Massively Parallel Classification of EEG Signals Using Min-Max Modular Neural Networks

Bao-Liang Lu, Jonghan Shin, and Michinori Ichikawa

Lab. for Brain-Operative Device, RIKEN Brain Science Institute
2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan
{lu,shin,ichikawa}@brainway.riken.go.jp

Abstract. This paper presents a massively parallel method for classifying electroencephalogram (EEG) signals based on min-max modular neural networks. The method has several attractive features. a) A large-scale, complex EEG classification problem can be easily broken down into a number of independent subproblems as small as the user needs. b) All of the subproblems can be easily learned by individual smaller network modules in parallel. c) The classification system acts quickly and facilitates hardware implementation. To demonstrate the effectiveness of the proposed method, we perform simulations on a set of 2,127 non-averaged single-trial hippocampal EEG data. Compared with a traditional approach based on multilayer perceptrons, our method converges very much faster and recognizes with high accuracy.

1 Introduction

Neurophysiologists generate large amounts of time-series data such as EEG data as they record the electrical activity in the brain. Artificial neural networks have been applied to neurophysilogical data analysis such as classification of EEG signals [1,2,10]. However, training large networks on large sets of high-dimensional EEG data is a hard problem because no efficient algorithm is available for training large networks and a very long training time is required to achieve satisfactory learning accuracy [10]. To avoid this problem, existing methods usually use a few number of features extracted from EEG data as inputs. For example, only eight features were used in [10]. However, extremely reducing the number of features will cause the loss of useful information of the original EEG signals and the decrease in correct classification rate.

In this paper, we present a massively parallel method for classifying EEG signals based on the min-max modular (M^3) neural network, an alternative committee machine proposed in our previous work [4]. The method has the following several attractive features. a) A large-scale, complex EEG classification problem can be easily broken down into a number of independent subproblems as small as the user needs. b) All of the subproblems can be simply learned by individual smaller network modules in parallel, and therefore, a large set of high-dimensional EEG data can be learned efficiently. c) The classification system acts quickly and facilitates hardware implementation; consequently, it can

G. Dorffner, H. Bischof, and K. Hornik (Eds.): ICANN 2001, LNCS 2130, pp. 601–608, 2001.

be applied to implementing a hybrid brain-machine interface [5], which relies on the real-time sampling and processing of large-scale brain activity to control artificial devices.

2 EEG Data

It has been shown that hippocampal EEG signals are associated to cognitive process and behaviors, such as attention, learning, and voluntary movement [7]. The hippocampal EEG signals used in this study were recorded from eight adult male hooded rats between 300 and 400 g. These rats were housed in individual cages with food and water provided until the behavioral training. One week after surgery for implanting hippocampal electrodes, the rats were water-deprived and trained in a chamber by means of an oddball paradigm [7], in which occasional 'target' stimuli have to be detected in a train of frequent 'non-target' stimuli. We used a low frequency tone (so called odd tone) as 'target' stimuli and a high frequency tone (so called frequent tone) as 'non-target' stimuli. The animals were rewarded by water whenever they discriminate 'target' tone and cross the light beam in the water tube.

A total of 2,127 non-averaged single-trial hippocampal EEG signals were recorded from the rats. Each of the EEG signals is 6 sec in duration and belongs to one of the four classes, namely FR, FW, OR, and OW, where 'FR' means frequent tone and right behavior (no go), 'FW' frequent tone and wrong behavior (go), 'OR' odd tone and right behavior (go), and 'OW' odd tone and wrong behavior (no go). Figure 1 illustrates four non-averaged single-trial EEG signals belonging to FR, FW, OR, and OW, respectively. In the simulations below, we use 1,491 EEG signals for training and the rest of 636 EEG signals for testing. Table 1 shows the distributions of the training and test data sets.

Table 1. Distributions of the training and test data

	FR	FW	OR	OW
Training	1027	136	307	21
Test	430	68	128	10

3 Method

3.1 Feature Extraction with Wavelet Transform

In order to quantify changes in single-trial hippocampal EEG signals in both frequency and amplitude, we use wavelet transform techniques [9] to extract the features of EEG signals. The original EEG signals were convolved by the Morlet wavelet $w(t, w_o)$ with Gaussian shape both in the time domain and in the frequency domain around its central frequency w_o:

$$W(t, w_o) = \exp\left(jw_0 t - \frac{t^2}{2}\right) \tag{1}$$

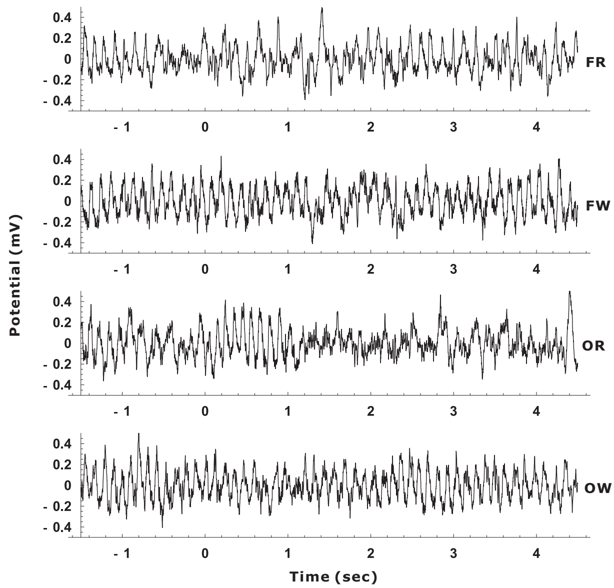


Fig. 1. Four EEG signals belonging to FW, FR, OR, and OW, respectively.

These wavelets can be compressed by a scale factor a and a shifted in time by a parameter b. Convolving the signal and the shifted and dilated wavelet leads to a new signal,

$$S_a(b) = \frac{1}{\sqrt{a}} \int W\left(\frac{t-b}{a}\right) x(t)dt \tag{2}$$

where W is the conjugate of the complex wavelet and $x(t)$ is the hippocampal EEG signal.

The new signals $S_a(b)$ are computed for different scaling factors a. In order to generate maps of hippocampal theta activity, the features of EEG signals were extracted between 5 Hz and 12 Hz out of the time-frequency maps. By selecting different numbers of samples in the time domain and using the same five wavelet coefficients in the theta frequency bandwidth, two sets of data were created, which have 200 and 2,000 features, respectively. Figure 2 shows the contour plots of the time-frequency representations of the four EEG signals illustrated in Fig. 1 with 2,000 features.

3.2 Task Decomposition

By using the task decomposition method proposed in our previous work [4], a K-class classification problem can be divided into $\binom{K}{2}$ two-class subproblems as follows:

$$\mathcal{T}_{ij} = \{(X_l^{(i)},\ 1-\epsilon)\}_{l=1}^{L_i} \cup \{(X_l^{(j)},\ \epsilon)\}_{l=1}^{L_j} \tag{3}$$

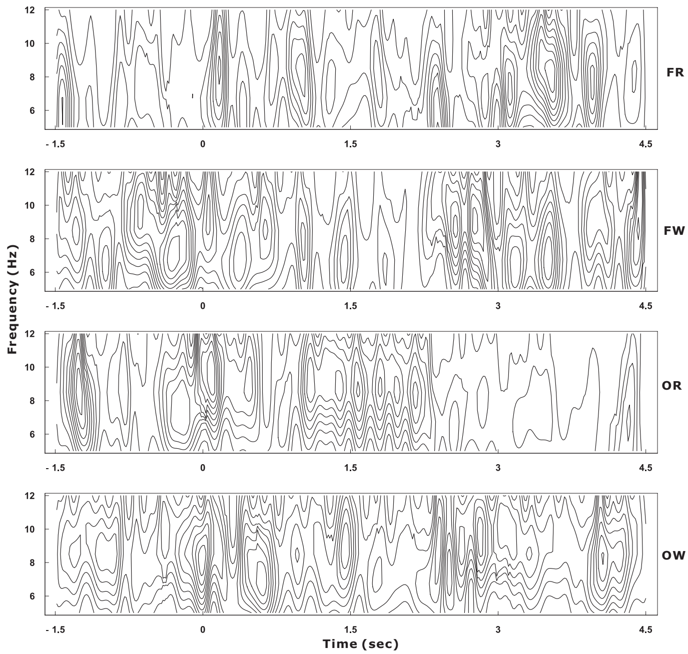


Fig. 2. The contour plots of the time-frequency representations of the four EEG signals illustrated in Fig. 1. Here, the number of features is 2,000.

where $i = 1, \cdots, K$, $j = i+1, \cdots, K$, ϵ is a small real positive number, $X_l^{(i)} \in \mathcal{X}_i$ and $X_l^{(j)} \in \mathcal{X}_j$ are the training inputs belonging to class $\mathcal{C}_i$ and class $\mathcal{C}_j$, respectively, $\mathcal{X}_i$ is the set of training inputs belonging to class $\mathcal{C}_i$, L_i denotes the number of data in $\mathcal{X}_i$, $\sum_{i=1}^{K} L_i = L$, and L is the total number of training data.

If some of the two-class problems defined by (3) are still large and hard to be learned, each of these subproblems can be further decomposed into a number of two-class problems as small as the user needs. Assume that $\mathcal{X}_i$ is partitioned into N_i $(1 \leq N_i \leq L_i)$ subsets in the form

$$\mathcal{X}_{ij} = \{X_l^{(ij)}\}_{l=1}^{L_i^{(j)}} \tag{4}$$

where $j = 1, \cdots, N_i$, $i = 1, \cdots, K$, and $\cup_{j=1}^{N_i} \mathcal{X}_{ij} = \mathcal{X}_i$. According to the above partition of $\mathcal{X}_i$, the two-class problem $\mathcal{T}_{ij}$ defined by (3) can be further divided into $N_i \times N_j$ much smaller and simpler two-class subproblems as follows:

$$\mathcal{T}_{ij}^{(u,v)} = \{(X_l^{(iu)}, 1-\epsilon)\}_{l=1}^{L_i^{(u)}} \cup \{(X_l^{(jv)}, \epsilon)\}_{l=1}^{L_j^{(v)}} \tag{5}$$

where $u = 1, \cdots, N_i$, $v = 1, \cdots, N_j$, $i = 1, \cdots, K$, $j = i+1, \cdots, K$, $X_l^{(iu)} \in \mathcal{X}_{iu}$ and $X_l^{(jv)} \in \mathcal{X}_{jv}$ are the training inputs belonging to class $\mathcal{C}_i$ and class $\mathcal{C}_j$, respectively. From (3) and (5), we see that a K-class problem can be decomposed into $\sum_{i=1}^{K} \sum_{j=i+1}^{K} N_i \times N_j$ two-class subproblems in a top-down approach.

According to (3), the four-class EEG classification problem is divided into $\binom{4}{2} = 6$ two-class subproblems, namely $\mathcal{T}_{1,2}$, $\mathcal{T}_{1,3}$, $\mathcal{T}_{1,4}$, $\mathcal{T}_{2,3}$, $\mathcal{T}_{2,4}$, and $\mathcal{T}_{3,4}$. From Table 1, we see that the number of training data for the smallest two-class subproblem $\mathcal{T}_{2,4}$ is 157, while the number of training data for the largest two-class subproblem $\mathcal{T}_{1,3}$ is 1,334. Although these two-class subproblems are smaller than the original problem, they are not adequate for massively parallel computation and efficient learning due to the following reasons. a) The subproblems are rather 'load imbalanced'. Since the speed of parallel learning is limited by the speed of the slowest subproblem, the unduly burdening of even a single subproblem can dramatically degrade the overall performance of the learning. b) Some of the subproblems are still too big for training. c) Some of the subproblems are very imbalanced, i.e., the training set contains many more data of the 'dominant' class than the other 'subordinate' class.

To speed-up learning, the bigger subproblems are further decomposed into a number of relatively smaller and simpler subproblems. According to (4), three larger training input data sets belonging to FR, FW, and OR are randomly broken down into 49, 6, and 15 subsets, respectively. As a result, the original four-class EEG classification problem is divided into $\sum_{i=1}^{4}\sum_{j=i+1}^{4} N_i \times N_j = 1,189$ balanced, two-class subproblems, where $N_1 = 49$, $N_2 = 6$, $N_3 = 15$, and $N_4 = 1$. The number of training data for each of the subproblems is about 40.

3.3 Massively Parallel Learning

An important feature of the above task decomposition method is that each of the two-class subproblems can be treated as a completely independent, non-communicating subproblem in the learning phase. Consequently, all of the subproblems can be learned in parallel.

In comparison with existing parallel implementations of neural network paradigms [8], the advantage of our massively parallel learning scheme is that it can be easily implemented not only on general-purpose parallel computers, but also on large numbers of individual serial machines and distributed Internet applications [3].

3.4 Module Combination

After learning, all of the trained individual network modules are integrated into an M^3 network by using three integrating units, namely MIN, MAX, and INV units, according to two simple module combination laws [4].

In contrast to the task decomposition mentioned above, the module combination process is bottom-up and involves the following two main steps.

a) The $N_i \times N_j$ network modules corresponding to the subproblems $\mathcal{T}_{ij}^{(u,v)}$ defined by (5) are integrated by using N_i MIN and one MAX units as follows:

$$\mathrm{MIN}_{ij}^{(u)} = \min(\mathrm{M}_{ij}^{(u,1)}, \mathrm{M}_{ij}^{(u,2)}, \cdots, \mathrm{M}_{ij}^{(u,N_j)}) \text{ for } u = 1, \cdots, N_i \tag{6}$$

and

$$\mathrm{M}_{ij} = \max(\mathrm{MIN}_{ij}^{(1)}, \mathrm{MIN}_{ij}^{(2)}, \cdots, \mathrm{MIN}_{ij}^{(N_i)}) \tag{7}$$

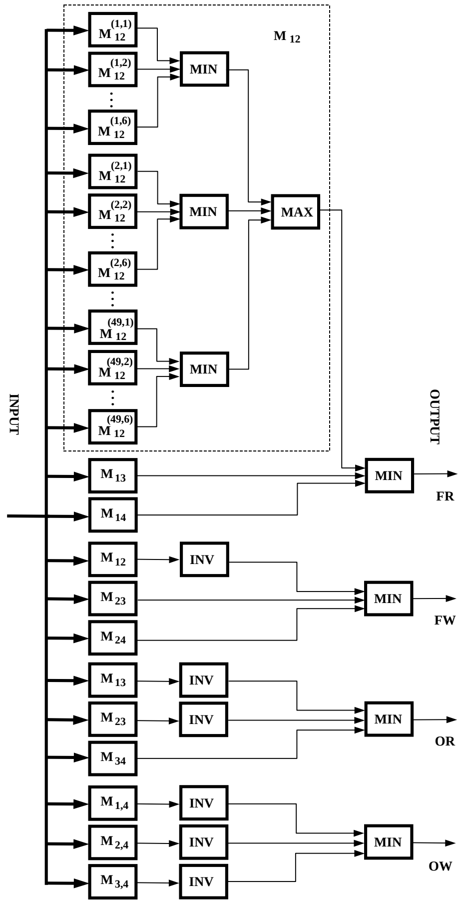


Fig. 3. The min-max modular network used for solving the four-class EEG classification problem, where thin arrowed-lines represent scalar inputs or outputs, and thick arrowed-lines represent vector inputs. Note that only module $\mathrm{M}_{1,2}$ is plotted in detail, and the other modules are roughly illustrated due to space requirements.

where $\mathrm{M}_{ij}^{(u,\,v)}$ denotes both the name of trained module corresponding to the subproblem $\mathcal{T}_{ij}^{(u,\,v)}$ and its actual output, and $\mathrm{MIN}_{ij}^{(u)}$ denotes the output of the combination of N_j network modules. For example, according to (6) and (7), the 294 modules $\mathrm{M}_{12}^{(u,v)}$ for $u = 1, \cdots, 49$ and $v = 1, \cdots, 6$ are integrated into the module M_{12} illustrated by the open dashed box in Fig. 3.

b) The $\binom{K}{2}$ network modules corresponding to $\mathcal{T}_{ij}$ and their $\binom{K}{2}$ inversions are integrated as

$$\mathrm{MIN}_i = \min(\mathrm{M}_{i1}, \cdots, \mathrm{M}_{ij}, \cdots, \mathrm{M}_{ik}) \text{ for } i = 1, \cdots, K \text{ and } i \neq j \qquad (8)$$

where MIN_i denotes the output of the combination of the $K-1$ modules by using the MIN unit. The M^3 network is guaranteed to produce solutions to the

Table 2. Performance of M^3 networks and single MLPs

No. of features	Method	No. of modules	No. of hidden units	CPU time (s.)		Correct rate (%)	
				Max	Total	Training	Test
200	MLP	1	60	120,011	120,011	100.0	78.9
	M^3	1189	2	7	2,209	100.0	79.7
2000	MLP	1	80	727,355	727,355	100.0	80.5
	M^3	1189	5	103	11,919	100.0	81.8

original problem as follows:

$$\mathcal{C} = \arg \max_{i} \{\mathrm{MIN}_i\} \text{ for } i = 1, \cdots, K \tag{9}$$

where $\mathcal{C}$ is the class that the M^3 network has assigned to the input.

4 Computer Simulations

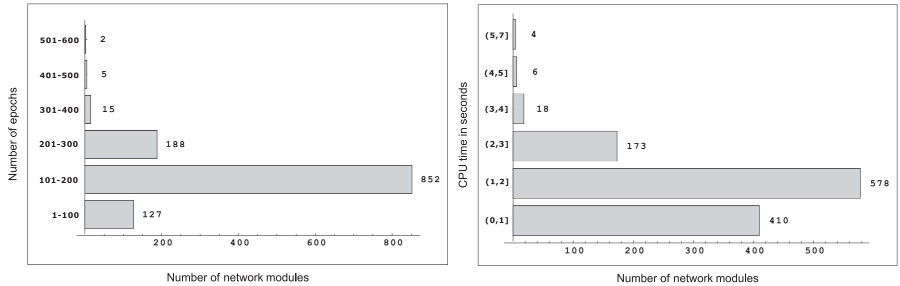
In the simulations, 1,189 three-layer MLPs were selected as network modules to learn the corresponding 1,189 subproblems. To compare M^3 networks with traditional MLPs, the original problems were also learned by three-layer MLPs. All of the network modules and the three-layer MLPs were trained by standard back-propagation algorithm [6]. All of the simulations were performed on an HP 9000 C240 workstation.

The simulation results are shown in Table 2, where 'Max' means the maximum CPU time required for training any networks. Figures 4(a) and 4(b) show the distribution of the 1,189 trained network modules measured by the number of epochs and CPU time (sec), respectively. From these figures we see that over 98% of the network modules converged within 300 epochs. From Table 2, we can see that our method is about 17,144 and 7,061 times faster than the method based on traditional three-layer perceptrons for learning the two EEG data sets that have 200 and 2,000 features, respectively, providing that learning in our method was performed in parallel. Even though all of the network modules were trained in serial, the results show that our method is still very much faster than the conventional method.

After the learning, the 1,189 individual trained network modules were integrated into the M^3 network shown in Fig. 3. Looking at Fig. 3, we see that our method has two useful features. a) The response time of the classification system is almost independent of the number of modules, that is, the system response time is almost independent of the problem size. b) The system might be easily implemented in hardware because of its modular structure.

5 Conclusions

In this paper we have presented a massively parallel method for classifying high-dimensional EEG data based on min-max modular neural networks. The advantages of the method over existing approaches are its high modularity and



(a) (b)

Fig. 4. Summary of the 1,189 network modules used for classifying the EEG data with 200 features. a) Distributions of trained modules measured by the number of epochs, and b) distributions of trained modules measured by CPU time in seconds.

parallelism, good scalability, and very fast learning speed. We have also demonstrated that the method is superior to the traditional approach based on multilayer perceptrons in generalization performance. By using the proposed method, we have begun doing computer experiments on a big data set containing 10,635 single-trial hippocampal EEG signals.

References

1. Bankman, I. N., Sigillito, V. G., Wise, R. A., Smith, P. L.: “Feature-based detection of the K-complex wave in the human electroencephalogram using neural networks”, *IEEE Trans. Biomedical Engineering*, vol. 39, no. 12, pp. 1305-1310, 1992.
2. Jansen, B. H., Desai, P. R.:“K-complex detection using multi-layer perceptrons and recurrent networks”, *Bio-Medical Computing*, vol. 37, pp. 249-257, 1994.
3. Lawton, W: “Distributed net applications create virtual supercomputers”, *Computer*, vol. 33, no. 6, pp. 16-21, 2000.
4. Lu, B. L., Ito, M.: “Task decomposition and module combination based on class relations: a modular neural network for pattern classification”, *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 1244-1256, 1999.
5. Nicolelis, M. A. L.: “Actions from thoughts”, *Nature* (insight feature), vol. 409, pp. 403-407, 2001.
6. Rumelhart, D. E. Hinton, G. E., Williams, R. J.: “Learning internal representations by error propagation”, in ‘Parallel Distributed Processing: Exploration in the Microstructure of Cognition’, Vol. 1, (MIT Press, Cambridge, Mass, 1986)
7. Shin, J., Lu, B. L., Talnov, A., Matsumoto, G., Brankack, J.: “Reading auditory discrimination behavior of freely moving rats from hippocampal EEG”, *Neurocomputing*, vol. 38-40, pp. 1557-1566, 2001.
8. Sundararajan, N., Saratchandran, P.: *Parallel Architectures for Artificial Neural Networks*, Los Alamitos, California, IEEE Computer Society Press, 1998.
9. Torrence, C., Compoo, G. P. : “A practical guide to wavelet analysis”, *Bulletin of the American Meteorological Society*, 1998, Vol. 79, pp. 61-78
10. Tsoi, A. C., So, D., Sergejew, A. A.: “Classification of electroencephalogram using artificial neural networks”, in ‘Advances in Neural Information Processing Systems’, Vol. 6, (Morgan Kaufmann, San Mateo, CA 1994), pp. 1151-1158