study area's climate transitions between a Continental and a Cordilleran climate. Average precipitation values have summer and winter maximums of 102.6 mm and 22.3 mm respectively. Minimum values occur in autumn and again in late winter (Janz & Storr, 1977). The lowest average temperatures occur in January with a value -6^{o}C, and the highest average occurs in July with a value of 14^{o}C (Kananaskis Field Stations, 1410 m above sea level). The climate also changes spatially with elevation (Storr & Ferguson, 1972). From 474.5 mm at 1400 m above sea level to 1110.5 mm at 2400 m above sea level, the average annual precipitation increased linearly during an 8 year study of a sub-basin in the watershed. The same study revealed that the average July daily maximum temperature decreased from 21.9^{o}C at 1400 m above sea level to 14.2^{o}C at 2400 m above sea level (Janz & Storr, 1977).

Human disturbances have had little impact on most of the study area (Johnson & Fryer, 1987). After 62 years (1886-1947), logging ended due to the loss of leases burned by a large fire in 1936. However, logging did not significantly impact the composition of the area's species and human impact is now limited to recreational facilities, hydroelectric dams and reservoirs. Fire is the area's dominant disturbance due to the intensity and reoccurrence of large fires within the lifetime of post-fire generation trees (Johnson & Larsen, 1991). However, the disturbance rate is equivalent throughout the study area as frontal fire intensities are so high that all is burned regardless of vegetation type or elevation (Fryer & Johnson, 1988).

Ranging in elevation from 1350 to approximately 2350 m above sea level, this study area's subalpine forest is mainly composed of conifers. Four of the area's vegetative classes are being investigated in this research: trembling aspen (*Populus* *tremuloides* Michx.); lodgepole pine (*Pinus contorta* Dougl. ex Loud. var. *latifolia* Engelm.); a combination of lodgepole pine and of Engelman/white spruce hybrids (*Picea glauca* x *Picea engelmanni*); and a combination of subalpine fir (*Abies lasiocarpa* (Hook.) Nutt.) and of Engelman/white spruce hybrids. These four vegetative classes will henceforth be abbreviated as the aspen (AS), pine (PI), pine/spruce (PS) and spruce/fir (SF) output classes respectively.

3.3 Remotely sensed feature set

The first data group in this study is the remotely sensed feature set consisting of information recorded by the Landsat-TM satellite. Acquired on August 8^{th}, 1984 at a 30 m resolution, figure 3.2 depicts the study area's satellite image. Remote sensing images represent spatially continuous electromagnetic properties of the Earth's surface by measuring electromagnetic radiation. Such data can be used to estimate physical properties of the landscape such as canopy density, but are most often applied to classification algorithms, associating measured radiance to landscape categories (Collins et al., 2001). RS is very useful for this last application as there is a strong correlation between land cover classes and measured radiance, particularly with electro-optical reflectance (Asrar, Myneni, & Kanemasu, 1989).

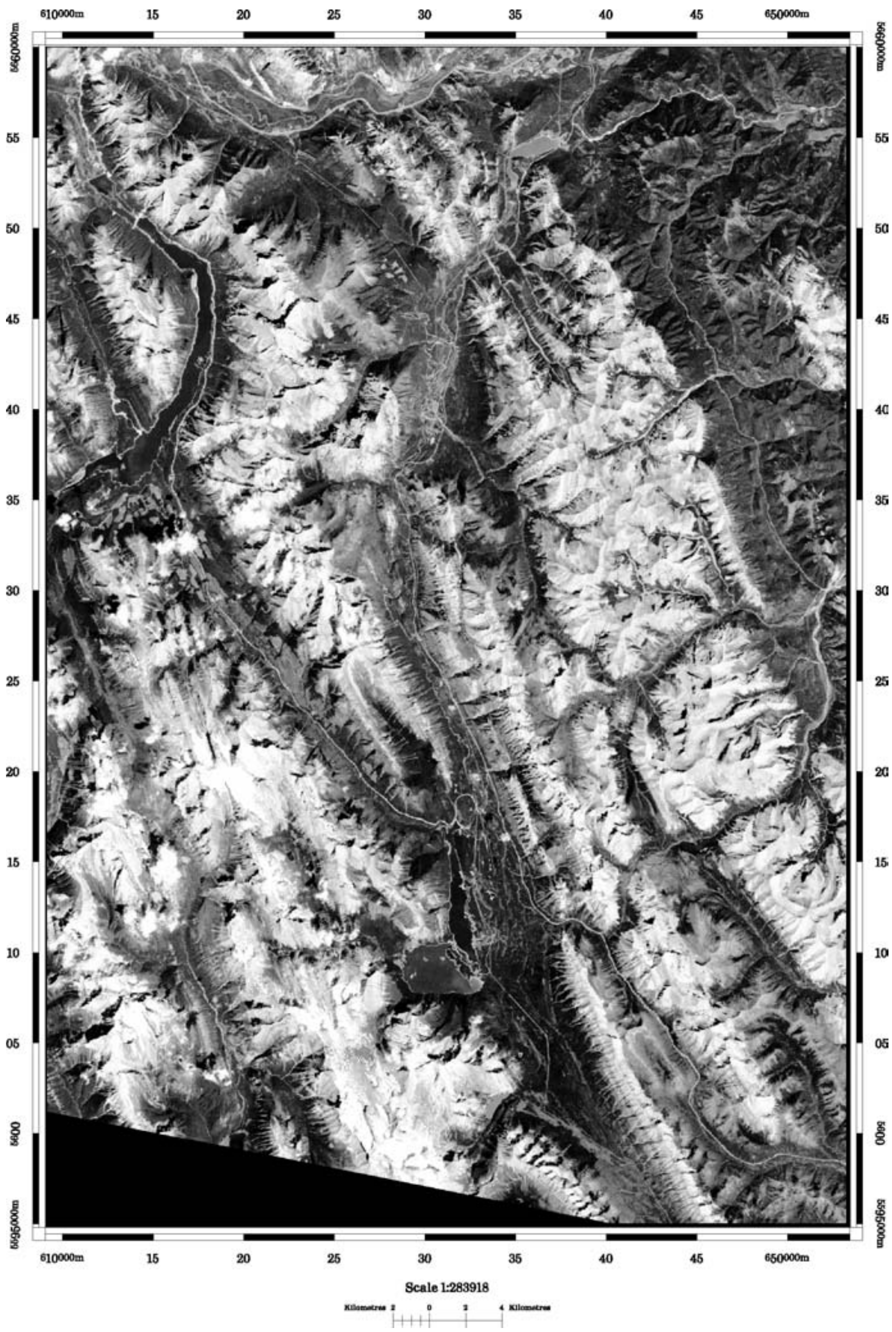

Figure 3.2: Landsat-TM image of study area using the blue, green and red electro-optical channels.

Landsat-TM consists of 7 spectral bands as outlined in table 3.1 below. The remotely sensed data type most often applied in ecological studies is electro-optical reflectance. Therefore, channels 1 to 3 were employed as the inputs to the RS network's feature set. Channels 4, 5 and 7 were also used; they are reflected infrared channels. Channel 6 was omitted from this feature set; it is not a reflectance channel and has a different resolution. This thermal infrared channel is useful in measuring such things as soil moisture and plant heat stress, but is not as useful in distinguishing between vegetation types as the electro-optical channels are (Campbell, 1996).

Channel	Wavelength (μm)	Band	Resolution (m^2)
1	0.45 - 0.52	blue/green	30 x 30
2	0.52 - 0.60	green	30 x 30
3	0.63 - 0.69	red	30 x 30
4	0.76 - 0.90	near IR	30 x 30
5	1.55 - 1.75	mid IR	30 x 30
6	10.40 - 12.50	thermal IR	120 x 120
7	2.08 - 2.35	mid IR	30 x 30

Table 3.1: Characteristics of Landsat-TM's spectral bands.

Caution should be exercised when using data recorded by spaceborne optical sensors. These sensors not only record the solar radiance reflected at the Earth's surface, but also the radiance scattered by the atmosphere, as depicted in figure 3.3. It has been suggested that improvements in land cover classifications can be achieved by atmospherically correcting the data under study (Kaufman, 1984). PCI Geomatics' EASI/PACE® Atmospheric Correction package was therefor applied to the Landsat-TM feature set (PCI, 1997). Appendix B outlines the settings of the atmospheric correction process employed.

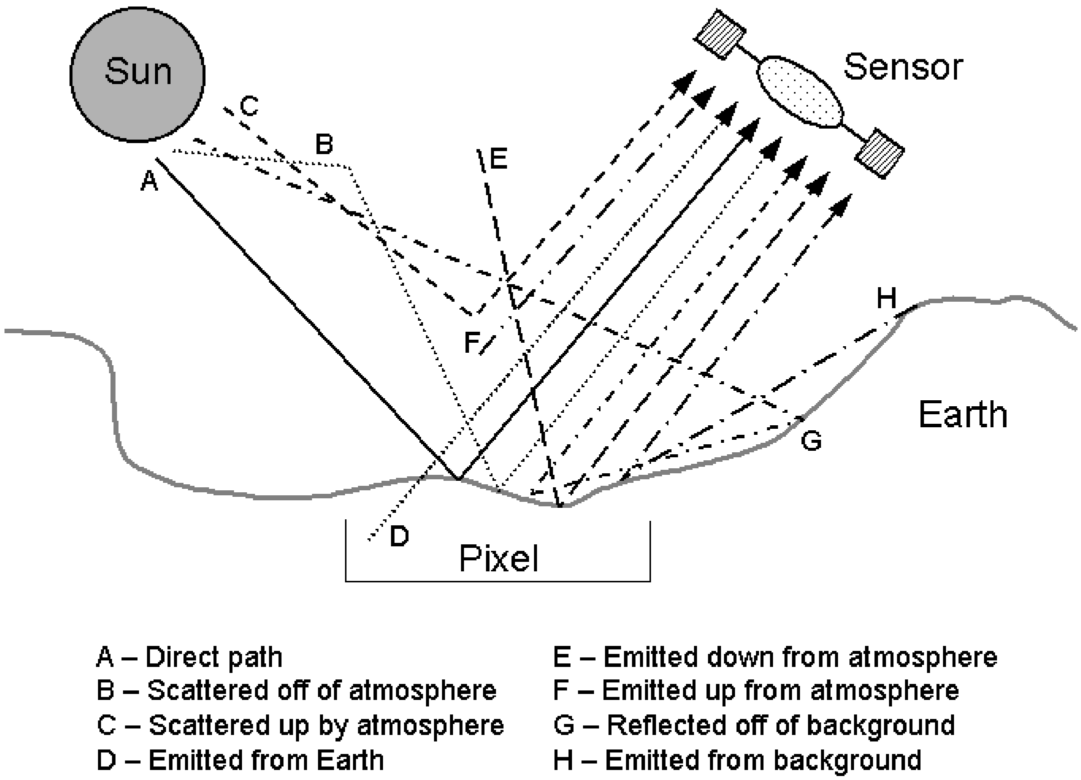

Figure 3.3: Possible solar radiance paths from the sun, atmosphere, Earth and background to the sensor.

This feature set also provides valuable information regarding the spectral overlap of the output classes. Vegetative classes are very difficulty to distinguish from one another due to their reflectance overlap. In figure 3.4, Collins *et al.* (2001) plotted +/- one standard deviation of each channel's mean and interpolated the data using a cubic spline. The figure clearly depicts the output classes' overlap. Aspen is the only vegetative class distinguishable from the others on the basis of its channel 4 near infrared measurements. However, it will be easily confused with the non-vegetative alpine class. The non-vegetative classes are more distinguishable from one another, especially the two water classes, except for rock which covers a wide range of reflectance values (Collins et al., 2001).

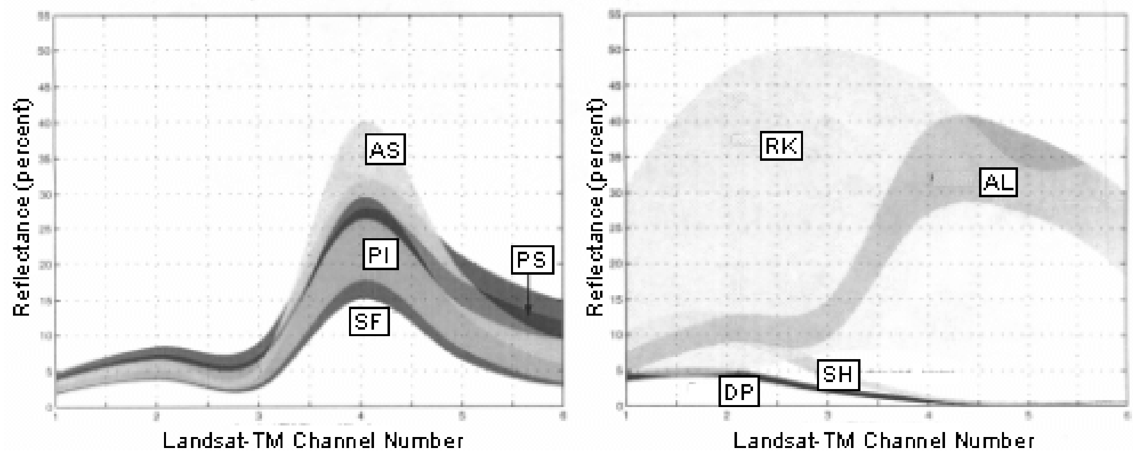

Figure 3.4: Reflectance overlap of vegetative (aspen (AS), pine (PI), pine/spruce (PS) and spruce/fir (SF)) and non-vegetative (alpine (AL), deep water (DP), rock (RK) and shallow water (SH)) classes, source: Collins *et al.* (2001).

The remotely sensed feature set is being employed in this research because it provides observational information regarding the area's vegetative and non-vegetative coverings. It will now be referred to as the Landsat-TM feature set and the ANN classifying it as the LTM network. LTM's corresponding input vector is denoted by,

$$X_i = \begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_t \end{bmatrix} = X_{LTM} = \begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_6 \end{bmatrix}, \tag{3.1}$$

where $x_1, ..., x_6$ are channels 1 to 5 and 7's reflectance percentages respectively, scaled to a [0,1] range. This reduces network computational time by simplifying the mathematics involved.

3.4 Geomorphology feature set

The second feature set employed is an indirect environmental gradient - elevation. A digital elevation model (DEM) is a representation of the Earth's topography using coordinates and altitude values. The Landsat-TM image of figure 3.2 was registered to the DEM by overlapping the two and matching their points to correspond with ground coordinates. The DEM had an accuracy of 5 m and was resampled from 25 m to 30 m.

A DEM can be used to estimate other indirect gradients such as slope and aspect. Slope describes a pixel's upward or downward slant; aspect describes its geometric orientation in the horizontal plane. Every pixel's slope and aspect were estimated using algorithms described in Guth (1995) by Dr. Michael J. Collins of the University of Calgary's Geomatics Engineering Department (Guth, 1995).

The indirect gradient feature set is being employed in this research because it provides localised site descriptions of the environment in which the vegetation grows. It will now be referred to as the geomorphology feature set and the ANN classifying it as the GEO network. GEO's corresponding input vector is denoted by,

$$X_{GEO} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \tag{3.2}$$

where x_1, x_2 and x_3 are elevation, slope and aspect respectively, scaled to a [0,1] range.

3.5 Process model feature set

The final feature set employed is a resource gradient developed by Caren Dymond of the University of Calgary's Biological Sciences Department. The two most important vegetation resources are light/energy and moisture. In Dymond (1998), species tolerance is redefined using energy and moisture processes and is performed on the same data set utilised in this research. These models' parameters were termed available energy and available moisture. They were estimated using two very simple process models which employed elevation, slope and aspect as inputs. A photosynthesis model scaled the relative impacts of solar radiation and temperature into an available energy variable, while a distributed hydrological model employed climate water flux due to elevation for the available moisture variable (Dymond, 1998). Readers are referred to Dymond (1998) for a full mathematical description of these two resource gradient variables.

The resource gradient feature set is being employed in this research because its variables directly effect plant growth. It will now be referred to as the process model feature set and the ANN classifying it as the PM network. PM's corresponding input vector is denoted by,

$$X_{PM} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{3.3}$$

where x_1 and x_2 are available energy and available moisture respectively, scaled to a [0,1] range.

Chapter 4

Experimental Design and Methodology

4.1 Introduction

In any research, it is necessary to fully explain and justify how results were obtained, which in turn supports the conclusions drawn. This research is divided into three components - classification of data by ANNs, combination of ANN classification results, and comparison and analyse of all results achieved. This chapter outlines the design and methods involved with these components by thoroughly explaining and justifying all experimental parameters and approaches employed.

4.2 Classification by artificial neural networks

Several advantages regarding the use ANNs for classification problems were outlined in §1 - from their tolerance of outliers within the feature set, to their ability to classify multiple feature sets of differing and unknown distributions. Unfortunately, there is one main disadvantage to using ANNs and it brings forth many other disadvantages. There are no set rules regarding a network's structure and parameter settings, such as initialising weight values, choosing a proper transfer function, selecting an appropriate learning algorithm, deciding on the number of hidden layers and nodes, using a valid training set and of adequate size, etc... Should the initial weights be randomly chosen or set equal to a pre-determined value? There are various suggestions for setting the initial weights and biases however, choosing the method most appropriate for the problem at hand could prove to be difficult. Is BP the most appropriate learning algorithm for the problem at hand? BP is the most commonly applied network learning algorithm when using remotely sensed data. In Tenorio *et al.* (1990) however, a SONN needed considerably less time to train than a BPNN. Should one use a 2- or 3-layered ANN? How many nodes per hidden layer should be included? If a large and complex network is used, there is a risk of overtraining the network - the training data will be learned so well that the network will not be able to properly generalise once presented with data it has never seen. However, if the network is too small, poor generalisation might also occur due to the fact that the network has not learned enough about the training data to properly classify new data. How many data points should one use to train the ANN? Which pixels are better for training? The quality of the training set is very important; it is desired to use representative data to train the network. It can be quite costly to acquire data points which are representative of the classes under study. Some good news arises when wondering how large a training set should be. Conventional classifiers usually need a large training set in order to produce good classification results when tested. The opposite has been shown for ANNs; smaller training sets have provided accurate classification results. However, there are still no set rules on the exact amount of training points to be used. In fact, if the training set is too large, a risk of once again overtraining the network is highly likely. All these factors make it difficult and time consuming to determine the proper network structure to use for every particular problem (Atkinson & Tatnall, 1997; Li et al., 1993; Paola & Schowengerdt, 1995; Zhuang, Engel, Lozano-Garcia, Fernández, & Johannsen, 1994).

4.2.1 Single versus multiple feature set networks

The feature sets from §3.3 to §3.5 have been partitioned into three single feature set networks: LTM, GEO and PM. They were partitioned to be classified by different networks as they are fundamentally different types of data. As previously mentioned, the Landsat-TM feature set provides observations of the land cover classes, the geomorphology feature set provides information regarding the environment in which these classes grow and finally, the process model feature set directly effects vegetation growth. It is reasonable to use different networks to classify these feature sets due to their different natures.

Once individual networks have classified their specific feature set, there are many ways in which the classification results can be combined. The combination methods presented in §2.4 outline several algorithms for combining multiple ANNs. Two advantages of joining multiple network results are the ability to merge data of differing natures, and the ability to combine any number or type of network as long as they produce likelihood estimates.

However, a simpler method of integrating several feature sets is to stack them and use a larger ANN. For example, stacking the LTM and GEO feature sets together would produce the LTM/GEO network with the following input vector,

$$X_{LTM/GEO} = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_8 \\ x_9 \end{bmatrix}, \tag{4.1}$$

where x_1 to x_6 are the six Landsat-TM features and x_7 to x_9 are the three geomorphology features. Adding data in such a fashion increases the dimensionality of the network's input vector, making the network larger and more complex. This type of network is referred to as a multiple feature set ANN as it stacks more than one feature set in its input space. Two more networks of such a nature will be analysed in this research - LTM/PM and LTM/GEO/PM. LTM/PM's corresponding input vector is denoted by,

$$X_{LTM/PM} = \begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_7 \\ x_8 \end{bmatrix}; \tag{4.2}$$

while LTM/GEO/PM's is denoted by,

$$X_{LTM/GEO/PM} = \begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_{10} \\ x_{11} \end{bmatrix}. \tag{4.3}$$

The LTM/PM network stacks the six Landsat-TM features and the two process model features, while the LTM/GEO/PM network stacks all feature sets in question. Which method - combining network results or stacking feature sets - provides better land cover classification accuracies? This research investigates both methods and compares all results in order to provide an answer to this question.

4.2.2 Approach to problematic pixels

A pixel is considered to be a pure pixel when it belongs to a single class. For example, imagine having a satellite image of the Atlantic Ocean which is being classified to one of three output classes: rock, water or tree. The image pixels obviously belong to the water class; they cannot possibly belong to more than one of the three classes at a time. This satellite image therefore contains pure water pixels. Mixed pixels differ from pure pixels in that they belong to more than one class. For example, if a pixel depicts 25 km^2 of the Earth's surface, it will most probably contain water, trees and rocks. If one were to make these classes even more specific i.e., spruce trees, fir trees, deep water, shallow water, the image would contain many mixed pixels.

How should mixed pixels be classified? One way is to categorise the mixed pixel to its most representative output class. Assume that a mixed pixel composed of 95% spruce trees and 5% fir trees is being classified. It would be best to categorise this pixel as a spruce pixel rather than a fir pixel. Another option is to use classes which represent the various possible mixed situations. Assume that a second mixed pixel is known to be composed of 45% spruce trees, 50% fir trees and 5% rocks. Should this pixel be classified as a spruce pixel or as a fir pixel? Neither, it would be best if there was a class designated to represent such pixels containing high percentages of both types of trees such as a spruce/fir class. If properly trained, a network should be able to categorise mixed pixels to their best representative class. During training, the network's parameters are adjusted in order to accommodate any number of output classes as long as it is trained with pixels representative of each output class (Murai & Omatu, 1997; Omatu & Yoshida, 1993).

Some pixels can be quite troublesome to classify and should be eliminated from the network's training and testing processes. Omatu & Yoshida (1993) applied the Kohonen's algorithm to select their training pixels. Kohonen's algorithm produces cluster centers for each output class such that the density function of the centers approximates the probability density function of the inputs (Omatu & Yoshida, 1993). Once their training pixels were chosen in such a fashion, Omatu & Yoshida (1993) applied them to their ANN resulting in a small mean squared error of 0.00956. However, they also tried to eliminate some of this error by removing pixels from the training set which were not properly classified during the first training iteration. Once completed, the network was retrained with this new set of pixels. The mean squared error was reduced to a value of 0.000819. Other forms of pixel preprocessing have been shown to reduce training time as well. Choosing training sets using a K-means clustering algorithm reduced network training time significantly when compared to the training times of randomly chosen training sets (Murai & Omatu, 1997). If certain pixels are continuously being misclassified by the network, one should consider the following questions and solutions in order to help rectify the problem:

- Is the pixel truly representative of its output class? If no, reclassify the pixel to an appropriate output class or remove the pixel from analysis completely if no such class is determined.

- Has the pixel's raw data been recorded correctly? If no, attempt to acquire new raw data but if this is impossible, remove the pixel from analysis completely.

- Has the feature set's scaling process been performed properly? If no, rescale all data.

- Should the pixel be considered a mixed pixel? If yes, reclassify the pixel to the appropriate mixed output class as well as other pixels that may be of such a mixed nature.

In this research, almost every shallow water pixel was consistently being misclassified during the training and testing phases. At first, it was assumed that the various networks' architectures and other parameter settings were incorrectly chosen. Several unsuccessful adjustments were made to these parameters in an attempt to properly classify these problematic pixels. Next, the DN values for each shallow water pixel were reviewed. Several pixels were found to have reflectance values not representative of shallow water or of any other output class in this research. They were therefore removed from any further analysis leading to significant improvements in network training and testing classifications.

Problems were also encountered when classifying mixed pixels composed of pine and spruce trees - pine/spruce pixels. A large number of pine/spruce pixels were continuously being classified as pine pixels. Once again, the networks' architectures and other parameter settings were examined and altered without improving classification accuracies. The pine and pine/spruce pixels' Landsat-TM feature sets were then analysed, but both classes have very similar signatures and are therefore very difficult to separate. The actual number of pine and spruce trees present in each pixel were then reviewed. It was discovered that several pixels labelled as pine/spruce pixels were in fact pure pine pixels. The count of pine trees in those pixels outnumbered the spruce trees' count significantly. These pixels were then recategorised as pine pixels. The incorrect labelling of these pixels lead to poor classification accuracies. Once these pixels were properly relabelled, improvements in the classification of the pine/spruce pixels were recorded. However, these pixels are still very difficult to differentiate from the pure pine pixels which is apparent in the final classification results.

In both of the above situations, the original hypothesis of having poor network architecture was incorrect. The poor classifications resulted from the networks being trained with unrepresentative data of their output classes.

4.2.3 Design of network architecture

The following paragraphs outline the various architecture parameters employed by the various ANNs in this research.

Initialisation: All weights and biases were randomly set to initial values within [-0.5,0.5] range. Smaller ranges of [-0.25,0.25] and [-0.15,0.15] have been suggested for RS classification problems (Kavzoğlu, 2001). These ranges were tested without resulting in significant changes in the final classification results. The classifications resulting from randomly initiated weights and biased also proved to be significantly better than setting all initials weights and biases equal a certain constant such as 0 or 1.

Hidden layers: A network without any hidden layers, only an input and an output layer, can classify data separable by a half plane bounded by a hyperplane. A single hidden layer network can form convex open or closed regions around the data to be classified. By using a network with two hidden layers, any form of a decision region can be created (Bishop, 1995). It is suggested in Paola & Schowengerdt (1995) that a single hidden layer is sufficient for classifying remotely sensed data because "the class distributions do not usually consist of disconnected or other non-convex regions" as formed by a two hidden layer network. Kanellopoulos & Wilkinson (1997) agree with their statement that a single hidden layer network is appropriate for most RS classification problems. A two hidden layer network should be used if the number of classes nears 20, allowing for the additional flexibility that such a complex problem requires (Kanellopoulos, Varfis, Wilkinson, & Mégier, 1992). For these reasons, all ANNs employed in this research consisted of a single hidden layer, otherwise referred to as a 2-layered ANN.

Hidden layer nodes: Various different approaches for determining the number of nodes per hidden layer were investigated. The suggestions made in Kanellopoulos & Wilkinson (1997) were implemented when choosing the number of nodes in the various networks' hidden layer. If a two hidden layer network is used, the first layer should have at least twice as many nodes as the number of inputs and perhaps even four times as many. The second hidden layer should have two to three times as many nodes as the number of output classes. If a single hidden layer network is used, the number of nodes should equal the greater of the values that would have been computed for a two hidden layer network (Kanellopoulos & Wilkinson, 1997). This resulted in 24 hidden layer nodes for the LTM, GEO and PM networks, 36 for the LTM/GEO network, 32 for the LTM/PM network, and 44 nodes in the LTM/GEO/PM network's hidden layer.

Transfer functions: A simple logistic sigmoid function, as described in equation (4.4), transferred all hidden layer node's weighted sum to the output layer. As mentioned in §3, all feature sets were scaled to a [0,1] range. A sigmoid function outputs values within the same range and was therefore chosen as the transfer function for all ANNs. The linear function $f(x) = x$ was employed at the output layer because it does alter the nodes' computed weighted sum; it simply produces the final classification values.

$$\text{logsig}(x) = \frac{1}{1 + e^{-x}} \tag{4.4}$$

Output layer nodes: The number of nodes in the output layer was originally set to equal 9 - 8 output classes and 1 unknown class. None of the ANNs were classifying any pixels to the unknown class which was to be expected. The networks were not trained with data representative of the unknown class and therefor, all pixels were classified to one of the 8 output classes. These classes consisted of four vegetative classes: aspen, pine, pine/spruce and spruce/fir, and four non-vegetative classes: alpine, deep water, rock and shallow water. The output vector of estimated a posteriori class probabilities for network i is denoted by

$$Y_i = \begin{bmatrix} y_i(c_1) \\ y_i(c_2) \\ \ldots \\ y_i(c_8) \end{bmatrix}. \tag{4.5}$$

The notations and abbreviations associated with each output class in this research are: c_1 - Aspen (AS), c_2 - Pine (PI), c_3 - Pine/Spruce (PS), c_4 - Spruce/Fir (SF), c_5 - Alpine (AL), c_6 - Deep Water (DP), c_7 - Rock (RK), and c_8 - Shallow Water (SH).

The PM network's architecture was analysed closely due to its poor training results. However, changing the number of its hidden layers and nodes did not significantly change training accuracy. These parameters were therefore left to equal their original values as described above.

4.2.4 Network training

As mentioned in §2.2, an ANN is a system of interconnected processing nodes that is first trained, then tested on the data to be classified. There are two types of training: supervised and unsupervised. Supervised training occurs when the network is trained with data for which its true classification is known. Unsupervised training is the direct opposite; the true classification of the data to be classified is not known. All networks were trained using a supervised training algorithm. The training set for all networks consisted of 4153 pixels. The objective of training is to adjust the weights and other network parameters in order to "optimise" the ANN's performance over the entire feature set of interest. A network is deemed optimised when the researcher is satisfied with the classification results achieved. Once a network has been trained, it is tested. Testing consists of processing the testing data once through the network with the weights determined during the training phase. The testing data set can consist of new data as well as some, all or none of the data used during training. The networks in this research were tested on a data set which had not been used during training and consisted of 885 pixels. The training and testing data sets were selected previously in Dymond (1998). The same training and testing data sets were employed in this research so that the results obtained here could be compared to a study performed by Collins *et al.* (2001). Patternwise training refers to updating weights after each pixel from the training set has been processed through the network, while epochwise training refers to updating weights only after a certain number of pixels from the training set have been processed. The number of pixels to be processed before weights are updated is referred to as the epoch size. Due to significant training time increases using the patternwise method, epochwise training was applied in this research. The epoch size for all networks equalled the number of pixels in the training set; every pixel in the training set was processed through the network once before weights were updated. All training pixels were presented in the same random order to all networks. Randomising the order of which a network views the training pixels prevents it from learning one type of pixel better than another.

Output Class	Number of Training Pixel	Number of Testing Pixels
Aspen	62	12
Pine	139	28
Pine/Spruce	94	26
Spruce/Fir	118	25
Alpine	569	92
Deep Water	1397	328
Rock	1171	250
Shallow Water	603	124
Total	4153	885

Table 4.1: Number of training and testing pixels per output class.

As mentioned in §2.2, BP is one of the most commonly applied training processes when analysing remotely sensed data with ANNs. BP is a steepest descent training algorithm which moves in the direction of the negative gradient because the performance function decreases most rapidly in this direction. Due to its wide use in RS and its simplicity, BP was utilised by all networks in this research with the addition of a momentum term. The momentum helps prevent the training algorithm from getting stuck in a local minima of the performance function by allowing past weight changes to influence current changes. The learning rate determines how fast the network will learn the problem at hand. If the learning rate is too large, the network will learn too quickly and will not be able to properly generalise whereas if the learning rate is too small, the network will take too long to converge (Bishop, 1995). The momentum and learning rate parameters were originally set equal to 0.9 and 0.1 respectively. This is a good starting point for these two parameters and could be considered their standard values. They can then be easily increased and/or decreased with respect to network performance. Different learning rate can also be employed for the different layers in a network. In this research, changing these values did not result in significant classification accuracy improvements. They were therefore returned to the original values mentioned above. The BP software (Dyn_BpNN) employed was developed by Mr. Kevin Croft of the University of Calgary's Geomatics Engineering Department. This software is based on Pao's (1989) training algorithm. Appendix C outlines the mathematical process of the BP training algorithm.

Choosing how long to train a network is always an interesting challenge when working with ANNs. There are various suggestions for choosing the number of training iterations, including stopping once the network's MSE falls below a preset value (Bishop, 1995). This research concentrated on each output class's conditional kappa (κ_j) performance during training. Training was stopped once the classes no longer improved their training classification accuracies. By simply looking at the MSE graph for each network, it is very difficult to determine when the network should stop training. Should training stop when the network's MSE value falls below 0.05 or 0.01? It would be very difficult for one to justify his/her reasoning for stopping the training process. However, the use of class-conditional kappa suggests an objective method for stopping: statistical significance. In figures 5.1 to 5.6, a black symbol is located at a training iteration which resulted in a class-conditional kappa value statistically better than the previously symbolled training iteration. For example, figure 5.1 depicts that the 250th training iteration resulted in the pine/spruce class's conditional kappa value to be statistically better than the value achieved after 125 iterations. In turn, pine/spruce's class-conditional kappa value after 125 iterations was statistically better than the first computed value at 25 iterations. It is very easy and clear to monitor the network's improvements in classification accuracies by using a graphical representation of each class's conditional kappa values.

In order to determine the number of training iterations for each network, the class-conditional kappa values will be computed every 25 iterations. The values will be compared to one another, noting which iterations have class-conditional values statistically better than previous iterations. Training will be stopped based on the following considerations:

- whether or not there had occurred improvements in any of the class-conditional kappa values over 200 iterations,
- whether or not the class-conditional kappa values were starting to stabilise and level off towards some specific value, and
- whether or not improvements were seen in the vegetative classes due to their difficulty in being properly classified.

4.3 Combination of classification results

Once all networks are trained and tested, their classification results will be combined using the various methods outlined in §2.4. There are two types of combinations that will be encountered in this research. They consist of combining either three or two ANNs' results. The combinations of three networks will consist of LTM's, GEO's and PM's results, while combinations of two networks will consist of LTM/GEO's and PM's results or LTM/PM's and GEO's results. These final combined classification results are the main target of this research. A full presentation of all results is given in §5, followed by a thorough discussion in §6.

4.4 Analysing results

It is of interest to know whether or not the classification results achieved by the ANNs and combination methods are statistically equivalent or better than one another. Once all networks have been tested and their results combined, the many overall and class-conditional kappa values will be compared in order to determine whether using combinations of smaller networks results in better classification accuracies than using a single large network. The test statistic that will be employed for these comparisons is the Z-score (Z). It is computed using the following equations.

$$Z_K = \frac{|K_i - K_I|}{\sqrt{\sigma^2_{K_i} + \sigma^2_{K_I}}} \tag{4.6}$$

computes the Z-score comparing networks i's and I's overall kappa values, while

$$Z_\kappa = \frac{|\kappa_j - \kappa_k|}{\sqrt{\sigma^2_{\kappa_j} + \sigma^2_{\kappa_k}}} \tag{4.7}$$

computes the Z-score comparing class j's and k's conditional kappa values (Congalton & Green, 1999). In equations (4.6) and (4.7), the overall and class-conditional kappas' variances are denoted by $\sigma^2_{K_i}$ and $\sigma^2_{\kappa_j}$ respectively. The reader is referred to Congalton & Green (1999) for the mathematical formulas of these variables.

4.5 Research experiments

First, three single feature set networks will be constructed - LTM classifying the remotely sensed feature set, GEO classifying the geomorphology feature set and PM classifying the process model feature set. Next, three multiple feature set networks will be constructed - LTM/GEO classifying the remotely sensed and geomorphology feature sets, LTM/PM classifying the remotely sensed and process model feature set and finally, LTM/GEO/PM classifying all three features sets under study. Their architectures will be determined based on the information provided in §4.2.3. All networks will be trained using the parameters and methods described in §4.2.4. Once training is completed and weights determined for each network, the ANNs will be tested on their associated testing sets. All testing classification results will then be recorded and presented in confusion matrices providing a visual representation of the number of correctly classified pixels in each output class. The confusion matrices' results will be used to computed overall and class-conditional kappa values as described in §2.5. Next, the various network classification results will be combined using the methods presented in §2.4. The three combinations being performed are LTM + GEO + PM, LTM/GEO + PM and LTM/PM + GEO. These combined classification results' accuracies will then be assessed by computing their overall and class-conditional kappas. These kappa values will then be compared to the individual networks' values that they combine at a 95% confidence level. This will determine which combination methods performed better than the individual networks being combined and whether or not combining smaller networks performed better than using a single large network. Finally, using the kappa values computed for the various combination methods, a comparison of these methods will be performed in order to determine the better method for combination ANNs classifying remotely sensed data.

4.6 Methodology justification

The design and methods utilised in this research were chosen so as to construct simpler networks able to accurately classify various feature sets leading to improvements in canopy cover classifications through combinations of these various networks' results. Focus was placed on the simplicity, speed and accuracy of classifying data with ANNs as well as the advantages, practicality and versatility of reaching a consensus decision rather than using a single "best" solution.

Chapter 5

Results

5.1 Introduction

The following chapter outlines the results produced by this research. These results will be discussed in context of other published research in the next chapter, §6. It should be noted that the output classes' abbreviations employed in these chapters are outlined in §4.2.3 and the combination methods' abbreviations are described in table 5.8.

5.2 Class-conditional kappas and training

In figures 5.1 to 5.6, each output class's conditional kappa value is graphed over every 25 training iterations. The vegetative classes are depicted by solid lines while the non-vegetative classes are dotted lines. The dashed black line represents the network's MSE value over all training iterations.

Figures 5.1 to 5.6 are now presented following which, a full discussion regarding each network's number of training iterations is addressed. It should be noted that although the PM network was trained for 150 more iterations than the LTM/GEO/PM network, its training time in minutes was significantly less. Network training time in minutes seemed to increase exponentially with the size of the network. The PM network needed approximately 10 minutes to process 500 training iterations whereas the

LTM/GEO/PM network required approximately 30 minutes for the same number of iterations. The LTM/GEO/PM network had 9 more inputs and 20 more hidden layer nodes than the PM network, making it a larger and more complex ANN. In fact, the LTM/GEO/PM network took the longest time in minutes to train of all networks. It should be noted that all analysis was performed on an IBM Aptiva personal computer with a Pentium II processor at 400 MHz using 128 MB RAM.

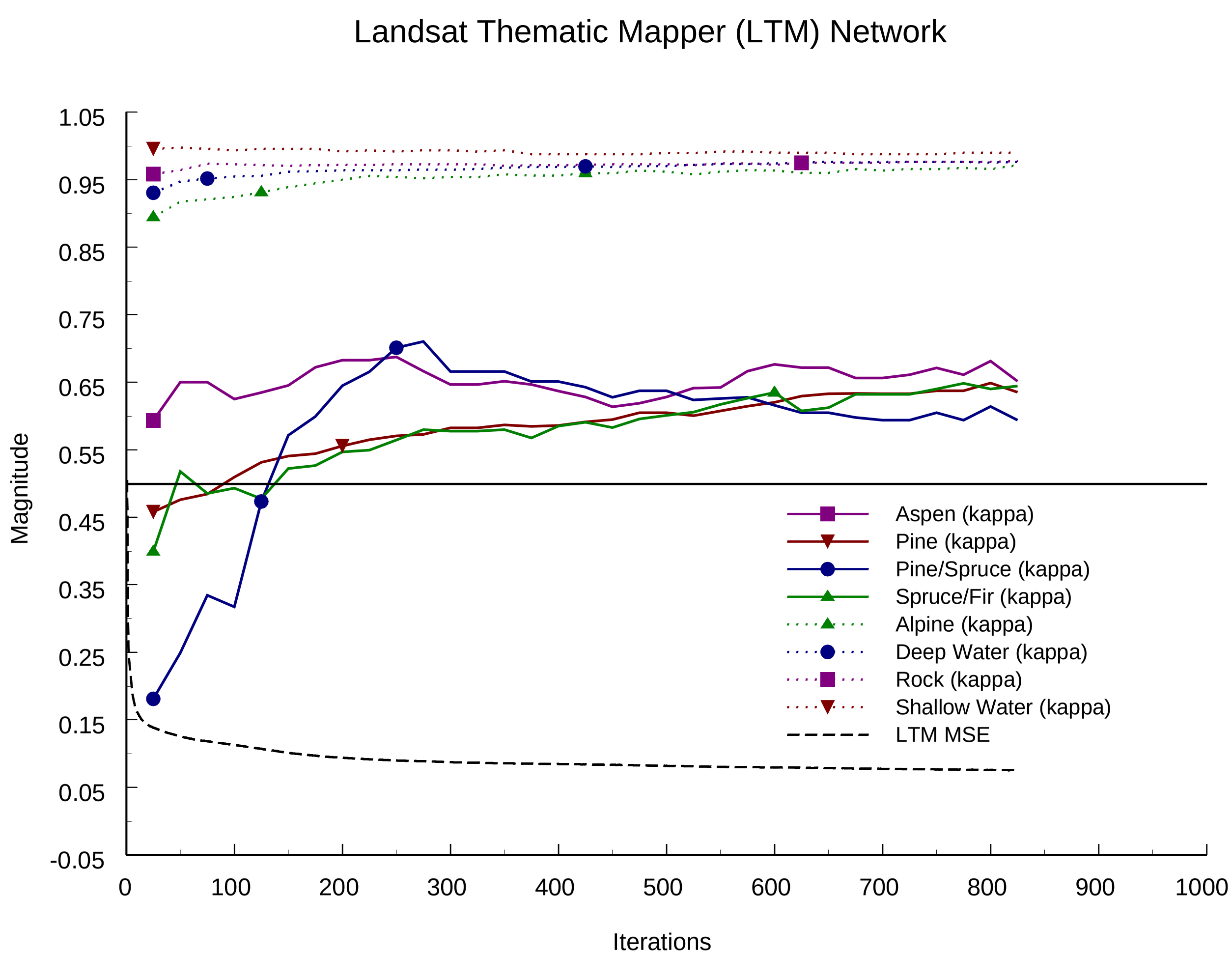

Figure 5.1: Class-conditional kappa (kappa) and mean squared error (MSE) values for the LTM network after every 25 iterations.

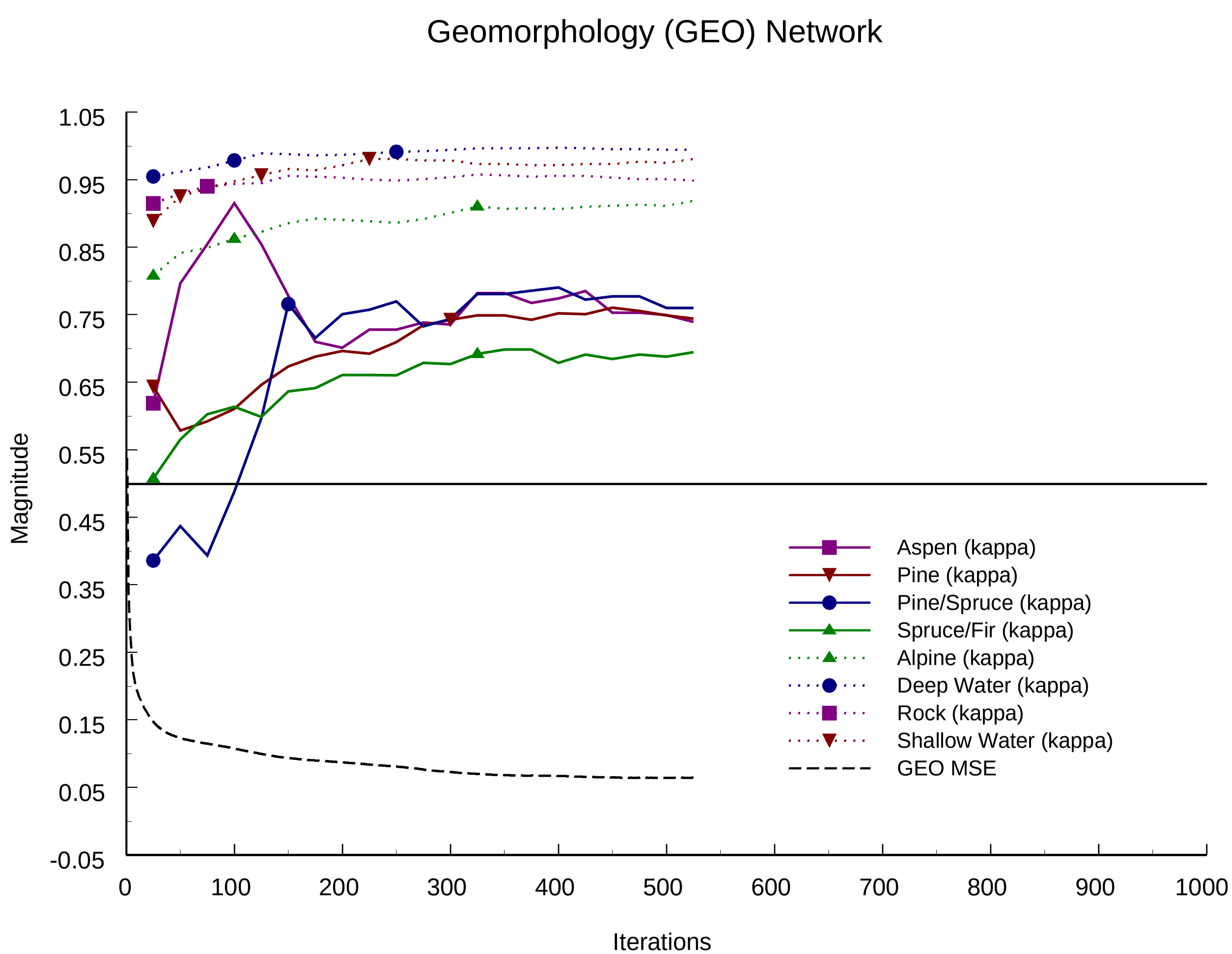

Figure 5.2: Class-conditional kappa (kappa) and mean squared error (MSE) values for the GEO network after every 25 iterations.

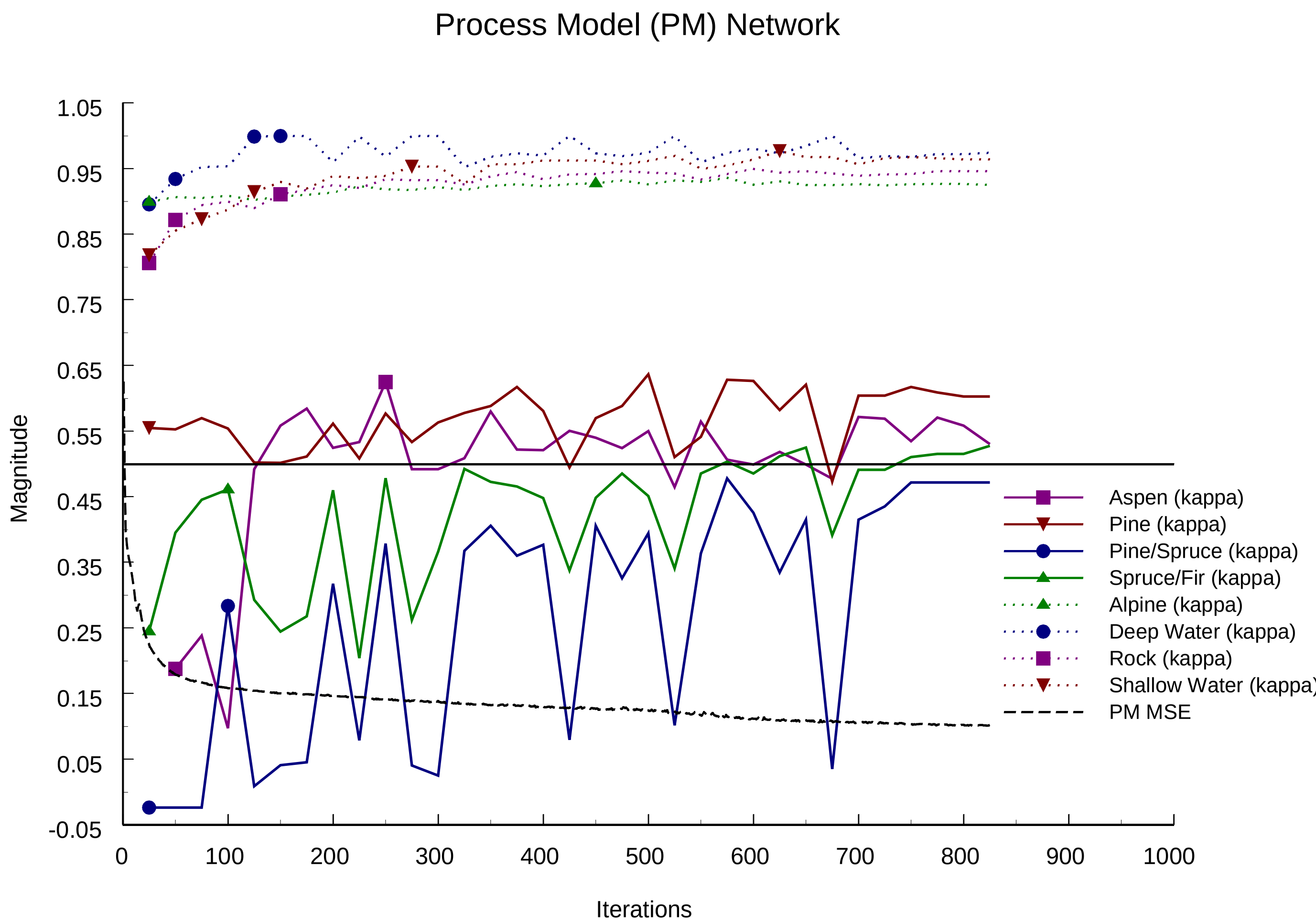

Figure 5.3: Class-conditional kappa (kappa) and mean squared error (MSE) values for the PM network after every 25 iterations.

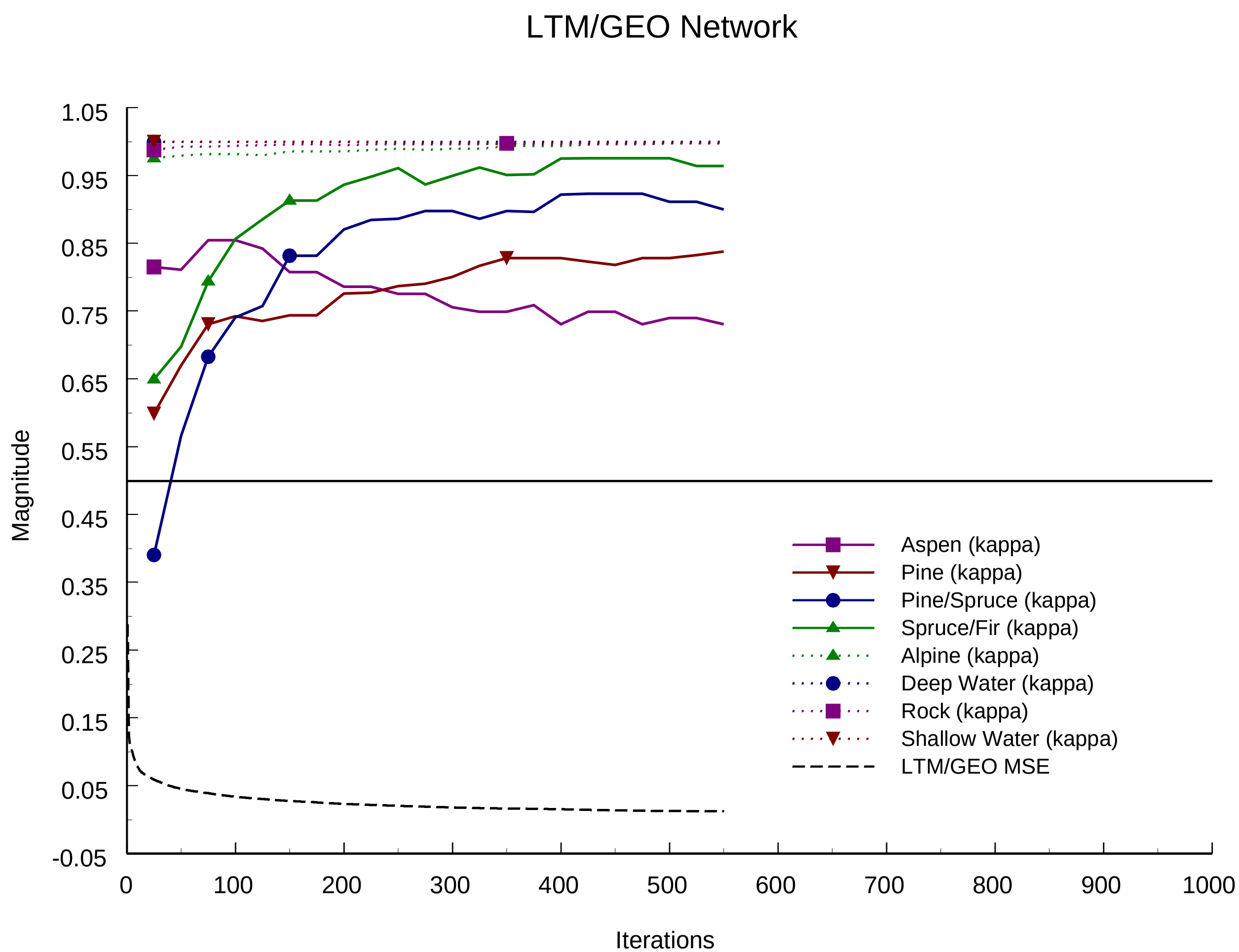

Figure 5.4: Class-conditional kappa (kappa) and mean squared error (MSE) values for the LTM/GEO network after every 25 iterations.

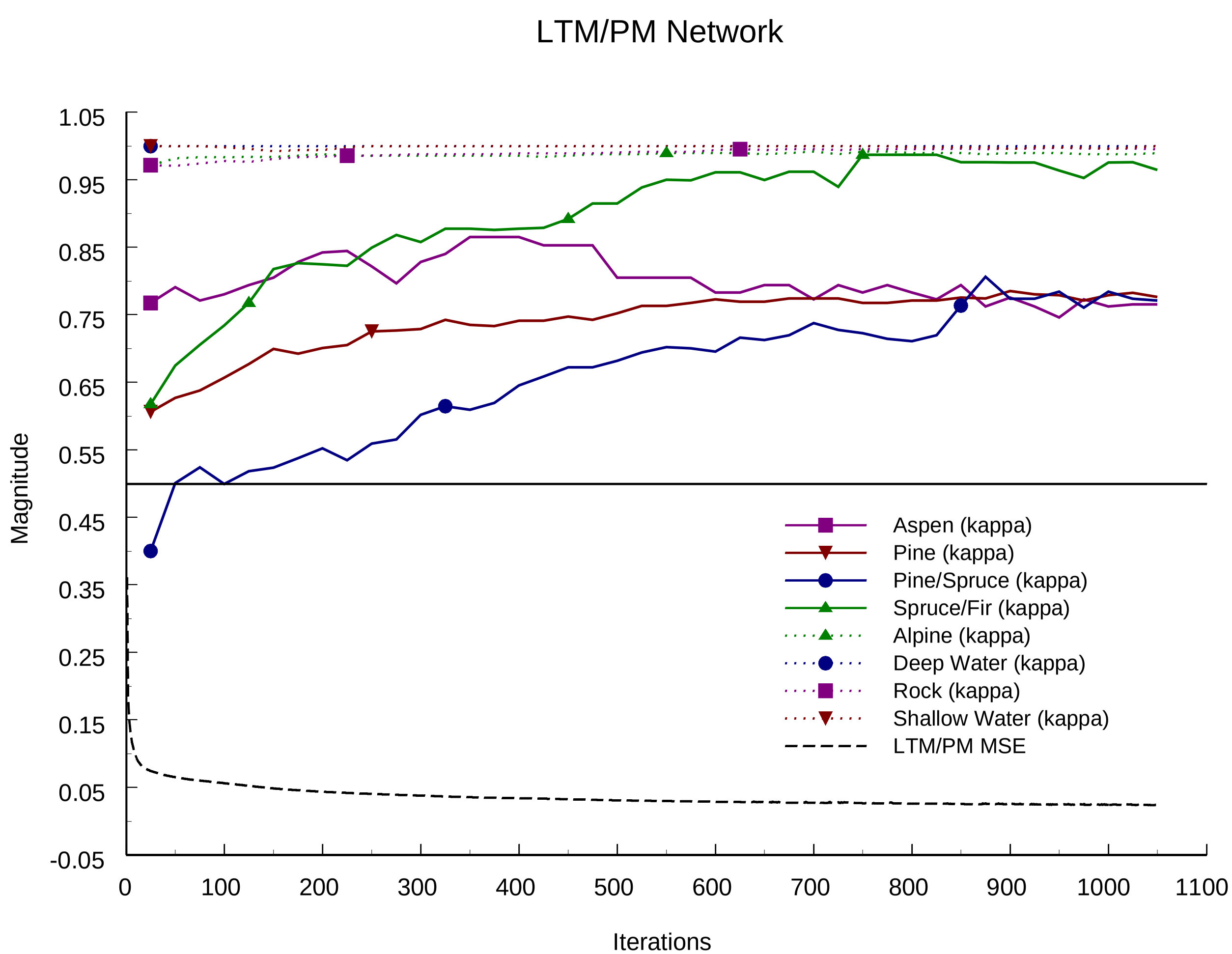

Figure 5.5: Class-conditional kappa (kappa) and mean squared error (MSE) values for the LTM/PM network after every 25 iterations.

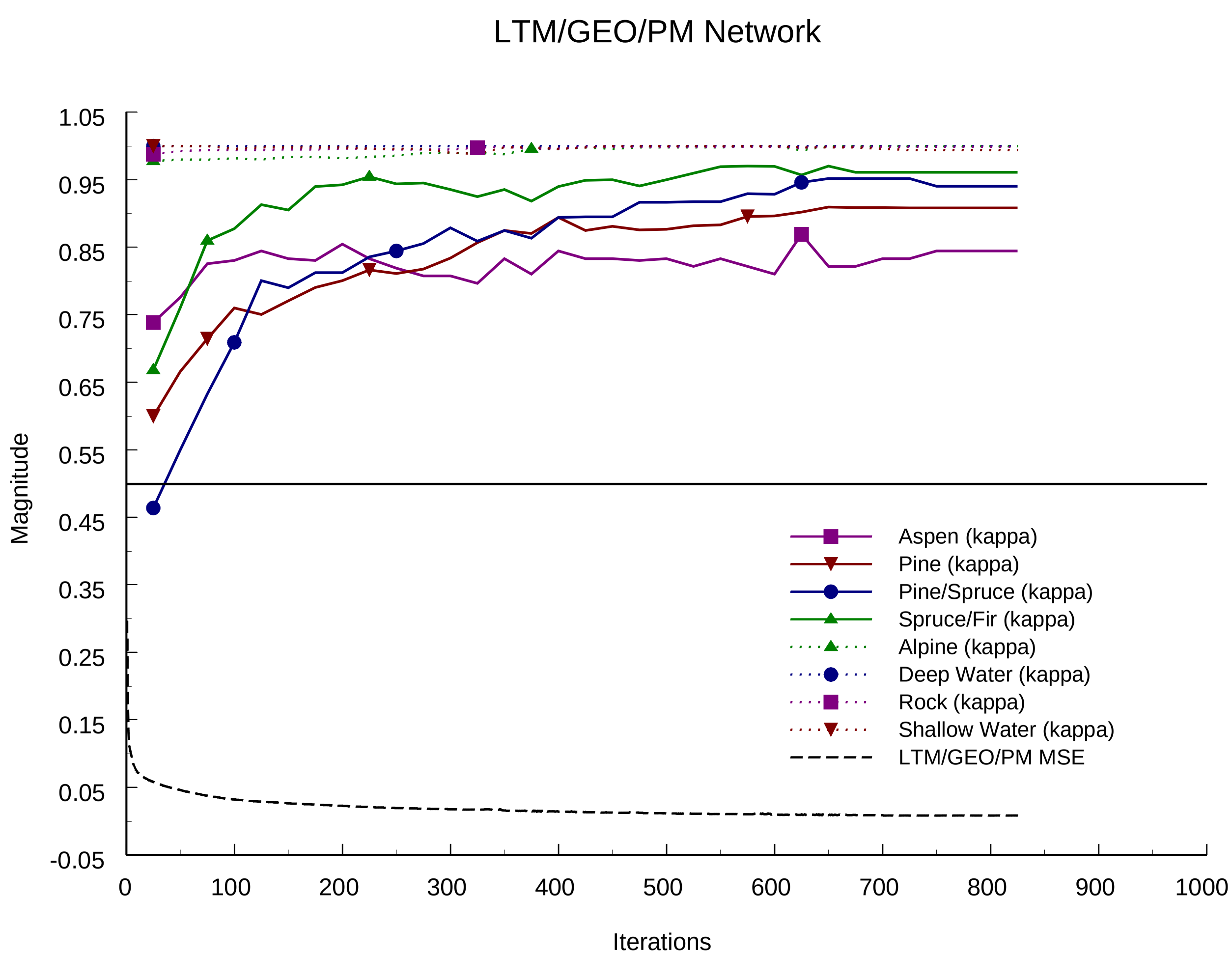

Figure 5.6: Class-conditional kappa (kappa) and mean squared error (MSE) values for the LTM/GEO/PM network after every 25 iterations.

All of the networks' MSE values converged nicely, to values less than 0.15. Assume that all networks stopped training when their MSE reached a value less than 0.02 (MSE thresholds are chosen within a [0.01, 0.10] range for ANN problems). The LTM/GEO/PM network would have halted training at 246 iterations. However, it is clearly visible from figure 5.6 that improvements in the aspen, pine, pine/spruce, rock and shallow water class-conditional kappa values would be attained by further training. Obviously, it was worthwhile to train the network for an additional 379 iterations as so many class accuracies are improved. Similarly with the LTM/GEO network depicted in figure 5.4, stopping this network at 262 iterations (when MSE < 0.02) would prevent improvements within two output classes - pine and rock (achieved at 350 iterations). Training for a mere 88 iterations would improve two classes' training accuracies, one of which is a vegetative class that requires greater effort to train properly. No improvements in either of these networks' class-conditional kappa values were achieved during an additional 200 iterations and were therefore stopped.

The LTM/PM network was trained for 850 iterations. At this point, the vegetative classes began to stabilise - from 850 to 1050 training iterations, the difference between each class's maximum and minimum conditional kappa values was less than 0.05. A better pine/spruce kappa value was also achieved at 850 iterations, see figure 5.5. Once again, the MSE values for LTM/PM converged nicely but did not help in determining when to stop training the network. Both the LTM and GEO networks' class-conditional kappa values began to stabilise at around 600 and 325 training iterations respectively, see figures 5.1 and 5.2. Their maximum and minimum class-conditional kappa values did not differ by more than 0.05 during an additional 200 training iterations. Unlike the GEO and LTM/PM networks which did not improve with an additional 200 training iterations, LTM's rock class did at 625 iterations. This network was halted from training as the rock class had a very high training accuracy throughout. It was therefore decided that a further 25 training iterations was unnecessary. An interesting point is the fact that all these networks' MSE values converged nicely without providing the information regarding accuracy improvements that the class-conditional kappa values provided.

The final ANN to discuss regarding this topic is the PM network, whose MSE and class-conditional kappa values are presented in figure 5.3. This network did have higher MSE values than the other networks involved in this research however, these values also converged nicely and left little to conclude about the overall performance of the network. By looking at the various class-conditional kappa graphs, it was very noticeable that the PM network was having great difficulty in classifying the vegetative classes. The kappa values were rising and falling drastically until around 700 training iterations. This was not visible by looking at the dashed MSE graph. Knowing that the network was having such difficulties, the researcher was then aware that there could be problems with the architecture of the network, with the feature set being classified or with the training algorithm's settings. After having altered many network parameters i.e., number of layers, number of nodes, learning rate and momentum values, similar training results were still outputted. The benefit of adding this auxiliary feature set was then questioned. It was the fact that the class-conditional kappa values began to stabilise over the last 100 iterations - the difference between each class's maximum and minimum conditional kappa values was less than 0.05 - that the network's classification results were employed in this research and its structure was left as is described in §4.2. It simply emphasised the fact that not all ANNs can be perfectly trained for every type of feature set. The PM network was trained for a total of 775 iterations, a middle point in the beginning of the vegetative classes' stabilisation.

Ardö *et al.* (1997) determined network training time when a network's MSE had "decreased to a certain predetermined level". However, they did not explain how this "level" should be chosen or what "level" was used in their research. Another study involving the classification of remotely sensed data suggested to stop training when the network's weights did not change from one iteration to the next or when the change was less some threshold amount (Benediktsson, Swain, & Ersoy, 1990). How this threshold amount is determined was not discussed. Furthermore, Foody & Arora (1997) trained their networks for 750 iterations without justification. In Kavzoglu & Vieira (1998), the network performed 15 000 BP training iterations because this amount was "found to be relevant" and "allowed the network to reach local minima" (Kavzoğlu & Vieira, 1998). The network was relatively small with only 6 input nodes, one single hidden layer with 15 nodes and 8 output nodes. It classified 5835 testing pixels; the exact number of training pixels was never mentioned. Considering network and testing set sizes, 15 000 training iterations seems excessive. Did they employ patternwise training in which case weights are updated after each training pixel is represented, or were the weights updated only after a group of pixels in the training set were presented, otherwise known as epochwise training? If patternwise training was used, then this could possibly explain the large number of training iterations. Another method for choosing when to stop training a network involves the use of a validation set. Network training is stopped when the validation set's error begins to rise for a certain number of iterations. However, problems arise when a limited number of data samples are available that are only enough to form training and testing sets. In this research, it is clear that training time was dependent on the output classes' performance and accuracy which enabled straightforward justifications to be made regarding each network's number of training iterations. It has been shown why determining ANN training time by monitoring class-conditional kappa performance can be a reliable, justifiable and efficient alternative to other methods, particularly when based on a network's MSE performance.

Network	Training Iterations
Landsat Thematic Mapper (LTM)	600
Geomorphology (GEO)	325
Process Model (PM)	775
LTM/GEO	350
LTM/PM	850
LTM/GEO/PM	625

Table 5.1: Number of training iterations processed per artificial neural network.

5.3 Confusion matrix

Tables 5.2 to 5.4 below represent the confusion matrices of the LTM, GEO and PM networks' testing classification results respectively.

Classifier Categories	Known Categories								Sum
	AS	PI	PS	SF	AL	DP	RK	SH	
AS	7	2	1	3	0	0	0	0	13
PI	5	20	13	6	0	0	0	0	44
PS	0	1	6	3	1	0	0	0	11
SF	0	2	4	12	1	0	0	0	19
AL	0	1	2	1	89	0	0	0	93
DP	0	0	0	0	0	328	0	5	333
RK	0	2	0	0	1	0	250	0	253
SH	0	0	0	0	0	0	0	119	119
Sum	12	28	26	25	92	328	250	124	885

Table 5.2: Confusion matrix depicting the LTM network's classification results.

Classifier Categories	Known Categories								Sum
	AS	PI	PS	SF	AL	DP	RK	SH	
AS	8	0	2	0	0	0	0	0	10
PI	0	23	10	0	0	0	1	0	34
PS	0	3	4	0	0	0	0	0	7
SF	0	0	5	14	0	0	0	0	19
AL	0	0	1	4	86	0	5	0	96
DP	0	0	0	0	0	327	0	0	327
RK	0	0	1	7	6	1	244	0	259
SH	4	2	3	0	0	0	0	124	133
Sum	12	28	26	25	92	328	250	124	885

Table 5.3: Confusion matrix depicting the GEO network's classification results.

Classifier Categories	Known Categories								Sum
	AS	PI	PS	SF	AL	DP	RK	SH	
AS	6	0	3	0	0	0	0	0	9
PI	2	24	7	0	0	1	1	0	35
PS	0	0	1	3	0	0	0	0	4
SF	0	0	5	13	0	0	0	0	18
AL	0	0	0	6	86	0	0	0	92
DP	0	0	6	0	0	326	0	0	332
RK	0	1	3	3	6	1	249	0	263
SH	4	3	1	0	0	0	0	124	132
Sum	12	28	26	25	92	328	250	124	885

Table 5.4: Confusion matrix depicting the PM network's classification results.

The following two tables, 5.5 and 5.6, represent the confusion matrices of the LTM/GEO and LTM/PM networks' classification results respectively. As mentioned in §4.2.1, these networks classify two of the three individual feature sets simultaneously.

Classifier Categories	Known Categories								Sum
	AS	PI	PS	SF	AL	DP	RK	SH	
AS	9	2	0	0	0	0	0	0	11
PI	2	24	12	0	0	0	1	0	39
PS	0	2	10	0	0	0	0	0	12
SF	0	0	3	24	0	0	0	0	27
AL	0	0	0	0	91	0	0	0	91
DP	0	0	0	0	0	328	1	0	329
RK	0	0	0	1	1	0	248	0	250
SH	1	0	1	0	0	0	0	124	126
Sum	12	28	26	25	92	328	250	124	885

Table 5.5: Confusion matrix depicting the LTM/GEO network's classification results.

Classifier Categories	Known Categories								Sum
	AS	PI	PS	SF	AL	DP	RK	SH	
AS	10	2	2	0	1	0	0	0	15
PI	2	20	8	0	0	0	0	0	30
PS	0	2	10	0	0	0	0	0	12
SF	0	2	3	25	0	0	0	0	30
AL	0	0	1	0	90	0	0	0	91
DP	0	0	0	0	0	328	0	0	328
RK	0	2	0	0	1	0	250	0	253
SH	0	0	2	0	0	0	0	124	126
Sum	12	28	26	25	92	328	250	124	885

Table 5.6: Confusion matrix depicting the LTM/PM network's classification results.

The final confusion matrix depicted by table 5.7 represents the classification results of the LTM/GEO/PM network. As mentioned in §4.2.1, this network classifies all three individual feature sets simultaneously.

Classifier Categories	Known Categories								Sum
	AS	PI	PS	SF	AL	DP	RK	SH	
AS	11	1	0	0	0	0	0	0	12
PI	0	22	7	0	0	0	1	0	30
PS	0	3	16	0	0	0	0	0	19
SF	0	0	3	24	0	0	0	0	27
AL	0	0	0	0	91	0	0	0	91
DP	0	0	0	0	0	328	0	0	328
RK	0	2	0	1	1	0	249	0	253
SH	1	0	2	0	0	0	0	124	125
Sum	12	28	26	25	92	328	250	124	885

Table 5.7: Confusion matrix depicting the LTM/GEO/PM network's classification results.

Overall, the multiple feature set networks resulted in more correctly classified pixels. All single feature set networks had great difficulty in classifying the vegetative classes. The pine/spruce vegetative class had very poor classification results for all networks, particularly for the LTM, GEO and PM networks. The pine/spruce class was particularly difficult to distinguish from the pine class. As discussed in §3.3, the difficulty in distinguishing between vegetative classes due to their signature separability and spectral overlap was apparent in these networks' confusion matrices. The PM network had a significantly poorer pine/spruce classification. However, the LTM/GEO and LTM/PM networks produced similar classifications for this output class. From tables 5.11 and 5.13, these two networks had the same class-conditional kappa value of 0.8283 for the pine/spruce class. PM's difficulty in classifying the pine/spruce pixels was overcome when its feature set was combined with remotely sensed data in LTM/PM. All networks misclassified a large number of vegetative pixels to the pine class. This reflects the difficulty in distinguishing the vegetative classes from one another due to their reflectance overlap. As seen in Li *et al.* (1993), the multiple feature set networks' higher accuracies support the fact that using auxiliary feature sets in conjunction with remotely sensed data for the mapping of satellite imagery improves the overall classification accuracy. This is generally true as long as an optimal balance between accuracy improvements and data redundancy is attained.

Overall, the non-vegetative classes were accurately classified by all ANNs, even the single feature set networks. As depicted in §3.3, non-vegetative classes have very distinct reflectance properties which provided excellent separability from other classes. It should be noted that the non-vegetative classes also had more training pixels than the vegetative classes. In this research, the classes with significantly more training pixels - the non-vegetative classes - resulted in more correctly classified testing pixels. However, it should be noted that the classes with a small number of training pixels - the vegetative classes - did have a high number of correctly classified testing pixels when categorised using the multiple feature set networks. In fact, most networks had a high number of correctly classified vegetative pixels considering the small number of training pixels employed, supporting the fact that ANNs can train accurately with a small number of training data (Hepner et al., 1990).

The GEO network had more properly classified pixels per vegetative class than the LTM network, except for the pine/spruce class. The high correlation between GEO's feature set - elevation, aspect and slope - and subalpine vegetation type has been shown in other studies as well (Bolstad, Swank, & Vose, 1998; La Roi & Hnatiuk, 1980). The multiple feature set networks achieved significant improvements in the classification of the spruce/fir class. Once again, the additional feature sets contributed positively to the classification results. Overall, these larger networks classified with higher accuracies all output classes. It is now of interest to know whether or not the combination methods will improve upon the classification results achieved by the large LTM/GEO/PM network and by the networks that they combined.

5.4 Kappa values

Tables depicting the overall network and class-conditional kappa values are now presented. For simplicity, the LTM/GEO/PM network is abbreviated as LGP, the LTM/GEO network as LG and the LTM/PM network as LP. All table kappa values are statistically compared to the corresponding LTM/GEO/PM kappa value at a 95% confidence level. Kappa values that are statistically better than the corresponding LTM/GEO/PM kappa value are denoted by **. Kappa values that are greater in magnitude than the corresponding LTM/GEO/PM kappa value are denoted by *. Kappa values in **bold** are statistically equivalent to the corresponding LTM/GEO/PM kappa value. All combination method abbreviations are described in the table below.

Abbreviation	Description
Borda	Borda count vote
WMV	Weighted majority vote
SMV	Simple majority vote
DOS	Dempster's orthogonal sum combination
FUZ	Fuzzy integral combination
Mun	OLC-MSE combination using unconstrained weights
Mcon	OLC-MSE combination using constrained weights
Mal	OLC-MSE combination using weights equalling testing overall kappa values
AVE	Bayesian average combination
MAX	Bayesian maximum combination
MED	Bayesian medium combination
MIN	Bayesian minimum combination
SUM	Bayesian summation combination
PROD	Bayesian product combination

Table 5.8: Combination method abbreviations and their definitions.

Tables 5.9 and 5.10 compare the combinations of the three individual networks - LTM, GEO and PM - to the LTM/GEO/PM and LTM networks. Tables 5.11 and 5.12 compare the combinations of the following two networks - LTM/GEO and PM - to the LTM/GEO/PM and LTM networks. Tables 5.13 and 5.14 compare the combinations of the following two networks - LTM/PM and GEO - to the LTM/GEO/PM and LTM networks. It should be noted that when only two networks are being combined, the MED combination method was not performed. Also, it was noticed that the Mun and Mcon combination methods produced the same results. This is due to Mun's combination weights summing to one even though they were not constrained to do so.

Class		ANNs		Combination Methods of LTM + GEO + PM						
		LGP	LTM	Borda	WMV	SMV	DOS	FUZ	Mun/Mcon	Mal
AS	κ	0.9155	0.5321	**0.6959**	0.1199	0.2491	**1.000***	**1.000***	**0.7747**	**1.000***
	σ^2_κ	*0.0065*	*0.0194*	*0.0214*	*0.0011*	*0.0070*	*0*	*0*	*0.0196*	*0*
PI	κ	0.7246	0.4367	**0.6245**	**1.00****	**0.6871**	**0.6127**	**0.5534**	**0.6695**	**0.6902**
	σ^2_κ	*0.0068*	*0.0057*	*0.0073*	*0*	*0.0067*	*0.0060*	*0.0068*	*0.0091*	*0.0073*
PS	κ	0.8373	**0.5317**	0.3132	–	**1.000***	**0.7056**	**0.853***	0.5707	**0.5245**
	σ^2_κ	*0.0074*	*0.0237*	*0.0390*	–	*0*	*0.0308*	*0.0185*	*0.0105*	*0.0200*
SF	κ	0.8857	0.6209	**0.7344**	**0.6570**	**0.8375**	**0.7763**	**0.8417**	0.4855	0.4631
	σ^2_κ	*0.0038*	*0.0128*	*0.0064*	*0.0783*	*0.0074*	*0.0077*	*0.0106*	*0.0144*	*0.0112*
AL	κ	1.0000	0.9520	**0.9876**	**0.9869**	**0.9749**	0.9400	0.9089	0.8158	0.8520
	σ^2_κ	*0*	*0.0005*	*0.0002*	*0.0002*	*0.0003*	*0.0007*	*0.0009*	*0.0016*	*0.0014*
DP	κ	1.0000	0.9761	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9952**	0.9483	0.9169
	σ^2_κ	*0*	*0.0001*	*0*	*0*	*0*	*0*	*0.0000*	*0.0002*	*0.0003*
RK	κ	0.9780	**0.984***	**0.9673**	**0.994***	**0.9568**	**0.9673**	0.9362	**0.9664**	**0.9556**
	σ^2_κ	*0.0001*	*0.0001*	*0.0002*	*0.0000*	*0.0002*	*0.0002*	*0.0003*	*0.0002*	*0.0002*
SH	κ	0.9907	**1.000***	0.9295	**1.000***	0.9379	0.9379	0.9379	**1.000***	**1.000***
	σ^2_κ	*0.0001*	*0*	*0.0006*	*0*	*0.0005*	*0.0005*	*0.0005*	*0*	*0*
Overall	K	0.9698	0.9183	0.9365	0.8772	0.9244	0.9395	0.9286	0.9045	0.8996
	σ^2_K	*0.0000*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*

Table 5.9: Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) and LTM networks; and for the voting, DOS, fuzzy logic and OLC-MSE combination methods of the three individual networks - LTM, GEO and PM.

Class		ANNs		Combination Methods of LTM + GEO + PM					
		LGP	LTM	AVE	MAX	MIN	MED	SUM	PROD
AS	κ	0.9155	0.5321	**0.7747**	**0.7973**	0.4931	**0.7747**	**0.8733**	0.4931
	σ^2_κ	*0.0065*	*0.0194*	*0.0196*	*0.0164*	*0.0114*	*0.0196*	*0.0.140*	*0.0114*
PI	κ	0.7246	0.4367	**0.6195**	**0.6245**	**0.6940**	**0.6372**	**0.5970**	**0.6336**
	σ^2_κ	*0.0068*	*0.0057*	*0.0063*	*0.0073*	*0.0081*	*0.0064*	*0.0060*	*0.0070*
PS	κ	0.8373	**0.5317**	**0.5879**	**0.8283**	0.3990	**0.8283**	–	0.3132
	σ^2_κ	*0.0074*	*0.0237*	*0.0507*	*0.0245*	*0.0212*	*0.0245*	–	*0.0390*
SF	κ	0.8857	0.6209	**0.8285**	**0.7579**	**0.7106**	**0.7942**	**0.7625**	**0.7106**
	σ^2_κ	*0.0038*	*0.0128*	*0.0061*	*0.0111*	*0.0066*	*0.0067*	*0.0071*	*0.0066*
AL	κ	1.0000	0.9520	0.9288	0.9195	**0.9877**	0.9509	0.9089	**0.9877**
	σ^2_κ	*0*	*0.0005*	*0.0008*	*0.0008*	*0.0001*	*0.0006*	*0.0009*	*0.0001*
DP	κ	1.0000	0.9761	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	σ^2_κ	*0*	*0.0001*	*0*	*0*	*0*	*0*	*0*	*0*
RK	κ	0.9780	**0.984***	**0.9673**	**0.9413**	**0.983***	**0.9516**	**0.9780**	**0.983***
	σ^2_κ	*0.0001*	*0.0001*	*0.0002*	*0.0003*	*0.0001*	*0.0002*	*0.0001*	*0.0001*
SH	κ	0.9907	**1.000***	0.9379	0.9213	**1.000***	0.9379	0.9379	**1.000***
	σ^2_κ	*0.0001*	*0*	*0.0005*	*0.0006*	*0*	*0.0005*	*0.0005*	*0*
Overall	K	0.9698	0.9183	0.9380	0.9302	0.9413	0.9379	0.9365	0.9412
	σ^2_K	*0.0000*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*

Table 5.10: Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) and LTM networks, and for the Bayesian combination methods of the three individual networks - LTM, GEO and PM.

Class		ANNs		Combination Methods of LTM/GEO + PM					
		LGP	LG	Borda	WMV/SMV	DOS	FUZ	Mun/Mcon	Mal
AS	κ	0.9155	**0.8157**	**0.8157**	0.1610	**0.7466**	**0.6959**	**0.7747**	**1.000***
	σ^2_κ	*0.0065*	*0.0138*	*0.0138*	*0.0023*	*0.0159*	*0.0214*	*0.0196*	*0*
PI	κ	0.7246	**0.6028**	**0.6467**	**0.801***	**0.6930**	**0.6845**	**0.5697**	**0.7096**
	σ^2_κ	*0.0068*	*0.0063*	*0.0061*	*0.0063*	*0.0059*	*0.0061*	*0.0070*	*0.0066*
PS	κ	0.8373	**0.8283**	**0.8127**	**1.000***	**0.7424**	**0.897***	**0.5758**	**0.8283**
	σ^2_κ	*0.0074*	*0.0122*	*0.0143*	*0*	*0.0165*	*0.0095*	*0.0149*	*0.0245*
SF	κ	0.8857	**0.8857**	**0.8094**	**0.8530**	**0.8475**	**0.8813**	**0.8857**	0.5884
	σ^2_κ	*0.0038*	*0.0038*	*0.0059*	*0.0092*	*0.0049*	*0.0041*	*0.0038*	*0.0070*
AL	κ	1.0000	**1.0000**	**0.9879**	**1.0000**	**0.9879**	**0.9877**	**1.0000**	0.9535
	σ^2_κ	*0*	*0*	*0.0001*	*0*	*0.0001*	*0.0001*	*0*	*0.0014*
DP	κ	1.0000	**0.9952**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9904**	0.9714
	σ^2_κ	*0*	*0.0000*	*0*	*0*	*0*	*0*	*0.0000*	*0.0001*
RK	κ	0.9780	**0.989***	**0.994***	**0.994***	**0.994***	**0.9780**	**0.989***	**0.994***
	σ^2_κ	*0.0001*	*0.0001*	*0.0000*	*0.0000*	*0.0000*	*0.0001*	*0.0001*	*0.0000*
SH	κ	0.9907	**0.9815**	**0.9549**	**0.9815**	**0.9725**	0.9379	**0.9815**	**0.9815**
	σ^2_κ	*0.0001*	*0.0002*	*0.0004*	*0.0002*	*0.0002*	*0.0005*	*0.0002*	*0.0002*
Overall	K	0.9698	**0.9593**	**0.9563**	0.9128	**0.9608**	**0.9547**	**0.9502**	0.9304
	σ^2_K	*0.0000*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*

Table 5.11: Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) and LTM/GEO (LG) networks; and for the voting, DOS, fuzzy logic and OLC-MSE combination methods of following two networks - LTM/GEO (LG) and PM.

Class		ANNs		Combination Methods of LTM/GEO + PM				
		LGP	LG	AVE	MAX	MIN	SUM	PROD
AS	κ	0.9155	**0.8157**	**1.000***	**0.8874**	**0.8986**	**1.000***	**0.8986**
	σ^2_κ	*0.0065*	*0.0138*	*0*	*0.0113*	*0.0092*	*0*	*0.0092*
PI	κ	0.7246	**0.6028**	**0.6293**	**0.6293**	**0.6459**	**0.6222**	**0.6459**
	σ^2_κ	*0.0068*	*0.0063*	*0.0061*	*0.0061*	*0.0067*	*0.0058*	*0.0067*
PS	κ	0.8373	**0.8283**	**0.8127**	**0.7939**	**0.7622**	**0.886***	**0.7622**
	σ^2_κ	*0.0074*	*0.0122*	*0.0143*	*0.0169*	*0.0144*	*0.0116*	*0.0144*
SF	κ	0.8857	**0.8857**	**0.8475**	**0.8857**	**0.8162**	**0.8475**	**0.8162**
	σ^2_κ	*0.0038*	*0.0038*	*0.0049*	*0.0038*	*0.0055*	*0.0049*	*0.0055*
AL	κ	1.0000	**1.0000**	**0.9879**	**1.0000**	**0.9879**	**0.9879**	**0.9879**
	σ^2_κ	*0*	*0*	*0.0001*	*0*	*0.0001*	*0.0001*	*0.0001*
DP	κ	1.0000	**0.9952**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	σ^2_κ	*0*	*0.0000*	*0*	*0*	*0*	*0*	*0*
RK	κ	0.9780	**0.989***	**0.989***	**0.983***	**0.994***	**0.989***	**0.994***
	σ^2_κ	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0000*	*0.0001*	*0.0000*
SH	κ	0.9907	**0.9815**	**0.9637**	**0.9549**	**0.9549**	**0.9463**	**0.9549**
	σ^2_κ	*0.0001*	*0.0002*	*0.0003*	*0.0004*	*0.0004*	*0.0005*	*0.0004*
Overall	K	0.9698	**0.9593**	**0.9593**	**0.9577**	**0.9578**	**0.9562**	**0.9578**
	σ^2_K	*0.0000*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*

Table 5.12: Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) and LTM/GEO (LG) networks, and for the Bayesian combination methods of the following two networks - LTM/GEO (LG) and PM.

Class		ANNs		Combination Methods of LTM/PM + GEO					
		LGP	LP	Borda	WMV/SMV	DOS	FUZ	Mun/Mcon	Mal
AS	κ	0.9155	**0.6621**	**0.6101**	0.1857	**0.7661**	**0.7661**	**0.7973**	**1.000***
	σ^2_κ	*0.0065*	*0.0151*	*0.0185*	*0.0024*	*0.0139*	*0.0139*	*0.0164*	*0*
PI	κ	0.7246	**0.6558**	**0.5932**	**0.820***	**0.6871**	**0.6164**	**0.6336**	**0.762***
	σ^2_κ	*0.0068*	*0.0077*	*0.0075*	*0.0066*	*0.0067*	*0.0069*	*0.0077*	*0.0072*
PS	κ	0.8373	**0.8283**	**0.7424**	**0.6566**	**0.6566**	**0.6254**	**0.6780**	**0.6830**
	σ^2_κ	*0.0074*	*0.0122*	*0.0248*	*0.0784*	*0.0195*	*0.0221*	*0.0141*	*0.0172*
SF	κ	0.8857	**0.8285**	**0.8226**	**0.931***	**0.8285**	**0.8456**	**0.8285**	0.5884
	σ^2_κ	*0.0038*	*0.0048*	*0.0052*	*0.0044*	*0.0048*	*0.0067*	*0.0048*	*0.0082*
AL	κ	1.0000	**0.9877**	**0.9757**	**1.0000**	**1.0000**	**0.9640**	**0.9877**	0.8438
	σ^2_κ	*0*	*0.0001*	*0.0003*	*0*	*0*	*0.0004*	*0.0001*	*0.0014*
DP	κ	1.0000	**1.0000**	**0.9952**	**1.0000**	**1.0000**	**1.0000**	**0.9856**	**0.9530**
	σ^2_κ	*0*	*0*	*0.0000*	*0*	*0*	*0*	*0.0001*	*0.0002*
RK	κ	0.9780	**0.984***	**0.9725**	**0.994***	**0.989***	**0.9566**	**0.989***	**0.994***
	σ^2_κ	*0.0001*	*0.0001*	*0.0001*	*0.0000*	*0.0001*	*0.0002*	*0.0001*	*0.0000*
SH	κ	0.9907	**0.9815**	**0.9549**	**0.9907**	**0.9725**	**0.9637**	**0.9907**	**0.9907**
	σ^2_κ	*0.0001*	*0.0002*	*0.0004*	*0.0001*	*0.0002*	*0.0003*	*0.0001*	*0.0001*
Overall	K	0.9698	**0.9577**	0.9426	0.9144	**0.9593**	**0.9524**	**0.9547**	0.9273
	σ^2_K	*0.0000*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*

Table 5.13: Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) and LTM/PM (LP) networks; and for the voting, DOS, fuzzy logic and OLC-MSE combination methods of the following two networks - LTM/PM (LP) and GEO.

Class		ANNs		Combination Methods of LTM/PM + GEO				
		LGP	LP	AVE	MAX	MIN	SUM	PROD
AS	κ	0.9155	**0.6621**	**0.6621**	**0.7235**	0.5269	**0.7235**	0.4931
	σ^2_κ	*0.0065*	*0.0151*	*0.0189*	*0.0184*	*0.0168*	*0.0184*	*0.0158*
PI	κ	0.7246	**0.6558**	**0.7151**	**0.6902**	**0.7002**	**0.6355**	**0.7246**
	σ^2_κ	*0.0068*	*0.0077*	*0.0072*	*0.0073*	*0.0069*	*0.0070*	*0.0068*
PS	κ	0.8373	**0.8283**	**0.6566**	**0.6136**	**0.7424**	**0.6566**	**0.7711**
	σ^2_κ	*0.0074*	*0.0122*	*0.0260*	*0.0309*	*0.0248*	*0.0391*	*0.0203*
SF	κ	0.8857	**0.8285**	**0.7427**	**0.7256**	**0.7427**	**0.7505**	**0.7427**
	σ^2_κ	*0.0038*	*0.0048*	*0.0061*	*0.0068*	*0.0061*	*0.0058*	*0.0061*
AL	κ	1.0000	**0.9877**	**0.9757**	**0.9757**	**1.0000**	**0.9640**	**1.0000**
	σ^2_κ	*0*	*0.0001*	*0.0003*	*0.0003*	*0*	*0.0004*	*0*
DP	κ	1.0000	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	σ^2_κ	*0*	*0*	*0*	*0*	*0*	*0*	*0*
RK	κ	0.9780	**0.984***	**0.984***	**0.9673**	**0.983***	**0.989***	**0.989***
	σ^2_κ	*0.0001*	*0.0001*	*0.0001*	*0.0002*	*0.0001*	*0.0001*	*0.0000*
SH	κ	0.9907	**0.9815**	**0.9463**	**0.9463**	**0.9463**	**0.9463**	**0.9463**
	σ^2_κ	*0.0001*	*0.0002*	*0.0005*	*0.0005*	*0.0005*	*0.0005*	*0.0005*
Overall	K	0.9698	**0.9577**	**0.9486**	0.9440	0.9472	0.9457	**0.9487**
	σ^2_K	*0.0000*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*	*0.0001*

Table 5.14: Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) and LTM/PM (LP) networks, and for the Bayesian combination methods of the following two networks - LTM/PM (LP) and GEO.

LTM's only vegetative class to have a kappa value statistically equivalent to one of LTM/GEO/PM's was the pine/spruce class. Although these two pine/spruce kappa values were equivalent, they differed in magnitude greatly. LTM/GEO/PM achieved a value of 0.8373 whereas LTM's was only 0.5317. The LTM network did prove to classify the non-vegetative classes very well. Although its overall kappa value was greater than 0.9, it was not statistically equivalent to the LTM/GEO/PM network's kappa value. This supports the fact that there is a need to improve the classification accuracies of remotely sensed data, particularly when speaking of vegetation. All of LTM's vegetative classes' kappa values were quite low, including a value less than 0.5 for the pine class. These accuracies were increased by combining the LTM network with the GEO and PM networks using the DOS, FUZ, AVE, MAX and MED combination methods. However, no one combination resulted in an overall kappa value that was statistically equivalent to LTM/GEO/PM's value. Several combinations reflected the difficulty in classifying the pine and pine/spruce pixels properly. The WMV and SUM methods were not able to classify any of the pine/spruce pixels properly. However, significant classification improvements were achieved with the DOS and FUZ combination methods for these two vegetative classes.

The LTM/GEO network performed very well compared to the LTM network. The vegetative classes' accuracies were improved significantly. The LTM/GEO had a lower pine kappa value because of the large number of pine/spruce pixels being misclassified to it. However, this value is still greater than 0.5 and is statistically equivalent to the LTM/GEO/PM's pine kappa value. Overall, the smaller LTM/GEO network was statistically equivalent to the larger LTM/GEO/PM network. Many combinations of LTM/GEO and PM networks were equivalent but not better than the LTM/GEO/PM network. As seen in the combination results for the LTM, GEO and PM networks, the difficulty in properly classifying the vegetative classes is apparent in the WMV/SMV combination method of the LTM/GEO and PM networks. This method's high pine, pine/spruce and spruce/fir classification accuracies are the benefit of a very poor aspen classification accuracy due to many pixels being misclassified as this vegetation. It is required to achieve high accuracies in all vegetative classes and therefore this combination method is too simple to classify such data. On the other hand, the SUM combination method provides excellent classifications for the vegetative classes, unlike the results seen in table 5.10. Although many combination methods were equivalent to the LTM/GEO/PM network, their class-conditional kappa values were lower than those achieved by the LTM/GEO network. DOS resulted in lower kappa values for three vegetative and two non-vegetative classes, while FUZ produced lower kappa values for one vegetative and three non-vegetative classes when compared to LTM/GEO's corresponding kappa values. From these results, a question arises as to whether or not the process model feature set should be utilised for satellite image classification, or should the geomorphology be the only additional feature set? Is the addition of the process model feature set through combinations of multiple ANNs decreasing classification accuracies? The overall kappa value achieved by DOS is larger than LTM/GEO's. The SUM combination method provided excellent classification and resulted in many improvements. Therefore, the process model feature set does contribute positively to the classification and should not be omitted.

The LTM/PM network also improved significantly upon the classification accuracies achieved by the LTM network alone, the vegetative classes improving the most. The LTM/PM network had lower kappa values for the aspen and pine vegetative classes. Once again, this was due to the large number of pixels being misclassified from the pine/spruce class. However, the simpler LTM/PM network was statistically equivalent to the larger LTM/GEO/PM network. Fewer combinations of LTM/PM and GEO than combinations of LTM/GEO and PM were equivalent to the LTM/GEO/PM network. The DOS algorithm was the only combination method with an overall kappa value larger than the LTM/PM network's value. The WMV/SMV combination produced similar problematic results as previously explained. All combination methods reduced the accuracy of the pine/spruce class by a significant amount. However, the geomorphology feature set did contribute positively to the classification as the LTM/GEO network's results were overall better than the LTM/PM network's.

5.5 Further results

After having evaluated the results presented in §5.4, the LTM/GEO and LTM/PM networks' results were compared to combinations of their individual networks. These comparisons investigate whether adding a second feature set through a combination method resulted in better classification accuracies than adding it directly to the network's input space. DOS was employed for this task as it produced higher overall kappa values than the networks it combined.

The class-conditional and overall kappa values for the LTM/GEO/PM network; the DOS combination of the LTM, GEO and PM networks; the LTM/GEO network; the DOS combination of the LTM and GEO networks; the LTM/PM network; and

Class		LGP	LTM + GEO + PM	LG	LTM + GEO	LP	LTM + PM
AS	κ	0.9155	**1.000***	**0.8157**	**1.000***	**0.6621**	**0.6959**
	σ^2_κ	*0.0065*	*0*	*0.0138*	*0*	*0.0151*	*0.0214*
PI	κ	0.7246	**0.6127**	**0.6028**	**0.5306**	**0.6558**	**0.6450**
	σ^2_κ	*0.0068*	*0.0060*	*0.0063*	*0.0057*	*0.0077*	*0.0073*
PS	κ	0.8373	**0.7056**	**0.8283**	**0.7711**	**0.8283**	**0.7622**
	σ^2_κ	*0.0074*	*0.0308*	*0.0122*	*0.0203*	*0.0122*	*0.0144*
SF	κ	0.8857	**0.7763**	**0.8857**	**0.8456**	**0.8285**	**0.8456**
	σ^2_κ	*0.0038*	*0.0077*	*0.0038*	*0.0067*	*0.0048*	*0.0067*
AL	κ	1.0000	0.9400	**1.0000**	0.9413	**0.9877**	0.9413
	σ^2_κ	*0*	*0.0007*	*0*	*0.0006*	*0.0001*	*0.0006*
DP	κ	1.0000	**1.0000**	**0.9952**	**1.0000**	**1.0000**	**1.0000**
	σ^2_κ	*0*	*0*	*0.0000*	*0*	*0*	*0*
RK	κ	0.9780	**0.9673**	**0.989***	**0.9673**	**0.984***	**0.9620**
	σ^2_κ	*0.0001*	*0.0002*	*0.0001*	*0.0002*	*0.0001*	*0.0002*
SH	κ	0.9907	0.9379	**0.9815**	**0.9815**	**0.9815**	**0.9463**
	σ^2_κ	*0.0001*	*0.0005*	*0.0002*	*0.0002*	*0.0002*	*0.0005*
Overall	K	0.9698	0.9395	**0.9593**	0.9425	**0.9577**	0.9425
	σ^2_K	*0.0000*	*0.0001*	*0.0001*	0.0001	*0.0001*	*0.0001*

Table 5.15: Overall and class-conditional kappa values for the LTM/GEO/PM (LGP) network; the DOS combination of the LTM, GEO and PM (LTM + GEO + PM) networks; the LTM/GEO (LG) network; the DOS combination of the LTM and GEO networks (LTM + GEO); the LTM/PM (LP) network; and the DOS combination of the LTM and PM (LTM + PM) networks

the DOS combination of the LTM and PM networks are presented in table 5.15 above. This table is read in the same fashion as the tables presented in §5.4. All results are being compared to the LTM/GEO/PM network's kappa values. It is apparent that the large LTM/GEO/PM network produced the overall best classification. The various DOS combinations presented were not statistically equivalent to the LTM/GEO/PM network's overall kappa value. However, the DOS combination of the LTM and GEO networks was statistically equivalent to the LTM/GEO network, as well as the the DOS combination of the LTM and PM networks to the LTM/PM network. Although the DOS combinations of two networks were not equivalent to the large LTM/GEO/PM network, they were equivalent to their corresponding multiple feature set networks.

When compared to the other combination methods investigated, the DOS algorithm had the overall best performance. It produced reasonably high classification accuracies no matter which networks were being combined. In contrast, the Bayesian techniques achieved high classification accuracies when the LTM/GEO and PM networks were combined but failed to produce such accuracies when combining the LTM, GEO and PM networks. Of all combination methods, DOS achieved the highest overall kappa values for all three possible combinations, with the exception of MIN's and PROD's values when combining the LTM, GEO and PM networks. Although their overall kappa values were greater than DOS's for this situation, both MIN and PROD classified the aspen and pine/spruce vegetative classes poorly. Some combination methods classified certain output classes very accurately but other classes very poorly. For example, the Mal combination of LTM/PM and GEO produced significantly accurate aspen, pine and rock classifications but at the same time, it produced significantly poor pine/spruce and alpine classifications. The DOS method did not have certain classes with high accuracies at the benefit of other's poor values. In Xu *et al.* (1992), classifier combinations based on Bayesian principles, voting techniques and Dempster's orthogonal sum were compared. They conclude their research by recommending Dempster's method stating that "it can obtain high recognition and reliability rates simultaneously and robustly". The FUZ combination method achieved classification accuracies comparable to DOS. In Cho & Kim (1995), the superiority of FUZ over various voting techniques is addressed. Both DOS and FUZ performed well regardless of the networks being combined, demonstrating their ability to accurately classify land cover independent of individual network performance.

Chapter 6

Discussion of Results

Vieira & Mather (2000) investigated the classification of a SPOT image depicting various crop types. The four individual classifiers being combined were a MLC, a MDC and two differently structure BPNNs. These classifiers' overall kappa values were quite low - 0.58 for MLC, 0.43 for MDC, and 0.57 and 0.64 for the two ANNs. Their results support the use of ANNs for the classification of remotely sensed data. This research produced better individual ANN results with 0.92 for LTM, 0.92 for GEO and 0.91 for PM. The ANNs classifying the SPOT feature set resulted in overall kappa values (0.57 and 0.64) which were much lower in magnitude than the ANN classifying the Landsat-TM feature set (0.92). These differences could be due to the differing natures of the feature sets however, the ANNs classifying the geomorphology and process model feature sets resulted in very high overall kappa values similarly to LTM. Perhaps it is the design of the individual classifiers - how they were trained and what were their individual parameter settings - which influenced the testing classification results. The specifics of each individual classifier were not fully discussed in Vieira & Mather (2000) and therefore no direct comparison can be made with the network designs utilised in this research. By applying the methods and designs employed in this research to Vieira & Mather's (2000) SPOT classification problem, these hypothesis could be answered. If the SPOT classification accuracy increases to a value similar to those achieved by LTM, GEO and PM, then the methods and designs employed in this research perform much more accurately for the classification of remotely sensed data whereas, if the SPOT accuracy results in values similar to those achieved in Vieira & Mather (2000), then it is the nature of the feature set which is more difficult to classify.

Once the different classifiers' results were combined, both studies resulted in increased overall kappa values. Vieira & Mather (2000) employed several combination methods based on voting principles, Bayesian techniques and Dempster's orthogonal sum. They also used an ANN to combine the individual classifier results. Interestingly, it was not the combination methods that resulted in the best classification - the ANN achieved an overall kappa value of 0.77 whereas the range of kappas for the combination methods was from 0.38 to 0.65. Most of their combinations did improve the classification accuracy of the remotely sensed data, supporting the use of classifier combinations when initial individual classification results are poor. In this research, the various combination methods employed (for LTM + GEO + PM) resulted in overall kappa values in the range of 0.88 to 0.94. Similarly to Vieira & Mather (2000), most combinations resulted in improved classification accuracies however, these improvements were less significant. Perhaps this is due to the initially high LTM, GEO and PM classification accuracies. In addition to the research performed here, it would be of interest to apply an ANN for the combination of the individual network results in order to further investigate the use of such a classifier in improving the classification of land cover.

Collins *et al.* (2001), a similar study to this research, combined multiple 5-NN classifiers on the same data set using the DOS combination method, producing different results than those achieved here. The DOS combination of the LTM and GEO 5-NN classifiers resulted in overall kappa values which were statistically equivalent and higher in magnitude than the LTM/GEO 5-NN classifier. The same held true for the DOS combination of the LTM and PM 5-NN classifiers when compared to the LTM/PM 5-NN classifier. As discussed in §5.5, the DOS combination of LTM and GEO, and of LTM and PM produced overall kappa values statistically equivalent but less in magnitude than their corresponding multiple feature set networks' values. In Collins *et al.* (2001), the DOS combination of the LTM, GEO and PM 5-NN classifiers resulted in an overall kappa value that was statistically better than that of the LTM/GEO/PM 5-NN classifier. Here, the DOS combination of the LTM, GEO and PM networks resulted in an overall kappa value that was not statistically equivalent to the LTM/GEO/PM network's value. Collins *et al.*'s (2001) results provide support for using combinations of multiple classifiers for the improvement of classification accuracies. According to table 5.15, this research did not produce significant accuracy improvements when adding a second feature set through combinations of ANNs versus adding the additional data directly to a network's input space. Aspen was the only class for which all DOS combinations produced higher kappa values than the individual networks being combined. However, this research does provide support for other benefits of combining ANNs over using a single large network - reductions in training time and operator effort, use of previously classified results, and avoidance of individual network shortfalls. It would be of interest to evaluate combinations of other classifiers e.g., MLC or MDC, in order to deduce which classifiers produce the best canopy type classifications when combined using the DOS combination method.

Giacinto *et al.* (2000) combined the results of a statistical (k-NN) and various neural classifiers for the mapping of agricultural land use. For the two combination experiments performed on their entire image, all three BPNNs constructed had higher overall kappa values compared to those of the k-NN classifier. The best BPNN resulted in an overall kappa value of 0.86 while the k-NN classifier produced a smaller value of 0.81. The results achieved in both this research and Giacinto *et al.* (2000) support the use of BPNNs for the classification of remotely sensed data. The three combination methods investigated in Giacinto *et al.* (2000) were based on voting, Bayesian and belief function principles as previously investigated in Xu *et al.* (1992). All combination methods improved the problem's classification accuracy. However, similar to some of the results achieved in this research by the same combination methods, their combination methods' improvements were small in magnitude. Overall, both studies showed how combinations of multiple classifier results can improve the classification of remotely sensed data.

From the results presented in §5, all networks had overall kappa values greater than 0.9. The best classification results were achieved by the large multiple feature set network LTM/GEO/PM. As well as being simpler networks to work with, the LTM/GEO and LTM/PM networks obtained statistically equivalent results as the LTM/GEO/PM network. All resulted in better classification accuracies than the LTM network classifying remotely sensed data alone. Significant improvements in subalpine vegetative canopy classification were achieved using BPNNs in this study.

Many combination methods achieved results equivalent to those of the networks being combined. In certain cases, improvements were attained for specific output classes, particularly those of a vegetative nature. The DOS algorithm resulted in better overall classifications than the other combination methods. The fact that DOS used an uncertainty term associated with each network's classification classified helped in its better results. The combination methods worked better when either LTM/GEO and PM or LTM/PM and GEO were combined. The combinations of the single feature set networks LTM, GEO and PM produced less accurate classifications which could be credited to each network's initial poor classification.

It is recommended to use combinations of multiple ANNs results rather than a single large network classifying all feature sets at once. In this research, the use of a single large ANN did not result in significantly greater classification accuracies when compared with reduced training time (in minutes) and operator effort required when combining smaller and simpler ANNs.

Chapter 7

Conclusion

This research investigated the use of BPNNs to classify remotely sensed, geomorphology and process model feature sets individually and in various combinations; analysed various methods for combining ANN results and presented benefits of adding auxiliary features sets to remotely sensed data. Focus was placed on improving the accuracy of land cover classifications using remotely sensed data, particularly of canopy types in subalpine ecosystems.

The LTM/GEO, LTM/PM and LTM/GEO/PM multiple feature set networks resulted in better classification accuracies than the LTM network. Ten combinations of the LTM, GEO and PM single feature set networks had higher overall kappa values than the LTM network alone. As concluded in other studies, the addition of representative auxiliary feature sets to remotely sensed data increased the classification accuracy. In this research, these improvements were especially prominent in the vegetative classes. All vegetative classes' accuracies were increased with the addition of the process model and geomorphology feature sets, the later being the slightly better addition of the two.

The aspen class was only moderately improved with the addition of the process model feature set. It was the geomorphology feature set that drastically improved this vegetative class's classification accuracy. Both the pine/spruce and spruce/fir classes achieved classifications virtually equal to the LTM/GEO/PM network's results when the geomorphology or process model feature set was added to the network's input space i.e., LTM/GEO or LTM/PM. The pine class's accuracy was improved by similar amounts for most classifications. The LTM/GEO network, the LTM/PM network and the many combinations of three and two networks increased pine's class-conditional kappa value to a range of [0.6, 0.7]. This is much better than its kappa value of 0.4367 for the Landsat-TM feature set alone. The non-vegetative classes all had very high classification accuracies. The LTM network alone produced accuracies higher than the LTM/GEO/PM network for the rock and shallow water output classes. Overall, the addition of auxiliary feature sets to remotely sensed data improved the classification of all vegetative classes studied in this research.

Although the various combinations of single and multiple feature set networks did not produce statistically better results than the large LTM/GEO/PM network, many resulted in statistically equivalent classifications. These combination methods are still very beneficial - the DOS algorithm recommended for combinations of multiple BPNNs. The smaller networks needed less time to train on their respective feature sets. The usefulness of the individual feature sets, particularly the process model feature set, was evaluated during training and testing. The process model feature set was very difficult to train and its final classification results reflected this fact. After having altered many of PM's network settings, similar results were always achieved. This showed that the network simply had difficulty classifying this feature set. The problems with the bad shallow water and pine/spruce pixels were much more apparent in the smaller networks' results than in the larger networks. Another benefit of this research's use of network combination methods was that the new feature set was added to the classification results without having to reclassify the existing feature sets. Thinking of the LTM/GEO and PM networks, a network did not need to be retrained and retested on the Landsat-TM and geomorphology feature sets in order to add in the process model feature set. The LTM/GEO network's results were simply combined to the PM network's results. The LTM/GEO/PM network needed to be retrained and retested on the Landsat-TM and geomorphology feature sets because of the addition of the process model feature set.

In conclusion, BPNNs are a fast and reliable method of classifying remotely sensed data, limitations dependent on training data quality and representativeness of classification structure. Furthermore, the combination of multiple BPNNs have many benefits and is an efficient method for the classification of satellite imagery, particularly when utilising the DOS combination method.

References

Abuelgasim, A., Gopal, S., Irons, S. J., & Strahler, A. (1996). Classification of ASAS multi-angle and multispectral measurements using artificial neural networks. *Remote Sensing of Environment*, *59*, 79-87.

Ainsworth, E. J., & Jones, I. S. F. (1999). Radiance spectra classification from the ocean color and temperature scanner on ADEOS. *IEEE Transactions on Geoscience and Remote Sensing*, *37*(3), 1645-1656.

Ardö, J., Pilesjö, P., & Skidmore, A. (1997). Neural networks, multitemporal Landsat Thematic Mapper data and topographic data to classify forest damages in the Czech Republic. *Canadian Journal of Remote Sensing*, *23*(3), 217-229.

Asrar, G., Myneni, R. B., & Kanemasu, E. T. (1989). Estimation of plant-canopy attributes from spectral reflectance measurements. In *Theory and applications of optical remote sensing* (p. 252-296). New York, NY USA: John wiley and Sons.

Atkinson, P. M., & Tatnall, A. R. L. (1997). Neural networks in remote sensing. *International Journal of Remote Sensing*, *18*(4), 699-709.

Austin, M. P., Cunningham, R. B., & Fleming, P. M. (1984). New approaches to direct gradient analysis using environmental scalars and statistical curve-fitting procedures. *Vegetatio*, *55*, 11-27.

Azimi-Sadjadi, M. R., Shaikh, M. A., Tian, B., Eis, K. E., & Reinke, D. (1996). Neural network-based cloud detection/classification using textural and spectral features. *Proceedings of the International Geoscience and Remote Sensing Symposium*, *2*, 1105-1107.

Benediktsson, J. A., Ersoy, O. K., & Swain, P. H. (1991). A consensual neural network. *Proceedings of the International Geoscience and Remote Sensing Symposium*, *4*, 2219-2222.

Benediktsson, J. A., Swain, P. H., & Ersoy, O. K. (1990). Neural network approaches versus statistical methods in classification of multisource remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, *28*(4), 540-551.

Bishop, C. M. (1995). *Neural networks for pattern recognition.* New York, NY USA: Oxford University Press.

Bishop, Y., Fienberg, S., & Holland, P. (1975). *Discrete multivariate analysis: Theory and practice.* Cambridge, MA USA: MIT Press.

Bolstad, P. V., Swank, W., & Vose, J. (1998). Predicting southern Appalachian overstory vegetation with digital terrain data. *Landscape Ecology*, *13*, 271-283.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123-140.

Campbell, J. B. (1996). *Introduction to remote sensing* (2 ed.). New York, NY USA: Guilford Press.

Cho, S. B., & Kim, J. H. (1995). Combining multiple neural networks by fuzzy integral for robust classification. *IEEE Transactions on Systems, Man, Cybernetics*, *25*(2), 380-384.

Clemen, R. T. (1989). Combining forecasts: A review and annotating bibliography. *International Journal of Forecasting*, *5*(4), 559-583.

Collins, M. J., Dymond, C., & Johnson, E. A. (2001). *Mapping subalpine forest types using networks of nearest neighbors classifiers.* (Unpublished)

Congalton, R. G., & Green, K. (1999). *Assessing the accuracy of remotely sensed data: Principles and practices.* Boca Raton, FL USA: Lewis Publishers.

Decatur, S. E. (1989). Application of neural networks to terrain classification. *Proceedings of the International Joint Conference on Neural Networks*, *1*, 283-288.

Dymond, C. (1998). *Redefining species tolerances in a subalpine forest by using energy and moisture processes.* Unpublished master's thesis, University of Calgary, Calgary, AB CND.

Foody, G. M., & Arora, M. K. (1997). An evaluation of some factors affecting the accuracy of classification by an artificial neural network. *International Journal of Remote Sensing*, *16*(4), 799-810.

Foody, G. M., McCulloch, M. B., & Yates, W. B. (1995). Classification of remotely sensed data by an artificial neural network: Issues related to training data characteristics. *Photogrammetric Engineering and Remote Sensing*, *61*(4), 391-401.

Franklin, J. (1995). Predictive vegetation mapping: geographic modelling of biospatial patterns in relation to environmental gradients. *Progress in Physical Geography*, *19*(4), 474-499.

Fryer, G. I., & Johnson, E. A. (1988). Reconstructing fires behaviour and effects in a subalpine forest. *Journal of Applied Ecology*, *25*, 1063-1072.

Giacinto, G., Roli, F., & Bruzzone, L. (2000). Combination of neural and statistical algorithms for supervised classification of remote-sensing images. *Pattern Recognition Letters*, *21*, 385-397.

Gopal, S., Woodcock, C. E., & Strahler, A. H. (1999). Fuzzy neural network classification of global land cover from a 1^o AVHRR data set. *Remote Sensing of Environment*, *67*(2), 230-243.

Guth, P. L. (1995). Slope and aspect calculations on gridded digital elevation models: examples from a geomorphometric toolbox for a personal computer. *Zeitschrift fur Geomorphologie (supplements)*, *101*, 31-52.

Hansen, L. K., & Salamon, P. (1990). Neural networks ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(10), 993-1001.

Hashem, S., & Schmeiser, B. (1995). Improving model accuracy using optimal combinations of trained neural networks. *IEEE Transactions on Neural Networks*, *6*(3), 792-794.

Hepner, G. F., Logan, T., Ritter, N., & Bryant, N. (1990). Artificial neural network classification using a minimal training set: Comparison to conventional supervised classification. *Photogrammetric Engineering and Remote Sensing*, *56*(4), 469-473.

Huang, Y. S., & Suen, C. Y. (1995). A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(1), 90-94.

Janz, B., & Storr, D. (1977). *The climate of the contiguous mountain parks* (Project No. 30). Applications and Consulting Division, Environment Canada.

Johnson, E. A., & Fryer, G. I. (1987). Historical vegetation change in the Kananaskis Valley, Canadian Rockies. *Canadian Journal of Botany*, *65*, 853-858.

Johnson, E. A., & Larsen, C. P. S. (1991). Climatically induced change in fire frequency in the Southern Canadian Rockies. *Ecology*, *72*(1), 194-201.

Kanellopoulos, I., Varfis, A., Wilkinson, G. G., & Mégier, J. (1992). Land-cover discrimination in SPOT HRV imagery using an artificial neural network: a 20 class experiment. *International Journal of Remote Sensing*, *13*, 917-924.

Kanellopoulos, I., & Wilkinson, G. G. (1997). Strategies and best practice for neural network image classification. *Internatioal Journal of Remote Sesnsing*, *18*(4), 711-725.

Kaufman, Y. J. (1984). The atmospheric effect and the separability of field classes measured from satellites. *Remote Sensing of Environment*, *18*, 21-34.

Kavzoğlu, T. (2001). *An investigation of the design and use of feed-forward artificial* *neural networks in the classification of remotely sensed images.* Unpublished doctoral dissertation, University of Nottingham, Nottingham, UK.

Kavzoğlu, T., & Vieira, C. A. O. (1998). An analysis of artificial neural network pruning algorithms in relation to land cover classification accuracy. *Proceedings of the Remote Sensing Society Student Conference*, 53-58.

Kittler, J., Hatef, M., Duin, R. P. W., & Matas, J. (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(3), 226-239.

Kittler, J., Hojjatoleslami, A., & Windeatt, T. (1997). Strategies for combining classifiers employing shared and distinct pattern representations. *Pattern Recognition Letters*, *18*(11-13), 1373-1377.

Kumar, A. S., Basu, S. K., & Majumdar, K. L. (1997). Robust classification of multispectral data using multiple neural networks and fuzzy integral. *IEEE Transactions on Geoscience and Remote Sensing*, *35*(3), 787-790.

La Roi, G. H., & Hnatiuk, R. J. (1980). The pinus contorta forests of Banff and Jasper National Parks: a study in comparitive synecology and syntaxonomy. *Ecological Monographs*, *50*(1), 1-29.

Lee, T., Richards, J. A., & Swain, P. H. (1987). Probabilistic and evidential approaches for multisource data analysis. *IEEE Transactions on Geoscience and Remote Sensing*, *GE-25*(3), 283-293.

Leszeynski, K., Penczek, P., & Grochulskki, W. (1985). Sugeno's fuzzy measure and fuzzy clustering. *Fuzzy Sets and Systems*, *15*(2), 147-158.

Li, H., Liu, Z., & Sun, W. (1993). A new approach to pattern recognition of remote sensing image using artificial neural network. *Proceedings of the International Geoscience and Remote Sensing Symposium*, *2*, 713-715.

McClellan, G. E., DeWitt, R. N., Hemmer, T. H., Matheson, L. N., & Moe, G. O. (1989). Multispectral image processing with a three-layer backpropagation network. *Proceedings of the International Joint Conference on Neural Networks*, *1*, 151-153.

Moody, A., Gopal, S., Strahler, A. H., Borak, J., & Fisher, P. (1994). A combination of temporal tresholding and neural network methods for classifying multiscale remotely-sensed image data. *Proceedings of the International Geoscience and Remote Sensing Symposium*, *4*, 1877-1879.

Murai, H., & Omatu, S. (1997). Remote sensing image analysis using a neural network and knowledge-based processing. *Internatioal Journal of Remote Sesnsing*, *18*(4), 811-828.

Omatu, S., & Yoshida, T. (1993). Pattern classification for remote sensing using neural network. *Proceedings of the International Geoscience and Remote Sensing Symposium*, *2*, 899-901.

Pao, Y.-H. (1989). *Adaptive pattern recognition and neural networks.* Reading, MA USA: Addison-Wesley Publishing Company, Inc.

Paola, J. D., & Schowengerdt, R. A. (1995). A review and analysis of backpropagation neural networks for classification of remotely-sensed multi-spectral imagery. *International Journal of Remote Sensing*, *16*(16), 3033-3058.

Parker, D., & Wolff, M. (1965). Remote sensing. *International Science and Technology*, *43*, 20-31.

Parker, J. R. (1997). In *Algorithms for image processing and computer vision* (p. 330-338). New York NY: John Wiley & Sons.

PCI. (1997). *PACE atmospheric correction* (Software Manual No. 6.2). Richmond Hill, ON CDN: PCI Geomatics.

Prokop, E. J., & Marchette, D. J. (2000). Evaluation of classifier boosting. *Proceedings of SPIE - The International Society for Optical Engineering, Applications and Science of Computational Intelligence III*, 49-56.

Quilan, J. R. (1993). *C4.5: Programs for machine learning.* The Morgan Kaufmann Series in Machine Learning.

Rogova, G. (1994). Combining the results of several neural network classifiers. *Neural Networks*, *7*(5), 777-781.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & M. J. L. (Eds.), *Parallel distributed processing: Explorations in the microstructures of cognition* (Vol. 1, p. 318-362). Cambridge, MA USA: The MIT Press.

Serpico, S. B., Bruzzone, L., & Roli, F. (1996). An experimental comparison of neural and statistical non-parametric algorithms for supervised classification of remote-sensing images. *Pattern Recognition Letters*, *17*(13), 1331-1341.

Shafer, G. (1976). *A mathematical theory of evidencce.* Princeton NJ: Princeton University Press.

Storr, D., & Ferguson, H. L. (1972). The distribution of precipitation in some mountainous Canadian watersheds. In *Geilo symposium - distribution of precipitation in mountainous areas* (Vol. 1, p. 243-263). Geneva: World Meterological Organization, WMO/OMM No. 326.

Sugeno, M. (1977). Fuzzy measure and fuzzy integrals - A survey. In *Fuzzy automata and decision processes* (p. 89-102). Amsterdam, The Netherlands: North-Holland Publishing Company.

Tenorio, M. F., Safavian, S. R., & Kassebaum, J. (1990). Multisource remote sensing data analysis with self-organizing neural networks. *Proceedings of the International Geoscience and Remote Sensing Symposium*, *2*, 1289-1292.

Tumer, K., & Ghosh, J. (1996). Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, *29*(2), 341-348.

Tzeng, Y. C., Chen, K. S., Kao, W.-L., & Fung, A. K. (1994). A dynamic learning neural network for remote sensing applications. *IEEE Transactions on Geoscience and Remote Sensing*, *32*(5), 1096-1102.

Vieira, C. A. O., & Mather, P. M. (2000). A comparative study of multiple classifier combination methods in remote sensing. *Proceedings of the IC - Artificial Intelligence*, *1*, 39-45.

Xu, L., Krzyżak, A., & Suen, C. Y. (1992). Methods of combining multiple classifiers and their applications to handwritten recognition. *IEEE Transactions on Systems, Man, Cybernetics*, *22*(3), 418-435.

Yager, R. R. (1993). Element selection from a fuzzy subset using the fuzzy integral. *IEEE Transactions on Systems, Man, Cybernetics*, *23*(2), 467-477.

Yu, K., Jiang, X., & Bunke, H. (1997). Lipreading: A classifier combination approach. *Pattern Recognition Letters*, *18*, 1421-1426.

Zhuang, X., Engel, B. A., Lozano-Garcia, D. F., Fernández, R. N., & Johannsen, C. J. (1994). Optimization of training data required for neuro-classification. *International Journal of Remote Sensing*, *15*(16), 3271-3277.

Appendix A

Bayesian Combination Methods

Bayes theory assigns an image pixel to class k if

$$k = \arg max_{j=1}^{m} P(c_j|X_1, ..., X_n), \tag{A.1}$$

where the a posteriori class probabilities can be rewritten as

$$P(c_j|X_1, ..., X_n) = \frac{p(X_1, ..., X_n|c_j)P(c_j)}{p(X_1, ..., X_n)}, \tag{A.2}$$

and the unconditional input joint probability density is equal to

$$p(X_1, ..., X_n) = \sum_{k=1}^{m} p(X_1, ..., X_n|c_j)P(c_k). \tag{A.3}$$

A.1 Product rule

The product rule is derived from the equations of the last section by first assuming that the input vectors employed by each network are conditionally statistically independent. This is represented by

$$p(X_1, ..., X_n|c_j) = \prod_{i=1}^{n} p(X_i|c_j). \tag{A.4}$$

Substituting equations (A.4) and (A.3) into (A.2) gives

$$P(c_j|X_1, ..., X_n) = \frac{P(c_j)\prod_{i=1}^{n} p(X_i|c_j)}{\sum_{k=1}^{m} P(c_k)\prod_{i=1}^{n} p(X_i|c_k)}, \tag{A.5}$$

which in turn is substituted into equation (A.1) resulting in the pixel being assigned to class k if

$$k = \arg max_{j=1}^{m} P(c_j) \prod_{i=1}^{n} p(X_i|c_j). \tag{A.6}$$

Note that only the numerator of equation (A.5) is of any importance for this substitution. Finally, rewriting equation (A.6) in terms of each classifier's a posteriori class probabilities by first noting that

$$P(c_j|X_i) = \frac{p(X_i|c_j)P(c_j)}{p(X_i)} \tag{A.7}$$

from equation (A.2). Using only the numerator of the above equation and solving for $p(X_i|c_j)$ gives

$$p(X_i|c_j) = \frac{P(c_j|X_i)}{P(c_j)}. \tag{A.8}$$

substituting equation (A.8) into (A.6) yields the *product rule*

$$\begin{aligned} P(c_j)\prod_{i=1}^{n} p(X_i|c_j) &= max_{j=1}^{m} P(c_j)\prod_{i=1}^{n}\frac{P(c_j|X_i)}{P(c_j)} \\ &= max_{j=1}^{m}\frac{P(c_j)}{\prod_{i=1}^{n} P(c_j)}\prod_{i=1}^{n} P(c_j|X_i) \\ &= max_{j=1}^{m} P^{(1-n)}(c_j)\prod_{i=1}^{n} P(c_j|X_i) \\ &= max_{j=1}^{m} P^{-(n-1)}(c_j)\prod_{i=1}^{n} P(c_j|X_i). \end{aligned} \tag{A.9}$$

Assign the image pixel to class k if

$$k = \arg max_{j=1}^{m} P^{-(n-1)}(c_j)\prod_{i=1}^{n} P(c_j|X_i). \tag{A.10}$$

A.2 Summation rule

For the summation rule, yet another assumption is made: the computed a posteriori class probabilities do not deviate dramatically from the priori class probabilities. This is mathematically represented by

$$P(c_j|X_i) = P(c_j)(1+\varepsilon_{ji}), \tag{A.11}$$

where $\varepsilon_{ji} << 1$. By substituting the above equation into (A.10) it follows that

$$\begin{aligned} P^{-(n-1)}(c_j)\prod_{i=1}^{n}P(c_j|X_i) &= P^{-(n-1)}(c_j)\prod_{i=1}^{n}P(c_j)(1+\varepsilon_{ji}) \\ &= P^{-(n-1)}(c_j)P^{n}(c_j)\prod_{i=1}^{n}(1+\varepsilon_{ji}) \\ &= P(c_j)\prod_{i=1}^{n}(1+\varepsilon_{ji}). \end{aligned} \tag{A.12}$$

By expanding the right-hand side of equation (A.12) and ignoring all higher order terms, ε is so small that it would not have any real value, the following is derived

$$\begin{aligned} P(c_j)\prod_{i=1}^{n}(1+\varepsilon_{ji}) &= P(c_j)\left[(1+\varepsilon_{j1})(1+\varepsilon_{j2})...(1+\varepsilon_{jn})\right] \\ &= P(c_j)\left[(1+\varepsilon_{j1}+\varepsilon_{j2})(1+\varepsilon_{j3})...(1+\varepsilon_{jn})\right] \\ &= P(c_j)\left(1+\sum_{i=1}^{n}\varepsilon_{ji}\right) \\ &= P(c_j)+P(c_j)\sum_{i=1}^{n}\varepsilon_{ji}. \end{aligned} \tag{A.13}$$

The next step is to substitute equations (A.13) and (A.11) into (A.10)

$$\begin{aligned} P^{-(n-1)}(c_j)\prod_{i=1}^{n}P(c_j|X_i) &= P^{-(n-1)}(c_j)\prod_{i=1}^{n}P(c_j)(1+\varepsilon_{ji}) \\ &= P(c_j)\prod_{i=1}^{n}(1+\epsilon_{ji}) \\ &= P(c_j)+P(c_j)\sum_{i=1}^{n}\varepsilon_{ji}. \end{aligned} \tag{A.14}$$

ε must now be solved for using equation (A.11)

$$\varepsilon_{ji} = \frac{P(c_j|X_i)}{P(c_j)} - 1. \tag{A.15}$$

ε can now be substituted for in equation (A.14) resulting in the *summation rule*

$$\begin{aligned} P(c_j) + P(c_j)\sum_{i=1}^{n} \varepsilon_{ji} &= P(c_j) + P(c_j)\sum_{i=1}^{n}\left[\frac{P(c_j|X_i)}{P(c_j)} - 1\right] \\ &= P(c_j) - nP(c_j) + P(c_j)\frac{\sum_{i=1}^{n} P(c_j|X_i)}{P(c_j)} \\ &= (1-n)P(c_j) + \sum_{i=1}^{n} P(c_j|X_i). \end{aligned} \quad \text{(A.16)}$$

Assign the image pixel to class k if

$$k = \arg max_{j=1}^{m}\left[(1-n)P(c_j) + \sum_{i=1}^{n} P(c_j|X_i)\right]. \quad \text{(A.17)}$$

A.3 Maximum rule

Before deriving the remaining Bayesian combination schemes, it is necessary to note that

$$\prod_{i=1}^{n} P(c_j|X_i) \leq min_{i=1}^{n} P(c_j|X_i) \leq \frac{1}{n}\sum_{i=1}^{n} P(c_j|X_i) \leq max_{i=1}^{n} P(c_j|X_i), \quad \text{(A.18)}$$

which implies that the sum rule can be approximated by $max_{i=1}^{n} P(c_j|X_i)$ and the product rule by $min_{i=1}^{n} P(c_j|X_i)$. Knowing that $0 \leq P(c_j|X_i) \leq 1$, equation (A.18) can be interpreted as follows using this simple example. Let the following a posteriori class j probabilities be true for three networks: $P(c_j|X_1) = 0.2$, $P(c_j|X_2) = 0.3$ and $P(c_j|X_3) = 0.5$. The multiplication of these values will be less than or equal to the minimum individual value due to principles of multiplying numbers within the interval [0,1]:

$$\prod_{i=1}^{n} P(c_j|X_i) = 0.03 \leq min_{i=1}^{n} P(c_j|X_i) = 0.2. \quad \text{(A.19)}$$

The minimum individual value will be less than or equal to the average of all values, which in turn will be less than or equal to the maximum individual value because the average always falls within the interval [min,max]:

$$min_{i=1}^{n} P(c_j|X_i) = 0.2 \leq \frac{1}{n}\sum_{i=1}^{n} P(c_j|X_i) = 0.\overline{3} \leq max_{i=1}^{n} P(c_j|X_i) = 0.5. \quad (A.20)$$

In order to derive the maximum combination algorithm, the sum term in equation (A.17) must first be approximated by the maximum of a posteriori class probabilities

$$(1-n)P(c_j) + \sum_{i=1}^{n} P(c_j|X_i) = (1-n)P(c_j) + max_{i=1}^{n} P(c_j|X_i). \quad (A.21)$$

By applying the assumption of equal priors, the $P(c_j)$ and $P(c_k)$ terms of equation (A.17) may be omitted and the *maximum rule* is produced.

Assign the image pixel to class k if

$$k = \arg max_{j=1}^{m}\{max_{i=1}^{n} P(c_j|X_i)\}. \quad (A.22)$$

A.4 Minimum rule

Following the same principles as the maximum rule, the product term in equation (A.10) is first bounded from above giving

$$P^{-(n-1)}(c_j)\prod_{i=1}^{n} P(c_j|X_i) = P^{-(n-1)}(c_j)\ min_{i=1}^{n} P(c_j|X_i). \quad (A.23)$$

Applying once more the assumption of equal priors, the *minimum rule* is formulated.

Assign the image pixel to class k if

$$k = \arg max_{j=1}^{m}\{min_{i=1}^{n} P(c_j|X_i)\}. \quad (A.24)$$

A.5 Average rule

The summation rule can be interpreted as computing an average of a posteriori class probabilities over all networks when the assumption of equal priors is applied. The $P(c_j)$ and $P(c_k)$ terms of equation (A.17) are once again removed and the sum term is transformed into an average computation. This results in the following *average rule*.

Assign the image pixel to class k if

$$k = \arg max_{j=1}^{m} \frac{1}{n} \sum_{i=1}^{n} P(c_j|X_i). \tag{A.25}$$

A.6 Median rule

One criticism of the average rule is that if a network were to output an outlier, the average would be affected negatively. This could cause many incorrect classifications. A robust approximation of the mean is well known to be the median (Kittler et al., 1998). For the following *median rule*, the average computation found in equation (A.25) is replaced by a median value.

Assign the image pixel to class k if

$$k = \arg max_{j=1}^{m} \{med_{i=1}^{n} P(c_j|X_i)\}. \tag{A.26}$$

Appendix B

Atmospheric Correction Parameters

Atmospherically corrected Landsat-TM data is in the form of digital numbers (DNs) ranging from 0 to 255. A DN is assigned to each pixel in an image file and represents its brightness level. The next step consists of converting each DN value to percentages of reflectance. The correlation between DNs and reflectance is an incremental value of 0.257%. Every pixel's DN value was therefore multiplied by 0.257. The final range of reflectance for an image is 0% to 65.535% (PCI, 1997). The parameters utilised in PCI Geomatics' EASI/PACE® Atmospheric Correction package are outlined in tables B.1 to B.4. A deep water pixel located at (770,1633) on figure 3.2 was used as the reference target.

Determination of Ground Visibility (ATCOR0)	
FILE	/data/dlcw/kan$_e$cology.jun.02
AFILE	/data/pci/aausrura.atx
DBIC	1-5,7
SBAND	1-5,7
DBIW	default
CALIB1	-0.15, 0.0795
CALIB2	-0.05, 0.15
CALIB3	-0.05, 0.12
CALIB4	-0.12, 0.1
CALIB5	0, 0.128
CALIB6	0, 0.0075
SENSOR	TM
TFLSZ	3,3
AFLSZ	8,8
REFLECT	4,5,3,1,1,1
AELEV	1000
ZNANGLE	40.7
DATE	8,8,1984
REPORT	/export/home/dcwatts/School/AtmosCor/Atcr0.txt

Table B.1: Parameters used for determining ground visibility using the Atmospheric Correction's ATCOR0 task.

Creation of the Reflectance Image without Adjacency-Effect (ATCOR1)	
FILE	/data/dlcw/kan$_e$cology.jun.02
AFILE	/data/pci/aausrura.atx
DBIC	1-5,7
SBAND	1-5,7
DBOC	8-13
DBIW	default
CALIB1	-0.15, 0.0795
CALIB2	-0.05, 0.15
CALIB3	-0.05, 0.12
CALIB4	-0.12, 0.1
CALIB5	0, 0.128
CALIB6	0, 0.0075
SENSOR	TM
VISIBIL	40
AELEV	1000
ZNANGLE	40.7
DATE	8,8,1984

Table B.2: Parameters used for creating a reflectance image using the Atmospheric Correction's ATCOR1 task without taking into account the adjacency effect.

Average Filtering (FAV)	
FILE	/data/dlcw/kan$_e$cology.jun.02
DBIC	8-13 (individually)
DBOC	14-19 (individually)
FLSZ	3,3
MASK	default

Table B.3: Parameters used for average filtering using the Image Processing's FAV task.

Creation of the Reflectance Image with Adjacency-Effect (ATCOR2)	
FILE	/data/dlcw/kan$_{e}$cology.jun.02
AFILE	/data/pci/aausrura.atx
DBIC	1-5,7
DBFC	14-19
SBAND	1-5,7
DBOC	20-25
DBIW	default
SENSOR	TM
VISIBIL	40
AELEV	1000
ZNANGLE	40.7

Table B.4: Parameters used for creating a reflectance image using the Atmospheric Correction's ATCOR2 task taking the adjacency effect into account.

Appendix C

Back-Propagation Training Algorithm with Momentum

The ANN software used in this research was developed by Mr. Kevin Croft of the University of Calgary's Geomatics Engineering Department. The software, Dyn_BpNN, was written in C++ code and is based on the BP algorithm found in Pao (1989). (Downloadable from http://orb.geomatics.ucalgary.ca/.) It should be noted that the network weights were only updated after every pixel in the training set had been propagated forward through the network once. This is referred to as epochwise training (Pao, 1989). The epoch size for every network in this research was therefore set to equal the number of pixels in the training set - 4153.

Step 1 Initialise the iteration counter k and weight change accumulators $\Delta w_{IJ}^{(l)}$ to zero.

Step 2 Do steps 2a to 2d for all training pixels - for ρ_ι, $\iota = 1, ..., 4153$

{

Step 2a Propagate $X_i(\rho_\iota)$ forward through network i to find its output $Y_i(\rho_\iota)$.

Step 2b Compare the computed output to the true output for all pixels, resulting in the errors $e_j(\rho_\iota)$ for all classes $j = 1, ..., m$.

Step 2c Propagate the error vector

$$e(\rho_\iota) = [e_1(\rho_\iota), ..., e_m(\rho_\iota)]^T \tag{C.1}$$

backwards through the network to compute the "d" value for each node. The pixel notation, ρ_ι, has been removed for simplicity.

For the output layer L,

$$d_I^{(L)} = e_I f'(s_I^{(L)}). \tag{C.2}$$

For layers $l = L - 1, ..., 1$,

$$d_I^{(l)} = f'(s_I^{(l)}) \sum_{I=1}^{t_{l+1}} d_I^{(l+1)} w_{IJ}^{(l+1)}. \tag{C.3}$$

The derivative of the transfer function is denoted by $f'()$. The logistic sigmoid function is utilised in this research. It's derivative is equal to

$$f'(\text{logsig}(\chi)) = \text{logsig}(\chi)(1 - \text{logsig}(\chi)). \tag{C.4}$$

Step 2d Update the weight change accumulators for all weights in the network,

$$\Delta w_{IJ}^{(l)} = \Delta w_{IJ}^{(l)} - 0.9 d_I^{(l)} a_J^{(l-1)}. \tag{C.5}$$

The learning rate has been set to a value of 0.9 in equation (C.5).

}

Step 3 Update all weights in the network,

$$w_{IJ}^{(l)}(k+1) = w_{IJ}^{(l)}(k) - \Delta w_{IJ}^{(l)} + 0.1 \Delta w_{IJ}^{(l)}(k-1). \tag{C.6}$$

The momentum has been set to a value of 0.1 in equation (C.6).

Step 4 Reset the weight accumulators to zero, $\Delta w_{IJ}^{(l)} = 0$.

Step 5 Increase the number of training iterations by 1, $k = k + 1$.

Step 6 Go back to step 2 if stopping criteria is not met, else end training.