INCREASING CLASSIFICATION ACCURACY USING MULTIPLE NEURAL NETWORK SCHEMES

George N. Bebis[1], *Michael Georgiopoulos*[1],

George M. Papadourakis[2], *and Gregory L. Heileman*[3]

Department of Electrical Engineering, University of Central Florida, Orlando, FL 32816 USA[1]
Institute of Computer Science, P.O. Box 1385, Iraklion, Crete, GREECE[2]
Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131 USA[3]

ABSTRACT

Back propagation neural networks have been widely used as classifiers in many complex classification tasks. However, early experimental results show that as the number of classes involved in a classification task increases, the classification accuracy of these networks decreases, especially in the presence of noisy inputs. In addition, larger size networks are needed to be utilized in such cases, a fact that may not always be possible. In order to overcome both of these undesirable effects a new approach is proposed in this paper which utilizes multiple, relatively small size networks to perform the classification task. This approach has been applied on a machine printed character recognition experiment and it has demonstrated better classification accuracy than the one exhibited by the single, larger size, network approach.

1. Introduction

Back propagation neural networks (BPNNs) pose a number of properties which make them particularly suited to complex classification problems. Although most of the current research in the field of neural networks is focused on the design of efficient learning algorithms to train a neural network to perform a specific task, in this paper we consider the problem of designing a neural network classification scheme having a high classification accuracy. Most of the current approaches for solving a classification problem are implemented using a single neural network classifier [1]-[5]. However, preliminary results have shown that as the number of classes involved in a classification task increases, the classification accuracy of the neural classifier is greatly affected. In addition, as the number of classes increases, the size of the network must be increased in order for it to be able to converge in a reasonable number of training epochs. Both of these effects are quite undesirable and restrictive, especially when real life classification problems are considered.

In order to keep the classification accuracy of a classifier high and its size manageable, we investigate the use of a neural based classification scheme including multiple but relatively small size BPNNs. This scheme may have BPNNs placed in a single layer or in multiple hierarchically structured layers. The BPNNs located at the bottom level of such a scheme are trained to classify input data to subsets of the classes involved in the classification problem. If each subset includes a small number of classes, then the classification accuracy of the networks is expected to be high. The way that the upper levels of the scheme are structured ranges from very simple to very complex. Assuming that a given input belongs to a class C_i, the purpose of the BPNNs located at the upper levels is to assign this input, possibly through BPNNs located in lower levels, to a BPNN located at the bottom level. The proper bottom level BPNN is the one trained to classify input data to a subset of classes which includes C_i. The classification decision made by this network is the final classification. In other words, the purpose of the upper level BPNNs is to select the proper bottom level BPNN that will perform the classification of the input. In case that the classification scheme consists of a single layer of BPNNs he input is forwarded through all the BPNNs located in the single layer. The classification decision taken by the BPNN having the strongest response is the final classification. Since the bottom level BPNNs can be made to have a high classification accuracy, if the upper level BPNNs have also a high classification accuracy, then the whole scheme will result in a high classification accuracy.

The way that the structure of such a classification scheme is determined is not a trivial task. In this paper we present two approaches related to this issue. The first of them considers a single level structure of BPNNs while the second one considers a two level structure of BPNNs. It should be emphasized that, these are early experimental approaches and further research is needed before a higher classification accuracy scheme is achieved.

In order to illustrate the proposed approach we have applied it to a relatively simple machine printed character recognition problem. The paper has been organized as follows: In Section 2, we present experimental results, giving evidence to the fact that the classification accuracy of a BPNN decreases when the number of classes involved in its classification task increases. In Section 3, we present some early classification schemes involving multiple BPNNs. Some interesting experiments and results are included also in this section. It should be mentioned that the momentum back propagation algorithm was used in all the experiments [6]. Finally, Section 4 includes our conclusions.

2. Classification accuracy vs number of classes: An experimental evidence

In most neural network based approaches used to solve a classification problem, a single neural network plays the role of the classifier. However, it can be illustrated experimentally that when the number of classes involved in a classification task increases, the classification accuracy of the network classifier is greatly affected, especially when distorted or noisy inputs must be classified. To illustrate this fact we have considered a machine printed character recognition experiment. The character set we have used includes the ten numbers plus the twenty six upper-case letters of the English alphabet. Each character pattern is arranged in a 9x7 grid. All input character patterns are given in binary notation. It should be mentioned that although the data set chosen may not be absolutely representative, it gives a clear evidence to the fact mentioned above.

The experiment consists of the training of several BPNNs with subsets of the original character set. In a character recognition problem, each character has to be assigned to a distinct class. Thus each character represents a class in this case. If we train a BPNN with different character subsets then we actually change the classes involved in the classification problem. Furthermore, we can increase or decrease the number of classes by choosing a larger or smaller subset of the original character set. The number of output nodes in a BPNN is set equal to the number of classes involved in its training, that is, equal to the number of characters included in its training set. A specific class is represented by setting one of the output nodes of the network equal to one and all the other equal to zero. The number of input nodes is determined by the dimensionality of the character patterns, (63 in our case). Three layer networks were utilized in all the experiments. During the training phase of a network, training patterns are presented randomly to the network until the error associated with each output node becomes less than 0.1 [7]. This error is defined as the absolute difference between the actual and the desired value of this node. During the recognition phase of a network, an unknown pattern is presented to the network. The output node having the maximum actual value determines the class to which the unknown pattern belongs to.

TABLE 1. Recognition accuracy vs number of classes

Recognition Accuracy									
Percent of noise	0	5	10	15	20	25	30	35	40
5 classes	100	100	100	99	91	77	60	38	19
10 classes	100	100	99	88	66	47	35	15	11
15 classes	100	99	88	66	50	28	22	13	5
20 classes	100	93	84	57	40	26	15	6	1
25 classes	100	90	69	46	34	24	14	1	0

Table 1 shows collective results involving the recognition accuracy of networks trained with an increasing number of classes. Different network architectures were tried and the training subsets used were formed by characters chosen randomly from the original character set. Test patterns were original patterns corrupted with noise. Noise corruption is formed by randomly reversing each bit of a character with an appropriate probability. As Table 1 illustrates, the recognition accuracy of a network is greatly affected as the number of classes increases.

3. A multiple network classification approach

It is quite obvious from the above experimental results that if we want to keep the recognition accuracy of a network high, then we must somehow utilize a network trained to classify input data to a small number of classes. However, most of the real life classification problems involve a large number of classes. In this paper we examine several approaches which will enable us, even in cases where the classification problem involves a large number of classes, to use networks trained to classify data into a relatively small number of classes. To illustrate the key idea of our approach, let us assume that for a given problem we have to train a BPNN to classify a set of data to a set of classes C. If we can divide the set of classes C into a number of disjoint subsets C_i, then we can assign one BPNN to each subset C_i and we can train it with the subset of training examples which belong to the classes included in C_i. In this way, the original classification problem has been actually divided into a number of smaller subclassification problems and we have let a set of BPNNs to solve them. Thus, instead of using a single BPNN to solve the whole problem, we use multiple BPNNs to solve it. These networks have been trained to classify data into a significantly smaller number of classes, a fact which guarantees that each one of them will exhibit a much higher classification accuracy compared to a single BPNN that is required to perform the entire classification task by itself. However, since each one of these BPPNs solves actually a part of the original classification problem, a way for providing a solution to the original classification problem must exist. Two different approaches have been tried in this paper. The first of them considers that the classification scheme consists of a single layer of BPNNs and the solution to the original classification problem is simply the solution given by the BPNN having the strongest response. The second approach considers that the classification scheme consists of two different levels structured in a hierarchical manner. The upper layer consists of a single BPNN while the bottom layer consists of multiple BPNNs. The solution to the original classification problem the solution given by a specific bottom level BPNN which has been selected according to some rules from the upper level BPNN. We must emphasize at this point that both of the approaches presented here exhibit only preliminary results.

3.1 A single stage classification approach

The first of the approaches considers that the classification scheme consists of a single layer of BPNNs and the solution to the original classification problem is simply the solution given by the BPNN having the strongest response. This subsection contains a number of experiments illustrating the way that such a scheme can be designed. In order to implement this idea, it is needed to use a proper scheme for representing a class through the values obtained at the outpout nodes of a BPNN. According to this scheme, each class is related to one of the output nodes of the network. The desired value for this node is one while the desired value for all the other nodes is zero. Each time an input is presented to the network, only one of the output nodes turns on, while all the other output nodes turn off. The class to which this node is related is the class to which the input belongs. Such a scheme is usually called local scheme [8]. During the training of a BPNN, a task is considered solved when the error between the actual value of an output node and the desired one becomes less than 0.1, for all the output nodes and for all the training examples. This is actually an error definition based on the maximum norm [7]. After training has been established, an input pattern is classified to a class by forwarding it through the network and picking the output node having the maximum response (we don't treat the output values as binary, we use the continuous output values which in the case that the sigmoid output function is used, these values belong to the interval (0,1)).

Let us suppose that the original classification problem has been divided into a number of subclassification problems and multiple BPNNs have been trained to solve them. Given an unknown input, we forward it through all the BPNNs. Obviously, each network will give its own response but it is expected that the BPNN which has been trained with examples which belong to the same class that the unknown patterns belongs to, will give the maximum response. Thus, if we select the network having the maximum response, the unknown input is classified in the class related to the output node of this network having the maximum value. Figure (1) illustrates this approach.

A question that might arise at this point is the way in which a classification task is divided into several subtasks. Actually this can be done by clustering the data set of the classification task into a smaller number of classes using a traditional clustering approach. In this paper we have used a clustering technique involving classification trees. This technique is one of the most simple-minded methods. Each cluster is represented by the average of all it's members. Initially, each data represents a cluster with just itself in it. The algorithm iterates as follows:

```
while number_of_clusters > 1 do
look for a pair of cluster whose representatives have minimal distance;
replace the pair with a new cluster having the two as subclusters;
compute representative vector for new cluster;
end
```

Classification trees seem to be a very helpful since they provide many alternative ways for dividing a data set into a number of classes.

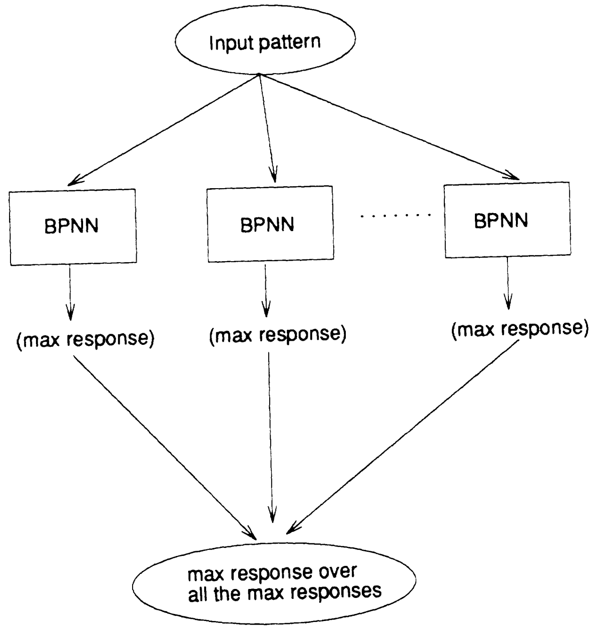

Figure 1. A single stage approach.

A simple character recognition experiment is utilized in order to illustrate the whole approach described in this subsection. The character set includes the numbers from 0 to 9. Each number pattern has the same characteristics as the character patterns described in Section 2. The problem we consider is the recognition of number patterns, especially when they have been distorted by noise. This is a classification task involving ten classes (each class corresponds to a number). The classification tree obtained by applying the above clustering technique on the number set is illustrated in Figure (2). Using this tree we are able to divide the number set into several subsets in many ways. For example, we can divide this set in two subsets ({4}, {0,1,2,3,5,6,7,8,9}), or in three subsets ({1}, {4}, {0,2,3,5,6,7,8,9}), or finally into ten subsets ({0}, {1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}). In this experiment, each number pattern represents a class. Dividing the data set into a set of classes has the effect that the set of classes to which the data belong to is also divided into subsets. In the general case that a class is represented by more than one pattern, the same procedure applies by considering the cluster centers of each class. As we can see, each one of the subsets obtained involves a smaller number of classes and at the final classification level each subset involves a single class.

Since the structure of this tree illustrates that the patterns (0,3,5,6,8,9) are very likely to belong to the same class, we initially consider the following subclasses: {4}, {1}, {7}, {2} and {3,5,9,6,0,8}. We can assume now that the original classification problem has been subdivided into five subclassification problems. The first four of them involve classification into one class while the fifth involves classification into six classes. A BPNN must be applied to each subclassification problem in order to solve it. Each network must be trained to classify input data to the classes involved in the subclassification problem it tries to solve. Specifically, the first BPNN must be trained to recognize the pattern 4, the second BPNN the pattern 1, the third BPNN the pattern 7, the fourth BPNN the pattern 2 and finally, the fifth BPNN the patterns 3,5,9,6,0,8. Since four of the subclassification tasks involve only one class, it seems to be wasteful to use four different BPNNs for each one of them. In order to avoid such a waste, we have merged these four subclassification tasks into a single task {4,1,7,2}. We will demonstrate later that such an action does not seem to have any serious impact to the performance of the whole approach.

Two BPNNs must be utilized now in order to solve the two subclassification problems obtained above. The first of the networks must be trained to classify the patterns 1,2,4 and 7 into four classes while the other one must be trained to classify the patterns 0,3,5,6,8, and 9 into six classes. Since the approach taken here is to train multiple networks and to select the maximum response given by one of them, we must somehow guarantee that when an input is presented to these networks, the only network that responds to this input most strongly is the network which has been trained with patterns belonging to the same class with the class that the input belongs to. This can be achieved by training each one of the networks with an augmented training set. The augmented training set will include not only the original training set, which we call regular examples, but also a set of training patterns which we call "negative" examples. The set of negative examples is formed by merging the training sets of all the other BPNNs. The reason for calling them negative examples is because when we train a BPNN with them, no output node is permitted to have a large output value for that network. That is, a negative example is required to turn off all the output nodes of a network, letting the BPPN for which this negative example is a regular training example to respond more strongly.

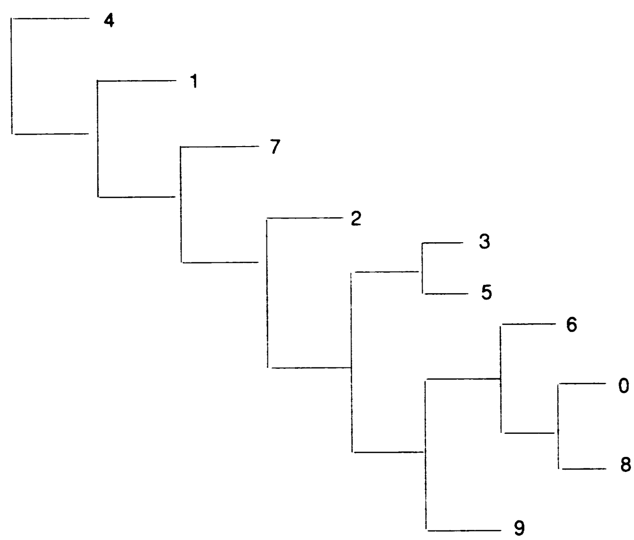

Figure 2. The classification tree of the number set.

Three layer networks were utilized with 63 nodes in the input layer and 5 nodes in each of the hidden layers. The number of nodes in the output layer was determined by the number of classes involved in the corresponding classification task (4 for the first network and 6 for the second). The number of nodes per hidden layer was set equal to 10. The training parameters are the same for both of the networks (learning rate 0.3, momentum term 0.9). After learning has been established, an unknown pattern is forwarded to both of the networks. The decision taken by the network having the strongest response is the final recognition result. Table (2) illustrates the results obtained by our approach. Table (3) illustrates the results obtained by the classical approach, that is, training a single BPNN (we have chosen a 3 layer BPNN, with 63 input nodes, 10 output nodes, 10 nodes per hidden layer, a learning rate equal to 0.3 and a momentum term equal to 0.9), with a training set equal to the entire number set. Comparing Table (2) to Table (3) it can be seen that the proposed approach performs better than the original one.

TABLE 2. Proposed approach - 2 BPNNs

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	100	98	91	82	70	56	41	25

TABLE 3. Traditional approach - 1 BPNN

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	98	92	88	70	61	44	29	22

Some interesting conclusions can be derived if we consider the performance of each one of these networks independently. Tables 4 and 5 illustrate the recognition accuracy that each one of these networks exhibits by itself (that is, when they are tested only on pure or noisy patterns of their training sets).

TABLE 4. BPNN trained to classify data into the classes {4,1,7,2}

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	100	98	97	94	89	79	65	48

TABLE 5. BPNN trained to classify data into the classes {3,5,6,0,8,9}

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	99	98	90	80	71	55	43	34

The above results indicate that the network of Table (5) which has been trained to classify data into six classes has a very low recognition accuracy compared to the network of Table (4). This is something we were expecting since the patterns belonging to the subclassification problem {3,5,6,0,8,9} are similar in contrast to the patterns belonging to the other subclassification problem {4,1,7,2}. This observation makes clear that the network related to the results of Table (4), is more responsible for most of the misclassification errors of the proposed approach. In order to increase the classification accuracy of the proposed approach we have subdivided the classification task of the less accurate network into two subclassification tasks and we have assigned two BPNNs to solve them. Since each of the subclassification problems involves a smaller number of classes it is expected that they will perform better in terms of accuracy, a fact which will upgrade also the recognition accuracy of the proposed approach. The classification tree of the number set suggest that the class {3,5,6,0,8,9} can be subdivided into the classes {3,5} and {6,0,8,9}. Thus the subclassification problems we must consider at this point are the following: ({4,1,7,2,}, {3,5} and {6,0,8,9}). The first subclassification problem {4,1,7,2,} has already solved by a BPNN. It is therefore needed to train two more BPNNs in order to solve the other two subclassification problems. Table (6) illustrates the classification accuracy of the proposed approach when three BPNNs are considered. Comparing Table (2) to Table (6), it can be seen that the recognition accuracy has been increased by a small amount but not by the amount we were expecting.

TABLE 6. Proposed approach - 3 BPNN

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	100	98	93	83	73	58	45	27

Some useful conclusions can be derived by testing the recognition accuracy of each one of these networks independently, a similar test that we did in the previous paragraph. The classification accuracy of each one of these BPNNs has been tested only on pure or noisy patterns derived from their training example sets. Tables (4), (7) and (8) illustrate the recognition results obtained.

TABLE 7. BPNN trained to classify data into the classes {3,5}

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	100	100	100	98	97	91	78	70

TABLE 8. BPNN trained to classify data into the classes {0,6,8,9}

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	99	98	91	81	67	58	45	42

The above results illustrate that the network trained to classify data into the classes {0,6,8,9} has a low recognition accuracy, indicating that this network might be the reason for most of the misclassification errors obtained. In order to further increase the recognition accuracy of the proposed method we have further subdivided this classification task into two subtasks. The classification tree of the number set suggest that the class {6,0,8,9} can be subdivided into the classes {6,9} and {0,8}. Thus, the subclassification problems we must consider at this point are the following: {4,1,7,2,}, {3,5}, {6,9} and {0,8}. The first two of them have already being solved by two BPNNs, thus two more BPNNs are needed to solve the rest of them. Table (9) illustrates the classification accuracy of the proposed approach. Comparing Table (6) to Table (9), it can be seen that the recognition accuracy of the proposed method has been considerably increased.

TABLE 9. Proposed approach - 4 BPNNs

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	100	99	97	92	80	65	48	30

The increment in recognition accuracy has been obtained due the fact that each of the BPNNs solving one of the subclassification problems has actually a relatively high recognition accuracy by itself. This is also illustrated through the results presented in Tables (4), (7), (10), and (11). These results have been derived by testing each BPNN only on pure or noisy patterns derived from their training examples set.

TABLE 10. BPNN trained to classify data into the classes {0,8}

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	100	100	100	98	96	91	84	72

TABLE 11. BPNN trained to classify data into the classes {6,9}

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	100	100	100	99	98	93	83	67

For illustration purposes we have subdivided also into two different tasks the classification problem involving the classes {4,1,7,2}. This classification problem has been formed by merging together four smaller classification problems. However, these smaller problems have the property that patterns belonging to the corresponding classes are relatively dissimilar. The two subclassification problems obtained by dividing the classification problem {4,1,7,2} into two subproblems are the following: {4,1} {7,2}. The subclassification problems that we must consider at this point are the following : {4,1}, {7,2,}, {3,5}, {6,9} and {0,8}. The last three of them have already solved by three BPNNs, thus two more BPNNs are needed to solve the first two. Table (12) illustrates the classification accuracy of the proposed approach. Comparing Table (9) to Table (12) it can be seen that recognition accuracies are about the same.

TABLE 12. Proposed approach - 5 BPNNs

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	100	99	97	93	84	65	48	32

In general, the classification accuracy of a network decreases as the number of classes involved in the classification task increases. However, the rate of decrease differs from case to case. We have observed that if the classes involved in a classification task are not well discriminated then the rate of decrease is high. However, when the classes are quite dissimilar, the rate of decrease is smaller. This means that we can merge together subclassification tasks involving quite dissimilar classes without significantly degrading the performance of the proposed approach. This fact has been already illustrated for the case of the {4,1,7,2} subclassification task.

3.2 A multiple stage classification approach

Although the above approach offers better recognition accuracies compared to the recognition accuracy of the traditional approach (Tables (3), and (9)), it is required that every time an unknown pattern needs to be recognized, it has be forwarded through all the networks in order to choose the network having the maximum response. In this section we describe a second approach which actually does not use information from all the bottom level networks in order to perform a recognition. This approach utilizes a two]layer scheme. The BPNNs located at the bottom level are trained to solve subclassification problems in a similar manner as described subsection 2.1. The upper level includes a single BPNN which is is trained to assign an unknown pattern to the most proper BPNN located at the bottom level of the scheme, which actually performs the recognition of the pattern. In other words, the BPNNs located at the upper level actually selects a BPNN located at the bottom level to perform the recognition. The way that the upper level BPNN, performs such a selection is based on the way it is trained. Since each bottom level BPNN has been trained to recognize patterns belonging to a subset of the original set of classes, the upper level BPNN learns to assign an input pattern to the bottom level BPNN which has been trained with patterns belonging to the same class that the input pattern belongs. Since the bottom level BPNNs can offer a high recognition accuracy, the performance of this approach depends on the recognition accuracy of the upper stage BPNN. The approach proposed in this subsection is illustrated in Figure (3).

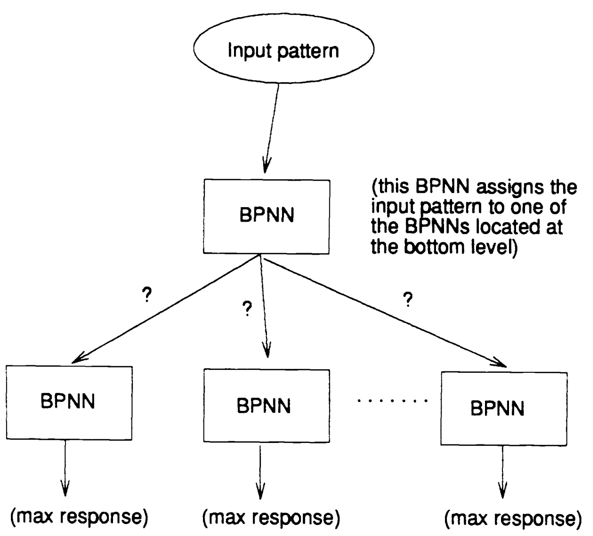

Figure 3. A two stage approach.

In order to illustrate this approach we have performed again a sequence of similar experiments. Initially we divided the original classification problem into the following subclassification problems: {4,1,7,2} and {3,5,9,6,0,8} (Figure (3)). The BPNN located at the upper level is trained to classify an input pattern into two different classes according to the subclassification problem that the input pattern belongs to. Since each subclassification problem is solved by one of the bottom level BPNNs, the classification performed by upper level BPNN is actually a selection of a bottom level BPNN. After this selection has been established, the selected bottom level BPNN performs the recognition of the input pattern. Table (13) illustrates the results obtained by this approach.

TABLE 13. Proposed approach - 1 BPNN at upper layer - 2 BPNN at bottom layer

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	100	98	92	83	73	56	40	24

Comparing Table (3) to Table (13) it can be seen that the proposed approach has given much better results. Comparing the two layer approach to the one layer approach (Tables (2) and (13)) we conclude that the results are almost the same. Tables (4) and (5) illustrate again that one of the bottom level BPNNs has a low recognition accuracy, a fact that degraded the performance of the one layer approach. Following the same ideas, we divide the original classification problem into the following subclassification problems: {4,1,7,2,}, {3,5} and {6,0,8,9}. Before we proceed with our experiments, let us test the classification accuracy of the upper level BPNN. As we have mentioned, its classification accuracy is of critical value since the performance of the whole approach depends greatly on it. Table (14) shows the classification accuracy of the upper level BPNN when pure or noisy inputs used.

TABLE 14. Upper level BPNN - classification into 2 classes

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	100	99	98	96	91	84	76	63

Table (15) illustrates the recognition accuracy of the proposed approach when three bottom level BPNNs are utilized. Although the results of Tables (13) and (15) are very similar there is actually a factor that will affect the performance of the entire scheme as we continue our experiments. This factor is the increasing number of errors made by the upper level BPNN (see Table (16)) when it is required to discriminate among a larger number of classes.

TABLE 15. Proposed approach - 1 BPNN at upper layer - 3 BPNN at bottom layer

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	100	98	94	83	74	54	36	23

TABLE 16. Upper level BPNN - classification into 3 classes

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	100	99	97	90	84	72	60	52

Tables (4), (7) and (8) illustrates once more time that one of the bottom level BPNNs has a low recognition accuracy, a fact that possible affects the performance of the whole scheme. Thus, subdividing the original classification problem into four subclassification problems ({4,1,7,2,}, {3,5}, {6,9} and {0,8}) may be a good reason for increasing the recognition accuracy of the whole scheme. Table (17) shows the results obtained. As it was expected, the recognition accuracy of the whole scheme was not increased, on the contrary, in some cases it was decreased. This occurs due the fact that the recognition accuracy of the upper level BPNN (Table (18)) decreases as the number of classes involved in its classification task increases. In addition, since the classes involved in its classification task are not well discriminated (for example, numbers 0,6,8,9 are similar), the rate with which the recognition accuracy is decreasing is expected to be high.

TABLE 17. Proposed approach - 1 BPNN at upper layer - 4 BPNN at bottom layer

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	100	97	90	80	68	53	35	22

TABLE 18. Upper level BPNN - classification into 4 classes

Simulation Results									
Percent of noise	0	5	10	15	20	25	30	35	40
Percent of accuracy	100	97	93	84	73	65	55	44	36

In order for the classification accuracy of the whole system to be increased, either the classification accuracy of the upper level BPNN must be increased or another approach must be followed. Hansen and Salamon [9] have suggested a method for improving the classification performance of a BPNN, which may become very useful for our approach. Another approach seems to be the use of multiple level BPNN classification schemes. The idea is that whenever a BPNN located at a specific level has a low recognition accuracy, new BPNNs positioned in the next level are added to the scheme in order to keep the recognition accuracy of the scheme high.

4. Conclusions

Early experimental results have shown that the accuracy of BPNN classifiers decreases as the number of classes involved in a classification task increases. A new approach was proposed in this paper which utilizes multiple, relatively small size BPNNs to perform a classification task. Each of these BPNNs is actually trained to classify data into a rather small number of classes, a fact which makes these networks very accurate. The multiple BPNN schemes proposed here may consist of a single layer of BPNNs or of more than one layers of BPNNs, structured in a hierarchical manner. A one and two layer approaches were considered in this paper. Both of them were applied on a machine printed character recognition experiment and demonstrated better classification accuracy than the one exhibited by the single, larger size, network approach. However, multiple layer schemes will be considered in our future research, since such schemes are expected to perform much better.

5. Acknowledgments

This research was supported in part by grants from Florida High Technology and Industry Council, the Boeing Computer Services under Contract W-300445 and the Sandia National Laboratories under Contract 87-5002. The authors also want to express their appreciation to Andreas Stolcke for making the clustering software package available.

6. References

[1] T. Sejnowski and C. Rosenberg, "Parallel networks that learn to pronounce eglish text", *Complex Systems*, vol. 1, pp. 145-168, 1987.

[2] R. Gorman and T. Sejnowski, "Learned classification of sonar targets using a massively-parallel network", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1135-1140, 1988.

[3] D. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network", In *Advances in Neural Information Processing Systems I*, ed. D. Touretzky, pp. 305-313, San Mateo: Morgan Kaufmann, 1989.

[4] Y. Le Cun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Back propagation applied to handwritten zip code recognition", *Neural Computation*, vol. 1, pp. 541-551, 1989.

[5] G. Bebis and G. Papadourakis, "Implementation of Character Recognition using Neural Networks and Traditional Classifiers", *Proceedings of the NEURONET International Symposium on Neural Networks and Neural Computing*,

pp. 33-36, Prague, September 1990.

[6] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, *Parallel Distributed Processing (PDP), Explorations in the Microstructure of Cognition*, Volume 1: Foundations, MIT Press, Cambridge, Massachusetts, 1986.

[7] G. Bebis, G. Papadourakis and M. Georgiopoulos,"Back Propagation: Increasing Rate of Convergence by Predictable Pattern Loading", *Intelligent Systems Review*, vol. 1, no 3, 1989.

[8] R. A. Jacobs, "Increased Rates of Convergence Through Learning Rate Adaptation", *Neural Networks*, vol 1, pp. 295-307, 1988.

[9] L. Hansen and P. Salamon, "Neural Network Ensembles", *IEEE Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993-1001, 1990.