Lessons in Neural Network Training: Overfitting May be Harder than Expected, Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI-97, AAAI Press, Menlo Park, California, pp. 540–545, 1997. Copyright AAAI.

Lessons in Neural Network Training: Overfitting May be Harder than Expected

Steve Lawrence[1], **C. Lee Giles**[1], **Ah Chung Tsoi**[2]*
[1] NEC Research, 4 Independence Way, Princeton, NJ 08540, [2] Faculty of Informatics, Uni. of Wollongong, Australia
{lawrence,giles}@research.nj.nec.com, Ah_Chung_Tsoi@uow.edu.au

Abstract

For many reasons, neural networks have become very popular AI machine learning models. Two of the most important aspects of machine learning models are how well the model generalizes to unseen data, and how well the model scales with problem complexity. Using a controlled task with known optimal training error, we investigate the convergence of the backpropagation (BP) algorithm. We find that the optimal solution is typically not found. Furthermore, we observe that networks larger than might be expected can result in lower training and generalization error. This result is supported by another real world example. We further investigate the training behavior by analyzing the weights in trained networks (excess degrees of freedom are seen to do little harm and to aid convergence), and contrasting the interpolation characteristics of multi-layer perceptron neural networks (MLPs) and polynomial models (overfitting behavior is very different – the MLP is often biased towards smoother solutions). Finally, we analyze relevant theory outlining the reasons for significant practical differences. These results bring into question common beliefs about neural network training regarding convergence and optimal network size, suggest alternate guidelines for practical use (lower fear of excess degrees of freedom), and help to direct future work (e.g. methods for creation of more parsimonious solutions, importance of the MLP/BP bias and possibly worse performance of "improved" training algorithms).

Introduction

Neural networks are one of the most popular AI machine learning models, and much has been written about them. A common belief is that the number of parameters in the network should be related to the number of data points and the expressive power of the network. The results in this paper suggest that the characteristics of the training algorithm should also be considered.

Generalization and Overfitting

Neural networks and other AI machine learning models are prone to "overfitting". Figure 1 illustrates the concept using polynomial approximation. A training dataset was created which contained 21 points according to the equation $y = \sin(x/3) + \nu$ where ν is a uniformly distributed random variable between -0.25 and 0.25. The equation was evaluated at $0, 1, 2, \ldots, 20$. This dataset was then used to fit polynomial models with orders between 2 and 20. For order 2, the approximation is poor. For order 10, the approximation is reasonably good. However, as the order (and number of parameters) increases, significant overfitting and increasingly poor generalization is evident. At order 20, the approximated function fits the training data very well, however the interpolation between training points is very poor. Overfitting can also be a very important problem in MLPs, and much work has been devoted to preventing overfitting with techniques such as model selection, early stopping, weight decay, and pruning.

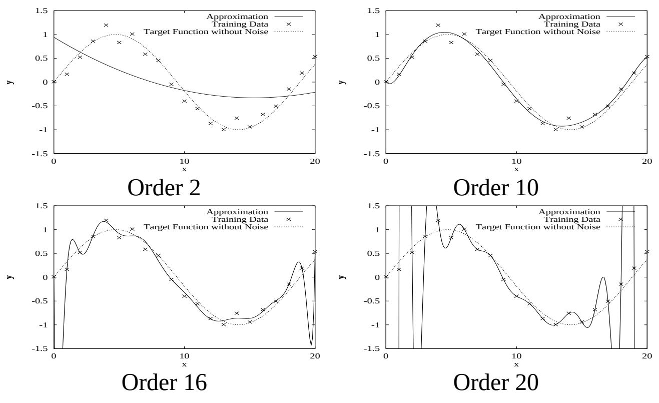

Figure 1. Polynomial interpolation of the function $y = \sin(x/3) + \nu$ in the range 0 to 20 as the order of the model is increased from 2 to 20. ν is a uniformly distributed random variable between -0.25 and 0.25. Significant overfitting can be seen for orders 16 and 20.

Theory

The selection of a model size which maximizes generalization is an important topic. There are several theories for determining the optimal network size e.g. the NIC (Network Information Criterion) which is a generalization of the AIC (Akaike Information Criterion) (Akaike 1973) widely used in statistical inference, the generalized final prediction error (GPE) as proposed by (Moody 1992), and the VC dimension (Vapnik 1995) – which is a measure of the expressive power of a network. NIC relies on a single well-defined minimum to the fitting function and can be unreliable when there are several local minima (Ripley 1995). There is very little published computational experience of the NIC, or the GPE. Their evaluation is prohibitively expensive for large networks. VC bounds have been calculated for various network types. Early VC-dimension work handles only the case of discrete outputs. For the case of real valued outputs, a more general notion of a "dimension" is required.

* Copyright 1997, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

Such a "pseudo-dimension" can be defined by considering a loss function which measures the deviation of predictions from the target values. VC bounds are likely to be too conservative because they provide generalization guarantees simultaneously for any probability distribution and any training algorithm. The computation of VC bounds for practical networks is difficult.

Student-Teacher Task

To investigate empirical performance we will use a student-teacher task (Crane *et al.* 1995) so that we know the optimal solution and can carefully control the problem. The task is as follows:

1. An MLP with m_i input nodes, m_h hidden nodes, and m_o output nodes (the "teacher" network, denoted by $m_i : m_h : m_o$) is initialized with uniform random weights in the range $-K$ to K except for the bias weights which are within the range $(-0.1, 0.1)$.
2. N_{tr} training data points and N_{te} test points are created by selecting Gaussian random inputs with zero mean and unit variance and propagating them through the network to find the corresponding outputs. N_{te} is 5,000.
3. The training data set is used to train new MLPs ("student" networks), with the following architecture: $m_i : m'_h : m_o$, where m'_h is varied from m_h to M, where $M >> m_h$. The initial weights of these new networks are set randomly using the procedure suggested in (Haykin 1994) (i.e. they are not equal to the weights in the network used to create the dataset). Theoretically, if $m'_h \geq m_h$, as it is throughout this paper, then the optimal training set error is zero (for the case where no noise is added to the data).

Simulation Results

This section investigates the training and generalization behavior of the networks for the student-teacher task with the teacher network size fixed but the student network size increasing. For all cases, the data was created with a teacher network architecture $20 : 10 : 1$ (where 20, 10, and 1 were chosen to represent a typical network where the number of inputs is greater than the number of hidden nodes and the specific values were chosen such that the total training time of the simulations was reasonable), and the random weight maximum value, K, was 1. The student networks had the following architecture: $20 : m'_h : 1$, where m'_h was varied from 10 to 50. Theoretically, the optimal training set error for all networks tested is zero, as $m'_h \geq m_h$. However, none of the networks trained here obtained the optimal error (using backpropagation (BP) (Rumelhart, Hinton, & Williams 1986) for 5×10^5 updates)[1].

Each configuration of the MLP was tested with ten simulations, each with a different starting condition (random weights). No method of controlling generalization was used (other than a maximum number of updates) in order to demonstrate this case (not because we advocate the practice). All networks were trained for an identical number of stochastic updates (5×10^5). It is expected that overfitting could occur. The initial learning rate was 0.5 and was reduced linearly to zero during training. We used the standard MLP. Batch update was also investigated – convergence was found to be very poor even when training times were extended by an order of magnitude. The quadratic cost function was used.

Considering that networks with more than 10 hidden units contain more degrees of freedom than is necessary for zero error, a reasonable expectation would be for the performance to be worse, on average, as the number of hidden units is increased. Figure 2 shows the training and test set error as the number of hidden units in the student network is varied from 10 to 50. The results are presented using both box-whiskers plots[2] and the usual mean plus and minus one standard deviation plots. We performed the experiments using three different values for N_{tr}, the number of training points (200, 2,000 and 20,000). On average, the best generalization error corresponds to networks with more than 10 hidden units (30, 40, and 40 respectively for N_{tr} = 200, 2,000, and 20,000)[3] [4]. The number of parameters in the networks is greater than 200, even for the case of 10 hidden units (the numbers of parameters as m'_h is varied from 10 to 50 are (221, 441, 661, 881, 1101). It is of interest to observe the effect of noise on this problem. Figure 2 also shows the results for the case of 200 training points when Gaussian noise is added to the input data with a standard deviation equal to 1% of the standard deviation of the input data. A similar trend is observed.

[1]Alternative optimization techniques (e.g. conjugate gradient) can improve convergence in many cases. However, these techniques often lose their advantage with larger problems and may sometimes be detrimental because the training algorithm bias in BP may be beneficial, see later in the paper.

[2]The distribution of results is often not Gaussian and alternative means of presenting results other than the mean and standard deviation can be more informative (Giles & Lawrence 1997). Box-whiskers plots (Tukey 1977) show the interquartile range (IQR) with a box and the median as a bar across the box. The whiskers extend from the ends of the box to the minimum and maximum values. The median and the IQR are simple statistics which are not as sensitive to outliers as the mean and the standard deviation. The median is the value in the middle when arranging the distribution in order from the smallest to the largest value. If the data is divided into two equal groups about the median, then the IQR is the difference between the medians of these groups. The IQR contains 50% of the points.

[3]Caruana presented a tutorial at NIPS 93 (Caruana 1993) with generalization results on a variety of problems as the size of the networks was varied from "too small" to "too large". "Too small" and "too large" are related to the number of parameters in the model (without consideration of the distribution of the data, the error surface, etc.). Caruana reported that large networks *rarely do worse* than small networks on the problems he investigated. The results in this paper partially correlate with that observation. Caruana suggested that "backprop ignores excess parameters".

[4]This trend varies according to the teacher network size (number of inputs, hidden nodes and outputs), the nature of the target function, etc. For example, the optimal size networks perform best for certain tasks, and in other cases the advantage of larger networks can be even greater.

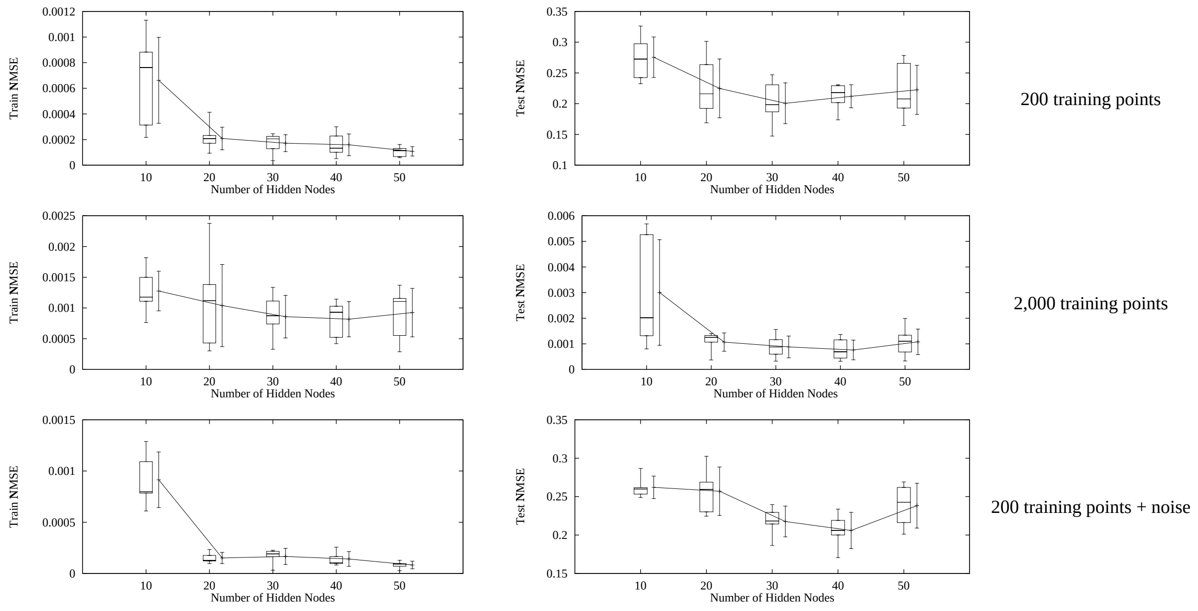

Figure 2. The error for networks with a topology 20:m'_h:1 using 200 (with and without noise) and 2,000 training points. For 20,000 points (not shown) the results were similar to the 2,000 points case. The graphs on the left are the training errors and the graphs on the right are the test errors. The abscissa corresponds to the number of hidden nodes. Box-whiskers plots are shown on the left in each case along with the mean plus or minus one standard deviation which is shown on the right in each case.

These results should not be taken to indicate that oversized networks should always be used. However, they do indicate that oversized networks may generalize well. Additionally, the results indicate that if training is more successful in the larger networks, then it is possible for the larger networks to also generalize better than the smaller networks. A few observations:

1. It remains desirable to find solutions with the smallest number of parameters.
2. A similar result would not be expected if a globally optimal solution was found in the small networks, i.e. if the 10 hidden unit networks were trained to zero error then it would be expected that any networks with extra degrees of freedom would result in worse performance.
3. The distribution of the results is important. For example, observe in figure 2 that the advantage of the larger networks for 2,000 training points is decreased when considering the minimum error rather than the mean error.
4. The number of trials is important. If sufficiently many trials are performed then it should be possible to find a near optimal solution in the optimal size networks (in the limit of an infinite number of random starting points, finding a global optimum is guaranteed with appropriate initialization). Any advantage from using larger size networks would be expected to disappear.
5. Note that there has deliberately been no control of the generalization capability of the networks (e.g. using a validation set or weight decay), other than a maximum number of updates. There are many solutions which fit the training data well that will not generalize well. Yet, contrary to what might be expected, the results indicate that it is possible for oversized networks to provide better generalization. Successive pruning and retraining of a larger network may arrive at a network with similar size to the smaller networks here but with improved training and generalization error.

Note that BP did not find the optimal solution in any of the cases presented here. Also of interest is how the solution found scales with problem complexity. The parameter K can be controlled to investigate this. As K is increased, the function mapping generally becomes more "complex" and less "smooth". Experiments with increasing K show that the solution found becomes progressively worse with respect to the optimal error of zero as K is increased. Analysis of the operation of BP (not given here) supports these results.

Degrees of Freedom Rules based on the degrees of freedom in the model have been proposed for selecting the topology of an MLP, e.g. *"The number of parameters in the network should be (significantly) less than the number of examples"* or *"Each parameter in an MLP can comfortably store 1.5 bits of information. A network with more than this will tend to memorize the data."*. These rules aim to prevent overfitting, but they are unreliable as the optimal number of parameters is likely to depend on other factors, e.g. the quality of the solution found, the distribution of the data points, the amount of noise, any bias in the training algorithm, and the nature of the function being approximated. Specific rules, such as those mentioned above, are not commonly believed to be accurate. However, the stipulation that the number of parameters must be less than the number of examples is typically believed to be true for common datasets. The results here indicate that this is not always the case.

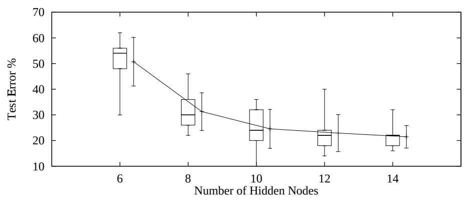

Figure 3. Face recognition example: the best generalizing network has 364 times more parameters than training points (18210 parameters).

Face Recognition Example This section presents results on real data. Figure 3 shows the results of training an MLP to classify 10 people from images of their faces[5]. The training set contains 5 images per person, for a total of 50 training patterns[6]. The test set contained a different set of 5 images per person. A small window was stepped over the images and the image samples at each point were quantized using a two dimensional self-organizing map. The outputs of the self-organizing map for each image sample were used as the inputs to the MLP. In each case, the networks were trained for 25,000 updates. The networks used contain many more parameters than the number of training points (for hidden layer sizes of (6, 8, 10, 12, 14) the number of weights is (7810, 10410, 13010, 15610, 18210)) yet the best training error and the best generalization error corresponds to the largest model. Note that a) generalization has not been controlled using, for example, a validation set or weight decay, and b) overfitting would be expected with sufficiently large networks and sufficiently "successful" training.

When simulated on serial machines, larger networks require longer training times for the same number of updates. Hence, it is of interest to compare what happens when the smaller networks are trained for longer than the larger networks. For this and other problems we have investigated, training for equal time rather than equal numbers of updates does not significantly affect the results or conclusions.

[5]This is not proposed as a practical face recognition technique.

[6]The database used is the ORL database which contains a set of faces taken between April 1992 and April 1994 at the Olivetti Research Laboratory in Cambridge and is available from `http://www.cam-orl.co.uk/facedatabase.html`. There are 10 different images of 40 distinct subjects in the database. There are variations in facial expression and facial details. All the images are taken against a dark homogeneous background with the subjects in an up-right, frontal position, with tolerance for some tilting and rotation of up to about 20 degrees. There is some variation in scale of up to about 10%. The images are greyscale (256 levels) with a resolution of 92x112.

Polynomial and MLP Interpolation

Figure 4 shows the results of using an MLP to approximate the same training set as used earlier in the polynomial approximation example[7]. As for the polynomial case, the smallest network with one hidden unit (4 weights including bias weights), did not approximate the data well. With two hidden units (7 weights), the approximation is reasonably good. In contrast to the polynomial case however, networks with 10 hidden units (31 weights) and 50 hidden units (151 weights) also resulted in reasonably good approximations. Hence, for this particular (very simple) example, MLP networks trained with backpropagation do not lead to a large degree of overfitting, even with more than 7 times as many parameters as data points. It is certainly true that overfitting can be a serious problem with MLPs. However, this example highlights the possibility that MLPs trained with backpropagation may be biased towards smoother approximations. We list a number of possibilities which can lead to such a bias:

1. Training an MLP is NP-complete in general and it is well known that practical training algorithms used for MLPs often results in sub-optimal solutions (e.g. due to local minima)[8]. Often, a result of attaining a sub-optimal solution is that not all of the network resources are efficiently used. Experiments with a controlled task have indicated that the sub-optimal solutions often have smaller weights on average (Lawrence, Giles, & Tsoi 1996). An intuitive explanation for this is that weights typically start out reasonably small (for good reason), and may get trapped in local minima before reaching large values.
2. MLPs are universal approximators (Hornik, Stinchcombe, & White 1989). However, the universal approximation result requires an infinite number of hidden nodes. For a given number of hidden nodes a network may be incapable of representing the required function and instead implement a simpler function which approximates the required function.
3. Weight decay (Krogh & Hertz 1992) or weight elimination (Weigend, Rumelhart, & Huberman 1991) are often used in MLP training and aim to minimize a cost function which penalizes large weights. These techniques tend to result in networks with smaller weights.
4. A commonly recommended technique with MLP classification is to set the training targets away from the bounds of the activation function (e.g. (-0.8, 0.8) instead of (-1, 1) for the $\tanh$ activation function) (Haykin 1994).

MLP networks are, of course, not always this resistant to

[7]Training details were as follows. A single hidden layer MLP, backpropagation, 100,000 stochastic training updates, and a learning rate schedule with an initial learning rate of 0.5 were used.

[8]The results in this paper show that BP training often results in sub-optimal solutions. Commonly, these solutions are referred to as local minima, about which much has been written and proven (e.g. (Yu 1992)). However, it is not only local minima that create trouble for BP – other error surface features such as "ravines" and "plateaus" or "flat spots" can also be troublesome. The error surface for two different problems may have no local minima yet one may be far more amenable to gradient descent optimization.

overfitting. For example, when repeating the above experiment but only evaluating the equation at $0, 1, 2, \ldots, 5$ (creating 6 data points), overfitting is seen with only three hidden nodes.

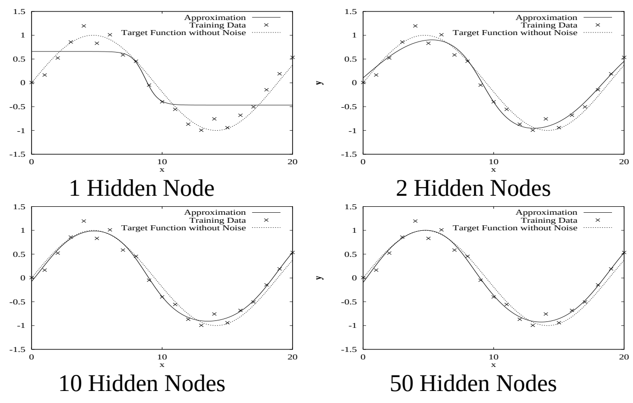

Figure 4. MLP interpolation of the function $y = \sin(x/3) + \nu$ in the range 0 to 20 as the number of hidden nodes is increased from 1 to 50. ν is a uniformly distributed random variable between -0.25 and 0.25. A large degree of overfitting can not be observed.

Network Size and Degrees of Freedom

A simple explanation for why larger networks can sometimes provide improved training and generalization error is that the extra degrees of freedom can aid convergence, i.e. the addition of extra parameters can decrease the chance of becoming stuck in local minima or on “plateaus”, etc. (Kröse & van der Smagt 1993). This section presents a visualization technique for showing the weights in the student networks as the network size is varied. A smaller task was used to aid visualization: the teacher network topology was 5:5:1 and the student networks contained 5, 15, and 25 hidden nodes. 1,000 training points were used and K was 2.

Figures 5 show the weights in the student networks for the case when Gaussian noise with standard deviation 5% of the input standard deviation was added to the inputs (we also performed experiments with 0% and 10% which produced similar results). The diagrams are plotted as follows: The columns (1 to 6) correspond to the weights from the hidden nodes to the bias and the 5 input nodes. The rows are organized into groups of two with a space between each group. The number of groups is equal to the number of hidden nodes in the student network. For the two rows in each group, the top row corresponds to the teacher network and the bottom row corresponds to the student network. The idea is to compare the weights in the teacher and student networks. A couple of difficulties arise in this comparison which are resolved as follows. Firstly, there is no reason for hidden node 1 in the teacher network to correspond to hidden node 1 in the student network, etc. This problem is resolved by finding the best matching set of weights in the student network for each hidden unit in the teacher network, and matching the hidden nodes accordingly. These matches are ordered according to the quality of the match, i.e. the top two rows shows the teacher network hidden node which was best approximated by a student hidden node. Likewise, the worst match is at the bottom. A second problem is that trying to match the weights from the hidden nodes to the input nodes does not take into account the output layer weights, e.g. exactly the same hidden node function could be computed with different weights if the hidden nodes weights are scaled and the output layer weights are scaled accordingly. For the case of only one output which is considered here, the solution is simple: the hidden layer weights are scaled according to the respective output layer weight. Each individual weight (scaled by the appropriate output weight) is plotted as follows: the square is shaded in proportion to the magnitude of the weight, where white equals 0 and black equals the maximum value for all weights in the networks. Negative weights are indicated by a white square inside the outer black square which surrounds each weight.

Observations: a) the teacher network weights are matched more closely by the larger networks (consider the fourth and fifth best matching groups of two rows), b) the extra weights in the larger networks contribute to the final approximation in only a minor way, c) the hidden units in the larger networks do not appear to be used redundantly in this case – this may be related to the artificial nature of the task, and d) the results indicate that pruning (and optionally retraining) the larger networks may perform well. A conclusion is that backpropagation can result in the underutilization of network resources in certain cases (i.e. some parameters may be ineffective or only partially effective due to sub-optimal convergence).

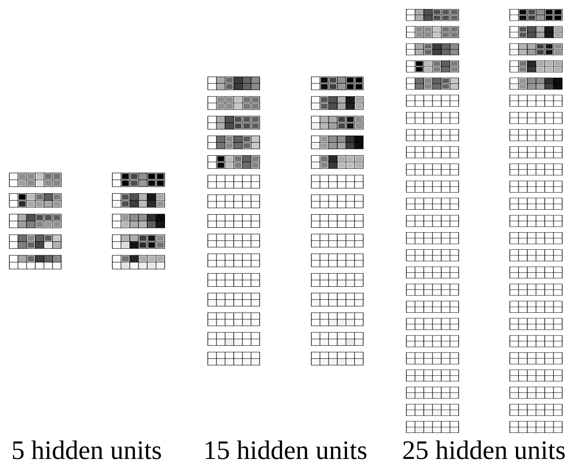

Figure 5. The weights after training in networks with 5, 15, and 25 hidden units for the case of Gaussian noise with standard deviation 5% of the standard deviation of the inputs. In each case, the results are shown for two networks with different random starting weights. The plotting method is described in the text.

Learning Theory

The results are not in contradiction with statistical learning theory. (Vapnik 1995) states that machines with a small VC dimension are required to avoid overfitting. However, he also states that *"it is difficult to approximate the training data"*, i.e. for a given problem in MLP approximation, the goal is to find the appropriate network size in order to minimize the tradeoff between overfitting and poor approximation. Vapnik suggests that the use of *a priori* knowledge may be required for small training error and small generalization error. For the case of linear output neurons, Barron 1991 has derived the following bound on the total risk for an MLP estimator: $O\left(\frac{C_f}{m_h}\right) + O\left(\frac{m_h m_i}{N_{tr}} \log N_{tr}\right)$, where C_f is the first absolute moment of the Fourier magnitude distribution of the target function f and is a measure of the "complexity" of f. Again, a tradeoff can be observed between the accuracy of the best approximation (which requires larger m_h), and the avoidance of overfitting (which requires a smaller m_h/N_{tr} ratio). The left-hand term (the approximation error) corresponds to the error between the target function and the closest function which the MLP can implement. For the noise-free artificial task, the approximation error is zero for $m'_h \geq 10$. Based on this equation, it is likely that $m'_h = 10$ would be selected as the optimal network size (note that the results reported here use sigmoidal rather than linear output neurons). Why do the theory and practical results differ? Because the domain of applicability of the theory does not cover the practical case and the assumptions incorporated in the theory are not always reasonable. Specifically, this theory does not take into account limited training time, different rates of convergence for different f, or sub-optimal solutions.

Recent work by (Bartlett 1996) correlates with the results reported here. Bartlett comments: *"the VC-bounds seem loose; neural networks often perform successfully with training sets that are considerably smaller than the number of network parameters"*. Bartlett shows (for classification) that the number of training samples only needs to grow according to A^{2l} (ignoring log factors) to avoid overfitting, where A is a bound on the total weight magnitude for a neuron and l is the number of layers in the network. This result and either an explicit (weight decay etc.) or implicit bias towards smaller weights leads to the phenomenon observed here, i.e. larger networks may generalize well and better generalization is possible from larger networks if they can be trained more successfully than the smaller networks (e.g. reduced difficulty with local minima). For the task considered in this paper, the distribution of weights after training moves towards smaller weights as the size of the student network increases.

Conclusions

It can be seen that backpropagation fails to find an optimal solution in many cases. Furthermore, networks with more weights than might be expected can result in lower training and generalization error in certain cases. Overfitting behavior is significantly different in MLP and polynomial models – MLPs trained with BP are biased towards smoother solutions. Given infinite time and an appropriate alternate training algorithm, an optimal solution could be found for an MLP. However, the examples in this paper illustrate that the mode of failure exhibited by backpropagation can in fact be beneficial and result in better generalization over "improved" algorithms, in so much as the implicit smoothness bias created by the network structure and training algorithm matches the desired target function. This bias may account for part of the success MLPs have encountered over competing methods in real-world problems.

Acknowledgments

We would like to acknowledge useful discussions with and comments from R. Caruana, S. Hanson, G. Heath, B.G. Horne, T. Lin, K.-R. Müller, T. Rognvaldsson, and W. Sarle. The opinions expressed herein are those of the authors. Thanks to the Olivetti Research Laboratory and F. Samaria for compiling and maintaining the ORL database.

References

Akaike, H. 1973. Information theory and an extension of the maximum likelihood principle. In Petrov, B. N., and Csaki, F., eds., *Proceeding 2nd International Symposium on Information Theory*. Budapest: Akademia Kiado. 267–281.

Barron, A. 1991. Complexity regularization with application to artificial neural networks. In Roussas, G., ed., *Nonparametric Functional Estimation and Related Topics*. Dordrecht, The Netherlands: Kluwer Academic Publishers. 561–576.

Bartlett, P. 1996. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. Technical report, Australian National University.

Caruana, R. 1993. *Generalization vs. Net Size*. Denver, CO: Neural Information Processing Systems, Tutorial.

Crane, R.; Fefferman, C.; Markel, S.; and Pearson, J. 1995. Characterizing neural network error surfaces with a sequential quadratic programming algorithm. In *Machines That Learn*.

Giles, C. L., and Lawrence, S. 1997. Presenting and analyzing the results of AI experiments: Data averaging and data snooping. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI-97*. Menlo Park, California: AAAI Press. 362–367.

Haykin, S. 1994. *Neural Networks, A Comprehensive Foundation*. New York, NY: Macmillan.

Hornik, K.; Stinchcombe, M.; and White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2:359–366.

Krogh, A., and Hertz, J. 1992. A simple weight decay can improve generalization. In Moody, J.; Hanson, S. J.; and Lippmann, R. P., eds., *Advances in Neural Information Processing Systems*, volume 4. San Mateo, CA: Morgan Kaufmann. 950–957.

Kröse, B., and van der Smagt, P. 1993. *An Introduction to Neural Networks*. University of Amsterdam, fifth edition.

Lawrence, S.; Giles, C. L.; and Tsoi, A. 1996. What size neural network gives optimal generalization? Convergence properties of backpropagation. Technical Report UMIACS-TR-96-22 and CS-TR-3617, Institute for Advanced Computer Studies, University of Maryland, College Park MD 20742.

Moody, J. 1992. The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In Moody, J.; Hanson, S. J.; and Lippmann, R. P., eds., *Advances in Neural Information Processing Systems*, volume 4. San Mateo, CA: Morgan Kaufmann. 847–854.

Ripley, B. 1995. Statistical ideas for selecting network architectures. Invited Presentation, Neural Information Processing Systems 8.

Rumelhart, D.; Hinton, G.; and Williams, R. 1986. Learning internal representations by error propagation. In Rumelhart, D., and McClelland, J., eds., *Parallel Distributed Processing*, volume 1. Cambridge: MIT Press. chapter 8, 318–362.

Tukey, J. 1977. *Exploratory Data Analysis*. Reading, MA: Addison-Wesley.

Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. Springer.

Weigend, A.; Rumelhart, D.; and Huberman, B. 1991. Generalization by weight-elimination with application to forecasting. In Lippmann, R. P.; Moody, J.; and Touretzky, D. S., eds., *Advances in Neural Information Processing Systems*, volume 3. San Mateo, CA: Morgan Kaufmann. 875–882.

Yu, X.-H. 1992. Can backpropagation error surface not have local minima. *IEEE Transactions on Neural Networks* 3:1019–1021.