

Research Article

A Sensitivity-Based Improving Learning Algorithm for Madaline Rule II

**Shuiming Zhong,^{1,2} Yu Xue,² Yunhao Jiang,² Yuanfeng Jin,³
Jing Yang,⁴ Ping Yang,² Yuan Tian,⁵ and Mznah Al-Rodhaan⁵**

¹ Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, China

² School of Computer & Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

³ Department of Mathematics, Yanbian University, Jilin, Yanji 133002, China

⁴ Beijing Normal University, Zhuhai 519087, China

⁵ Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh 11362, Saudi Arabia

Correspondence should be addressed to Yuanfeng Jin; yfkim@ybu.edu.cn

Received 6 May 2014; Revised 30 July 2014; Accepted 2 August 2014; Published 27 August 2014

Academic Editor: Yuri Vladimirovich Mikhlin

Copyright © 2014 Shuiming Zhong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a new adaptive learning algorithm for Madalines based on a sensitivity measure that is established to investigate the effect of a Madaline weight adaptation on its output. The algorithm, following the basic idea of minimal disturbance as the MR II did, introduces an adaptation selection rule by means of the sensitivity measure to more accurately locate the weights in real need of adaptation. Experimental results on some benchmark data demonstrate that the proposed algorithm has much better learning performance than the MR II and the BP algorithms.

1. Introduction

The ability to learn is the uppermost function of neural networks. Hence, to build a proper learning mechanism is a key issue for all kinds of neural networks. This paper focuses on the learning mechanism of a Madaline, especially improving its learning performance.

A Madaline (many Adalines) [1] is a binary feedforward neural network (BFNN) with supervised learning mechanism, which is suitable for handling inherently discrete tasks, such as logical calculation, pattern recognition, and signal processing. Theoretically, a discrete task can be regarded as a special case of a continuous one, and the BP algorithm [2] based on continuous techniques is by now the most mature learning algorithm of feedforward neural networks; that is, why the continuous feedforward neural networks (CFNNs) with the BP algorithm are more popular than Madalines. However, compared with the CFNNs, Madalines do have some obvious advantages in nature, that is, (1) it is easy for description of discrete tasks without extra requirement of discretization, (2) it is simple in computation and interpretation

with hard-limit activation function and limited input and output states, and (3) it is facilitative for hardware implementation with the available VLSI technology. Further, the process of discretizing CFNN's output for classification tasks is quite application-dependent and not suitable to be involved in a general learning algorithm. So, a learning algorithm for BFNNs without relying on continuous technique and discretization is worthy of being explored.

However, Madalines have not yet had an effective learning algorithm. In literatures, there have been many studies on Madaline learning since Madaline model was brought forward in early 1960s. On the whole, two main approaches on Madaline learning are well known. One is an adaptive approach that extends the perceptron rule [3] or something like that to Madalines. For example, Ridgway's algorithm [4] called MRI (Madaline rule I) by Winter [5] and the MR II (an extension of the MRI) [5, 6] apply Mays rule [7], a variation of the perceptron rule, to Madalines. Unfortunately, these algorithms are still too poor in performance to meet practical applications. The other is called geometrical construction approach [8, 9] which fabricates a set of hyperplanes based on

Adaline (neuron) structure feature to meet the input-output mapping in the training data set. The main disadvantage of this approach is that it usually results in a Madaline with much larger architecture than the one resulted from an adaptive learning algorithm, and thus it not only complicates the computation and hardware implementation but also degrades the generalization ability of the Madaline. In fact, the geometrical construction approach is not a learning algorithm in adaptive sense. Therefore, it is still significant to investigate the learning algorithm of Madalines.

It is obvious that the basic idea of minimal disturbance [1, 5, 6] is crucial to almost all adaptive supervised learning algorithms such as the MRII and BP. As a key principle, the idea tries, in each iteration of weight adaptation, to not only better satisfy the current training sample but also avoid as much as possible the disturbance on the effects established by previous training samples. Although the BP algorithm was derived from the steepest descend method, it actually also follows this idea [1]. Unfortunately, the MRII does not well implement the principle, and this is the main cause of its poor performance. It can be found that the confidence level (summation of weighted inputs of a neuron) [5, 6] adopted by the MRII as a measure for implementing the principle cannot guarantee to select proper neurons for the weight adaptation during learning.

In Madaline learning, one of the most important issues is the effect of variation of network parameters on its output; so, Madaline sensitivity (i.e., the effect of parameter variation on network output) can be used to properly assess this effect. Based on the Madaline sensitivity measure, a new learning algorithm (SBALR) [10] of Madalines is proposed, and it has performed well in learning. However, why is MRII (a previous learning algorithm for Madalines) poor in learning performance? This problem has still not been solved in theory. This paper tries to theoretically analyze MRII's disadvantage and to further improve it.

This paper presents an improving Madaline learning algorithm based on a Madaline sensitivity measure. The main contribution of the algorithm is that (1) it analyzes MRII's shortage in learning performance from the sensitivity point of view and points out that the confidence level in MRII cannot properly measure the output perturbation due to weight adaptation; (2) it proposes an adaptation selection rule by means of the sensitivity measure to improve MRII. The adaptation selection rule for neurons could more accurately locate the neurons and thus their weights in real need of adaptation during learning so as to better implement the minimal disturbance principle for greatly improving the learning performance of MRII.

Although both this paper and [10] take the Madaline sensitivity theory as the important theory, there are two main differences between them. (1) In this paper, the sensitivity is mainly taken as a measure to better locate the neurons in real need of adaptation during learning, while, in [10], the sensitivity is used to guide weight learning rule development; (2) in goal, this paper adopts the sensitivity theory to analyze MRII's shortages in performance and then to improve MRII, while [10] takes the sensitivity theory to guide learning rule design and thus to develop a completely new learning algorithm for Madalines independent of MRII.

The rest of this paper is organized as follows. In the next section, the Madaline model and its sensitivity are briefly described. Measures for evaluating the effects of weight adaptation are discussed in Section 3. An adaptation selection rule based on the Madaline sensitivity is proposed in Section 4. Following, in Section 5, is the new Madaline learning algorithm based on the rule. Experimental evaluations and results are given in Section 6. Finally, Section 7 concludes the paper.

2. Madaline Model and Sensitivity Measure

2.1. Madaline Model and Notations. A Madaline is a kind of binary multilayer feedforward neural network with a supervised learning mechanism and consists of a set of neurons, called Adalines, with binary input, output, and hard limit activation function. The input of an Adaline, which is represented by $X = (x_0, x_1, \dots, x_n)^T$ including an extra element $x_0 (\equiv 1)$ corresponding to a bias w_0 , is weighted by the weight $W = (w_0, w_1, \dots, w_n)^T$ containing the bias and then fed into an activation function $f(\cdot)$ to yield an output of the Adaline as

$$y = f(X^T W) = \begin{cases} -1 & X^T W < 0 \\ +1 & X^T W \geq 0. \end{cases} \quad (1)$$

Generally, a Madaline has L layers, and each layer l ($1 \leq l \leq L$) has n^l ($n^l \geq 1$) Adalines. The form of $n^0 - n^1 - \dots - n^L$ is used to represent the Madaline, in which each n^l ($0 \leq l \leq L$) not only stands for a layer but also indicates the number of Adalines in the layer. n^0 is an exception, which denotes the input dimension of the Madaline. For the j th layer, X^j ($1 \leq j \leq L$) denotes the input of all Adalines in the layer, and Y^j ($1 \leq j \leq L$) denotes the output of the layer. They meet $Y^{k-1} = X^k$ ($2 \leq k \leq L$). Particularly, X^1 denotes not only the input of all Adalines in the first layer but also the input of the entire Madaline; n^L denotes the output layer, and Y^L is the output of both the last layer and the entire Madaline.

It is well known that a network with a single hidden layer and enough hidden neurons is adequate to deal with all mapping problems [11]. For simplicity and without loss of generality, the following discussion only focuses on the Madalines with single hidden layer.

2.2. Madaline Sensitivity Measure. Usually, an adaptive supervised learning is a process of iterative weight adaptation until the input-output mapping indicated by a training data set is established. So, in each iteration, how to correctly locate the weights in real need of adaptation is a key issue for the success of a Madaline learning algorithm. In order to successfully locate the weights in need of adaptation, it is vital to analyze Madaline output variation caused by the weight adaptation. Since the study on Madaline sensitivity aims at exploring the effects of a Madaline weights' variation on its output, it is reasonable to investigate the sensitivity as a measure to locate the weights.

The following subsections will briefly introduce the latest research results on the Madaline sensitivity, which will be

employed as a technical tool to support the investigation of Madaline learning mechanism. For further details, please refer to [12–14].

2.2.1. Adaline Sensitivity

Definition 1. The sensitivity of an Adaline is defined as the probability of the Adaline's output inversion due to its weight variation with respect to all inputs, which is expressed as

$$s(\Delta W) = \frac{V_{\text{var}}}{V_n}, \quad (2)$$

where V_{var} is the number of inputs whose Adaline's output is inverted due to the Adaline's weight variation and V_n is the number of all inputs.

The research results have shown that the Adaline sensitivity can be approximately computed as

$$\begin{aligned} s(\Delta W) &\approx \widehat{s}(\Delta W) \\ &= \begin{cases} \frac{\arccos\left(\frac{(WW')}{(|W||W'|)}\right)}{\pi}, \\ \frac{(|\Delta W|/|W|)}{\pi}, \end{cases} \quad \text{for } |\Delta W| \ll |W|, \end{aligned} \quad (3)$$

where W , ΔW , and W' , respectively, refer to the original weight, the weight variation, and the varied weight.

Due to the information propagation between layers in a Madaline, the Adaline sensitivity will lead to the corresponding input variation of all Adalines in the next layer. So, the Adaline sensitivity to its input variation also needs to be taken into account. However, it can be easily tackled by transforming the input variation to an equivalent weight variation as

$$\begin{aligned} s(\Delta X) &\approx \widehat{s}(\Delta X) \\ &= \frac{\arccos\left(\frac{\sum_{i=0}^n w_i w'_i / \sum_{i=0}^n w_i^2}{\pi}\right)}{\pi}, \quad (4) \\ w'_i &= \begin{cases} -w_i & i \in \{j_1, j_2, \dots, j_K\} \\ w_i & \text{others,} \end{cases} \end{aligned}$$

where ΔX denotes the input variation in which only K input elements are varied and j_t ($1 \leq t \leq K$) denotes that the j_t th input element of the Adaline is varied, and its corresponding equivalent varied weight element is w'_i . n is the input dimension of the Adaline.

Usually, each weight element of an Adaline during training is in the same magnitude; thus, according to the study result of [11], (4) can further be simplified to

$$s(\Delta X) \approx \frac{\sqrt{4K/(n+1)}}{\pi} = \frac{(|\Delta X|/|X|)}{\pi} \quad \text{for } K \ll n. \quad (5)$$

2.2.2. Madaline Sensitivity. Based on the structural characteristics of Madalines and the sensitivity of Adalines, the sensitivity of a layer and a Madaline can separately be defined as follows.

Definition 2. The sensitivity of layer l ($1 \leq l \leq L$) is a vector in which each element is the sensitivity of the corresponding Adaline in the layer due to its input and weight variations, which is expressed as

$$S^l = (s_1^l, s_2^l, \dots, s_{n^l}^l)^T. \quad (6)$$

Definition 3. The sensitivity of a Madaline is the sensitivity of its output layer; that is,

$$S_{\text{net}} = S^L = (s_1^L, s_2^L, \dots, s_{n^L}^L)^T. \quad (7)$$

During training, it could be helpful to quantitatively evaluate the output variation of a Madaline due to its weight adaptation. Usually, there are two ways to evaluate the output variation. One is the number of inputs at which the Madaline output is varied; the other is the number of output elements whose values are varied before and after the weight adaptation. Apparently, for Madalines with a vector output, the latter can more truly reflect their output variation before and after the weight adaptation. Therefore, the sensitivity of a Madaline can be further quantified as follows:

$$s_{\text{net}} = \frac{\sum_{i=1}^{n^L} s_i^L \cdot V_n}{n^L \cdot V_n} = \frac{1}{n^L} \sum_{i=1}^{n^L} s_i^L, \quad (8)$$

where V_n is the number of all inputs.

From (8), the Madaline sensitivity is equal to the average of sensitivity values of all Adalines in the output layer.

3. Measures for Evaluating the Effects of Weight Adaptation

During the training of a Madaline, a weight adaptation will inevitably lead to its output variation. In order to make the Madaline obtain the desired output for the current input sample by weight adaptation and meanwhile meet the minimal disturbance principle, it is necessary to find a measure to evaluate if the effects of weight adaptation on the output of the Madaline are acceptable.

3.1. Sensitivity Measure. According to the above Madaline sensitivity definition, a Madaline output variation due to its weight adaptation is just the Madaline sensitivity due to its weight adaptation; that is,

$$\text{output variation} = s_{\text{net}}(\Delta W_i^l), \quad 1 \leq l \leq L, \quad 1 \leq i \leq n^l. \quad (9)$$

Considering the computation difference of the Adaline sensitivity between the hidden layer and the output layer, we divide the computation of (9) into the following two cases:

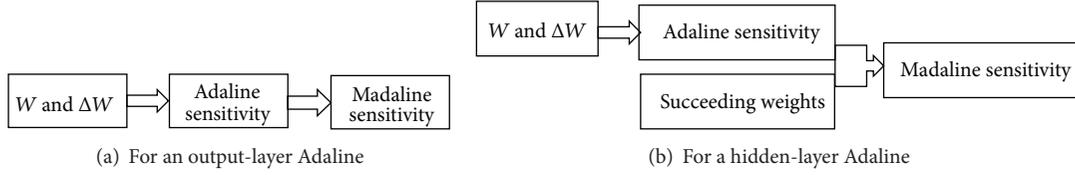


FIGURE 1: The computing process of Madaline sensitivity.

- (a) for the weight adaptation of the i th ($1 \leq i \leq n^L$) Adaline in the output layer, its sensitivity can be computed by (3) as

$$s_i^2(\Delta W_i^2) = \frac{\arccos\left(\frac{(W_i^2(W_i^2 + \Delta W_i^2))}{(|W_i^2||W_i^2 + \Delta W_i^2|)}\right)}{\pi}. \quad (10)$$

- (b) For the weight adaptation of the j th ($1 \leq j \leq n^1$) Adaline in the hidden layer, the input variation of Adalines in its succeeding layer will occur and this will propagate layer by layer to the output layer. Thus, the sensitivity of the hidden-layer Adaline due to its weight adaptation is firstly computed by (3) and the sensitivity of the Adalines in the output layer due to its corresponding input variation is computed by (4), and then the sensitivity of each Adaline in the output layer can be computed as

$$s_i^2(\Delta W_j^1) \approx s_j^1(\Delta W_j^1) s_i^2(\Delta X_i^2) = s_j^1\left(\frac{\arccos\left(\frac{(|W_i^2|^2 - 2w_{i,j}^2)/|W_i^2|^2}{\pi}\right)}{\pi}\right), \quad (11)$$

$$1 \leq i \leq n^2.$$

Based on the result of (10) or (11), the Madaline sensitivity due to its weight adaptation can be calculated by (8).

3.2. Confidence Level. In order to facilitate analysis, it is necessary to firstly introduce the weight adaptation rule in the MRIL, namely, Mays rule [5], as follows:

$$W' = W + d\eta\left(\frac{(R - dX^T W)}{(n+1)}\right)X \quad \text{for } |X^T W| < \delta, \quad (12)$$

where W , W' , and X , respectively, represent the original weight, the varied weight, and the current input of an Adaline; d is the desired output of the Adaline for the current input; η (> 0) and R (> 0), respectively, represent an adaptation constant and an adaptation level; n is the input dimension of the Adaline; and δ (> 0) is a dead zone value.

When the output of an Adaline needs to be reversed, it would have $dX^T W \leq 0$. So, according to Mays rule (12), it further has

$$\Delta W = d\eta\left(\frac{(R - dX^T W)}{(n+1)}\right)X = d\eta\left(\frac{(R + |X^T W|)}{(n+1)}\right)X. \quad (13)$$

In the MRIL, the absolute value of weighted input summation $|X^T W|$, called confidence level, is used as a measure to evaluate the effects of weight adaptation on Madaline output during training. It is obvious that the measure has some shortcomings for evaluating the effects because the value of $|X^T W|$ is only related to the current input and does not take all inputs into consideration. However, the Madaline sensitivity measure covers all inputs with no functional relation to any individual input. In this sense, the confidence level is a local measure for the network output variation at a given input, while the Madaline sensitivity is a global measure for all possible inputs.

From the sensitivity study, one could make further analysis about the shortcomings of the confidence level. The weight adaptation of an Adaline will directly affect the input-output mapping of the Adaline. If the input-output mapping varies, this variation will propagate through the network and finally may cause a variation of the input-output mapping of the Madaline. Since both Adaline sensitivity and Madaline sensitivity are only functions of W and ΔW , they can, respectively, reflect the output variations of Adalines and Madalines. According to (10) and (11), the network output variation due to the weight adaptation of an Adaline in a Madaline can be illustrated as in Figure 1.

However, according to (13), ΔW is an increasing function of the confidence level $|X^T W|$ under given parameters η , R , n , and X , and its direction is the same as dX . Unfortunately, W cannot be reflected by the confidence level $|X^T W|$ either in the magnitude or in the direction. So, it can be seen from Figure 1 that the confidence level $|X^T W|$ of an Adaline is unable to exactly reflect the output variation of the Adaline and thus the output variation of the corresponding Madaline based on weight adaptation rule (12). This shortcoming of the confidence level makes it unable to correctly guide the design of Madaline learning algorithm.

3.3. Simulation Verification for the Two Measures. In order to verify the correctness of the above theoretical analysis, computer simulations were carried out. A Madaline with the architecture of 10-5-1 and random weights was chosen. For

TABLE 1: Comparison of three different measures.

Hidden Adaline number	Measures		Number of varied output elements in simulation
	Confidence	Sensitivity	
1	2.214	0.023	26
2	0.058	0.008	5
3	1.429	0.054	64
4	1.685	0.022	28
5	2.959	0.038	47

each hidden-layer Adaline, from the first one to the last one, its weights were adapted by (12) (in which parameter δ was ignored), and then the corresponding values of the two measures (the confidence level $|X^T W|$ and the sensitivity measure) and the number of varied output elements due to the weight adaptation were computed and simulated. The experimental results are listed in Table 1.

According to the values of the two measures and simulation results in Table 1, all hidden-layer Adalines are queued in a sequence with an ascending order. Table 2 gives, in three rows, three sequences, in which the first one is regarded as the standard and each wrongly located Adaline in the other two sequences is marked with bold.

From Tables 1 and 2, one could find that the Madaline sensitivity measure is obviously superior to the confidence level. In Table 2, there are four wrong locations in the sequence of the confidence level $|X^T W|$ and two wrong locations in the sequence of the Madaline sensitivity. It could be further found, from Table 2, that the two Adalines wrongly located by Madaline sensitivity, namely, Adaline 4 and Adaline 1, are adjacent Adalines in the standard sequence. In addition, one could find from Table 1 that the actually varied output elements of them, 26 output elements for Adaline 1 and 28 output elements for Adaline 4, are very close. This slight mismatch of the Madaline sensitivity measure with simulation results may mainly come from the approximate computation of the sensitivity measure.

Tables 1 and 2 show that our conclusion drawn from the above theoretical analysis about the two measures is consistent with the result of the experiments, which further verifies the fact that the Madaline sensitivity is a more appropriate measure to evaluate the effects of weight adaptation on a Madaline output.

4. An Adaptation Selection Rule

For CFNNs, with the support of the steepest descent technique, all neurons take part in weight adaptation during training. However, because of Madalines' discrete features, the determination of which Adaline being in need of adaptation is more complicated.

For a Madaline, when output errors occur, the easiest way is to directly adapt the weights of the Adalines in the output layer whose outputs are in error. But it is well known that a single-layer neural network can handle only linearly separable problems. So, a precondition of being able to directly adapt the Adalines in the output layer is that the

TABLE 2: Hidden-layer Adaline sequence ordered by the actual variation and the two measures.

Measures	Sequence of Adalines in hidden layer
Number of varied output elements in simulation	2 → 1 → 4 → 5 → 3
Sensitivity measure	2 → 4 → 1 → 5 → 3
Confidence level	2 → 3 → 4 → 1 → 5

hidden-layer outputs must be linearly separable. If the precondition is not satisfied, it is impossible to train a Madaline to solve a nonlinearly separable problem by only adapting Adalines in output layer. For this consideration, in the layer level, the priority of adaptation would be given to Adalines in hidden layer. As the information flow in a Madaline is always one-way from the input layer to the output layer, it is apparent that the former layer would be in general prior to its succeeding layer in a Madaline with many hidden layers.

In the same hidden layer, when the network output error for the current input occurs, there may be many selections of Adaline for adaptation to reduce the error due to the binary feature of the Adaline output. Then, a question is how to select the Adaline or the Adaline combination that is really in need of adaptation for improving training precision. Actually, there are two aspects that need to be considered for the selection. One is that the adaptation of the selected Adaline(s) must be able to reduce output errors of the Madaline for the current input. This is easy to be judged by the following way, called "trial reversion": reverse the output(s) of the selected Adaline(s) and then compute the output of the Madaline to check if the number of output element errors for the current input is reduced. If it is, view this selection as a useful one. The other is that the adaptation of the selected Adaline(s) also must minimize the Madaline's output disturbance for all noncurrent inputs. According to the analysis in Section 3, a Madaline's output disturbance due to its weight adaptation can be properly evaluated by the Madaline sensitivity. So, Madaline sensitivity measure can be used to establish an adaptation selection rule as follows: "give priority of adaptation to the Adaline(s) that can reduce the output errors and meanwhile minimize the sensitivity measure."

In order to simplify the computation of the sensitivity measure during training, it is noted that the weight adaptation according to (12) is always a small real value; so, the constraint in (3) can be met. Besides, the constraint in (5) can be also met as long as the number of hidden-layer Adalines n^1 is more than one. Thus, (11) can be further simplified into

$$\begin{aligned}
 s_i^2(\Delta W_j^1) &= s_j^1(\Delta W_j^1) s_i^2(\Delta X_i^2) \\
 &\approx \left(\frac{2}{\pi^2}\right) \sqrt{\frac{1}{(n^1+1)}} \left(\frac{|\Delta W_j^1|}{|W_j^1|}\right). \quad (14)
 \end{aligned}$$

From (14), it can be seen that the sensitivity of an output-layer Adaline in a Madaline due to its hidden Adaline's weight adaptation only depends on the weight variation ratio, that is, $|\Delta W_j^1|/|W_j^1|$ ($1 \leq j \leq n^1$). Hence, by (8), the sensitivity

Input: A Madaline with given architecture and random initial weights, a set of training data, learning parameters R, η , and the requirements of training precision and the Maximal epochs.

- (1) Randomly arrange training samples;
- (2) Loop for all training samples starting with $1 \rightarrow i$:
 - (2.1) Feed the i th training sample into the Madaline;
 - (2.2) If the output of the Madaline is correct for the i th sample, $i + 1 \rightarrow i$, go to Step 2;
 - (2.3) For each hidden layer l, l from 1 to $L - 1$, do:
 - (2.3.1) Determine weight adaptations of all Adalines in the l th layer by (12), and then calculate values of their sensitivity measure by (9) or (15);
 - (2.3.2) Sort l th-layer Adalines according to their sensitivity measure values in ascending order;
 - (2.3.3) For j from 1 to $\lfloor n^l/2 \rfloor + 1$ do:
 - (2.3.3.1) For all possible adjacent Adaline combinations with j elements in the queue do:
 - ① Implement the trial reversion for the current Adaline combination;
 - ② If output errors of the Madaline don't reduce, reject the adaptation, and continue to do for next Adaline combination;
 - ③ Weight(s) of Adaline(s) in the current combination are adapted by (12); Count the Madaline's output errors.
 - ④ If the Madaline errors are equal to zero, $i + 1 \rightarrow i$, go to Step 2; Else $1 \rightarrow l$ and go to Step 2.3.
 - (2.4) For the k th-Adaline in output layer, k from 1 to n^l , do:
 - If the output of the k th Adaline isn't correct to the i th sample, its weight is adapted by (12).
 - (3) Go to Step 1 unless the training precision meets the given requirement for all training samples or training epochs reach the given number.

Output: all weights and training errors under all training samples.

ALGORITHM 1

measure for hidden-layer Adalines can be further simplified into

$$\text{the sensitivity measure} = \frac{|\Delta W_j^1|}{|W_j^1|}, \quad 1 \leq j \leq n^1. \quad (15)$$

5. New Madaline Learning Algorithm

A Madaline learning algorithm aims to assign proper weights to every Adaline so that the input-output mapping could be established to maximally satisfy all given training samples. The basic idea of the Madaline learning algorithm can be briefly described as follows. All training samples are iteratively trained one by one until output errors of the Madaline for all the samples are zero or meet a given precision requirement. Each time, one training sample is fed into the Madaline, and then selected weight adaptations are conducted in a layer from the first layer to the output layer until the output of the Madaline meets the desired output of the current sample. As to the selection of weights for adaptation, it can be treated by distinguishing two cases: the selection of Adaline(s) in a hidden layer and the selection of Adaline(s) in the output layer. In the former case, Adalines in the layer are selected to adapt according to the adaptation selection rule; in the latter case, those Adalines that have erroneous outputs are selected to adapt. The details of an adaptive learning algorithm for a Madaline based on its sensitivity measure can be programmed as shown in Algorithm 1.

6. Experimental Evaluations

Usually, the learning performance and the generalization performance are two main indexes to evaluate a learning algorithm. Due to the discrete characteristic of Madalines, MSE (mean square error) is no longer suitable to evaluate the learning performance of the learning algorithm of Madalines. Herein, instead of MSE, the sample success rate and the network convergence rate are used to evaluate the learning performance. The success rate is the percentage of successful training samples by a Madaline in training, while the convergence rate is the percentage of Madalines that reach a complete solution under specified requirements among a group of Madalines participating in training. Besides, the generalization rate that shows the percentage of the successful testing samples by a Madaline after training is used to evaluate the generalization performance.

To evaluate the efficiency of the proposed algorithm, some experiments are carried out using the algorithms, the MR II and the BP, respectively. In the experiments, Madalines and MLPs (multilayer perceptron) with a single hidden layer were organized to solve several representative problems, and 5 of them are chosen from UCI repository [15]. They are three Monks problems, two Led display problems, and the And-Xor problem. The Monks problems are Monks-1 with 124 training samples and 432 testing samples, Monks-2 with 169 training samples and 432 testing samples, and Monks-3 with 122 training samples and 432 testing samples. The Led display problems are Led-7 with 7 attributes and Led-24 with

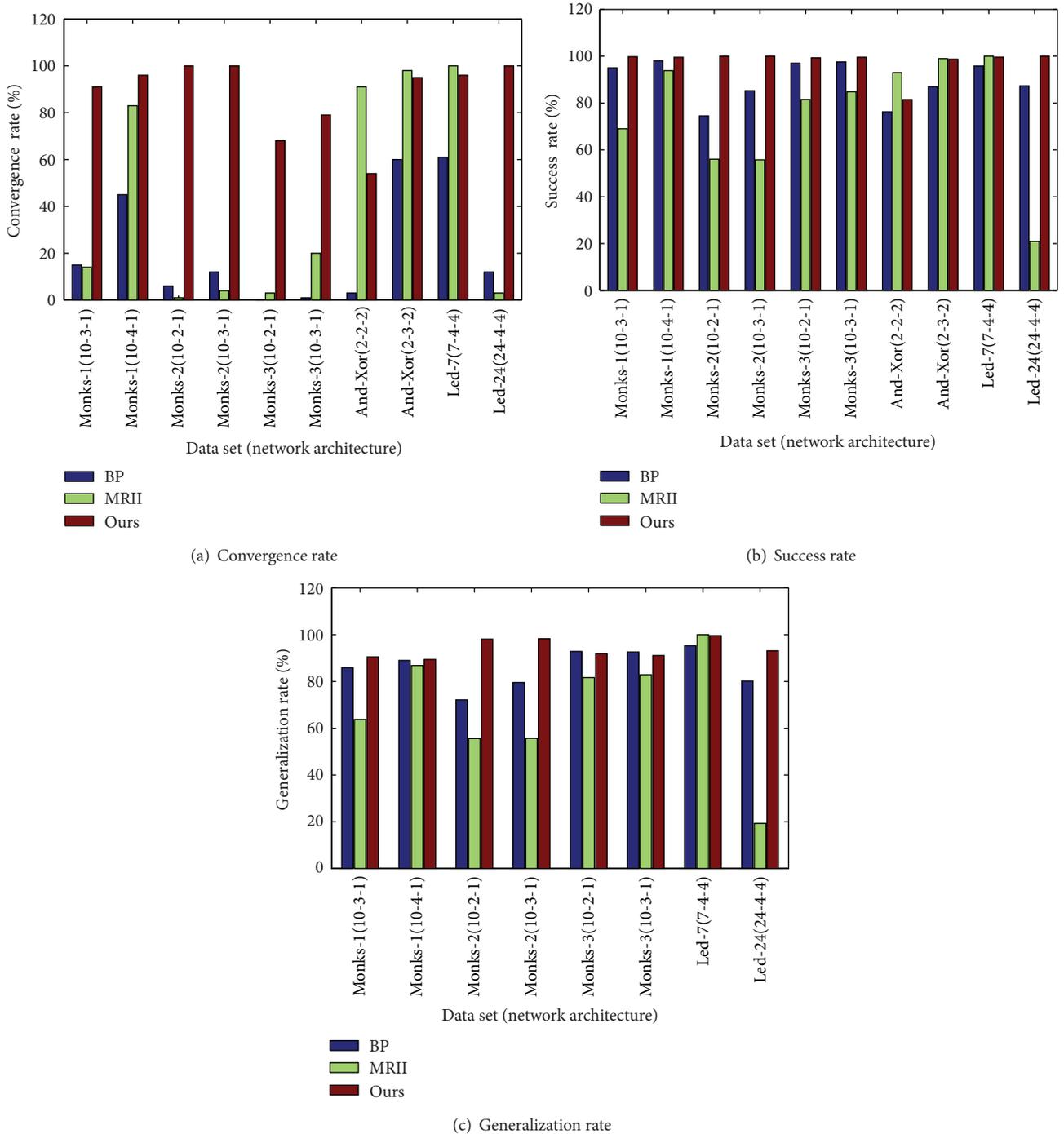


FIGURE 2: Performance comparison among BP, MRII, and our algorithm.

24 attributes; both of them have 200 training samples and 1000 testing samples, and the latter adds 17 irrelevant attributes on the basis of the former. The And-Xor with two inputs and two outputs is a representative nonlinear logical calculation problem, in which one output implements the “AND” calculation of two inputs while the other implements the “XOR” calculation of them. For each experiment, the training goal, that is, output error, was set to 0, and epochs of Monks problems, Led problems, and And-Xor problem were

set to 2000, 2000, and 200, respectively. Besides, for MLPs, the momentum gradient descent BP algorithm was used to train them.

In order to guarantee the validity of experimental results, all results presented in Figure 2 are the average of 100 runs’ results. Figure 2 shows that our algorithm has better performance than MRII not only on learning performance but also on generalization performance, especially for the difficult classification problems such as Led-24; only for several

relative simple problems such as And-Xor and Led-7, MRII has a good performance. Compared with BP algorithm, our algorithm also shows better learning performance and generalization performance, especially on convergence rate; only for monks-3 problem, BP algorithm is slightly better than our algorithm. The experimental results of Figure 2(a) show that the BP algorithm is rather poor on the convergence rate, which highlight the BP algorithm's shortage of easily falling into the local minimum.

7. Conclusion and Future Work

This paper presents a new adaptive learning algorithm for Madalines based on a Madaline sensitivity measure. The main focus of the paper is how to implement the minimal disturbance principle in the algorithm. An adaptation selection rule based on the sensitivity measure is proposed to carry out the minimal disturbance principle. Both theoretical analysis and experimental evaluations demonstrate that the sensitivity measure is superior to the confidence level used in the MRII. With the proposed adaptation selection rule, the algorithm can more accurately locate the weights in real need of adaptation. Experiments on some representative problems show that the proposed algorithm has better learning ability not only than that of the MRII but also than BP algorithm.

Although the proposed learning algorithm of Madalines has better performance, it is noticed that there still exist some weaknesses because of the usage of the Mays rule in the algorithm. One is that too many parameters need to be set in advance, which can hamper the application of Madalines. The other is that the Mays rule is unable to guarantee weight adaptation to exactly follow the minimal disturbance idea. In our future works, we will try to solve these two issues to develop a more perfect Madaline learning algorithm.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the Research Foundation of Nanjing University of Information Science and Technology (20110434), the National Natural Science Foundation of China (11361066, 61402236, and 61403206), Natural Science Foundation of Jiangsu Province (BK20141005), University Natural Science Research Program of Jiangsu Province (14KJB520025), a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions, and the Deanship of Scientific Research at King Saud University (RGP-264).

References

- [1] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, Madaline, and backpropagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415–1442, 1990.
- [2] E. D. Rumelhart, E. G. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, vol. 1, chapter 8, MIT Press, Cambridge, Mass, USA, 1986.
- [3] F. Rosenblatt, "On the convergence of reinforcement procedures in simple perceptrons," Cornell Aeronautical Laboratory Report VG-1796-G-4, Buffalo, NY, USA, 1960.
- [4] W. C. Ridgway, "An adaptive logic system with generalizing properties," Tech. Rep. 1557-1, Stanford Electron. Lab, Stanford, Calif, USA, 1962.
- [5] R. Winter, *Madalines rule II: a new method for training networks for adalines [Ph.D. thesis]*, Department of Electrical Engineering, Stanford University, 1989.
- [6] R. Winter and B. Widrow, "Madaline rule II: a training algorithm for neural networks," in *IEEE International Conference on Neural Networks*, vol. 1, pp. 401–408, San Diego, Calif, USA, July 1988.
- [7] C. H. Mays, "Adaptive threshold logic," Tech. Rep. 1556-1, Stanford Electronics Lab, Stanford, Calif, USA, 1963.
- [8] M. Frean, "The upstart algorithm: a method for construction and training feedforward networks," *Neural Computation*, vol. 2, no. 2, pp. 198–209, 1990.
- [9] J. H. Kim and S. K. Park, "Geometrical learning of binary neural networks," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 237–247, 1995.
- [10] S. Zhong, X. Zeng, S. Wu, and L. Han, "Sensitivity-based adaptive learning rules for binary feedforward neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 480–491, 2012.
- [11] N. E. Cotter, "The Stone-Weierstrass theorem and its application to neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 4, pp. 290–295, 1990.
- [12] S. Zhong, X. Zeng, H. Liu, and Y. Xu, "Approximate computation of Madaline sensitivity based on discrete stochastic technique," *Science China: Information Sciences*, vol. 53, no. 12, pp. 2399–2414, 2010.
- [13] Y. Wang, X. Zeng, D. S. Yeung, and Z. Peng, "Computation of Madalines' sensitivity to input and weight perturbations," *Neural Computation*, vol. 18, no. 11, pp. 2854–2877, 2006.
- [14] X. Zeng, Y. Wang, and K. Zhang, "Computation of Adalines' sensitivity to weight perturbation," *IEEE Transactions on Neural Networks*, vol. 17, no. 2, pp. 515–519, 2006.
- [15] <http://www.ics.uci.edu/~mllearn/MLRepository.html>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

