REIHE INFORMATIK
3/95

Quantifying a critical training set size for generalization and overfitting using teacher neural networks

R. Lange und R. Männer
Universität Mannheim
Container B6
D-68131 Mannheim

Abstract:
Teacher neural networks are a systematic experimental approach to study neural networks. A teacher is a neural network that is employed to generate the examples of the training and the testing set. The weights of the teacher and the input parts of the examples are set according to some probability distribution. The input parts are then presented to the teacher neural network and recorded together with its response. A pupil neural network is then trained on this data. Hence, a neural network instead of a real or synthetic application defines the task, according to which the performance of the pupil is investigated. One issue is the dependence of the training success on the training set size. Surprisingly, there exists a critical value above which the training error drops to zero. This critical training set size is proportional to the number of weights in the neural network. A sudden transition exists for the generalization capability, too: the generalization error measured on a large independent testing set drops to zero, and the effect of overfitting vanishes. Thus, there are two regions with a sudden transition in-between: below the critical training set size, training and generalization fails, and severe overfitting occurs; above the critical training set size, training and generalization is perfect and there is no overfitting.

Also appeared as:
R. Lange and R. Männer. Quantifying a critical training set size for generalization and overfitting using teacher neural networks. In M. Marinaro and P. Morasso, editors, *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, volume I, pages 497–500, London, GB, May 1994. Springer.

Contact:

- Dipl. Phys. Rupert Lange
  Dept. Computer Science V
  University of Mannheim
  B6
  68131 Mannheim
  Germany

phone	+49-621-292-5707
fax	+49-621-292-5756
e-mail	lange@mp-sun1.informatik.uni-mannheim.de

- Prof. Dr. Reinhard Männer
  Dept. Computer Science V
  University of Mannheim
  B6
  68131 Mannheim
  Germany

phone	+49-621-292-5758
fax	+49-621-292-5756
e-mail	maenner@mp-sun1.informatik.uni-mannheim.de

1 Introduction

Neural networks are trained to map a given training set of input-output pairs as accurate as possible. In most applications, the training set models a task, which the neural network has to complete — it should not just learn the training data. This *generalization* is estimated by measuring how accurate the neural network maps an independent testing set. However, it is hard to know in advance how well a neural network can generalize. Two major questions arise: Is sufficient data available to model the task? Which architecture fits the given task best? Typically, both questions must be answered by trial and error due to the lack of sound knowledge that would allow to predict the number of training samples required and the optimal architecture.

We have tackled these questions by means of *teacher neural networks*. This is an experimental approach which is sufficiently general to provide meaningful insights. The results reported here show, to start with, that there exists a critical training set size at which both the generalization error and overfitting drop to zero and, secondly, that this critical training set size is proportional to the number of parameters of the neural network.

2 Experiments

Teacher. A teacher is a neural network with fixed weights[1] which is employed to generate the training and testing sets. Hence, the teacher defines the task, instead of a (real or synthetic) data set. There are several advantages of this approach. First, one knows that a perfect solution of the task exists. Second, as many input-output pairs as desired can be produced. Third, the complexity of the task can be scaled by varying the architecture of the teacher.

Our teacher has an n–n–n architecture: an input layer, one hidden layer, an output layer; each n neurons. The weights are uniformly drawn from $[-1,+1]$. The transfer function is $tanh$. Pairs for the training and testing set are produced by presenting random input $\vec{\xi} \in \{-1,+1\}^n$ to the teacher in order to get the corresponding target output $\vec{\zeta} \in [-1,+1]^n$. Let p^t and p^g be the size of the training and testing set.

Pupil. The *pupil neural network* is trained on the training set produced by the teacher using online back propagation (Rumelhart et al., 1986). The pupil and teacher architectures are identical. We have scanned a wide range of training lengths. For reasons of lucidity, none of the well-known modifications that increase convergence speed have been applied. Thus, only one parameter is left, the learning rate γ.

Error measure. The training error is $\epsilon^t = \frac{1}{np^t} \sum_{\mu=1}^{p^t} \sum_{i=1}^{n} |z_i^\mu - \zeta_i^\mu|$, where z_i^μ is the actual output of the pupil for input $\vec{\xi^\mu}$. Regardless of n and p^t, this

[1] Here and in the following weights include thresholds.

error measure is bounded to $[0, 2]$. Analogously, the generalization error ϵ^g is defined on the testing set.

Training set size. Our interest is to quantify how generalization depends on the number of training samples. The training samples are used to adjust the parameters of the pupil neural network. Therefore, we expected the training set size to scale with the number of pupil parameters to be fixed, i.e. the number $w = 2n(n+1)$ of weights. This choice is upheld by bounds on the generalization $\epsilon^g(p^t)$ found for linear threshold networks (Baum and Haussler, 1989). Accordingly, we have measured the training set size in units of w, in order to investigate comparable ranges for different neural network sizes.

Parameter settings. We have studied $\epsilon^g(p^t)$ for 13 different n–n–n networks, with $n = 5, 10, \ldots, 65$. The size of the training set varied over the range $[0, w]$, $\frac{p^t}{w} = \frac{1}{16}, \frac{2}{16}, \ldots, \frac{16}{16}$. The testing set must be large enough to measure generalization accurately; we used $p^g = w$. The pupil was trained with a learning rate $\gamma = 0.01$. Weights were initialized with random values uniformly drawn from $[-0.01, +0.01]$.

3 Results

Quantities. We measured the training error ϵ^t and the generalization error ϵ^g during the training process, at $\tau = 0, 1, 2, 4, \ldots, 131072 = 2^{17}$ weight updates. So, instead of the usual choice of *epochs* (presentations of the entire training set), τ gives the number of presentations of input-output *pairs* that are presented to the pupil. The reason for this is that we wanted to compare the results for different training set sizes; this would not be possible if we measured training time in epochs, because in this case a larger training set size would mean longer training.

Generalization and overfitting. Figure 1 shows the generalization error during training for a 50-50-50 network. The different curves correspond to different training set sizes p^t, and illustrate that the behaviour depends dramatically on this parameter; one has either overfitting and bad generalization ($p^t \leq \frac{5}{16}w$) or no overfitting and perfect generalization ($p^t \geq \frac{6}{16}w$). This is true for all neural network sizes that we have observed.

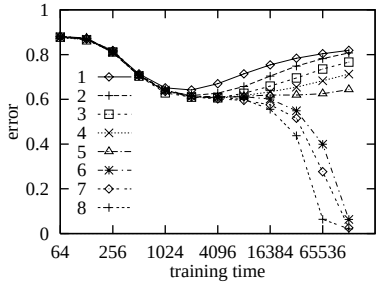

Figure 1: Generalization error as a function of training time for different training set sizes $\frac{16p^t}{w} = 1, 2, \ldots, 8$. The first out of 40 runs for $n = 50$ is shown.

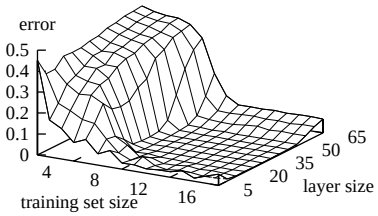

Figure 2: Perfect generalization above critical training set size. The generalization error ϵ^g vanishes.

Scaling behaviour. For the results presented now, we picked the smallest values ϵ^t_{opt} and ϵ^g_{opt} and recorded these together with the corresponding times τ^t_{opt} and τ^g_{opt}. The difference $2^{17} - \tau^g_{opt}$ between the entire training time and the time for optimal generalization is then a measure for overfitting. This becomes clear in Figure 1: the training time for optimal generalization τ^g_{opt} differs from the training length 2^{17} for curves 1–5; overfitting has occured.

The results for $\epsilon^g_{opt} and \tau^g_{opt}$ presented in the following are the mean of 40 values measured for different weight initializations.

Figure 2 shows the dependence of the generalization error on the training set size for different neural networks. Over a wide range of layer sizes n, the neural networks show the behaviour discussed above for Figure 1. A critical training set size $p^t_c \approx \frac{6}{16}w$ exists with no generalization below and perfect generalization above. The results deviate from this idealized behaviour for both the smallest and the largest neural networks ($n \leq 10$ and $n \geq 55$).

First, we discuss the graph in the region of perfect generalization. The curves are rather ragged for small n, due to bad statistics; for large n, the residual error increases. The latter effect has two reasons. The first reason is trivial; the training length is not sufficient large, as Figure 1 indicates already. The second reason is not that evident; the choice of $\gamma = 0.01$ is not adequate. We know from other experiments we have carried out with teacher neural networks that larger networks require smaller γ. To check on that, we have rerun the simulation for $n = 65$ with $\gamma = 0.004$. This yielded perfect generalization after approximately 2^{19} pairs.

Secondly, considering the region with low generalization, the residual error increases with the layer size n, particularly for small n. This suggests better generalization for small n, which is not really true, because the condition for measuring generalization — independent training and testing set — is no longer fulfilled[2].

Figure 3 displays the time at which generalization is optimal for different neural networks. Indirectly, this shows the dependence of overfitting on the training set size. Be aware that the training set size axis is reversed compared to Figure 2! Again, over a wide range of layer sizes n, the neural networks show the behaviour described above for Figure 1. A critical training set size $p^t_c \approx \frac{6}{16}w$ exists with no overfitting above and clear overfitting below. As to the smallest

[2]For small n, the training set size $p^t \propto 2n(n+1)$ becomes comparable to 2^n, the number of possible inputs. Hence, the lower n, the more input-output pairs exist that reside in both the training and testing set.

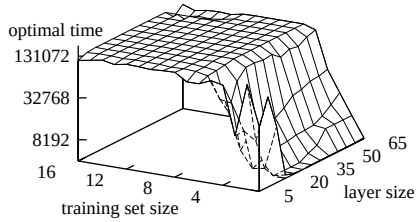

Figure 3: No overfitting above critical training set size. $\tau^g_{opt} = 2^{17}$. The smallest generalization error ϵ^g is at the very end of training.

training set, for example, the best generalization is achieved at approximately 4096 pairs, i.e. only the 256th part of the entire training time is exploited.

4 Discussion

We have trained neural networks on tasks defined by teacher neural networks. It has been shown how generalization and overfitting depend on the training set size. We expected that the generalization error decreases as the number of training samples is increased. This has been observed for tasks defined by (real or synthetic) data, too. Surprisingly, our studies revealed that there are two distinct regions — one with low generalization and severe overfitting and one with perfect generalization without overfitting — with a sudden transition between them. This transition identifies a critical training set size, which we could show to be proportional to the number of weights in the pupil neural network. Although tasks defined by teacher neural networks do not compare directly with real applications, teacher neural networks are a valuable tool to investigate neural networks beyond the scope of specific applications. Our experience with neural networks used for signal processing proved this conjecture. Regarding a critical learning rate, our results from teacher experiments applied directly to the application. Further results dealing with the question of the optimal architecture are promising, too.

References

[1] E. Baum and D. Haussler. What size net gives valid generalization? *Neural Computation*, 1:151–160, 1989.

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