Proceedings of the 1995 IEEE ICNN, Perth, Australia.

A Parallel Recurrent Cascade-Correlation Neural Network with Natural Connectionist Glue

Ingrid Kirschning, Hideto Tomabechi and Jun-Ichi Aoe
Dept. of Information Science and Intelligent Systems
Faculty of Engineering, University of Tokushima,
2-1 Minami Josanjima-Cho, Tokushima Shi, 770 Japan.
E-mail : ingrid@is.tokushima-u.ac.jp

ABSTRACT

Some problems of "unlearning" were encountered when using Fahlman's Recurrent Cascade Correlation Learning Architecture (RCC) for phoneme recognition. In this paper we present a parallel-modular RCC. The original RCC is transformed into a modular RCC, trained with natural connectionist glue. This is done in order to concentrate the "knowledge" about a group of patterns in a module, instead of distributing it across the whole network. The modules are connected in parallel, in contrast to the completely cascaded structure of the original RCC. This new approach provides an improvement in the recognition rates for tasks involving large numbers of features to be learned. The modularity, besides providing a better learning, makes training of large sample-sets easier and faster.

1. Introduction

The Recurrent Cascade-Correlation Learning Architecture (RCC) [2] offers several advantages over other neural networks, such as providing a near minimal multilayer topology by defining its own size during training. However, when attempting to learn large groups of patterns involving a great number of different features, as in spectrograms for speech recognition, we observed that this structure presents some disadvantages. The network has difficulties to generalize from various inputs and it tends to forget several previously learned features, resulting in a low recognition rate.

The Cascade-Correlation architecture was proposed by Fahlman and Lebiere [1], with the purpose to generate a network where each unit learns a specific task as fast as possible, avoiding the random motion of the hidden units in space, as happens in standard backpropagation [8]. Later, the recurrent version of this architecture, the RCC, where the hidden units have also a self-recurrent link that feeds the units activation back to itself (see figure 1) was proposed by Fahlman [2].

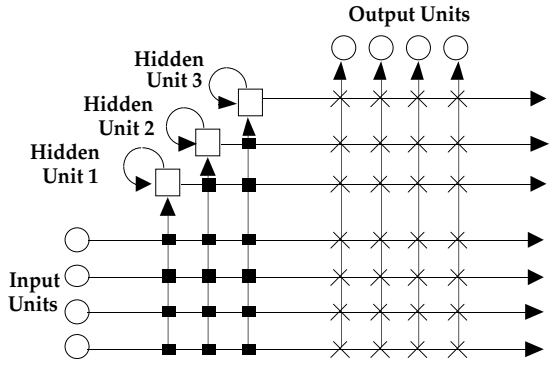

Figure 1: The RCC architecture. The connections marked with a black square as well as the self-recurrent links are frozen and the connections marked with an X are trained repeatedly [2].

The RCC network is created in a cascaded fashion. It stars without hidden units and adds them one by one during the training while the error surpasses a determined threshold.

Each new hidden unit receives the activation from all the previously installed ones and from its own recurrent link. Its own activation is, however, never passed to the previously installed units, thus the cascaded structure.

Every hidden unit in this kind of network can be viewed as a new layer, becoming a specialized feature detector. Since it is not possible to change it after it was installed, further training becomes cumulative.

This learning architecture eliminates the need to guess the neural network's size, depth and topology in advance and provides a near-minimal multilayer topology fitted to the problem to solve.

The hidden units cooperate in the solution of the problem, as every new hidden unit is affected by the activation of all the previous ones, as in the example of the two spirals problem presented in [1], or in learning sequences of symbols as in the Reber grammar [2]. Each new unit specializes even more on the problem supported by the "knowledge" acquired by the previous hidden units.

However, when the task to be trained is complex, like the learning of the spectral representation of all the phonemes in an alphabet, the original RCC learning architecture presents some strong difficulties to generalize. To solve this problem we propose a parallel-modular RCC, as an alternative to the

original RCC architecture, which provides a better recognition rate than the original approach.

The next section introduces the parallel RCC architecture and our concept of "natural" connectionist glue. Further, section 3 presents the experiment performed to show the improvement achieved by this new approach, applying it to continuous speech recognition. Finally, section 4 presents the conclusions drawn from the experiment.

2. Modularity in the Recurrent Cascade-Correlation Network

To train the cascade-correlation network with large training sets, Fahlman proposed to divide the training set into a series of short "lessons", and train them one after the other, going from the simplest to the most complicated one. Then re-train the network with all the samples in a single training set [2; 3; 9]. The resulting network structure is shown in figure 2.

The re-training is done in the same way each lesson was trained, that is, keeping the incoming connections of the previously installed hidden units frozen, as well as their respective self-recurrent links.

The groups of hidden units generated during the training of each lesson grow in a cascaded fashion one on top of the other. When the network is finally re-trained with the complete training set, one last group of hidden units is created. We will refer to each of these groups of hidden units as *modules*.

2.1 The Parallel RCC

We propose to alter the way the connections of the original RCC interrupting the cascade and locating every new module parallel to the previous ones, with no connections between the modules. In this way the cascaded RCC of figure 2 is transformed into the parallel RCC shown in figure 3. Each module is totally independent from the activation of the others. This also opens the possibility to train the modules separately, which makes training faster and easier.

When the training of each subset of the training-set is concluded, the obtained modules are merged parallely into one single network, which is re-trained with all the training samples in one single set. This creates an additional group of hidden units, which we call *natural* connectionist glue (see fig. 3).

2.2 Natural Connectionist Glue

The *Connectionist Glue* is a concept developed by Waibel et al. for modularity and scaling in large phonemic neural networks [7; 10]. Several neural networks can be trained individually for small subsets of the training set, making them specialized modules. These modules are merged together into a greater network with common input and output layers. The connections from the input layer to these modules are fixed and an extra group of hidden units, the "connectionist glue", is added to the network. This network is then re-trained with that extra group of hidden units, whose connections are free to learn any missing features to supplement the features learned by the frozen modules.

Due to the nature of the RCC learning algorithm, the new hidden units are added one by one during the training process. This creates a *naturally added* connectionist glue (figure 3). The size of this module, the natural connectionist glue, depends on the problem being trained, in contrast to the connectionist glue used for the TDNN, where its size is fixed and determined before training starts [10].

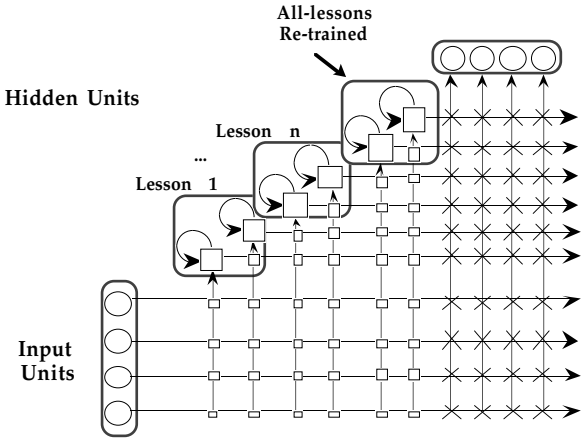

Figure 2 : Original RCC trained with the training set divided into "lessons". (The boxed connections as well as the self-recurrent links are frozen and the rest are trained repeatedly.)

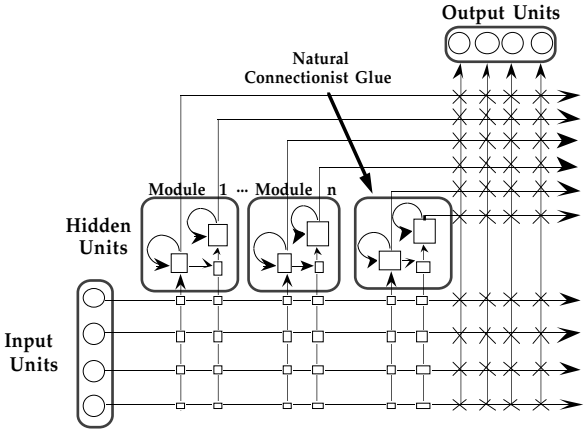

Figure 3 : The new RCC with **parallel** modules and Natural Connectionist Glue. (The boxed connections as well as the self-recurrent links are frozen and the rest are trained repeatedly.)

3. Experiment

To compare the performance of a non-modular cascade-correlation network, the traditional cascaded architecture and the proposed parallel structure, an experiment was performed using the following three approaches:

1) A non-modular network (trained with all the samples in one single training set);
2) the cascaded RCC (original version) where the training set is split into groups which are trained one after the other, and the resulting modules are built up in a cascade;
3) the parallel RCC where the training set is divided in exactly the same way as for the cascaded net, but the modules are built up in parallel.

These three approaches were trained to recognize a subset of the phonemes of the Japanese language /a/, /i/, /u/, /e/, /o/, /b/, /d/, /g/, /p/, /t/ and /k/ using the Time-Sliced Paradigm [5]. Since some of these phonemes sound very similar, it is desirable to teach the neural network to recognize the subtle differences between phonemes like /b/ and /d/. Thus the phonemes were divided between 3 groups according to their coarseness (based on [6; 10]). These three groups are : vowels (/a//e//i//o//u/), voiced stops (/b//d//g/) and unvoiced stops (/p//t//k/). By training each of these groups separately the networks will be able to learn the differences between similar sounding phonemes.

With this training set the performance and learning of the three neural networks, as well as their resulting sizes were compared.

All three networks have 217 input units plus a bias unit, and 242 output units [4]. Table 1 shows the sizes of the networks. Clearly there is no difference between the non-modular and the cascaded network, only the parallel network has 1200 connections less than the others.

All three networks were trained with the same samples, i.e. seven samples for each phoneme randomly extracted from a set of 43 different words, and they were tested to spot the phonemes inside the words. The testing set consisted of a list of 143 words, including those from which the training samples were extracted. We are including those words into the test-set because a small fraction of only one phoneme of the whole word was extracted for the training set, and the rest of the phonemes of the word could be used for testing. Naturally, for the recognition statistics the trained samples were not included into the counting.

Each network was trained and its progress recorded. Figure 4 compares the decrease of the overall error vs. the number of installed hidden units during the training for each network. While the error curve of the non-modular net descends smoothly, the other two jump up at the moment the network is presented with the complete training set and retrained.

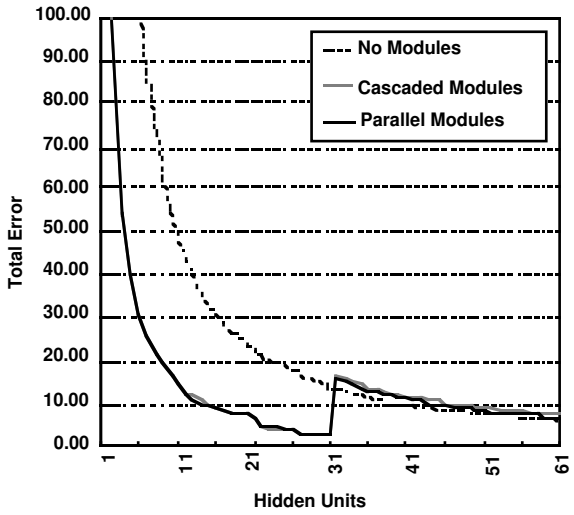

Figure 4 : Error curve for the three different approaches.

Network :	Non-Modular	Cascaded	Parallel
Total No. of Units	520	520	520
Total No. of Hidden Units	60	60	60
With Natural Connectionist Glue	No	Yes	Yes
Total No. of Connections	82,186	82,186	80,986

Table 1 : Comparing the sizes of the networks.

Phoneme Group	# of samples (test set only)	Recognition Rates		
		Non-Modular	Cascaded	Parallel
Vowels	313	93.93 %	96.49 %	96.81 %
Stops	81	34.57 %	40.74 %	43.21 %
Total	394	81.73 %	85.03 %	85.79 %

Table 2 : Recognition rates for correctly spotted phonemes. To test the network the complete words of the test set were passed through the net and the correctly spotted phonemes were registered. The test-results do not include the training samples.

	Increment in the recognition rate compared to :	
Group:\Architecture:	Non-Modular	Cascaded
vowels	+ 2.88	+ 0.32
stops	+ 8.64	+ 2.47
average	+ 4.06	+ 0.76

Table 3: Improvement obtained by the Parallel Architecture.

While it is adding the connectionist glue, the error decreases again. It shows that in essence, considering the total output error, all three networks have the same tendency. However, the difference is in their performance.

Interestingly, even though the three networks have the same level of output error, the results, shown in Table 2, show an increase in the recognition rate for the parallel network above the other two. The amount of this improvement can be seen more clearly in Table 3.

4. Conclusion

As the number of units increases during the training of large and complex training sets, the cascaded network looses generalization and recognition capability, producing even a null recognition for some phonemes. The modular training (or by lessons) gives each module a chance to specialize on a small group of patterns. However, due to the cascaded structure of the modules, the influence of one module on the next can be negative, as it filters out information and enhances other features that can confuse the following modules.

Thus, the proposed parallel-modular RCC lets each module learn to distinguish a specific group of patterns independently from whatever the other modules learned. Then the modules are merged and retrained adding the natural connectionist glue. This permits the modules to cooperate better with the others combining their very specific "expertise". In other words, it proves better to train the RCC in a parallel-modular fashion, so that the independently acquired "knowledge" is "localized" in each module, instead of distributing it through all the network. The result is an improvement in the recognition rate of phonemes in continuous speech recognition.

The RCC learning algorithm was originally chosen because it generates a near minimal multilayer neural network, which provides us with a speech recognition system small enough to run even on a PC.

The /b//d//g/-/p//t//k/ distinction task in speech recognition is well known as to be difficult because of the similitude between the phonemes. The low recognition rate itself only indicates that there are still several changes that need to be done to achieve good continuous speech recognition.

References

[1] S. Fahlman and C. Lebiere, "The Cascade-Correlation Learning Architecture", Technical Report # CMU-CS-90-100, Carnegie Mellon University, Pittsburgh, February 1990.

[2] S. Fahlman, "The Recurrent Cascade-Correlation Architecture", Technical Report # CMU-CS-91-100, Carnegie Mellon University, Pittsburgh, May 1991.

[3] S. Fahlman, Personal communication, (1994).

[4] I. Kirschning, "Continuous Speech Recognition Using the Time-Sliced Paradigm", M.E. Thesis, Tokushima University, Dept. of Information Science and Intelligent Systems, Tokushima, Japan, 1995.

[5] I. Kirschning, H. Tomabechi, "Phoneme Recognition Using the Time-Sliced Recurrent Recognizer", IEEE Proceedings of the 1994 ICNN-WCCI, Orlando, Fla., 1994, pp. 4437-4441.

[6] K. Kita, A Study on Language Modeling for Speech Recognition", *Ph.D. Thesis*, Waseda University, Japan, 1992.

[7] K. Lang, A. Waibel, G. Hinton, "A Time-Delay Neural Network Architecture for Isolated Word Recognition", Neural Networks, Vol. 3, 1990, pp. 33-34.

[8] D.E. Rumelhart and J.L. McClelland, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. I and II, Cambridge, MA: M.I.T. Press, 1986.

[9] H. Sawai, Y. Minami, M. Miyatake, A. Waibel, K. Shikano, "Connectionist Approaches to Large Vocabulary Continuous Speech Recognition", IEICE Transactions, Vol. E 74, No. 7, July 1991, pp. 1834-1844.

[10] A. Waibel, H. Sawai, K. Shikano, "Consonant Recognition by Modular Construction of large Phonemic Time-Delay Neural Networks, (1989), Readings in Speech Recognition, edited by A. Waibel, K. Lee, Morgan Kaufmann Publishers, 1990, pp. 405-408.