The Effective Capacity of Multilayer Feedforward Network Classifiers

Martin A. Kraaijveld[1,2] and Robert P.W. Duin[1]

Pattern Recognition Group, Faculty of Applied Physics, Delft University of Technology,
P.O. Box 5046, 2600 GA Delft, The Netherlands, e-mail: ks115@ksepl.nl, duin@ph.tn.tudelft.nl

Abstract

Theoretical results on the capacity, or Vapnik-Chervonenkis dimension, of a multilayer feedforward (neural) network classifier leads to much larger training sets than is used in many applications. In this paper it is shown that the effective capacity, that takes into account the training rule, is much smaller than the upper bounds on the capacity that are derived from these theoretical considerations. The success of many network applications can thereby be understood from the restricted possibilities of the optimization technique that is used for training the network.

1. Introduction

Nowadays feed forward network classifiers are a very popular tool for solving pattern classification problems. From the traditional pattern recognition point of view they yield sometimes remarkable results. They are complex in the sense that many different classification functions can be implemented. As a consequence large sets of objects can be arbitrarily labeled by the classifier. The capacity, or Vapnik-Chervonenkis dimension, (Vapnik [18], Devroye [4], Baum and Haussler [3], Kraaijveld [13]) is a complexity measure that is based on this observation: *it is the size V of the largest set of objects that can be arbitrarily classified*, i.e. for each labeling (2^V for a 2-class problem) there exist a network realization (a set of weights) that perfectly classifies it. As a consequence no generalization is to be expected before the size of the training set $m >> V$. The Vapnik-Chervonenkis theory yields upper bounds for the necessary sample size [18]. Baum [2] has shown that for 2-layer networks with h hidden units and a d-dimensional input space holds that:

$$V \geq 2 \lfloor h / 2 \rfloor d \qquad (1)$$

which is roughly equivalent to

$$V \geq W - 2h \qquad (2)$$

with W the number of free parameters (weights) in the network. As an upper bound for the network capacity Baum and Haussler [3] found:

$$V \leq 2 W \log_2 (e (h + 1)) \qquad (3)$$

Here e is the base of the natural logarithm. So $m >> W$ is necessary in order to be sure that the network generalizes and is not biased by the training set. However, many successful applications have been published with $m << W$, see table 1. Several explanations for this can be raised, see Duin [5]. A major point is that the most popular training rule for feed forward networks, the back propagation rule does most certainly not perform an exhaustive search over the parameter space. In fact only a small subset of weight combinations is investigated. One might say that the effective capacity V^* of a network classifier is much smaller than the network capacity as a consequence of the training rule: $V^* << V$. Of course, V^* is a function of the parameters determining the rule, e.g. weight initialization, scaling, step-size η, momentum term α, etc.

A second reason for the large difference between the theoretically needed sample sizes and the successfully used ones is that the Vapnik-Chervonenkis theory is based on a worst case approach in the sense that the bounds hold for the worst possible training set. As a consequence one may define an actual capacity V^*_A that

[1] Supported 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 (NWO).

[2] Now with SHELL Research, P.O. Box 60, 2280 AB Rijswijk, The Netherlands.

The authors thank their colleague Wouter Schmidt for many useful discussions.

holds for the given training set as well as for the training rule:

$$V^*_A << V^* << V \qquad (4)$$

The actual capacity is of importance for a particular application, the effective capacity is a more general number as it characterizes the training rule. This difference is not made by Guyon [9], in which a slightly different definition of effective capacity is used. In this paper we will experimentally work out the difference between the effective capacity V^* and V using an artificial problem of two Gaussian distributions. It will be shown that V^* can be very small. We think that this is an important explanation for understanding the good results of table 1. A network is, after training, not at all as complex as its number of free adjustable weights suggests.

Table 1: Some experiments in the neural networks literature in which the sample size m is considerably smaller than the number of free parameters W. The performance on an independent test set is shown in the last column.

Ref.	Appl.	W	m	Perf.
[16]	Mapping text to phonemes	> 20,000	≈ 5,000	> 95%
[6]	Discriminate sex of human faces	> 36,000	90	91.9% (better than humans)
[10], [17]	Back-gammon	> 11,000	3000	world-champion computer program
[14]	Control of an autono-mous vehicle	> 36,000	1200	> 90% (better than any other algorithm)
[7], [8]	Sonar target recognition	1,105	192	84.7%, (nearest neighbor class. 82%)
[15]	OCR	> 10,900	5000	94.5%

2. The Estimated Effective Capacity of Multi-Layer Networks.

In this section a number of experiments are described in which we estimate the effective capacity of a multi-layer network that is trained with the backpropagation algorithm. As the non-linear dynamics of the backpropagation algorithm may be sensitive to particular choices for the parameters of the algorithm, it is investigated how the effective capacity is influenced by such factors as the learning rate, the scaling of the training data, etc.

2.1 The Experimental Procedure

Conceptually, the approach is very simple. The procedure is started with an estimate for the effective capacity V^* that is equal to one. Then, a subset of size V^* is randomly selected from the dataset. If the learning procedure is capable of inducing all 2^{V^*} dichotomies on this subset, the effective capacity is obviously at least equal to V^*. In that case, the estimate of the effective capacity V^* is increased by one and the procedure is repeated. However, if the learning procedure is not capable of inducing all 2^{V^*} dichotomies on the subset, another subset of size V^* is randomly selected from the dataset. If the learning procedure is not capable of inducing the 2^{V^*} dichotomies on 1,000 subsets of size V^*, it is concluded that the effective capacity is equal to $V^* - 1$.

Despite the conceptual simplicity of the method, it is clear that the approach results in considerable computational requirements. If at some moment in the procedure V^* is equal to 10, $2^{10} = 1024$ dichotomies have to be investigated. Therefore, the method quickly becomes impractical for larger values of V^*. Also, the choice to investigate at most 1,000 subsets of size V^*, is rather arbitrary, and primarily motivated by computational considerations. Note, however, that the repeated selection of subsets is only necessary to cope with the problem that an unfortunate subset of points could (almost) be hidden in some subspace. Such a set of points could only be labeled arbitrarily with great difficulty. It is reasonable to assume, therefore, that 1,000 different subsets is sufficient to yield at least some subsets for which this is not the case, unless the total dataset is indeed hidden in a subspace.

2.2 The Training Data and the Learning Procedure.

In the experiments that are described in the rest of this section, the influence of various parameters of the backpropagation algorithm on the effective capacity is investigated. However, as there many of such parameters, it is not possible to explore the complete space of parameter settings. Instead, we have defined one "*standard experiment*" and then in subsequent experiments the effect of changing only one parameter

was investigated. The settings of this standard experiment were taken as follows:

• A *standard* dataset of 200 points was generated from bivariate normally distributed data. The dataset was saved to disk, such that in all experiments exactly the same dataset could be used.

• As the *standard* network of the experiments, a network with two inputs, 100 hidden units in one hidden layer, and one output unit was generated by assigning small random values to the parameters of the network. The initial values of the parameters were generated according to a uniform distribution in the range [-0.01, 0.01]. All units of the network were provided with the standard inner product activation functions and sigmoidal output functions. The network was also saved, such that all experiments could be performed with the same initial parameters.

• During the training procedure, the parameters of the network were updated for at most 100,000 times with the backpropagation algorithm. The learning rate η was chosen as 0.1 and no momentum-term was applied ($\alpha = 0$). The simulations were performed in a custom-made environment that was written in C [12].

2.3 Experimental Verification for Linear Classifiers

To verify the procedure and to gain some insight in its characteristics, some experiments were performed in which the capacity was exactly known. As for linear classifiers the capacity is equal to $d + 1$ a number of experiments were performed in which the capacity was estimated with the experimental procedure described above. As the value $d + 1$ was also found in these experiments, the experimental setup and the environment was proven to be correct.

2.4 Experimental Results

In the following tables is shown how the effective capacity behaves as a function of: the number of hidden units (table 2), the initialization of the network parameters (table 3), the learning rate of the backpropagation algorithm (table 4), the learning time (table 5), the scaling of the training data (table 6), and the dimensionality of the feature space (table 7). The result of the "standard experiment" is shown in the shaded cell in each table.

3. Discussion

A striking aspect of the experimental results of the previous section, is that the estimated effective capacity V^* of a multi-layer feedforward network classifier, that is trained with the backpropagation algorithm, is so low. For all the variants of the standard experiment, the training procedure has not been able to find some set of 8 points that can be labeled in an arbitrary way, despite the 401 free parameters of the network, a guaranteed lower bound on the capacity of 200, and a huge investment of roughly 10^{15} CPU instructions. Apparently, the application of the backpropagation rule, instead of performing an exhaustive search, reduces the maximum number of points that a classifier can label in an arbitrary way with orders of magnitude. This finding confirms that the backpropagation algorithm is simply not capable of exploiting all the free parameters of the network. A closer look at the results reveals the following aspects:

Table 2: The estimated effective capacity V^* as a function of the number of hidden units h, estimated on the standard dataset. The result of the "standard experiment" is shown in the shaded cell.

nr. of hidden units h	1	2	5	10	20	50	100
nr. of free param. W	5	9	21	41	81	201	401
lower bound on V, cf. (1)	3	4	8	20	40	100	200
upper bound on V, cf. (3)	24	54	169	401	945	2860	6496
$\hat{V}^*$	3	5	6	7	7	6	5

Table 3: The estimated effective capacity V^* for various realizations of the initial parameter values, and as a function of the statistics that generate the initial parameter values.

domain of uniform. distr.	realization				
for the initial parameters	1	2	3	4	5
$W \in$ [-0.01, 0.01]	5	6	6	6	6
$W \in$ [-1.0, 1.0]	6	6	7	7	7
$W \in$ [-100.0, 100.0]	1	2	2	1	2

Table 4: The estimated effective capacity V^* as a function of the learning rate η.

η	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	10
$\hat{V}^*$	1	1	2	3	4	5	5	1

Table 5: V^* as a function of the number of parameter updates.

# of upd.	1	10	10^2	10^3	10^4	10^5	10^6	10^7
$\hat{V}^*$	1	1	1	3	4	5	6	6

Table 6: The estimated effective capacity V^* as a function of the scale of the training data.

scaling factor	0.01	0.03	0.1	0.3	1	3	10	30	100
$\hat{V}^*$	2	3	3	4	5	7	7	7	7

Table 7: The estimated effective capacity V^* as a function of the dimensionality of the feature space d.

d	W	lower bound on V cf. (1)	upper bound on V cf. (3)	$\hat{V}^*$
1	301	100	4876	4
2	401	200	6496	5
5	701	500	11357	7
10	1201	1000	19458	11
20	2201	2000	35660	15

• Table 2 presents the consequences of changing the number of hidden units on the estimated effective capacity. It is clear that the number of hidden units, or the number of free parameters has a negligible influence on the effective capacity of the network. Changing the number of hidden units from 2 to 100 made the estimated effective capacity vary between 5 and 7. A remarkable fact is that the estimated effective capacity seems to have a maximum for 10 and 20 hidden units in table 2. An explanation for this effect, is that an increase of the number of hidden units generally slows down the learning process. This is caused by the fact that a large number of hidden units results in a high correlation of their outputs, as the number of inputs is constant. It can be shown that, in the beginning of the learning phase, such a high correlation results in a decrease of the learning speed of the output unit [1].

• Further influence of the total number of parameter updates is found in table 5. When the number of parameter updates is chosen too small, the learning procedure is not capable of reaching a desired location in parameter space. This causes the effective capacity of the system to be very small. Of course, this is not a property of multi-layer networks classifiers only, as it also holds for linear classifiers. However, the table shows that an increase of the number of parameter updates beyond 1,000,000 has no influence on the estimated effective capacity.

A related parameter of the backpropagation algorithm is the learning rate η. If the learning rate is too small, the learning procedure may be stopped before a suitable (local) minimum or saddle point has been reached. Making the learning rate too large, however, will result in a non-stable, or even chaotic, behavior of the learning procedure. This also prevents the learning procedure of reaching a certain desired location. As table 4 reveals, the optimum is found somewhere in-between: the maximum estimated effective capacity is found for a learning rate of 0.1 or 1.0.

• It is well-known that the behavior of backpropagation critically depends on the initial choice of the parameter vector, Kolen [11]. The experiments in table 3, however, show that as long as the initial parameters are chosen sufficiently small, the estimated effective capacity is not heavily influenced. Only when the initial parameters are chosen much too large, the dynamic behavior of the learning procedure is affected. This is caused by the fact that the large initial parameter values cause the sigmoidal non-linearities of the network to saturate. For properly chosen initial parameter values, however, the estimated effective capacity again varies between 5 and 7.

• A related parameter of the backpropagation algorithm is the scaling of the training data. A too small scale of the training data will make it difficult to label data in an arbitrary way, as a decision function has to be positioned accurately in a very small area. This is not always possible since the learning rate, i.e. the step-size of the adaptations of the parameters, is fixed. On the other hand, if the scale of the training data is too large, the sigmoidal non-linearities of the network will saturate, which would imply a small effective capacity. However, this is an effect that is not visible in table 6. This can be explained by the fact that the training data is of a stochastic nature; the data is spread out over some area and some of the points of the training set are on locations that have the optimal balance between separability and saturation of the sigmoids. By repeated selection of subsets from the training set, some of these subsets will have many points on these ideal locations. Therefore, the fact that the estimated capacity is constant for large scaling factors, is a property of the procedure to estimate the effective capacity.

• Finally, the last table 7 shows the effect of an increase of the dimensionality of the feature space. Again, it is clear that the estimated effective capacity is orders of magnitude smaller than the true capacity or the number of free parameters. However, it should be noted

that increasing the dimensionality of the feature space corresponds to other problems than the standard two-dimensional problem, and that it therefore might be necessary to reinvestigate some of the other parameter settings of the backpropagation algorithm.

As a conclusion from all the tables, it is clear that the effective capacity of a multi-layer network that is trained with the backpropagation algorithm depends on the various settings of the parameters of the algorithm. The experimental results also reveal that the settings that were initially chosen for the standard experiment were not too bad. In the standard experiment an effective capacity of 5 was estimated, and in many tables this was close to the maximum. In fact, the maximum estimated effective capacity that was found in all experiments with a two-dimensional feature space was equal to 7. As the experiments did not perform an exhaustive search of the parameter settings of the backpropagation algorithm, this maximum value of 7 is likely to be an underestimate of the true effective capacity. However, the fact that the value 7 is based on 40,000 attempts to use a 100 hidden unit network to label some subsets in an arbitrary way (tables 3 - 6), does provide some confidence in its value.

4. Conclusions

In this paper it was discussed how the specific properties of an iterative learning procedure influence the generalization behavior of a classifier. Some experiments were performed to estimate the effective capacity of a multi-layer feedforward network classifier that was trained with the backpropagation algorithm. The experiments show that the estimated effective capacity may be orders of magnitude smaller than the true capacity of a network. This proves that the searching capabilities of the backpropagation algorithm are very limited. An implication of this finding, however, is that the performance of such a network on a training set is much more significant than the number of free parameters would suggest.

5. References

[1] Bakker, R.R.N., Kraaijveld, M.A., Duin, R.P.W., and Schmidt, W.F., "On the Speed of Training Networks with Correlated Features" Proc. of the IEEE Conference on Neural Networks, (San Francisco, Mar. 28 - April 1, 1993).

[2] Baum, E.B., "On the Capabilities of Multi-layer Perceptrons", Journal of Complexity 4, pp. 193 - 215, 1988.

[3] Baum, E.B., and Haussler, D., "What Size Net Gives Valid Generalization?", Neural Computation 1, pp. 151-160, 1989.

[4] Devroye, L., "Automatic Pattern Recognition: A Study of the Probability of Error", IEEE Tr. on PAMI, Vol. 10, No. 4, July 1988.

[5] Duin, R.P.W., "Superlearning Capabilities of Neural Networks?," Proc. of the 8th Scandinavian Conf. on Image Analysis", Tromsø, Norway, May 1993.

[6] Golomb, B.A., Lawrence, D.T., and Sejnowski, "SEXNET: A Neural Network Identifies Sex from Human Faces", in: Neural Inform. Proc. Systems 3, Lippmann, Moody and Touretzky (eds.), Morgan Kaufmann Publishers, U.S.A., pp. 572 - 577, 1991.

[7] Gorman, R.P., and Sejnowski, T.J., "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets", Neural Networks, Vol. 1, No. 1, 1988.

[8] Gorman, R.P., and Sejnowski, T.J., "Learned Classification of Sonar Targets Using Massively Parallel Network", IEEE Tr. on ASSP, Vol. 36, No. 7, 1988, pp. 1135 - 1140.

[9] I. Guyon., V. Vapnik, B. Boser, L. Bottou, and S.A. Solla, Capacity Control in Linear Classifiers for Pattern Recognition, Proceedings 11th ICPR, II, The Hague, 1992, 385-388.

[10] Hertz, J., Krogh, A., and Palmer, R.G., Introduction to the Theory of Neural Computation, Addison Wesley, 1991.

[11] Kolen, J.F., and Pollack, J.B., "Back-Propagation is Sensitive to Initial Conditions", in: Neural Inform. Proc. Systems, Lippmann, Moody and Touretzky (eds.), Morgan Kaufmann, 1991, pp. 860 - 867.

[12] Kraaijveld, M.A., and Schmidt, W.F., "ANN-lib - Neural Networks Simulation Library", Pattern Recognition Group, Dpt. of Applied Physics, Delft University of Technology, Delft, The Netherlands, 29 pages, 1990 / 1991 / 1992 / 1993.

[13] Kraaijveld, M.A., Small Sample Size Behavior for Multi-Layer Feedforward Network Classifiers: Theoretical and Practical Aspects, Ph.D. Thesis, Delft University of Technology, Delft, 1993

[14] Pomerleau, D.A., "ALVINN: An Autonomous Land Vehicle in a Neural Network", in: Adv. in Neural Inform. Proc. Systems I, ed. D.S. Touretzky, San Mateo, Morgan Kaufmann, pp. 305 - 313, 1989.

[15] Sato, A., Yamada, K., Tsukumo, J., and Temma, T., Neural Network Models for Incremental Learning", Proc. of the International Conference on Neural Networks, Helsinki, 1991.

[16] Sejnowski, T.J., and Rosenberg, C.R., "Parallel Networks that learn to Pronounce English Text", Complex Systems 1 (1987) 145-168.

[17] Tesauro, G., Neurogammon Wins Computer Olympiad", Neural Computation, Vol. 1, pp. 321 - 323, 1990.

[18] Vapnik, V.N., Estimation of Dependencies Based on Empirical Data, Springer Verlag, New York, 1982.