The Use of Vector Quantization in Neural Speech Synthesis

G.C. Cawley and P.D. Noakes
Neural and VLSI Systems Laboratory
Department of Electronic Systems Engineering
University of Essex, UK

Abstract

Our previous work has indicated that multi-layer perceptrons (MLPs) trained using the backpropagation (BP) algorithm, have great difficulty in learning continuous mappings with sufficient accuracy for speech synthesis. The use of vector quantization allows networks to be trained to select a sequence of entries from a codebook of speech parameter vectors. For the network to be able to generalise meaningfully some correlation must exist between codebook vectors and the indices by which they are recalled (otherwise the network will be attempting to learn an essentially random mapping). This paper describes the use of the Hamming learning vector quantizer (H-LVQ), which is used to generate a codebook of speech vectors in which such a correlation exists.

1 Introduction

In our previous work [?, ?, ?, ?, ?], multi-layer perceptrons were trained using the backpropagation algorithm to generate successive frames of speech parameters corresponding to the allophones comprising the utterance. Each allophone is described by a vector of articulatory features. The input layer forms a sliding window over the input, as employed in the NETtalk system [?], and consists of three such vectors corresponding to the current allophone and left and right context allophones. In addition one input neuron is used to indicate the duration of the current allophone, and an index neuron which indicates how much of the current allophone has already been generated. During synthesis a ramp is applied to the index neuron, as the input rises the output neurons step out the speech parameters required to synthesize the current allophone. A number of parametric descriptions of speech based on linear predictive coding (LPC) [?] have been used in an attempt to reduce the sensitivity to the inevitable recall error. Line spectral pair representation (LSP) [?] has been found to produce the best results, but large training times are required to sufficiently reduce the error on the training set, and generalization remains poor [?, ?, ?]. This paper describes the use of vector quantization to replace continuous outputs which must be learned accurately, with binary outputs which represent indices into a codebook of speech parameters.

1.1 Vector Quantization

The vector space described by the LPC coefficients obtained from the analysis of human speech is not populated with uniform density, but instead vectors tend to be concentrated in clusters. Vector quantization [?] is a technique widely used in low bit rate speech coding by which any vector within a cluster is replaced by a reference vector representing the centroid of the cluster (in practice large clusters may be assigned more than one reference vector). A codebook containing the centroids of cluster is stored at both transmitter and receiver. During transmission, each vector is compared with the codebook to find the most similar reference vector, the index of this codebook entry is then transmitted, using typically 10 bits, rather than the vector of speech parameters requiring around 50–70 bits. However, this reduction in bit rate is achieved at the expense of a small increase in spectral distortion.

1.1.1 Kohonen Learning Vector Quantizer

The Kohonen self organising feature map can be used to perform vector quantization [?]. Typically the Kohonen feature map consists of a two dimensional array of linear neurons. During

training the same pattern is presented to the inputs of each neuron, the neuron with the greatest output value is selected as the winner, and its weights updated according to the following rule:

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + \eta[\boldsymbol{x}(t) - \boldsymbol{w}_i(t)]$$

Where: $\boldsymbol{w}_i(t)$ is the weight vector of neuron i at time t, η is the learning rate and $\boldsymbol{x}(t)$ is the training vector.

Those neurons within a given distance, the neighbourhood, of the winning neuron also have their weights adjusted according to the same rule. This procedure is repeated for each pattern in the training set to complete a training cycle or epoch. The size of the neighbourhood is reduced as training progresses. In this way the network generates over many cycles an ordered map of the input space, neurons tending to cluster together where input vectors are clustered, similar input patterns tending to excite neurons in similar areas of the network. When trained, the weight matrix of the network forms a codebook of the vector space described by the training set, which may be used for vector quantization.

2 The Hamming-LVQ Algorithm

We have developed the Hamming learning vector quantizer which is a variation on the standard Kohonen LVQ algorithm. Training patterns are presented and the winning neuron selected as before, but the neurons comprising the neighbourhood of the winning neuron are selected according to the Hamming distance of the neurons index rather than the manhattan distance over a two dimensional array of neurons. This effectively realises a network arranged as an N-dimensional hypercube with a neuron at each vertex. When used for vector quantization such a network produces a codebook in which similar vectors are associated with indices which have similar binary patterns (hence the Hamming LVQ algorithm).

3 Experimental Evaluation of H-LVQ Algorithm

To evaluate the H-LVQ algorithm, conventional LVQ and H-LVQ networks were trained using speech vectors obtained using 10th order LSP analysis of eight sentences taken from the TIMIT corpus [?]. Figures 1 and 2 show graphs of rms error and the sample cross correlation of codebook entries and indices against cycles trained. From these it can be seen that the H-LVQ algorithm forms a similarly accurate mapping of the vector space, but creates a map ordered such that similar codebook vectors are recalled by indices with similar binary patterns. Note that the conventional LVQ network was dimensioned so as to maximise correlation.

4 Vector Quantization in Neural Speech Synthesis

In order for vector quantization to be used in neural speech synthesis, a multi-layer perceptron is trained to recall a sequence of codebook indices instead of generating the speech vectors directly. The same network structure is used as before except the output neurons now have binary targets and it therefore has greater tolerance to learning errors. Four networks were trained, each with 50 hidden layer neurons: (i) a conventional MLP, (ii) an MLP trained to access a randomly ordered codebook, and MLPs trained to access codebooks generated using (iii) the LVQ and (iv) H-LVQ algorithms. Figure 3 shows a graph of RMS error against cycles trained for each network. It can be seen that the use of vector quantization yields better performance, at the expense of the extra computation required to generate the codebook. Also the codebooks which are more correlated are shown to produce the better results.

5 Summary and Conclusions

We have shown that vector quantization can be successfully applied to the outputs of multilayer perceptron networks trained using the back propagation algorithm for speech synthesis. For optimal results the codebook should be ordered such that similar codebook vectors are recalled by indices with similar binary patterns. An additional benefit of vector quantization is that the generation of a codebook requires only raw speech vectors which are easily obtained, unlike the time aligned phonetic transcription required for the supervised training of the MLP.

Acknowledgements

The authors would like to acknowledge the support of the United Kingdom Science and Engineering Research Council (SERC).

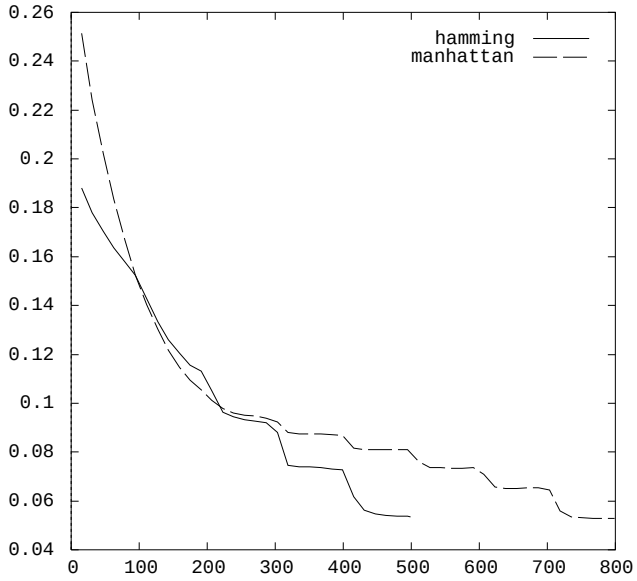

Figure 1: Graph of RMS error against cycles trained for different vector quantizers

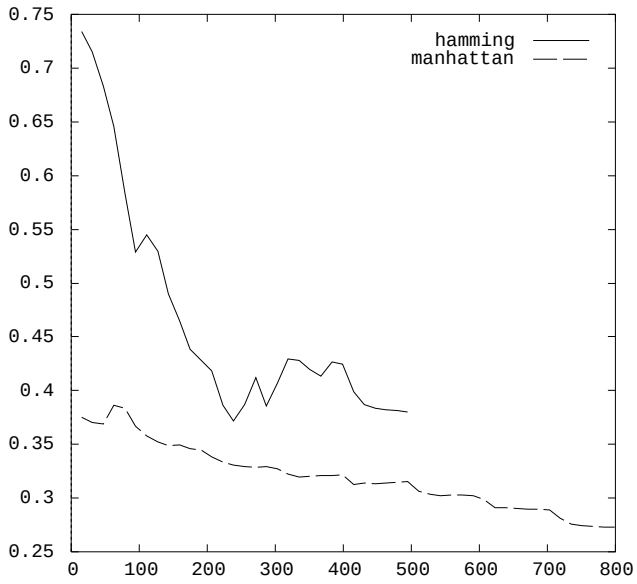

Figure 2: Graph of sample cross correlation between similarity of binary indices and similarity of the corresponding codebook vectors for different vector quantisers

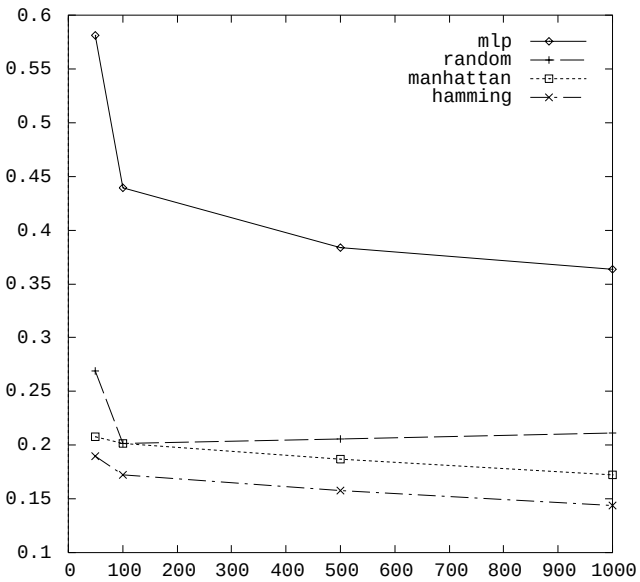

Figure 3: Graph of RMS error against cycles trained for conventional MLP and MLPs trained using vector quantisation