Superlearning Capabilities of Neural Networks?

Robert P.W. Duin
Pattern Recognition Group
Faculty of Applied Physics
Delft University of Technology
The Netherlands

Abstract

The literature on neural networks shows some spectacular examples in which large networks are trained by small sample sets. It is discussed how such results relate to the insights on the complexity of pattern recognition systems. The capacity as introduced by Vapnik [2] is considered but found to be insufficient for explanation. A proposal is made for the definition of an actual capacity that gives a more clear understanding of the observed phenomena. Based on these concepts situations of superlearning are defined and explained. Finally it is discussed how networks in superlearning situations might be trained such that generalization is to be expected.

1. Introduction

The application of artificial neural networks for pattern classification shows a number of surprising characteristics, at least from the traditional pattern recognition point of view. Huge nonlinear classifiers, in term of numbers of adjustable parameters, are trained by sometimes very small sets of examples. In many real world applications successes are reported where one might wonder why traditional pattern recognition methods have not been tried earlier. Obviously such studies are often done by people not familiar with the field of pattern recognition. On the other hand, from inside this field the interest in the neural network wave arose relatively late. Clearly a number of prejudices had to be removed.

The purpose of this paper is to discuss as clearly as possible some of the aspects of neural network training and to explain the behavior for a readership from the pattern recognition society. Some existing concepts will be clarified and some new ones will be introduced in order to be able to explain what might be going on. It is not the purpose of this paper to present new scientific experiments and results. We will focus on the understanding of old ones.

A typical example is the NETtalk experiment as reported by Sejnowski and Rosenberg [8]. One of their experiments concerns a network with more than 25000 adjustable parameters, trained by about 5000 examples distributed over 26 classes. These are large numbers. The striking point, however, is that the number of parameters is a multiple of the number of examples. One would expect, and we will explain that further, that the 25000 parameters can be given such values that all 5000 training examples are classified correctly, resulting in an *apparent error* (resubstitution error) of zero: $\varepsilon_A = 0$. As this is to be expected regardless of the distribution of classes, there is no generalization to be expected, resulting in a high classification error (more than 50% because of the 26 classes) on a test set: $\varepsilon > 0.5$. Both expectations are disproved experimentally. Sejnowski and Rosenberg find $\varepsilon_A \approx 0.05$ and $\varepsilon \approx 0.10$.

In the literature on neural networks many such examples can be found. See also [5], [9] and [10], discussed below. It is not the intention of this paper to give an extensive review.

In section 2 it will be discussed what generalization capabilities might be expected from trained classifiers in relation with the classifier *capacity* as introduced by Vapnik [2]. This concept will be modified as *actual capacity* in section 3, giving us the possibility to explain the concept of *superlearning*. Some artificial examples and some experiments from literature are presented in section 4. In section 5 the way neural networks are trained is related to the surprising results. Finally some general conclusions are presented.

2. On the expectation of generalization

Suppose a training set of *m* training objects is given and a classifier is found that classifies α φραχ–τιον ε_A of them incorrectly. What is the generalization of this result?, i.e. what can be expected for the error ε of the application of this classifier to new objects? Before considering the answers that have been given to this question we will first try to formulate why there should be a generalization at all.

Why is it that a classifier that classifies a set A of *m* objects correctly or almost correctly, is expected to show a similar behavior to another set B? A good reason may be that both sets are selected in a similar way (in statistical terms: randomly drawn from the same universe) and that thereby a generalization of A is valid for their common universe and consequently also for B. However, this is not sufficient as not each classifier is a generalization of its training set. Only under particular conditions this holds. In the generalization process the specific peculiarities of A should be deleted. A common way to do this is to limit the flexibility of the classifier. The less alternatives that are investigated for a classifier, the better. E.g., if the entire set of continuous functions is inspected for the classification of A, one can be sure that the resulting classifier is adapted to details in A that do not hold for B. In this context it is not a reassuring property of neural networks that they are universal approximators: if enough hidden units are provided, they can approximate almost any function, see Hornik [14] and Funahashi [15].

So, for generalization the training set A should be representative for B and the set of possible classifiers should be small. However, one likes to know: how small, and moreover, how does this relate to the size *m* of the training set A? A first answer has been given in 1965 by Cover [12] for a set of linear and polynomial classifiers: If *k* is the number of parameters (features in case of linear classifiers), then if $k = m$ any randomly labeled set of objects can be classified correctly ($\varepsilon_A = 0$). If $k = m/2$ there is a probability of 50% that a randomly labeled set of *m* objects can be classified entirely correctly. As the labeling is random an error of zero has no meaning for generalization. Cover chose the 50% level for his bound $k<m/2$. Later Foley [13] published Monte Carlo results by which for a desired level of significance bounds like $k < m/5$ or $k < m/10$ can be chosen.

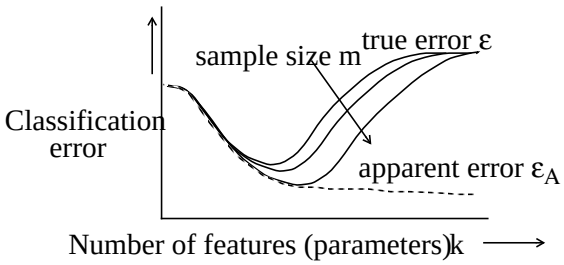

Fig. 1. The peaking phenomenon

The typical behavior of the classification error ε and the apparent error ε_A as a function of *k* is shown in fig. 1. This behavior is called “the peaking phenomenon”, see Jain and Chandrasekaran [11]. The deviation $\varepsilon - \varepsilon_A$ is a stochastic quantity depending of the realization of the training set. The entire figure depends on the application and can only be found for very large training sets and for simulated examples. For such an example the optimal value of *k* can be estimated. Raudys [18], [19] has published many of these simulation studies, resulting in some practical guidelines, see also Raudys and Jain [20]. It should be emphasized that these studies give insight in the expected behavior over repeated sets of training objects of classes that are at best similar to the application one is dealing with.

A worst case approach on the deviation $\varepsilon - \varepsilon_A$ has been made by Vapnik [2]. He gives probabilistic bounds for this deviation based on the concept of the capacity of the classifier. The *capacity* V_C (also called the Vapnik-Chervonenkis dimension) is the size of the largest training set for which all dichotomies can be realized by the classifier, i.e. for each labeling of this training set a classifier with $\varepsilon_A = 0$ can be found. The idea is that the actual learning set may be the worst one. For this set the probability of a given deviation is maximum. Vapnik gives bounds for such probabilities in terms of V_C and *m*. Devroye [21] used the same theory for investigating the consistency of a number of classifiers.

It is important to understand the difference between the capacity V_C and the number of free parameters *k* in a classifier. For linear and polynomial classifier they are directly related: $V_C = k + 1$. For these cases Vapnik’s probability bounds are a direct extension of the work of Cover [12]. For other classi-

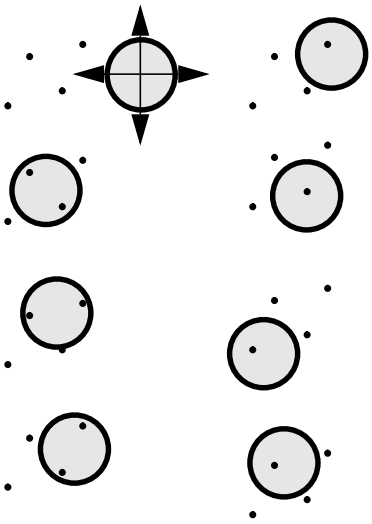

Fig. 2. A circular decision function with fixed radius can classify 4 points in R_2 in any way.

fiers, or if not all parameters are free, no such relations exist. For some simple examples see fig. 2 and 3. In fig. 2 a circular classifier in R_2 is shown for which just its center and *not* its radius can be adjusted. By adjusting these two parameters all dichotomies of the four given points can be reached. So $V_C = 4$, at least. For a linear classifier in R_2 with also two free parameters $V_C = 3$. Note that this holds for the given four points, and not for an arbitrary set of four points.

In fig. 3 the classifiers in R_1, $S_1(x) = w_1 x + w_0$ and $S_2(x) = \sin(\omega x)$ are compared. $S_1(x)$ can separate two points at most, so $V_C = 2$. However, for almost

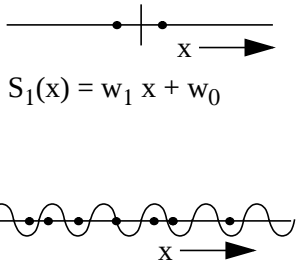

Fig. 3. A linear classifier in R_2 has a capacity of 2. A nonlinear classifier may have an unbounded capacity.

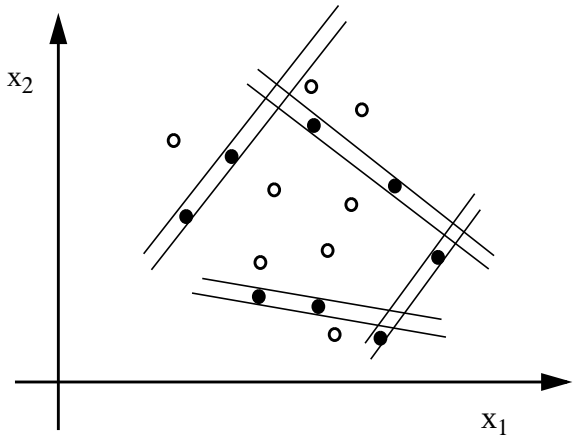

Fig. 4. Almost any set of m points in R_κ can be arbitrarily classified by a network with m/k hidden units and m +m/k weights

any finite set of points and for each desired labeling a value of ω can be found such that $S_2(x)$ separates them, so for this classifier with just one parameter holds $V_C \to \infty$.

It is now of interest what the capacity is for a neural network. Baum and Haussler [3] give the upper bound $V_C \leq 2W \log(eN)$ for a neural network with W weights and N units. Whether this bound can be reached and for what set of points is not clear. In fig. 4 is shown how m points in R_k can be classified arbitrarily by a neural network with m/k hidden units and $m + m/k$ weights, providing a lower bound for V_C. In this network the first layer of m/k units isolates one of the classes by pairs of linear functions. The second layer provides the final assignment. See also Baum [1]. For a multiclass situation $V_C > W/2$. So, such constructions show that for classifying m objects correctly ($\varepsilon_A = 0$), a feedforward network with about $W = 2m$ weights exists that can do it. The remarkable thing is now that this is not found in many applications with large networks, e.g. [5] and [8]. The training rule is apparently not able to find such solutions. Reasons can be: finite training time, parameter settings, etc. This defect of the training rule is very profitable, as zero-error solutions as given in fig. 4 have no generalization power whatsoever.

So we have to conclude that as a consequence of the fact that the training rule (usually backpropagation) does not do what it is supposed to do: minimiz-

ing the error, a good generalization is obtained. A better training rule, in the sense that a better minimization is reached, may thereby be counterproductive: a larger generalization error is reached.

This result can be explained by introducing a *training rule related capacity* V_R that is smaller than the classifier capacity V_C. If now the following holds:

$$V_R < m < V_C$$

then the m training objects are used to train a classifier with such a large capacity that no generalization has to be expected, but as it is trained by a rule for which the capacity is small enough, generalization can still be reached. An example of this situation is the estimation of a linear classifier for m objects in R_k while $m < k$. Here Fisher's linear discriminant can not be used as the scatter matrix will be singular. However, the more simple nearest mean method may still yield a reliable classifier.

Focussing or restricting the training rule to a small set of solutions can also be described in terms of regularization, a technique for solving underdetermined sets of equations, e.g. see Sjöberg and Ljung [6] and Moody [7]. Here expressions are derived for the expectation of the deviation $\varepsilon - \varepsilon_A$ like:

$$E\{\varepsilon(\lambda) - \varepsilon_A(\lambda)\} = 2\,\sigma^2_{eff}\,p_{eff}(\lambda)/m$$

as presented by Moody [7]. λ is an adjustable regularization parameter, $p_{eff}(\lambda)$ is the effective number of parameters as result of the regularization and σ^2_{eff} is the effective output noise of the classifier. Three points are important to notice:

1. Here ε and ε_A are mean square errors and not probabilities of error!
2. It has already been illustrated above that the number of parameters is not a good measure for the capacity of a classifier. This does not seem to hold for the topic of function approximation as studied by Moody.
3. The expression tells something of the deviation, the difference between the true error ε and the observed error ε_A. By measuring ε_A and estimating the right hand term, an *expected value* for $\varepsilon(\lambda)$ is found. Vapnik's theory presents a probabilistic upperbound for this error.

3. The actual capacity of a classifier and superlearning

The above mentioned theory on classifier capacity is based on a possible worst case situation: the data configuration for which the largest number of dichotomies can be realized by the classifier. However, in a practical situation a dataset is given and in theory it can be verified whether the data is in the worst case or not. If not, a smaller number of dichotomies are possible, resulting into a smaller *actual capacity* V_A. So we have:

$$V_A \le V_R \le V_C$$

This suggests that generalizable results can be obtained for training sets that are smaller than the training rule capacity:

$$V_A \le m \le V_R \le V_C$$

In such a situation the special data configuration makes it possible to get a generalizable classifier by a training rule from which it cannot be expected as $m \le V_R$. This will be called *superlearning*. Such a special data configuration might be caused by a high correlation in the data. Also nonlinear dependencies are of importance as the neural network is a nonlinear classifier. If the data is located in a linear or nonlinear subspace of the feature space large numbers of possible classifiers will not be distinguishable anymore as the intrinsic dimensionality of the data does not allow it. The following example may elucidate this. If we are looking for a linear classifier in R_{100} based on 50 training objects no generalization can be expected. However, if these 50 objects are located in a 5-dimensional subspace of R_{100}, generalization may very well be possible.

4. Examples

We will now discuss a few examples that illustrate the presented ideas about the actual capacity and the superlearning phenomenon.

4.1 Example 1

Fig. 5 shows two normally distributed classes in R_2. For each class just 5 training objects are given. Both classes have a very large variance in one direction. Methods like the nearest mean and the nearest neighbor will fail. If R_2 is rotated into a higher dimensional space, say R_{20} and becomes a subspace of it, also parametric methods based on the estimation of

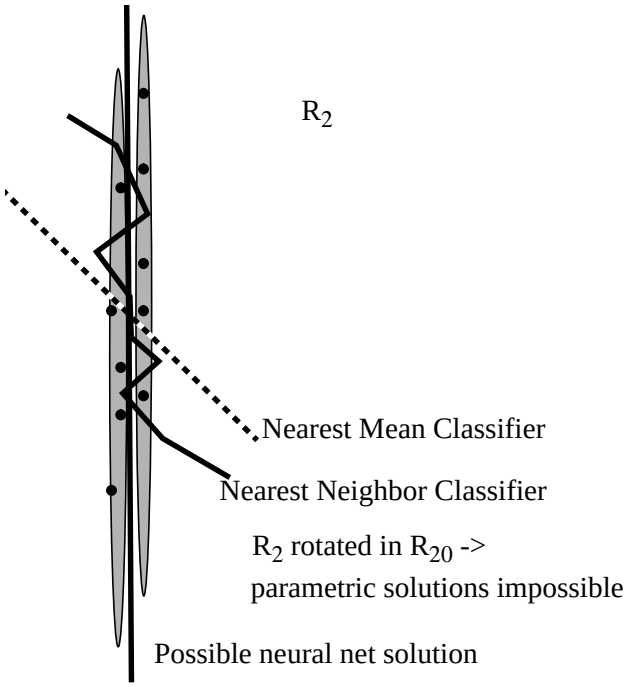

Fig. 5. 5+5 points in a 2-dimensional subspace may be sufficient to construct a generalizable neural net classifier.

normal distributions will fail. However, an attempt to optimize the separability between the classes by an error minimizing classifier like a neural network, can yield a generalizable result in spite of the large capacity of the classifier ($V_C = V_R = 20$, $m = 10$). Such a solution may also be found by a principal component analysis, followed by Fisher's linear discriminant. The neural network may combine this in a single approach.

4.2 Example 2

Suppose f(x) is some arbitrary function on R_1, say some frequency spectrum. Let the samples x_1, x_2,... x_k be the feature input to a classifier that has to distinguish between two types of curves for f(x). Suppose that the intrinsic variations between the function are just caused by shifts and amplitude variations: $\forall$i: $f_i(x) = a\, f_0(x\text{-}b)$. This implies that all feature vectors $\boldsymbol{x}$, independent of k, are located on some, possibly non-linear, 2-dimensional subspace of R_k. Independent of k, about 10 training funtions($m = 10$) may now already give a generalizable result for a linear classifier as in spite of $V_C = k$ we find for the actual capacity $V_A = 2$. Here we see that increasing k by adding noise free new samples of the functions does not necessarily result in a peaking result, in agreement with the intuitive notion that a better functional description is obtained. This is a possible explanation of the results of a paper by Ciftcioglu et al. [10] in which the authors report to find reliable results with a feed forward network with 512 weights trained by just 6 examples!

4.3 Example 3

In the ICPR11 conference, 1992, Guyon, Vapnik and others [4] presented an interesting paper in which they study the *effective capacity*, which seems to combine our training rule capacity V_R and actual capacity V_A . They show experimentally that a decreasing effective capacity first increases the performance of the classifier (increased generalizability caused by a smaller set of inspected classifiers) and finally decreases the performance (the good classifiers are not represented anymore in the set of inspected classifiers).

This paper is for another reason also of interest. The application is character recognition and the features are just pixel values. In the preprocessing the characters are heavily blurred. Blurring is a linear operation in the feature space. It is equivalent to a linear operation on the weights in the first layer of a network. So any network classifier that is found after blurring the input characters, is equivalent to a network of the same size for non-blurred characters. Why is it that using pixel values of the original characters are worse features than pixel values of blurred characters as the resulting feature spaces differ only by a linear operation? A possible answer in our terminology is that the blurring operation transforms the data such that the actual capacity of the classifier *decreases*. The blurring operation probably results in a feature space in which the distances between objects better represent the resemblance between the original characters.

4.4 Example 4

Kamata et al. [9] have studied a remote sensing application using 7 bands. In windows varying from 1 x 1 to 5 x 5 pixels recognition tasks are performed with 5 classes. So the dimensionality of the feature space ranges from 7 x 1 x 1 = 7 to 7 x 5 x 5 = 175. A network with one hidden layer with 10 units is used. So the number of weights ranges from 80 to 1800. The authors use 5 x 10 = 50 training examples and 5 x 30 = 150 test examples. The network is al-

.

Table 1: Review of some results by Kamata et al. [9]

window size	network size	no. weights	no. iterations	test result ε
2 x 2	28 x 10 x 5	330	1720	0.033
3 x 3	63 x 10 x 5	680	240	0.000
4 x 4	112 x 10 x 5	1170	170	0.047
5 x 5	175 x 10 x 5	1800	170	0.053

ways trained until all training objects are classified correctly ($\varepsilon_A = 0.0$). This convergency cannot be reached for the 1 x 1 window, which is thereby not tested. In table 1 the test results for the other window sizes are listed. The test results in the right column show the typical peaking characteristic. First an increase of performance, then, for higher dimensionalities and more complex networks a deterioration. The remarkable point in this table is the zero error for the 3 x 3 window size. Here the authors train in a 63 dimensional feature space a network with 680 weights and yield a perfect generalizable result. The network is trained until a zero error for the training set is found. This implies an almost unbounded training effort. The generalizable result for the 3 x 3 window is thereby obtained by a very low value of the actual capacity V_A, probably caused by the high correlation between the bands and between the pixels in a window. This also explains the not perfect, but very good results for the larger windows. A larger window increases the dimensionality, but hardly the intrinsic dimensionality if the pixel noise is small.

5. Neural network training

Many user adjustable parameters influence the result of training a neural network: the number of features, data normalization, the number of hidden units, input and output coding, step size, momentum term, batch size, stopping criterion, number of repeated initializations, value of the initial weights, etcetera. They all influence the training rule capacity and thereby they also possibly influence the actual capacity as they increase or decrease the set of possible classifiers on the dataset. By each of these parameters the good classifiers may be deleted from this set and cannot be found thereby. On the other side, by each of these parameters the set of possible classifiers can be made that large that the good classifiers cannot be selected by the small training set ($m < V_A$). “Training a neural network can become a user’s nightmare”, Weiss and Kulikowski [16].

In almost any application a number of initial experiments have to be done in order to find the right settings of the above parameters. They are sometimes mentioned in the paper, but almost never reported in detail. However, they influence the deviation $\varepsilon - \varepsilon_A$. A second problem in neural network training is that it is very time-consuming. A third point to notice is that each neural network training result has a random component due to the set of random initializations. These three aspects make it almost impossible to set up the correct leave-one-out or bootstrap procedures as developed in the statistical error estimation literature for correcting the above deviation and using the complete set of training objects exhaustively, see Hand [17]. An advisable procedure might therefore be the following:

1. Hold out a set of test objects from the training set that is used for the initial experiments.

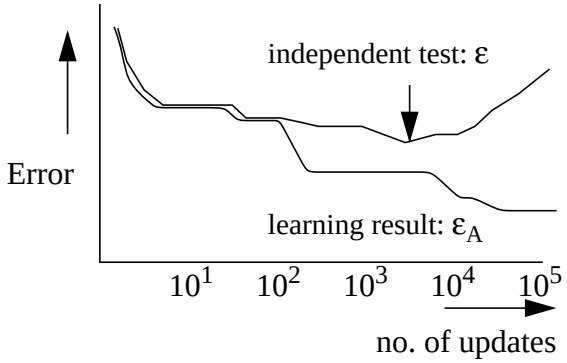

Fig. 6. Training using an independent test set

2. Train the network a number of times (e.g. 10) with different sets of initial weights and use one of the test sets for a stopping rule, see fig. 6. This prevents the network from overtraining by reducing the number of classifiers to be inspected.
3. The best network according to the first test set gives for this test set again a positively biased result. This bias can be estimated by Hoeffding's inequality, see Vapnik [2].

In most traditional pattern recognition methods one would afterwards combine the testset and trainingset for obtaining a more reliable result. This yields in neural network training the problem that without a testset it is hard to find best training result. Thereby there is no guarantee that this extended training set produces a better classifying network.

6. Conclusions and discussion

The classifier capacity as introduced by Vapnik is a good general framework for understanding the selection of generalizable classifiers. For linear and polynomial classifiers it is directly related to studies based on the number of parameters. However, as it is based on a worst case situation, it is far too pessimistic for practical use. By introducing the training rule capacity and the actual capacity it is made understandable how generalizable results can be obtained in applications with a far too small set of training objects in relation with the classifier capacity.

Generalizable training results with a sample size under the training rule capacity, called superlearning, can be understood from strong dependencies in the data, resulting in a low intrinsic dimensionality. By some examples the existence of such data is demonstrated.

Following these attempts to get some insight, some suggestions are given how to train a neural network in practice such that generalizable results are obtained.

In retrospect one might argue that the main conclusion on the cause of superlearning is obvious: if the data is such that the classes are well separated, then any classifier will do, very simple ones and also very complex ones. This also may hold if the training set is very small: e.g., two widely separated classes can be distinguished by almost any classifier using two objects in the training set. So superlearning should exist for all very simple problems The word superlearning itself becomes questionable for such situations.

However, these very simple problems are not the ones of interest here. In practice they even will hardly arise as they will already be solved before a pattern recognition system is considered at all. The problems discussed here, are those that cannot be solved by a simple classifier, e.g. a linear one, or other classifiers with a capacity in the order of the feature size ($V_C = O(k)$), but the problems for which a complex classifier is needed, but that can still be trained by a surprisingly small training set. That is the superlearning phenomenon.

7. Acknowledgment

The author thanks Martin Kraaijveld and Wouter Schmidt, both of Delft University of Technology, and Prof. S. Raudys of the Institute of Mathematics and Informatics in Vilnius, Lithuania, for stimulating discussions on the topic. Arnold Smeulders of the University of Amsterdam is gratefully acknowledged for his stimulating observations.

This work was sponsored by the Dutch Government as a part of the SPIN-FLAIR-DIAC project, and by the Foundation of Computer Science in the Netherlands (SION) with financial support from the Dutch Organization for Scientific Research.

8. References

[1] E.B. Baum, "On the capabilities of multilayer perceptrons," Journal of Complexity, vol. 4, pp. 193 - 215, 1988.

[2] V. Vapnik, Estimation of Dependences based on Empirical Data, Springer, New York, 1982

[3] E.B. Baum and D. Haussler, "What size nets gives valid generalization?," Neural Computation, vol. 1, pp. 151-160, 1989.

[4] I. Guyon, V. Vapnik, B. Boser, L. Bottou, and S.A. Solla, "Capacity Control in Linear Classifiers for Pattern Recognition," in: Proceedings 11th IAPR International Conference on Pattern Recognition, Volume II, Conference B: Pattern Recognition Methodology and Systems (ICPR11, The Hague, The Netherlands, August 30 - September 3, 1992), pp. 385 - 388, IEEE Computer Society Press, Los Alamitos, California, 1992

[5] R. Sabourin, J-P. Drouhard, “Off-line signature verification using directional PDF and neural networks” in: Proceedings 11th IAPR International Conference on Pattern Recognition, Volume II, Conference B: Pattern Recognition Methodology and Systems (ICPR11, The Hague, The Netherlands, August 30 - September 3, 1992), pp. 321 - 325, IEEE Computer Society Press, Los Alamitos, California, 1992

[6] J. Sjöberg and L. Ljung, “Overtraining, regularization, and searching for minimum in neural networks,” Neuroprose archive, February 1992

[7] J.E. Moody, “The effective number of parameters: an analysis of generalization and regularization in nonlinear systems,” in: Advances in neural information processing systems 4, ed. J.E. Moody, S.J. Hanson, R.P. Lippman, Morgan Kaufmann Publishers, San Mateo, CA, 1992

[8] T.J. Sejnowski and C.R. Rosenberg, NETtalk: a parallel network that learns to read aloud, The John Hopkins University Electrical Engineering and Comp. Science, 1986.

[9] S.-I. Kamata, R.O. Eason, A. Perez, and E. Kawaguchi, “A Neural Network Classifier for LANDSAT Image Data,” in: Proceedings 11th IAPR International Conference on Pattern Recognition, Volume II, Conference B: Pattern Recognition Methodology and Systems (ICPR11, The Hague, The Netherlands, August 30 - September 3, 1992), pp. 573 - 576, IEEE Computer Society Press, Los Alamitos, California, 1992

[10] O. Ciftcioglu, E. Turcan, and S. Seker, “Failure detection studies by layered neural networks,” Int. AMSE Conf. Neural Networks, San Diego, U.S.A., 29-31 May 1991.

[11] A.K. Jain and B. Chandrasekaran, “Dimensionality and Sample Size Considerations in Pattern Recognition Practice,” in: Handbook of Statistics, vol. 2, ed. P.R. Krishnaiah and L.N. Kanal, pp. 835 - 855, North-Holland, Amsterdam, 1987

[12] Cover, T.M., “Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition,” IEEE Trans.Elec.Comp, vol. EC-14, pp. 326-334, 1965.

[13] D.H. Foley, “Considerations of sample and feature size,” IEEE Trans. on Information Theory, vol. IT-18, no. 5, pp. 618-626, September 1972.

[14] K. Hornik, M. Stinchcombe, and H. White, “Multi-layer Feedforward networks are Universal Approximators,” Neural Networks, vol. 2, pp. 359-366, 1989.

[15] K. Funahashi, “On the Approximate Realization of Continuous Mappings by Neural Networks,” Neural Networks, vol. 2, pp. 183-192, 1989.

[16] S.M. Weiss and C.A. Kulikowski, Computer Systems that Learn, Morgan Kaufmann, San Mateo, California, 1991.

[17] D.J. Hand, Recent advances in error rate estimation, Pattern Recognition Letters, vol. 4, 1986, 335-346.

[18] S. Raudys and V. Pikelis , “Collective selection of the best version of a pattern recognition system,” Pattern Recognition Letters, vol. 1, no. 1, pp. 7 - 14, 1982.

[19] S. Raudys, “On dimensionality, Learning, sample size and complexity of classification algorithms,” in: Proc. of the 3rd Int. Conf. on Pattern Recognition, pp. 166-169, Coronado, California, November 1976.

[20] S.J. Raudys and A.K. Jain, “Small sample size effects in statistical pattern recognition: recommendations for practitioners,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 3, pp. 252-264, 1991.

[21] L. Devroye, “Automatic pattern recognition: a study of the probability of error,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-10, no. 4, pp. 530 - 543, July 1988.