International Journal of Remote Sensing
Vol. 30, No. 8, 20 April 2009, 2133–2147

Taylor & Francis
Taylor & Francis Group

How many hidden layers and nodes?

D. STATHAKIS*

Joint Research Centre (JRC) of the European Commission, Institute for the Protection and Security of the Citizen (IPSC), Monitoring Agriculture with Remote Sensing (MARS) Unit, Food Security Action, TP266, 21020 (VA), Ispra, Italy

(*Received 19 May 2007; in final form 10 May 2008*)

The question of how many hidden layers and how many hidden nodes should there be always comes up in any classification task of remotely sensed data using neural networks. Until today there has been no exact solution. A method of shedding some light to this question is presented in this paper. A near-optimal solution is discovered after searching with a genetic algorithm. A novel fitness function is introduced that concurrently seeks for the most accurate and compact solution. The proposed method is thoroughly compared to many other methods currently in use, including several heuristics and pruning algorithms. The results are encouraging, indicating that it is time to shift our focus from suboptimal practices to efficient search methods, to tune the parameters of neural networks.

1. Introduction

During any application of neural networks for the classification of remotely sensed data, the same question always rises; how many hidden layers and how many nodes in each layer should be used? Although it has been almost two decades now since the first introduction of neural networks in remote sensing (Benediktsson *et al* 1990) there exists no exact method to answer this question (Mas and Flores 2008), and it is a critical question since the selection of topology has a profound impact on classification results.

Traditionally identification of topology has been based on trial and error, on heuristics sometimes followed by trial and error, and on pruning or constructive methods as discussed in the following section. None of these methods has the theoretical rigor of revealing optimal or at least near-optimal solutions. The objective of this paper is to bridge this chasm by presenting a method based on genetic algorithms (Goldberg 1989, Holland 1992a, b). The proposed solution can be summarized as the synergy of a neural network and a genetic algorithm that searches for topologies, based on a novel fitness function, aiming to concurrently optimize performance while minimizing network complexity. The pressure on compactness can be varied according to specific needs.

In the sequel current practices with respect to setting hidden layer topology are reviewed in §2. The proposed method based on searching with the genetic algorithm is introduced in §3. It is then applied in a case study and compared with the state-of-the-art methods in §4. Finally, conclusions are briefly discussed in §5.

*Email: dstath@uth.gr

International Journal of Remote Sensing
ISSN 0143-1161 print/ISSN 1366-5901 online
http://www.tandf.co.uk/journals
DOI: 10.1080/01431160802549278

2. Current methods

In this section the methods currently deployed to identify efficient network topologies are presented. Before that the theoretical bounds of optimal topologies are discussed.

2.1 *Theoretical bounds of optimal topology*

The root of any discussion on topological bounds is probably Kolmogorov's theorem stating that any continuous function defined on an n-dimensional cube can be represented by sums and superpositions of continuous functions of one variable (Kolmogorov 1957). Hecht-Nielsen (1987) imported this theorem later in neuro-computing by proving that any continuous function can be represented by a neural network that has only one hidden layer with exactly $2n+1$ nodes, where n is the number of input nodes. Several authors in remote sensing use this $2n+1$ figure as a panacea to either fix the number of nodes in the first hidden layer or to justify the lack of need to search for topologies that have two hidden layers. This is not the case however as Hecht-Nielsen stated that the $2n+1$ rule is not for any class of activation functions but for a specific one (Hecht-Nielsen 1987). This activation function is much more complex, compared with the commonly used sigmoidal functions. It has been suggested (Kurkova 1992) that two hidden layers should be used to compensate for lost efficiency when using regular activation functions. The argument that it is sufficient to use a single hidden layer still holds when using regular transfer functions (e.g. sigmoidal) but the number of required hidden nodes can be as high as the number of training samples (Huang and Babri 1997, Huang 2003). The purpose of using a second hidden layer is to drastically reduce the total required number of hidden nodes. Huang (2003) proved that in the two-hidden-layer case, with m output neurons, the number of hidden nodes that are enough to learn N samples with negligibly small error is given by

$$2\sqrt{(m+2)N}. \tag{1}$$

Specifically, he suggests that the sufficient number of hidden nodes in the first layer is

$$\sqrt{(m+2)N}+2\sqrt{N/(m+2)}, \tag{2}$$

and in the second it is

$$m\sqrt{N/(m+2)}. \tag{3}$$

As the optimal topology is judged by the capacity to generalize on unseen data, the most accurate structure will have fewer nodes than that suggested by equations (1)–(3) that are good to over-fit the training data.

In summary, topologies with a first and a second hidden layer should be searched having at worst as many hidden nodes as that suggested by equations (2) and (3) respectively.

2.2 *State of the art*

In the absence of an exact paradigm to estimate an optimally or near-optimally performing neural network structure, four methods are currently deployed.

2.2.1 Trial and error. This is the most primitive path, and it is not uncommon to yield severely suboptimal structures especially when adopted by inexperienced users.

2.2.2 Heuristic search. Several heuristics exist in the literature amalgamating knowledge gained from previous experiments on where a near-optimal topology might exist. The objective is principally to devise a formula that estimates the number of nodes in the hidden layers as a function of the number of input and output nodes. The estimate can take the form of a single exact topology to be adopted (Hecht-Nielsen 1987, Hush 1989, Wang 1994, Ripley 1993) or of a range of topologies that should be searched (Fletcher and Goss 1993, Kanellopoulos and Wilkinson 1997). Some heuristics come from the literature regarding neural networks in general (Hecht-Nielsen 1987, Fletcher and Goss 1993, Ripley 1993), whereas others have been introduced by experimenting with spatial data (Paola 1994, Wang 1994, Kanellopoulos and Wilkinson 1997). In practice these heuristics are frequently used as points of departure for subsequent search by trial and error. This seems to be a wise approach as in most cases the heuristics lack the theoretical evidence to support the discovery of an optimal structure. Note that none of the heuristics presented here takes into consideration the number of samples which is an integral part in equations (1)–(3).

2.2.3 Exhaustive search. Searching through all possible topologies is normally not an option for any real-world application. It is not that the number of alternatives is exceedingly large but rather that the time required to evaluate each alternative is long. An exhaustive search is further complicated due to the noisy fitness evaluation problem (Yao 1993), i.e. the fact that neural networks produce different results due to different initialization conditions even when everything else is kept fixed. This is why a single run is actually not enough to evaluate a topology. Multiple runs are in fact needed. In this experiment, it has been found that the side-effects of the noisy fitness evaluation problem are reduced by increasing the number of samples. It is known that in remote sensing, the availability of samples varies widely with circumstances. If, however, samples are available, the number of samples used should be progressively increased while observing the variance of the classification results. There is a point where the inclusion of additional samples yields no benefit towards the stabilization of results.

2.2.4 Pruning and constructive algorithms. Pruning and constructive algorithms aim at devising an efficient network structure by incrementally adding or removing links (weights) in a network that has redundantly more or initially none respectively. Optimal Brain Damage (Le Cun *et al.* 1990) is a commonly used pruning algorithm that progressively removes the weight that causes the least increase in training error. To simplify computation, it makes the assumption that the network's Hessian matrix is diagonal. Optimal Brain Surgeon (Hassibi and Stork 1993) makes no such assumptions, and it demands no retraining after the pruning of a weight. Kavzoglu and Mather (1999) recently conducted a comparison of the most common methods and concluded that Optimal Brain Surgeon outperforms the rest.

3. Proposed method

As will be apparent shortly, the proposed method is novel in two aspects. First, although genetic algorithms are used in other domains, it seems that they are frequently neglected in remote sensing where methods with limited capacity in

estimating network structures are still in use. As a result, the effectiveness of neural networks is frequently compromised. A second contribution of this work is that a novel fitness function is introduced to evaluate each solution. The function aims at concurrently maximizing classification accuracy and at the same time minimizing network complexity.

3.1 *Symbiosis of genetic algorithms and neural networks*

The proposed method is based on the synergy of genetic algorithms and neural networks. The problem can be formulated as a search task in the architecture space where each point represents an architecture (Liu *et al.* 2000). Following the discussion in §2.1, topologies with up to two hidden layers are searched. The number of nodes of the two hidden layers of the network structure is directly coded in a binary chromosome. The length ℓ of the chromosome is 10 bits; the first six are reserved for the first hidden layer, whereas the remaining four are for the second hidden layer as shown in figure 1. The process involves the transformation of the binary numbers, corresponding to the number of nodes in each hidden layer, to decimal numbers. A constant is then added to reduce computation requirements. Ideally, to be able to search for a wider range of topologies, more bits would be required. Preferably, up to the numbers set by equations (2)–(3) should be searched for the first and second hidden layer respectively. Due to performance limitations the string is restricted to what is mentioned above. This does not skew the results, as the lower topologies excluded are clearly suboptimal by containing too few nodes. The actual results justify also the use of the higher limit of the topological structures as the optimal topologies are found below the upper bound of searched topologies. As shown in figure 1, no constant is added to the number of nodes in the second hidden layer. This is done in order to maintain the possibility of constructing and evaluating networks with only one hidden layer, i.e. having zero nodes in the second hidden layer. The importance of this implementation path is that it gives the chance to the genetic algorithm to answer the question of how many hidden layers are sufficient. In summary, topologies between 10:[20–83]:[0–15]:5 are searched. The

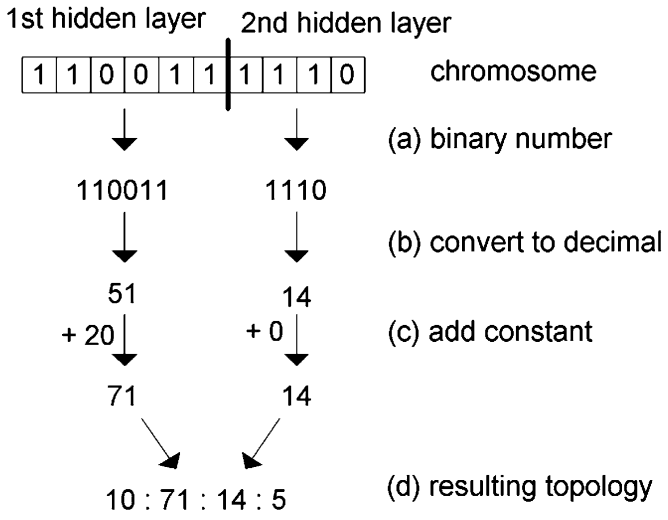

Figure 1. Mapping of the two hidden layer topology of the neural network to a chromosome to be optimized by the genetic algorithm.

network is eventually built based on the number of nodes in each layer dictated by the genetic algorithm. It is then trained and its performance evaluated according to the fitness function set. The accuracy matrix (Congalton 1998) is constructed after each run in order to evaluate the results. The performance metric used here is overall accuracy. This could easily be extended however to include other indices such as the Kappa coefficient (k-hat).

The neural network settings are kept constant throughout the process. Training is performed with a fully connected feed forward multilayer perceptron using the Levenberg–Marquardt back-propagation variant (Werbos 1974, Rumelhart *et al.* 1986, Hagan and Menhaj 1994). Learning rate is set to 0.03 and momentum to 0.9. Both values are constant though time and are used in all methods compared. Note however that the specific training algorithm ensures very fast and efficient training. The transfer function is tan-sigmoid. Training is terminated when the overall testing data set accuracy drops. The winner-takes-all rule is applied to the output of the neural network to obtain a single output class per input vector. All input data are normalized to $[-1, 1]$.

On the side of the genetic algorithm most choices are based on the recommendations of Goldberg (2002). Two-point crossover is set to 0.75, and tournament selection with a size of 4 (Goldberg 2002, chap. 8) and uniform mutation of 0.01 is used. Fitness is scaled by rank. Population n, maximum number of function evaluations and maximum number of generations are set respectively, according to Goldberg (2002, chap. 10), as:

$$n = \ell^2 \log_{10}^2(\ell) = 100 \tag{4}$$

$$\max(\text{function-evaluations}) = \ell \log_{10} \ell = 10 \tag{5}$$

$$\max(\text{generations}) = \ell^2 \log_{10}^2(\ell)/n = 10. \tag{6}$$

These settings have been found to work well in practice.

3.2 *Accuracy-over-compactness fitness function*

The assumption is made that compactness, i.e. having as few nodes in the topology as possible, is considered an additional merit to overall verification set classification accuracy. It is essentially a multi-objective task of finding the most efficient and at the same time the less redundant network skeleton. The formula of the novel fitness function is:

$$f = e + s\frac{c - c_{\min}}{c_{\max} - c_{\min}}. \tag{7}$$

Overall verification classification error e corresponds to that of the current topology. The complexity factor c is measured as the number of weights in each searched topology. The parameters $c_{\max}$ and $c_{\min}$ correspond to the maximum and minimum complexity allowed given the length of the chromosome used to code solutions. The chromosome used in this case, shown in figure 1, contains 6 bits for the first hidden layer. The maximum decimal number that can be represented by 6 bits is 63. Given that we choose to add the constant of 20 to the number of nodes in the first hidden layer (figure 1), the maximum number of nodes in the first hidden layer becomes $63 + 20 = 83$. In a similar manner the maximum number of nodes for

the second hidden layer is 15 as we have four reserved bits in the chromosome and zero is added (figure 1). The highest allowed topological structure is thus 10 : 83 : 15 : 5 which has 2150 degrees of freedom; hence $c_{max}=2150$. At the other end, the minimum topology allowed is 10 : 20 : 5 with 300 degrees of freedom or $c_{min}=300$. The two values c_{min} and c_{max} are easily calculated based on the number of bits used for each layer in the chromosome plus a constant that we chose to add. The overall objective of using c_{max} and c_{min} is to normalize the complexity of each topology tested to the interval [0,1] so that it can be smoothly incorporated to the fitness function. In the ideal case this should be $c_{min}=1$ and $c_{max}=+\infty$ but in practice these values are set according to the computing power available so that the algorithm converges at a reasonable time. The accuracy sacrifice percentage s is the only user-specified parameter. The meaning of it is how much accuracy we are willing to sacrifice for a more compact solution. If s is set to 1 (default) in effect it is assumed that solutions that are up to 1% less accurate can be considered more fit if they are more compact. If we set $s=0$, compactness ceases to be an objective. In any case, the s parameter is an absolute threshold. Essentially it is assumed that the objective is merely the minimization of overall verification classification error. By exerting no other pressure it is expected that the most accurate individuals in the population with prevail.

With this fitness function accuracy is preferred to compactness. The range of s parameter is problem-dependent. In easy classification problems it can be relaxed whereas in difficult ones it has to remain low. Note in addition that in multi-objective optimization problems, for the non-dominated solutions, following the Paretto terminology, it is not trivial to decide which is best (Goldberg 1989, chap. 5).

This fitness function can be easily used by other researches since the only user-defined coefficient has quite a straightforward meaning. Note also that there is no need for logarithmic scaling of complexity or accuracy since rank selection is adopted for the genetic algorithm. This in turn relieves us from setting the curvature of the logarithmic function which used to be one of the main problems in previous fitness functions such as in (Siedlecki and Sklansky 1989). We are quite confident in this choice as rank-based selection procedures outperform proportional ones (Goldberg 2002, chap. 8, p. 114).

4. Experimental results

4.1 *Data set and sampling*

The experimental data set refers to Lefkas island in the western part of the Hellenic Republic. Inputs to the system are seven LANDSAT 7 ETM + bands; plus elevation, slope and aspect (table 1) which are derived from SRTM data (SRTM, 2000). The LANDSAT scene was acquired on July 2000. Outputs are five CORINE Level 1 land use classes (CORINE, 2000), *viz*. artificial surfaces, agricultural areas, forest and semi-natural areas, wetlands, and water bodies. The CORINE landcover data set is used as the reference to annotate the output vector. Notably the dimensions of this multisource dataset are relatively small compared with hyper-spectral imagery. However, most of the classification tasks in remote sensing are done in that order of dimensionality. In addition, using the proposed method with hyper-spectral data would require more powerful non-standard computer hardware.

As many as 19 044 samples are selected corresponding to 1% of the total available. A stratified random sampling strategy is adopted so that all classes are equally

Table 1. Characteristics of the samples used for training, validation and testing (original values).

Band	Mean	Median	Standard deviation	Min	Max
LANDSAT 1	92.9	90	14.2	71	202
LANDSAT 2	75.1	73	20.4	42	206
LANDSAT 3	76.5	72	30.0	29	251
LANDSAT 4	62.7	74	30.2	11	148
LANDSAT 5	91.8	99	53.9	11	254
LANDSAT 6	154.7	158	12.7	130	187
LANDSAT 7	61.6	60	37.7	9	217
DEM	145.6	26	240.5	−40	1140
Slope	9.9	6.9	10.7	0	65.8
Aspect	129.6	109.25	119.4	−1	35.96

represented (~3800 samples per class). That percentage is set by trial and error. The classification is done also for the remaining 99% of the data and the results evaluated after the topology is fixed. The samples are split into equal parts (i.e. 6348 samples each) to form three sets namely training, testing, and validation. The terminology is consistent with Bishop (1995, chap. 9). The testing set is used to terminate training done with the training set. The validation set is kept independent and used in accuracy assessment only after training has converged. Re-sampling has been tested several times with negligible impact on the results.

4.2 *Method comparison*

4.2.1 Heuristics. In table 2 the results of several commonly used heuristics are displayed. Many of those heuristics have also been used in Kavzoglu and Mather (1999) although somewhat differently. For example, the Kanellopoulos–Wilkinson rule originally refers to a range of topologies rather than exact ones as implied in Kavzoglu and Mather (1999). Also, the upper bound is up to four times the number of nodes in the input layer (Kanellopoulos and Wilkinson 1997, Stahakis and Vasilakos 2006). Thus, results for the low, medium, and high bound of this rule are shown. The general observation is that larger topologies yield better results or, stated another way, most of the heuristics tend to underestimate the complexity of this classification problem. This conforms to findings in previous work in a completely different context (radar for oil spill detection; Stathakis *et al.* 2006). In that paper the number of input features is concurrently evolved with the number of nodes, in a binary output classification problem (oil spill or look-alike) with only one hidden layer and a standard fitness function. Conversely, the current paper is focused on selecting an optimal hidden topology while keeping the features to be used constant in a much more complex classification problem with five land-cover types, one or two hidden layers and a novel fitness function. Furthermore, small topologies produce more unstable results. None of the heuristics suggests the use of a second layer for that many nodes in the input and output layers.

4.2.2 Pruning. Two pruning methods are tested *viz*. Optimal Brain Damage and Optimal Brain Surgeon. The maximum topology proposed among all heuristics, a 10 : 40 : 5 structure, is adopted as the starting point. All testing samples are used to guide pruning, and all verification set samples are used to calculate the overall accuracy. The progress of pruning for both algorithms is shown in figure 2. It is

Table 2. Method comparison for the same data set. Results for the heuristics are averaged over 50 runs.

Method name	Reference	Range	Topology	Mean	Max	Min	σ
Hecht–Nielsen rule	Hecht-Nielsen (1987)		10 : 21 : 5	70.68	71.92	69.45	0.63
Kanellopoulos–Wilkinson rule also Hush rule	Kanellopoulos and Wilkinson (1997)	Low	10 : 20 : 5	70.54	71.87	68.48	0.69
		Medium	10 : 30 : 5	73.42	74.98	71.19	0.71
	Hush (1989); Kanellopoulos and Wilkinson (1997)	High	10 : 40 : 5	73.91	75.21	71.77	0.75
Wang rule	Wang (1994)		10 : 7 : 5	62.97	65.23	20.78	6.17
Ripley rule	Ripley (1993)		10 : 8 : 5	64.05	67.06	22.93	6.19
Fletcher–Goss rule	Fletcher and Goss (1993)						
	Low		10 : 11 : 5	66.29	68.76	63.43	0.90
	Medium		10 : 16 : 5	70.30	72.23	63.17	1.25
	High		10 : 21 : 5	70.68	71.92	69.45	0.63
Paola rule	Paola (1994)		10 : 22 : 5	70.39	73.02	20.12	7.29
Garson	(1998)	$r=5$	10 : 42 : 5	72.88	75.18	19.41	7.74
Garson		$r=10$	10 : 85 : 5	74.12	77.45	19.96	9.14
Optimal Brain Surgeon (OBD)	LeCun *et al.* (1990)			–	75.48	–	–
Optimal Brain Damage (OBS)	Hassibi and Stork (1993)			–	75.26	–	–
Genetic algorithm ($s=1$)	Proposed method	10 : [20–83] : [0–15] : 5	10 : 73 : 10 : 5	70.53	78.34	19.96	15.29
			10 : 73 : 5	71.23	77.03	19.96	15.29
Genetic algorithm ($s=0$)		10 : [20–83] : [0–15] : 5	10 : 74 : 14 : 5	71.62	79.63	24.33	13.44

Mean, max, and min accuracy (%) are measured for the verification set. Standard deviation is σ. Both Optimal Brain pruning methods start by a 10 : 40 : 5 topology. Furthermore, it is practically infeasible to run the two Optimal Brain methods 50 times, as the time required is prohibiting. Hence the corresponding statistics are not reported here.

evident that while both algorithms can remove a number of links without any significant accuracy degradation, there is hardly any improvement in accuracy. These results are well in accordance with those in Kavzoglu and Mather (1999). The best classification results are shown on table 1. The computation resources required for both algorithms are high, with Optimal Brain Surgeon being more demanding.

4.2.3 Genetic algorithm with accuracy over compactness. Note that in this classification problem ten input and five output neurons are needed. Furthermore, 6348 vectors consist each of the sample data sets. Hence, according to equations (2)–(3), the theoretical upper bound for this problem is a 10 : 271 : 151 : 5 structure.

Setting in the proposed fitness function $s=1$, the complexity of two representative solutions evolves, as shown in figure 3. In both cases the number of nodes in the first hidden layer is increased to about 70 (maximum is 83). It follows that this is the minimum number of nodes to achieve near-optimal performance. Regarding the number of nodes in the second hidden layer, there are two different results. In the first run, the number increases to 10 nodes (the maximum is 15) indicating that a

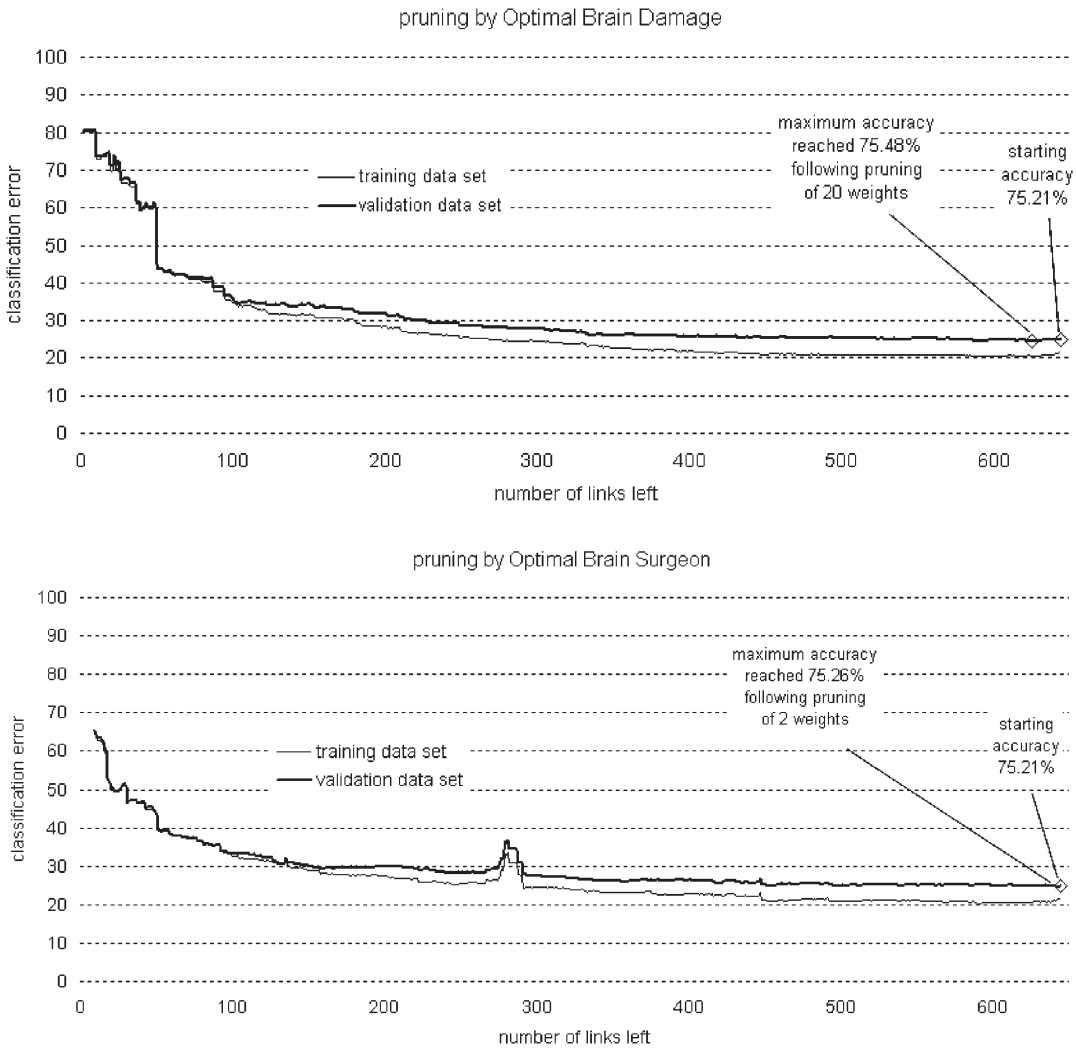

Figure 2. Pruning a 10 : 40 : 5 network by Optimal Brain Damage (top) and by Optimal Brain Surgeon (bottom). The algorithm can find a network that performs equally well and has approximately 150 fewer links. Performance is only negligibly increased, however.

10 : 73 : 10 : 5 solution is optimal by yielding 78.34% accuracy with respect to the verification set. In the second run, however, after five generations it becomes apparent that a second hidden layer might not be required at all. Thus, the discovered optimal solution is a 10 : 73 : 5 structure resulting in a slightly worse accuracy of 77.03%. The solution space searched as well as the locations where near-optimal neural network skeletons have been discovered are presented in figure 4. The solution space for this data set is quite small, and as a result it is quite probable that the same architecture can be assessed more than once. This fact has a positive effect as it might smoothe the impact of the noisy fitness evaluation problem somewhat.

When the sacrifice percentage is set $s=0$, solutions that optimize performance regardless of complexity are in fact sought. Quite naturally when no pressure on complexity is put ($s=0$) the accuracy is higher as opposed to when some pressure is exerted ($s=1$). The evolution of solutions is shown in figure 5 where five runs are averaged. Accuracies between 76.41% and 79.63% are achieved, in all cases better

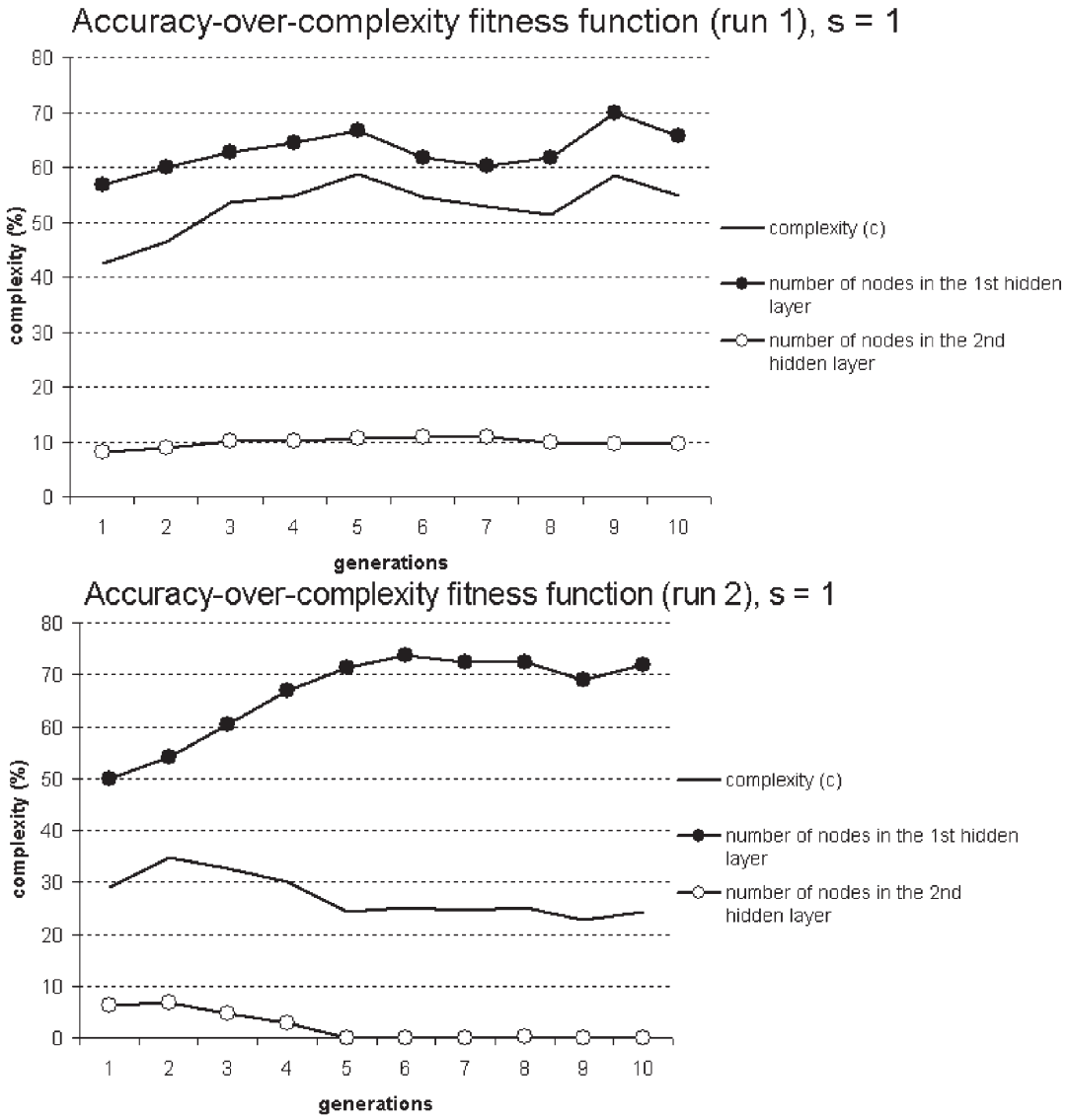

Figure 3. Two typical runs with the accuracy-over-complexity fitness function. The sacrifice percentage is set to $s=1$. In run 1, a 10 : 73 : 10 : 5 solution is discovered after 10 generations, yielding an accuracy of 78.34%. In run 2, a different evolution path reveals a single-hidden-layer topology of 10 : 73 : 5 with accuracy of 77.03% as optimal. This figure shows complexity not accuracy.

than that of conventional methods. As is evident in figure 5, because no pressure is put on it, complexity grows generation by generation.

Both the classification results using the most accurate solution (10 : 73 : 10 : 5) and the reference data are shown on figure 6 for comparison. The accuracy matrix for this topology is presented in table 3. Let us recall the fact that genetic algorithms do not converge to a single point in the solution space. This known property can be turned into a strong advantage as several near-optimal individuals are produced, highlighting different aspects of feasible solutions. In this case some solutions are slightly more accurate whereas others are less complex. Some solutions have one whereas others have two hidden layers. This phenomenon gave rise to the theory of ensembles (Liu *et al.* 2000).

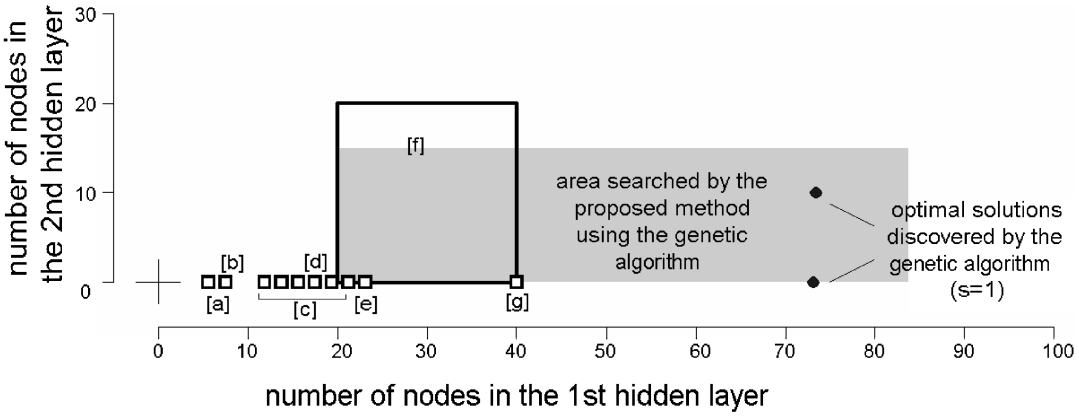

Figure 4. Areas searched by different methods tested in this paper as shown in table 1. The proposed method searches in the shaded area and reveals several solutions that are superior to those suggested by heuristics. The small squares show actual topologies suggested by heuristics in the literature that were compared here. Letters in the square brackets show the reference in which the heuristic was introduced, i.e. [a]=Wang (1994), [b]=Ripley (1993), [c]=Fletcher and Goss (1993), [d]=Hecht-Nielsen (1987), [e]=Paola (1994), [f]=Kanellopoulos and Wilkinson (1997), [g]=Hush (1989).

The fact that in this experiment setting $s=0$ or $s=1$ produces accuracy of a similar level does not mean that that complexity plays a negligible role in the optimization of the fitness function proposed. The explanation is rather that this classification problem is particularly difficult. Because this classification problem is so difficult, it is pointless to relax the sacrifice percentage, i.e. test for $s=2$, $s=3$, etc. Putting more pressure on complexity will eventually lead to less accurate solutions as there is no space for large improvements here. It is expected that this will be different in easier classification problems.

5. Conclusion

In summary, a method is presented here that addresses the design of topology in neural networks for classification problems. The solution is achieved in a relatively automated fashion. The proposed fitness function contains essentially one

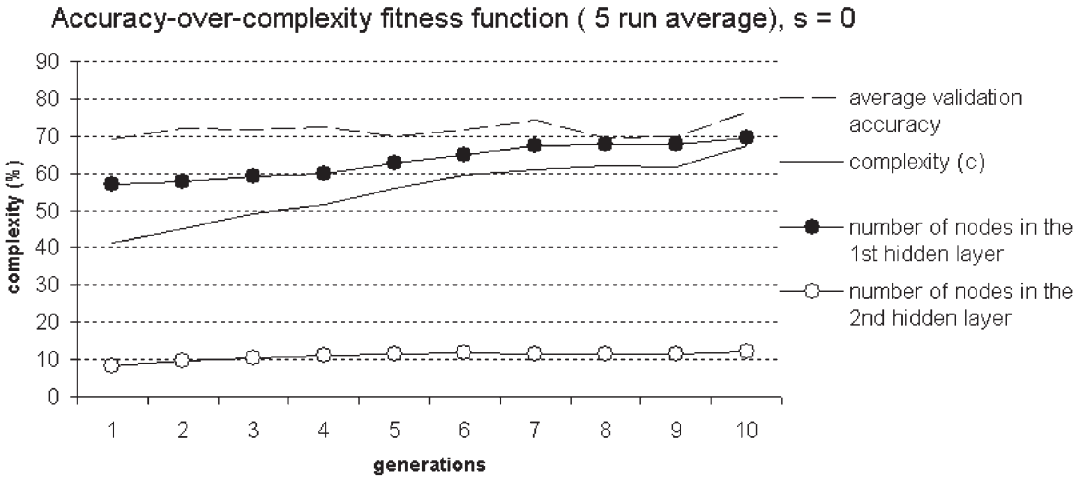

Figure 5. Average of five runs after setting $s=0$ in the proposed fitness function. Complexity grows by generation as a result of putting no pressure on it.

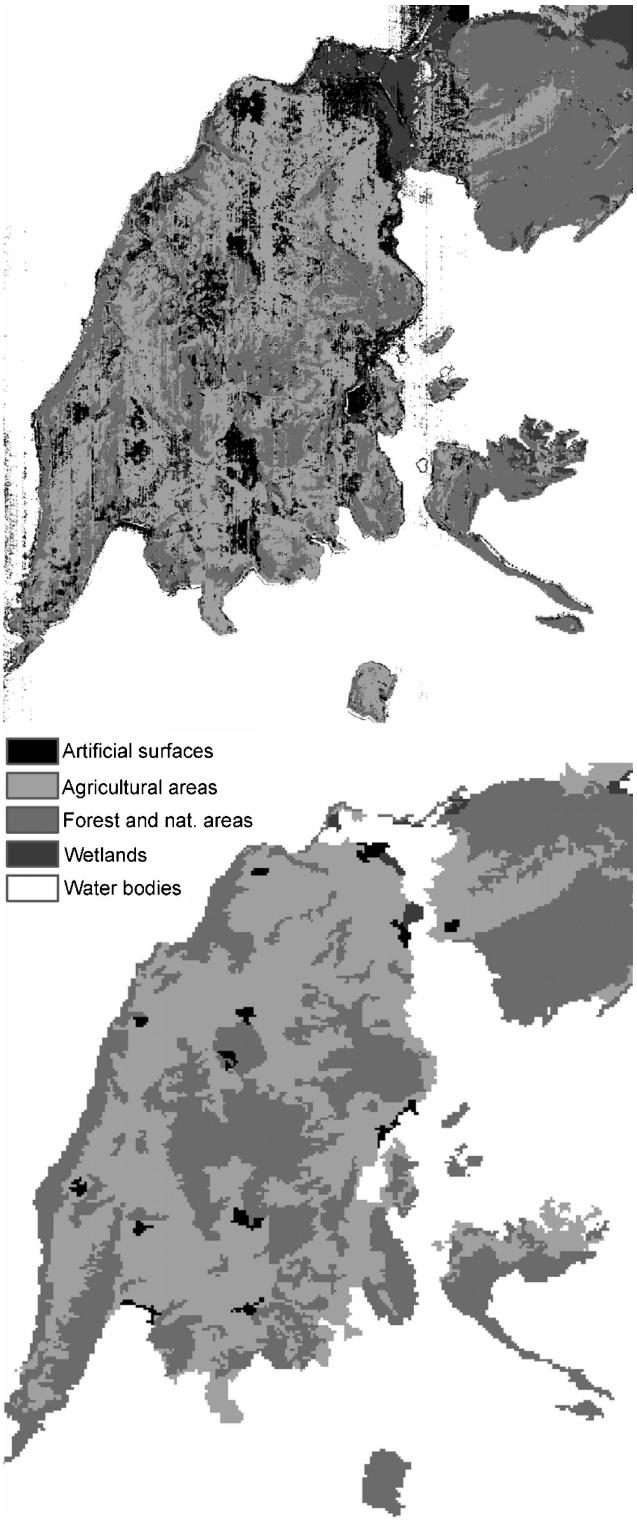

Figure 6. Classification results (top) and the reference source used CORINE (bottom) for comparison. The best-performing network discovered by the proposed method ($s=1$) is used for the classification yielding 78.34% accuracy for the validation data set and 80.41% accuracy for the complete image. The striping present in the top image is not present in the bottom image because the reference source (CORINE) was compiled using manual photo-interpretation as well as a subset of the available input bands.

Table 3. Accuracy matrix for the best topology, discovered by the proposed method, i.e. 10 : 73 : 10 : 5.

Training Khat=0.81	ATF	WET	FOR	WAT	AGR	User's
ATF	1123	137	64	51	12	81.0
WET	57	870	212	16	9	74.7
FOR	35	192	937	7	11	79.3
WAT	38	30	24	1163	19	91.3
AGR	32	10	31	1	1258	94.4
Producer's	87.4	70.2	73.9	93.9	96.1	84.41
Testing Khat=0.73	ATF	WET	FOR	WAT	AGR	User's
ATF	1018	196	82	93	14	72.6
WET	94	770	242	31	10	67.1
FOR	52	217	847	17	15	73.8
WAT	61	47	41	1171	50	85.5
AGR	40	21	41	3	1141	91.6
Producer's	80.5	61.6	67.6	89.0	92.8	78.34

'Producer's' and 'User's' refer to the percentage accuracy.

user-defined parameter with intuitive meaning. The other two parameters ($c_{\min}$ and $c_{\max}$) are bound to the computing resources available. It is found to outperform all conventional paradigms tested. Specifically, the heuristics tend to underestimate complexity. Of these, the Kanellopoulos–Wilkinson rule gives the best results in terms of accuracy. State-of-the-art pruning methods are, by design, efficient in pruning but not in increasing accuracy towards the discovery of a near-optimal solution. On the number of hidden layers, when seeking to optimize accuracy the use of a second layer is desirable. Overall the results of the neural networks tested in this experiment are heavily dependent on the topology chosen. The variance of classification accuracy across all topologies explored, shown in table 1, is approximately $\pm 14\%$.

Regarding the computational requirements it became evident that pruning has comparable demands to the proposed method based on the genetic algorithm. Currently a little more than one day is required to complete training with the proposed method and a little less than one day with pruning. All tests were done on a computer with a dual 2.80-GHz processor and 3 GB of RAM.

Heuristic searching is fast but clearly sub-optimal. Speed may appear as a desirable trade-off in some applications, but here the goal set is primarily accuracy. In conclusion, it should be considered preferable to search for one day and have evidence that a near-optimal solution is discovered as opposed to searching for practically the same amount of time based on heuristics without any evidence that a better solution is infeasible. The proposed method also contains some elements that need to be set ad hoc. Nevertheless searching is based on a far more efficient method than the deterministic method. Also, the setting of the parameters is quite easy and straightforward as opposed to the previously used fitness functions such as that used in Stathakis *et al* (2006).

Ideally searching with the genetic algorithm should cover for the theoretical bounds of the number of hidden nodes per layer, but this can be relaxed in practice. It is important to note that when the search range set is insufficient, the algorithm

indicates so by pointing to solutions on the edge of it. In the future, the same fitness function can be tested to concurrently perform feature selection as well, by incorporating the number of input nodes in the complexity coefficient. The method was tested here in one example that is not particularly large. There should be no reason why this method will not work on larger problems provided that computing power is available. It is interesting to note that the concurrent evolution of input dimensionality will lead to feature reduction, which in turn reduces the topology needed. It is expected that this property will aid the method to scale up in larger classification problems.

References

Benediktsson, J., Swain, P. and Ersoy, O., 1990, Neural network approaches versus statistical methods in classification of multisource remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, **28**, pp. 540–552.

Bishop, C., 1995, *Neural Networks for Pattern Recognition* (Oxford: Oxford University Press).

Congalton, R., 1998, *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices* (Boca Raton, FL: Lewis).

COoRdinate INformation on the Environment (CORINE), program of the European Commission to provide consistent information on land cover across Europe. Available online at: http://image2000.jrc.it/ (accessed 2 May 2008).

Fletcher, D. and Goss, E., 1993, Forecasting with neural networks: an application using bankruptcy data. *Information and Management*, **24**, pp. 159–167.

Garson, G.D., 1998, *Neural Networks: An Introductory Guide for Social Scientists* (London: Sage).

Goldberg, D., 2002, *The Design of Innovation: Genetic Algorithms and Evolutionary Computation*, 1st ed. (New York: Springer), p. 272.

Goldberg, D.E., 1989, *Genetic Algorithms in Search, Optimization & Machine Learning* (Reading, MA: Addison-Wesley).

Hagan, M.T. and Menhaj, M., 1994, Training feed-forward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, **5**, pp. 989–993.

Hassibi, B. and Stork, D.G., 1993, Second-order derivatives for network pruning: optimal brain surgeon. In *Advances in Neural Information Processing Systems 5*, S.J. Hanson, J.D. Cowan and C.L. Giles (Eds), pp. 164–171 (San Mateo, CA: Morgan Kaufmann).

Hecht-Nielsen, R., 1987, Kolmogorov's mapping neural network existence theorem. In *IEEE First Annual International Conference on Neural Networks*, **3**, pp. 11–13.

Holland, J., 1992a, *Adaptation in Natural and Artificial Systems*, 2nd ed. (Cambridge, MA: MIT Press).

Holland, J., 1992b, Genetic algorithms: Computer programs that 'evolve' in ways that resemble natural selection can solve complex problems even their creators do not fully understand. *Scientific American*, **July**, pp. 44–50.

Huang, G.-B., 2003, Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Transactions on Neural Networks*, **14**, pp. 274–281.

Huang, G.-B. and Babri, H., 1997, General approximation theorem on feedforward networks. In *International Conference on Information, Communications and Signal Processing*, ICICS '97, Singapore, 9–12 September, pp. 698–702.

Hush, D.R., 1989, Classification with neural networks: a performance analysis. In *Proceedings of the IEEE International Conference on Systems Engineering*, Dayton, OH, pp. 277–280.

Kanellopoulos, I. and Wilkinson, G., 1997, Strategies and best practice for neural network image classification. *International Journal of Remote Sensing*, **18**, pp. 711–725.

Kavzoglu, T. and Mather, P.M., 1999, Pruning artificial neural networks: an example using land cover classification of multi-sensor images. *International Journal of Remote Sensing*, **20**, pp. 2787–2803.

KOLMOGOROV, A.N., 1957, On the representational of continuous functions of many variables by superpositions of continuous functional of one variable and addition. *Doklady Akademii Nauk USSR*, **114**, pp. 953–956.

KURKOVA, V., 1992, Kolmogorov's theorem and multilayer neural networks. *Neural Networks*, **5**, pp. 501–506.

LE CUN, Y., DENKER, J.S. and SOLLA, S.A., 1990, *Optimal Brain Damage, in Advances in Neural Information Processing Systems 2*, D.S. Touretsky (Eds), pp. 598–605 (San Mateo, CA: Morgan Kaufmann).

LIU, Y., YAO, X. and HIGUCHI, T., 2000, Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, **4**, pp. 380–387.

MAS, J.F. and FLORES, J.J., 2008, The application of artificial neural networks to the analysis of remotely sensed data. *International Journal of Remote Sensing*, **29**, pp. 617–663.

PAOLA, J.D., 1994, Neural network classification of multispectral imagery. MSc thesis, University of Arizona, Tucson.

RIPLEY, B.D., 1993, Statistical aspects of neural networks. In *Networks and Chaos—Statistical and Probabilistic Aspects*, O.E. Barndorff-Nielsen, J.L. Jensen and W.S. Kendall (Eds), pp. 40–123 (London: Chapman & Hall).

RUMELHART, D.E., MCCLELLAND, J. and the PDP RESEARCH GROUP, 1986, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1—Foundations* (Cambridge, MA: MIT Press), p. 547.

SHUTTLE RADAR TOPOGRAPHY MISSION (SRTM), 2000, Available online at: http://www2.jpl.nasa.gov/srtm/ (accessed 2 May 2008).

SIEDLECKI, W. and SKLANSKY, J., 1989, A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, **10**, pp. 335–347.

STATHAKIS, D., TOPOUZELIS, K. and KARATHANASSI, V., 2006, Large-scale feature selection using evolved neural networks. In *Proceedings of 13th SPIE Symposium on Remote Sensing*, The International Society for Optical Engineering, **6365**, 11–14 September, Stockholm, Sweden.

STAHAKIS, D. and VASILAKOS, A., 2006, Comparison of several computational intelligence based classification techniques for remotely sensed optical image classification. *IEEE Transactions in Geoscience and Remote Sensing*, August, **44**, pp. 2305–2318.

WANG, F., 1994, The use of artificial neural networks in a geographical information system for agricultural land-suitability assessment. *Environment and Planning A*, **26**, pp. 265–284.

WERBOS, P., 1974, Beyond regression: new tools for prediction and analysis in the behavioral sciences. PhD thesis, Harvard University.

YAO, X., 1993, Evolutionary artificial neural networks. *International Journal of Neural Systems*, **4**, pp. 203–222.