SINGLE LAYER LOOK-UP PERCEPTONS

G.D.Tattersall, S.Foster and P.Linford

University of East Anglia, U.K

INTRODUCTION

In this paper we report a new type of single layer perceptron which incorporates n-tuple pattern recognition techniques [1] in an SLP architecture to produce a *single layer look up perceptron* (SLLUP) which can learn the same types of non-linear mappings as an multi-layer perceptron, MLP, but with a fraction of the training and computation. An additional very desirable property of the SLLUP is that it produces a quadratic error surface and so convergence to optimal performance is assured.

It will be argued that the SLLUP is basically an interpolation system which is able to generate an estimate of a continuous mapping function from a sparse set of training examples and, as will be demonstrated, is well suited to dealing with simple non-linear mappings such as parity detection.

The ability of the SLLUP to work on the very complex mapping problems of speech recognition and text to speech synthesis is also examined and compared with the performance obtainable using the multi-layer perceptron. It will be seen that the SLLUP can very nearly equal the performance of the MLP in these problems, suggesting that the MLP also does little more than a straightforward sample interpolation.

SLLUP ARCHITECTURE AND OPERATION

All neural networks can be thought of as vector transformers in which the transformation is learnt. The learning is normally supervised as depicted in Fig 1. A training example of an input vector is applied to the system and a target output vector is shown to the system by the supervisor. The difference between the actual output and the target is used to modify the internal parameters of the system so that the actual output becomes more like the target.

In the case of the SLLUP, the vector transformer has the form shown in figure 2. The input vector X is encoded as an image of black and white pixels formed by bits of the code representing the scalar elements of X. The image is placed in a retina onto which random connections are made in groups of n to form n-tuples which are used to address a large number of RAMs. The RAMs themselves are grouped into units and the outputs of all the RAMs in the ith unit are added to form, y_i, the value of the i th element of the output vector Y. The bits used to represent the elements of X in the retina can range from themometer code to natural binary or Gray code depending on the desired SLLUP characteristics.

The system is trained by applying a vector X to its input. This causes a specific set of n-tuple addresses to be generated, which access corresponding contents in each of the RAMs. The summation of the output of each group of RAMs produces the elements of the output vector Y. This vector is compared with the desired output T and the error vector, E, is used to modify the values of the currently addressed RAM locations so that next time the same input vector is applied, the output, Y, is nearer to the desired output T.

Repeated application of different training vectors allows the system to learn the required input - output mapping Y= f(X).Moreover, appropriate choice of n-tuple order and number of RAMs in each neuron block, enables the system to estimate the best function f(X) to fit a rather sparse training set.

Analysis of the SLLUP Learning Procedure.

The adaption of the RAM contents to develop the required mapping function is done using a gradient descent procedure. The required error gradient is easily shown to be given by equation 1 in which $C_{ij}(X)$ is the content of the location in the jth RAM in the ith neuron block which is addressed via its n-tuple connections by the pattern X. y_i , t_i , and e_i are the actual output ,target output and output error respectively.

$$\frac{\partial \overline{E^2}}{\partial C_{ij}^{n}(X)} = 2.(y_i - t_i) = 2e_i \quad \ldots\ldots\ldots\ldots \quad 1$$

This gradient expression can be used to define a simple steepest descent training algorithm in which the contents of the currently addressed RAM locations are updated by subtracting a fraction of the error between actual and target outputs of the neuron block.

$$C_{ij}^{n+1}(X) = C_{ij}^{n}(X) + k.(y_i - t_i) \quad \ldots\ldots\ldots\ldots \quad 2$$

As well as this very simple training procedure, the SLLUP has an other important advantage compared to other neural networks such as the MLP: examination of the expression for the error surface shown in equation 3 indicates that it is quadratic in Cij (X) which means that convergence to a single global minimum is guaranteed.

$$\overline{E^2} = \frac{1}{N}\sum_{i=1}^{N}\left\{\sum_{j=1}^{Q} C_{ij}(\bar{x}) - t_i\right\}^2 \quad \ldots\ldots\ldots\ldots \quad 3$$

Learning a Mapping Function & Interpolation.

Supervised learning machines are required to learning a mapping function Y = f (X) without exposure to all possible input - output pairs of Y and X values. This is only possible if

the system can *interpolate* between sample values of the function which are given during training. A straightforward approach to generating a continuous interpolated function from a set of discrete samples of the function is to convolve the samples with a suitable low pass filter response or kernel function. This is approximately the function performed by the SLLUP [2], [3] which implicitly generates a low pass filter kernel function having a form governed by the order of the n-tuple connections, n, the number of dimensions, D, of the input pattern space and the way in which the input pattern is encoded in the input retina. If a 'thermometer code' is used for each of the input pattern elements, the kernel function is defined by equation 4 in which s(x) is the value of the kernel at a distance x from its centre and W is the width of each input space dimension.

$$s(x) = t.(1 - e^{\frac{-n.|x|}{W.D}}) \quad \ldots\ldots\ldots\ldots \quad 4$$

Three important points are raised by this view of the generation of a continuous function from discrete examples.

1. Sufficient training examples must be given such that there is an average of two samples per cycle of the highest frequency in the mapping function.i.e The function must be sampled at the Nyquist Rate. This suggests that the complexity of a mapping function be specified in terms of its bandwidth, B.

2. To obtain a perfect, continuous function from the training examples, the interpolation filter should have a rectangular frequency response, cutting off at a frequency of B. However, practical filters will never have this response and will give an error in the estimate of the continuous function.

3. It is very unlikely that the training examples supplied to the SLLUP will be uniformly distributed across the pattern space. This means that the distances between samples of the required mapping function are non uniform. The Nyquist Sampling Theorem requires at least two samples per cycle of the function if it is to be recovered without loss of information. However, a uniform sample interval is not specified and the irregularity of the training points does not necessarily mean that the continuous mapping function cannot be recovered. Unfortunately, a simple interpolation filter is unable to recover a continuous function from a set of irregular samples because the function will be non-uniformly scaled in proportion to the density of the samples. However this is not a problem, if the learning is done iteratively, because incorrect scaling of the function will be corrected by the negative feedback action of the adaption algorithm.

The SLLUP Interpolation Kernel.

The form of interpolation kernel produced by the SLLUP depends strongly on the way in which the input vector X is encoded. Thermometer coding is the simplest way of encoding X in 'image' form which can be sampled by the random n-tuple connections. This type of input coding produces the smooth kernel defined in equation 4 whose 1/e width and hence bandwidth is simply controlled by choosing the appropriate n-tuple order. Unfortunately, this technique makes inefficient use of memory because one pixel must be provided for every increment along every dimension of X.

An alternative coding is natural binary which leads to a minimal retina size, and hence memory. However, it produces a kernel consisting of a central impulse surrounded by lower level impulses. and the effective bandwidth of this kernel function is much higher than the thermometer code function. Consequently, many more training examples are required for it to synthesise a smooth mapping function. Many other coding schemes can be devised which yield kernels which lie between the extremes of form generated by the thermometer and natural binary codes and these are discussed in [3].

LEARNING A FUNCTION - THE FUZZY EXOR PROBLEM.

A series of experiments have been done on a SLLUP to test it ability to synthesize simple mapping functions and to see how much training is required and illustrate the operation of the SLLUP.

Just one classical example is presented here: a fuzzy exclusive OR as shown in figure 3. The SLLUP is required to map any patterns lying within the two rectangles, marked C_1 in the input space, to the single point marked C_1 is in the output space. Similarly any pattern in the C_2 region of the input space should map through to the point C_2 in the output space.

To test the accuracy of the mapping learnt by the SLLUP, an error function has been defined where $e(x_1, x_2)$ is the Euclidean distance between the target output and actual output when an input pattern $[x_1\ x_2]$ is applied to the machine.

As an example, Fig 4 shows the error function for C1 after the SLLUP has been trained on one point in the centre of each region. The resultant mapping closely approximates the specified function even though the system has only been trained on a very small subset of possible input patterns.

SLLUP Learning Times.

Fig. 5 shows the time variation of the RMS error between the output of a SLLUP and the target values given during training on the fuzzy EXOR problem described earlier in the paper. In this experiment only four distinct input patterns have been used for training in each of the class regions shown in Fig 3. The SLLUP converges to low error value *at the training points* after only 50 iterations although the value of error with inputs other than those used during training is of course much higher.

For comparison, the learning curves of a 2-layer 4 hidden unit MLP are also shown in Fig 5, although direct comparison of the MLP and SLLUP learning times is rather difficult because it depends on the number of hidden units used in the MLP. The minimum number of hidden units depends in turn on the complexity of the mapping function to be generated whereas the SLLUP complexity does not depend directly on the function complexity. Four hidden units was chosen in this case because it is more than sufficient to deal with the fuzzy EXOR problem.

SPEECH RECOGNITION AND SYNTHESIS PROBLEMS USING THE SLLUP.

It has been shown in previous sections that

the SLLUP is able to perform simple non linear mappings such as the fuzzy EXOR problem. This is achieved using only small amounts of training and with little computation compared to the MLP. We now examine the SLLUP performance on two speech mapping problems of very great complexity on which MLPs and some other neural nets have already been tested.

The first mapping problem is speaker independent recognition of utterances of the letters of the alphabet. A defined cepstral coefficient representation of many utterances of the letters of the alphabet from one large set of talkers must be classified by the SLLUP after it has been trained on examples from a separate set of talkers. The database used in this experiment was compliled by British Telecom Research Labs and is Known as the CONNEX S1 data [5]. The SLLUP is trained on approximately 4000 utterances from a balanced mix of 52 talkers and then tested on approximately 4000 utterances from another 52 talkers. The utterance length is normalised by linear time warping and is presented to the SLLUP as a set of 15 frames of 8 Mel Cepstral coefficients.

The second complex problem to which the SLLUP has been applied is text to speech synthesis. In this case orthographic text has to be mapped to a sequence of phoneme codes which are then used to drive a hardware synthesiser. The experiment uses the same database as NETSPEAK [4] and is identical in all respects except that the MLP is replaced by a SLLUP. The SLLUP is presented with a character taken from English orthographic text and has to produce an appropriate phoneme code as output. Clearly the pronounciation of a particular character often depends on the word in which it is embedded and so 3 characters on either side of the target character are simultaneously presented to the SLLUP. Thus, the complete input pattern consists of a context window of 7 characters, each encoded using 11 bits. The output phoneme is represented using a 19 bit code to represent each of 55 phonemes.

It is interesting to consider the types of mapping which the SLLUP has to develop to deal with each of these two problems. In the speech recognition case, input patterns belonging to the same utterance class are likely to cluster together in their N-space and the SLLUP has to map the region of N-space enclosing the cluster to a single specified point in the output space. There may be several clusters belonging to one class but overall the mapping between input and output is smooth without abrupt transitions. This proposition is supported by the fact that moderately good speech recognition systems can be made using nearest neighbour classification of the input pattern. The task of the SLLUP in this case is to *interpolate* so that previously unseen input patterns which lie between training examples of the same class are mapped to the same output code.

The text to speech mapping is very different. The distances between the codes representing different characters does not have a simple relationship to the distances between the codes for the corresponding output phoneme codes. In other words, the patterns are really symbolic and just happen to be represented in a Euclidean space for processing by the neural net. Thus, the task of the SLLUP is to detect any *logical* structure in the data and failing this, to act as a look up table.

Experimental Results on The Speech Recognition Problem.

Tables 1 to 3 summarise the performance of the SLLUP as a speech recogniser. All the results were obtained after only 8000 training iterations. i.e Exposing the SLLUP to the training set twice. Table 1 shows that a SLLUP using natural binary coding in the retina is able to learn the training set very well, but performs poorly on the test set. Moreover, the performance tends to improve as the order of n-tuple decreases. Taken together, these two factors suggest that the SLLUP is unable to interpolate sufficiently betwen the training examples because the effective width of the interpolation kernel is too small. Reduction of the n-tuple order causes the kernel to become wider, with a consequent improvement in performance on the test set. Increasing the order makes the system behave more like a look up table, giving better recognition of the training set but an inability to generalise.

8 Bit Natural Binary Coding		
N-tuple Order	Training set	Test set
2	95%	65%
3	98%	62%
4	95%	52%

Table 1.
Speech Recognition Using Natural Binary Code.

The kernel width produced using natural binary code is very narrow and so a possible solution to the poor test set performance is to use a different code in the retina as demonstrated by the results of Table 2. These results were obtained by quantising each of the cepstral coefficients to 8 levels and representing them by thermometer code. As expected the performance improves on the test set and gets worse on the training set. This confirms our hypothesis that the natural binary code leads to an over specific system. The results in Table 2 show an improvement in performance as the n-tuple order increases, indicating that in this system,the kernel is actually too wide so that with low values of n, over generalisation is taking place. This is supported by the fact that the system has been unable to accurately recognise the training set.

8-Level Barchart Coding		
N-tuple Order	Training Set	Test Set
2	69%	66%
3	74%	69%
4	76%	70%

Table 2. Speech Recognition Using 8 Level Thermometer Code.

An additional factor which possibly reduces the system accuracy is the very coarse quantisation and this is born out by the improved results in Table 3 for a system

using thermometer code with 16 levels per coefficient. The recognition accuracy obtained using this system with n=6 is comparable with results obtained using a 2 layer, 25 hidden unit MLP on the same data which produced a test set accuracy of 81%.

16 Level Thermometer Code		
Tuple Order	Training set	Test set
2	76%	71%
3	81%	75%
4	83%	77%
6	85%	78%

Table 3. Speech Recognition Using 16 Level Thermometer Code.

Experimental Results On The Text to Speech Synthesis Problem.

In these experiments each of the seven characters in the input window are represented by 11 bit codes containing approximately equal numbers of '1's and '0's. This is important when using a SLLUP because an imbalance in the number of'1's and '0's will cause most n-tuple values to always consist of n '1's or n '0's respectively and this renders the n-tuple values insensitive to changes in the input vector X.

The results obtained using a SLLUP in the text to speech application are presented in Tables 4 and 5. Table 4 shows the performance of the SLLUP when the 11 bit codes are placed at approximately equidistant positions in 11-space. This coding is therefore completely unstructured. As expected, the performance is very poor because the input patterns are really symbolic and the interpolation between arbitrary codes effected by the SLLUP is inappropriate. Using a high n-tuple order of 8 gives improved performance on the training set because the SLLUP starts to operate as a look up table. However, the test set performance remains poor.

Coding Codes for each character are approximately equidistant are each 11 bits long consisting of 5 '1's and 6 '0's.		
Tuple Order	Training Set	Test set
4	34.5%	33.4%
8	60.8%	55.2%
Training: 4 blocks of 10,000 characters Testing: 1 block of 10,000 characters		

Table 4. Text to Speech Sythesis Results - Unstructured Codes.

Table 5 shows the performance of the SLLUP working on a modified set of input codes which are chosen so that their mutual distances approximately reflect the perceptual distances between the phonemes which map most frequently to each letter. Using these structured codes, distances in the 11-space have some meaning and so interpolation becomes a more appropriate means of generating an output on unseen input data. Predictably the results in Table 5 are much better, with high accuracies obtained both on training and test data if sufficiently large n-tuples are used. A furthur improvement can be obtained if the frequency of commonly occuring words is relected in the content of the training and test sets. This is because the very common words in English often have irregular pronunciation rules which are hard for the SLLUP to learn unless seen very frequently. McCulloch reports [4] that a 2-layer, 77 hidden unit MLP can give an 86% letter to phoneme mapping accuracy which is slightly better than the SLLUP result. However, the SLLUP converges relatively quickly and shows a trend of improving performance as n-tuple order increases.

Coding Each code is 11 bits consisting of 5 '1's and 6 '0's. Distance between each code reflects the letter group.		
Tuple Order	Training Set	Test set
4	52.2%	52.2%
8	72.7%	71.3%
10	78.4%	75.0%
10 **	83.9%	79.9%
Training: 8 blocks of 10,000 characters **Testing:** 5 blocks of 10,000 characters **frequency weighted training and test data		

Table 5. Text to Speech Synthesis Results - Structured Codes.

CONCLUSIONS

It has been argued that the purpose of any supervised learning network is to synthesise a continuous non-linear mapping function from a sparse set of training examples of the function. The continuous function can be generated by *interpolation* between the discrete examples of the function. An important implication of this argument is that the number of training examples must be sufficient such that the function is sampled at least at the Nyquist rate.

SLLUPs synthesise the required mapping function by effectively convolving the discrete training function samples with a kernel function which is analogous to the impulse response of a low pass interpolation filter.

The SLLUP uses comparable amounts of memory to the MLP for all but the most trivial functions and in general will learn the required mapping function much faster than an MLP because it is a single layer machine in which error gradients used for its adaption can be calculated directly from the output error. Moreover, because it is a single layer machine, the error surface for the SLLUP is and therefore always converges to a minimum error.

It has been shown that the SLLUP is able to operate as a speaker independent recognicer

with almost as high accuracy as an MLP which suggests both that speech recognition can be effectively performed by interpolation and, perhaps more important, that the MLP also appears to be doing little more than interpolation. This is supported by the use of a SLLUP for text to speech synthesis which again gave a performance only slightly inferior to an MLP.

Acknowledgement.

Acknowledgement is made to the director of British Telecom Research Labs for supporting this work and providing the CONNEX S1 and NETSPEAK databases.

REFERENCES

[1] W.W.Bledsoe and I. Browning. 1959. Pattern Recognition and Reading by Machine. Proc. Eastern Joint Comp.Conf. Boston, Mass.

[2] G.D.Tattersall and R.D.Johnston.1984. Speech Recognisers Based on N-tuple Sampling. Proc.Institute of Acoustics.Spring Conf.1984.

[3] G.D.Tattersall. Single Layer Look Up perceptrons. To be published in British Telecom Technology Journal. Autumn 1989.

[4] N. McCulloch, W.A. Ainsworth, R.Linggard. Multi-Layer Perceptrons applied to Speech Technology.. British Telecom Technology Journal. Vol.6 No.2 April 1988.

[5] R. Linggard and P. Woodland. CONNEX S1 Database. British Telecom Research Labs Ipswich IP5 7RE.

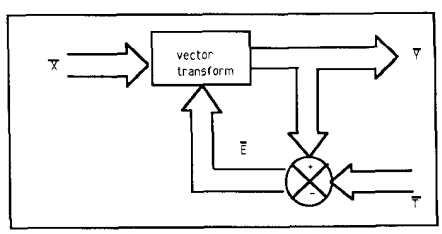

Fig 1. Supervised Learning System.

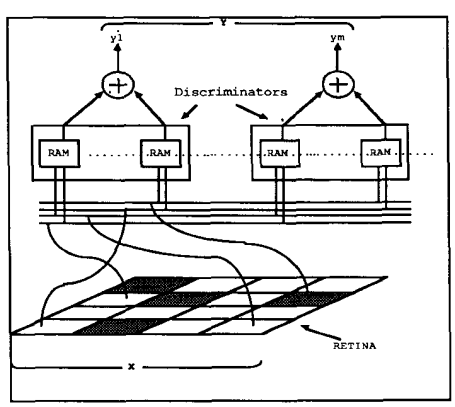

Fig 2. SLLUP Architecture

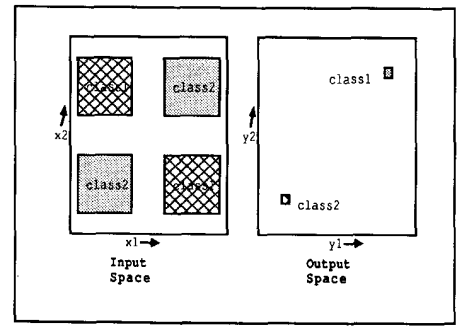

Fig 3. The Fuzzy EXOR Mapping.

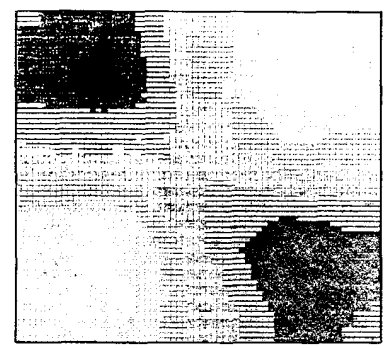

Fig 4. Class 1 Error Function for EXOR Mapping.

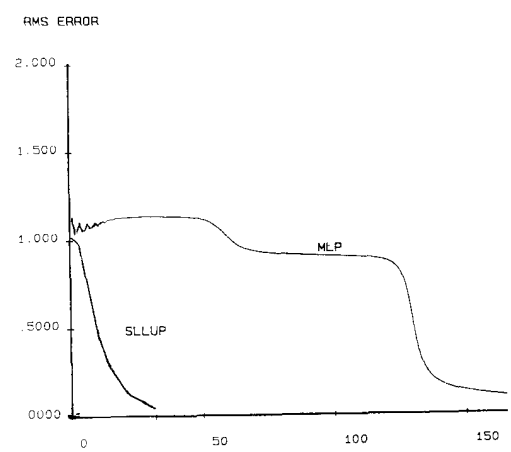

Fig 5. Learning Times For SLLUP and MLP on EXOR.