Salzburg Interest Group on Integrated Systems

siGis-002-1996 May 24, 1996

Evolution of Low Complexity Artificial Neural Networks for Land Cover Classification from Remote Sensing Data

20th Workshop of the Austrian Pattern Recognition Group (ÖAGM)
May 9–10, 1996, Schloß Seggau, Leibnitz, Austria

Roland Schwaiger
rschwaig@cosy.sbg.ac.at

Helmut A. Mayer
helmut@cosy.sbg.ac.at

Reinhold Huber
reini@cosy.sbg.ac.at

Department of Computer Science
University of Salzburg

Correspondence to:

Universität Salzburg
Institut für Computerwissenschaften und Systemanalyse
Jakob–Haringer–Straße 2
A–5020 Salzburg
Austria

Evolution of Low Complexity Artificial Neural Networks for Land Cover Classification from Remote Sensing Data

Roland Schwaiger
rschwaig@cosy.sbg.ac.at

Helmut A. Mayer
helmut@cosy.sbg.ac.at

Reinhold Huber
reini@cosy.sbg.ac.at

Department of Computer Science
University of Salzburg

Abstract **- Artificial Neural Networks (ANN) have gained increasing popularity as an alternative to statistical methods for classification of Remote Sensing Data. Their superiority to some of the classical statistical methods has been shown in the literature. Therefore, ANNs are commonly used for segmentation and classification purposes. We address the problem of generating an appropriate low complexity network topology, the right number of training epochs and preprocessing the training data set for multi-layer feed-forward ANNs. A method based on Genetic Algorithms (GA) for the automatic generation of problem–adapted topologies is employed with the parallel netGEN system which has been designed by the authors. A land cover classification problem using multi–spectral Landsat Thematic Mapper (TM) data is presented so as to demonstrate the capabilities of netGEN.**

1 Introduction

Artificial Neural Networks (ANN) have gained increasing popularity as an alternative to statistical methods for classification of Remote Sensing Data. Results on land cover classification from various imagery have been reported, e.g. Bischof *et al.* for Landsat Thematic Mapper (TM) data [BSP92]. Their superiority to some of the classical statistical methods has been shown in the literature [Lip93]. Therefore, ANNs are commonly used for segmentation and classification purposes.

ANN design is a key issue in ANN research. The architecture of the ANN has direct impact on training time, convergence and generalization ability. ANN architecture or complexity is determined by the number of neurons, number of connections, and their connectivity. While networks tend to lose generalization ability with increasing complexity, they are less able to *memorize* input patterns with decreasing complexity. A way to avoid a too complex ANN is to limit the growth of the weights through some kind of weight decay [KH92], which is a special kind of regularization. Various methods have been suggested for the automatic construction

of ANN topologies. Among these are *Network Pruning* (OBS [HSW93], OBD [CDS90], etc.), *Network Growing* (Cascade Correlation [FL90], etc.), and *Evolutionary Design* of networks.

Adopting the evolutionary design approach by means of *Genetic Algorithms* (GA) the topology search is governed by a fitness function. With our direct encoding scheme implemented in the netGEN system which is an extension to the representation suggested by *Miller & Todd* [MTH89] *Feed–Forward* ANNs are generated and trained in parallel by *Error–Back–Propagation*. The fitness is evaluated as the ANN performance on the validation set. netGEN genotypes comprise *Neuron Markers* [HMS95] which indicate the absence or presence of a maximum preset number of hidden neurons, hereby, increasing the variability of the neuron number and the space of ANN topologies. In order to combat *overfitting* we also encoded the number of training epochs.

We present the ANN topologies found by evolutionary search for a land cover classification problem on Landsat TM data taken from the Austrian National Park "Hohe Tauern". Specifically, we consider the influence of an additional *regularization term* used with the GA fitness function.

2 Evolution of Artificial Neural Networks

2.1 Genotype Representation of an ANN

Out of numerous existing genotype representations of an ANN [BH95], we adopted a *direct encoding* method introduced by *Miller & Todd* [MTH89]. Their original encoding - in the following called *Miller–Matrix* (MM) - uses an adjacency matrix to represent the topology of an ANN. Each entry $w_{ij} \in \{0(\text{no connection}), L(\text{learnable connection})\}$ in the matrix at position (i, j) indicates the absence or presence of a directed connection between unit j and unit i. Further, they added a column for the biases of the units.

In our *Modified–Miller–Matrix* MMM encoding scheme [HMS95] we dropped the bias column and added *Neuron Markers* in the main diagonal (i, i) of the MMM, indicating the absence or presence of a particular neuron i. As a consequence, the only topology parameters which have to be set are the number of input and output neurons (depending on the problem) and the maximum number of neurons.

In order to combat overfitting [RF95], we also encoded the number of training epochs with the further benefit of another ANN parameter being set automatically.

In addition to our modifications of the MM encoding scheme, the GA does not operate on the matrix representation, but on the concatenation of the rows of the non–zero part of the MMM's

lower triangle matrix representing the possible connections. All MMM entries clamped to zero need not be processed by the GA, hence, they are not encoded in the genotype.

The general form of the genotype looks like:

$$e_1 e_2 \ldots e_{n_1} m_1 m_2 \ldots m_{n_2} x_1 x_2 \ldots x_{n_3},$$

where

1. $e_i \in \{0,1\}, i = 1, \ldots, n_1$ is the binary representation of the number of epochs,
2. $m_i \in \{0,1\}, i = 1, \ldots, n_2$ are the neuron markers, where $m_i = 0$ indicates the absence and $m_i = 1$ indicates the presence of a neuron and
3. $x_i \in \{0,1\}, i = 1, \ldots, n_3$ indicate the absence/presence of a connection. The x_i represent possible connections (non–zero entries in the MMM). Reflexive connections, connections within input and output layer, and connections from unit i to unit j with $i < j$, i. e. constant zero entries in the MMM, are not encoded.

This enables the use of the standard genetic operator *2–Point–Crossover* [Gol89] which has a more global effect on the bit string representation (and therefore, the ANN topology) than the MM crossover operator exchanging rows or columns of two individuals. The mutation operator used in netGEN is the standard bit–flip with mutation probability p_m.

For the GA selection phase we chose *Binary Tournament Selection*, for it showed mostly better results than *Proportionate Selection Methods*. First analytical steps, possibly confirming this observation, have been reported recently [BT95].

2.2 Fitness Functions

Besides a "good" encoding of the problem, the choice of an appropriate fitness function is essential to the GA and its capability of finding (sub)optimal solutions. As for the land cover classification problem, we experimented with various fitness functions using penalties so as to assess the distance of a misclassification to a correct classification. An ANN whose observed winner neuron in the output layer is close to the wanted winner with one particular test pattern receives a small penalty which increases with distance [RPLH89]. However, at the time we arrived at the best solutions with a simple fitness function without penalties using the classification error $\mathcal{E}_s$ of an individual ANN on the validation set and is given by $\mathcal{E}_s = \frac{e_v}{n_v}$, with e_v being the number of misclassified patterns and n_v the total number of patterns in the validation set.

Table 1 shows the GA and ANN parameters used in all the experiments in this paper.

GA Parameters		ANN Parameters	
Population Size	50	Network Topology	Feed–Forward
Generations	50	Activation Function (all neurons)	Sigmoid
Crossover Probability p_c	0.6	Output Function (all neurons)	Identity
Mutation Probability p_m	0.005	Training	Back–Propagation
Crossover	2–Point	Learning Parameter η	0.2
Selection Method	Binary Tournament	Momentum Term	No
		Number of Epochs	Evolutionary
		Training Set Size	197
		Validation Set Size	1000

Table 1: Parameter settings for the GA and the ANN used throughout this study.

2.3 Regularization Terms

It has been shown that free parameters of the network such as weights (number, magnitude), biases (magnitude), units (number) have a direct influence on the generalization ability of neural nets [KH92]. To include these parameters in the network error function we have to add another term to the classification error term $\mathcal{E}_s$: i.e.

$$\mathcal{E} = \alpha_s \mathcal{E}_s + \alpha_c \mathcal{E}_c,$$

where $\mathcal{E}_c$ is called a complexity or regularization term [Hay94] . The constants α_s and α_c emphasize both error measures, accordingly.

The first regularization term examined was proposed by Hinton [Hin86], where C_{total} refers to all the weights in the network. The second term was suggested by Rumelhart [HP89][1]:

$$\text{Hinton:} \qquad \mathcal{E}_c = \sum_{i \in C_{total}} w_i^2, \qquad \text{Rumelhart:} \qquad \mathcal{E}_c = \sum_{i \in C_{total}} \frac{w_i^2}{1 + w_i^2}.$$

Harp & Samad [HS91] introduced a general fitness function for ANNs

$$\mathcal{F} = \sum_{k=1}^{m} \alpha_k \Phi_k \left(\mathcal{E}_k \right),$$

where m is the number of freely adjustable parameters, α_k are the weights of the various error measures of the different parameters denoted by $\mathcal{E}_k$, and the Φ_k transform the error measures. For the problem of finding an appropriate network topology for classification problems we established a fitness function which allows to minimize learning error and irregularity. Hence, we take $\mathcal{E}_c$ and $\mathcal{E}_s$ and scale each of them such that they fall in the interval $[0, 1]$. Thus the generic fitness function looks like,

$$\mathcal{F} = \alpha_1 (1 - \mathcal{E}_s) + \alpha_2 \frac{1}{1 + \mathcal{E}_c},$$

[1] Weigend et al. [WRH91] modified the regularization term by dividing the weights by a preset constant w_0.

where $\alpha_1 + \alpha_2 = 1$. In the following we analyze the impact of different regularization parameters and various values of α_i on the topology and the classification accuracy of the ANN.

Figure 1 shows an evolutionary generated ANN with seven input and 6 output neurons, with α_1 and α_2 set to 0.5 and Rumelhart term used for regularization.

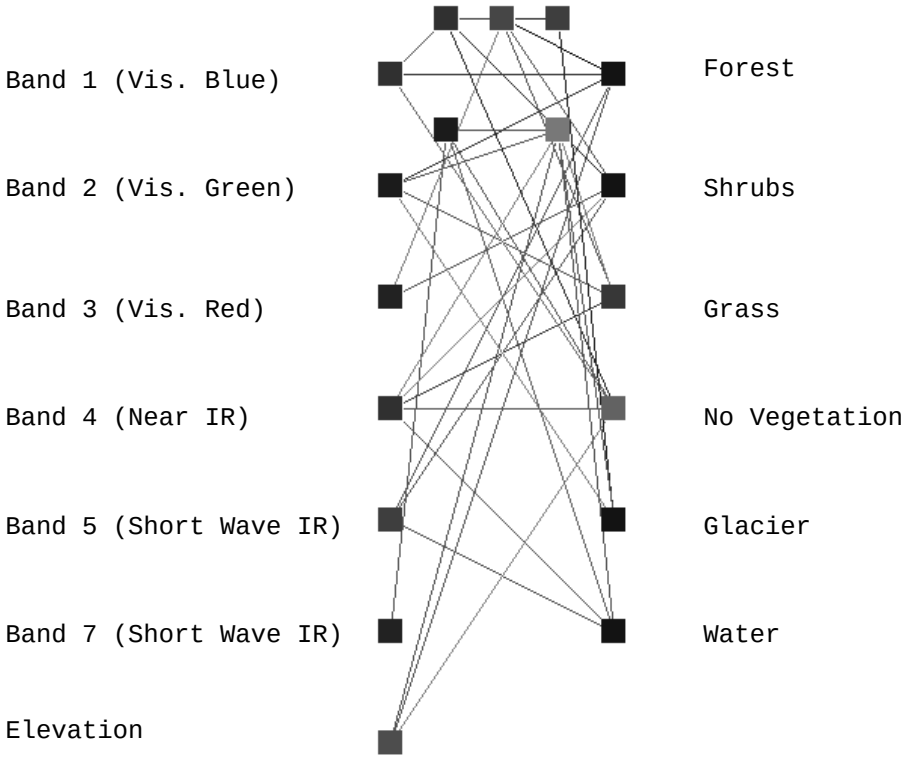

Figure 1: An ANN for Land–Cover classification generated by netGEN, 7 input neurons, 6 output neurons, $\alpha_1 = 0.5$, $\alpha_2 = 0.5$, Regularization term: Rumelhart.

3 Implementation of a System Evolving ANN Topologies

The netGEN system [HMS95] has been implemented on a network cluster of DEC AXP workstations in order to parallelize the learning phase of ANNs. It consists of three main components, the Simple Genetic Algorithm (SGA)[SGE91], the Genotype/Phenotype Mapping (GPM), and the Neural Network Manager (NNM) as shown in figure 2. For the message passing between the individual modules we employed the *PVM* (Parallel Virtual Machine) library [GBD+94].

SGA generates a blueprint for each ANN, a genotype in GA terminology, which is processed by the GPM module mapping the ANN genotype to a phenotype adapted to the Neural Network Simulator [ZMV+94]. The mapping module's task is to prepare valid configuration files and to dismiss ANN topologies which cannot be processed by netGEN. If the topology is valid, a command, e.g. *learn*, is passed to the NNM module. The NNM interfaces to the ANN

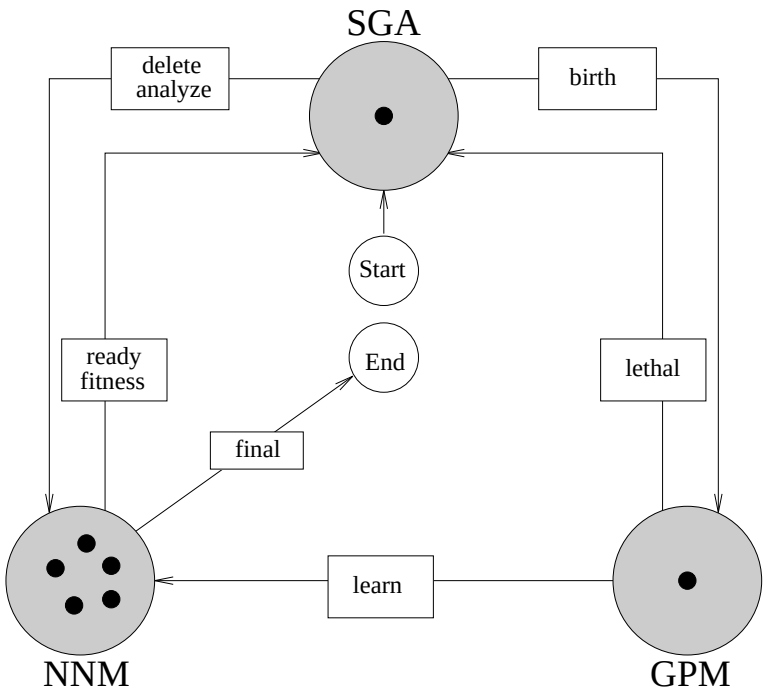

Figure 2: The netGEN system consisting of the Simple Genetic Algorithm (SGA), the Genotype/Phenotype Mapping (GPM), and the Neural Network Manager (NNM).

simulator, evaluates the fitness of each ANN and passes it back to the SGA. With the parallel implementation a set of NNM processes run on different machines and/or different processors.

Furthermore, we like to mention that we achieve *implicit load balancing* which means that more ANNs are passed to the faster machines within the PVM. A simple time–out mechanism prevents the system from deadlock and performance reducing situations when waiting for results from crashed or very slow machines.

4 The Land Cover Classification Problem

Figure 3(a) shows the ground–truth map comprising 6 land cover classes and the corresponding map derived from the netGEN ANN shown in figure 1 applied to the Landsat TM data. The six nonthermal bands of Landsat TM for August 7, 1991, have been geocoded and oversampled to a grid size of 25 meter pixel resolution. For each channel one input neuron was used and the pixel data was scaled to the range zero to one. Additionally, the elevation, also scaled to the range of zero to one, taken from a DEM (Digital Elevation Model) was fed into a seventh input neuron.

Table 2 shows a confusion matrix, where the columns are the reference categories, i.e. the ground–truth categories, and the rows are classification categories. Each entry is given in percent of the number of samples in the considered reference class. The bottom line gives the

producer's accuracy [ZEXJ95], i.e. the number of correctly classified samples from a category to the total number of reference samples of that category.

		Reference Class					
		Forest	Shrubs	Grass	No Veg.	Glacier	Water
Resultant Class	Forest	60.16	15.35	4.73	0.16	0.00	9.08
	Shrubs	36.43	78.58	1.43	0.25	0.00	2.42
	Grass	0.29	3.83	76.94	13.42	0.00	22.31
	No Veg.	0.29	1.26	16.31	82.58	34.91	17.38
	Glacier	0.00	0.09	0.05	3.02	64.01	2.24
	Water	0.90	0.86	0.52	0.55	1.06	46.54
	Overall	71.07					

Table 2: Classification accuracy of the netGEN ANN classifier for each class.

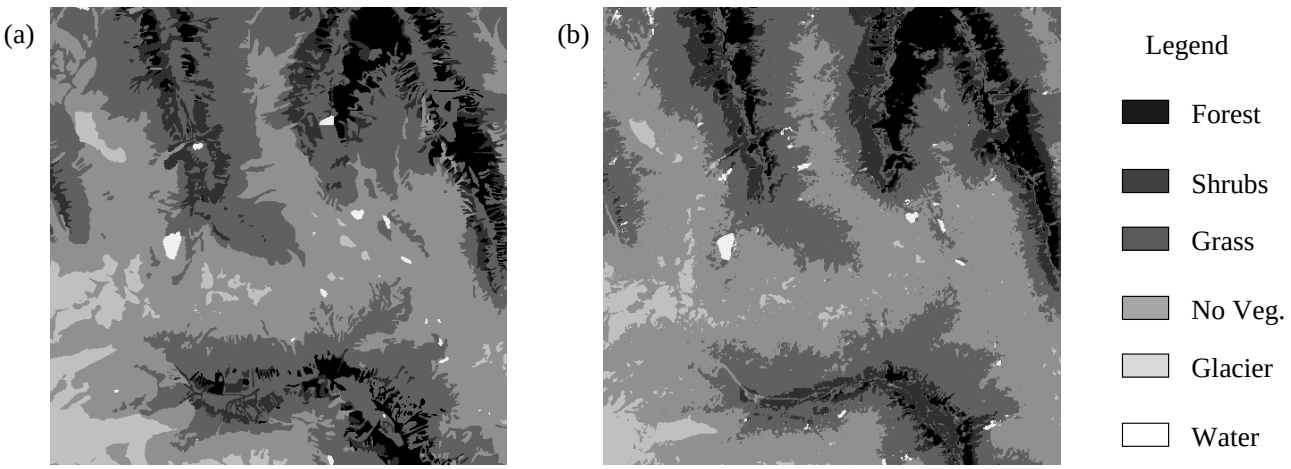

Figure 3: Landsat TM classification: (a) Ground–Truth map, (b) Classification map for the netGEN ANN classifier designed with regularization.

A Maximum–Likelihood classification for the given region has been performed for comparison. The GRASS system (U.S. Army Corps of Engineers, Construction Engineering Research Labs) [WSG88] functions `i.gensig` and `i.maxlik` have been used to generate the training signatures and to perform the Maximum–Likelihood classification. The resulting error matrix is shown in Table 3, figure 4 shows the corresponding classification map compared to the ground–truth.

The large time lag (6 years) between the completion of the vegetation map and the Landsat data take results in a degradation of the ground–truth accuracy. Therefore, two filters have been applied in the preprocessing of the training set, a *k–Nearest Neighbor* (k–NN)–filter in the feature space and another one in the image space [SHM95]. Figure 5(a) shows the k–NN–filter in the image space for $k = 8$. The majority of the neighbor pixels belong to a different class so the pattern is rejected. Figure 5(b) shows the k–NN–filter in the feature space for $k = 8$. In the

		Reference Class					
		Forest	Shrubs	Grass	No Veg.	Glacier	Water
Resultant Class	Forest	83.9	43.5	20.1	0.9	0.0	5.7
	Shrubs	12.3	45.6	19.8	1.5	0.0	0.7
	Grass	2.8	8.5	53.5	38.1	0.8	22.2
	No Veg.	0.1	0.3	4.4	46.7	19.9	6.1
	Glacier	0.0	0.0	0.0	7.4	73.1	4.2
	Water	0.9	2.1	2.1	5.3	6.1	55.0
	Overall	53.6					

Table 3: Classification accuracy of the Maximum–Likelihood classifier for each class.

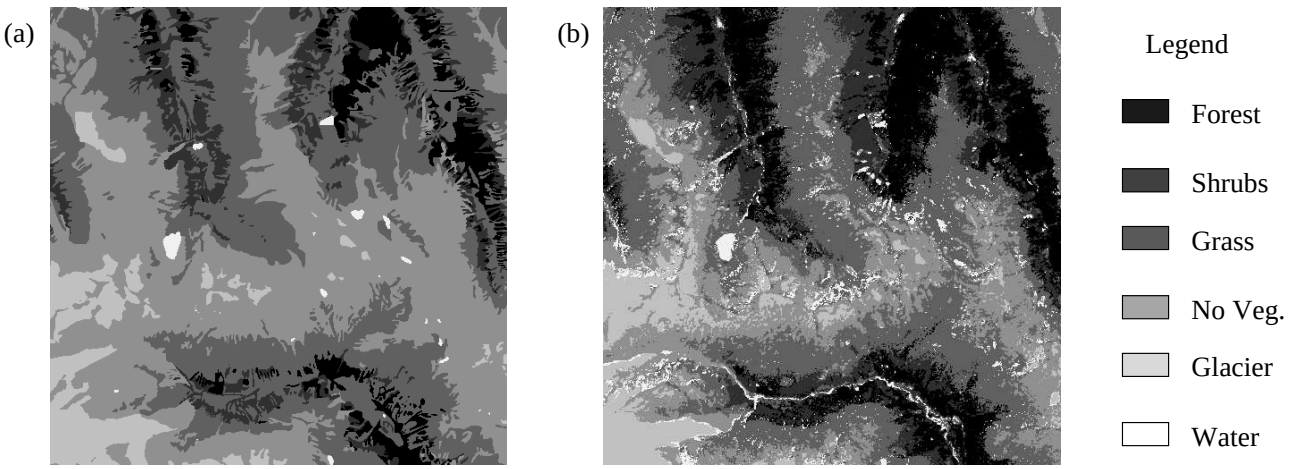

Figure 4: Landsat TM classification: (a) Ground–Truth map, (b) Classification map for the Maximum–Likelihood classifier.

feature space the eight most adjacent feature vectors belong to the same class so the pattern is retained in feature space filtering. In the preprocessing for the land cover classification example we applied both filters with k set to 8 and used only those patterns which are retained by both filters.

5 Discussion

Table 4 reports the parameters and results of the best nets for the land cover problem. As the Hinton term is very small for a large number of weights and has little effect on the overall–fitness (with the exception of $\alpha_2 = 1.0$), we present only one result. With both regularization terms the ANN classification accuracy remains on a good level in a very narrow range for different settings of the α_i, except for the case when choosing α_2 too large, causing an unproportional high influence of the regularization term on the fitness value. As we could observe in all of our

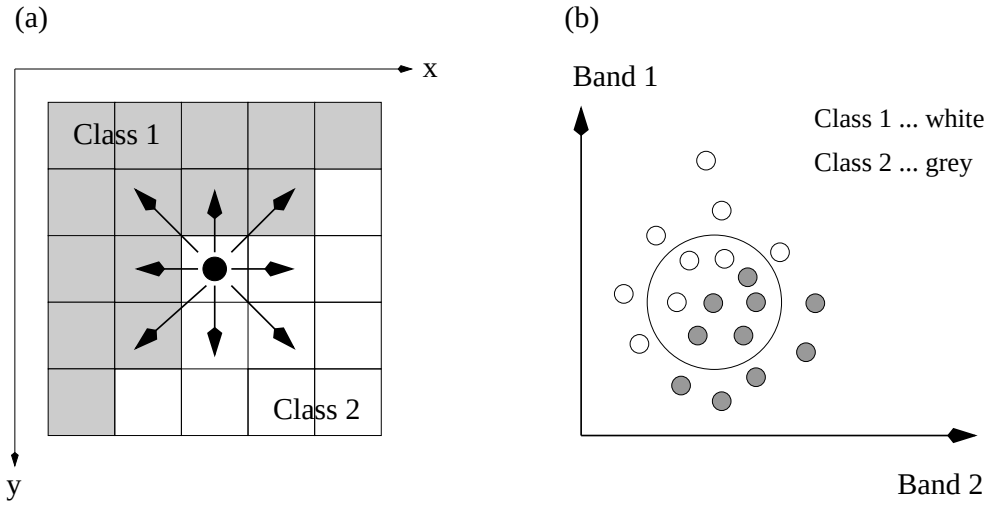

Figure 5: k–Nearest Neigbor filters used in the preprocessing of the training set: (a) k–NN filter in the image space, a pattern is rejected if the majority of its k spatial neigbors is of a different ground–truth class, (b) k-NN filter in the feature space, a pattern is rejected if the majority of its k most adjacent feature vectors is of a different ground–truth class.

previous work, e.g. [HMS95], evolution tends to decrease the number of connections which is enhanced by the use of neuron markers.

α_1	α_2	$\mathcal{F}$	# **hu**	# **c**	**w**			**e**	**a**
					$\sum \lvert w \rvert / C_{total}$	min	max		
1.0	0.0	0.698000	12	165	3.291597	-9.848700	15.475420	1173	0.692665
Regularization: *Hinton*, Neuron Markers: *ON*									
0.5	0.5	0.348625	12	217	3.105659	-13.020380	14.891870	1651	0.700981
Regularization: *Rumelhart*, Neuron Markers: *ON*									
0.9	0.1	0.627934	13	189	3.216468	-14.867800	13.173060	1423	0.688240
0.7	0.3	0.491544	11	151	3.883498	-14.005290	15.469020	1683	0.689571
0.5	0.5	0.363108	5	36	4.793943	-16.778370	11.925910	752	0.710735
0.3	0.7	0.238423	1	14	5.095423	-17.616619	10.360710	580	0.700027
0.1	0.9	0.132638	5	25	0.591409	-2.061530	0.968830	2	0.647305
0.0	1.0	0.136515	1	17	0.485132	-0.883940	0.994490	0	0.070446

Table 4: ANN complexity and accuracy for different weightings of classification and regularization term in fitness evaluation. The abbreviations stand for: (α_i: $i = 1, 2$) = weights for the fitness function $\mathcal{F}$; **hu** = hidden units; **c** = connections; **w** = weights; **e** = epochs, evolutionary determined; **a** = accuracy (image).

With the additional use of regularization parameters the excess weights not only assume values close to zero [Hay94], but disappear, especially, when using the Rumelhart term. Note that the best net discovered (accuracy 0.711) was found with $\alpha_1 = 0.5$ and has less than half

of the hidden units and approximately a quarter of the connections of a net found without using any regularization term. Furthermore, the number of training epochs is also significantly lower. Interestingly, this number is close or equal to zero for fitness functions focusing on regularization. This can be explained by the fact that the initial weights are randomly selected between -1.0 and 1.0, i.e. a net with high regularity. Back–propagation immediately shifts the weights towards irregularity, thus, in this case creating ANNs with lower fitness and driving the number of training epochs towards zero.

6 Acknowledgements

We wish to thank the *Austrian Center for Parallel Computation* (ACPC), Group Salzburg, for making available a cluster of DEC AXP workstations, the *Center for Geographic Information Processing* (ZGIT), Department of Geography, University of Salzburg, for providing TM and ground–truth data and Thomas Fischer of the Austrian Forest–Technical Service in Torrent and Avalanche Control, District Salzburg, for helpful comments on the vegetation map.

References

[BH95] Karthik Balakrishnan and Vasant Honavar. Evolutionary Design of Neural Architectures – A Preliminary Taxonomy and Guide to Literature. Technical Report CS TR #95-01, Iowa State University, Department of Computer Science, Ames, Iowa 50011–1040, U.S.A., January 1995.

[BSP92] H. Bischof, W. Schneider, and A.J. Pinz. Multispectral Classification of Landsat-images using Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*, 30(3):482–490, May 1992.

[BT95] Tobias Blickle and Lothar Thiele. A Mathematical Analysis of Tournament Selection. In L. Eshelman, editor, *Genetic Algorithms: Proceedings of the 6th International Conference (ICGA95)*, San Mateo, California, 1995. Morgan Kaufman Publishers, Inc.

[CDS90] Y.L. Le Cun, J.S. Denker, and S.A. Solla. Optimal Brain Damage. In D. Touretzky, editor, *Advances in Neural Information Processing Systems, Vol. 2*, volume 2, pages 599–605. Morgan Kaufmann, 1990.

[FL90] S.E. Fahlman and C. Lebiere. The Cascade-Correlation Learning Architecture. In D. Touretzky, editor, *Advances in Neural Information Processing Systems, Vol. 2*, volume 2, pages 524–532. Morgan Kaufmann, 1990.

[GBD+94] Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Manchek, and Vaidy Sunderam. *PVM 3 User's Guide and Reference Manual.* Oak Ridge National Laboratory, 1994.

[Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning.* Addison-Wesley, 1989. (ISBN 0-201-15767-5).

[Hay94] Simon Haykin. *Neural Networks. A Comprehensive Foundation.* MacMillan, 1994.

[Hin86] G.E. Hinton. Learning Distributed Representations of Concepts. *Proceeding of the Eighth Annual Conference of the Cognitive Science Society*, 1986.

[HMS95] Reinhold Huber, Helmut A. Mayer, and Roland Schwaiger. netGEN - A Parallel System Generating Problem-Adapted Topologies of Artificial Neural Networks by means of Genetic Algorithms. In *Beiträge zum 7. Fachgruppentreffen Maschinelles Lernen der GI-Fachgruppe 1.1.3, Forschungsbericht Nr. 580, Dortmund*, August 1995.

[HP89] S.J. Hanson and L.Y. Pratt. Comparing biases for minimal network construction with back-propagation. In D. Touretzky, editor, *Advances in Neural Information Processing Systems, Vol. 1*, volume 1, pages 177–185. Morgan Kaufmann, 1989.

[HS91] S.A. Harp and T. Samad. Genetic Synthesis of Neural Network Architectures. In L. Davis, editor, *Handbook of Genetic Algorithms*, pages 202–221. Van Nostrand Reinhold, 1991.

[HSW93] B. Hassibi, D.G. Stork, and G.J. Wolff. Optimal Brain Surgeon and General Network Pruning. In *Proceeding IEEE International Conference on Neural Networks, San Francisco*, volume 1, pages 293–299, 1993.

[KH92] A. Krogh and J.A. Hertz. A Simple Weight Decay can Improve Generalization. In J.E. Moody, S.J. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems, Vol. 4*, volume 4, pages 950–957. Morgan Kaufmann, 1992.

[Lip93] Richard P. Lippman. Neural Networks, Bayesian a Posteriori Propabilities, and Pattern Classification. In *Proceedings of the NATO Advanced Study Institute From Statistics to Neural Networks, Theory and Pattern Recognition Applications, Les Arcs, France*, June 1993.

[MTH89] Geoffrey F. Miller, Peter M. Todd, and Shailesh U. Hegde. Designing Neural Networks using Genetic Algorithms. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 379–384, San Mateo, California, 1989. Philips Laboratories, Morgan Kaufman Publishers, Inc.

[RF95] Paul L. Rosin and Freddy Fierens. Improving Neural Network Generalisation. In *Proceedings of IGARSS'95, Firenze, Italy*, July 1995.

[RPLH89] Jon T. Richardson, Mark R. Palmer, Gunar Liepins, and Mike Hilliard. Some Guidelines for Genetic Algorithms with Penalty Functions. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 191–197, San Mateo, California, 1989. Philips Laboratories, Morgan Kaufman Publishers, Inc.

[SGE91] Robert E. Smith, David E. Goldberg, and Jeff A. Earickson. SGA-C: A C–Language Implementation of a Simple Genetic Algorithm. TCGA Report 91002, The Clearinghouse for Genetic Algorithms, The University of Alabama, Department of Engineering Mechanics, Tuscaloosa, AL 35487, May 1991.

[SHM95] Roland Schwaiger, Reinhold Huber, and Helmut A. Mayer. Land–Cover Classification of Landsat Images Using Problem–Adapted Artificial Neural Networks. In *International Workshop on Soft Computing in Remote Sensing Data Analysis, Milan, Italy*, December 1995.

[WRH91] A.S. Weigend, D.E. Rumelhart, and B.A. Huberman. Generalization by weight elimination with application to fore casting. In D. Touretzky, editor, *Advances in Neural Information Processing Systems, Vol. 3*, volume 3, pages 875–882. Morgan Kaufmann, 1991.

[WSG88] James D. Westervelt, Michael Shapiro, and William D. Goran. GRASS User's Reference Manual. Technical Report N-87/22, USACERL ADP Report, September 1988.

[ZEXJ95] Xin Zhuang, Bernard A. Engel, Xiaoping Xiong, and Chris J. Johannsen. Analysis of Classification Results of Remotely Sensed Data and Evaluation of Classification Algorithms. *Photogrammetric Engineering and Remote Sensing*, 61(4):427–433, April 1995.

[ZMV+94] Andreas Zell, Guenter Mamier, Michael Vogt, Niels Mach, Ralf Huebner, Kai-Uwe Herrmann, Tobias Soyez, Michael Schmalzl, Tilman Sommer, Artemis Hatzigeogiou, Sven Doering, and Dietmar Posselt. *SNNS Stuttgart Neural Network Simulator, User Manual.* University of Stuttgart, 1994.