Research Article

An efficient learning algorithm for periodic perceptron to test XOR function and parity problem

Chittaranjan Mallick[1] · Sourav Kumar Bhoi[2] · Sanjaya Kumar Panda[3] · Kalyan Kumar Jena[2]

Received: 5 October 2019 / Accepted: 3 January 2020 / Published online: 7 January 2020

Abstract

Artificial neural network (ANN) is an important tool, which is used in numerous fields, such as computer vision, pattern recognition, signal processing, solving optimization problems, and voice analysis and synthesis. Many real-life problems, where the future events play a vital role, are based on the past history. For example, predicting the behavior of stock market indices and electrical load forecasting. In this paper, we establish an efficient learning algorithm for periodic perceptron (PP) in order to test in realistic problems, such as the XOR function and the parity problem. Here, the periodic threshold output function guarantees the convergence of the learning algorithm for the multilayer perceptron. By using the binary Boolean function and the PP in single and multilayer perceptron, XOR problem is solved. The performance of PP is compared with multilayer perceptron and the result shows superiority of PP over the multilayer perceptron.

Keywords Multilayer perceptron · Periodic perceptron · Learning algorithm · Parity problem

1 Introduction

Now-a-days, research on ANN is very much challenging and it is an emerging part of artificial intelligence (AI) [1–6]. It is a technique, which attempts to simulate the function of the human brain and implement it in machine intelligence. The main parts of the human brain are a network of neurons, synapses, axons, dendrites and others. The neurons (i.e., information processor) are interconnected through dendrites. The different neurons and dendrites meet to form synapses, which is the roadways for passing the information. The neuron receives electrochemical impulses through synapses. If the total impulses received by the neuron exceed a firm threshold value, the neuron fires and sends another impulse down to other neurons through the axon. Synapses help to create the connection between axon and other neurons. Therefore, a neuron receives a set of input impulse and sets out another pulse, which depends on the total input to it and its activation level in the brain. The information is processed through a large number of such neurons. Essentially, ANN is a graph with a set of nodes and arcs [6]. A generalized view of network structure and model of neuron is as follows (Fig. 1).

Here, y is the output of neuron and it is defined as follows.

$$f\left(\sum w_i x_i - \theta\right) \tag{1}$$

where w_i = weight of input signal i, $1 \le i \le n$, x_i = input signal i, $1 \le i \le n$, θ = threshold level and $f(\cdot)$ = a non-linear function.

A weight is multiplied with each input, which is analogous to a synaptic strength. The activation value is the product of the inputs and its corresponding weights ($\sum w_i x_i$) and it is compared with a given threshold value. If the summation is more than the given threshold

✉ Sanjaya Kumar Panda, sanjayauce@gmail.com; Chittaranjan Mallick, cmallick75@gmail.com; Sourav Kumar Bhoi, souravbhoi@gmail.com; Kalyan Kumar Jena, kalyan.cse@pmec.ac.in | [1]Department of Mathematics, Parala Maharaja Engineering College (Govt.), Berhampur 761003, India. [2]Department of Computer Science and Engineering, Parala Maharaja Engineering College (Govt.), Berhampur 761003, India. [3]Department of Computer Science and Engineering, National Institute of Technology, Warangal 506004, India.

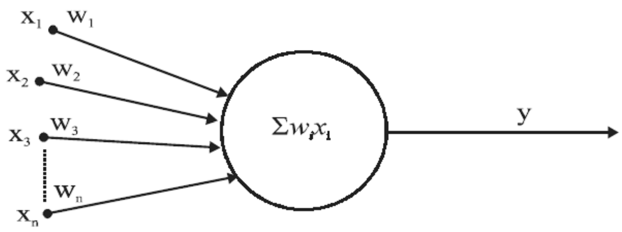

Fig. 1 A simple neuron model

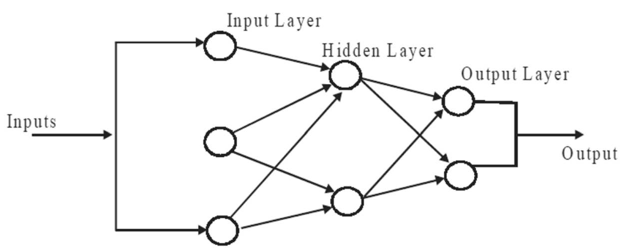

Fig. 2 Neural information processor

value, the threshold element calculates an output signal $(\sum_{i=1}^{n} w_i x_i - \theta)$ using activation. Note that the threshold function may be a sigmoid or a hyperbolic tangent function.

The primary contribution of this paper is stated as follows.

- A two-layer perceptron is developed with a periodic non-monotonous activation function (referred as PP) to compute any Boolean function.
- An efficient learning algorithm is proposed and a comparison is performed between PP and multilayer perceptron. It is observed that PP gives better result than the multilayer perceptron.
- PP is tested with realistic problems, such as the XOR function and the parity problem.

The paper is structured as follows. Section 2 discusses the background work. Section 3 shows the two-layer perceptron with non-monotonous activation function. Section 4 presents the results and discussion. At last, the conclusion is presented in Sect. 5.

2 Related works

Figure 2 gives the structure of a neural network (NN) that is represented by a set of nodes and a set arrows. This structure mainly deals with three layers, such as input, hidden and output layers. In order to function the NN, the weights are being initialized. As a result, the network is made to learn by using some learning methods and rules [2–15]. The connection weights are adjusted during the training. When the training is completed, the weights are fixed to some value. Note that learning of NN indicates parameter change and synaptic changes in brain or nervous system. There are various learning rules for NN, such as simple Hebbian, delta rule and generalized delta rule.

The two popular methods of learning are supervised and unsupervised. A number of well-known NN models have been built, such as perceptron, multilayered perceptron, adaptive resonance theory network and Boltzmann machine. ANN has gained immense popularity as a useful tool in modeling and simulation. Moreover, it is a mathematical model inspired by the working of a biological brain and borrows heavily from the literature of brain and memory modeling. It comes in many flavors, but, the most popular is the backpropagation model, which is synonymous with ANN. The most significant use of a backpropagation ANN is creating an approximate model of a system, whose response to a large set of stimuli is known and there is no need for creating a mathematical model of any particular kind. A system, which takes an *N*-dimensional stimulus vector, *X* and produces *M*-dimensional response vector *Y*, can generate a set of *P* stimulus pairs of the form {*X*, *Y*}. To model this system, we create a backpropagation network with a set of unknown weights that is attached to each connection and a non-linear transformation function on the internal nodes. Each internal node works on the following model.

$$\text{Output} = f(\text{sum of inputs}) \tag{2}$$

where *f* is highly non-linear function. The most popular one is the sigmoid function. The weights are estimated using an iterative stage, called training. The problem of training is as follows.

Given a function $y' = g(W, X)$, where, *X* is a stimulus vector, *y′* is the response vector and *W* is the set of weights assigned to the internal connections to find the value. It is required to minimize the least square error (*Er*) of the periodic function as follows.

$$E(W) = Er = \sum_{k=1}^{p} \sum_{j=1}^{M} (y'_{k,j} - y_{k,j})^2 \tag{3}$$

In Fig. 3, the inputs are denoted as $\{v_1, v_2, \ldots, vn\}$ and the weights are denoted as $\{w_1, w_2, \ldots, wn\}$.

The total input of the neuron is calculated as follows.

$$x = \sum_{i=1}^{n} v_i w_i \tag{4}$$

The output of the neuron is as follows.

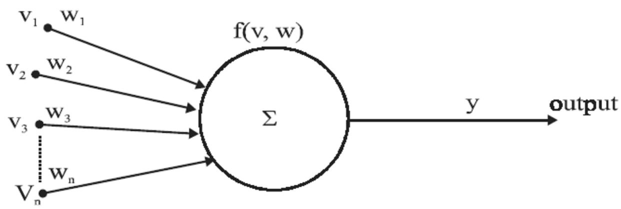

Fig. 3 A simple (McCulloch-Pitts) neurons

$$x = \sum_{i=1}^{n} v_i w_i - \theta \tag{5}$$

where θ is called as threshold associated with this neuron. In addition, there is a transfer function *f*(*x*), which provides the discrete and continuous output as follows.

$$f(x) = \begin{Bmatrix} 0 \text{ if } x \le 0 \\ 1 \text{ if } x > 0 \end{Bmatrix} \tag{6}$$

The below perceptron is called as Rosenblatt's perceptron [7].

$$f(x) = \frac{1}{(1 + e^{-x})} \tag{7}$$

Hornik et al. [16] have stated that a perceptron with a huge number of hidden layers can estimate any type of function. But, finding an optimal solution remains a crucial problem as addressed by Hinton [17]. Brady [18] has used periodic activation function to study the convergence of learning algorithm. Gioiello et al. [19] have used multilayer perceptron to study handwritten classification. Filliatre and Racca [20] have studied the PP for speech synthesis. Many such works have been presented in [21–25]. Hu et al. [21] have used two distributions, namely Cauchy and laplace and one error function, called Gaussian to generate novel activation functions. Moreover, they have compared three functions, namely sigmoid, hyperbolic tangent and normal distribution functions. Fawaz et al. [23] have focused on binary neural network and presented the usefulness of quantum amplitude amplification. Godfrey [25] has stated that most of the literautres are relying on one or two activation functions throughout the network. As a result, they have studied various heterogeneous activation functions and their possible applications.

3 Two-layer perceptron with non-monotonous activation function

Let N_i be the *i*th neuron receiving input signals $\{s_1, s_2, \ldots, s_n\}$. Let I_i be the total stimulus (input) and O_i be the output, which are mathematically expressed as follows.

$$I_i = \sum_{i=1}^{n} s_i \text{ and } O_i = f(I_i) \tag{8}$$

where *f* is denoted as activation function (Fig. 4).

Let us consider that there are three neurons (Fig. 5) in which two neurons in the first layer and one neuron in the second layer. The input and output of the neurons are binary (i.e., 0 and 1). The activation of two neurons, namely N_1 and N_2 are given by the system, when the input neuron is equal to their excitation, that is, $O_1 = I_1$ and $O_2 = I_2$. The activity of the neuron N_3 is given as follows.

$$O_3 = C_r\left(W_{1,3}O_1 + W_{2,3}O_2\right) \tag{9}$$

where C_r is the crenel function, which is defined as follows.

$$C_r(x) = \begin{Bmatrix} I & \text{if } T_1 \le x \le T_2 \\ O & \text{Otherwise} \end{Bmatrix} \tag{10}$$

where T_1 and T_2 are the threshold of the neuron N_1 and N_2.

The crenel function of the PP is given as follows (Fig. 6).

The weights $W_{1,3}$ and $W_{2,3}$ can be taken as follows.

$$w_{1,3} = w_{2,3} = \frac{T_1 + T_2}{2} \tag{11}$$

The XOR function calculation by PP is shown in Table 1.

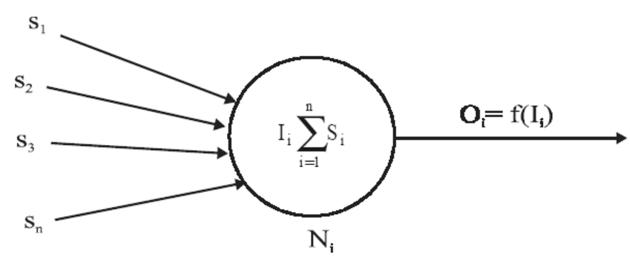

Fig. 4 Output signal of the input given to the neuron

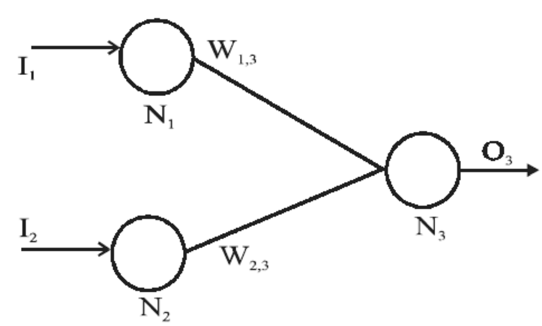

Fig. 5 The network of PP with three neurons

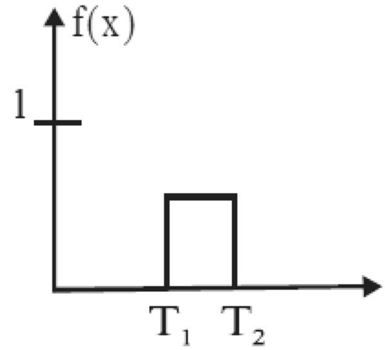

Fig. 6 The crenel function of PP

Table 1 XOR function calculation by PP

O_1=I_1	O_2=I_2	I_3=$W_{1,3}\,O_1+W_{2,3}\,O_2$	O_3
0	0	0	0
0	1	$\frac{T_1+T_2}{2}=\frac{1}{2}$	1
1	0	$\frac{T_1+T_2}{2}=\frac{1}{2}$	1
1	1	T_1+T_2	0

We can adopt the following rule for changing periodicity of the activation function.

Let $f_r^k(x)$ be the activation function with period 2 *k*. Here, $f_r^k(x)$=$f_r(\frac{x}{k})$ with weight matrix W_k. Based on the above facts, the following theorem is proved.

Theorem 1 *Every Boolean function can be evaluated by using a periodic perceptron with three neurons.*

Proof Consider the network of the PP of three neurons as shown in Fig. 5, where the output neuron (i.e., N_3) is provided by the activation function f_r. The following need to be considered to compute the Boolean function *f*.

$$\phi(0,0) = f_r(0) \tag{12}$$

$$\phi(1,0) = f_r(W_{1,3}) \tag{13}$$

$$\phi(0,1) = f_r(W_{2,3}) \tag{14}$$

$$\phi(1,1) = f_r(W_{1,3} + W_{2,3}) \tag{15}$$

Here, *r* is determined as $\left\{ \begin{array}{l} r = 0 \text{ if } \phi(0,0) = 0 \\ r = 1 \quad \text{Otherwise} \end{array} \right\}.$ □

The weights $W_{1,3}$ and $W_{2,3}$ are selected randomly in order to satisfy Eqs. (13) and (14). As f_v is a periodic (i.e., period 2), there exists an interval length *l* around $W_{1,3}$, where Eqs. (13) and (14) are fulfilled. If [x_1, x_2] is the interval, where $W_{1,3}$ satisfying Eq. (13) and [x_3, x_4] is the interval satisfying Eq. (14), then $W_{1,3}$ and $W_{2,3}$ sweep through these intervals and through an interval of length 2 ([x_1+x_3, x_2+x_4]), where f_r ($W_{1,3}+W_{2,3}$) is equals to 0 and 1 alternately.

The learning algorithm for PP is shown in Table 2, which is adopted based on the delta learning method.

Table 2 Leaning algorithm for PP

Input: *n* input neurons, {N_1, N_2,…, N_n}
Output: N_{n+1}

1. Give the input vector {I_1, I_2,…, I_n} for every neuron N_j and $I_j = O_j$, where O_j is the output vector
2. Initialize all weights (w_{ij}) to zero
3. Calculate $I_{n+1} = \sum W_{i,n+1}$ for N_{n+1} neuron
4. Compute $O_{n+1} = f_r(I_{n+1})$
5. Check that O_{n+1} is the desired output and if not, it gives no right parity
6. Change the weights $W_{i,n+1}$ for all i = 1, 2,…, n
7. **if** $I_{n+1} - f_r(I_{n+1}) \leq 0.5$
8. Compute $\Delta_{n+1} = f_r(I_{n+1}) - I_{n+1} - 0.5$
9. else
10. Compute $\Delta_{n+1} = f_r(I_{n+1}) - I_{n+1} + 1.5$
11. Δ_n being the *Er*
12. $W_{i,n+1}(t+1) = W_{i,n+1}(t) - \alpha r \Delta_{n+1}$

Table 3 Test results

Test ID	Error (E_r)	Learning rate (a)
1	116	0.950
	37	0.950
	27	0.803
	15	0.613
	12	0.385
	6	0.328
	4	0.214
	2	0.176
	0	0.138
2	337	0.950
	35	0.950
	22	0.765
	17	0.518
	12	0.423
	10	0.328
	5	0.290
	3	0.195
	0	0.157

4 Results and discussion

In this section, the results are computed to test the performance of the learning algorithm for PP in order to find the Boolean function. The algorithm uses the XOR function and the parity concept to get the results.

1. The XOR function: In this problem, we train PP with three neurons and we compute XOR function.
2. The parity problem: Let A be a set of n-bit vector. The set splits into A_0 and A_1, where A_0 includes odd number of 0's and A_i includes the others.

For t instance of time, let a be the learning rate and r is the correction factor and $0 \le r \le 1$. The test results for the activation function with the delta learning rule are shown in Table 3. Periodic perceptron is used in such a way that the remaining hidden layer gives the same output. The algorithm is an efficient one for finding the Boolean function.

5 Conclusion

In this paper, we have observed that a two-layer perceptron with a periodic non-monotonous activation function can compute any Boolean function. An efficient learning algorithm for periodic perceptron has been proposed to test two realistic problems, such as the XOR function and the parity problem. The performance of PP have compared with multilayer perceptron and it has been observed that PP gives better result than the multilayer perceptron. In future, this work can be extended by adding the deep nueral network (DNN) and/or convolution neural network (CNN) concept to analyze the error.

Compliance with ethical standards

Conflict of interest The authors declare that they have no competing interests.

References

1. Esteves JT, de Souza Rolim G, Ferraudo AS (2019) Rainfall prediction methodology with binary multilayer perceptron neural networks. Clim Dyn 52(3–4):2319–2331
2. Mileiko S, Shafik R, Yakovlev A, Edwards J (2019). A pulse width modulation based power-elastic and robust mixed-signal perceptron design. In: 2019 design, automation & test in Europe conference & exhibition (DATE). IEEE, pp 1603–1606
3. Sakar CO, Polat SO, Katircioglu M, Kastro Y (2019) Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks. Neural Comput Appl 31:6893–6908
4. Yamamoto AY, Sundqvist KM, Li P, Harris HR (2018) Simulation of a multidimensional input quantum perceptron. Quantum Inf Process 17(6):128
5. Amaral RPF, Ribeiro MV, de Aguiar EP (2019) Type-1 and singleton fuzzy logic system trained by a fast scaled conjugate gradient methods for dealing with binary classification problems. Neurocomputing 355:57–70
6. Struye J, Latré S (2019) Hierarchical temporal memory and recurrent neural networks for time series prediction: an empirical validation and reduction to multilayer perceptrons. Neurocomputing. https://doi.org/10.1016/j.neucom.2018.09.098
7. Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65(6):386–408
8. Heidari AA, Faris H, Aljarah I, Mirjalili S (2019) An efficient hybrid multilayer perceptron neural network with grasshopper optimization. Soft Comput 23(17):7941–7958
9. Lima-Junior FR, Carpinetti LCR (2019) Predicting supply chain performance based on SCOR metrics and multilayer perceptron neural networks. Int J Prod Econ 212:19–38
10. Li Y, Tang G, Du J, Zhou N, Zhao Y, Wu T (2019) Multilayer perceptron method to estimate real-world fuel consumption rate of light duty vehicles. IEEE Access 7:63395–63402
11. Bhowmik M, Muthukumar P, Anandalakshmi R (2019) Experimental based multilayer perceptron approach for prediction of evacuated solar collector performance in humid subtropical regions. Renew Energy 143:1566–1580
12. Tang X, Zhang L, Ding X (2019) SAR image despeckling with a multilayer perceptron neural network. Int J Dig Earth 12(3):354–374
13. Pełka P, Dudek G (2019) Pattern-based forecasting monthly electricity demand using multilayer perceptron. In: International conference on artificial intelligence and soft computing. Springer, Cham, pp 663–672

14. Wang SH, Zhang Y, Li YJ, Jia WJ, Liu FY, Yang MM, Zhang YD (2018) Single slice based detection for Alzheimer's disease via wavelet entropy and multilayer perceptron trained by biogeography-based optimization. Multimedia Tools Appl 77(9):10393–10417
15. Díaz-Álvarez A, Clavijo M, Jiménez F, Talavera E, Serradilla F (2018) Modelling the human lane-change execution behaviour through Multilayer Perceptrons and Convolutional Neural Networks. Transp Res Part F Traffic Psychol Behav 56:134–148
16. Hornik K (1991) Approximation capabilities of multilayer feedforward networks. Neural Netw 4(2):251–257
17. Hinton GE (1989) Connectionist learning procedures. Artif Intell 40(1–3):185–234
18. Brady MJ (1990) Guaranteed learning algorithm for network with units having periodic threshold output function. Neural Comput 2(4):405–408
19. Gioiello G, Sorbello F, Vassallo G, Vitabile S (1996) A new VLSI neural device with sinusoidal activation function for handwritten classification. In: Proceedings of 2nd international conference on neural networks and their applications, pp 238–242
20. Filliatre B, Racca R (1996) Multi-threshold neurones perceptron. Neural Process Lett 4(1):39–44
21. Hu J, Xu L, Wang X, Xu X, Su G (2018) Effects of BP algorithm-based activation functions on neural network convergence. J Comput 29(1):76–85
22. Du KL, Swamy MNS (2019) Multilayer perceptrons: architecture and error backpropagation. In: Neural networks and statistical learning. Springer, London, pp 97–141
23. Fawaz A, Klein P, Piat S, Severini S, Mountney P (2019). Training and meta-training binary neural networks with quantum computing. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. ACM, pp 1674–1681
24. Laudani A, Lozito GM, Fulginei FR, Salvini A (2015) On training efficiency and computational costs of a feed forward neural network: a review. Comput Intell Neurosci 2015:83
25. Godfrey LB (2018) Parameterizing and aggregating activation functions in deep neural networks. Theses and Dissertations. University of Arkansas, Fayetteville. Retrieved from https://scholarworks.uark.edu/etd/2655