# COMPARISON OF NEURAL AND CONVENTIONAL CLASSIFIERS ON A SPEECH RECOGNITION PROBLEM

M. D. Bedworth†, L. Bottou‡, J. S. Bridle†, F. Fallside*, L. Flynn†, F. Fogelman‡, K. M. Ponting†, R. W. Prager*

† Royal Signals and Radar Establishment, UK.
‡ Lab. D'Intelligence Artificielle de Paris V, France.
* Cambridge University, UK.

## 1 Introduction

The term "Neural Networks" has been used for a variety of ideas, most of which have been presented as solutions to problems in Pattern Recognition. It is often difficult to tell whether results presented in Neural Network papers are really significant in practical applications, compared with more conventional methods. We set out to compare several different neural network and other methods on a common dataset relevant to Automatic Speech Recognition. The particular problem chosen (speaker-independent EE-set recognition) is rather specialised, but particularly difficult.

Speaker-independent EE-set word recognition is very difficult, because some of the distinctions are rather subtle and there are large variations due to the speaker. It is also of practical importance, because a spelling mode would be useful in some applications of speech recognition. Even people sometimes resort to spelling out difficult names or other uncommon words. (Most of the errors in alphabet-word recognition are due to the EE-set.) Because the sequential structure of these words is relatively simple we might expect basically static methods (all except the HMMs and TDNN in our case). to have an unfair advantage compared with real speech problems.

Among the studies which have concentrated on speaker independent EE-set recognition there have been impressive results using statistical pattern recognition applied to the output of specially designed feature detectors [1] and HMM systems using discrimination-based training [2]. (The datasets used in these references included the American pronunciation of "Z".)

## 2 Data

The data is a portion of a larger dataset from British Telecom Research Labs (BTRL). It is part of one of the CONNEX datasets which BTRL are preparing for experiments on neural networks. 104 speakers had been asked to say the (British) English names of the letters of the alphabet. We used only the first utterance, by each speaker, of the "EE" words: "B, C, D, E, G, P, T, V" (not "Z", since "Z" is not pronounced with an "ee" in British English). Endpoints of the utterances had been noted using a semi-automatic system. We used only the data between these endpoints. Durations ranged from 260ms to 930ms, with a mean of approximately 470ms. The speakers had been divided into training and test sets, approximately balanced for sex and age. Thus the training and testing sets consisted of approximately 400 words each with different speakers in training and test sets.

Initial acoustic analysis, which was the same for all experiments, used the "SRUbank" filterbank analysis system. This has 27 bandpass filters, followed by squaring, lowpass filters, and downsampling to the "frame rate". Filters 2 to 20 are compatible with the 19 channels of the JSRU Vocoder analysis [3]. Filter 1 is a lowpass filter (zero frequency bandpass). Filters 21 to 27 extend beyond the vocoder's 19 channels from 3.2kHz up to 10kHz. The frequency scale is approximately the same as mel. The lowpass filters are designed to be suitable for frame rates down to 100 frames per second, which is the rate used here.

Most experiments used the first eight cosine components of this spectrum representation, producing a represenation usually known as "mel frequency cepstrum coefficients" (mfccs). One of the mfccs is the average value of the log power spectrum (see, for example, [4]). The HMM methods add up scores for a sequence of frames, and their performance may be improved by augmenting the individual vectors so they can be more sensitive to higher-order properties. Some experiments used augmented frames of mfccs plus differences between values in the previous and next frames ("delta-cepstrum"). For some experiments the data was truncated: only the first 26 frames of data were used. (This is the length of the shortest utterance.)

## 3 Pattern Recognition Systems Used

In order to present a meaningful evaluation of the performance of these "Neural Network" methods we have included standard general purpose methods as well as existing methods specialised for the application area in our comparison.

### 3.1 General-Purpose Methods

- Nearest Class Mean (NCM)

  There is one reference point for each class. Its position is the mean of the training data for the class. We compute Euclidean distances from a test pattern to all the reference points. The class of the closest reference point is the answer. Data was 27 channel or 8 mfccs, truncated. (Reference points and test vectors are $26 * 27$ or $26 * 8$ dimensional)

- k-Nearest Neighbours (kNN)

  Euclidean distances are computed to all 400 training samples. The class with most representatives among the $k$ nearest neighbours wins. Results are produced for various values of $k$, for 27 channels and 8 mfccs, truncated.

### 3.2 Speech-Specific Methods

- 15-state HMM Word Models

  15-state hidden Markov models, with one Gaussian output distributions per state, diagonal covariance matrix estimated for each state. Transitions are to same state (repeat loops) or to next state. Training uses the Forward-Backward algorithm to compute posterior probability distributions, and the Baum-Welch re-estimation to hill-climb on the likelihood of the data for a given class.

In the basic system each observation is an 8 component vector (the 8 mfccs). Two variations and their combination were tried: a) Each observation is a concatenation of the 8 mfccs and 8 differences between values in the previous and the next frame. b) The initial and final states are "tied" so that their parameters are identical (i.e. a silence state), and states 6 to 14 are tied across models, in an attempt to use the knowledge that the same vowel is used in all words.

- Template-style HMMs

  Many dynamic programming template matching word recognition methods can be thought of as particular forms of HMM-based recogniser [5]. Picone [6] suggests using as many states as the average number of frames in the word, and a fixed transition matrix allowing repeats and single-state skips, with probabilities based on the average distances between frames corresponding to the same sound. The most template-like methods tried estimated a mean for each state, and a common diagonal covariance matrix. The recogniser could then use Euclidean distances on weighted acoustic vectors. Results were also produced for the case where a diagonal covariance matrix is estimated for each state separately.

### 3.3 Network Methods

All these networks have 8 outputs (one for each class) and the class with the largest output is taken as the response. The mapping from inputs to outputs is controlled by weights, which are adjusted to minimise the difference between the outputs and the desired outputs. Large enough networks are always able to fit the training data well — the real problem is to obtain good performance on unseen data (the test data in these experiments). For most networks the smaller the size of the network the better the performance on the training data will carry over to test data. It is however necessary to have a certain minimum size of network in order to cope with the scale of the problem. For best results any prior knowledge about the structure of the patterns should be incorporated into the structure of the network. Most of the network methods investigated are without special structure. (For instance the two-dimensionality of set of inputs is lost.) In the Locally connected MLPs used here, we exploit the idea that a set of properties of short segments of the signal should be sufficient. (The same basic idea lies behind the HMM systems too.) The TDNN version also imposes a shift-invariance.

- Radial Basis Function Networks (RBFn)

  A Radial Basis Function Network consists of a nonlinear mapping followed by a linear mapping [8]. The input nonlinear mapping computes functions of the distances between the input vector and a set of reference vectors. These "radial basis functions" are the input to a linear mapping to the output nodes. We used Euclidean distance ($x$) to reference vectors, and an $x^2 \log x$ nonlinearity.

  The simplest way to train an RBFN is to fix the reference vectors (we used a subset of the training points) and solve for output weights which produce minimum mean square error. We randomised the order of the training data and used the first $n$ points as centres, for $n = 10$ to 200 in steps of 10.

- Kanerva Memory Model (KMM)

  The modified Kanerva model [9,10] is a static network consisting of a fixed non-linear mapping (location matching) followed by a single layer of adaptive links. The KMM can be thought of as a radial basis function network with a threshold nonlinearity. The present system used random points in input space for reference vectors.

  The modified Kanerva model experiment was performed with the full 27 element input vectors of filter outputs. Each training pattern was presented in many versions, with offsets from the start of up to 30 frames, in tenths of a frame. In this experimant 9600 location units were used.

  For testing the output of the model was determined by taking the majority vote of the output units as the input window was scanned across the spectral data. About 30 different positions in the input buffer were considered in each case.

- Fully-connected semi-linear logistic multi-layer perceptrons (MLPn)

  A series of "no frills" backpropagation networks [7], with one layer of hidden units. Results were produced for various numbers of hidden units (5 to 40 in steps of 5), using one run for each, with small random starting weights.

- MLP with NCM initialisation (MLPNCM)

  A set of weight values was calculated for a layered logistic MLP network which was functionally equivalent to the nearest class mean classifier. The perpendicular bisectors of each pair of class means were implemented as hidden unit hyperplanes. Those hyperplanes which did not contribute to the NCM decision boundaries were removed. The second layer of weights formed the conjunction of the relevant hidden unit outputs. The network was then trained using the conjugate gradients optimization scheme (an essentially parameterless method). The need for exploratory experiments with different numbers of hidden units and different random weight starts is obviated by this technique (the results are for the first and only run for each data set). The 27 channel data MLP had 18 hidden units. The 8 channel mfcc data MLP had 20 hidden units.

- Multi-layer network with independent local connections (LMLP)

  Most of the results used truncation to 26 frames, but as a check one set of results was obtained using 50 frames of 27 channels. For the 26 frame cases, the input layer is divided into 13 overlapping windows of three frames each. A cluster of 10 hidden units is connected to each window. This first hidden layer is divided into 4 windows, each comprising four sets of 10 units, overlapping by one set. Six cells in the second hidden layer are connected to each of these 4 windows. The second hidden layer is fully connected to an eight cell output layer. During the training we added Gaussian noise to the input data. Both the learning rate and the amplitude of the noise were progressively reduced during training. Results were produced using the 27 channel data, 8 mfccs, 12 mfccs, and 12 mfccs plus 12 time differences.

- Time delay neural network (TDNN)

  These networks are of the same configuration as the LMLPs, but weights which are in equivalent positions at different times in the pattern are "shared" (constrained to have the same values) [11,12]. This forces the lower levels of the network to be shift-invariant, and instantiates the

idea that the absolute time of an event is not important. This general theme of constrained back-propagation has been addressed in [13]. The experimental procedure was the same as for LMLP.

## 4 Results

Here we report only the percentage error rates on the test data (see tables 1–5). For some methods there is an order parameter, which changes the number of degrees of freedom, or alters the amount of smoothing of the input distributions. In these cases we have tried to indicate the range over which the performance is good. Ideally these parameters would be estimated from the available training data, by use of cross-validation or equivalent techniques. With this exception, any parameters of the methods were left in their default settings (performance was not optimised on the test data).

## 5 Discussion

The first thing to note about the results is the poor performance of all the methods. It is conceivable that an automated telephone directory assistance system could make use of a speech front-end with a 15% error rate on the EE-set, but we have no reason to believe that this performance would be maintained in the field.

Although not reported in detail here, the systems differed greatly in the amount of computation and storage needed for training and for recognition. It could be argued that for a speaker-independent word recogniser the training time is irrelevant, as long as it is feasible. On the other hand, any serious attempt at this task would use a *very* much larger training set, from many thousands of speakers.

Five systems had error rates of about 17% (measured on the test set). They were the fully connected MLP started from NCM, the locally-connected MLP, the RBF networks, and two HMMs. The MLPs need about an order of magnitude less computation at recognition time than the other systems. The best result (14%) was by an HMM, which was the most expensive HMM tested. All three best HMM systems used cepstrum differences. For the systems that rely on Euclidean distances (NCM, kNN, RBF), the 8mfccs performed better than the full 27 channel data. In other words, smoothing the log power spectrum removes confusing detail. It is rather surprising that the cepstral inputs perform so badly with the locally connected networks.

As expected, the speech-specific methods (HMM and TDNN) did not show an advantage on this data. This is well segmented data, whose discriminative areas are in predictable positions. However we can say that the large reduction of the number of different weights for the TDNN compared with the LMLP has not reduced the performance much.

This is only a preliminary comparison, and has shown up many conditions which should be tested so that arbitrary differences between methods are minimised. For example, the HMM

| Input | NCM | $k = 1$ | Best range for $k$ | Error range |
|---|---|---|---|---|
| 27ch | 44% | 37% | 8 – 19 | 30% – 32% |
| 8mfcc | 31% | 28% | 4 – 11 | 22% – 24% |

Table 1: Error rates for Nearest Neighbour methods (including NCM)

| | 15 states | | Template Style | |
|---|---|---|---|---|
| Input | Basic | Tied | Common variances | Separate variances |
| 8 mfccs | 37% | 28% | 24% | 25% |
| 8mfccs + diffs | 23% | 17% | 17% | 14% |

Table 2: Error rates for HMM methods

| Input | Best range for $n$ | Error range | Kanerva |
|---|---|---|---|
| 27ch | 70 – 240 | 22% – 24% | 25% |
| 8mfcc | 155 – 230 | 17% – 18% | |

Table 3: Error rates for Radial Basis Function networks

| | Random start | | NCM start | |
|---|---|---|---|---|
| Input | Best range for #HU | Error range | #HU | Error |
| 27ch | | | 20 | 19% |
| 8mfcc | 25 – 40 | 21% 22% | 18 | 17% |

Table 4: Error rates for fully-connected MLPs

| | Independent weights (LMLP) | | Shared weights (TDNN) | |
|---|---|---|---|---|
| Input | num wts | Error | num wts | Error |
| 26*27 ch | 1450 | 17% | 373 | 20% |
| 50*27ch | 2900 | 20% | 457 | 25% |
| 26*8mfcc | 631 | 32% | 316 | 35% |
| 26*12mfcc | 865 | 25% | 328 | 27% |
| 26*(12+12d) | 1333 | 31% | 364 | 35% |

Table 5: Error rates for locally connected MLPs

methods could be tested on the truncated data, and the Euclidean distance based methods (NCM, kNN, RBF) could use the data weighting derived in the template-style HMM (or some other weighting, such as that produced by Linear Discriminant Analysis).

As a reference point we would like to know the performance of human listeners, using both the original waveforms and resynthesised from the parametric representation used in the experiments.

Finally, there are many other interesting kinds of neural network worth trying out on this data, together with hybrids of HMMs and networks.

## 6 Acknowledgements

This collaboration was made possible by funding of the project "Connectionist Approaches to Adaptive Intelligence", under the European Economic Community DG12 programme BRAIN (Basic Research in Adaptive Intelligence and Neurocomputing).

We are grateful to British Telecom Research Laboratories for use of part of their "CONNEX" dataset for this work.

## References

[1] R A Cole et al. Performing fine phonetic distinctions: templates vs features. In J.Perkell and D.H.Klatt, editors, *Invariance and variability of speech processes*, Erlbaum, Hillsdale N.J., 1984.

[2] P F Brown. *The acoustic-modeling problem in automatic speech recognition.* Technical Report RC 12750, IBM TJW Research Center, 1987.

[3] J N Holmes. The JSRU Channel Vocoder. *Proc. IEE*, 127 Pt.F(1):53–60, 1980.

[4] J N Holmes. *Speech Synthesis and Recognition.* van Nostrand Reinhold (UK), 1988.

[5] J S Bridle. Stochastic models and template matching: some important relationships between two apparently different techniques for automatic speech recognition. In *Proc. Inst. of Acoustics Autumn Conf., Windermere*, November 1984.

[6] J Picone. On modelling duration in context in speech recognition. In *Proc. IEEE ICASSP89*, 1989.

[7] D E Rumelhart, G E Hinton, and R J Williams. Learning Internal Representations by Error Propagation. In D E Rumelhart and J L McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations*, MIT Press, 1986.

[8] D Broomhead and D Lowe. *Multi-variable interpolation and adaptive networks.* Technical Report 4148, Royal Signals and Radar Est., 1988.

[9] R W Prager and F Fallside. The Modified Kanerva Model for automatic speech recognition. *Computer Speech & Language*, 3:61–81, 1989.

[10] R W Prager, T J W Clarke, and F Fallside. The Modified Kanerva Model: results for real time word recognition. In *IEE Conf. Artificial Neural Networks*, September 1989.

[11] K Lang and G E Hinton. *The development of TDNN architecture for speech recognition.* Technical Report CMU-CS-88-152, Carnegie-Mellon University, 1988.

[12] A Waibel, T Hanazawa, G Hinton, K Shikano, and K Lang. Phoneme recognition: neural networks vs. hidden Markov models. In *Proc. IEEE ICASSP88*, pages 107–110, 1988.

[13] Y le Cun. A theoretical framework for back propagation. In D Touretzsky, editor, *Connectionist Models: a Summer School*, Morgan-Kaufmann, 1988.