A STUDY OF ENGLISH WORD CATEGORY PREDICTION BASED ON NEURAL NETWORKS

Masami NAKAMURA, Kiyohiro SHIKANO

ATR Interpreting Telephony Research Laboratories
Seika-chou, Souraku-gun, Kyoto 619-02, JAPAN

ABSTRACT

Using traditional statistical approaches, it is difficult to make an N-gram word prediction model to construct an accurate word recognition system because of the increased demand for sample data and parameters to memorize probabilities.To solve this problem, NETgrams, which are neural networks for N-gram word category prediction in text, are proposed. NETgrams can easily be expanded from Bigram to N-gram networks without exponentially increasing the number of free parameters. Training results show that the NETgrams are comparable to the statistical model and compress information. Results of analyzing the hidden layer (Micro Features) show that the word categories are classified into some linguistically significant groups. Also we confirmed that NETgrams performed effectively for unknown data,that is to say, NETgrams interpolate sparse training data naturally just like the deleted interpolation. In addition, this paper proposes a new method to speed up the Back-Propagation algorithm, which can automatically determine better parameters and achieve a shorter training time.

1. INTRODUCTION

There is a method of predicting word category using the appearance probabilities of the next word to correct word recognition errors in text [1] [2]. The point of improving prediction ability is to use as much past word information as possible. However, by using this statistical approach, it is difficult to make an N-gram word prediction model because of the increased demand for sample data and parameters to learn probabilities.

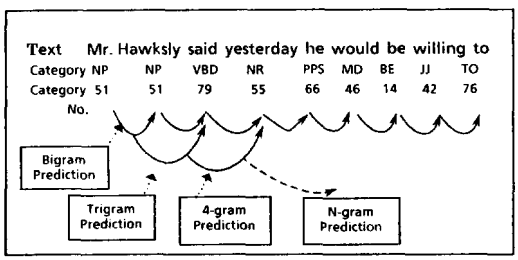

Fig. 1 Word Category Prediction Using Brown Corpus Text Data

Neural networks are interesting devices which can learn general characteristics or rules from limited sample data. Neural networks are particularly useful in pattern recognition. In symbol processing, NETtalk [3], which produces phonemes from English text, has been successful. Now we are trying to apply neural networks to word category prediction in English text. It is expected that this task will be very difficult to train because of a many-to-many mapping problem with many exceptions for output data, and a symbol-to-symbol mapping problem rather than a pattern-to-symbol mapping problem as with pattern recognition. We are interested in learning to what degree a neural network can be applied to symbol processing.

This paper describes NETgrams, which are neural networks for N-gram word category prediction in text. NETgrams are constructed by a trained Bigram network with two hidden layers. Each hidden layer learns the coarse-coded Micro Features (MF1 or MF2) of the input or output word category. NETgrams can easily be expanded from Bigram to N-gram networks without exponentially increasing the number of free parameters. NETgrams are tested by training experiments with the Brown Corpus English Text Database [4].

2. NETgram (NETwork for N-gram word category prediction)

To design neural networks for word category prediction in text, the following are required :

a. Training data are the category names(tags) in Brown Corpus English Text Database. Categories are 88 tags, corresponding to parts of speech in Brown Corpus, and every sentence is headed by a blank.

b. The network input layer has several blocks corresponding to the number of input words. For example, a Bigram Network has one input block and a Trigram Network has two. Each block has 89 units and local representation for an input word category. Therefore, in one input block, only one unit corresponding to word category number is turned ON(1); The others are turned OFF(0).

c. Outputs are prediction values for the next possible word categories. Therefore, an output layer has 89 units.

d. Training algorithm is the Back-Propagation algorithm[5].

In addition, we also consider the following :

e. After learning, hidden layers obtain the coarse-coded Micro Features of the input and output word categories.

f. NETgrams can easily be expanded from Bigram to N-gram networks without exponentially increasing units and connections between them.

Two NETgrams are proposed considering the above requirements. Each NETgram is expanded from one basic Bigram network.

CH2673-2/89/0000-0731 $1.00 © 1989 IEEE

2.1. Basic Bigram Network

The basic Bigram network is a 4-layer feed-forward network, as shown in Fig.2, which has 2 hidden layers so that each hidden layer obtains coarse-coded MF (Micro Features) of the input or output word category. Because this network is trained for the next word category as the output for an input word category, hidden layers are expected to learn some linguistic structure from the relationship between one word category and the next in the text.

2.2. Expansion to N-gram Network

We propose two models for expansion to N-gram networks.

2.2.1. Model 1

Model 1 consists of basic Bigram networks put side by side as shown in Fig.3. An upper hidden layer (MF2) of a basic Bigram network is fully connected to that of the next basic Bigram network with the link weight set w4. Each basic network link weight set (w1, w2 or w3) has the same values. Model 1 can learn the Micro Features (MF1) for each input word category independently and can be expanded to a recurrent network with the self-loop link weight set w4.

2.2.2. Model 2

Model 1 will probably be difficult to train because the number of layers from the first input block to the output layer increases as the number of grams increases. In order to hold the number of layers from all input blocks to the output layer at four, Model 2 is proposed as shown in Fig.4. Model 2 has a structure such that every new input block produced as the number of grams increases is fully connected to the lower hidden layer of one basic Bigram network with the link weight set w1'. Initial values of the link weight set w1' are all zero. Therefore, the training starts at the output values equal to the trained output values of the basic Bigram network. However, as all input word category information is compressed to one lower hidden layer (MF1), input word category information must to some degree be lost as input blocks increase. Therefore, when expanding from Trigram network to 4-gram network, one lower hidden layer block is added and the first and second input blocks are fully connected to one lower hidden layer block, and the second and third input blocks are fully connected to the other lower hidden layer block.

3. HOW TO TRAIN NETgram

As input data, word categories in the Brown Corpus text are given in order from the first word in the sentence to the last word. In one input block, only one unit corresponding to the word category number is turned ON (1); The others are turned OFF (0). As output data, only one unit corresponding to the next word category number is trained by ON (1) ; The others are trained by OFF (0).

The training algorithm is a new method to speed up the Back-Propagation algorithm, which is proposed in section 5.

How to train a NETgram, e.g. a Trigram network, is shown in Fig.5. First, the basic Bigram network is trained, and next, the Trigram networks are trained with the link weight values trained by the basic Bigram network as initial values. 4-gram networks are trained in the same way.

This task is a many-to-many mapping problem. Thus it is difficult to train because the updating direction of the link weight

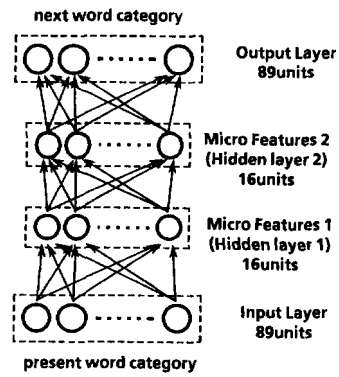

Fig.2 Basic Bigram Network for Word Category Prediction

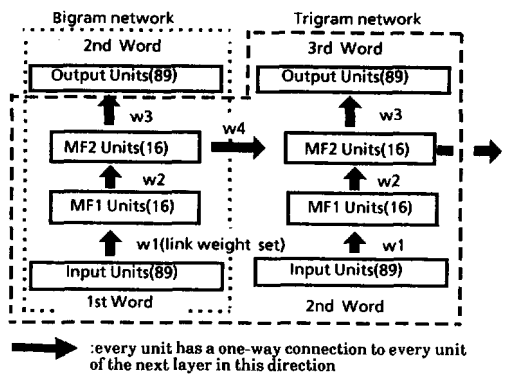

Fig.3 NETgram Model 1 for Word Category Prediction

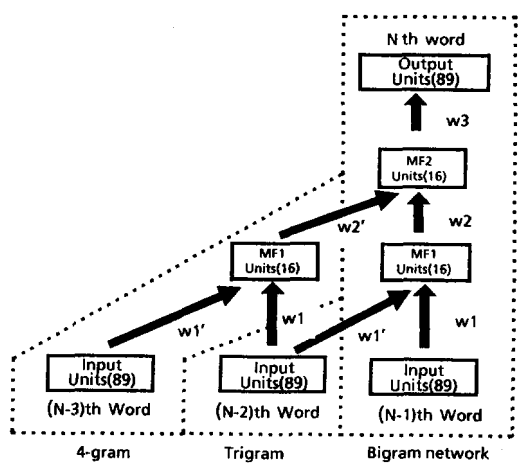

Fig.4 NETgram Model 2 for Word Category Prediction

vector easily fluctuates. In a two-sentence training experiment of about 50 words, we have confirmed that the output values of the basic Bigram network converge on the next occurrence probability distribution. However, for many training data, considerable time is required to train. Therefore in order to increase training speed, we use the next word category occurrence probability distribution calculated for 1,024 sentences (about 24,000 words) as output training data in the basic Bigram network. Of course, in the Trigram and 4-gram training, we use the next one-word category as output training data.

4. TRAINING RESULTS

4.1. Basic Bigram Network

Word category prediction results show that NETgram (the basic Bigram network) is comparable to the statistical Bigram model.

Next, we consider whether the hidden layer has obtained the Micro Features. We calculated the similarity of every two lower hidden layer (MF1) output vectors for 89 word categories and clustered them. Similarity S is calculated by

$$S(Ci,Cj) = \frac{(M(Ci),M(Cj))}{\| M(Ci) \| \| M(Cj) \|} \qquad (1)$$

where M(Ci) is the lower hidden layer (MF1) output vector of the input word category Ci. (M(Ci),M(Cj)) is the inner product of M(Ci) and M(Cj). $\| M(Ci) \|$ is the norm of M(Ci). The clustering result is shown in Fig.6. Clustering by the threshold of similarity, 0.985, the word categories are classified into linguistically significant groups, which are the HAVE verb group, BE verb group, subjective pronoun group, group whose categories should be before a noun, and others. Therefore NETgrams can learn linguistically significant structure naturally.

4.2. Trigram Network

Word category prediction results are shown in Fig.7. Two NETgrams (Trigram networks) are comparable to the statistical Trigram model for test data.

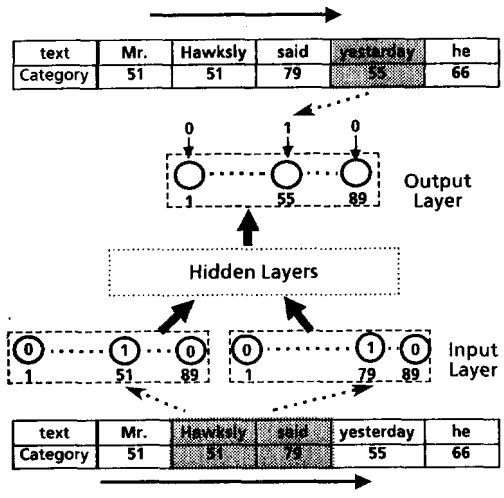

Fig.5 How to Train NETgram (Trigram Model)

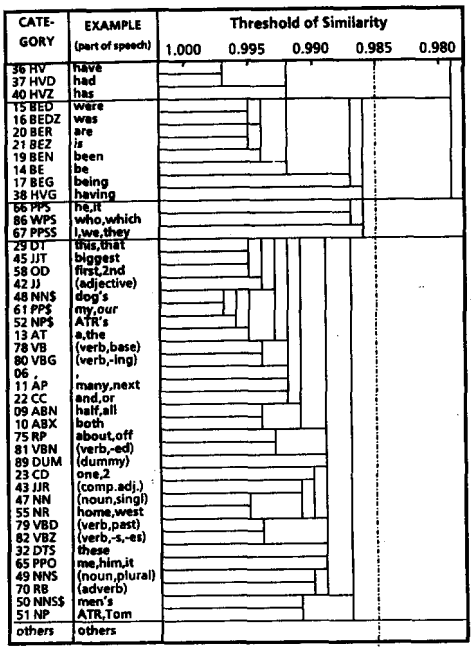

Fig. 6 A Clustering Result of MF1 Output Values of Basic Bigram Network

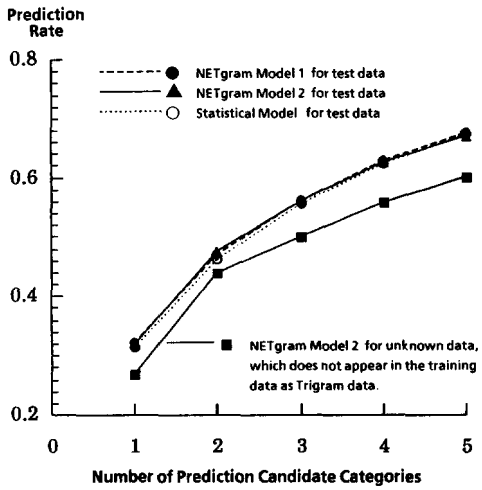

Fig.7 NETgram (Trigram) Prediction Rates

Table 1 Number of Free Parameters (ratio; NETgram/statistical model)

	Statistical Model	NETgram Model 1	NETgram Model 2
Bigram	7,921 $=89^2$	3,225 (1/2.5)	3,225 (1/2.5)
Trigram	704,969 $=89^3$	5,193 (1/136)	4,649 (1/152)

We discuss the free parameters of the NETgram. The number of free parameters of the statistical model is a power of 89 in this task, e.g. $89^3 = 704{,}969$ in the Trigram model, and increases exponentially as the number of grams increases. On the other hand, the number of free parameters of NETgram is the number of link-weights, e.g. 5,193 in the Model 1 Trigram network, and increases linearly though the number of grams increases. Therefore, the NETgram in Trigram prediction compresses information more than 130 times as shown in Table 1.

Next, we discuss a Trigram prediction result for unknown data,whose trigram never appeared in the training data. So we picked up the unknown data from the test data. NETgram prediction rates for unknown data are reasonable as shown in Fig.7. On the other hand, the statistical Trigram model cannot predict unknown data. This result shows that NETgrams interpolate sparse training data naturally, such as the deleted interpolation in the statistical model [6].

5. A NEW METHOD TO SPEED UP THE BACK-PROPAGATION ALGORITHM [7]

Considerable time is required to train NETgrams. It is also very difficult to converge the global minimum because of the many-to-many mapping problem. In this section, a new method to speed up the Back-Propagation algorithm, called DCP (Dynamic Control training Parameters), is proposed.

A basic theory of the Back-Propagation [5] is the gradient descent. The rule for changing link weights is given by

$$\Delta wij(k) = \eta \cdot (-\partial Ep/\partial wij) + \alpha \cdot \Delta wij(k-1) \qquad (2)$$

where Ep is the error between the output values and the desired training values and is a function of the link weights. wij is the link weight from the ith unit to the jth unit. The first term is the direction of the gradient descent and the second term is the memory of the last updating step size. This provides a kind of momentum in weight space. Each term has a parameter, η or α, which decides the current real updating step size. The optimal values of these parameters depend on the shape of the weight space, determined by the type of task and the size of the training data, and depend on the degree of training. The DCP method dynamically changes the training parameters (η,α) every N training iterations so that Ep is at a minimum as in the following equation.

$$Ep(wij(k) + \Delta wij(k)(\eta(k),\alpha(k)) \qquad (3)$$
$$= \min_{l,m} Ep(wij(k) + \Delta wij(k)(\eta_l,\alpha_m))$$

where one of the combinations of N η_l and α_m is chosen.

As a result of the DCP method, a shorter training time is attained (4.3 times faster in the basic Bigram network) and some local minima are avoided.

6. CONCLUSION

In this paper we have presented two NETgrams, neural networks for N-gram word category prediction in the text. Each model is constructed by a trained basic Bigram network with two hidden layers. NETgrams can easily be expanded from Bigram to N-gram networks without exponentially increasing the number of free parameters.

The training results showed that the Trigram word category prediction ability of NETgrams was comparable to that of the statistical Trigram model and compressed information more than 130 times. Also we confirmed that NETgrams performed effectively for unknown data which never appeared in the training data, that is to say, NETgrams interpolate sparse training data naturally just like the deleted interpolation. The comparison with the deleted interpolation is being planned.

The results of analyzing the hidden layer (Micro Features) after training showed that the word categories were classified into some linguistically significant groups, that is to say, a NETgram learns a linguistically significant structure naturally.

In addition, this paper proposed a new method to speed up the Back-Propagation algorithm, which Dynamically Controls the training Parameters (DCP), updating step size and momentum. As a result of the DCP method, a shorter training time is attained and unsuitable local minima are avoided.

At present, the NETgram for 4-gram prediction is being trained. The 4-gram prediction ability of the NETgram is gradually increasing. But at present, it is not yet significantly superior to that for Trigram prediction. Training takes much time because 4-gram prediction requires much training data. The next problem is to speed up training much more.

ACKNOWLEDGEMENT

The authors would like to express their gratitude to Dr. Akira Kurematsu, president of ATR Interpreting Telephony Research Laboratories, which made this research possible for his encouragement and support. We are also indebted to the members of the Speech Processing Department at ATR, for their help in the various stages of this research.

REFERENCES

[1] F.Jelinek, "Continuous Speech Recognition by Statistical Methods", Proceedings of the IEEE, Vol.64, No.4 (1976.4)

[2] K.Shikano, "Improvement of Word Recognition Results by Trigram Model", ICASSP 87, 29.2 (1987.4)

[3] T.J.Sejnowski et.al., "NETtalk, A Parallel Network that Learns to Read Aloud", Tech. Report, The Johns Hopkins University EESS (1986)

[4] Brown University, "Brown Corpus", Tech. Report, Brown University (1967)

[5] D.E.Rumelhart et al., "Parallel Distributed Processing", M.I.T. Press (1986)

[6] F.Jelinek et al., "Interpolated Estimation of Markov Source Parameters from Sparse Data", Pattern Recognition in Practice, ed. E.S. Gelsema and L. N. Kanal, North Holland (1980)

[7] M.Nakamura et al., "A Study of English Word Category Prediction Based on Neural Networks", Tech. Report, TR-I-0052, ATR Interpreting Telephony Research Labs. (1988)