ANALYSIS OF THE INTERNAL REPRESENTATIONS IN NEURAL NETWORKS FOR MACHINE INTELLIGENCE

Lai-Wan CHAN
Computer Science Department
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
email : lwchan@cucsd.cuhk.hk (bitnet)

Abstract

The internal representation of the training patterns of multi-layer perceptrons was examined and we demonstrated that the connection weights between layers are effectively transforming the representation format of the information from one layer to another one in a meaningful way. The internal code, which can be in analog or binary form, is found to be dependent on a number of factors, including the choice of an appropriate representation of the training patterns, the similarities between the patterns as well as the network structure; *i.e.* the number of hidden layers and the number of hidden units in each layer.

1 Introduction

In supervised neural networks, such as multi-layer perceptrons [Rumelhart, Hinton & Williams 1986], information is acquired by presenting some training examples to the network in the training process. These examples are pairs of input and output patterns. A set of connection weights is then found iteratively using the generalised delta rule and it is reserved for the classification process in the recalling phase. At present, there is no explicit guide-lines for both the choice of the size of the network and the representation format of the training examples. Trial and error has been used to decide the number of hidden layers and the number hidden units in each layer. Previous studies have shown that the number of hidden units in a multi-layer perceptron affects the performance of the network. For examples, the convergence speed and the recognition rate vary with the number of hidden units [Burr 1988]. In this paper, the approach of regarding the hidden layers as the transformation process in the hyper-space were used. We illustrate that a back propagation network with internal layers solves some classification problem intelligently by using this transformation idea. In addition, we show that the internal representation of information can be affected by some characteristics of the training patterns and the architecture of the network.

2 Method of analysis

In previous studies, it has been pointed out that the multi-layer perceptron networks store information distributively [Rumelhart & McClelland 1986]. The distribution of information may be uneven among the hidden units, thus, some hidden units may be more important than others and some may carry no information at all. In this respect, care has to be taken to choose the appropriate hidden units for studying. In our study examples, we used low dimensional hidden space so that all information has to be packed in limited dimensions and the distributive representation of information can be avoided. All experiments described in this section involved training a network to perform a particular task until the total error dropped below 0.1%. The relation between the states of the hidden units and the training patterns was studied and displayed in the form of diagrams. From these diagrams, we are able to visualize how the training patterns are transformed and encoded in the hidden layer due to the connection weights. This also enables us to tell the distribution of the encoded information in the hidden space. This training procedure was repeated with different initial settings to exhaust all other possibilities of the hidden space patterns and to check the reproducibility of the results. The initial settings include the random initial weights, the gradient and the momentum coefficients and the use of other training algorithms such as the

Training patterns							
Input States				Output States			
1	0	0	0	1	0	0	0
0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1

Table 1: The training patterns used in the encoder problem.

adaptive training [Chan & Fallside 1987].

3 The encoder case

3.1 The 4-2-4 encoder

We started the investigation with the 4-2-4 encoder problem [Hinton, Sejnowski & Ackley 1984]. A three-layered network in which the input and output units have four units each and a hidden layer with two units was used. Table 1 shows the input and target states of the training patterns. After successful training process, the resulting hidden space pattern of this 4-2-4 encoder is shown in Fig. 1. Each axis of the space represents the state value of one of the hidden unit and the crosses indicate the values of the hidden units when one of the training patterns is presented to the network. Similar diagrams are obtained when we repeated the experiment with different initial settings. The hidden space pattern indicates that the hidden units encoded the four input patterns in a more or less binary fashion and this encoding schema is the same as what we expected from the traditional encoding method in a multiplexing system.

3.2 The $2^n - n - 2^n$ encoder

When the 4-2-4 encoder problem was further extended into $2^n - n - 2^n$ encoder problems, the hidden space pattern distribution is observed to be quite differently. Theoretically, the back-propagation network is able to encode the training patterns into a binary representation as in the case of 4-2-4 encoder problem. However, as n increases, the hidden space becomes less likely to encode patterns in a binary-valued representation. The total error of the network during the training process dropped to a very low value without the formation of a binary coding system inside the hidden space. Fig. 2 shows the hidden space patterns of the 8-3-8 encoder problem with different initial conditions. The binary-valued pattern shown in Fig. 2a had the highest occurrence whereas the patterns in Fig. 2b and Fig. 2c, where the locations of the patterns had shifted from the corners of the space to the edges, appeared with a much lower probability. In the 16-4-16 encoder problem, almost none of the trials gave a binary form representation in the hidden space. This demonstrates that a back propagation network can provide encoding schema in both binary and analog forms in the hidden space. This schema is not unique and depends on the initial setting of the training parameters. One suggested explanation of this phenomenon is that the number of hidden units has been increased and therefore, the training patterns have more freedom to be distributed in the hidden space. In the next section, we show that the analog encoding schema of the multi-layer perceptron is depending on the architecture of the network.

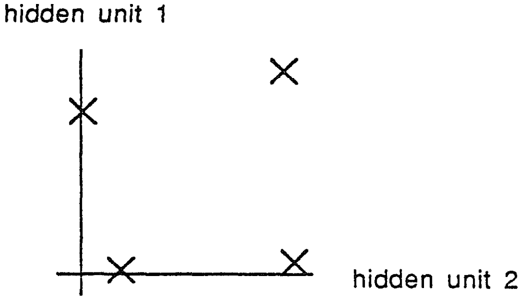


Figure 1: The hidden space of 4-2-4 encoder problem. The x and y axes show the ranges of the two hidden states from 0 to 1 respectively. The hidden states of the four training patterns are represented by the four crosses.

4 The bottle-neck problem

The previous section showed that information can be encoded in either binary or analog manner in the hidden layer. We are now concentrating on the study of the internal representation using analog values. In order to force the use of analog representation in the hidden space, we constructed networks to encode more than four patterns in two hidden units.

4.1 One hidden layer

A n-2-n network (*i.e.* a network with n input units, 2 hidden units and n output units) was used to encode n training patterns. In this section, the orthogonal training patterns used in Section 3 were used and they were extended to n orthogonal patterns in

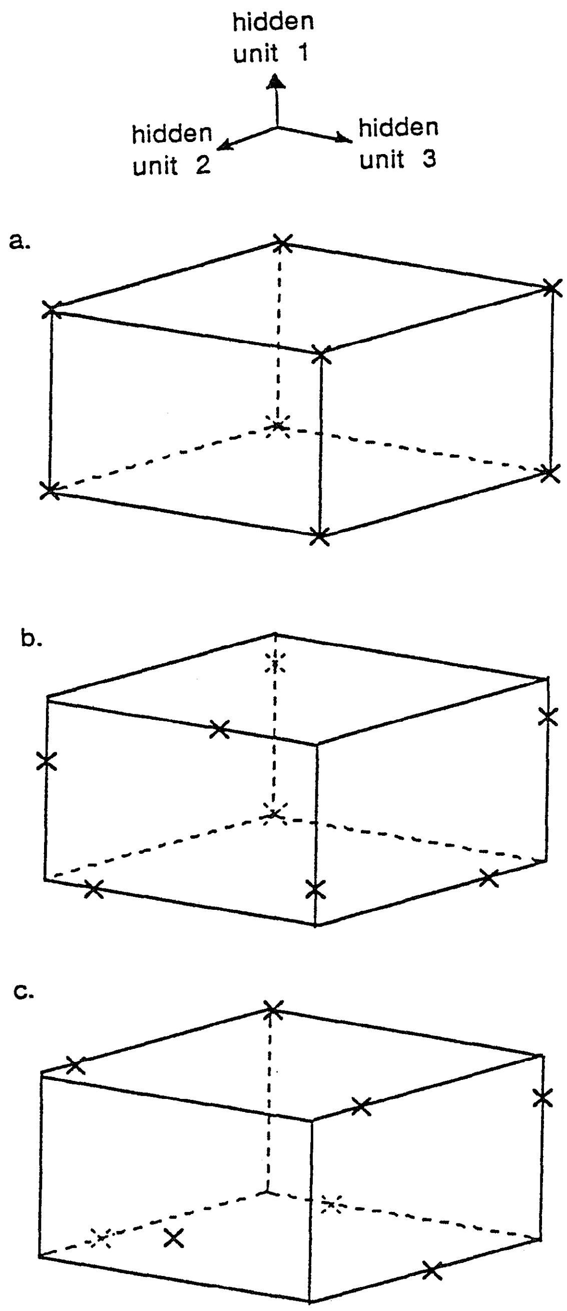


Figure 2: Three hidden spaces of the 8-3-8 encoder problem. The x, y and z coordinates of the cubes are the ranges of three hidden units respectively. The encoding of the eight training patterns in the hidden space are represented by eight crosses in each cube.

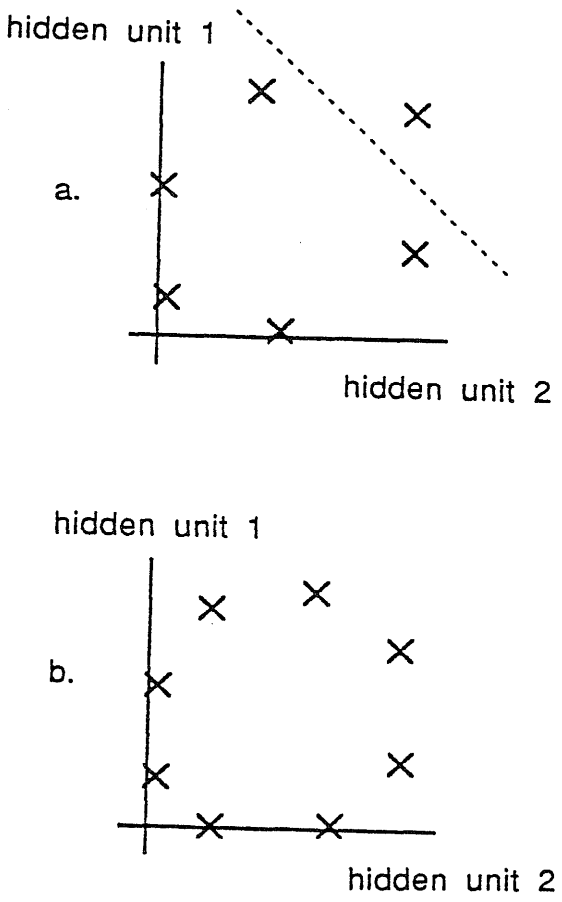


Figure 3: The hidden space pattern of (a) the 6-2-6 encoder problem and (b) the 8-2-8 encoder problem.

n-dimensions. These training patterns forced the hidden units to encode the binary input and output in an analog manner. Experimental results shown that the patterns in the hidden layer were arranged in an orderly manner and were lying approximately on a circle (Fig. 3). This particular arrangement has a special feature; each cross can be separated from the others by a single straight line and this enables each pattern be distinguished from the others by using the decision boundary in a 2-dimensional space. Therefore, the presence of the hidden layer provides the transformation of the input patterns into other dimensional space which enables the classification to take place more easily.

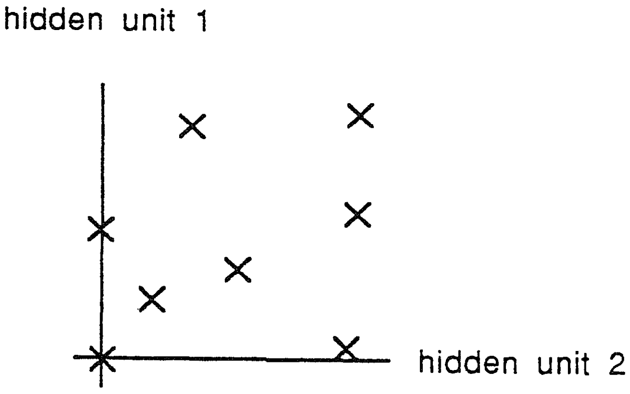


Figure 4: The middle layer hidden space of the 8-4-2-4-8 encoder problem.

4.2 More hidden layers

A different hidden space pattern was obtained if more hidden layers were included in the back propagation network. A network with 8 input units, 3 hidden layers with 4 units, 2 units and 4 units respectively and an output layer with 8 output units was used to encode 8 orthogonal patterns. Fig. 4 shows the hidden space patterns of the middle hidden layer with two hidden units. Patterns were no longer arranged in a circle as seen in Fig. 3b, but spread all over the space. These patterns were organised in a different way because the presence of extra hidden layers in between this middle hidden layer and the input/output layers brought an extra transformation between them. Again, the extra transformation allows the network to have more freedom to form the coding methodology.

4.3 Non-orthogonal patterns

The above hidden space patterns were not obtainable if we altered the training patterns. When a different set of training patterns was used, the resultant hidden space diagram was different. In this section, we study the encoding property of a 4-2-4 network using non-orthogonal training patterns. Table 2 shows six non-orthogonal patterns used in the experiments. When any four of the patterns were chosen and included in the training domain, the hidden space arrangement was the same as that of the orthogonal 4-2-4 encoder problem shown in Fig. 2. With the addition of an extra training pattern, the network was able to encode all training patterns successfully and this training process took about 254 cycles. The resultant hidden space pattern of this network is shown in Fig. 5. It can be seen that one of the training patterns no longer resides on a circle but locates at the centre of the space. When six input patterns were presented, the network was unable to encode all patterns successfully.

Training patterns							
Input States				Output States			
1	1	0	0	1	1	0	0
0	1	1	0	0	1	1	0
0	0	1	1	0	0	1	1
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
1	0	0	1	1	0	0	1

Table 2: The non-orthogonal training patterns.

The difference between the arrangements of the patterns in the hidden space is suggested to be due to the use of a different set of training patterns. In the experiments using orthogonal patterns, only one unit is active in each pattern whereas in the non-orthogonal case, more than one units are active. In the latter experiment, the connection weights were found to form the decision boundaries as shown in Fig. 5a. This allows the patterns be separated by the decision boundaries and satisfied the the input/output requirements as stated in the training sets.

5 Conclusions and Discussions

We can deduce from the above experiments that back propagation networks classify different input patterns by transforming the pattern in a layer to another location in the hidden space of a higher layer. The rule of the transformation depends on the distribution of training patterns. The number of hidden layers can affect the arrangements of the training patterns in each hidden layer. For patterns with both distinct input and distinct output states, their locations in the hidden space are far apart so that their separations are large enough for discrimination to take place. Their distributions depend on the number of hidden layers and the distribution of training input and output states. On the other hand, for patterns with different input patterns but the same output states, their hidden space patterns are more difficult to describe. The network will organise itself in such a way that these patterns are grouped together so that they can be separated from the others by hyperplanes in the higher layers, *e.g.* the XOR problem

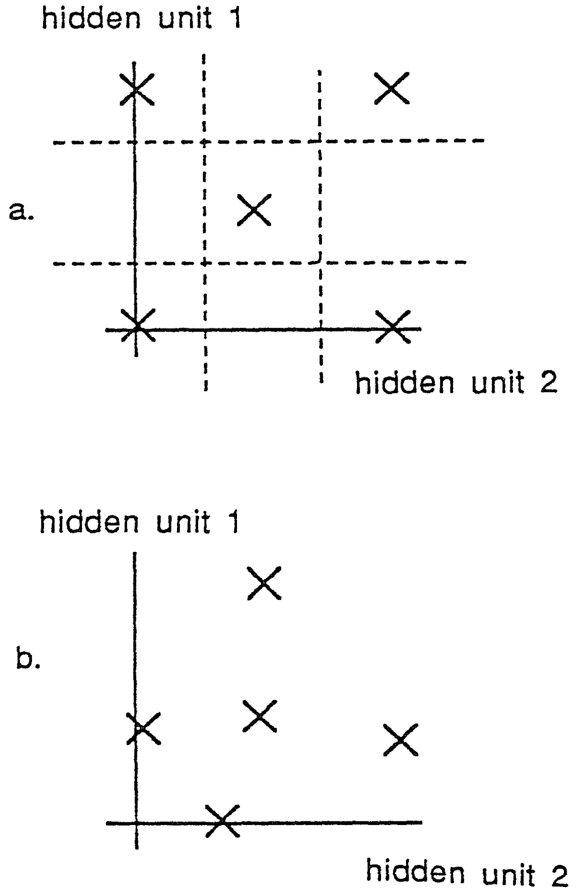


Figure 5: The hidden space patterns of the 4-2-4 network when five non-orthogonal training patterns were used.

Fig. 6 [Rumelhart & McClelland 1986].

Acknowledgement. Part of the work in this paper was carried out in Cambridge University Engineering Department and the author thanks Prof. Frank Fallside for his guidance during that period.

References

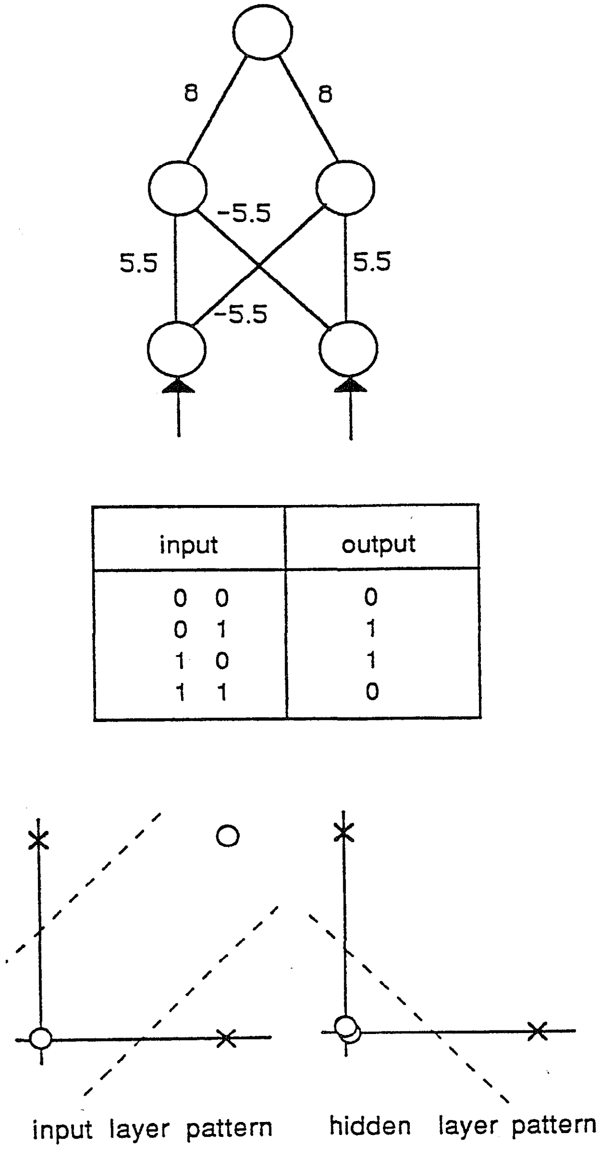
[Burr 1988] Burr, D.J. 1988. Experiments on Neural Net Recognition of Spoken and Written Text. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1162-1168.

[Chan & Fallside 1987] Chan, L-W. and Fallside, F. 1987. An Adaptive Training Algorithm for Back Propagation Networks. *Computer Speech and Language*, 2:205-218.

[Hinton, Sejnowski & Ackley 1984] Hinton, G.E.; Sejnowski, T.J. and Ackley, D.H. 1984. Boltzmann Machine: Constraint satisfaction networks that learn. Technical Report: CMU-CS-84-119, Carnegie-Mellon University.

[Rumelhart & McClelland 1986] Rumelhart, D.E. and McClelland, J.L. eds. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Vol. 1:Foundations, Bradford Books/MIT Press.

[Rumelhart, Hinton & Williams 1986] Rumelhart, D.E.; Hinton, G.E. and Williams, R.J. 1986. Learning Internal Representations By Error Propagation, in *Parallel Distributed Processing: Explorations in the Microstructure of cognition*, Vol. 1, ed. by Rumelhart D.E. & McClelland J.L. Bradford Books/MIT Press.



input	output
0 0	0
0 1	1
1 0	1
1 1	0

Figure 6: The XOR problem.