S10.5

Unstable Connectionist Networks In Speech Recognition

Richard Rohwer, Stephen Renals and Mark Terry

Centre for Speech Technology Research,
University of Edinburgh,
Edinburgh EH1 1HN
UK

ABSTRACT

Connectionist networks evolve in time according to a prescribed rule. Typically, they are designed to be stable so that their temporal activity ceases after a short transient period. However, meaningful patterns in speech have a temporal component: therefore it seems natural to attempt to map the temporality of speech patterns onto the temporality of an *unstable* network.

We have begun some exploratory experiments to train networks to recognise temporal patterns. We have designed fully connected networks that are trained to emulate and classify sequences by regarding each temporal state of a network as a layer in a feed-forward network. Training is then performed by a variant of the back-propagation algorithm. We have conducted initial experiments using the output of a peripheral auditory model.

INTRODUCTION

In the past few years there has been a resurgence of interest in connectionist systems [1, 2] as pattern recognisers, for tasks requiring parallel constraint satisfaction and as cognitive models. However, the various learning procedures that have been developed, such as back-propagation of error by first-order gradient descent [1, 3, 4] and procedures based around simulated annealing, such as the Boltzmann Machine [5] are limited in that they are methods for training *static* networks: that is the network will undergo a short period of temporal instability before settling into a stable, final state. Such a learning procedure is clearly incompatible with the learning of time-dependent sequences. Hence current connectionist network learning procedures are severely limited to the stable, time-invariant case.

One of the major problems to which connectionist techniques have been applied is that of speech recognition Several studies have been reported of tackling a speech recognition problem using a connectionist system [6, 7, 8, 9, 10, 11]. However these studies are all characterised in that they attempt short time base recognition; they all perform the recognition of short speech patterns that may be presented to the network as a single, static signal. It is clearly essential that learning procedures be developed that enable networks to learn to emulate and classify time-varying signals.

In this paper we present our first steps in tackling the problem of developing learning algorithms for unstable connectionist networks. The basic learning procedure is "back-propagation technology"; however the back-propagation occurs over time [4] and the actual network is fully connected and assymmetric. This time-dependent back propagation learning procedure attempts to train the network to develop abstract time-dependent features which enable it to predict the next time-frame, as these features are likely to be relevant to the classification of the input patterns.

We have carried some initial, exploratory experiments using this learning procedure in attempting to train a network to recognise isolated digits, spoken by various speakers, male and female. The temporal input to the network was generated by a peripheral auditory model.

NETWORK ARCHITECTURE

We are studying fully connected connectionist network models with no restrictions on the weight matrix, such as the symmetry required by Boltzmann machines or the feed-forward structure required by back-propagation. Specifically, every node i outputs a real number $Y_{(t,i)}$ at integral time t according to the law of motion

$$Y_{(t,i)} = f(\sum_j W_{ij} Y_{(t-1,j)})$$

where:

$$f(x) = \frac{1}{1+e^{-x}}$$

and W_{ij} is a real number modelling the synaptic weight from node j to node i. The function f causes the node value $Y_{(t,i)}$ to lie between 0 and 1

The state of the network (the set of node values) varies in time in a way governed by the weight matrix, and at least to some extent the initial state. This law of motion may be overruled for some nodes at certain times by *clamping* nodes, by which is meant overwriting the node value produced by the law of motion with a prescribed value.

We designate some of the nodes as *input* nodes, some as *output* nodes, and the rest as *hidden* nodes. We wish to adjust the weight matrix so that when a temporal sequence of auditory model output, derived from a spoken word is clamped on the input nodes, the network will respond by the time the word has been spoken by setting an output node corresponding to that word to a value near 1, and the other output nodes to 0.

CH2561-9/88/0000-0426 $1.00 © 1988 IEEE

The standard way to use back-propagation to train this network would be to train an equivalent feed-forward network consisting of at least as many layers as there are time steps in the utterances of the words to be classified, back-propagating errors from the output nodes. We have modified this method by giving the network some extra "clues" to assist its training. In order to distinguish words, the network must learn to respond to temporal features in the input which bear more directly than the raw, "time-frozen" input on the distinctions to be cast. We presume that some of these features may also be found amongst features which can help to predict the future evolution of the input signal itself. Therefore we train the network to predict its next input as well as the overall classification of the signal. The network is initialised in a standard state, and trained to predict a return to this state after each utterance, thereby priming itself to respond to another utterance.

LEARNING PROCEDURE

The learning procedure adopted was a variant of the back-propagation of error procedure, in which the time-dependency was simulated by constructing the equivalent strictly-layered, feed-forward network, with each layer corresponding to a time-slice of the input signal.

Let T be defined as the total number of time slices in each training sequence, I be the set of input nodes, H be the set of hidden nodes and O be the set of output nodes. If $i \in I$, $S^p_{(t,i)}$ is the training signal p on node i at time t and if $i \in O$, then $S^p_{(t,i)}$ is the target value for signal p on node i, except when $t = 1$ and $t = T$, at which times $S^p_{(t,i)} = h$, where h is a constant "idle value".

The element of the weight matrix representing the weight from node j to node i is represented by W_{ij}. The weight matrix of the corresponding feedforward network has additional subscripts, representing the timestep in question. So the element of the weight matrix connecting element j at time t, to element i at time $t+1$ is represented as $W_{(t+1,i),(t,j)}$. It should be emphasised that the values of the weight matrix for the feed-forward network do not *depend* on t:

$$W_{(t+1,i),(t,j)} = W_{ij}$$

The network was iterated over the set of signals, p. The $t = 1$ layer of the feed-forward network was initialised:

$$Y^p_{(1,i)} = S^p_{(1,i)} \quad \text{if } i \in I$$

$$Y^p_{(1,i)} = h \qquad \text{if } i \in H \bigcup O$$

That is, the hidden and target nodes are initialised to the idle value.

The net is run for $R-1$ steps, where $R \leq T$ is the level being trained to:

$$Y^p_{(t,i)} = f(\sum_j W_{ij} Y^p_{(t-1,j)})$$

f is the activation function for each node (its equation of motion, in terms of a dynamical systems analysis). In this work f was taken to be the semilinear sigmoid function, the most commonly used activation function:

$$f(x) = \frac{1}{1+e^{-x}}$$

The derivative of f, with respect to x is given by:

$$f'(x) = f(x)\,(1 - f(x))$$

The error is propogated using a modified version of the generalised delta rule: When training to a layer R, the value for δ, the error signal for a node is given by:

$$\delta^p_{(R,j)} = f'^p_{(R,j)}(S^p_{(R,j)} - Y^p_{(R,j)}) \quad \text{if } j \in I \bigcup O$$

The hidden nodes are only targeted for this final layer, T; in this case they are targeted to the idle value, h, as the signal has finished:

$$\delta^p_{(T,j)} = f'^p_{(T,j)}(h - Y^p_{(T,j)}) \qquad \text{if } j \in H$$

When $R \leq T$:

$$\delta^p_{(R,j)} = 0 \qquad \text{if } j \in H$$

For the layers below the current level of training, R, the error signal for each node is given by:

$$\delta^p_{(t,j)} = f'^p_{(t,j)}(\sum_{i \in I \bigcup O} f'^p_{(t+1,i)}(S^p_{(t+1,i)} - Y^p_{(t+1,i)})W_{(t+1,i)(t,j)} + \sum_{i \in H} \delta^p_{(t+1,i)} W_{(t+1,i),(t,j)})$$

The error propagates normally through the hidden nodes, but the input and output nodes are targeted to the signal value and classification pattern for the time step corresponding to that layer.

The change in the weight from node j to node i, ΔW_{ij} is given by the average over t of:

$$\Delta W_{(t+1,i),(t,j)} = \eta \sum_p \delta^p_{(t+1,i)} Y^p_{(t,j)}$$

The training schedule for such a network is simple. First a two layer net is trained, for a few iterations, then a three layer net, and so on until a T layer net, representing the entire signal length, has undergone training for a few iterations. This process of training each feed-forward network for a few iterations is repeated until an error function, E, defined below, reaches a suitable minimum.

$$E = \frac{1}{2}\sum_p (\sum_{t=2}^{T} \sum_{j \in I \bigcup O} (S^p_{(t,j)} - Y^p_{(t,j)})^2 + \sum_{j \in H} (h - Y^p_{(T,j)})^2)$$

When training to level $R < T$:

$$E = \frac{1}{2}\sum_p \sum_{t=2}^{R} \sum_{j \in I \bigcup O} (S^p_{(t,j)} - Y^p_{(t,j)})^2$$

The modified delta rule used here is a first-order gradient descent of this error function. A "momentum" term was also used; this function of the previous weight change is added to the current weight change, and has the effect of damping down oscillations. Hence the expression for change of weights (at the nth iteration) becomes:

$$\Delta W_{(t+1,i),(t,j)}(n) = \eta \sum_p \delta^p_{(t+1,i)} Y^p_{(t,j)} + \alpha \Delta W_{ij}(n-1)$$

Here, α is a constant between 0 and 1 specifying the influence of the momentum term. Generally, α is large, in the region of 0.9.

Changes were also made to the activation function, shifting its output down by 0.5, thus changing its range from (0,1) to (-0.5,0.5). This is done by redefining f as:

$$f(x) = \frac{1}{1+e^{-x}} - \frac{1}{2}$$

EXPERIMENTAL

The network algorithm was simulated in C using a vector-accelerated MassComp 5700 under the AUDLAB interactive speech and signal processing package.

The input signals used to train the model were derived from a band-pass non-linear (BPNL) peripheral auditory model [9], in the form of quantised "auditory spectrograms". These spectrograms are simulated neural firings plotted against time, with frequency measured on the non-linear Bark scale. Additional features representing high and low frequency rms energy have also been used, as the simple auditory spectrogram does not represent energy features (such as the high frequency energy, characteristic of frication) very well. The speech patterns processed by the auditory model were isolated utterances of the English digits "zero" to "ten" spoken by two male and two female speakers with varying accents and at varying rates. This same data set had been previously used successfully in training a static feed-forward network, using the standard back-propagation technique.

These exploratory experiments are in progress at the time of writing: initial results are encouraging, although the minimisation of the error function is slow and local minima and fluctuations are encountered. Work is currently underway to improve the minimisation algorithm by incorporating some form of variable step-size and using second order information.

DISCUSSION

It is apparent that connectionist models offer a promising approach to the problem of automatic speech recognition. However, most previous approaches have failed to map the temporality of the speech signal onto the dynamics of the connectionist network. Those approaches which have attempted to impart temporality into a connectionist network [10, 11] have done so by using specific restricted architectures; in contrast, this work attempts to produce temporal learning algorithms for unconstrained, fully-connected networks.

It is reasonable to suppose that a lack of time-dependence inhibits the possibility of continuous classification or emulation of speech pattern sequences. The work presented here is a first suggestion for developing unstable connectionist networks that can learn to classify and emulate temporal signals. It is premature to say whether this approach has the potential to perform this task; however it is clear that such approaches should be investigated.

REFERENCES

[1] D. E. Rumelhart and J. L. McClelland (1986), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume I and II, MIT Press, Cambridge MA.

[2] J. A. Feldman and D. H. Ballard (1982), *Connectionist Models and Their Properties*, Cognitive Science, **6**, 205 - 254.

[3] D. E. Rumelhart, G. E. Hinton and R. J. Williams (1986), *Learning Representations by Back-Propagating Errors*, Nature, **323**, 533 - 536.

[4] D. E. Rumelhart, G. E. Hinton and R. J. Williams (1986), *Learning Internal Representations by Error Propagation*, in [4], 318 - 362.

[5] D. H. Ackley, G. E. Hinton and T. J. Sejnowski (1985), *A Learning Algorithm for Boltzmann Machines*, Cognitive Science, **9**, 147 - 169.

[6] J. L. Elman and Zipser, D., (1987), *Learning the Hidden Structure of Speech*, ICS Report 8701, University of California, San Diego.

[7] S. M. Peeling, R. K. Moore and M. J. Tomlinson (1986), *The Multi-layer Perceptron as a Tool for Speech Pattern Processing Research*, Proceedings of the Institute of Acoustics, **8(7)**, 307 - 314.

[8] R. W. Prager, T. D. Harrison and F. Fallside (1986), *Boltzmann Machines for Speech Recognition*, Computer Speech and Language, **1**, 3 - 27.

[9] M. Terry, S. Renals, R. Rohwer and J. Harrington (1988), *A Connectionist Approach To Speech Recognition Using Peripheral Auditory Modelling*, This Conference.

[10] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. Lang (1987), *Phoneme Recognition Using Time-Delay Neural Networks*, TR-I-0006, ATR Interpreting Telephony Research Laboratories, Osaka.

[11] R. L. Watrous, L. Shastri and A. H. Waibel (1987), *Learned Phonetic Discrimination Using Connectionist Networks*, Proceedings of European Conference on Speech Technology, Edinburgh, 377 - 380.