ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 193-198

Noise Reduction by Multi-Target Learning

John A. Bullinaria

Neural Networks Research Group, Psychology Department,
Edinburgh University, 7 George Square, Edinburgh

Abstract. We review the problems associated with neural networks learning from noisy and/or ambiguous training data and propose a simple procedure that appears to alleviate these problems. By making use of the readily available output error information, a network is able to choose the correct output targets from sets of possibilities and generate new targets if any of the correct ones appear to be missing from the given training data.

Introduction

Consider a system (such as an artificial neural network) that is expected to learn and generalize from training data consisting of a set of input vectors with corresponding output vectors. We generally say that the training data is *noisy* if at least some of the output vectors differ from the 'correct' output vector by some 'small' amount. If this is the case, it is well known that, unless care is taken to prevent the system from over-learning (i.e. modelling the details of the noise rather than just the underlying regularities), the systems generalization performance (i.e. its ability to deal appropriately with new input data) will be poor.

Sometimes the noise will result in there being more than one distinct output vector corresponding to a given input vector (and this set of output vectors may or may not include the 'correct' output vector), in which case we say the training data is *ambiguous*. Ambiguous training data that arises in other ways, in which the differences between the output vectors are not necessarily small, will also result in over-learning problems. Such ambiguous data may occur because:

1. The input information is incomplete.
2. The different output vectors are actually equivalent in some way.

Both causes are familiar to us from reading. First, the same set of letters may be pronounced in different ways, but given sufficient additional context information, we know which pronunciation is appropriate. Second, given only a set of letters and the corresponding set of phonemes for a word, it is not obvious how the letters and phonemes line up - there will be many possible output targets for each input letter.

It is well known that human brains can handle noisy data and both types of ambiguity very well, but artificial neural networks tend not to be so successful. In this short paper we will review the problems involved and propose a simple approach which appears to alleviate these problems.

Conventional Learning

Let us begin by looking at how a simple input-output mapping is learnt by a conventional feedforward back-propagation network with one hidden layer. We will

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 193-198

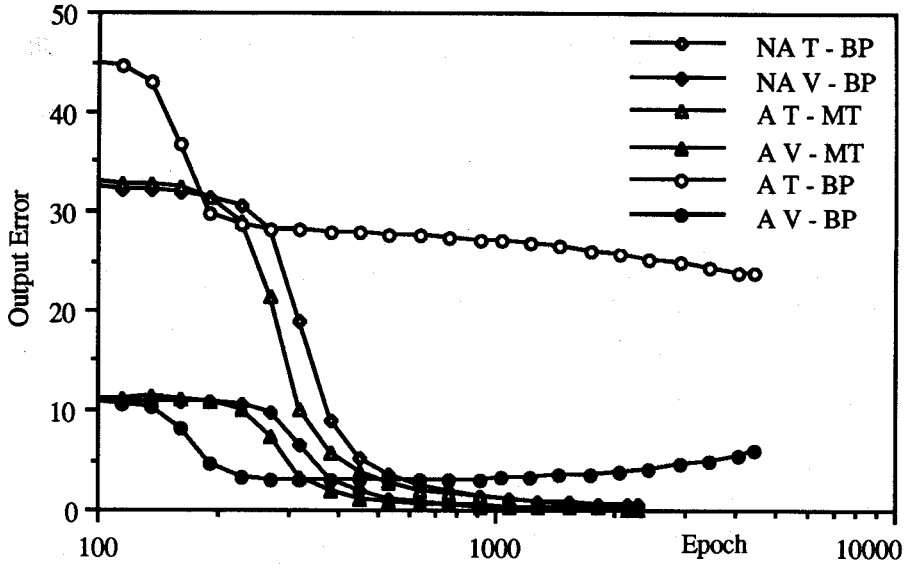

Fig. 1. Evolution of training (T) and validation (V) error measures for the non-ambiguous (NA) and ambiguous (A) data during standard (BP) and multi-target (MT) training.

illustrate what happens by considering the simple, but non-trivial, case of a nine bit cyclic exclusive-or. We split the 512 possible training patterns randomly into 462 for training and 50 for validation (i.e. testing of generalization ability). Figures 1 and 2 show how leaning proceeds for a network with 100 hidden units, a learning rate of 0.02 and no momentum. We see that the sum squared error score and the number of output errors decease for both the training and validation sets.

Now suppose we introduce noise into the training data such that there are now two possible outputs for each input, the original correct output plus another with two of the nine bits flipped at random. Clearly the network will have problems with this ambiguous data since it cannot possibly produce two different outputs for the same input. We see from Figures 1 and 2 that the network first identifies the main regularities in the data (and hence the correct outputs) and the validation errors reach a minimum, but then the network goes on to attempt to accommodate the erroneous outputs. The training data error score continues to decrease as we would expect of a gradient descent training algorithm, but the number of output errors, the validation error score and the number of validation output errors all begin to increase again. This is a typical manifestation of the 'over-training' or 'over-fitting' that occurs in neural networks with too many parameters for the data they are trying to model (Baum & Haussler, 1989). One obvious solution (from the point of view of Figures 1 and 2) is to terminate the training just as the validation error starts increasing again (e.g. Morgan & Bourlard, 1990; Weigend, Huberman & Rumelhart, 1990) but at this point the error measure is still quite large. Other possible solutions involve reducing the effective number of free parameters (i.e. connection weights) in various ways such as network pruning (e.g. Karnin, 1990), weight decay (e.g. Krogh & Hertz, 1992) or weight sharing (e.g. Nowlan & Hinton, 1992). Here we propose an alternative approach that is perhaps more biologically and psychologically plausible: we learn to ignore the noise.

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 193-198

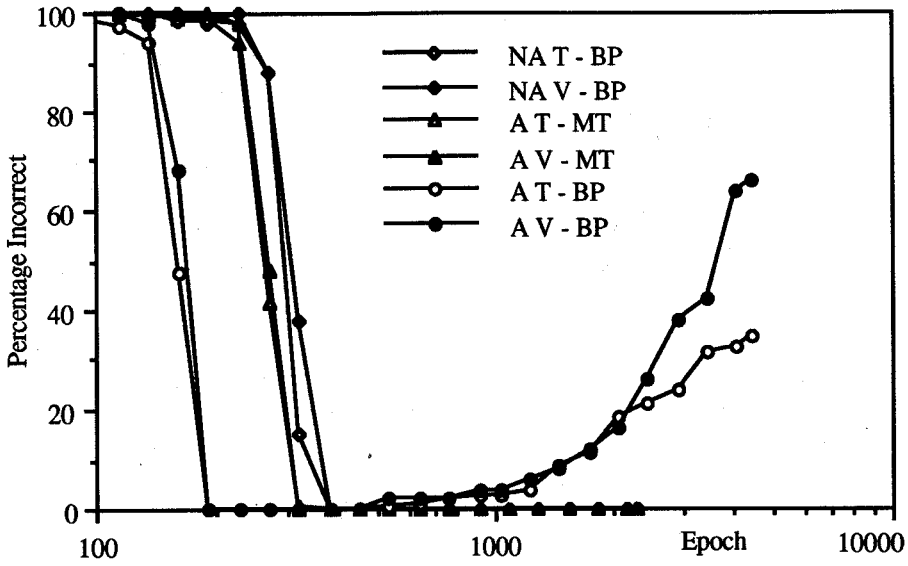

Fig. 2. Evolution of training (T) and validation (V) output errors for the non-ambiguous (NA) and ambiguous (A) data during standard (BP) and multi-target (MT) training.

Multi-Target Learning

Suppose we have several possible target outputs for a given input. Whether one is right and the others wrong, or they are all equivalent but one is preferable, we need to choose to train on one and ignore the others. If we had to choose consciously between the targets we would naturally choose the one that best fitted in with our existing knowledge of the domain in question taking into account any other information we might have concerning the appropriateness of each target. In neural network terms this means choosing the target that already gives the lowest output error score weighted by a prior probability that the target is the appropriate one.

In Bullinaria (1993a) it was shown that, if E_α is the output error score for target α and P_α is the 'prior probability' that target α is the correct one, then by training only on the target for each input with the lowest value of $S = (1 - P_\alpha).E_\alpha$ the network can, under certain circumstances, *learn* which is the appropriate target for each input. Lack of space prevents us from going through the full analysis and all the empirical results here, but intuitively what happens is that by the time the validation errors would start increasing again the network has settled down into using only the correct (or best) targets and the other targets are simply not there to cause any problems. It is important that we have sufficient numbers of hidden units and training examples and sufficiently small learning rate that the training proceeds smoothly, or the network can get stuck with the wrong set of targets, but apart from that the procedure appears to be quite robust. Figures 1 and 2 show that the learning curves for our ambiguous data using this multi-target approach (with equal P_α for the two targets) are not significantly different from the non-ambiguous case.

Such a multi-target approach was successfully applied in Bullinaria (1993b) to a realistic problem in which the different output vectors were not the result of errors but

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 193-198

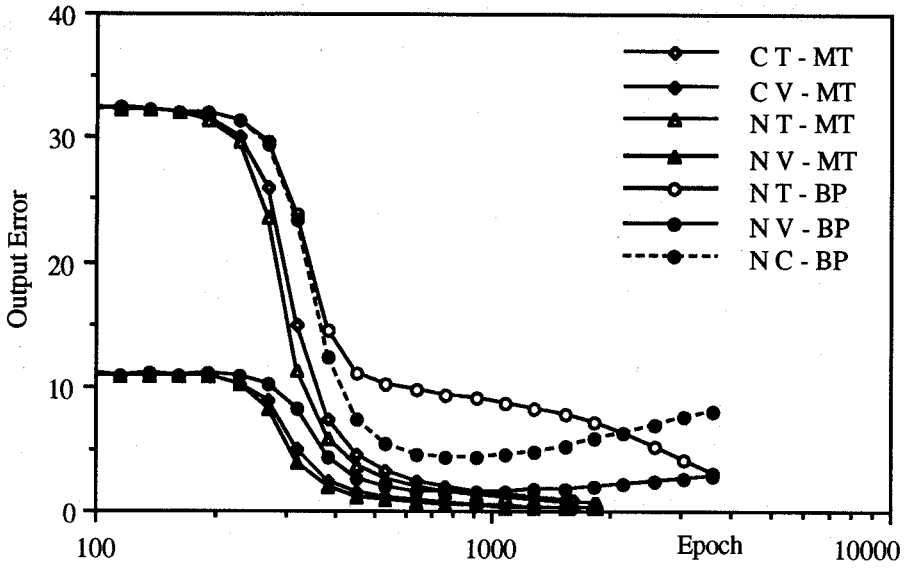

Fig. 3. Evolution of training (T) and validation (V) error measures for the correct (C) and noisy (N) data for standard (BP) and multi-target (MT) training.

were introduced specifically to allow the network to choose which of a number of possible ways of representing a given output was best. The domain in question was that mentioned in the introduction, namely reading aloud. The training data here consists of sets of letters and the corresponding sets of phonemes. Since there is not generally a one-to-one correspondence between the letters and phonemes, we have the so-called alignment problem whereby we do not know which is the appropriate target phoneme for a given input letter. Previously this problem was solved by hand prior to training (Sejnowski & Rosenberg, 1987). The multi-target approach obviated the need for such pre-processing of the training data by using a set of equally probable targets, one for each possible alignment, and allowing the network to choose one of them. A word like 'ace' thus had three targets (/As–/, /A–s/ and /–As/) and the multi-target approach allowed the network to learn that /As–/ is the one that ties in best with the optimal set of grapheme to phoneme rules.

Noise Reduction

Having a network pick the right target from a set of possibilities solves part of our problem, but what about noisy data in which some of the correct targets have been lost completely? Again we consider how we would proceed consciously. It is usually reasonable to assume that the correct data will not be much different from the data we actually have. Moreover, if a slight deviation from the actual data fits in much better than the actual data with everything else, then it is likely that the deviation results in the correct data and we should use that rather than the actual data. In neural network terms this means also considering targets that differ from the given ones and then using the above multi-target approach to determine which of the targets to use. Clearly, we should consider the deviating targets to be less probable than the

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 193-198

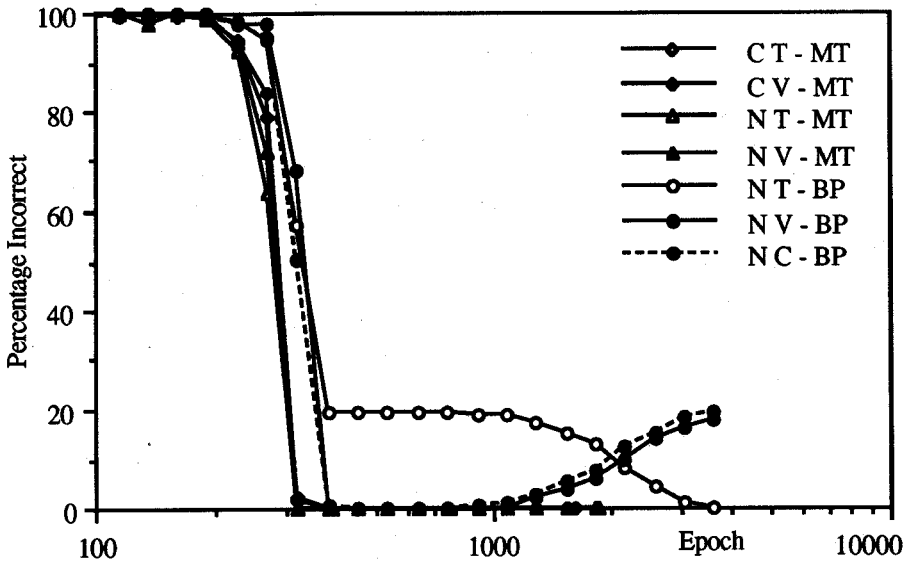

Fig. 4. Evolution of the number of training (T) and validation (V) output errors for the correct (C) and noisy (N) data for standard (BP) and multi-target (MT) training.

given targets (and the more the deviation, the less the probability), or the network will be free to train itself on virtually anything it chooses. The easiest way to proceed is to reduce the targets' probability by a factor F for each bit flipped from the original target then, in the measure of suitability $S = (1 - P_\alpha F^m).(E_\alpha - \Delta_m E)$, the reduction in error score $\Delta_m E$ is offset against the reduced probability $P_\alpha F^m$ and changes are only made when there is good reason. Naively, this sounds like a lot of potential targets, e.g. 2^n for each input if there are n output bits. In fact, in practice, the number of extra targets will be less than n for each given target since (assuming each change of output bit reduces the probability by the same factor) the best target with m bits changed can be obtained from the original target by simply flipping the m bits contributing most to the total error of that target. Moreover, all the necessary error information is produced in the course of standard back-propagation training.

Suppose we introduce noise into the 9 bit cyclic exclusive-or data used previously by allowing each output bit to be flipped with a probability of 3%. Figures 3 and 4 show a typical run in which the training data included 82 output patterns with one bit wrong and 8 with two bits wrong. We used the same parameters as before with $P_\alpha = 0.8$ and $F = 0.8$ throughout. There are three cases to consider. First, we check that our noise reduction procedure can learn from the correct data without unnecessarily changing that data. Next, we train by standard back-propagation with the noisy data and find the expected over-fitting problems. Finally, we see that our noise reduction procedure is capable of correcting all the errors in the training data and ends up with the same perfect generalization performance as the non-noisy case. In 25 runs with different initial weights, the noisy data was corrected resulting in perfect generalization every time and in all but one run the correct data was learnt without any unnecessary changes. This 2% failure rate rose to 14%, 29% and 50% for learning rates of 0.04, 0.08 and 0.16 respectively.

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 193-198

Discussion

Despite the apparent successes outlined above, there are still many potential problems with this approach. Firstly, as mentioned before, it is important to keep the parameters set such that the learning is smooth or the wrong targets are chosen and unnecessary target changes occur. Fortunately, when the learning goes wrong it is usually easy to tell: in the one case in our main set of 50 runs which had any incorrect target changes, there were 116 of them! Moreover, this 'all or nothing' effect seems to persist for the less successful higher learning rate cases. Another problem is that, if we set the probability reduction factor F too low, it will be difficult for the network to correct all the errors, particularly if there are several in one training pattern. On the other hand, if it is set too high, the network will make corrections where they are not necessary and get totally confused. It is not immediately obvious under what circumstances the F 'window' will exist and be large enough to be useful. It is possible that we will always have to err on the side of caution and fail to correct all the errors, in which case we may need to use this approach in conjunction with other solutions to the over-fitting problem. Finally, we have only considered the case of simple binary input-output mappings with noisy or ambiguous outputs: there are many other classes of problem we have yet to deal with.

In conclusion then, the multi-target approach to learning from noisy and ambiguous training data has not yet been shown to be universally useful but the indications are that it is at least worthy of further investigation.

References

Baum, E.B. & Haussler, D. (1989) What Size Net Gives Valid Generalization? *Neural Computation*, **1**, 151-160.

Bullinaria, J.A. (1993a) Neural Network Learning from Ambiguous Training Data. Submitted to *Connection Science*.

Bullinaria, J.A. (1993b) Representation, Learning, Generalization and Damage in Neural Network Models of Reading Aloud. In preparation.

Karnin, E.D. (1990) A Simple Procedure for Pruning Back-Propagation Trained Neural Networks. *I.E.E.E. Transactions on Neural Networks*, **1**, 239.

Krogh, A. & Hertz, J. A. (1992) A Simple Weight Decay Can Improve Generalization. In J. E. Moody, S. J. Hanson & R. P. Lippman (Eds.) *Advances in Neural Information Processing Systems 4*, pages 950-957. San Mateo, CA: Morgan Kauffmann.

Morgan, N. & Bourlard, H. (1990) Generalization and Parameter Estimation in Feedforward Nets: Some Experiments. In D.S. Touretzky (Ed.) *Advances in Neural Information Processing Systems 2*, pages 630-637. San Mateo, CA: Morgan Kauffmann.

Nowlan, S.N. & Hinton, G.E. (1992) Simplifying Neural Networks by Soft Weight Sharing. *Neural Computation*, **4**, 473-493.

Sejnowski, T.J. & Rosenberg, C.R. (1987) Parallel Networks that Learn to Pronounce English Text. *Complex Systems*, **1**, 145-168.

Weigend, A.S., Huberman, B.A. & Rumelhart, D.E. (1990) Predicting the Future: A Connectionist Approach. *International Journal of Neural Systems*, **1**, 193.