# Improved Learning Strategies for Multimodular Fuzzy Neural Network Systems: A Case Study on Image Classification

Steven A. Israel
Nikola K. Kasabov

**University of Otago**
**Department of Information Science**
**PO Box 56**
**Dunedin, New Zealand**
***israel@spheroid.otago.ac.nz***
***nkasabov@otago.ac.nz***

***Abstract***

*This paper explores two different methods for improved learning in multimodular fuzzy neural network systems for classification. It demonstrates these methods on a case study of satellite image classification using 3 spectral inputs and 10 coastal vegetation covertype outputs. The classification system is a multimodular one; it has one fuzzy neural network per output. All the fuzzy neural networks are trained in parallel for a small number of iterations. Then, the system performance is tested on new data to determine the types of interclass confusion. Two strategies are developed to improve classification performance. First, the individual modules are additionally trained for a very small number of iterations on a subset of the data to decrease the false positive and the false negative errors. The second strategy is to create new units, 'experts', which are individually trained to discriminate only the ambiguous classes. So, if the main system classifies a new input into one of the ambiguous classes, then the new input is passed to the 'experts' for final classification. Two learning techniques are presented and applied to both classification performance enhancement strategies; the first one reduces omission, or false negative, error; the second reduces comission, or false positive, error. Considerable improvement is achieved by using these learning techniques and thus, making it feasible to incorporate them into a real adaptive system that improves during operation.*

## 1. Introducing the case study of satellite image classification

### 1.1 Sampling image data for the experiment

A System Pour l'Observation de la Terre (SPOT) image of the Otago Harbour, Dunedin, New Zealand, provided the inputs for the classification. The SPOT image has 20 metre spatial resolution and 3 spectral bands sensing the green, red and infrared portions of the electromagnetic spectrum. Ten covertypes, containing intertidal vegetation and substrates, were recorded during a ground reference survey. From the SPOT image, three spatially separable reference areas were extracted for each of ten covertypes. All of the sample pixels for a given covertype were amalgamated and randomly sorted into training and test sets.

### 1.2 Natural confusion among classes

The biggest problem with mapping natural systems (inputs) to human determined classes (outputs) is that some confusion will occur. There are 2 major types of confusion: (1) errors of omission, false negative errors, and (2) errors of comission, false positive errors. For the case study problem, considerable confusion exists among classes 3, 4 and 5 (*hisand*, *lowsand* and *lowzost* respectively). To graphically illustrate this confusion among the classes, scatterplots were produced (Figure 1, Figure 2, and Figure 3)

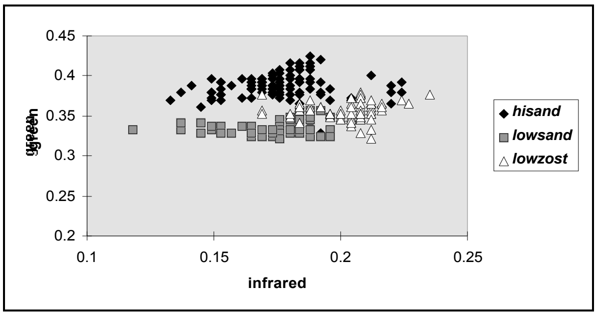


**Figure 1** Scatterplot of 3 Sample Classes (infrared versus green feature space)

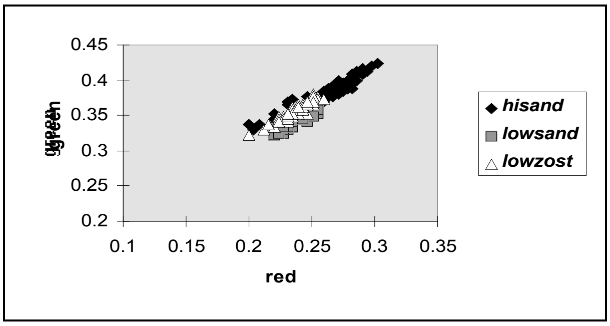


**Figure 2** Scatterplot of 3 Sample Classes (red versus green feature space)

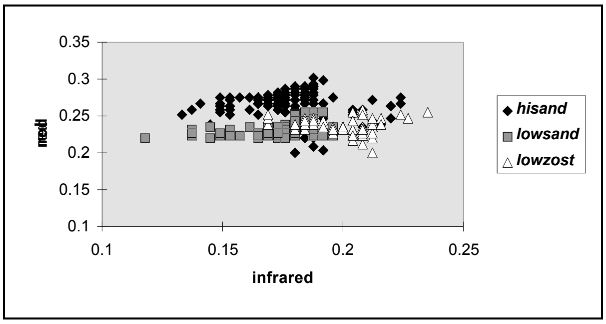


**Figure 3** Scatterplot of 3 Sample Classes (infrared versus red feature space)

### 1.3 Classification

For image classification, a variety of different algorithms have been explored, among them: statistical methods, connectionist methods, and fuzzy inference methods. Independent of the algorithm used, sample classification is determined as the highest score among the individual class transfer functions that associate inputs to outputs. The most common traditional classifier is the maximum likelihood classifier [1,2,3]. It operates by performing 2 passes through the data. The first pass creates the transfer function between the input and output classes; the second one performs the classification. Statistical techniques in general are not adaptive. For example, in order for a system to adapt to new data, the entire training set and new data must be analysed and updated in an iterative process [4].

Connectionist systems have been used for classification [5,6,7,8,9,18]. These systems require a large number of iterations to adjust connection weights for error minimisation. This coupled with large training sets associated with image reference data, places a large processing load on computing resources. Very often, connectionist systems cannot be adapted to improve the performance on individual classes. These systems are designed to reduce the RMS error of the neural networks which does not insure a corresponding increase in classification accuracy.

Fuzzy inferences methods have been used extensively for pattern recognition [9,11]. The difficulty with fuzzy systems is the vaguely known expert fuzzy rules and membership functions must be adapted to the new data. However, once the systems are created, classification is very efficient. Fuzzy inferences have also been combined with connectionist techniques as hybrid techniques [10,13,18]. Fuzzy rules may be extracted from hybrid systems to determine what the system has learned [13,17,18]. Previous experiments compared the utility of connectionist-based systems to conventional parametric classifiers and to fuzzy classifiers [9,11,14,15,16]. However, this research will introduce effective methods to reduce interclass confusion and to improve adaptation.

## 2. Fuzzy Neural Networks

### 2.1 Different types of fuzzy neural networks

A fuzzy neural network (FNN) is a connectionist model for fuzzy rules implementation and inference. There are a wide variety of architectures and functionalities of FNN [10,17,18,21]. They differ mainly in the following parameters:

- *type of fuzzy rules* implemented; this reflects in the connectionist structure used;
- *type of inference method* implemented; this reflects in the selection of different neural network parameters and neuronal functions, such as summation, activation, output function; it also influences the way the connection weights are initialised before training, and interpreted after training;
- *mode of operation*; we shall consider here three major modes of operation :

⇒ *Fixed mode* - fixed membership functions-fixed set of rules, i.e. a fixed set of rules is inserted in a network; the network performs inference, but does not change its weights. It cannot learn and adapt. It does not forget either.

⇒ *Learning mode, i.e.,* a neural network is structurally defined to capture knowledge in a certain format, e.g., some type of fuzzy rules. The network architecture is randomly initialised and trained with a set of data. Rules are then extracted from the structured network. The rules can be interpreted either in the same network structure or by using other inference methods.

⇒ *Adaptation mode* - A neural network is either randomly initialised or structurally set according to a set of fuzzy rules, 'hints', and heuristics. The network is then trained with data and updated fuzzy rules are extracted from its structure following some rule extraction algorithm. The rules can either be interpreted in a fuzzy inference engine or can be inserted back to the fuzzy neural network structure in the same way initial set of rules have been inserted. The network is further trained with new data and new updated rules extracted, etc.

The FNN model [17,18] facilitates learning from data, fuzzy rules extraction, fuzzy rules insertion, approximate reasoning, and adaptation. FNN uses a multi-layered perceptron (MLP) network and a backpropagation training algorithm. It is an adaptable FNN where the membership functions of the fuzzy predicates, as well as the fuzzy rules inserted before training (adaptation), may adapt and change according to the training data. The general architecture of FNN consists of five layers. Figure 4 depicts a FNN for two exemplar fuzzy rules [17,18].

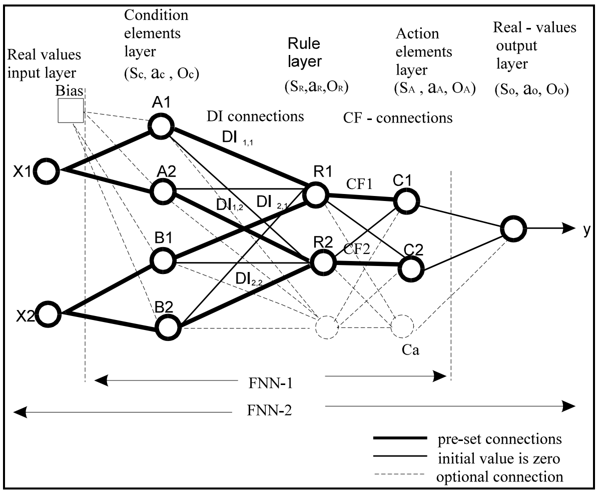


**Figure 4** A FNN structure for the two fuzzy rules Explicitly: $R_1$: IF $x_1$ is $A_1$ ($DI_{1,1}$) and $x_2$ is $B_1$ ($DI_{2,1}$) THEN y is $C_1$ ($CF_1$); $R_2$: IF $x_1$ is $A_2$ ($DI_{1,2}$) and x2 is $B_2$ ($DI_{2,2}$) THEN y is $C_2$ ($CF_2$), where DIs are degrees of importance attached to the condition elements an CFs are certainty factors attached to the consequent parts of the rules [17,18].

In the following experiments, FNN that consist of only the condition element layer, the rule layer and the action element layer are considered. The membership functions are defined by the user. For the experiments in the next section, the membership functions are of the standard triangular type with a uniform distribution over the universe of discourse. Fuzzification and defuzzification are performed outside the structure.

### 2.2 Rules extraction from fuzzy neural networks

One of the advantages of fuzzy neural networks is that structured information (knowledge) can be inserted and extracted. A FNN can be interpreted in linguistic terms after training. The structure of a FNN also restricts the information (knowledge) representation and interpretation.

An algorithm called REFuNN (Rules Extraction From Neural Networks) for rules extraction from a trained FNN is presented in [17,18]. The method is based on the following assumptions: simple operations are used and a low computational cost achieved; hidden nodes in a MLP learn features, rules, and groups in the training data; fuzzy quantisation of the input and the output variables is performed outside the algorithm; automatically extracted rules require additional manipulation depending on the reasoning method applied afterwards. The algorithm uses thresholds above which connection weights are kept and represented in a linguistic form as fuzzy rules.

### 2.3 Towards adaptive learning strategies in modular connectionist structures

A general architecture of an adaptive multimodular system is given in Figure 5. It consists of: (1) a classification module built as a multimodular system, (2) a module for training, modification and adaptation, and (3) a module for rules extraction.

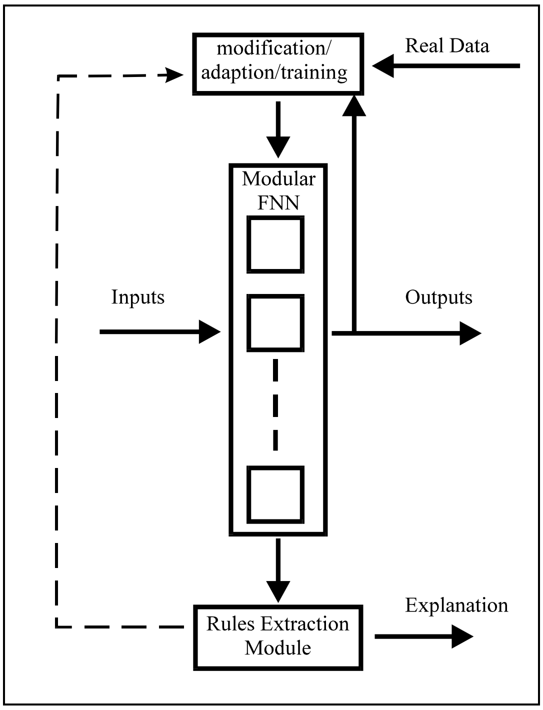


**Figure 5** A general architecture of an adaptive intelligent multimodular classifier

This paper discusses the classification module only. The classification module has a single connectionist unit for each of the output classes. Initially, all the modules are trained with identical data. After that, each of the units can be tuned using learning techniques. This approach has been presented in [18] and illustrated on a phoneme classification task where different learning strategies have been experimented, such as: additional training of class units with negative examples to suppress false positive error; using averaged over three time units data. These methods are further developed and experimented here on an image classification case problem.

## 3. A Classification System based on Multimodular Fuzzy Neural Network Classifier: Initial System for the Case Study Problem

### 3.1 Image classification in a multimodular fuzzy-neural network system

The classification module of the architecture given in Figure 5 is implemented here for the case study problem. The multimodular sub-system has one fuzzy neural network per output class. The number of the membership functions are fixed. The input variables are quantised into 5 membership functions and the output variables (the class variables) are quantised into 2 membership functions for this case study (Figure 6).

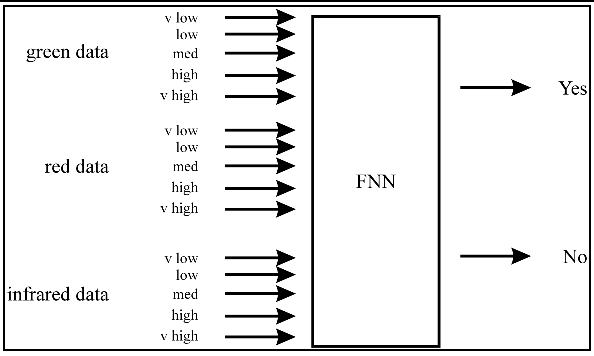


**Figure 6a**

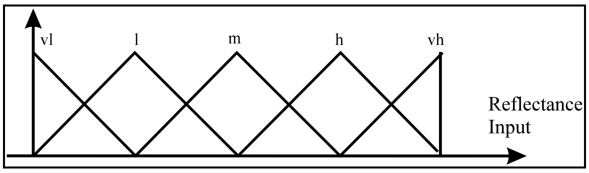


**Figure 6b**

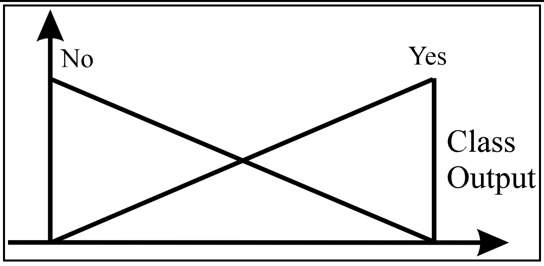


**Figure 6c**

**Figure 6** (a) The single module classifier for the case study problem of image classification; (b) Membership functions of the input variables and (c) the output, class variables

## 3.2 Training and testing the multimodular neuro-fuzzy classifier

All the neural networks are trained in parallel for a small number of iterations. Then, the performance of the system is tested and the poorly performing class units are identified. Training and testing sets are identical for all classes.

| | training data | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| random sort data 10 fuzzified neural network 15-10-2 in parallel - maximum of 200 iterations or 0.001 rms error | | | | | | | | | | | | | |
| | **denzost** | **dryzost** | **hisand** | **lowsand** | **lowzost** | **red** | **shalsub** | **shell** | **shellzost** | **wetzost** | | sums | percent |
| **denzost** | **251** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | | 256 | 98 |
| **dryzost** | 2 | **40** | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 5 | | 49 | 82 |
| **hisand** | 0 | 0 | **102** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | 103 | 99 |
| **lowsand** | 0 | 0 | 0 | **15** | 0 | 0 | 0 | 0 | 0 | 0 | | 15 | 100 |
| **lowzost** | 0 | 1 | 24 | 67 | **77** | 0 | 0 | 2 | 0 | 1 | | 172 | 45 |
| **red** | 1 | 0 | 0 | 0 | 0 | **303** | 0 | 0 | 0 | 0 | | 304 | 100 |
| **shalsub** | 0 | 0 | 0 | 0 | 0 | 0 | **58** | 0 | 0 | 0 | | 58 | 100 |
| **shell** | 0 | 2 | 0 | 0 | 0 | 0 | 0 | **100** | 0 | 0 | | 102 | 98 |
| **shellzost** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | **64** | 2 | | 67 | 96 |
| **wetzost** | 10 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 8 | **80** | | 101 | 79 |
| | | | | | | | | | | | **1090** | | |
| sum | 264 | 43 | 126 | 83 | 78 | 306 | 58 | 103 | 76 | 90 | | **1227** | |
| percent | 95 | 93 | 81 | 18 | 99 | 99 | 100 | 97 | 84 | 89 | | | **88.83456** |
| *iterations* | *200* | *200* | *200* | *200* | *200* | *200* | *26* | *200* | *200* | *200* | | | |
| | | | | | | | | | | | | | |
| | test data | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | **denzost** | **dryzost** | **hisand** | **lowsand** | **lowzost** | **red** | **shalsub** | **shell** | **shellzost** | **wetzost** | | sums | percent |
| **denzost** | **129** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | 130 | 99 |
| **dryzost** | 0 | **16** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | 17 | 94 |
| **hisand** | 0 | 0 | **50** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 50 | 100 |
| **lowsand** | 0 | 0 | 0 | **5** | 0 | 0 | 0 | 0 | 0 | 0 | | 5 | 100 |
| **lowzost** | 0 | 2 | 12 | 35 | **38** | 0 | 0 | 3 | 0 | 0 | | 90 | 42 |
| **red** | 0 | 0 | 0 | 0 | 0 | **149** | 0 | 0 | 0 | 0 | | 149 | 100 |
| **shalsub** | 0 | 0 | 0 | 1 | 0 | 0 | **29** | 0 | 0 | 0 | | 30 | 97 |
| **shell** | 0 | 3 | 0 | 0 | 0 | 0 | 0 | **47** | 0 | 0 | | 50 | 94 |
| **shellzost** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | **33** | 2 | | 36 | 92 |
| **wetzost** | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | **43** | | 47 | 91 |
| | | | | | | | | | | | **539** | | |
| sums | 130 | 21 | 62 | 41 | 38 | 151 | 29 | 50 | 37 | 45 | | **604** | |
| percent | 99 | 76 | 81 | 12 | 100 | 99 | 100 | 94 | 89 | 96 | | | **89.23841** |

**Figure 7** Initial Confusion matrix of the 10 classes

Note the confusion between classes 3, 4, and 5 (*hisand*, *lowsand*, and *lowzost* respectively). The reduction of this confusion will be the discussed further. The columns represent the input classes and the rows represent the "as classed" values.

The reflectance information from each of the satellites spectral bands (inputs for the networks) were fuzzified. This effectively increased the ratio between inputs and outputs. The initial training data was identical for each network. All connectionist-based processing was performed using a hybrid software environment FuzzyCOPE [13]. Acceptable conversion tolerance and error were considerably less than the optimum tolerance and required fewer iterations. For this case, acceptable training was chosen to be an RMS error of 0.001 or 200 iterations for each class. Figure 7 shows the confusion matrix for the training and test data for the ten output classes. Intuitively, the greater number of samples, the fewer iterations are required.

After the interclass confusion was identified, the next step of the process was to reduce it. For this, two strategies are described. The first is highly specific additional training on the ambiguous class modules. The second is the creation of new expert systems for those classes where sufficient confusion exists. For both strategies, two learning techniques were applied. The learning techniques involve the reduction of comission and omission errors.

# 4. Quick Additional Training of Individual Class Networks for Reducing Comission and Omission Errors

The first strategy operates on the existing ambiguous networks. The individual networks contain sufficient information so that only tuning is required. The ambiguous classes are *hisand*, *lowsand*, and *lowzost*. The additional training was performed to reduce the confusion among these classes.

## 4.1 Comission correction

Once the system was trained, one can identify the confusion between classes. Figure 8 shows the dramatic improvement of the training and test error when one additional training iteration is performed on the class “lowzost” network with all the negative examples used for the initial training in Figure 7. This was the first technique applied.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | training data | | | | | | | | | | | |
| random sort data 10 fuzzified neural network 15-10-2 in parallel - maximum of 200 iterations or 0.001 rms error + 1 it. class 5 "not" examples class 3 and | | | | | | | | | | | | |
| | **denzost** | **dryzost** | **hisand** | **lowsand** | **lowzost** | **red** | **shalsub** | **shell** | **shellzost** | **wetzost** | | sums | percent |
| **denzost** | **251** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | | 256 | 98 |
| **dryzost** | 2 | **41** | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 5 | | 51 | 80 |
| **hisand** | 0 | 0 | **118** | 0 | 15 | 0 | 0 | 0 | 0 | 0 | | 133 | 89 |
| **lowsand** | 0 | 0 | 0 | **77** | 2 | 0 | 0 | 0 | 0 | 0 | | 79 | 97 |
| **lowzost** | 0 | 0 | 4 | 5 | **59** | 0 | 0 | 0 | 0 | 0 | | 68 | 87 |
| **red** | 1 | 0 | 0 | 0 | 0 | **303** | 0 | 0 | 0 | 0 | | 304 | 100 |
| **shalsub** | 0 | 0 | 3 | 0 | 0 | 0 | **58** | 0 | 0 | 0 | | 61 | 95 |
| **shell** | 0 | 2 | 0 | 0 | 1 | 0 | 0 | **102** | 0 | 1 | | 106 | 96 |
| **shellzost** | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | **64** | 2 | | 68 | 94 |
| **wetzost** | 10 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 8 | **80** | | 101 | 79 |
| | | | | | | | | | | | **1153** | | |
| sum | 264 | 43 | 126 | 83 | 78 | 306 | 58 | 103 | 76 | 90 | | **1227** | |
| percent | 95 | 95 | 94 | 93 | 76 | 99 | 100 | 99 | 84 | 89 | | | **93.96903** |
| *iterations* | *200* | *200* | *200* | *200* | *201* | *200* | *26* | *200* | *200* | *200* | | | |
| | | | | | | | | | | | | | |
| | test data | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | **denzost** | **dryzost** | **hisand** | **lowsand** | **lowzost** | **red** | **shalsub** | **shell** | **shellzost** | **wetzost** | | sums | percent |
| **denzost** | **129** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | 130 | 99 |
| **dryzost** | 0 | **18** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | 19 | 95 |
| **hisand** | 0 | 0 | **60** | 0 | 8 | 0 | 0 | 0 | 0 | 0 | | 68 | 88 |
| **lowsand** | 0 | 0 | 0 | **30** | 0 | 0 | 0 | 0 | 0 | 0 | | 30 | 100 |
| **lowzost** | 0 | 0 | 1 | 10 | **29** | 0 | 0 | 0 | 0 | 0 | | 40 | 73 |
| **red** | 0 | 0 | 0 | 0 | 0 | **149** | 0 | 0 | 0 | 0 | | 149 | 100 |
| **shalsub** | 0 | 0 | 0 | 1 | 0 | 0 | **29** | 0 | 0 | 0 | | 30 | 97 |
| **shell** | 0 | 3 | 1 | 0 | 1 | 0 | 0 | **50** | 0 | 0 | | 55 | 91 |
| **shellzost** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | **33** | 2 | | 36 | 92 |
| **wetzost** | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | **43** | | 47 | 91 |
| | | | | | | | | | | | **570** | | |
| sums | 130 | 21 | 62 | 41 | 38 | 151 | 29 | 50 | 37 | 45 | | **604** | |
| percent | 99 | 86 | 97 | 73 | 76 | 99 | 100 | 100 | 89 | 96 | | | **94.37086** |

**Figure 8** Advantageous forgetting using all classes; 1 iteration of training with “not” examples for class 5 (lowzost) network

## 4.2 Omission correction

The second technique uses synthetically generated data based upon the training set’s parametric information. The synthetic data produced ‘yes’ examples bounded by ‘not’ examples. Figure 9 shows the results after training the *hisand* and *lowsand* networks with positive examples only.

Positive examples were randomly generated in an “n” dimensional space 1σ from the cluster mean, “n” is the number of inputs. Negative examples were randomly ‘placed’ in a ring 2σ from the cluster ring with a random variation of 2%. Additional individual training was performed separately for each class. At most, 16 iterations were required. This approach of placing examples in a neighbourhood is similar to the Mexican hat paradigm of updating the connections weights in the Kohonen self organising networks [19].

- For positive examples the equation is:

$$\bar{x}(i) + \sigma(i) * Rand(\ ) * (-1)\ \verb|^|Int(Rand(\ ))$$

- For negative examples the equation is:

$$\bar{x}(i)+[2\sigma(i) + (\sigma(i)*Rand(\ ))/50)] * (-1)\ \verb|^|Int(Rand(\ ))$$

where:

| | | |
|---|---|---|
| i | = | the input number (this process is performed for each input) |
| $\bar{x}$ | = | average of all the training samples |
| $\sigma$ | = | standard deviation of the training samples |
| Rand( ) | = | a random number generated between 0 and 1 |
| Int | = | integer rounding of real value |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | training data | | | | | | | | | | | | |
| random sort data 10 fuzzified neural network 15-10-2 in parallel - omission correction class 3 and 4 | | | | | | | | | | | | | |
| | **denzost** | **dryzost** | **hisand** | **lowsand** | **lowzost** | **red** | **shalsub** | **shell** | **shellzost** | **wetzost** | | sums | percent |
| **denzost** | **251** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | | 256 | 98 |
| **dryzost** | 2 | **40** | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 5 | | 49 | 82 |
| **hisand** | 0 | 0 | **103** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | 104 | 99 |
| **lowsand** | 0 | 0 | 0 | **67** | 1 | 0 | 0 | 0 | 0 | 0 | | 68 | 99 |
| **lowzost** | 0 | 1 | 23 | 16 | **76** | 0 | 0 | 2 | 0 | 1 | | 119 | 64 |
| **red** | 1 | 0 | 0 | 0 | 0 | **303** | 0 | 0 | 0 | 0 | | 304 | 100 |
| **shalsub** | 0 | 0 | 0 | 0 | 0 | 0 | **58** | 0 | 0 | 0 | | 58 | 100 |
| **shell** | 0 | 2 | 0 | 0 | 0 | 0 | 0 | **100** | 0 | 0 | | 102 | 98 |
| **shellzost** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **64** | 2 | | 66 | 97 |
| **wetzost** | 10 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 8 | **80** | | 101 | 79 |
| | | | | | | | | | | | **1142** | | |
| sum | 264 | 43 | 126 | 83 | 78 | 306 | 58 | 103 | 76 | 90 | | **1227** | |
| percent | 95 | 93 | 82 | 81 | 97 | 99 | 100 | 97 | 84 | 89 | | | **93.07253** |
| *iterations* | *200* | *200* | *216* | *203* | *200* | *200* | *26* | *200* | *200* | *200* | | | |
| | | | | | | | | | | | | | |
| | test data | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | **denzost** | **dryzost** | **hisand** | **lowsand** | **lowzost** | **red** | **shalsub** | **shell** | **shellzost** | **wetzost** | | sums | percent |
| **denzost** | **129** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | 130 | 99 |
| **dryzost** | 0 | **16** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | 17 | 94 |
| **hisand** | 0 | 0 | **51** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 51 | 100 |
| **lowsand** | 0 | 0 | 0 | **27** | 0 | 0 | 0 | 0 | 0 | 0 | | 27 | 100 |
| **lowzost** | 0 | 2 | 11 | 14 | **38** | 0 | 0 | 3 | 0 | 0 | | 68 | 56 |
| **red** | 0 | 0 | 0 | 0 | 0 | **149** | 0 | 0 | 0 | 0 | | 149 | 100 |
| **shalsub** | 0 | 0 | 0 | 0 | 0 | 0 | **29** | 0 | 0 | 0 | | 29 | 100 |
| **shell** | 0 | 3 | 0 | 0 | 0 | 0 | 0 | **47** | 0 | 0 | | 50 | 94 |
| **shellzost** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | **33** | 2 | | 36 | 92 |
| **wetzost** | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | **43** | | 47 | 91 |
| | | | | | | | | | | | **562** | | |
| sums | 130 | 21 | 62 | 41 | 38 | 151 | 29 | 50 | 37 | 45 | | **604** | |
| percent | 99 | 76 | 82 | 66 | 100 | 99 | 100 | 94 | 89 | 96 | | | **93.04636** |

**Figure 9** Omission correction using positive examples to train class 3 and class 4 networks

Both the omission and comission corrections operate very quickly. In this experiment, presenting the system with negative (not) examples required 1 iteration and 16 iterations at the most for the positive (yes) examples.

# 5. Creating Individual 'Expert' Class Networks

If a significant amount of confusion existed between a small number of classes, the following procedures were followed. A new set of network 'experts' were trained using a reduced number of classes (the ambiguous classes only) for a small number of iterations. The training data was limited to the training data of those covertypes alone. If the initial system classifies a new input into one of the ambiguous classes, then the input was passed to the 'experts' for final classification. Figure 10 shows the confusion table of the classification of training and test data for expert networks for *hisand*, *lowsand* and *lowzost*. The RMS convergence was noted to be higher; however, confusion between classes was less. Training times for the expert systems are significantly less due to the smaller training set. As with the first strategy to improve classification performance, learning data will be presented to reduce both omission and comission error.

| | | | | | | |
|---|---|---|---|---|---|---|
| | training data | | | | | |
| | random sort (3) 15-10-2 fuzzy neural networks in parallel | | | | | |
| | **hisand** | **lowsand** | **lowzost** | | sums | percent |
| **hisand** | **116** | 0 | 1 | | 117 | 99 |
| **lowsand** | 0 | **80** | 2 | | 82 | 98 |
| **lowzost** | 10 | 3 | **75** | | 88 | 85 |
| | | | | **271** | | |
| sum | 126 | 83 | 78 | | **287** | |
| percent | 92 | 96 | 96 | | | **94.42509** |
| *iterations* | *200* | *200* | *300* | | | |
| | | | | | | |
| | test data | | | | | |
| | | | | | | |
| | **hisand** | **lowsand** | **lowzost** | | sums | percent |
| **hisand** | **53** | 0 | 1 | | 54 | 98 |
| **lowsand** | 0 | **33** | 0 | | 33 | 100 |
| **lowzost** | 9 | 8 | **37** | | 54 | 69 |
| | | | | **123** | | |
| sums | 62 | 41 | 38 | | **141** | |
| percent | 85 | 80 | 97 | | | **87.23404** |

**Figure 10** Newly trained class 'experts' for the classes that contained significant confusion

## 5.1 Expert system - comission correction

A single additional training iteration was performed on the "lowzost class" using 'not' examples from the other ambiguous classes. The results have improved as shown in Figure 11. If required, additional individual training should be performed. For both the initial and expert classifier strategies, the user is taking advantage of system forgetting.

| | training data | | | | | |
|---|---|---|---|---|---|---|
| random sort (3) 15-10-2 networks in parallel + 1 its class 5 with 3 & 4 nots | | | | | | |
| | **hisand** | **lowsand** | **lowzost** | | sums | percent |
| **hisand** | **121** | 0 | 9 | | 130 | 93 |
| **lowsand** | 0 | **83** | 5 | | 88 | 94 |
| **lowzost** | 5 | 0 | **64** | | 69 | 93 |
| | | | | **268** | | |
| sum | 126 | 83 | 78 | | **287** | |
| percent | 96 | 100 | 82 | | | **93.37979** |
| *iterations* | *200* | *200* | *301* | | | |
| | | | | | | |
| | test data | | | | | |
| | | | | | | |
| | **hisand** | **lowsand** | **lowzost** | | sums | percent |
| **hisand** | **58** | 0 | 4 | | 62 | 94 |
| **lowsand** | 0 | **39** | 1 | | 40 | 97 |
| **lowzost** | 4 | 2 | **33** | | 39 | 85 |
| | | | | **130** | | |
| sums | 62 | 41 | 38 | | **141** | |
| percent | 94 | 95 | 87 | | | **92.19858** |

**Figure 11** After additional training of class 5 'expert' network for 1 iteration with all the negative examples on classes 3 and 4

However, additional iterations using negative examples may lead to deteriorating the performance of the trained network expert, as shown in Figure 12 and Figure 13.

| | training data | | | | | |
|---|---|---|---|---|---|---|
| random sort (3) 15-10-2 networks in parallel + 2 its class 5 with 3 & 4 nots | | | | | | |
| | **hisand** | **lowsand** | **lowzost** | | sums | percent |
| **hisand** | **121** | 0 | 14 | | 135 | 90 |
| **lowsand** | 0 | **83** | 11 | | 94 | 88 |
| **lowzost** | 5 | 0 | **53** | | 58 | 91 |
| | | | | **257** | | |
| sum | 126 | 83 | 78 | | **287** | |
| percent | 96 | 100 | 68 | | | **89.54704** |
| *iterations* | *200* | *200* | *302* | | | |
| | | | | | | |
| | test data | | | | | |
| | | | | | | |
| | **hisand** | **lowsand** | **lowzost** | | sums | percent |
| **hisand** | **59** | 0 | 6 | | 65 | 91 |
| **lowsand** | 0 | **41** | 6 | | 47 | 87 |
| **lowzost** | 3 | 0 | **26** | | 29 | 90 |
| | | | | **126** | | |
| sums | 62 | 41 | 38 | | **141** | |
| percent | 95 | 100 | 68 | | | **89.3617** |

**Figure 12** Less advantageous forgetting; 2 iterations of "not" examples to additionally train class 5 network

| | training data | | | | | |
|---|---|---|---|---|---|---|
| random sort (3) 15-10-2 networks in parallel + 5 its. class 5 with 3 & 4 nots | | | | | | |
| | **hisand** | **lowsand** | **lowzost** | | sums | percent |
| **hisand** | **122** | 0 | 25 | | 147 | 83 |
| **lowsand** | 0 | **83** | 13 | | 96 | 86 |
| **lowzost** | 4 | 0 | **40** | | 44 | 91 |
| | | | | **245** | | |
| sum | 126 | 83 | 78 | | **287** | |
| percent | 97 | 100 | 51 | | | **85.36585** |
| *iterations* | *200* | *200* | *305* | | | |
| | | | | | | |
| | test data | | | | | |
| | | | | | | |
| | **hisand** | **lowsand** | **lowzost** | | sums | percent |
| **hisand** | **61** | 0 | 12 | | 73 | 84 |
| **lowsand** | 0 | **41** | 7 | | 48 | 85 |
| **lowzost** | 1 | 0 | **19** | | 20 | 95 |
| | | | | **121** | | |
| sums | 62 | 41 | 38 | | **141** | |
| percent | 98 | 100 | 50 | | | **85.8156** |

**Figure 13** Reduced performance; 5 iterations of "not" examples to additionally train class 5 network

## 5.2 Expert system - omission correction

The omission correction was applied to the *hisand* and *lowsand* networks (Figure 14). The result is an improvement on the initial state (Figure 10). It was noted that as the ratio of positive to negative examples increased, the optimum number of iterations reduced. For this case study, 50 positive to 10 negative examples was empirically determined as optimum.

| | training data | | | | | |
|---|---|---|---|---|---|---|
| random sort (3) 15-10-2 networks in parallel +omission correction | | | | | | |
| | **hisand** | **lowsand** | **lowzost** | | sums | percent |
| **hisand** | **116** | 0 | 3 | | 119 | 97 |
| **lowsand** | 3 | **83** | 6 | | 92 | 90 |
| **lowzost** | 7 | 0 | **69** | | 76 | 91 |
| | | | | **268** | | |
| sum | 126 | 83 | 78 | | **287** | |
| percent | 92 | 100 | 88 | | | **93.37979** |
| *iterations* | *201* | *203* | *301* | | | |
| | | | | | | |
| | test data | | | | | |
| | | | | | | |
| | **hisand** | **lowsand** | **lowzost** | | sums | percent |
| **hisand** | **57** | 0 | 2 | | 59 | 97 |
| **lowsand** | 0 | **39** | 2 | | 41 | 95 |
| **lowzost** | 5 | 2 | **34** | | 41 | 83 |
| | | | | **130** | | |
| sums | 62 | 41 | 38 | | **141** | |
| percent | 92 | 95 | 89 | | | **92.19858** |

**Figure 14** Confusion matrix after additional training of the class expert networks with synthetic data

| Trial | $F_N$ | $F_P$ | $R_{P/N}$ | Iterations | Improvement |
|---|---|---|---|---|---|
| **Initial** | | | | | |
| *hisand* | 1101 | 126 | 0.114 | 200 | ------ |
| *lowsand* | 1144 | 83 | 0.073 | 200 | ------ |
| lowzost | 1149 | 78 | 0.068 | 200 | ------ |
| ***Comission*** | | | | | |
| *lowzost* | 209 | 1 | 0.005 | +1 | 31 |
| ***Omission*** | | | | | |
| *hisand* | 10 | 24 | 2.4 | +16 | 1 |
| *lowsand* | 10 | 50 | 5 | +3 | 22 |
| **Expert** | | | | | |
| Initial *hisand* | 161 | 128 | 0.8 | 200 | 3 |
| Initial *lowsand* | 204 | | 0.407 | 200 | 28 |
| Initial *lowzost* | 209 | 78 | 0.373 | 300 | -1 |
| ***Comission*** | | | | | |
| *lowzost* | 209 | 1 | 0.005 | +1 | Initial 37<br>Expert 7 |
| ***Omission*** | | | | | |
| *hisand* | 10 | 50 | 5 | +1 | Initial 7<br>Expert 4 |
| *lowsand* | 10 | 50 | 5 | +3 | Initial 34<br>Expert 6 |
| *lowzost* | 10 | 50 | 5 | +1 | Initial -4<br>Expert -3 |

**Figure 15** The relationship among positive and negative examples used for learning, the iterations required, and the improvement on the system for each trial.

Note: $\mathbf{F_N}$ and $\mathbf{F_P}$ are the number of negative and positive examples respectively, $\mathbf{R_{P/N}}$ is the ratio between positive and negative examples, and **Improvement** is given as additional correctly classed examples; for omission it is based on the particular class and for comission it is based on the whole system.

# 6. Conclusion and Directions for Further Research

This paper offers two distinct strategies to improve classification performance and two different methods of learning that were applied to both strategies. The omission and comission error reduction techniques both performed adequately and the application of these techniques depend upon the user's precision requirements. The two strategies of operation, using the entire data stream, or new expert classes for error reduction, also depend upon the ambiguity of the data. If ambiguity exists between a small selected group of classes, the 'expert' strategy is feasible. The initial expert system accuracy over the ambiguous classes was 30%. After using either the omission or comission correction, the improvement was approximately 8% (Figure 15).

Further research has been planned in the following direction. Further development of the learning strategies in relation with rules extraction techniques; tuning the individual class-units to perform in an automated mode, thus achieving real adaptive image classification systems.


# 7. Acknowledgements

The initial funding for the ground reference survey and baseline environmental research was provided by the New Zealand Department of Conservation, Otago Conservancy and reported in [20]. The SPOT imagery was purchased through a grant from the New Zealand Lottery Board.


# References


[1] Congalton, R.G., Oderwald, R.G. and Mead, R.A., "Assessing Landsat Classification Accuracy Using Discrete Multivariate Analysis Statistical Techniques", Photogrammetric Engineering and Remote Sensing, 49(12), 1983: 1671-1678.

[2] Schowengerdt, R.A., "Techniques for Image Processing and Classification in Remote Sensing", Academic Press, Inc., Orlando, Florida, 1983.

[3] Lillesand, T.M. and Kiefer, R.W., "Remote Sensing and Image Interpretation, 3rd edition", John Wiley & Sons, New York, 1994.

[4] Gong, P. and Howarth, P.J., "An Assessment of Some Factors Influencing Multispectral Land-Cover Classification", Photogrammetric Engineering and Remote Sensing, 56(5), 1990: 597-603.

[5] Anand, R., Mehrotra, K., Mohan, C.K. and Ranka, S., "Efficient Classification for Multiclass Problems Using Modular Neural Networks", IEEE Transactions on Neural Networks, 6(1), 1995: 117-124.

[6] Bezdek, J.C. and Pal, N.R., "An Index of Topological Preservation for Feature Extraction", Pattern Recognition, 28(3), 1995: 381-391.

[7] Dreyer, P., "Classification of Land Cover Using Optimized Neural Nets on SPOT Data", Photogrammetric Engineering and Remote Sensing, 59(5), 1993: 617-621.

[8] Foody, G.M., McCulloch, M.B. and Yates, W., “Classification of Remotely Sensed Data by an Artificial Neural Network: Issues Related to Training Data Characteristics”, Photogrammetric Engineering and Remote Sensing, 61(4), 1995: 391-401.

[9] Hall, L.O., Bensaid, A.M., Clarke, L.P., Velthuizen, R.P., Silbiger, M.S. and Bezdek, J.C., “A Comparison of Neural Network and Fuzzy Clustering Techniques in Segmenting Magnetic Resonance Images of the Brain”, IEEE Transactions on Neural Networks, 3(5), 1992: 672-682.

[10] Horikawa, S.-I., Furuhashi, T. and Uchikawa, Y., “On Fuzzy Modelling Using Fuzzy Neural Networks with the Back-Propagation Algorithm”, IEEE Transactions on Neural Networks, 3(5), 1992: 801-806.

[11] Bezdek, J.C., Hall, L.O. and Clarke, L.P., “Review of MR Image Segmentation Techniques Using Pattern Recognition”, Medical Physics, 20(4), 1993: 1033-1048.

[12] Kasabov, N., “Learning, Generalisation, Adaptation and Forgetting in Fuzzy Neural Networks and Hybrid Systems”, Proceedings of the International Conference on Neural Information Processing (ICONIP'95- Beijing), Beijing, Publishing House of Electronics Industry, Beijing, 1995: 973-976.

[13] Kasabov, N.K., “Hybrid Connectionist Fuzzy Production System: Towards Building Comprehensive AI”, Intelligent Automation and Soft Computing, 1(4), 1995: 351-360.

[14] Bezdek, J.C., “A Review of Probabilistic, Fuzzy, and Neural Models for Pattern Recognition”, Journal of Intelligent and Fuzzy Systems, 1(1), 1993: 1-25.

[15] Paola, J.D. and Schowengerdt, R.A., “A Review and Analysis of Backpropagation Neural Networks for Classification of Remotely-Sensed Multi-spectral Imagery”, International Journal of Remote Sensing, 16(16), 1995: 3033-3058.

[16] Zhuang, X., Engel, B.A., Xiong, X. and Johannsen, C.J., “Analysis of Classification Results of Remotely Sensed Data and Evaluation of Classification Algorithms”, Photogrammetric Engineering and Remote Sensing, 61(4), 1995: 427-433.

[17] Kasabov, N.K., “Learning Fuzzy Rules and Approximate Reasoning in Fuzzy Neural Networks and Hybrid Systems”, Fuzzy Sets and Systems, Special Issue, 1996:1-19.

[18] Kasabov, N., “Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering”, MIT Press, Cambridge, Massachusetts, 1996.

[19] Kohonen, T., “The self-organising map”, Proceedings of the IEEE, 78:1464-1480

[20] Israel, S.A. and Fyfe, J.E., “Determining the Sensitivity of SPOT XS Imagery for Monitoring Intertidal and Sublittoral Vegetation of the Otago Harbour”, Department of Conservation, Otago Conservancy, Dunedin: Reference 9, 1995.

[21] Gupta, M. and Rao, D.H. On the principles of fuzzy neural networks, Fuzzy Sets and Systems, 61(1), 1994: 1-18.