Hindawi
Computational Intelligence and Neuroscience
Volume 2018, Article ID 9390410, 22 pages
https://doi.org/10.1155/2018/9390410

Research Article

A Pruning Neural Network Model in Credit Classification Analysis

Yajiao Tang,[1,2] Junkai Ji,[1] Shangce Gao,[1] Hongwei Dai,[3] Yang Yu,[1] and Yuki Todo[4]

[1]*Faculty of Engineering, University of Toyama, Toyama-shi 930-8555, Japan*
[2]*College of Economics, Central South University of Forestry and Technology, Changsha 410004, China*
[3]*School of Computer Engineering, Huaihai Institute of Technology, Lianyungang 222005, China*
[4]*School of Electrical and Computer Engineering, Kanazawa University, Kanazawa-shi 920-1192, Japan*

Correspondence should be addressed to Shangce Gao; gaosc@eng.u-toyama.ac.jp

Received 8 November 2017; Revised 11 January 2018; Accepted 14 January 2018; Published 11 February 2018

Academic Editor: Pietro Aricò

Nowadays, credit classification models are widely applied because they can help financial decision-makers to handle credit classification issues. Among them, artificial neural networks (ANNs) have been widely accepted as the convincing methods in the credit industry. In this paper, we propose a pruning neural network (PNN) and apply it to solve credit classification problem by adopting the well-known Australian and Japanese credit datasets. The model is inspired by synaptic nonlinearity of a dendritic tree in a biological neural model. And it is trained by an error back-propagation algorithm. The model is capable of realizing a neuronal pruning function by removing the superfluous synapses and useless dendrites and forms a tidy dendritic morphology at the end of learning. Furthermore, we utilize logic circuits (LCs) to simulate the dendritic structures successfully which makes PNN be implemented on the hardware effectively. The statistical results of our experiments have verified that PNN obtains superior performance in comparison with other classical algorithms in terms of accuracy and computational efficiency.

1. Introduction

In the past few decades, credit classification has been continuously attracting a great deal of attention from academic researchers and financial institutions, resulting in various algorithms, known as the credit classification models [1]. Credit classification fits for predicting potential risk corresponding to credit portfolio; thus, it plays a fundamental role for financial institutions to improve their liquidity and reduce any possible risk [2]. Concerning the financial institutions' profitability, the possibility of differentiating good applicants from bad ones correctly is extremely important and urgent [3]. Even we can conclude that it is significant to make the accurate credit granting decision because any little improvement such as even a percent fraction in accuracy could be converted into a large future saving for the financial institutions [4].

In general, credit classification approach is used to classify applicants (including individuals and companies) into either good (with credit accepted decisions) or bad (with credit rejected decisions). It is based on the applicants' information such as individual's monthly income, bank balance, vocation, family status, educational background and company's balance sheets, financial ratios, and capital mobility. In detail, good applicants are creditworthy and more capable of repaying loan, bad applicants are not creditworthy, and their capability of loan repayment is low. Consequently, credit analysts have to undertake the responsibility to gather and analyze the relevant information about the loan applicants [5]. As we know, proper credit classification model can reduce credit analysis costs, provide quicker decisions, and reduce potential risk [6]. However, credit classification is a tough task because it is difficult to separate the credit data correctly by using the ordinary approach [7]. The properties of the credit assessment always cause heterogeneity and asynchrony of the information transmitted to the applicants and analysts [8]. Therefore, credit classification always results in high misclassification rate beyond the extent that people normally consider acceptable.

Credit classification techniques are usually estimated through three properties, namely, accuracy, interpretability, and computational efficiency [9]. Accuracy is the essential requirement which represents that the maximum possible number of correct decisions can be generated. And a minor improvement of accuracy means a significant saving for a financial institution. The interpretability is quite important to not only decision makers but also credit applicants, since it represents the ability to generate an understandable evaluation mechanism to the applicants, which includes the choice of the most essential input attributes of the analysis model in the meantime. The computational efficiency represents the speed of classification. It is helpful for the assessors to make the decision whether credit should be granted or not as quickly as possible according to the classification result. Therefore, the credit classification model which owns the above-mentioned properties can be considered as an appropriate tool in the business and finance fields, especially under the conditions with high uncertainty.

Till now, based on particular computation patterns, several models which include artificial neural networks (ANNs) [10–12], decision tree [13], expert systems [14], and genetic algorithms [15, 16] have been proposed to classify bank credit applicants. Among them, the adoption of ANNs in bankruptcy prediction has been studied since the 1990s [17–19]. In 1991, it was the first time for a literature using the neural network to set up the credit classification model and the relative analysis [10, 20, 21]. During the similar period, ANNs have been applied to credit risk assessment and consumer credit scoring research [10, 20, 21]. According to the previous literatures, the general conclusion has been that ANNs outperformed conventional statistical algorithms and inductive learning methods [22]. However, as credit classification methods, the classical ANNs still suffer from the following disadvantages: firstly, they are easy to be trapped into local minimum; thus, sometimes they will cause extraordinarily incorrect experimental results to the financial decision makers and lead to extremely unsatisfactory choices, which will bring a lot of risks for the financial institutions to make the appropriate credit decisions and better investment decisions [23]. Secondly, they are often referred to as black boxes because it is difficult to interpret how the results are concluded [6]. In other words, ANNs are lack of interpretability. Last but not least, most ANNs studies thus far have adopted only limited datasets because when dealing with the high-dimensional credit classification problems, the structures of ANNs are quite large which makes ANNs become time-consuming and thus influence the timeliness of financial decisions [6].

Besides, Lim and Sohn have argued that a single classification algorithm is ineffective because it is not capable of distinguishing the slight differences of various customers [24]. Therefore, in their opinions it is necessary to introduce hybrid classification algorithms to credit classification so as to make the right judgement. However, in terms of average prediction accuracy, it is worth mentioning that some scholars have pointed out that multiple neural networks classifiers are not always superior to a single best neural networks classifier in many cases [25]. This offers us the inspiration to apply a single neural network model to make credit classification.

In order to overcome the above shortcomings of ANNs, we propose a pruning neural network (PNN) which is inspired by the synaptic nonlinearity of biological neural models. It has three layers. Firstly, the axons of the other neurons transform the signals to the synaptic layer; then the interaction of the synaptic signals transfers to every branch of the dendrites; afterwards, the outputs of the dendritic layer are collected and sent to the soma body. Since PNN is a feedforward neural model, we adopt the error back-propagation algorithm to train it. During the training process, PNN owns the neuronal pruning function to eliminate the unnecessary synapses and superfluous dendrites; thereafter a tidy dendritic morphology will be formed without sacrificing the classification accuracy. Furthermore, we replace the final dendritic structure by logic circuits which are in the form of comparators and logic NOT, AND, and OR gates. Thus, the results of applying PNN on credit classification can be easily implemented in hardware. And fast computational speed will make PNN become a suitable tool for the financial institutions. Experiments are conducted based on Australian and Japanese credit datasets and the results verify that PNN can classify the credit applicants effectively and efficiently in terms of accuracy rate, sensitivity, specificity, and the area under the operating characteristic curve (AUC).

The remaining of the paper is constructed as follows. Section 2 presents a review of the relative algorithms in credit classification. Section 3 introduces the proposed neural model PNN in detail. The learning algorithm of PNN is described in Section 4. Section 5 presents the experimental results of PNN in comparison with other algorithms through using UCI machine learning datasets, namely, Australian and Japanese credit datasets. Section 6 is dedicated to the discussion. Section 7 concludes this paper.

2. Related Works

In the modern studies of neural networks, the McCulloch-Pitts neuron model has been extensively applied [26]. Concretely, the synapses and dendrites are independent of each other and there is no effect on them from one to the other. In its basic unit, each input vector is multiplied by a weight value, and then the result passes through a threshold gate with nonlinearity (see Figure 1). The prevailing view concluded from the previous biological neural networks' literatures has revealed that the brain has great computational capability because of the complicated connectivity of neural networks which implies that McCulloch-Pitts' model is oversimplified to deal with complicated computation [27]. Various modes of synaptic and dendritic plasticity and nonlinearity mechanisms endue the synapses and dendrites the ability to play a significant role in the computation [28]. Individual neuron could act more powerfully whenever the synaptic nonlinearities in a dendrite tree are considered [29–32]. Furthermore, recent research has identified that the already-known

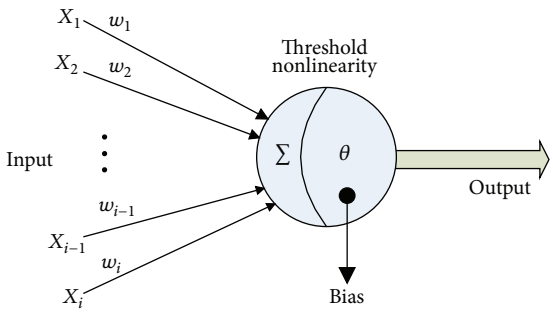

Figure 1: McCulloch-Pitts neuron model.

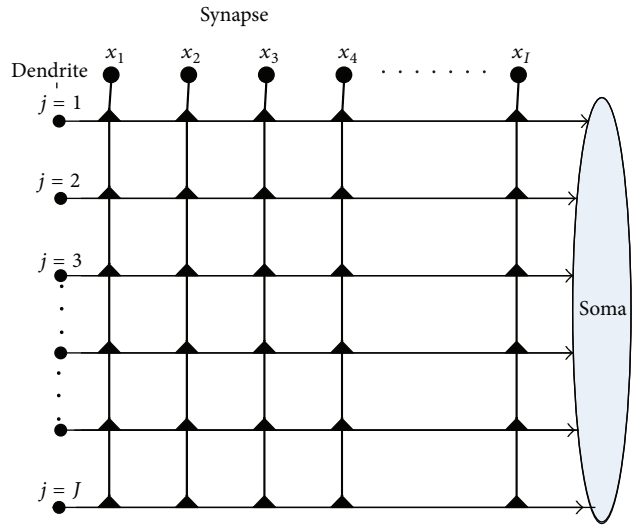

Figure 2: The architecture of PNN.

neurons all hold a unique shape of dendrite tree [33]. And a small morphological difference would result in a large functional variation. Mode-specific dendritic morphology has its important functional implication in determining what signals a neuron would receive and how these signals would integrate [33].

However, dendritic computation mechanism can provide concrete explanation on targeting the synaptic inputs at the appropriate locations [34]. To be more exact, in the early stage, redundant synapses and dendrites are found in the neural system, while the unnecessary ones will soon be filtered out and the rest will be strengthened and fixed, then a ripened neural network function will be formed [35]. These phenomena offer us the train of thought to propose our model.

According to the measurements made by adopting histological theories, Koch et al. have revealed that the interactions between excitatory synapses and inhibitory synapses have strong nonlinearity, and shunting inhibitory inputs can specifically occlude an excitatory input if they locate on the same path to the soma directly [29]. They have posited that the interactions among synapses and the response at the connection point of a branch could be thought of as logic operations [36]. Nevertheless, their model cannot distinguish whether the excitatory or inhibitory synapse is kept, where it is located, and which branch of dendrite needs to be strengthened [37]. Hence, Koch et al. have pointed out that we need a learning algorithm which is based on the plasticity in dendrites to answer the above questions [38]. It is worth mentioning that, in biological pyramids neurons, manifold plasticity mechanisms have been identified [39–41]. It benefits us to understand the role of plasticity in ANNs. In our previous research [42–45], the well-evolved neurons can be approximately substituted by a logic circuit which is simply composed of the so-called comparators and logic NOT (negation), AND (conjunction), and OR (disjunction) gates according to Boolean algebra. Meanwhile, the locations and types of synapses on the dendrite branches will be formulated by learning [46]. And extra and useless synaptic and dendritic connections would be removed so as to enhance the efficiency of the neurological system [47]. These perspectives and research findings are helpful for us to yield a more realistic model which adopts the single neuron computation to solve the linearly nonseparable problems and improve the neuronal pruning mechanism and then apply it to solve some practical problems such as classifying the credit applicants.

3. Proposed Model

Inspired by biological neurons, we build up a novel single neuronal structure with dendrites, namely, PNN. PNN has three layers: a synaptic layer, a dendritic layer, and a soma layer, which are shown in Figure 2. The inputs (x_1 to x_I) which come from the axons of the prior neurons will enter the synaptic layer. Then, the interactions of the synaptic signals occur on each branch of dendrites. After that, the interactions will be collected and sent to the soma body. The mathematical expressions of PNN are depicted as follows.

3.1. Synaptic Layer. The synaptic layer is the region where nerve impulses are transmitted and received among neurons, encompassing the axon terminal of a neuron where neurotransmitters are released in response to an impulse. A synaptic connection to the dendrites of a neuron is implemented by its receptors which have a certain pattern of the specific ion. When the receptors receive an ion, the potential of the receptors converts and determines whether the connection synapse is excitatory or inhibitory [55]. The direction of the flow in the synaptic layer is feed-forward, which always begins from a presynaptic neuron to a postsynaptic neuron. And the equation of the jth ($j = 1, 2, 3, \ldots, J$) synaptic layer receiving the ith ($i = 1, 2, 3, \ldots, I$) input is expressed as follows:

$$Y_{ij} = \frac{1}{1 + e^{-k(w_{ij}X_i - q_{ij})}}, \tag{1}$$

where k denotes a positive constant, X_i is the input of the synapse, and w_{ij} and q_{ij} are synaptic parameters that need to

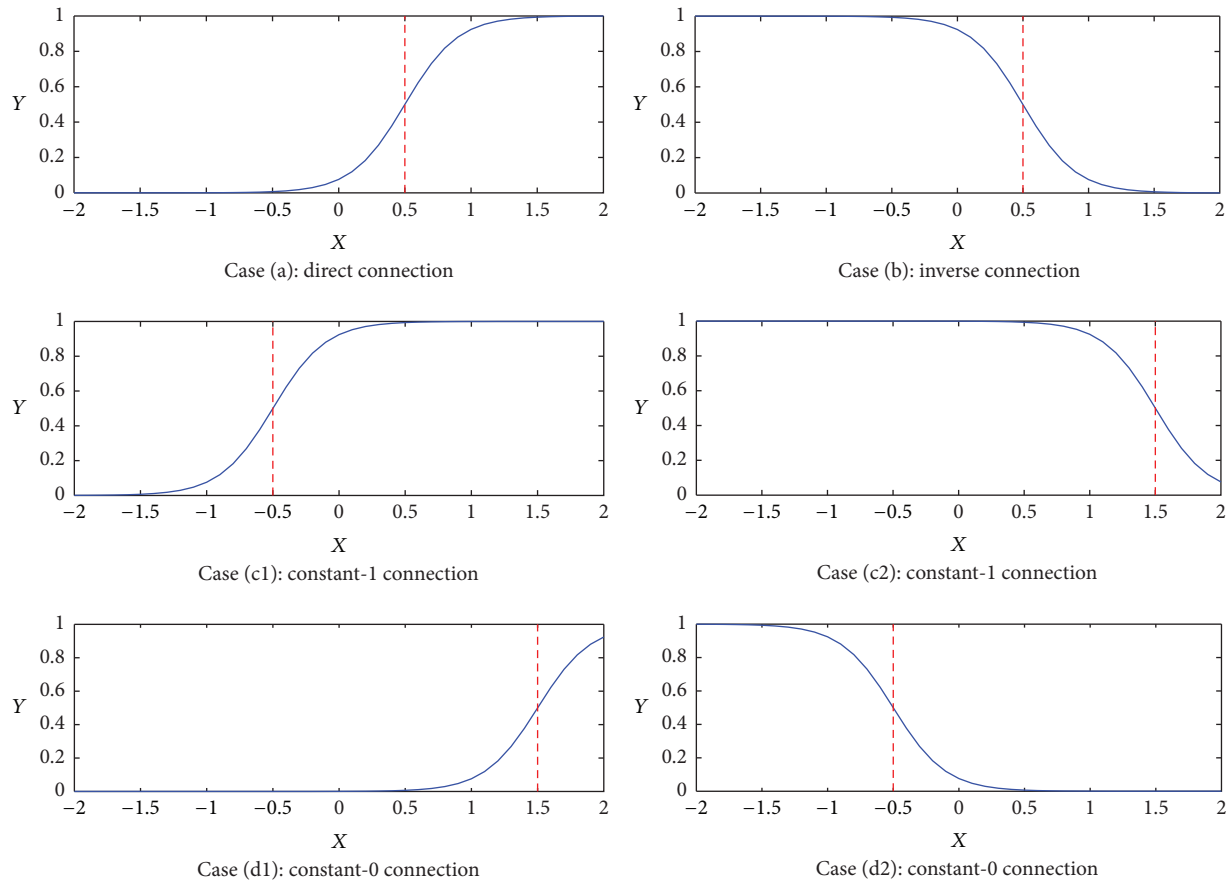

Figure 3: Six connection cases of the synaptic layer.

be trained. We use θ_{ij} to represent the threshold of a synaptic layer, which can be calculated in the following:

$$\theta_{ij} = \frac{q_{ij}}{w_{ij}}. \tag{2}$$

Depending on different values of w_{ij} and q_{ij}, there will appear four kinds of connection states: a direct connection, a reverse connection, a constant-1 connection, and a constant-0 connection as shown in Figure 3.

3.1.1. Direct Connection. Case (a): $0 < q_{ij} < w_{ij}$: for example, $q_{ij} = 0.5$ and $w_{ij} = 1.0$. As shown in Figure 3, Case (a), it corresponds to a direct connection. Once $x_i > \theta_{ij}$, the output Y_{ij} converges towards 1 which shows that when the input owns higher potential in comparison with the threshold θ_{ij}, the synapse turns to be excitatory which will depolarize the soma body. And when $x_i \leq \theta_{ij}$, the corresponding output will tend to 0 which represents that once the input possesses low potential, the synapse will change into inhibitory which will hyperpolarize the soma body transiently.

3.1.2. Inverse Connection. Case (b): $w_{ij} < q_{ij} < 0$: for example, $q_{ij} = -0.5$ and $w_{ij} = -1.0$. As shown in Figure 3, Case (b), it leads to an inverse connection. Once $x_i > \theta_{ij}$, the output Y_{ij} approximates to 0 which shows that when the input possesses low potential compared with its thresholds, the synapse turns to be inhibitory. And it will hyperpolarize the soma layer transiently. And, on the contrary, when $x_i \leq \theta_{ij}$ happens, the output Y_{ij} approximates to 1 which represents that when the input is of high potential, the synapse will become excitatory, and it will depolarize the soma layer.

3.1.3. Constant-1 Connection. There are two cases in the constant-1 connection: Case (c1): $q_{ij} < 0 < w_{ij}$: for example, $q_{ij} = -0.5$ and $w_{ij} = 1.0$. Case (c2): $q_{ij} < w_{ij} < 0$: for example, $q_{ij} = -1.5$ and $w_{ij} = -1.0$. Case (c1) and Case (c2) shown in Figure 3 represent the constant-1 connection. In these cases, no matter whether the input signal x_i exceeds the threshold θ_{ij}, the corresponding output tends to 1 all the time. In other words, the signals from the synaptic layer have little impact on the dendritic layer. When the excitatory input signals transport, depolarization will occur in the next soma layer.

3.1.4. Constant-0 Connection. The constant-0 connection also contains two cases: Case (d1): $w_{ij} < 0 < q_{ij}$: for example, $q_{ij} = 0.5$ and $w_{ij} = -1.0$. Case (d2): $0 < w_{ij} < q_{ij}$: for example, $q_{ij} = 1.5$ and $w_{ij} = 1.0$. Case (d1) and Case (d2) illustrated in Figure 3 denote the constant-0 connection. Regardless of the numeric value of the input, the outputs always approximate to

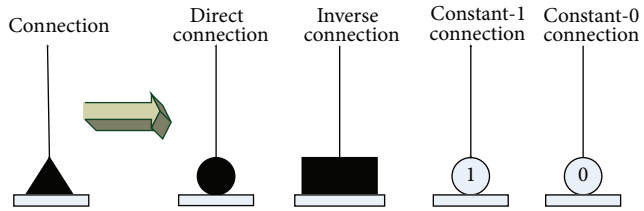

Figure 4: Four connection states of the synaptic layer.

0. The signals from the synaptic layer always degenerate the output signals into an inhibitory one.

The values of w_{ij} and q_{ij} are initialized randomly between −1.5 and 1.5. It means that the synapses are connected to each dendritic branch with randomly chosen connection cases. After being trained by learning algorithms, the values of w_{ij} and q_{ij} are changed, and the corresponding connection case of synapses will be changed at the same time. Figure 4 shows these four connection cases of synapses in our model's structure: a direct connection (•), an inverse connection (■), a constant-1 connection (①), and a constant-0 connection (⓪).

3.2. Dendrite Layer. A dendrite layer stands for the typical nonlinear interaction of synaptic signals on each branch of dendrites. Since the multiplication operation plays an important role in the process of transferring and disposing neural information, the nonlinearity calculation among the synapses on a dendrite can be implemented by a typical multiplication instead of summation. Thus, the interaction among synapses on a dendritic branch corresponds to a logic AND operation. The corresponding equation of the dendrite layer is defined as follows:

$$Z_j = \prod_{i=1}^{I} Y_{ij}. \tag{3}$$

3.3. Soma Layer. A soma layer accumulates the summation of the dendritic signals from each dendritic layer. Its function is thought to be the same as a logic OR operation approximately. This logic OR operation implies that the soma body will generate the value 1 when at least one of the variables is equal to 1. Its equation is shown as follows:

$$O = \sum_{j=1}^{J} Z_j. \tag{4}$$

3.4. Neuronal Pruning Function. Pruning technique means the removal of the superfluous nodes and weights through learning and training the neural network [56]. In our neural model, pruning function can be achieved by eliminating unnecessary synapses and dendrites. And a simplified and unique neural structure will be formed for each specific problem. Neuronal pruning function of our model contains two parts: synaptic pruning and dendritic pruning.

Synaptic Pruning. When the input transmits to the synaptic layer which is in the constant-1 connection case, the synaptic output is always 1, because the result of any arbitrary value multiplying 1 equals itself in the dendrite layer. It is obvious that the synaptic input in constant-1 connection has little impact on the output of the dendrite layer. Therefore, this kind of synaptic input could be absolutely neglected.

Dendritic Pruning. If the input transmits to the synaptic layer which is in the constant-0 connection case, the output is always 0. Consequently, the output of the corresponding dendrite layer also becomes 0 because of the multiplication operation. It means that this entire dendrite layer should be omitted because it has little influence on the soma layer.

An example of a synaptic and dendritic pruning procedure is illustrated in Figure 5. The original structure is composed of four synaptic layers and two dendrite layers as shown in Figure 5(a). Since the connection case of input x_1 is ① in Dendrite-1 layer, this synaptic layer can be deleted. The connection case of input x_3 is ⓪ in the Dendrite-2 layer; the whole Dendrite-2 can be completely omitted because of the dendritic pruning function. The unnecessary synaptic layers and dendrite layers which could be removed are shown in dotted lines as shown in Figure 5(b). The final simplified dendritic morphology is shown in Figure 5(c), in which only a synaptic layer and a dendritic layer are retained.

4. Learning Algorithm

As all the equations of PNN are differential, the error backpropagation algorithm (BP) is valid to be utilized as the learning algorithm. The BP algorithm adjusts the values of w_{ij} and q_{ij} to cut down the differences between the actual output O and desired output T. The Least Squared Error (LSE) between the actual output and desired output is defined in (1):

$$E = \frac{1}{2}(T - O)^2. \tag{5}$$

In PNN, the error minimization is realized by modifying the connection parameters in the negative gradient direction during the learning process. Hence, the differential changes of these connection parameters should be collected as shown in the following equations:

$$\begin{aligned} \Delta w_{ij} &= -\eta \frac{\partial E}{\partial w_{ij}}, \\ \Delta q_{ij} &= -\eta \frac{\partial E}{\partial q_{ij}}, \end{aligned} \tag{6}$$

where η denotes the learning rate and it is always set to be a positive constant. However, a low learning rate makes the convergence speed very slow, whereas a high learning rate makes the error become very difficult to converge to a certain connection pattern. The updating rules for connection parameters w_{ij} and q_{ij} are as follows:

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \Delta w_{ij}(t), \\ q_{ij}(t+1) &= q_{ij}(t) + \Delta q_{ij}(t), \end{aligned} \tag{7}$$

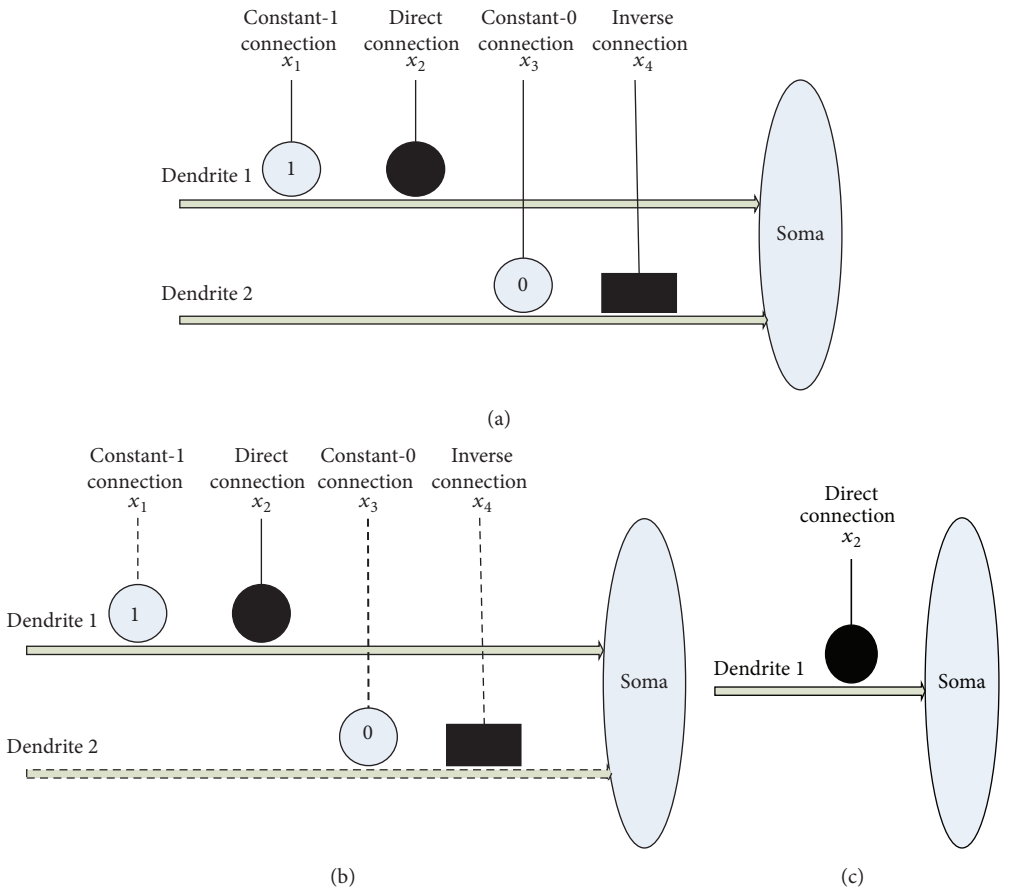

Figure 5: An example of a synaptic and dendritic pruning procedure.

where t represents the current learning epoch. Moreover, the partial differentials of E with respect to w_{ij} and q_{ij} are computed in the following:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial Z_j} \cdot \frac{\partial Z_j}{\partial Y_{ij}} \cdot \frac{\partial Y_{ij}}{\partial w_{ij}},$$
$$\frac{\partial E}{\partial q_{ij}} = \frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial Z_j} \cdot \frac{\partial Z_j}{\partial Y_{ij}} \cdot \frac{\partial Y_{ij}}{\partial q_{ij}}. \tag{8}$$

The following shows the components of the above-mentioned partial differential:

$$\frac{\partial E}{\partial O} = O - T,$$
$$\frac{\partial O}{\partial Z_j} = 1,$$
$$\frac{\partial Z_j}{\partial Y_{ij}} = \prod_{L=1, L\neq i}^{I} Y_{Lj},$$
$$\frac{\partial Y_{ij}}{\partial w_{ij}} = \frac{k x_i e^{-k(X_i w_{ij} - q_{ij})}}{\left(1 + e^{-k(X_i w_{ij} - q_{ij})}\right)^2},$$
$$\frac{\partial Y_{ij}}{\partial q_{ij}} = \frac{-k e^{-k(X_i w_{ij} - q_{ij})}}{\left(1 + e^{-k(X_i w_{ij} - q_{ij})}\right)^2}. \tag{9}$$

5. Simulation

5.1. Credit Dataset Description. In this experiment, we have adopted two benchmark datasets, namely, the Australian and Japanese credit datasets (all from the UCI repository), to test different classification models. With a good mixture of different attributes which includes not only continuous but also nominal attributes with both small and large numbers of values, these two real world datasets are very meaningful to financial decision makers and managers. The details of the attributes can be found from the UCI repository [57].

Australian credit dataset is used to classify credit card applications and it contains 690 examples that record the applicants' data. This dataset contains 307 examples of creditworthy applicants ("good" and "accepted") and 383 examples which are not creditworthy ("bad" and "rejected"). Each instance consists of 8 categorical and 6 numerical input attributes. In order to protect the secrecy of the credit applicants, the applicants' names and values of the attributes have been converted to meaningless symbols.

Table 1: Attributes for evaluating credit risk in the Australian credit dataset.

Attributes	Type	Values (after preprocessing)	Values (before preprocessing)
A1	Categorical	0, 1	a, b
A2	Numerical	13.75–80.25	13.75–80.25
A3	Numerical	0–28	0–28
A4	Categorical	1, 2, 3	p, g, gg
A5	Categorical	1, 2, 3, . . . , 14	ff, d, i, k, j, aa, m, c, w, e, q, r, cc, x
A6	Categorical	1, 2, 3, . . . , 9	ff, dd, j, bb, v, n, o, h, z
A7	Numerical	0–28.5	0–28.5
A8	Categorical	0, 1	t, f
A9	Categorical	0, 1	t, f
A10	Numerical	0–67	0–67
A11	Categorical	0, 1	t, f
A12	Categorical	1, 2, 3	s, g, p
A13	Numerical	0–2000	0–2000
A14	Numerical	0–100,000	0–100,000
Class	Categorical	0, 1	−, +

Table 2: Attributes for evaluating credit risk in the Japanese credit dataset.

Attributes	Type	Values (after preprocessing)	Values (before preprocessing)
A1	Categorical	0, 1	a, b
A2	Numerical	13.75–80.25	13.75–80.25
A3	Numerical	0–28	0–28
A4	Categorical	1, 2, 3, 4	u, y, l, t
A5	Categorical	1, 2, 3	g, p, gg
A6	Categorical	1, 2, 3, 4, . . . , 14	c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff
A7	Categorical	1, 2, 3, 4, . . . , 9	v, h, bb, j, n, z, dd, ff, o
A8	Numerical	0–28.5	0–28.5
A9	Categorical	1, 0	t, f
A10	Categorical	1, 0	t, f
A11	Numerical	0–67	0–67
A12	Categorical	1, 0	t, f
A13	Categorical	2, 3, 1	g, p, s
A14	Numerical	0–2000	0–2000
A15	Numerical	0–100,000	0–100,000
Class	Categorical	0, 1	−, +

Japanese credit dataset also contains 690 instances, which are classified into two groups. Among them, 307 are labeled as class "+" and the rest 383 are labeled as "−." And each sample is characterized by 6 numerical and 9 categorical features. Similar as Australian credit dataset, the applicants' names and the attribute values have been converted to meaningless symbols to protect the data confidentiality.

5.2. Data Preprocessing. Data preprocessing is the first and crucial step to make data analysis. The classification task would be misleading and redundant if the data are not understood and considered completely in advance. Firstly, it sometimes shows a few missing values in the dataset, and the majority of learning algorithms are lack of the ability to handle the datasets with missing values. It needs us to utilize some methods to replace them [58]. In our experiments, we replace the numerical attribute with the average values and categorical ones with the mode of attributes, respectively.

Secondly, some learning algorithms such as ANNs require that each data sample is expressed as a real number vector. Thus, we need to transform the categorical attributes into numerical ones before we input them into the classifier. The attribute information of Australian credit dataset has been changed for the convenience of statistics. For example, the fourth attribute of this dataset has 3 labels, namely, "p," "g," and "gg." And these labels have been changed to 1, 2, and 3 in our experiments. According to this method, all the categorical attributes of Australian credit dataset and Japanese credit dataset have been changed, which are presented in Tables 1 and 2, separately.

Last but not least, for the sake of preventing the large numerical attributes from dominating those with small numerical values, all the numerical values should be

Table 3: Contingency matrix of prediction results.

Hypothesis class	Real class	
	Positive	Negative
Positive	True positive (TP)	False positive (FP)
Negative	False negative (FN)	True negative (TN)

normalized. In general, all the attributes are normalized to a range of $[0, 1]$ with a min-max normalization rule. And the min-max normalization procedure uses a linear transformation to change the original input range into a new specified range, which can be shown in the following equation:

$$x_{\text{normalized}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{10}$$

5.3. Performance Measures. In our experiments, overall accuracy rate, true positive rate, true negative rate, and AUC which is the area under the receiver operating characteristic curve (ROC) are utilized to construct the performance evaluation system. Firstly, the classification accuracy rate is regarded as one of the most popular classification performance metrics. It is measured by using the following equation:

$$\text{Accuracy rate} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} (\%), \tag{11}$$

where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively. True positive (TP) indicates the number of the instances which are predicted as creditworthy and their corresponding teacher target labels are creditworthy too. True negative (TN) represents the number of the instances whose prediction label and teacher target label are uncreditworthy at the same time. And false positive (FP) denotes the number of the samples which are detected as uncreditworthy, while the teacher target label is creditworthy. On the contrary, false negative (FN) stands for the number of the samples which are detected as creditworthy, but their teacher target labels are not. The results of a classifier containing TP, TN, FP, and FN can be measured by a 2-dimensional contingency matrix, which is demonstrated in Table 3.

Sensitivity and specificity are also important performance metrics in classification problems. Sensitivity measures the percentage that actual positives are correctly identified. It implies how successfully a classifier can identify the normal records which means that the applicants are creditworthy in the case of credit classification. Therefore, financial institutions can reduce their possible financial losses by adopting the classifier with higher sensitivity. Specificity measures the number of the observed bad applicants occupying a certain proportion of the total number of the observed bad applicants and those classified as bad. Thus, it represents how successfully a classifier can distinguish the abnormal records, so it means the proportion of true negative. Higher specificity can help the financial institutions to reduce the possibility of accepting the applicants with bad credit. And the expressions of sensitivity and specificity are shown as follows:

$$\begin{aligned} \text{Sensitivity} &= \frac{\text{TP}}{\text{TP} + \text{FN}} (\%), \\ \text{Specificity} &= \frac{\text{TN}}{\text{TN} + \text{FP}} (\%). \end{aligned} \tag{12}$$

In this study, AUC is also designed as significant metric to evaluate the model. And it can be calculated from the graph in which the sensitivity is plotted on the y axis and specificity is plotted on x axis, respectively. AUC reveals the difference between the classification groups predicted by a classifier. In other words, a score of 100% indicates that two classes can be perfectly discriminated by the classifier, while a score of 50% illustrates that the classifier owns insignificant discriminatory quality. The value of AUC can be demonstrated as follows:

$$\text{AUC } (\%) = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}}\right) \times 100 \ (\%). \tag{13}$$

Besides, in order to compare the convergence speed of different classification algorithms, the mean squared error (MSE) of PNN and MLP at each iteration is calculated by the following equation:

$$\text{MSE} = \frac{1}{R}\sum_{a=1}^{R}\left[\frac{1}{S}\sum_{b=1}^{S}\left(E_{ab} - O_{ab}\right)^2\right], \tag{14}$$

where E_{ab} and O_{ab} represent the predicted output and the actual output separately. S is the number of instances applied for training. R denotes the running times of the experiments which is set to be 30 to classify both Australian and Japanese credit datasets in our experiments.

5.4. Optimal Parameters Setting. Three user-defined parameters are considered to be sensitive to the classification performance of PNN, namely, k, η, and M. k represents a constant of the sigmoid function in the synaptic layer, η denotes the learning rate, and M means the branch number of the dendritic layer. It is necessary to determine an optimal set of parameters to obtain high accuracy rate and fast convergence speed. Thus, we employ the Taguchi method to produce the orthogonal arrays [59], which can reduce the number of trails to control the cost of time, manpower, and materials effectively. Each parameter is defined to own four levels in PNN. We provide $L_{16}(4^3)$ orthogonal arrays for both benchmark datasets, which are illustrated in Tables 4 and 5. The corresponding accuracies of PNN with each parameter set are also shown in these tables. We can find that, for Australian credit dataset, the parameter set of the 3rd row ($k = 2, \eta = 0.08, M = 30$) has better performance than the other sets. And the highest testing accuracy of Japanese credit dataset occurs on the 8th row ($k = 2.5, \eta = 0.07, M = 30$). These parameter combinations are reasonable to obtain acceptable performance; to some extent they reveal the effects of the parameters on the performance of PNN. These parameter sets are reasonable to obtain acceptable performance for two benchmark datasets, and we use these parameter sets as

Table 4: $L_{16}(4^3)$ orthogonal array and factor assignment of the Australian credit dataset.

Expe. number/parameter	k	η	M	Testing accuracy
1	2	0.01	25	85.26 ± 1.13
2	2	0.05	28	85.54 ± 1.27
3	2	0.08	30	85.64 ± 1.74
4	2	0.1	32	84.88 ± 1.59
5	3	0.01	25	84.22 ± 5.68
6	3	0.05	28	85.45 ± 1.56
7	3	0.08	32	84.98 ± 1.73
8	3	0.1	30	85.34 ± 1.66
9	4	0.01	30	82.48 ± 5.55
10	4	0.05	32	82.89 ± 5.26
11	4	0.08	25	83.73 ± 3.35
12	4	0.1	28	83.22 ± 4.27
13	5	0.01	32	74.16 ± 8.85
14	5	0.05	30	76.03 ± 9.87
15	5	0.08	28	73.17 ± 12.33
16	5	0.1	25	71.03 ± 12.68

Table 5: $L_{16}(4^3)$ orthogonal array and factor assignment of the Japanese credit dataset.

Expe. number/parameter	k	η	M	Testing accuracy
1	2	0.01	25	84.37 ± 2.16
2	2	0.03	28	85.39 ± 1.36
3	2	0.05	30	84.85 ± 1.61
4	2	0.07	32	85.39 ± 1.08
5	2.5	0.01	28	85.44 ± 1.40
6	2.5	0.03	25	85.29 ± 1.13
7	2.5	0.05	32	85.25 ± 1.52
8	2.5	0.07	30	85.54 ± 1.42
9	3	0.01	30	81.90 ± 8.84
10	3	0.03	32	85.01 ± 1.52
11	3	0.05	25	85.08 ± 1.51
12	3	0.07	28	84.90 ± 1.37
13	3.5	0.01	32	82.12 ± 7.96
14	3.5	0.03	30	84.79 ± 1.05
15	3.5	0.05	28	83.42 ± 5.14
16	3.5	0.07	25	83.94 ± 2.42

the optimal ones to make further comparison with the other classifiers in our experiments.

In our experiments, PNN is compared with the classical multilayer perceptron (MLP) to solve both benchmark problems. PNN and MLP have different neuronal structures, but they utilize the same learning algorithm. For a relatively fair comparison, the number of weights and thresholds of both models should be approximately equal, which can be calculated as follows:

$$N_{\mathrm{MLP}} = I \times L + 2L + 1, \tag{15}$$

where N_{MLP} denotes the amount of the relevant weights and thresholds which need to be adjusted in the structure of MLP. I represents the number of neurons in the input layer. And L means the neuron numbers in the hidden layers.

$$N_{\mathrm{PNN}} = 2I \times J, \tag{16}$$

where N_{PNN} refers to the number of the relevant weights and thresholds which need to be adjusted in the structure of PNN. I represents the number of synapses on each branch of dendrites. And J means the numbers of the dendrite branches. The numbers of the adjusted parameters of both benchmark datasets in our simulation are summarized in Table 6.

5.5. Performance Comparison. To evaluate the performances of different classification methods, each dataset is randomly

Table 6: Structures of PNN and MLP for the Australian and Japanese credit datasets.

Dataset	Model	Number of input	Number of branch hidden node	Number of output	Number of adjusted parameters
Australian	PNN	14	30	1	840
	MLP	14	53	1	849
Japanese	PNN	15	30	1	900
	MLP	15	53	1	902

Table 7: Comparison of the simulation results between PNN and MLP of the Australian credit dataset.

Method	Accuracy (%)	p value	Sensitivity	Specificity	AUC
PNN	85.64 ± 1.74	N/A	0.9484	0.9111	0.9411
MLP	84.23 ± 1.73	0.0038	0.9063	0.7789	0.8976

separated into two subsets, one is for training and the other is for testing. The training subset is used to train the classification model, and the testing one is adopted to verify the validity of the model. And the percentages of the training and the testing subset are set to be 50% and 50% [60], respectively. All the experiments of the two benchmark datasets run 30 times, the average (mean) and standard deviation (Std) of the results are provided in the form of Mean ± Std.

In the classification investigation field, cross-validation is widely applied to test the model's robustness, especially under the uncertainty with unknown class labels [61]. In contrast with the single-fold validation method, the multifold cross-validation (CV) such as K-fold CV has the advantage to minimize the bias caused by random sampling, whereas it has the disadvantages of excessive computation time and cost requirement [62]. In our experiments, 5-fold CV and 10-fold CV methods are applied to compare PNN with the other classifiers.

In addition, a nonparameter statistical test, namely, Wilcoxon rank-sum test was adopted to detect the significant difference between PNN and MLP in our experiments. The null hypothesis means that there is no difference between two models, and the required significance level is set to be 0.05. If the p value is less than 0.05, there is a strong evidence to reject the null hypothesis. And if it is larger than 0.05, the null hypothesis cannot be rejected. N/A represents "Not Applicable" which indicates that the relevant algorithm does not need to be compared with itself.

5.5.1. Australian Credit Dataset. In this section, PNN is firstly compared with MLP to solve Australian credit problem. The learning rate is set to be 0.08 for both models. As shown in Table 7, the proposed PNN acquires an average testing accuracy of 85.64%, which is higher than the 84.23% accuracy rate obtained by MLP. The p value of Wilcoxon rank-sum test is 0.0038, which is smaller than the required significance level (0.05). It implies that there is a significant difference between PNN and MLP to solve Australian credit problem. Moreover, PNN also performs better than MLP in the aspects of sensitivity and specificity. Convergence speed is also a performance metric which affects the efficiency of a model. The convergence curves of PNN and MLP for Australian credit dataset are compared in Figure 7. It can be observed that, at the beginning, the convergence speed of MLP is higher than that of PNN, while PNN converges more quickly since the 50th iteration of the training process.

Based on the sensitivity and specificity values of PNN and MLP in Table 7, we can conclude that a higher sensitivity value indicates that PNN is more powerful to identify the applicants who are creditworthy. And a higher specificity value represents that PNN has a smaller probability to misjudge a creditworthy applicant when solving Australian credit problem. Figure 6 shows the ROC curves of PNN and MLP. By calculating the area under the curves, the AUC value of PNN (0.9411) is found to be larger than that of MLP (0.8976). Besides MLP, we compare PNN with some other classifiers, such as support vector machine (SVM), Kth nearest neighbor (KNN), and Bayesian network. The corresponding results have been illustrated in Table 8. We can find that all the three cases of PNN, namely, 50%-50%, 5 × CV, and 10 × CV, have performed higher classification accuracy rates than the other classifiers. It has once again proved that PNN is capable of providing superior performances to solve Australian credit problem.

5.5.2. Japanese Credit Dataset. When dealing with the Japanese credit dataset, learning rate of both PNN and MLP is set to be 0.07. As shown in Table 9, the average testing accuracy rate of 30 times experiments of PNN is 85.54%, which is higher than that of MLP. p value of Wilcoxon test is $6.4811e^{-05}$, and it is smaller than the required significance level (0.05). Thus, we can conclude that the accuracy of PNN is significantly higher than that of MLP. What is more, PNN obtains higher values of sensitivity and specificity than MLP, which implies that PNN is more powerful to retain creditworthy applicants and remove uncreditworthy applicants, when dealing with Japanese credit problem. It also can be observed from the ROC curves, which are illustrated in Figure 8. By calculating the area under ROC, we can find that the AUC of PNN is 0.9301, which is larger than that of MLP. The convergence curves of PNN and MLP are provided in Figure 9. As it is observed, PNN converges very quickly and nearly achieves the best convergence performance at the 20th iteration. At the end of the training process, PNN presents lower training error than MLP.

Table 8: Classification accuracy rates comparison between PNN and other algorithms obtained from literatures of the Australian credit dataset.

Authors (published year)	Algorithms (train-to-test ratios)	Classification accuracy rate (%)
Luo et al. (2009) [48]	SVM (10 × CV)	80.43
Peng et al. (2011) [49]	Bayesian network (10 × CV)	85.22
	KNN (10 × CV)	79.42
	RBF network (10 × CV)	83.04
Yu et al. (2011) [1]	C4.5 (10 × CV)	84.3
	LVQ (10 × CV)	82.97
Chang and Yeh (2012) [50]	SVM (10 × CV)	84.7
	C4.5 (10 × CV)	82.5
	Naive Bayes (10 × CV)	84.9
Zhu et al. (2013) [51]	QDA (5 × CV)	80.02
	DT (5 × CV)	83.18
Tsai et al. (2014) [52]	MLP (10 × CV)	82.44
	DT (10 × CV)	84.91
Lessmann et al. (2015) [8]	CART (10 × CV)	66.4
	ELM (10 × CV)	69.8
	LDA (10 × CV)	78.9
	Logistic regression (10 × CV)	80.7
	ADT (10 × CV)	79.8
	Bag (10 × CV)	76.8
	Boost (10 × CV)	81
	Random forest (10 × CV)	85.2
Khashei and Mirahmadi (2015) [7]	QDA (50%-50%)	80.1
	SVM (50%-50%)	77.5
Our method (2017)	PNN (50%-50%)	85.64
	PNN (5 × CV)	85.31
	PNN (10 × CV)	85.19

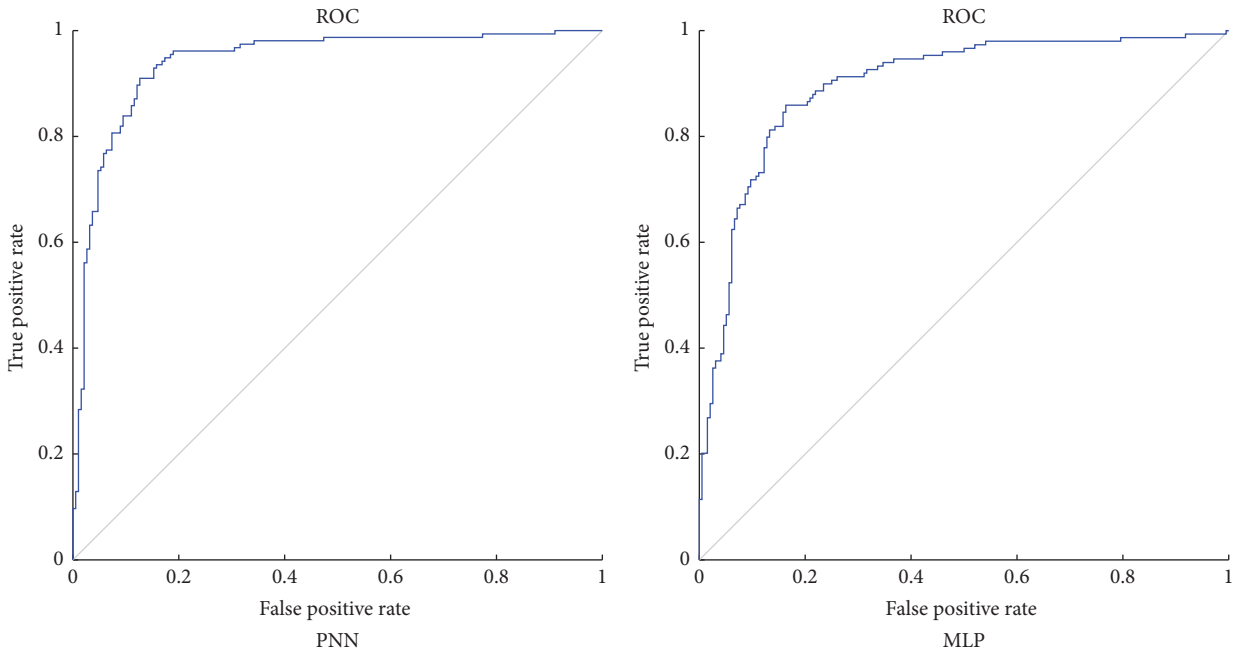

Figure 6: The ROC of PNN and MLP for Australian credit dataset.

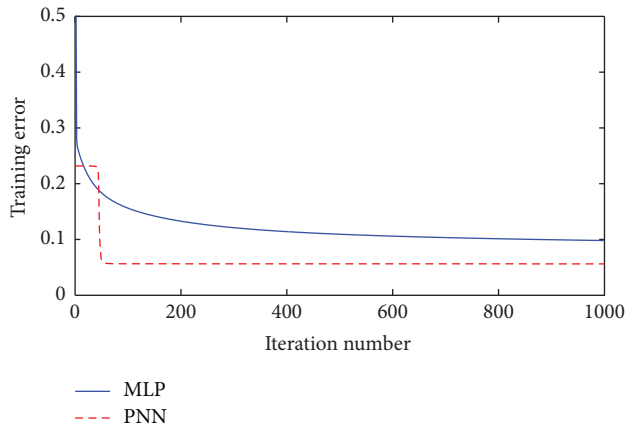

Figure 7: Comparison of convergence speed between PNN and MLP of Australian credit dataset.

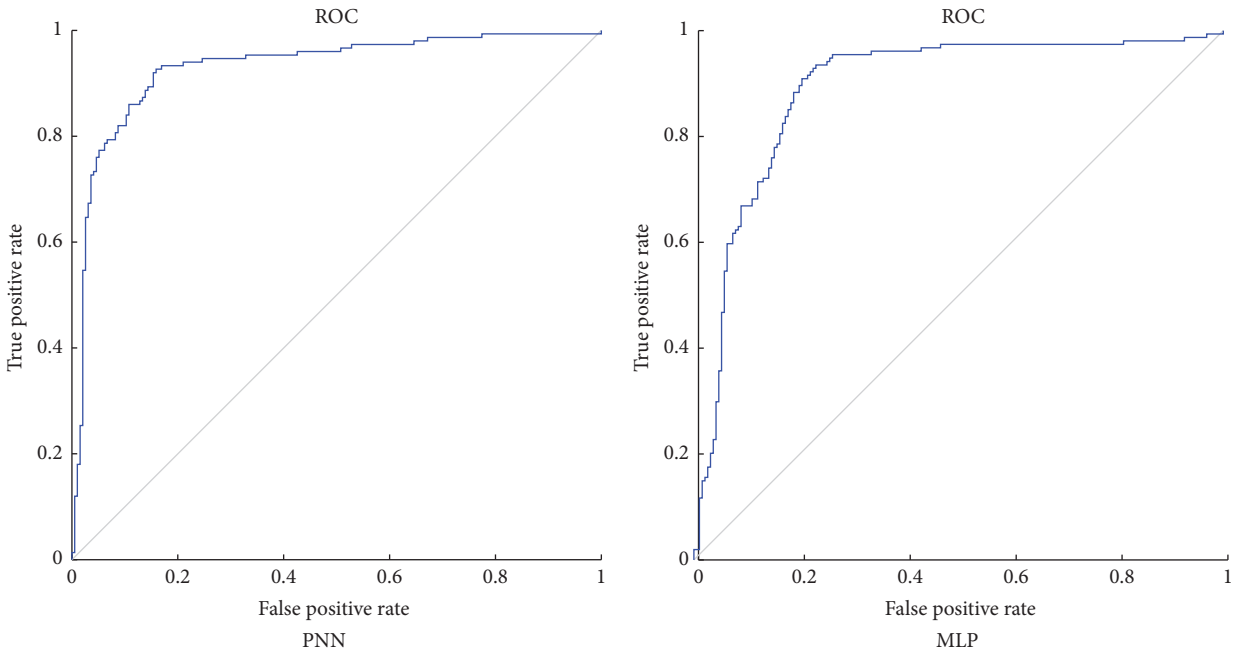

Figure 8: The ROC of PNN and MLP for Japanese credit dataset.

In addition, we compare the classification performance of PNN with some other classifiers, and the comparison has been summarized in Table 10. The accuracies of three cases of our method (50%-50%, 5 × CV, and 10 × CV) are 85.54%, 85.23%, and 85.27%, respectively. All of them are obviously higher than the other classifiers. Based on these results, it can be concluded that PNN possesses a relatively high convergence speed and accuracy to solve Japanese credit problem.

5.6. Dendrite Morphology Reconstruction

5.6.1. The Ultimate Synaptic and Dendritic Morphology. As mentioned above, PNN utilizes synaptic pruning and dendritic pruning to realize structural plasticity, and the superfluous synapses and useless dendrites can be removed during the process of learning. Hence, a simplified and distinct structural morphology is formed and it can be replaced by a logic circuit. In this section, we verify the effectiveness of the neuronal pruning function and the accuracy of the logical circuit by applying Australian and Japanese credit datasets.

Figure 10 shows the dendritic structure of Australian credit dataset before learning. As it shows, there are 30 branches of dendrites in the structure, and each branch has 14 synapses which connect to 14 input features. All the connection cases of these synapses are determined by the randomly chosen weights and thresholds. Figure 11 presents the relative structure after learning. We use the symbol "×" to represent that this dendrite can be removed by dendritic pruning. Figure 12 shows that all the unnecessary branches of dendrites which own the synapses in constant-0 connection case are detected; only dendrite 7 and dendrite 12 are retained. After removing all the synapses in the constant-1 connection cases,

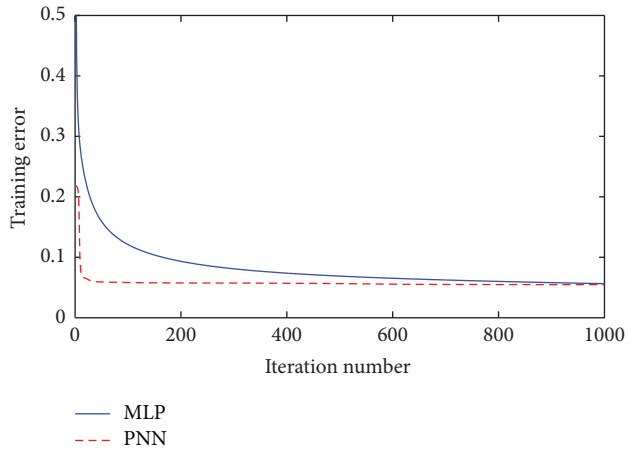

Figure 9: Comparison of convergence speed between PNN and MLP of Japanese credit dataset.

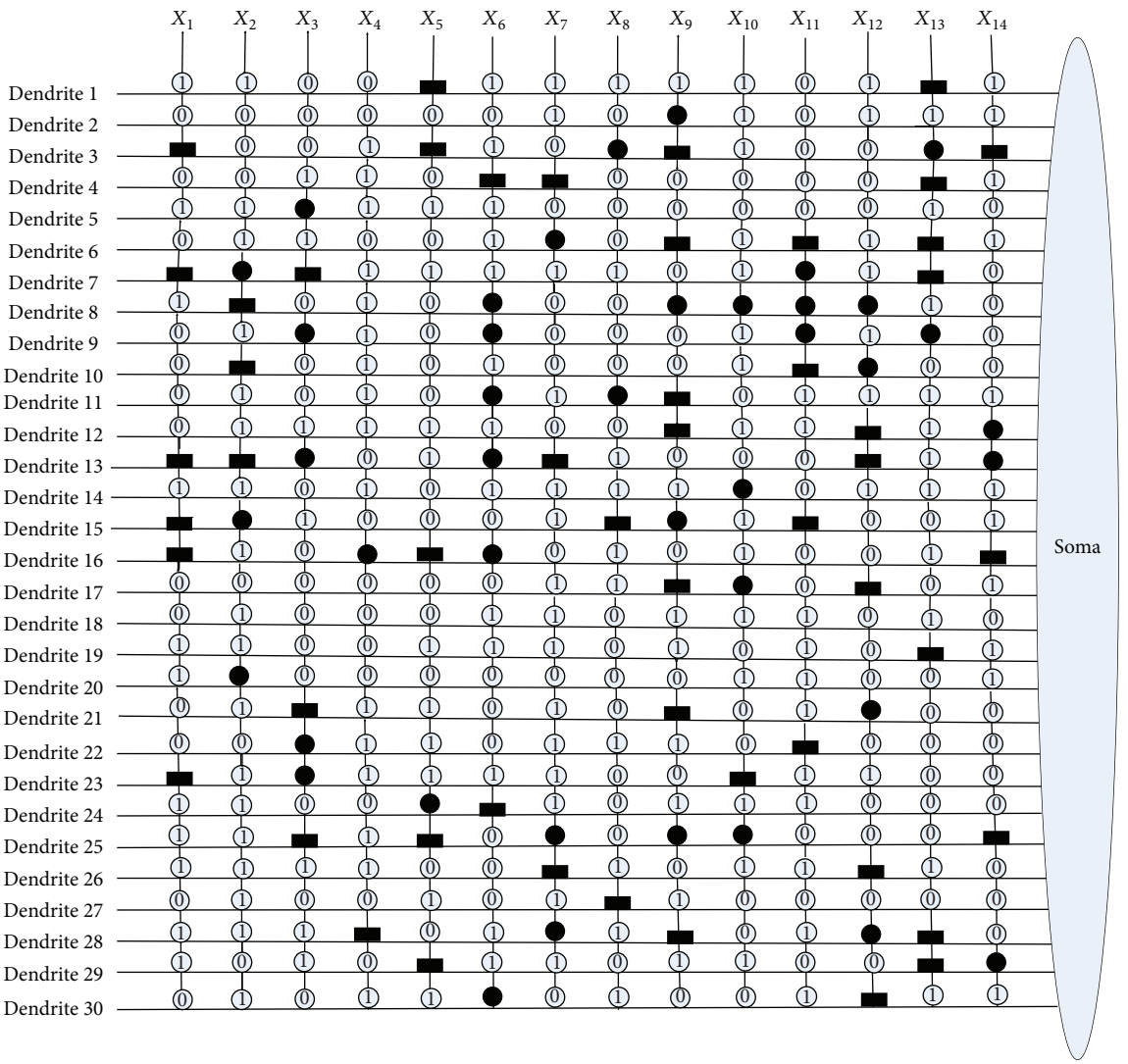

Figure 10: The dendritic morphology of the Australian credit dataset before learning.

Table 9: Comparison of the simulation results between PNN and MLP of the Japanese credit dataset.

Method	Accuracy (%)	p value	Sensitivity	Specificity	AUC
PNN	85.54 ± 1.42	N/A	0.9333	0.8316	0.9301
MLP	83.92 ± 1.32	$6.4811e^{-05}$	0.8247	0.8148	0.8900

Table 10: Classification accuracy rates comparison between PNN and other algorithms obtained from the literatures of the Japanese credit dataset.

Authors (published year)	Algorithms (train-to-test ratios)	Classification accuracy rate (%)
Yu et al. (2008) [53]	log R (10 × CV)	75.82
	ANN (10 × CV)	80.77
	SVM (10 × CV)	79.91
	Neuro-fuzzy hybrid (10 × CV)	77.91
	Fuzzy SVM hybrid (10 × CV)	83.94
Tsai et al. (2014) [52]	MLP (10 × CV)	84.38
Our method (2017)	PNN (50%-50%)	85.54
	PNN (5 × CV)	85.23
	PNN (10 × CV)	85.27

Table 11: Model structure comparison between Australian and Japanese credit datasets.

Dataset	Feature input		Dendritic layer		Adjusted weight	
	Initial value	Selected value	Initial value	Selected value	Initial value	Selected value
Australian credit dataset	14	4	30	2	840	16
Japanese credit dataset	15	4	30	2	900	16

the final synaptic and dendritic morphology is described in Figure 13, and it can be observed that we delete all the unnecessary synapses which connect to X_1, X_2, X_3, X_5, X_6, X_7, X_9, X_{10}, X_{11}, and X_{14}. The final reserved features are only X_4, X_8, X_{12}, and X_{13} for Australian credit dataset.

Then, the same process of disposing Japanese credit dataset has been illustrated in Figures 14, 15, 16, and 17. It can be observed that the neuronal pruning function has totally abandoned 28 unnecessary branches of dendrites and 11 redundant features. Only features X_4, X_7, X_9, and X_{13} are reserved in the final structure of PNN. The model structure comparison between Australian and Japanese credit datasets is summarized in Table 11.

5.6.2. The Simplified Logic Circuit (LC) of the After-Learning Morphology. After implementing the neuronal pruning function, we have obtained two simplified model structures for both benchmark problems. Then, these models are replaced by logic circuits (LCs) which consist of an analog-to-digital converter, namely, "comparator," logical "NOT," "AND," and "OR" gates. LCs of both benchmark datasets are presented in Figures 18 and 19, respectively. A comparator is used to compare the practical input with the threshold θ. Once the input x_i is less than the threshold θ, the "comparator" will output 0. On the contrary, if the input exceeds the threshold θ, the output will be 1. By these LCs, we can classify the applicants into accepted and rejected for the Australian credit dataset and Japanese credit dataset. Moreover, we calculate the accuracy of these LCs and provide the results in Table 12. It is obvious that the test accuracy of the Australian credit dataset is 85.80%, and the test accuracy of the Japanese credit dataset is 85.51%. They are nearly equal to the accuracies of PNN before simplification which are 85.64% and 85.54%.

Moreover, we compare the results of another pruning method (named as "correlation pruning (CP)") to simplify the PNN structure. Specifically, the method detects the pair of the most highly correlated branches of dendrites in the morphology structure. Each dendritic branch is represented by the vector B_i which consists of its synaptic parameters w_{ij} and q_{ij} in the ith dendritic branch. Then, one of the dendritic branches in the pair will be deleted randomly. The process repeats until the branch number of PNN satisfies a predetermined number pn. In our experiments, the values of pn of the Australian and Japanese credit datasets are set to be 15, and both experiments are run 30 times independently. The correlation coefficient r is defined as follows:

$$r\left(B_i, B_j\right) = \frac{\operatorname{cov}\left(B_i, B_j\right)}{\sqrt{\operatorname{var}\left[B_i\right] * \operatorname{var}\left[B_j\right]}}, \tag{17}$$

where $\operatorname{cov}(B_i, B_j)$ represents the covariance of B_i and B_j and $\operatorname{var}[B_i]$ and $\operatorname{var}[B_j]$ denote the variance of B_i and B_j, respectively. Figures 20 and 21 illustrate the simplified structures of PNN, and CP discards 15 highly correlated branches for both datasets. The average of the test accuracy rate of these simplified PNN structures is also presented in Table 12. It can be observed that the test accuracy rates of the Australian and Japanese credit datasets are 61.21% and 65.56%, respectively. They are smaller than the results of PNN and the simplified LCs. It means that the replacement of LCs is more effective,

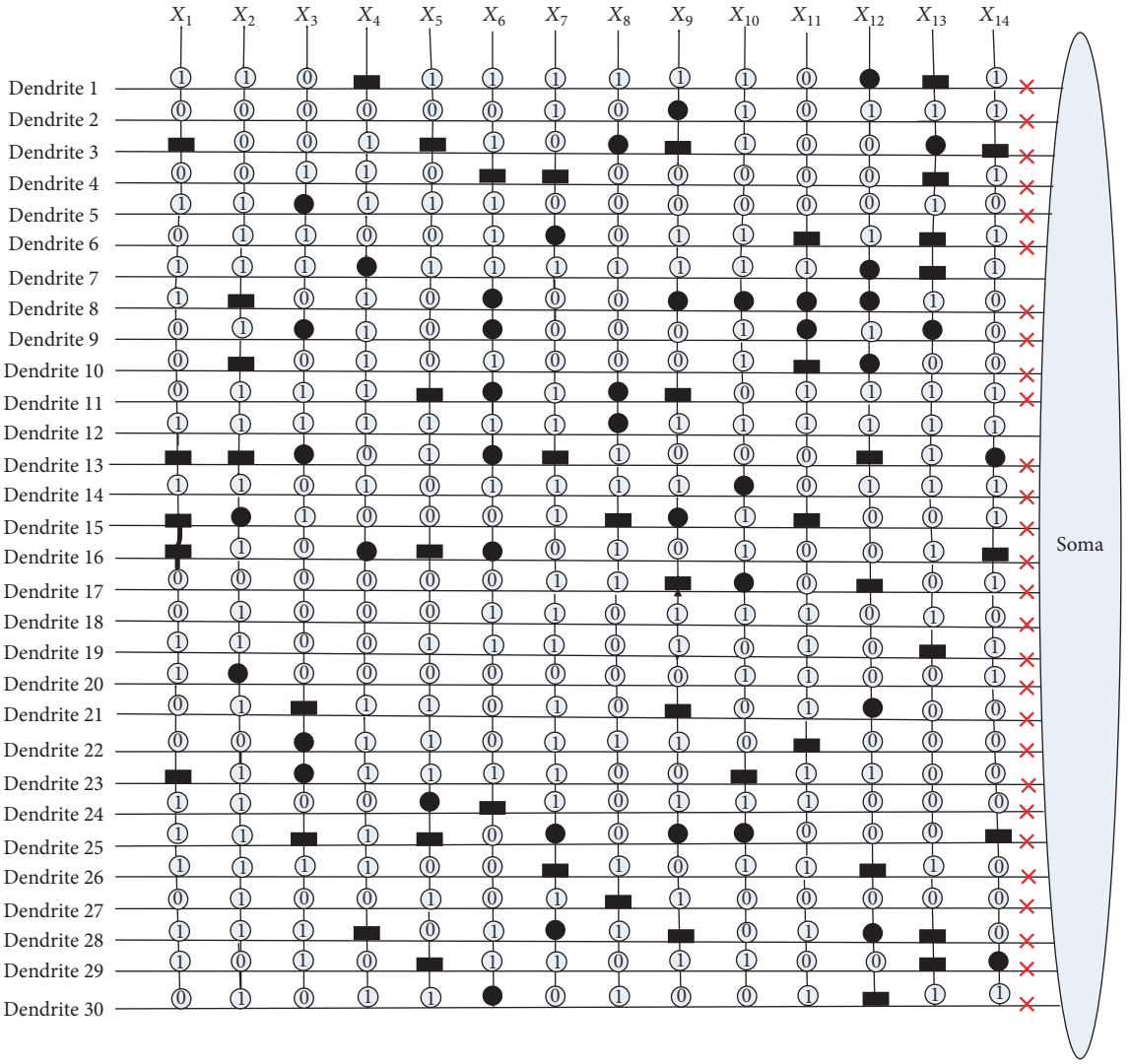

Figure 11: The dendritic morphology of the Australian credit dataset after learning.

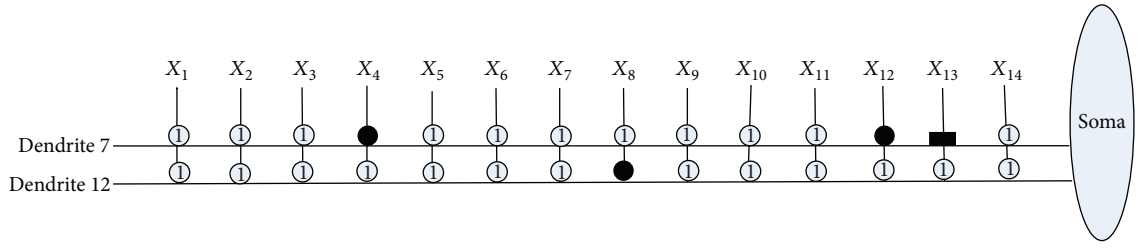

Figure 12: The dendritic morphology of the Australian credit dataset after dendritic pruning.

Table 12: Verification between LC and CP.

Credit dataset	Test accuracy of PNN (%)	Test accuracy of LC (%)	Test accuracy of CP (%)
Australian	85.64	85.80	61.21
Japanese	85.54	85.51	65.56

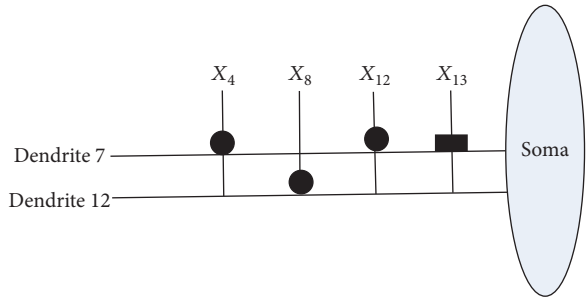

Figure 13: The dendritic morphology of the Australian credit dataset after synaptic pruning.

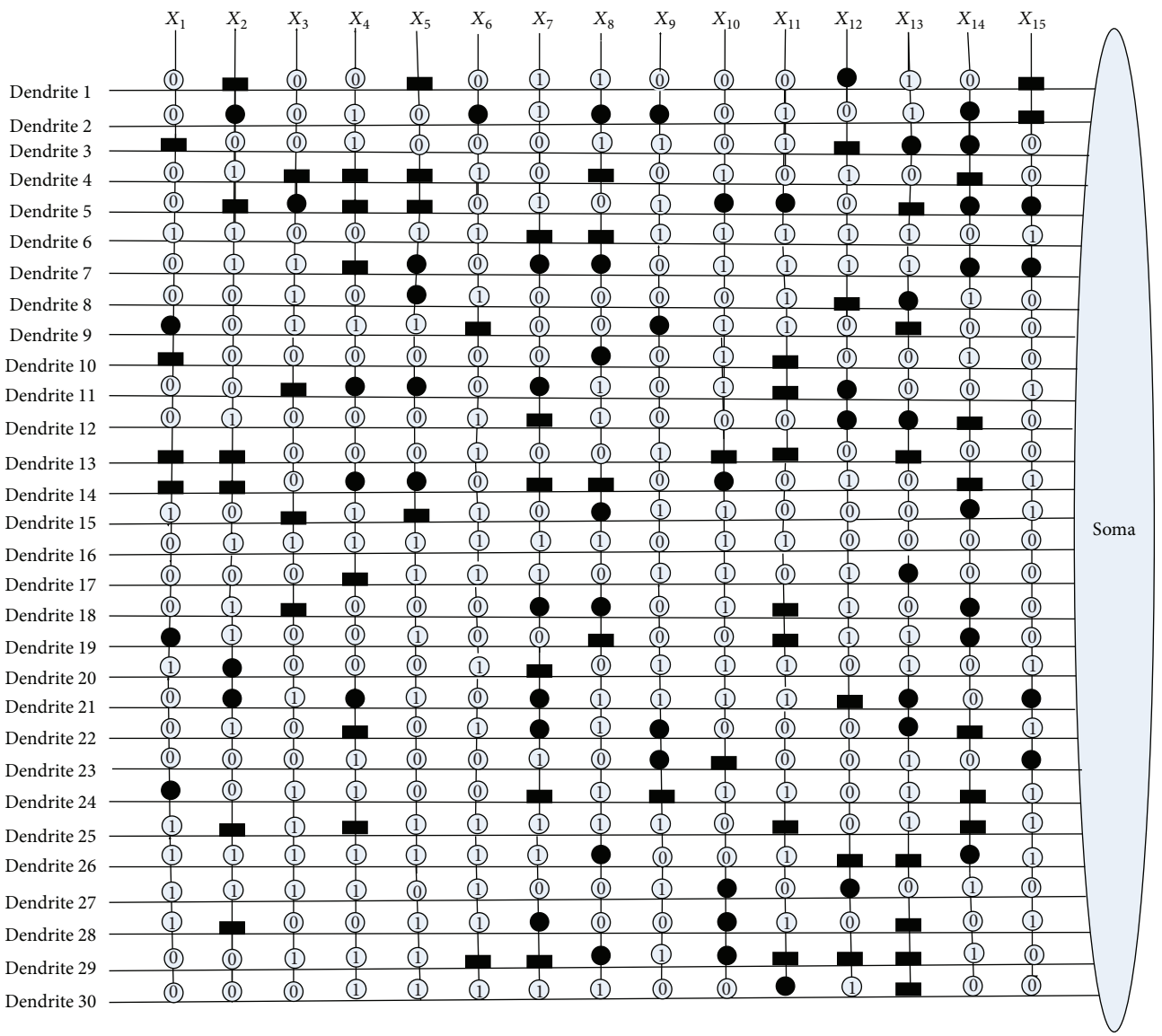

Figure 14: The dendritic morphology of the Japanese credit dataset before learning.

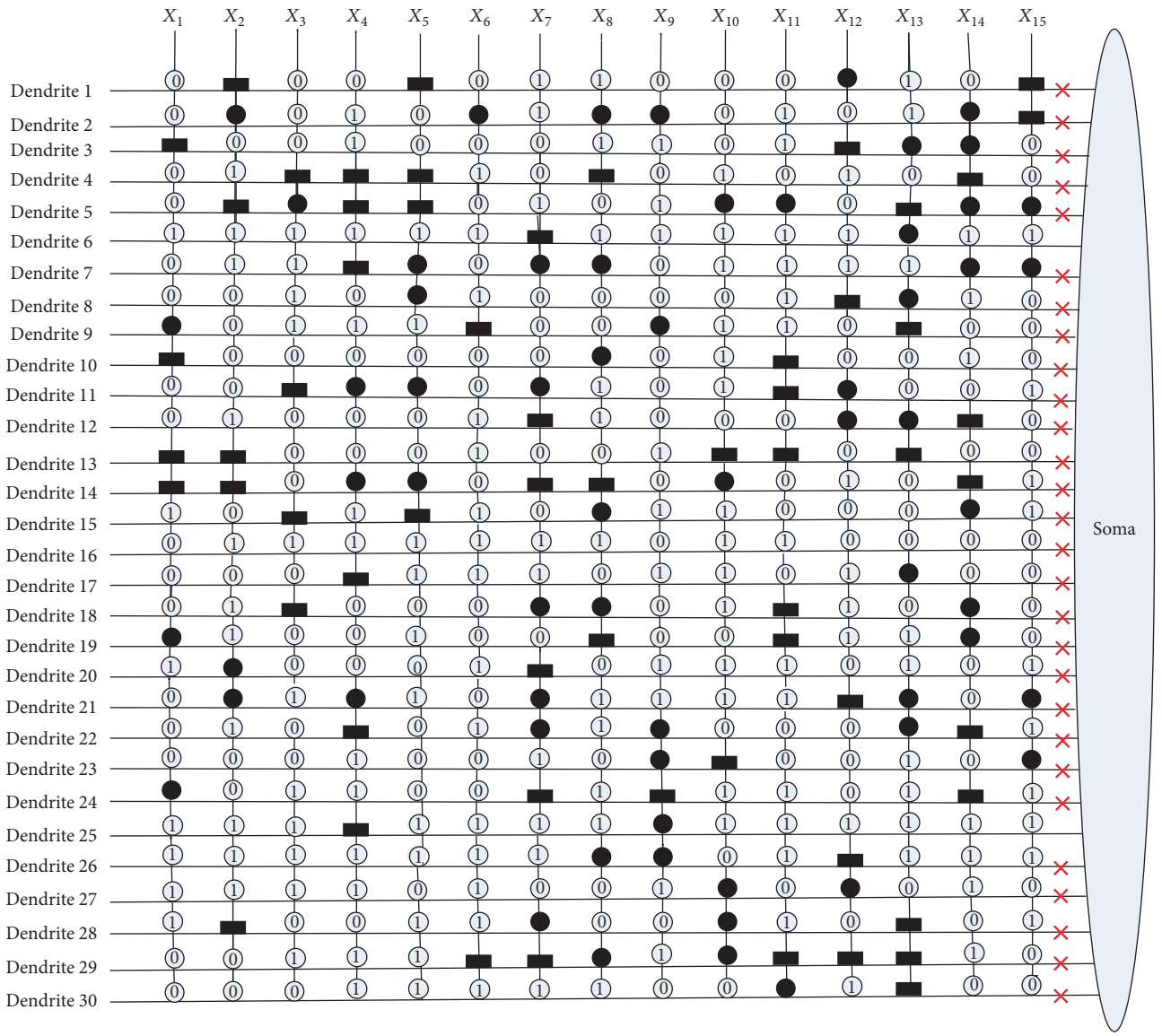

Figure 15: The dendritic morphology of the Japanese credit dataset after learning.

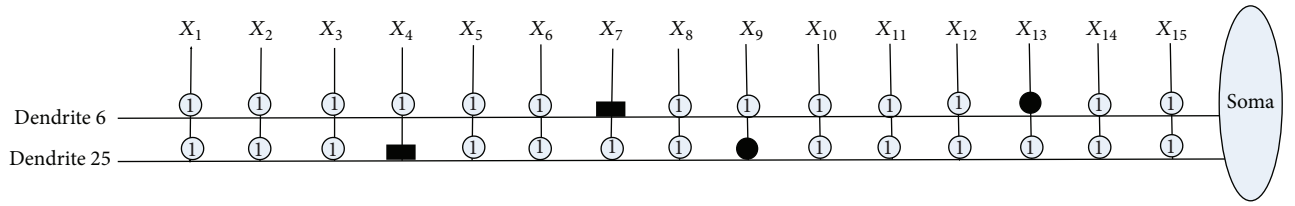

Figure 16: The dendritic morphology of the Japanese credit dataset after dendritic pruning.

and it does not sacrifice the classification accuracy rate of PNN. In addition, LCs can be easily implemented in hardware, which can achieve a high computational speed. Based on these excellent characteristics, the obtained LCs are thought to be powerful classifiers for the benchmark datasets.

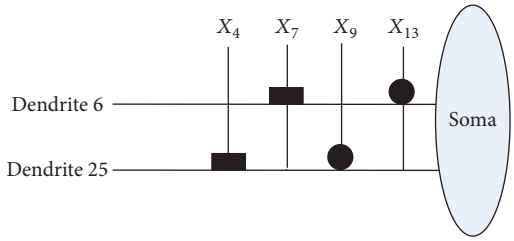

Figure 17: The dendritic morphology of the Japanese credit dataset after synaptic pruning.

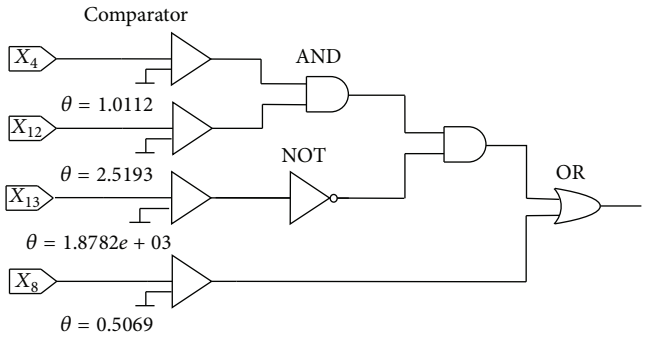

Figure 18: LC of the Australian credit dataset.

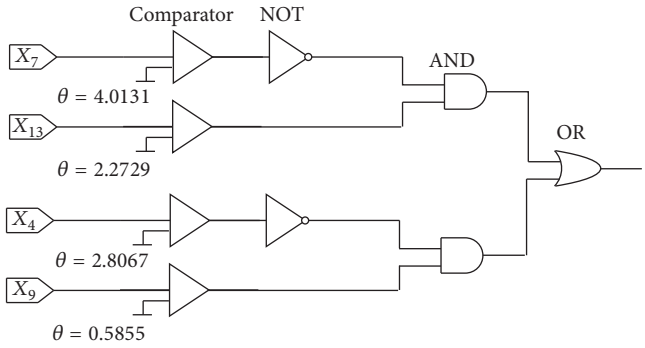

Figure 19: LC of the Japanese credit dataset.

6. Discussion

Based on the above experimental results, the following is clear. Firstly, PNN has higher accuracy rate than MLP on both benchmark datasets. It means PNN can offer much more correct decision support for the financial institutions. Secondly, higher values of sensitivity and specificity imply that PNN is able to retain better applicants and remove worse applicants with a high probability, respectively. Thirdly, PNN has a larger value of AUC which represents that the differences between creditworthy and uncreditworthy groups classified by PNN are more obvious. This point is very important in the credit risk assessment because it is helpful to make the financial institutions and credit applicants accept the classification result more easily. Lastly, PNN provides two LCs for both benchmark datasets. These LCs have satisfactory classification performances and they will extremely speed up the classification to offer correct and quick decision for the decision makers. Although many novel algorithms are constantly emerging to solve the credit classification problems, many approaches have still merely focused on the credit classification models' ability of improving the classification accuracy rate, while PNN provides a brand new perspective to improve the efficiency of ANNs.

Feature selection is one of the most important steps of machine learning in the process of data mining. It focuses on filtering out the redundant and irrelevant features from the original large-scale datasets, which can reduce the running time of a learning algorithm and improve the model's performance consequently as well as reducing the effort of training the model [63]. Many algorithms have been proposed to select effective features from the input attributes such as t-test, stepwise, and related matrix [54]. It is worth mentioning that PNN can also implement feature selection during the training process because the pruning function can reduce not only the superfluous branches of dendrites but also the unnecessary synapses. And each synapse connects to the input of a feature. If all these kinds of synapses are eliminated, the features will be extracted. The extraction rates of the two benchmark datasets are shown in Tables 13 and 14, respectively. It can be observed that, for the Australian credit dataset, although the accuracy rate of PNN is not the best one, the feature extraction rate of PNN is obviously higher than the other five methods. As for Japanese dataset, PNN has the highest accuracy rate and extraction rate simultaneously. Therefore, we can conclude that PNN owns the best extraction rate among the six feature selection methods. It is notable that the features selected by PNN are only verified to be effective in our neural model. In our future research, we intend to investigate whether these selected features can remain suitable in other classification algorithms.

It is inevitable that although PNN performs satisfactorily on several aspects, it also has its disadvantages such as its results lack of interpretability especially on analyzing the pruning reasons for the input variables. Interpretability represents whether the classification results can be explained clearly to the applicants [51]. This will be a major drawback and cause a reluctance to use the approach. Even it may go further that when credit application has been refused to a client, the financial institutions should provide definite reasons legally why the application is rejected. Previous literature reviews show that some algorithms perform well in one or two aspects at most but bad in the remaining aspects. In a word, there is nearly no algorithm which can balance accuracy, complexity, and interpretability; PNN is no exception. In order to acquire useful and understandable knowledge, adopting a visual and interactive framework will be an inevitable trend to integrate the users into the black-box process.

7. Conclusion

In this paper, a pruning neural network classifier PNN is proposed for credit classification. We can conclude that,

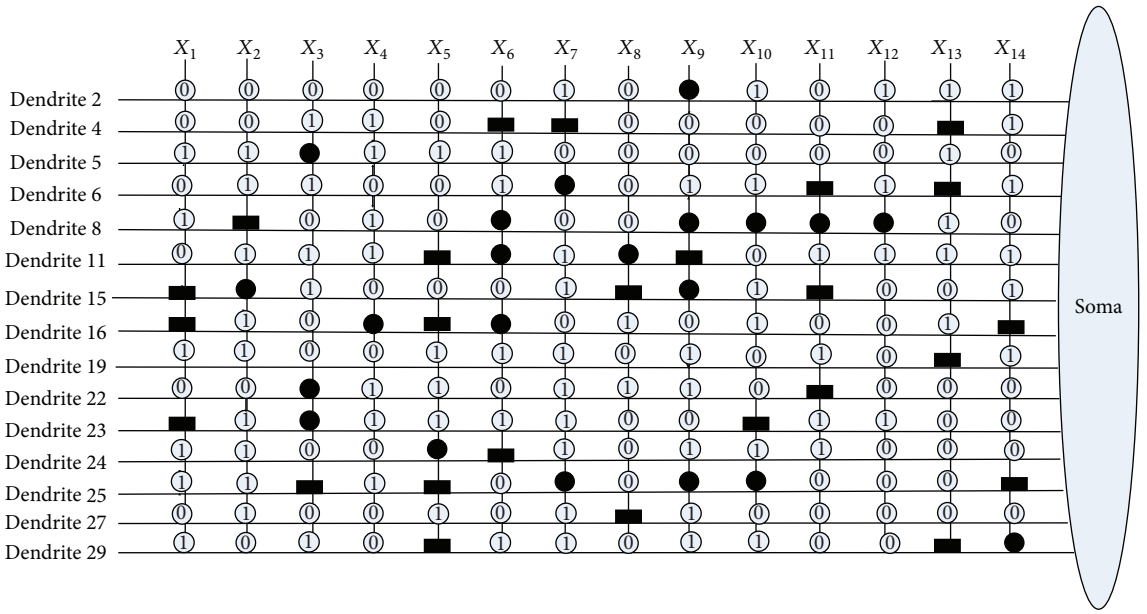

Figure 20: The simplified structure of the Australian credit dataset acquired by CP.

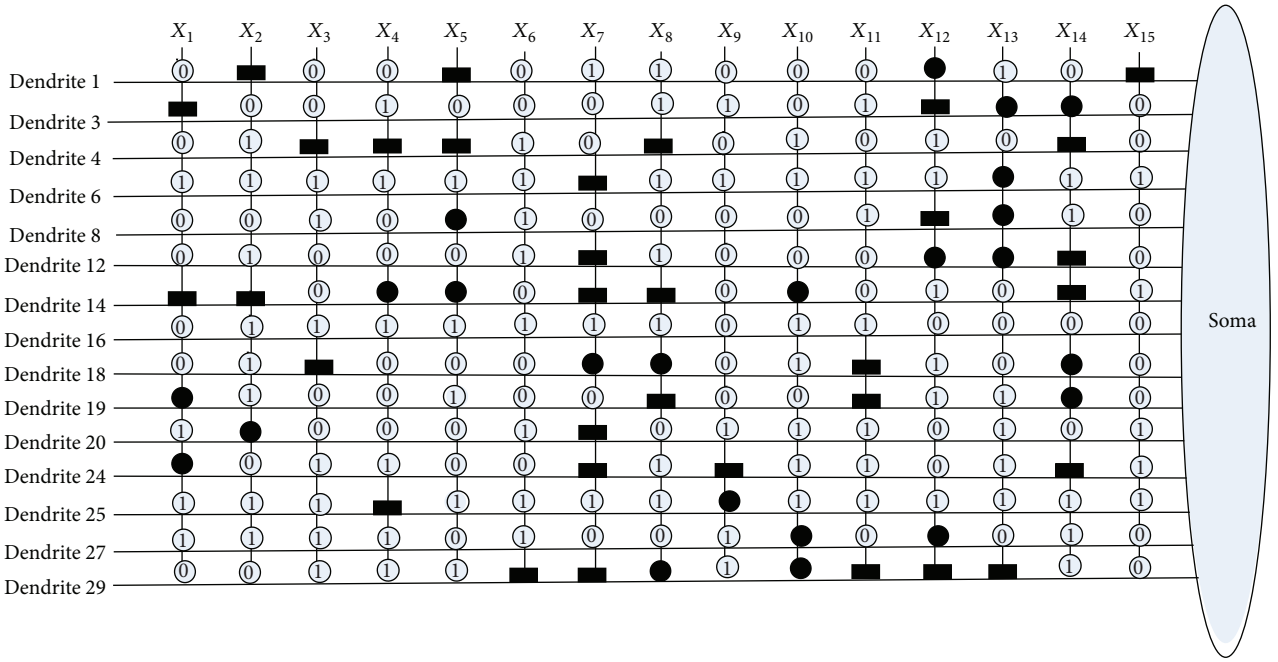

Figure 21: The simplified structure of the Japanese credit dataset acquired by CP.

in contrast with MLP and other classifiers, PNN performs the best in terms of the average accuracy rate, sensitivity, specificity, and AUC for the two popularly applied benchmark datasets, namely, Australian credit dataset and Japanese credit dataset. Besides, PNN has provided tidy neuronal morphologies and LCs by synaptic and dendritic pruning for both datasets. And the efficiency of LCs has been verified in our experiments. Therefore, PNN will be a very effective and efficient method to solve the classification problems.

In summary, the contributions of this paper are as follows. (1) The PNN model that we have proposed gets further access to the realistic biological neural model in comparison with other ANNs. (2) PNN can simplify its structure during the training process by its synaptic and dendritic pruning mechanisms. (3) PNN settles the credit classification issue efficiently in terms of accuracy and convergence speed. (4) PNN offers another perspective to realize feature selection. (5) LCs of the two classification benchmark problems obtain satisfactory accuracy and higher computation speed. These points imply that the proposed PNN owns great potential to be applied in solving other real world classification problems in the big data era.

Table 13: Extraction rate and accuracy rate comparison between PNN and other feature selection methods of the Australian credit dataset.

Methods	Numbers of variables before feature selection	Numbers of variables after feature selection	Extraction rate (%)	Accuracy rate (%)
T-test [54]	14	12	14.3	89.27
Stepwise [54]	14	7	50	84.74
Related matrix [54]	14	12	14.3	89.31
Factor analysis [54]	14	9	35.7	86.08
PCA [54]	14	9	35.7	89.93
PNN	14	4	71.4	85.80

Table 14: Extraction rate and accuracy rate comparison between PNN and other feature selection methods of the Japanese credit dataset.

Methods	Numbers of variables before feature selection	Numbers of variables after feature selection	Extraction rate (%)	Accuracy rate (%)
T-test [54]	15	12	20	65.53
Stepwise [54]	15	5	66.7	82.64
Related matrix [54]	15	12	20	60.16
Factor analysis [54]	15	11	26.7	74.22
PCA [54]	15	8	46.7	74.00
PNN	15	4	73.3	85.51

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was partially supported by the JSPS KAKENHI Grants nos. JP17K12751 and JP15K00332, Hunan Provincial Status and Decision-Making Advisory Research (no. 13C1165), Hunan Educational Bureau Research Item (no. 2014BZZ270), and the Prospective Joint Research of University-Industry Cooperation of Jiangsu (no. BY2016056-02).

References

[1] L. Yu, X. Yao, S. Wang, and K. K. Lai, "Credit risk evaluation using a weighted least squares SVM classifier with design of experiment for parameter selection," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15392–15399, 2011.

[2] T.-S. Lee, C.-C. Chiu, C.-J. Lu, and I.-F. Chen, "Credit scoring using the hybrid neural discriminant technique," *Expert Systems with Applications*, vol. 23, no. 3, pp. 245–254, 2002.

[3] D. West, "Neural network credit scoring models," *Computers & Operations Research*, vol. 27, no. 11-12, pp. 1131–1152, 2000.

[4] L. C. Thomas, D. B. Edelman, and J. N. Crook, *Credit scoring and its applications*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, USA, 2002.

[5] W.-Y. Lin, Y.-H. Hu, and C.-F. Tsai, "Machine learning in financial crisis prediction: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 4, pp. 421–436, 2012.

[6] H. A. Abdou and J. Pointon, "Credit scoring, statistical techniques and evaluation criteria: a review of the literature," *Intelligent Systems in Accounting, Finance and Management*, vol. 18, no. 2-3, pp. 59–88, 2011.

[7] M. Khashei and A. Mirahmadi, "A soft intelligent risk evaluation model for credit scoring classification," *International Journal of Financial Studies*, vol. 3, no. 3, pp. 411–422, 2015.

[8] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," *European Journal of Operational Research*, vol. 247, no. 1, pp. 124–136, 2015.

[9] D. J. Hand and W. E. Henley, "Statistical classification methods in consumer credit scoring: A review," *Journal of the Royal Statistical Society. Series A: Statistics in Society*, vol. 160, no. 3, pp. 523–541, 1997.

[10] B. Baesens, R. Setiono, C. Mues, and J. Vanthienen, "Using neural network rule extraction and decision tables for credit-risk evaluation," *Management Science*, vol. 49, no. 3, pp. 312–329, 2003.

[11] E. I. Altman, G. Marco, and F. Varetto, "Corporate distress diagnosis: comparisons using linear discriminant analysis and neural networks," *Journal of Banking & Finance*, vol. 18, no. 3, pp. 505–529, 1994.

[12] A. F. Atiya, "Bankruptcy prediction for credit risk using neural networks: a survey and new results," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 12, no. 4, pp. 929–935, 2001.

[13] C. Carter and J. Catlett, "Assessing Credit Card Applications Using Machine Learning," *IEEE Expert-Intelligent Systems and their Applications*, vol. 2, no. 3, pp. 71–79, 1987.

[14] K. J. Leonard, "Detecting credit card fraud using expert systems," *Computers & Industrial Engineering*, vol. 25, no. 1-4, pp. 103–106, 1993.

[15] V. S. Desai, D. G. Conway, J. N. Crook, and G. A. Overstreet, "Credit-scoring models in the credit-union environment using neural networks and genetic algorithms," *IMA Journal of Management Mathematics*, vol. 8, no. 4, pp. 323–346, 1997.

[16] F. Varetto, "Genetic algorithms applications in the analysis of insolvency risk," *Journal of Banking & Finance*, vol. 22, no. 10-11, pp. 1421–1439, 1998.

[17] M. D. Odom and R. Sharda, "A neural network model for bankruptcy prediction," in *Proceedings of the International Joint Conference on Neural Prediction Networks (IJCNN '90)*, pp. 163–168, 1990.

[18] G. Klersey and M. Dugan, "Substantial doubt: Using artificial neural networks to evaluate going concern," *Advances in accounting information systems 3*, pp. 137–159, 1995.

[19] T. E. McKee and M. Greenstein, "Predicting bankruptcy using recursive partitioning and a realistically proportioned data set," *Journal of Forecasting*, vol. 19, no. 3, pp. 219–230, 2000.

[20] S. Piramuthu, "Financial credit-risk evaluation with neural and neurofuzzy systems," *European Journal of Operational Research*, vol. 112, no. 2, pp. 310–321, 1999.

[21] H. C. Koh, W. C. Tan, and G. C. Peng, "Credit scoring using data mining techniques," *Singapore Management Review*, vol. 26, no. 2, 2004.

[22] T. Bellotti and J. Crook, "Support vector machines for credit scoring and discovery of significant features," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3302–3308, 2009.

[23] B. S. Trinkle and A. A. Baldwin, "Research opportunities for neural networks: the case for credit," *Intelligent Systems in Accounting, Finance and Management*, vol. 23, no. 3, pp. 240–254, 2016.

[24] M. K. Lim and S. Y. Sohn, "Cluster-based dynamic scoring model," *Expert Systems with Applications*, vol. 32, no. 2, pp. 427–431, 2007.

[25] H. Wang, Q. Xu, and L. Zhou, "Large unbalanced credit scoring using lasso-logistic regression ensemble," *PLoS ONE*, vol. 10, no. 2, Article ID e0117844, 2015.

[26] F. Rosenblatt, *Principles of Neurodynamics*, Spartan Book, 1962.

[27] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biology*, vol. 5, no. 4, pp. 115–133, 1943.

[28] L. F. Abbott and W. G. Regehr, "Synaptic computation," *Nature*, vol. 431, no. 7010, pp. 796–803, 2004.

[29] C. Koch, T. Poggio, and V. Torre, "Nonlinear interactions in a dendritic tree: Localization, timing, and role in information processing," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 80, no. 9 I, pp. 2799–2802, 1983.

[30] S. Blomfield, "Arithmetical operations performed by nerve cells," *Brain Research*, vol. 69, no. 1, pp. 115–124, 1974.

[31] N. Brunel, V. Hakim, and M. J. E. Richardson, "Single neuron dynamics and computation," *Current Opinion in Neurobiology*, vol. 25, pp. 149–155, 2014.

[32] V. Torre and T. Poggio, "A synaptic mechanism possibly underlying directional selectivity to motion," *Proceedings of the Royal Society B Biological Science*, vol. 202, no. 1148, pp. 409–416, 1978.

[33] Y.-N. Jan and L. Y. Jan, "Branching out: Mechanisms of dendritic arborization," *Nature Reviews Neuroscience*, vol. 11, no. 5, pp. 316–328, 2010.

[34] Q. Wen and D. B. Chklovskii, "A cost-benefit analysis of neuronal morphology," *Journal of Neurophysiology*, vol. 99, no. 5, pp. 2320–2328, 2008.

[35] Y. Komatsu, K. Fujii, S. Nakajima, K. Umetani, and K. Toyama, "Electrophysiological and morphological correlates in the development of visual cortical circuitry in infant kittens," *Developmental Brain Research*, vol. 22, no. 2, pp. 305–309, 1985.

[36] C. Koch, T. Poggio, and V. Torre, "Retinal ganglion cells: a functional interpretation of dendritic morphology," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 298, no. 1090, pp. 227–263, 1982.

[37] A. Destexhe and E. Marder, "Plasticity in single neuron and circuit computations," *Nature*, vol. 431, no. 7010, pp. 789–795, 2004.

[38] C. Koch, "Computation and the single neuron," *Nature*, vol. 385, no. 6613, pp. 207–210, 1997.

[39] J. N. J. Reynolds and J. R. Wickens, "Dopamine-dependent plasticity of corticostriatal synapses," *Neural Networks*, vol. 15, no. 4–6, pp. 507–521, 2002.

[40] Q. Gu, "Contribution of acetylcholine to visual cortex plasticity," *Neurobiology of Learning and Memory*, vol. 80, no. 3, pp. 291–301, 2003.

[41] A. Losonczy, J. K. Makara, and J. C. Magee, "Compartmentalized dendritic plasticity and input feature storage in neurons," *Nature*, vol. 452, no. 7186, pp. 436–441, 2008.

[42] W. Chen, J. Sun, S. Gao, J.-J. Cheng, J. Wang, and Y. Todo, "Using a single dendritic neuron to forecast tourist arrivals to Japan," *IEICE Transaction on Information and Systems*, vol. E100D, no. 1, pp. 190–202, 2017.

[43] Y. Yu, Y. Wang, S. Gao, and Z. Tang, "Statistical modeling and prediction for tourism economy using dendritic neural network," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 7436948, 9 pages, 2017.

[44] T. Jiang, S. Gao, D. Wang, J. Ji, Y. Todo, and Z. Tang, "A neuron model with synaptic nonlinearities in a dendritic tree for liver disorders," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 12, no. 1, pp. 105–115, 2017.

[45] J. Ji, S. Gao, J. Cheng, Z. Tang, and Y. Todo, "An approximate logic neuron model with a dendritic structure," *Neurocomputing*, vol. 173, pp. 1775–1783, 2016.

[46] Z. Tang, H. Tamura, M. Kuratu, O. Ishizuka, and K. Tanno, "A model of the neuron based on dendrite mechanisms," *Electronics and Communications in Japan, Part III: Fundamental Electronic Science (English translation of Denshi Tsushin Gakkai Ronbunshi)*, vol. 84, no. 8, pp. 11–24, 2001.

[47] R. C. Paolicelli, G. Bolasco, F. Pagani et al., "Synaptic pruning by microglia is necessary for normal brain development," *Science*, vol. 333, no. 6048, pp. 1456–1458, 2011.

[48] S.-T. Luo, B.-W. Cheng, and C.-H. Hsieh, "Prediction model building with clustering-launched classification and support vector machines in credit scoring," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7562–7566, 2009.

[49] Y. Peng, G. Wang, G. Kou, and Y. Shi, "An empirical study of classification algorithm evaluation for financial risk prediction," *Applied Soft Computing*, vol. 11, no. 2, pp. 2906–2915, 2011.

[50] S.-Y. Chang and T.-Y. Yeh, "An artificial immune classifier for credit scoring analysis," *Applied Soft Computing*, vol. 12, no. 2, pp. 611–618, 2012.

[51] X. Zhu, J. Li, D. Wu, H. Wang, and C. Liang, "Balancing accuracy, complexity and interpretability in consumer credit decision making: A C-TOPSIS classification approach," *Knowledge-Based Systems*, vol. 52, pp. 258–267, 2013.

[52] C.-F. Tsai, Y.-F. Hsu, and D. C. Yen, "A comparative study of classifier ensembles for bankruptcy prediction," *Applied Soft Computing*, vol. 24, pp. 977–984, 2014.

[53] L. Yu, S. Wang, and K. K. Lai, "Credit risk assessment with a multistage neural network ensemble learning approach," *Expert Systems with Applications*, vol. 34, no. 2, pp. 1434–1444, 2008.

[54] C.-F. Tsai, "Feature selection in bankruptcy prediction," *Knowledge-Based Systems*, vol. 22, no. 2, pp. 120–127, 2009.

[55] C. Koch, *Biophysics of Computation: Information Processing in Single Neurons*, Oxford University Press, Oxford, UK, 2004.

[56] J. Sietsma and R. J. Dow, "Neural net pruning-why and how," in *IEEE International Conference on Neural Networks*, vol. 1, pp. 325–333, 1988.

[57] A. Asuncion and D. J. Newman, UCI Machine Learning Repository, http://www.ics.uci.edu/mlearn/MLRepository.html, Irvine, CA: University of California , School of Information and Computer Science.

[58] C.-M. Wang and Y.-F. Huang, "Evolutionary-based feature selection approaches with new criteria for data mining: A case study of credit approval data," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5900–5908, 2009.

[59] G. Taguchi, R. Jugulum, and S. Taguchi, *Computer-Based Robust Engineering: Essentials for DFSS*, ASQ Quality Press, 2004.

[60] A. Khashman, "Credit risk evaluation using neural networks: emotional versus conventional models," *Applied Soft Computing*, vol. 11, no. 8, pp. 5477–5484, 2011.

[61] K. Coussement and W. Buckinx, "A probability-mapping algorithm for calibrating the posterior probabilities: A direct marketing application," *European Journal of Operational Research*, vol. 214, no. 3, pp. 732–738, 2011.

[62] S. S. Haykin, *Neural Networks and Learning Machines*, vol. 3, Pearson Upper Saddle River, New Jersey, NJ, USA, 2009.

[63] J. Yang and S. Olafsson, "Optimization-based feature selection with adaptive instance sampling," *Computers & Operations Research*, vol. 33, no. 11, pp. 3088–3106, 2006.