2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

A NEURAL NETWORK BASED ON SUBNETS——SNN

Feng Liu, Jianxin Jiang, Jun Cheng, Kechu Yi

Lab 106, Dept. of Info. Engineering

Xidian University, Xi'an, China

10.21437/ICSLP.1992-389

ABSTRACT

In this paper, we present a new kind of neural network model based on subnets--SNN. The network is composed of a set of subnets and a decision layer. Each subnet is a Multilayer Perceptron that has two outputs units and used as a simple pattern classifier. The decision layer takes all the outputs of the subnets into account, and makes the final decision. We also give an training algorithm. Each pattern can be trained independently, which is possible to train a NN on a personal computer. In the paper, we compared SNN with the MLP, and show that SNN greatly decreases the compl exity of the networks.

Evaluation experiments were conducted, using 10 Chinese vowel syllables. The results show that the SNN is effective and has more potential in the speech recognition.

1 INTRODUCTION

Artificial neural networks (ANN), and more partiuculary Multilayer Perceptrons(MLP), have been recognized as attractive tools for information processing. One of their main advantages lies in the existence of the effective training algorithm, called back-propagation(BP). Another important featrue of the MLP is their capability to approximate a very wide range of continuous mappings between input and output vector spaces. It has been proved that a network with one hidden layer of sigmoid units can approximate arbitrarily any continuous function.

In speech recognition, MLPs are usually employed as speech pattern discriminators, where output layer units represent class names. In case of a small number of classses, higher perrformances than achieved by conventional methods have been reported. At present, however, there are practical problems in applying MLPs to large vacabulary speech recognition. First, in order to learn complex discrimination hyperplanes for a large number of classes, the MLP discriminators requires a very large training data set and its structure becomes much more complex. These mean larger computational expense in training and impossible to perform on a PC. Second, if new recognition classes are added, the structure of the MLP must be changed, and the discriminators in the system must be retrained using a training data set for all classses. Finally, the more complex the MLP structure is, the more difficult the covergence of the network. The network is easier to go into local minimum.

To accelerate the convegence of the network, a lot of adaped algorithms have been proposed, but these proposals make the algorithms more complex, and the local minum remains unsolved. To overcome these problems, we present a new speech recognition model based on Sub Neural Networks(SNN). The SNN uses a group of MLPs as discriminators. Each subnet only needs to seperate two classes, so is simpler than the coventional MLP which must discriminate all classes. Hence, problems inherent in MLP are avoided. The convergence is faster, a new pattern can be added without retraining all the subnets.

The paper is orgnised as follows. In Section 2, the structure of the SNN is discribed. The algorithm is given in Section 3. In section 4, we present the experiments conducted. Section 5 discusses the advantages of the SNN and the comparison with MLP.

2 THE NEURAL NETWORKS BASED ON SUBNETS--SNN

2.1 The Structure of A Subnet

See Figure 1. A subnet is a MLP that has some

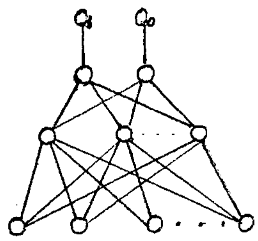

Fig 1. The structure of the subnet

special features. First, a subnet has only two output units O1 and O0. Second, each subnet represents a pattern to be recognised and is used as a classifier and trained to seperate the corresponding class from all the other classes. So if there are L classes of patterns to be recognised, there should be L subnets correspondingly. Finally, all the subnets has the same number of input units and are trained with the same training set.

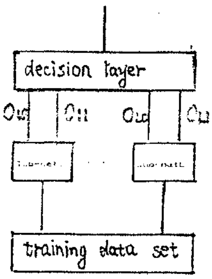

Fig 2. The structure of SNN

2.2 The Structure of The SNN

Figure 2 shows the architecture of the SNN. The SNN is composed of a group of subnets in the lower layer and a decision layer in the upper layer. A square represents a subnet.

In Figure 3, L is the number of the subnets (also the number of the patterns)

Oi0. Oi1 $(i = 1, 2, \ldots, L)$, the output units of the subnet i.

$Q = \{q1, q2, \ldots, qN\}$ is the training data set.

$P = \{P1, P2, \ldots, PL\}$ is the pattern set, each pattern

corrresponding a subnet.

All the outputs of the subnets are fed into the decision layer, where the final decision is made.

2.3 Learning In A SNN

The training of the SNN takes two steps, the training of the subnet and that of the decision layer. In the first step, each subnet is trained seperately like a MLP. Suppose subnet i represents pattern

Pi. Giving a training example $q \in Q$,

if $q \in Pi$, $Di0 = 1$, $Di1 = 0$;(Di0 and Di1 are the desired outputs of subnet i). else

if $q \notin Pi$, $Di0 = 0$, $Di1 = 1$.

The subnet i functions as a discriminator to seperate Pi from other kinds of patterns. In the second step, all the outputs of subnets are fed in as the inputs of the decision layer. The decision layer can be a simple net where the final result is obtained by finding the subnet whose output O0 is maximum. But in general the decision layer takes all the inputs into account and uses AI, fuzzy mathmatics or even NN to make the final decision. The method that uses fuzzy mathmatics to make the decision can be seen in Jianxin Jiang's term paper[1].

3 TRAINING ALGORITHM

In the training process, the subnets are trained independently using the same training data set. The Back– Propagation algorithm is used in training Multilayer Perceptrons as

$$Wij(n+1) = Wij(n) + \eta * j * Oi + \alpha * [Wij(n) - Wij(n-1)] \quad (1)$$

where $1 > \eta > 0$ is the learning rate and α is the momentum term. The BP algorithm is an iterative gradient descent of the mean square errror between the actual output of a MLP and desired output. It is the extesion of the LMS algorithm in the nonlinear space. In LMS algorithm, the weights are adjusted as

$$Wj(n+1) = Wj(n) - \eta * \nabla_j \quad (2)$$

where is the energy gradient, and $\nabla_j = -2 * [ej * xj]$. ej is the error, xj is the input. So we have

$$Wj(n+1) = Wj(n) + 2 * \eta * E[ej * xj] \quad (3)$$

For actual implementation, we replace ∇_j with $\hat{\nabla}_j = -2 * ej * xj$, that is

$$Wj(n+1) = Wj(n) + 2 * \eta * ej * xj \quad (4)$$

which is equal to (1).

When the input is stationary, $E[\hat{\nabla}_j] = <\hat{\nabla}_j> = \nabla_j$ ($< >$ is the time average, and E[ ] is the statistic average)

But this is not the case where a subnet is trained. Given L kinds of patterns, each having S training examples. A subnet is to seperate a pattern from all the other patterns, that is, to seperate S examples from the other (L– 1) * S examples. Hence

$$E[\hat{\nabla}_j] = 1/s * \sum_j ej * xj + 1/(LS - S) * \sum_j xj * ej \neq <\hat{\nabla}_j>$$

In the training process, the output Oj1 approchs 1 very fast, while output Oj0 is near to 0 all the time, the training fails.

To let $<\hat{\nabla}_j> = E[\hat{\nabla}_j]$, we adjusted η in the BP algorithm as follows.

$$\eta = c \quad if \quad qj \in Pi \quad and \quad (5)$$

$\eta = c/(L-1)$ if $qj \notin Pi$, where c is a constant value and $1 > c > 0$.

The experimental results shows that using the moderate algorithm, the net convergences very fast.

4 EXPERIMENTS AND THE RESULTS

4.1 The Training Data

In this experiment, we used 10 Chinese vowel syllables as the patterns to be seperated. Each syllable was uttered 100 times, hence 1000 utterances as the training set. All utterances were sampled at a 16KHz sampling rate, and Hamming windowed. LPCCEP coefficients were computed. All coefficients of an input token(in this case, 8 frames of speech) were first fed into a self– orgnised feature map(SOFM). The SOFM is used as a vector quantizer and the size of the map is 80. The outputs of the map is computed as $Zj = 1 / (1+\varepsilon * Dj * Dj)$, where ε is a factor and Dj is the distance of the input vector from the jth code vector of the map. (For the detailed discription, see[1]). Then we obtained 1000 vectors as the training data set.

4.2 Training With SNN

Every subnet has 80 input units, 4–8 hidden units(see table 1) and 2 output units. A subnet represents a syllable so there are ten subnets for the 10 syllables. In order to compare the results, all subnets are trained with the same iteration times(200 times here). The results are showed in the table 1, where the accuracy rates are within the training data set. The accuracy rates within the training data set can reach 100% as the iteration times increase.

Table 1 The Experiment Results of SNN

vowel	the number of hidden units	training time of subnets(hour)	recongnotion rate of the subnet (%)	the final recognition accuracy (%)
a	6	1.2	83	92
i	4	1.1	94	95
u	4	1.1	92	98
ai	8	2	79	87
ao	8	2	87	90
ou	8	2	69	67
an	8	2	71	85
ong	4	1.0	99	97
yao	8	1.5	97	90
uan	8	1.5	76	78

To compare SNN with MLP, we trained a MLP that has 80 input units, 20 hidden units and 10 output units with the BP algorithm and using the same data set. When iteration time reaches 2000, the net has no signs to convergence.

4.3 Adding A New Pattern

In this experiment, we add a new pattern to the trained SNN. Only the newly added subnet is trained, while all the other subnets remained unchanged. The decision layer had to be retrained, but it is very easy. The results (see table 2) shows that the SNN can easily add new patterns without much influence on the former results.

Table 2 Adding A New Pattern To SNN

vowle	a	i	u	ai	ao	ou	an	eng	ao	uan	iang
beforc adding a new pattern	92	95	98	87	90	67	85	97	90	78	
after adding a new pattern	91	93	97	89	92	60	80	97	88	77	80

5 DISSCUSION

5.1 The Complexity of The Network

The complexity of the network is deterimined by the number of the connection weights. In the MLP model, the hidden layer is for the substraction and extraction of the features of the input vectors. Fewer hidden units results in the missing of the features of the input vectors, while larger number of hidden units makes the network more sensitive to noise and the computation is expensive. So, to obtain higher performance, it is very important to select the number of hidden units. Often, the number of hidden units is obtained by experiments with different number of the hidden units. But for a MLP with many patterns, it is often impossible.

In our SNN model, the case is different. Each subnet is only to seperate a pattern from other patterns, so there are fewer features needed, which means fewer number of the fidden units are needed. From above experiments, we see that 4–8 hidden units is enough for a subnet.

The complexity of the network has an effect on the computation. The simpler the network is, the smaller the computational cost is.

5.2 The Convegernce Of The SNN

As we metioned above, the BP algorithm is the extention of the LMS algorithm to the nonlineary space. So it is very easy to go into local minimum. The fewer hidden units, the smaller the posibility to go into local mimimum. The SNN is proposed rightly on this fact. We know that the convergence of the network is a process to find global minimum enery in the nonlinear space. To add a hidden units is to increase the demesion of the nonlinear space, making it more difficult for the network to find global minimum.

In our SNN, each subnet needs only a few hidden units, hence avoid the local minimum greatly. In addition, since each subnet can be trained independently, the optional number of hidden units can be obtained by the experiment easily.

5.3 The Advantage

Each subnet functions like a MLP, but since a subnet is only to seperate two classes, its structure is simpler than that of a MLP and convergences more easilly. Besides, since the training of a subnet is very easy, we can get higher performance by optiminizing the structure and parameters of the subnets. Finally, the decision layer takes all the outputs into account, so the SNN is more reliable and flexible than MLP which makes the decision only by the maximum of the outputs. The experiments also proves the conclusion.

5.4 The Training Of A New Pattern

If we add a new pattern to a MLP, then the structure of the MLP must be changed and the network must be retrained. But for our SNN, what we must do is to trained a new subnet with all the training data and adjust the decision layer. The experimental results show that the rocognition accuracy has only a slight decrease. It is a very good feature especially for a system with many patterns.

6 CONCLUSION

In this paper, we proposed a new NN model—SNN. Compared with MLP, SNN has some advantages. (1) It decreases the complexity of the network. (2) It accelarates the convergence of the network and, to a great degree, overcomes the local minium. (3) Each subnet is trained independently, making it easy to addjust the parameter and the structure of the network. It also makes it possible to train the network on a PC. (4) The recognition accuracy is increased by the decision layer. (5) When adding a new pattern, the SNN need not be retained. The experimental results proves it applicable.

Though the experiment is insufficient, we think SNN a potential way wspecially in the real application.

REFFERENCES

〔1〕 J.X.Jiang "A Hybrid Neural–Fuzzy–Neural Frame Work For SpeechRecognition".

〔2〕 Licheng Jiao The System Theory on Neural Network.

〔3〕 T.P.Vogl, et al "Accelerating the Convergence of BP Method". Billogical Lybernetics, vol. 59, 1988.

〔4〕 S.Kirkpartrick, et al., "Optimization By Simuluated Annealing." Since, vol. 220, pp671, 1988.

〔5〕 R.P. Loppman "An Introduction to Computing With Neural Network." IEEE ICSSP pp4–22, April, 1989.

〔6〕 Allexander Waibel "Phoneme Recognition Using TDNN", IEEE ICASSP 1989