

Evolving Neural Network Using Variable String Genetic Algorithms (VGA) for Color Infrared Aerial Image Classification

Xiaoyang Fu

Institute of Computer Science and
Technology, Jilin University

Zhuhai, China 519041

Tel: 86 756 7626277

Email: dvndavidfu@vip.163.com

P.E.R. Dale

Griffith School of Environment, Griffith
University

Tel: 61 7 3875 7136

Email: p.dale@Griffith.edu.au

Shuqing Zhang

Northeast Institute of Geography and
Agricultural Ecology, Chinese
Academy of

Sciences, Changchun, China 130012

Tel: 86 431 85542230

Email: shqzhang@263.net

ABSTRACT

In this paper, we investigate an evolving neural network classifier using variable string genetic algorithms (VGA) and its application to the land cover classification of Color Infrared Aerial (CIR) Image. With VGA, the classifier that we designed is able to evolve automatically the appropriate number of hidden nodes for modeling the neural network topology optimally and to find a near-optimal set of connection weights globally. The effectiveness of the VGA-BP classifier is demonstrated on CIR images. Compared with standard classifiers, such as Bayes maximum-likelihood classifier, VGA classifier and BP-MLP classifier, it has shown that the VGA-BP classifier can have better performance on highly resolution land cover classification.

Keywords

Variable String Genetic Algorithm, Neural Network, Pattern Classification

1. INTRODUCTION

Classification of pixels for partitioning different land cover regions is an important investigation in remote sensing imagery classification research. So far, besides conventional classifiers such as Bayes maximum-likelihood classifier (Tou and Gonzalez, 1974), several pattern recognition classifiers have been adopted, for example, the Fuzzy ARTMAP Classifier (Carpenter et al, 1997, 1999, Gopal, et al, 1999), Genetic Classifier (Bandyopadhyay and Pal et al, 1998, 2001) and Neural Network Classifier (Benediktsson et al, 1990). Within these methods, a neural network classifier with Evolutionary Algorithms (EA's) has been developed rapidly (X. Yao, 1999).

The most widely used neural network model is the multi layer

perceptron (MLP), in which the connection weight training can be implemented used by a back-propagation (BP) learning algorithm (Rumelhart et al, 1986). The BP learning approach is to start with an untrained network, presenting a training pattern to input layer, passing the signals through the network and determining the output in the output layer. Here these outputs are compared to the target values, any difference corresponds to an error. The error is minimized when the network outputs match the desired outputs. Thus the weights are adjusted to reduce this measure of error. The essential character of the BP algorithm is gradient descent. Because of the gradient descent algorithm is strictly dependent on the shape of the error surface, and the error surface may have some local minima, multimodal and/or non differentiable. This may result in falling into some local minimum and premature convergence may occur (Hertz et al., 1991).

On the other side, Genetic Algorithms (GAs) are randomized and optimized techniques guided by the principles of evolution and natural genetics. They are efficient, adaptive and robust in search process, producing near-optimal solutions and have a large amount of implicit parallelism, and can handle large, highly complex and multimode spaces (Goldberg, 1989). Of course, it is not surprised to evolve the neural network with GAs (Van Rooij et al, 1996, Yao, 1999) and to study the remote sensing land cover classification using GA evolving neural network (Liu et al, 2004). By searching a near-optimal set of initial connection weights of BP network with GA, Liu et al (2004) showed that the hybrid GA/BP approach was more efficient than either the GA or BP algorithm used alone.

In general, the neural network model for remote sensing imagery classification is a three-layer feed forward neural network. Its input nodes are equal to numbers of the multi-spectral bands (the dimensional numbers in feature space), and output nodes are equal to numbers of the image category, therefore its network architecture is dependent on numbers of the hidden nodes uniquely. the architecture design is crucial in the successful application of ANN's because the architecture has significant impact on a network's information processing capabilities.

Without optimizing the number of hidden nodes, the former work assumed that the architecture of an ANN is predefined and fixed during the evolution of connection weights (Liu et al, 2004),

Exploration on methods to evolve ANN architecture and connection weights simultaneously is therefore necessary.

In this paper, a hybrid variable string genetic algorithm (VGA) and BP (VGA-BP) algorithm is proposed for evolving the three-layer neural network architecture and connection weights.

In VGA Algorithm, an near-optimal neural network topology (the number of hidden nodes) and a set of initial connection weights can be obtained simultaneously by competing with each other potential individual solutions (which are chromosomes in term of VGA's) to a problem through selection, crossover, and mutation operations. And then using BP algorithm, the best connection weights at the local error surface is found.

The paper is organized as follows: Section 2 introduces a brief outline of three-layer neural network, VGA classifier, and a hybrid VGA algorithm based neural network classifier (VGA-BP). Section 3 introduces the study area and materials used by classifiers. Section 4 provides the results of the classification and comparison with CIR Image using our VGA-BP classifier and other classifiers. Finally, in Section 5, some conclusions and discussion are proposed.

2. METHODOLOGY

2.1 The Three-layer Neural Network

Assuming a three-layer neural network with m inputs (spectral bands), k outputs(categories), and l hidden nodes, see Fig1.

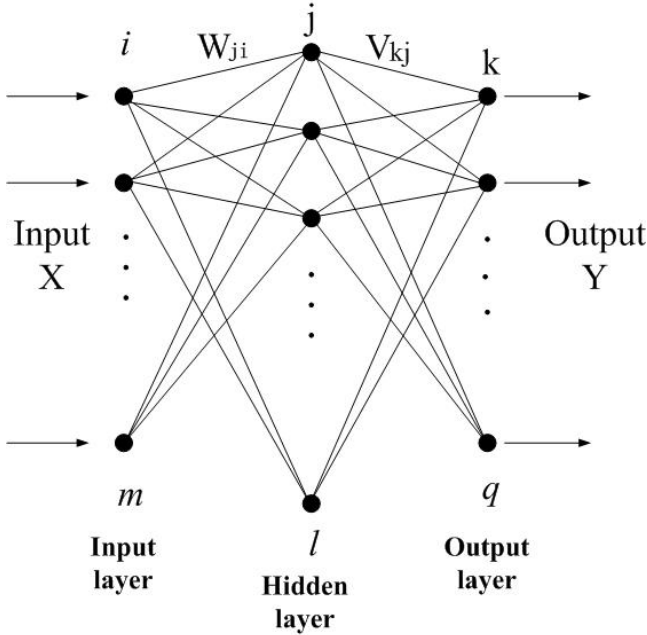


Fig. 1 Three layer neural network structure

Present a training pattern (x_i) to input nodes, pass the signals to hidden nodes through the connection weights w_{ji} , and produce a hidden output (y_j) through a threshold function (sigmoid), pass the signals to output nodes, and finally produce an output

(o_k) through a threshold function (sigmoid or other function).

We can formulate the relations of input and output as follows:

$$net_j = \sum_{i=1}^m w_{ji}x_i + w_{j0} \quad (1)$$

$$y_j = f(net_j) \quad (2)$$

$$net_k = \sum_{j=1}^l v_{kj}y_j + v_{k0} \quad (3)$$

$$o_k = f(net_k) \quad (4)$$

where w_{ji} is the connection weight between the j^{th} hidden node and the i^{th} input node, w_{j0} is its bias; v_{kj} is the connection weight between the j^{th} hidden node and the k^{th} output node, v_{k0} is its bias. $f(net)$ is a sigmoid activation function, it is defined as:

$$f(net) = \frac{1}{1 + e^{(-net)}} \quad (5)$$

where $net \in [-\infty, +\infty]$, It is better to adopt an anti-symmetric function as :

$$f(x) = a \cdot \tanh(bx) = a \cdot \left[\frac{1 - e^{-bx}}{1 + e^{-bx}} \right] \quad (6)$$

where $a=1.716$, $b=0.666$.

Suppose we have a set of training patterns $X = \{X_1, X_2, \dots, X_n\}$, where n is the number of training patterns, each training pattern X_i in set X is an m -dimensional feature vector. Let $T = \{T_1, T_2, \dots, T_n\}$ as set X 's corresponding output classes, $T_i = [t_1, t_2, \dots, t_q]$ is a q -dimensional class vector. If the target class for a specific pattern is k ($1 \leq k \leq q$), then we have $t_k = 1$, otherwise, $t_k = 0$.

Let's denote o_{ik} as the i^{th} actual neuron output for input training pattern x_i at the output node k while t_{ik} as its desired response. The mean square error function(MSE) for this neural network is defined as:

$$MSE(W) = \frac{1}{nq} \sum_{i=1}^n \sum_{k=1}^q (t_{ik} - o_{ik})^2 \quad (7)$$

where W represents all the weights in the network. Thus, this error is some scalar function of the weights and is minimized when the network outputs match the desired outputs.

2.2 VGA Classifier

2.2.1 Hyperplanes for Pattern Classification

Every pixel in a multi-spectral image with m -dimensional bands can be represented as a feature point in m -dimensional feature space. Thus we can formulate the pattern classification by putting a set of hyperplanes in the feature space appropriately such that the number of misclassified training points is minimized. From elementary geometry, the equation of a hyperplane in m -dimensional space ($X_1 - X_2 - \dots - X_m$) is given by

$$\beta_1 x_m + \beta_2 x_{m-1} + \dots + \beta_m x_1 = d \quad (8)$$

where $\beta_i = \cos \alpha_{m-i} \cdot \sin \alpha_{m-(i-1)} \dots \sin \alpha_{m-1}$. Here, α_j is the angle that the projection of the unit normal in the ($X_1 - X_2 - \dots - X_{j+1}$) space with the X_{j+1} axis. Since $\alpha_0 = 0$, so the $(\alpha_1, \alpha_2, \dots, \alpha_{m-1}, d)$ specifies a hyperplane uniquely in m -dimensional space, d is the perpendicular distance from the hyperplane to the origin.

2.2.2 Description of the VGA Algorithm

A. Hyperplane Encoding

In VGA algorithm, a set of hyperplanes can be encoded by a binary string (chromosome). The chromosome is represented by

$$Str = (\alpha_1^1, \alpha_2^1, \dots, \alpha_{m-1}^1, d_1, \alpha_1^2, \alpha_2^2, \dots, \alpha_{m-1}^2, d_2, \dots, \alpha_1^h, \alpha_2^h, \dots, \alpha_{m-1}^h, d_h)$$

where α_j^i is the α_j angle of the i^{th} hyperplane, d_i is its perpendicular distance. h is the number of hyperplanes. If each angle variable α is represented by b_1 number of bits and a perpendicular distance d is represented by b_2 number of bits, then the Str length is $[(m-1)b_1 + b_2] \cdot h$.

B. Misclassified Points

When partitioning a m -dimensional feature space with a set of hyperplanes, each hyperplane provides two half-spaces—a positive half-space and a negative half-space. For h hyperplanes, the maximum number of such regions is 2^h . Suppose there are some feature points in the i^{th} region, and the class corresponding to the most points in the i^{th} region is j ($1 \leq j \leq q$), then we consider that the i^{th} region belongs to class j , and all other points in the i^{th} region are misclassified points $miss_i$. If all misclassified points $miss_i$ are summed up for entire regions (2^h), then we obtain the total misclassified points $miss$.

$$miss = \sum_{i=1}^M miss_i \quad (9)$$

where $M = 2^h$.

C. Fitness Computation

The fitness function is defined as:

$$f_i = n - miss - \frac{h_i}{h_{\max}} \quad (10)$$

where n is the number of training patterns, h_{\max} is the maximum number of hyperplanes. h_i is the hyperplane number of the i^{th} string (chromosome). A string with zero hyperplane is defined to have zero fitness. Maximization of the fitness ensures the minimization of the number of misclassified points and the number of hyperplanes.

D. Genetic Operators

Selection: the roulette wheel selection procedure has been adopted to implement a proportional selection strategy (Holland, 1975, Goldberg, 1989)

Crossover: Two string, i and j , having lengths l_i and l_j respectively, are selected from the mating pool. Let $l_i \leq l_j$, then string i is padded with # so as to make the two lengths equal. Single point crossover is performed over these two strings with probability μ_c . The following two cases may now arise: (1) all the hyperplanes in the offspring are complete (all the bits corresponding to it are either defined (i.e., 0_s and 1_s) or #_s); (2) some hyperplanes are incomplete.

In the second case, the incomplete hyperplanes must be modified as follows: Let u is the number of defined bits (either 0 or 1) and t is the total number of bits per hyperplane. Then, for each incomplete hyperplane, all the #_s are set to defined bits (either 0 or 1 randomly) with probability u/t . Otherwise, all the defined bits are set to # with a probability $(1-u/t)$. Do modification mentioned above repeatedly until all incomplete hyperplanes become complete.

Mutation: In order to introduce greater flexibility in the method, the mutation operator is defined in such a way that it can both increase and decrease length. For this, the strings are padded with # such that the resultant length equal to l_{\max} which is max length of chromosome. Now for each defined bit position, its reverse bit is set with probability μ_m . Otherwise, the position is set to # with probability μ_{m1} . Each undefined position is set to defined bit (0 or 1 randomly) with probability μ_{m2} . Note that mutation may also result in some incomplete hyperplanes, so these are handled as the same way as crossover operation.

Details about VGA classifier are described in reference [16].

2.3 Evolving Neural Network Using VGA

The training of evolving neural network using VGA algorithm is composed of three major phases. (1) Connection weight encoding; (2) compute fitness through the feed forward neural network for a set of training patterns; (3) select parents from the current generation according to their fitness, and apply crossover operator and mutation operator to parents to generate offspring which form the new generation. This process is iterated until some stopping criterion is met.

2.3.1 Connection Weight Encoding

Assuming the connection weight chromosomes represented by two strings which have hidden nodes l .

$$StrV = (v_{10}, v_{20}, \dots, v_{q0}, v_{11}, v_{21}, \dots, v_{q1}, \dots, v_{1l}, v_{2l}, \dots, v_{ql})$$

$$StrW = (w_{10}, w_{20}, \dots, w_{m0}, w_{11}, w_{21}, \dots, w_{m1}, \dots, w_{1l}, w_{2l}, \dots, w_{ml})$$

The length of StrV and StrW is $(l+1)q$ and $(m+1)l$, respectively. We use a real number form to represent the connection weights. Because of the l is variable, so the length of StrV and StrW is also variable in evolving procedure. By convenient rule of thumbs (Duda et al., 2001), the number of hidden nodes is chosen such that the total number of weights in the network is roughly $n/10$, or more than this number, but it should not be more than the total number of training points, n . For the three-layer neural network, the total number of weights (excepting bias) is $l(m+q)$, so the maximum number of hidden nodes $l_{\max} \leq n / (m + q)$, the hidden node l can be chosen number from range $[l_{\max} / 10, l_{\max}]$.

In initialization, the connection weight values (including biases) are randomly chosen a real number from range $[-1, +1]$.

2.3.2 Fitness Computation

For a pair of strings (StrV_{*i*} and StrW_{*i*}) with the same length l_i , we can calculate the Error Mean Square (EMS) for a set of training patterns through formulation (1)–(5). The fitness function is defined as:

$$f_i(net) = e^{-\left(\alpha \cdot EMS - \frac{l_i}{l_{\max}}\right)} \quad (11)$$

where f_i is the fitness of the i^{th} string, l_i is its number of hidden nodes, α is a positive constant. Therefore, maximization of the fitness ensures the minimization of the EMS, and the term l_i / l_{\max} will force the minimization of the number of hidden nodes. If delete the term, then the neural network structure is evolved according to maximization of the fitness, not preferring the minimization of the number of hidden nodes.

2.3.3 Genetic Operators

Selection, crossover and mutation are adopted as the same way as above section 2.2.2. Note that the connection weight encoding is a real number from rang $[-1, +1]$.

In summary, the typical evolution flowchart of VGA-BP can be described as Fig.2

1. Construct a set of neural networks with randomly generated hidden structure and initial connection weights, and train them using a number of training patterns.
2. Calculate the fitness of each individual (chromosome) according to the average training result calculated from formulation (1)–(5).
3. Select parents from the current generation according to their fitness.

4. Apply crossover and mutation operators to parents to generate offspring that form the new generation.
5. Do the step 2 to 4 repeatedly until some stopping criterion is met. Now we get a near-optimal connection weights and the network topology.
6. Go on to train with BP algorithm based on both hidden nodes and connection weights trained by VGA and to find the best connection weights.

Fig. 2 A typical flowchart of the evolution of VGA-BP classifier

3. Study area and Materials

The study area is approximately 8 ha of a salt-marsh on Coomera Island ($S 27^{\circ} 51', E 153^{\circ} 33'$) in south-east Queensland, Australia. It is generally low-lying and hummocky with a relative relief of $<1m$. The island is approximately 10 km north of one of major tourist resorts in Australia, the Gold Coast, and 80 km south of Brisbane, the State capital. The main tidal flooding source is a shallow inlet to the north (with another, indirect, source to the south). The inlet is vegetated with the Grey Mangrove (*Avicennia marina* (Forsk.)). The salt marsh vegetation is mainly Marine Couch (*Sporobolus virginicus* (L. Kunth) and *Sarcocornia quinqueflora* (Bunge ex Ung.-Stern) in a fine mosaic. These will subsequently be referred to as *Sporobolus* and *Sarcocornia*. *Sporobolus* grows taller and more densely in less saline areas which are locally elevated, whereas *Sarcocornia* is mainly found in areas of higher salinity which are prone to waterlogging such as in local depressions and adjacent to drainage lines (P.E.R. Dale et al., 1986).

The CIR photography was provided by Dale et al(2006). The original photographs had been enlarged to a scale of 1:1100 as positive paper prints that could later be used in the field. The samples for classification were obtained through a neutral filter, and primary color filters of blue, green and red since these separate respectively the green, red and infrared spectral bands on the CIR image. For each sample, site values for image reflectance through each filter were treated as the attributes.

4. RESULTS

In the section, we will discuss some experimental results with VGA-BP classifier and compared with other classifiers. The effectiveness of algorithm has been demonstrated on CIR image.

4.1 Data Set

The dataset, which is used for training and testing patterns is randomly extracted from two part data: the site spectral values which were classified using a divisive classification procedure and data from the field sampling (reported in P.E.R. Dale et al., 1986). A total of 200 samples belonging to seven land cover classes are used for training and testing. The seven classes are Tall dense *Sporobolus* (TDSp.), Mixed *Sporobolus* and *Sarcocornia* relatively tall and dense (MTDSp.Sa.), Mixed plant species of medium density (MMDSp.Sa.), *Sarcocornia* (Sa), Mangrove (Man), Water Body (WB) and Bare Ground (BG).

4.2 Implementation Parameters of Classifiers

The control parameters for Bayes maximum-likelihood classifier, VGA classifier, BP neural network classifier and VGA-BP classifier are defined as follows,

(1) Bayes maximum-likelihood classifier

Assuming the conditional densities are the normal density, and the prior probabilities $P(w_i)$ can get from training patterns.

(2) VGA classifier

Population size, h_{\max} , μ_c , μ_{m1} and μ_{m2} is 20, 15, 0.8, 0.1, 0.1, respectively. μ_m is variable within the range[0.015,0.333]. The algorithm is terminated if the population contains at least one string with no misclassified points. Otherwise, the algorithm is executed for 6000 generations.

(3) BP-MLP classifier

A three-layer MLP with one hidden layer consisting of 8 hidden nodes is used. For the back propagation training algorithm, the learning rate is 0.15, the learning rate increment is set to 1.004, the momentum rate (Duda et al., 2001), which allows the MLP network learn more quickly when plateaus in the error surface exist, is 0.9, and the target training performance is 0.0185. The algorithm is end if performance is less than 0.0185. Otherwise, the algorithm is executed for 3000 epochs.

(4) VGA-BP classifier

Using VGA algorithm, we evolve the BP-MLP network structure dynamically to get a near-optimal architecture. Simultaneously, we get initial connection weights for BP-MLP network such that BP algorithm can find the globally optimal weights easily.

In VGA algorithm, there are two population groups, and each population size is 20, the maximum number of hidden nodes is $n/(m+q)$, crossover probability is 0.75, and mutation probabilities are the same as the VGA classifier.

In BP algorithm, all parameters are set to the same as BP-MLP for comparison fairly.

The VGA algorithm is executed for 500 generations, and BP algorithm is terminated if the target training performance is met. Otherwise it is executed for 3000 epochs.

4.3 Accuracy Assessment of classifiers

User accuracy and Kappa coefficient are used to quantitative assess the new classifier's capacity of classification (Congalton et al, 1983, Rosenfield and Fitzpatrick-Lins, 1986). If n_i ' points (of all the n points) are found to be classified into class i , then the user's accuracy (U) is defined as

$$U = n_{i_c} / n_i \quad (12)$$

where n_{i_c} points have been correctly classified. User accuracy denotes the level of purity associated with a classified region.

The coefficient of agreement called "kappa" measures the

relationship of beyond chance agreement to expected disagreement. The estimate of kappa(K) is the proportion of agreement after chance agreement is removed from consideration.

The estimate of kappa for class i (K_i) is define as

$$K_i = \frac{n \times n_{i_c} - n_i \times n_i}{n \times n_i - n_i \times n_i} \quad (13)$$

The numerator and denominator of the overall kappa are obtained by summing the respective numerators and denominators of K_i separately over all classes i .

For comparison, Bayes maximum-likelihood classifier, VGA classifier, BP neural network classifier and VGA-BP classifier are carried out on both the same training points (70% points of Data Set) and the testing points (30% points of Data Set). Bayes maximum-likelihood classifier is taken as a basis for its high performance in remote sensing imagery classification, as it is most widely and generally applied (Enrica Belluco et al., 2006). VGA classifier and BP neural network classifier are also chosen as we hope to know if VGA-BP classifier performs better.

Comparative classification results of VGA-BP classifier and other classifiers are shown in Table 1 and Table 2.

Table 1. Results of Classification Accuracy for Training (70% of data set)

Class	User's Accuracy (U%)				Kappa coefficient			
	Bayes	VGA	BP-MLP	VGA-BP	Bayes	VGA	BP-MLP	VGA-BP
TDSp.	96.67	86.67	90.00	93.33	0.959	0.840	0.880	0.919
MTD Sp.Sa.	80.00	90.00	90.00	90.00	0.774	0.883	0.886	0.886
MMD Sp.Sa.	55.56	55.56	55.56	66.67	0.512	0.518	0.509	0.627
Sa.	60.00	68.00	64.00	68.00	0.527	0.629	0.571	0.627
Man.	94.12	88.24	94.12	100.00	0.935	0.870	0.935	1.000
WB.	78.79	93.94	81.82	84.85	0.738	0.922	0.774	0.811
BG.	92.00	92.00	96.00	96.00	0.905	0.905	0.953	0.953

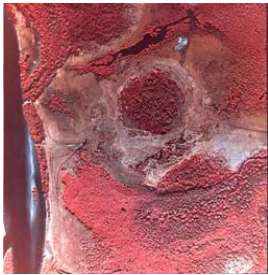
Table 2. Results of Classification Accuracy for Testing (70% of data set for training, 30% for testing)

Class	User's Accuracy (U%)				Kappa coefficient			
	Bayes	VGA	BP-MLP	VGA-BP	Bayes	VGA	BP-MLP	VGA-BP
TDSp.	100.00	85.71	100.00	100.00	1.000	0.831	1.000	1.000
MTD Sp.Sa.	60.00	80.00	60.00	80.00	0.570	0.760	0.564	0.771
MMD Sp.Sa.	28.57	28.57	28.57	42.86	0.232	0.265	0.255	0.395
Sa.	46.15	61.54	53.85	46.15	0.375	0.553	0.455	0.385
Man.	87.50	68.75	81.25	81.25	0.845	0.631	0.771	0.771
WB.	75.00	87.50	87.50	87.50	0.723	0.857	0.852	0.850
BG.	87.50	81.25	75.00	75.00	0.839	0.755	0.684	0.684

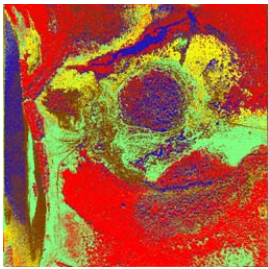
As seen from Table 1 and 2, the performance of the VGA-BP classifier is better than that of the BP-MLP classifier, either training or testing. VGA-BP classifier recognizes the different classes consistently with a high degree of accuracy. On the contrary, the other classifiers can recognize some classes very well, however, much they are poorer for other classes. For example, the Bayes classifier provides User's accuracy of 100.00% and 87.50% for TDSp. and Man., respectively, but its User's accuracy for MMDSp.Sa. is only 28.57%. Here we use Cross-Validation to realize a low generalization error.

4.4 Pixel Classification of CIR Image

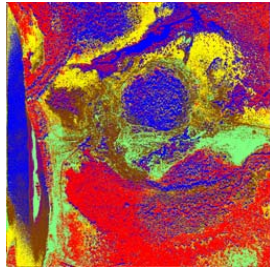
A 6609×6501 pixels CIR highly resolution image of Coomera Island ($S 27^{\circ} 51'$, $E 153^{\circ} 33'$) in southern Moreton Bay, Queensland, Australia is used for classification (Fig. 3(a)). Different land cover classification results are shown in Fig. 3(b)-Fig. 3(e).



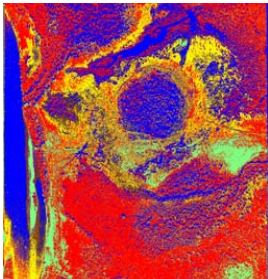
(a)Original CIR image



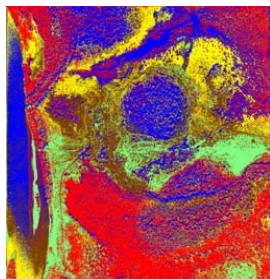
(b) Bayes classification



(c) VGA classification



(d) BP-MLP classification



(e) VGA-BP classificatoin



Fig. 3 Classification Results of CIR image of Coomera Island

As seen from the figure, although all the classifiers are able to identify Mangroves, Water Body and Bare Ground, only VGA-BP classifier and BP-classifier are able to identify different plant species more clearly. As well the spatial pattern in Fig. 3(e) most closely relates to the actual distributions on the ground. Note that the circular area marked in blue is outside the salt marsh. It is upland and so was not part of the training pixels. Having relatively low reflectance in the near infra red it has been classified as water, a not unreasonable conclusion.

The VGA-BP training performance is shown in Fig.4, Fig.5 and Fig. 6.

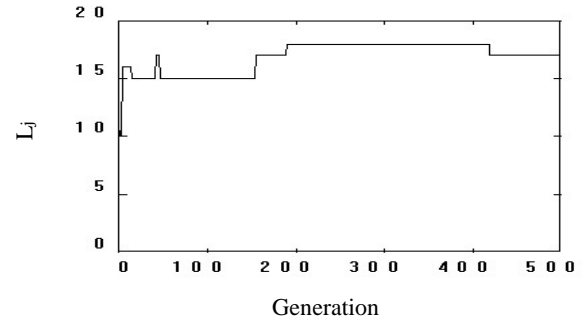


Fig.4 Best Neural Network Hidden Nodes by VGA Evolutionary Training

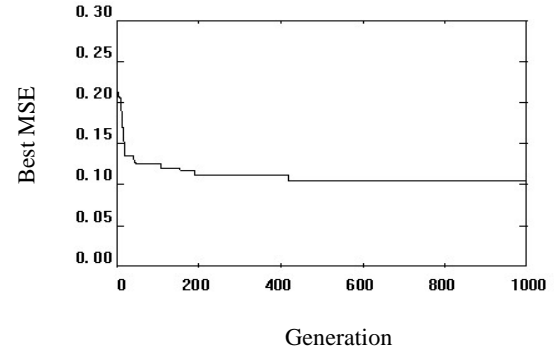


Fig. 5 Best MSE by VGA Evolutionary Training

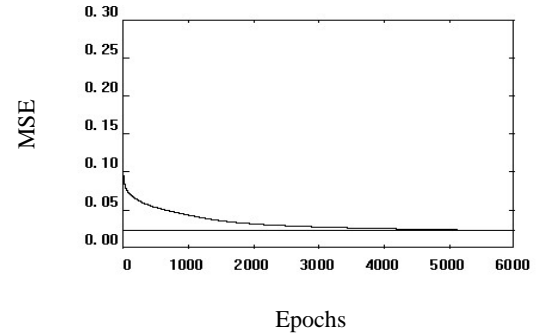


Fig. 6 MSE by BP Evolutionary Training

Fig. 4 is number of hidden nodes (L_j) corresponding to each generation during the evolutionary training. From this Figure, we can see the variation of best topology represented by the best

chromosome with the number of generation of the VGA algorithm. The best number of hidden nodes (17) is obtained just after the 430 generations. Fig.5 shows the best MSE corresponding to each generation during the evolutionary training of MLP architecture and initial connection weights. Fig.6 is the MSE after VGA algorithm corresponding to each epoch during BP training. It is found that although VGA algorithm has greatly reduced the total MSE of the neural network, a more improvement of training performance is achieved by applying a back propagation weight adjustment procedure. The time took during training and partitioning for CIR image is 24.98 and 9.88 seconds, respectively, which are calculated on a PC (Intel CPU 2.80GHz, 1 G RAM).

According to the results of the Classification accuracy of classifiers, we could conclude that water body and mangrove are easily classified while Sp. and Sa. are a little more difficult. However the hardest classes to be identified are the mixed two types. This may be because the differences between the mixed classes are not so obvious in the digital data. On the ground the *Sarcocornia* tends to have a sprawling life form and so the density estimates may be affected by this. Density was measured in terms of number of succulent stems per quadrat, yet if those are near horizontal the apparent density may be relatively large. So these two types suffer from poor classification accuracy from classified plot. Although the classification method is still having some works to improve the classified accuracy, it also provides us an alternative to extend our knowledge of nature and extent of the changes to a wider area than field experiment unobtrusively.

5. DISCUSSION AND CONCLUSIONS

In this paper, we have described an evolving neural network classifier using variable string genetic algorithm (VGA) in detail. We could find some important features of different classifiers we studied for image classification. The Bayes classifier is classic classifier, whose structure is determined by the conditional densities as well as by the prior probabilities. Although Gaussian density is popular of the various density functions, we have to get the prior-knowledge of density function for various complex classification problems. The VGA classifier attempts to place H hyperplanes in the feature space appropriately such that number of misclassified training points is minimized. However, although the classifier could handle with hyperplanes (H) in n dimensional space, how to choose the maximum value of H is still a practical problem (because the search space regions are equal to 2^H). The ANN could deal with complex nonlinear real-world problems so that the parameters governing the nonlinear mapping are learned at the same time as those governing the linear discriminant. However, BP algorithm often gets trapped in a local minimum of the error function and is incapable of finding a global minimum if the error function is multimodal. One way to overcome gradient-descent-based training algorithms' shortcomings is to adopt evolutionary ANN.

With VGA, the classifier that we designed is able to evolve automatically the appropriate number of hidden nodes for modeling the neural network topology optimally and to find a near-optimal set of initial connection weights globally. Then, using BP to perform local search from the initial connection weights and architecture, finally get the best connection weights for ANN. The effectiveness of the algorithm is demonstrated on

CIR images. Compared with standard classifiers, such as Bayes ML classifier, VGA classifier and BP-MLP classifier, it has shown that the hybrid VGA algorithm based neural network classifier can have better performance on highly resolution land cover classification. The VGA-BP classifier is not only used for CIR image but also for multi-spectral image classification.

The observed feature of the methodology is that the neural network structure is evolved automatically while connection weights are being evolved. Here some control parameters used in VGA algorithm and BP, such as crossover operator, mutation operator, learning rate, momentum rate etc, are experiential values that given by experts. However proper selection of control parameters for VGA algorithm and BP algorithm to different classification problems is still an open issue, which is a part of our further work.

6. ACKNOWLEDGMENTS

We would like to thank Northeast institute of Geography and Agricultural Ecology for its supports and Faculty of Environment Science, Griffith University for CIR image and field samples.

7. REFERENCES

- [1] Benediktsson, J.A., Swain, P.H. and Ersoy, O.K., 1990. Neural Network approaches versus statistical methods in classification of multi-spectral remote sensing data. *IEEE Transaction on Geoscience and Remote Sensing*, 28(4):540-552.
- [2] Carpenter, G.A., Gijaja M.N., Gopal S. and Woodcock C.E., 1997. ART neural networks for remote sensing vegetation classification from Landsat TM and terrain data. *IEEE Transaction on Geoscience and Remote Sensing*, 35(2):308-325.
- [3] Carpenter G.A., Gopal S., Macomber S., Martens Siegfriens and Woodcock C.E., 1999. A neural network method for mixture estimation for vegetation mapping. *Remote Sensing of Environment*, 70:138-152.
- [4] Congalton, R.G., Oderwald, R.G., and Mead, R.A., 1983, Assessing Landsat Classification Accuracy Using Discrete Multivariate Analysis Statistical Techniques. *Photogrammetric Engineering and Remote Sensing*, 49, 1671-1678
- [5] Dale, P.E.R., Hulsman, K. and Chandica, A.L. 1986. Seasonal Consistency of salt-Marsh Vegetation classes Classified from Large-Scale Color Infrared Aerial Photographs. *Photogrammetric Engineering and Remote Sensing Vol. 52, No.2 February 1986*, pp.243-250.
- [6] Dale, P.E.R., Hulsman, K. and Chandica, A.L. 1986. Classification of reflectance on colour infrared aerial photographs and sub-tropical salt-marsh vegetation types. *International J. Remote Sensing* 7:1783-1788
- [7] Dale, P.E.R., Chandica, A.L. and Evans, M. 1996. Using substraction and classification to evaluate change in sub-tropical intertidal wetlands. *International J. Remote Sensing* 1996, Vol. 17, No. 4, 703-719.

- [8] Dale P.E.R. and Dale, M.B. 2002. Optimal classification to describe environmental change : Picture from the exposition. *Community Ecology* 3(1):19-29
- [9] Dale, P.E.R. ,2001. Wetlands of conservation significance:mosquito borne disease and its control. *Arbovirus Research in Australia* 8:102-108
- [10] Enrica Belluco , Monica Camuffo, Sergio Ferrari, Lorenza Modenese, Sonia Silvestri, Alessandro Marani, Marco Marani, Mapping salt-marsh vegetation by multispectral and hyperspectral remote sensing. *Remote sensing of environment*,2006
- [11] Goldberg,D.E.,1998. Genetic algorithms in Search, Optimization and Machine Learning. *Addison-Wesley, New York*.
- [12] Richard O. Duda, Peter E.Hart and David G.Stork, Pattern Classification, Second Edition, *John Wiley & Sons, Inc.*
- [13] Harp,S.A.,Samad,T.,1992. Genetic synthesis of nwural network architecture. In:L.David(Ed),*Handbook of Genetic Algorithms*. New York,pp.202-221.
- [14] Hertz J., Krogh A., Palmer R., 1991. A Introduction to the Theory of Neural Computation. *Addison-Wesley. Readings, CA*
- [15] Holland,J.H.,1975. Adaptation in Natural and Artifical Systems. *The University of Michigan Press*.
- [16] Maniezzo V.,1994. Genetic evolution of the topology and weight distribution of neural networks. *IEEE Trans on Neural Networks* 5,39-53.
- [17] Richard O. Duda, Peter E. Hart, and David G. Stork: Pattern Classification, Second Edition(ISBN:0-471-05669-3). *John Wiley & Sons*.
- [18] Rosenfield, G.H. and Fitzpatrick-Lins, K.,1986,Coefficient of agreement as a measure of thematic classification accuracy. *Photogrammetric Engineering and Remote Sensing*, 52, 223-227
- [19] Rumelhart, D.E. , Hinton,G.E. and Williams,R.J., 1986. Learning representations by back-propagating errors. *Nature*,323:533-536
- [20] S. Bandyopadhyay, C.A. Murthy, Sankar K. Pal, 1998. Pattern classification using genetic algorithms: Determination of H. *Pattern Recognition Letters* 19(1998)1171-1181.
- [21] S. Bandyopadhyay and Sankar K.Pal,2001. Pixel Classification Using Variable String Genetic Algorithms with Chromosome Differentiation. *IEEE Transaction on Geoscience and Remote Sensing*, Vol. 39,No.2.
- [22] Sankar K. Pal, S.Bandyopadhyay, C.A. Murthy, 2001.Genetic Classifiers for Remotely Sensed Images: Comparison with Statndard Methods. *International Journal of Remote Sensing*, Vol.22, No.13,pp. 2545-2569
- [23] Shuqing Zhang, Junyan Zhang, Fang Li, Roger Cropp,2006. Vector analysis theory on landscape pattern (VATLP). *Ecological Modelling* Vol.193,page 492-502
- [24] Tou,T.T. and Gonzalez, R.C.,1974. Pattern Recognition Principles. *Addison-Wesley, New York*
- [25] Van Rooij A.J.F., et al. Neural Network Training Using Genetic Algorithm. *World Scientific*,1996,3:45-54
- [26] Yao X., 1999. Evolving artificial neural networks. *Proceeding of the IEEE*, 87(9):1423-1447
- [27] Zhengjun, Liu, Changyao,Wang, Aixia, Liu, and Zheng, Niu.,2004. Evolving Neural network Using Real Coded Genetic Algorithm(GA) for Multi-spectral Image Classification. *Future Generation Computer Systems* 2004,20(7):1119-1129,30