

Research Article

A Novel Single Neuron Perceptron with Universal Approximation and XOR Computation Properties

Ehsan Lotfi¹ and M.-R. Akbarzadeh-T²

¹ Department of Computer Engineering, Torbat-e-Jam Branch, Islamic Azad University, Torbat-e-Jam, Iran

² Electrical and Computer Engineering Departments, Center of Excellence on Soft Computing and Intelligent Information Processing, Ferdowsi University of Mashhad, Iran

Correspondence should be addressed to Ehsan Lotfi; esilotf@gmail.com

Received 9 February 2014; Accepted 7 April 2014; Published 28 April 2014

Academic Editor: Cheng-Jian Lin

Copyright © 2014 E. Lotfi and M.-R. Akbarzadeh-T. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

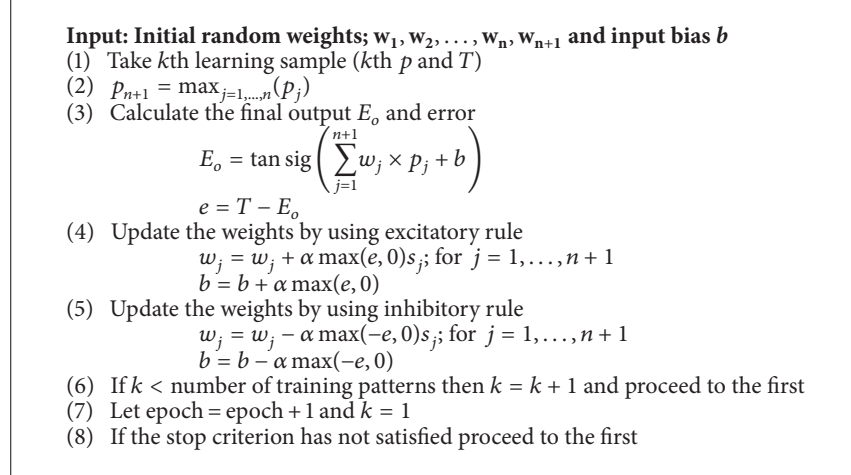
We propose a biologically motivated brain-inspired single neuron perceptron (SNP) with universal approximation and XOR computation properties. This computational model extends the input pattern and is based on the excitatory and inhibitory learning rules inspired from neural connections in the human brain's nervous system. The resulting architecture of SNP can be trained by supervised excitatory and inhibitory online learning rules. The main features of proposed single layer perceptron are universal approximation property and low computational complexity. The method is tested on 6 UCI (University of California, Irvine) pattern recognition and classification datasets. Various comparisons with multilayer perceptron (MLP) with gradient decent backpropagation (GDBP) learning algorithm indicate the superiority of the approach in terms of higher accuracy, lower time, and spatial complexity, as well as faster training. Hence, we believe the proposed approach can be generally applicable to various problems such as in pattern recognition and classification.

1. Introduction

In various computer applications such as pattern recognition, classification, and prediction, a learning module can be implemented by various approaches including statistical, structural, and neural approaches. Among these methods, artificial neural networks (ANNs) are inspired by physiological workings of the brain. They are based on mathematical model of single neural cell (neuron) named single neuron perceptron (SNP) and try to resemble the actual networks of neurons in the brain. As computational models, SNP has particular characteristics such as the ability to learn and generalize. Although the multilayer perceptron (MLP) can approximate any functions [1, 2], traditional SNP is not universal approximator. MLP can learn through the error backpropagation algorithm (EBP), whereby the error of output units is propagated back to adjust the connecting weights within the network. In MLP architecture, by increasing the number of neurons in input layer or (and) the number of neurons in output layer or (and) the number of neurons

in hidden layer(s), the number of learning parameters and the algorithm computational complexity are significantly increased. This problem is usually referred to as the curse of dimensionality [3, 4]. So many researchers have tried to propose more powerful single layer architectures and faster algorithms such as functional link networks (FLNs) and Levenberg-Marquardt (LM) and its modified and extended versions [5–20].

In contrast to the MLP, SNP and FLNs do not impose high computational complexity and are far from the curse of dimensionality. But because of disregarding the universal approximation property, SNP and FLNs are not very popular in the applications. In contrast to the previous knowledge about SNP, this paper aims to propose a novel SNP model that can solve the XOR problem and we show that it can be universal approximator. Proposed SNP can solve XOR problem only if additional nonlinear operator is used. As illustrated in the next section, the SNP universal approximation property can simply be achieved by extending the input patterns and using the nonlinear operator max. Like functional link networks



ALGORITHM 1: Proposed SNP algorithm.

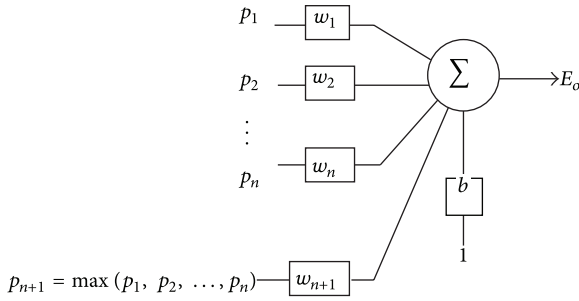


FIGURE 1: Proposed SNP.

(FLNs) [21], the proposed SNP does not include hidden units or expand the input vector, but guarantees universal approximation. FLNs are single-layer neural networks that can be considered as an alternative approach in the data mining to overcome the complexities associated with MLP [22] but they do not guarantee universal approximation.

The paper is organized as follows. Proposed SNP and universal approximation theorem are proposed in Section 2. Section 3 presents the numerical results, where the proposed SNP is compared with backpropagation MLP. There are various versions of backpropagation algorithms. In classification problems, we compare with gradient descent backpropagation (GDBP) [23], that is, the standard basic algorithm. Finally, conclusions are made in Section 4.

2. Proposed Single Neuron Perceptron

Figure 1 shows the proposed SNP. In the figure, the model is presented as $n + 1$ -inputs single-output architecture. The variable p is the input pattern and the variable T is related target applied in the learning process (3). Let us extend the input pattern as follows:

$$p_{n+1} = \max_{j=1, \dots, n} (p_j). \quad (1)$$

Actually, max operation increases the input dimension to $n + 1$.

So, the new input pattern has $n + 1$ elements. In Figure 1, the input pattern is illustrated by vector $p_{1 \leq j \leq n+1}$ and the E_o calculated by the following formula is the final output:

$$E_o(p) = f \left(\sum_{j=1}^{n+1} w_j \times p_j + b \right), \quad (2)$$

where f is activation function and w_1, w_2, \dots, w_{n+1} , and b are adjustable weights. So, error can be achieved as follows:

$$e = T - E_o \quad (3)$$

and the learning weights can be adjusted by the following excitatory learning rule:

$$w_j = w_j + \alpha \max(e, 0) p_j; \quad \text{for } j = 1, \dots, n + 1 \quad (4)$$

and then by the following inhibitory rule:

$$w_j = w_j - \alpha \max(-e, 0) p_j; \quad \text{for } j = 1, \dots, n + 1, \quad (5)$$

where T is target, E_o is output of network, e is related error, and α is the learning rate. Also b can be trained by

$$\begin{aligned} b &= b + \alpha \max(e, 0), \\ b &= b - \alpha \max(-e, 0). \end{aligned} \quad (6)$$

It should be added that max operation applied on the input pattern and also in the learning phase has been motivated from computational models of limbic system in the brain [24–26]. Limbic system is an emotional processor in the mammalian brain [27–29]. In these models [24–26], the max operator prepares the output and input of main parts of limbic system.

In summary, the feedforward computation and backward learning algorithm of proposed SNP, in an online form and with *tansig* activation function, is as in Algorithm 1.

In the algorithm, α can be picked empirically or changed adaptively during the learning process according to the adaptive learning [30, 31].

The proposed SNP solves the XOR problem. Consider 2–1 architecture with hardlim activation function and by using the following weights: $w_1 = -1$, $w_2 = -2$, $w_3 = 2$, and $b = -1$; thus,

$$\begin{aligned}\widehat{f}(1, 1) &= \text{hardlim}(w_1 \times 1 + w_2 \times 1 + w_3 \times \max(1, 1) + b) \\ &= 0 \\ \widehat{f}(0, 0) &= \text{hardlim}(w_1 \times 0 + w_2 \times 0 + w_3 \times \max(0, 0) + b) \\ &= 0 \\ \widehat{f}(1, 0) &= \text{hardlim}(w_1 \times 1 + w_2 \times 0 + w_3 \times \max(1, 0) + b) \\ &= 1 \\ \widehat{f}(0, 1) &= \text{hardlim}(w_1 \times 0 + w_2 \times 1 + w_3 \times \max(0, 1) + b) \\ &= 1,\end{aligned}\tag{7}$$

where hardlim is calculated by the following formula:

$$\text{hardlim}(x) = \begin{cases} 1 & \text{if } (x \geq 0) \\ 0 & \text{otherwise.} \end{cases}\tag{8}$$

Since \widehat{f} is in the form of (2), so \widehat{f} based on SNP can approximate the XOR function. The proposed model has a lower computational complexity than other methods such as spiking neural networks [32] that solved XOR problem. The computational complexity of proposed SNP is $O(n)$; this is while it profits from very simple questions adjusting the weights.

In the next section, we prove that SNP is a universal approximator and can approximate all real continuous functions.

2.1. Universal Approximation Theorem. Let us ignore the activation function from the model and rewrite (2) like this

$$E(s) = \sum_{j=1}^n (w_j) \times s_j + b.\tag{9}$$

Consider Y as the set of all equations in form (9) and $d_\infty(E_1, E_2) = \sup_{p \in U} |E_1(p) - E_2(p)|$ as a submetric; then (Y, d_∞) is ametric space [33]. The following theorem shows that (Y, d_∞) is dense in $(C[U], d_\infty)$, where $C[U]$ is the set of all real continues functions defined on U .

SNP Universal Approximation Theorem. For any given real continuous function g on the compact set $U \subset R^n$ and arbitrary $\varepsilon > 0$, there exists $E \in Y$ such that

$$\sup_{p \in U} |g(p) - E(p)| < \varepsilon.\tag{10}$$

We use the following Stone-Weierstrass theorem to prove the theorem.

Stone-Weierstrass Theorem (see [33, 34]). Let Y be a set of real continuous functions on compact set U . If (1) Y is algebra, that is, the set Y is closed under scalar multiplication (the closing under addition and multiplication is not necessary for real continuous functions [34]), (2) Y separates points on U , that is, for every $x, y \in U$ such that $x \neq y$, there exists $E \in Y$ such that $E(x) \neq E(y)$; and (3) Z vanishes at no point of U , that is, for each $x \in U$, there exists $E \in Y$ such that $E(x) \neq 0$, then the uniform closure of Y consists of all real continuous functions on U ; that is, (Y, d_∞) is dense in $(C[U], d_\infty)$.

SNP Universal Approximation Proof. First, we prove that (Y, d_∞) is algebra. Let $E \in Y$, for arbitrary $c \in R$:

$$E(p) = \sum_{j=1}^n w_j p_j + b\tag{11}$$

$$cE(p) = \sum_{j=1}^n cw_j p_j + cb$$

which is given in form (8). Thus, $cE \in Y$ and (Y, d_∞) is an algebra.

Next, we prove that (Y, d_∞) separates the points on U . We prove this by constructing a required E , for arbitrary $p^1, p^2 \in U$ such that $p^1 \neq p^2$, and we choose w, v such that $w_j = p_j^1$, for $j = 1, \dots, n$ and $v_{n+1} = w_{n+1} = 0$.

Thus,

$$\begin{aligned}p^1 \neq p^2 &\implies p^1 \cdot p^1 \neq p^1 \cdot p^2 \\ &\implies \sum_{j=1}^n (w_j) p_j^1 \neq \sum_{j=1}^n (w_j) p_j^2 \implies E(p^1) \neq E(p^2).\end{aligned}\tag{12}$$

Therefore, (Y, d_∞) separates the point on U .

Finally, we prove that (Y, d_∞) vanishes at no point of U . We choose $w_j = 0$, for $j = 1, \dots, n$, $v_{n+1} = 0$, $w_{n+1} > 0$, and $b = 1$.

Since, for all $p \in R^n$,

$$\sum_{j=1}^n (w_j) p_j = 0\tag{13}$$

and $b > 0$, then, for all $p \in R^n$, there exists E such that $E(p) \neq 0$.

So, SNP independently from activation function is universal approximator.

3. Numerical Results

One parameter that related to computational complexity of a learning method is the number of learning weights in each epoch. The lower number of learning weights concludes lower number of computations and lower computational complexity. To evaluate the number of proposed SNP learning weights

TABLE 1: Datasets and related learning information.

ID	Dataset Name	Dataset information			Parameters Learning rates	ENN model		MLP model		Comparison Rw
		Instance	Class	Attribute		Architecture	Weights	Architecture	Weights	
1	Diabetes	768	2	8	0.050	9-1	10	8-2-1	21	52%
2	Heart	270	2	13	0.050	14-1	15	13-2-1	31	51%
3	Pima	768	2	8	0.005	9-1	10	8-2-1	21	52%
4	Ionosphere	351	2	34	0.050	35-1	36	34-2-1	73	50%
5	Sonar	208	2	60	0.0005	61-1	62	60-2-1	125	50%
6	Tic-tac	958	2	9	0.0005	10-1	11	9-2-2	23	52%

with respect to the MLP, we propose a measure named the reducing ratio of number of weights (Rw) as follows:

$$Rw = \left(1 - \frac{\text{Number Of Learning Weights of SNP Model}}{\text{Number Of Learning Weights of MLP Model}} \right) \times 100. \quad (14)$$

The Rw is a measure that can be used to compare the computational complexity of proposed SNP and MPL. The higher Rw shows SNP has a lower number of learning weights. Thus, it has a lower number of computations and so has a lower computational complexity. Additionally, in the classification problems, the accuracy can be a proper performance measure to evaluate the algorithms. This measure is generally expressed as follows:

$$\text{Accuracy} = \frac{\text{Correct Detection}}{\text{All}}. \quad (15)$$

For all learning scenarios listed below, the training set contained 70% while the testing set contained 15% of the data and the remaining was used for the validation set. Input patterns have been normalized between [0 1]. Output targets are binary digits (i.e., the single class is labeled by digits “1” and “0,” the two classes are labeled as “01” and “10,” and the three classes are labeled as “001,” “010,” and “100,” and...). Also the initial weights were randomly selected between [0 1].

Here and prior to entering comparative numerical studies, let us analyze the computational complexity. Regarding the proposed learning algorithm, the algorithm adjusts $O(2n)$ weights for each learning sample, where n is number of input attributes. In contrast, computational time is $O(cn)$ for MLP, where c is number of hidden neurons (the lowest c is 2). Additionally, GDBP MLP compared here is based on derivative computations which impose high complexity, while the proposed method is derivative free. So, the proposed method has lower computational complexity and higher efficiency with respect to the MLP. This improved computing efficiency can be important for online predictions, especially when the time interval of observations is small.

To test and assess the SNP in classification, 6 single class datasets have been downloaded from UCI (University of California, Irvine) Data Center. In all datasets, the target

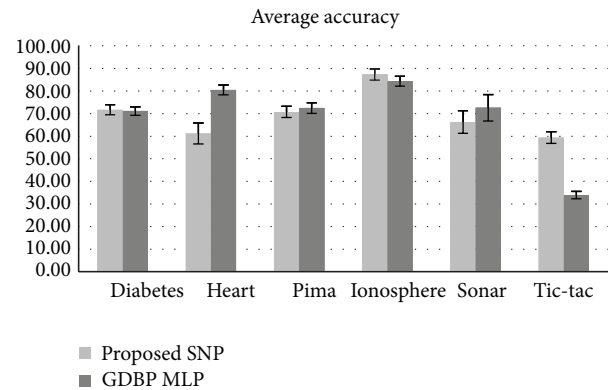


FIGURE 2: Comparisons between SNP and MLP.

labeling was binary. Table 1 shows the information related to the datasets that include the number of attributes and instances. Additionally, the SNP and MLP architectures and the number of learning weights and Rw are presented in the table too. As illustrated in Table 1, SNP reduces the number of learning weights approximately about 50% for each dataset.

In the proposed SNP algorithm, we consider $b = 0$. And the learning parameters values are shown in Table 1. The activation function was *tansig* and the stop criterion in learning process was the maximum epochs, which means the maximum number of epochs has been reached. The maximum and minimum values of each dataset were determined and the scaled data (between 0 and 1) were used to adjust the weights. The training was repeated 10 times and the average of accuracy in test set was recorded. Figure 2 presents the accuracy average and the confidence interval obtained from SNP and MLP. It is obvious that SNP is more accurate than MLP with GDBP algorithm in some datasets. The results indicated in Figure 2 are based on student’s *t*-test with 95% confidence.

Although, according to Figure 2, it seems that GDBP is better in some cases, what is very important in the results is number of learning epochs. Table 2 shows the learning epoch comparisons. According to Table 2, MLP needs many epochs to reach the results of SNP. It is the main feature of proposed SNP, fast learning with lower computational complexity, that makes it suitable for usage in various applications and especially in online problems.

TABLE 2: Number of learning epoch comparison.

Model	Proposed SNP	GDBP MLP
Diabetes	6091 ± 1973	9500 ± 1046
Heart	3833 ± 2098	10000 ± 0
Pima	6245 ± 2021	10000 ± 0
Ionosphere	2632 ± 1638	10000 ± 0
Sonar	922 ± 1027	9111 ± 2247
Tic-tac	628 ± 576	9952 ± 98
Average	3392	9760

4. Conclusion

In this paper, we prove that a single neuron perceptron (SNP) can solve XOR problem and can be a universal approximator. These features can be achieved by extending input pattern and by using max operator. SNP with this extension ability is a novel computational model of neural cell that is learnt by excitatory and inhibitory rules. This new SNP architecture works with fewer numbers of learning weights. Specifically, it only generates $O(2n)$ learning weights and only requires $O(2n)$ operations during each training iteration, where n is size of input vector. Furthermore, the universal approximation property is theoretically proved for this architecture. The source code of proposed algorithm is accessible from <http://www.bitools.ir/projects.html>. In numerical studies, SNP was utilized to classify 6 UCI datasets. The comparisons between proposed SNP and backpropagation MLP present the following conclusions. Firstly, the number of learning parameters of SNP is much lower with respect to the standard MLP. Secondly, in classification problems, the performance of supervised excitatory and inhibitory learning algorithm is higher than gradient descent backpropagation (GDBP). Thirdly, lower computational complexity caused from the fewer learning parameters and faster training of proposed SNP make it suitable for real time classification. In short, SNP is a universal approximator with a simple structure and is motivated by neurophysiological knowledge of the human's brain. We believe, based on the multiple case studies as well as the theoretical results in this report, that SNP can be effectively used in pattern recognition and classification problems.

Conflict of Interests

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

References

- [1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [2] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [3] S. Bengio and Y. Bengio, "Taking on the curse of dimensionality in joint distributions using neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 550–557, 2000.
- [4] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *Computational Intelligence and Bioinspired Systems*, vol. 3512, pp. 758–770, Springer, Berlin, Germany, 2005.
- [5] B. Verma, "Fast training of multilayer perceptrons," *IEEE Transactions on Neural Networks*, vol. 8, no. 6, pp. 1314–1320, 1997.
- [6] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.
- [7] J.-X. Peng, K. Li, and G. W. Irwin, "A new Jacobian matrix for optimal learning of single-layer neural networks," *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 119–129, 2008.
- [8] J.-M. Wu, "Multilayer Potts perceptrons with Levenberg-Marquardt learning," *IEEE Transactions on Neural Networks*, vol. 19, no. 12, pp. 2032–2043, 2008.
- [9] B. M. Wilamowski and H. Yu, "Improved computation for levenbergmarquardt training," *IEEE Transactions on Neural Networks*, vol. 21, no. 6, pp. 930–937, 2010.
- [10] B. M. Wilamowski and H. Yu, "Neural network learning without backpropagation," *IEEE Transactions on Neural Networks*, vol. 21, no. 11, pp. 1793–1803, 2010.
- [11] K. Y. Huang, L. C. Shen, K. J. Chen, and M. C. Huang, "Multilayer perceptron learning with particle swarm optimization for well log data inversion," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '12)*, pp. 1–6, Brisbane, Australia, June 2012.
- [12] N. Ampazis and S. J. Perantonis, "Two highly efficient second-order algorithms for training feedforward networks," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1064–1074, 2002.
- [13] C.-T. Kim and J.-J. Lee, "Training two-layered feedforward networks with variable projection method," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 371–375, 2008.
- [14] S. Ferrari and M. Jensenius, "A constrained optimization approach to preserving prior knowledge during incremental training," *IEEE Transactions on Neural Networks*, vol. 19, no. 6, pp. 996–1009, 2008.
- [15] Y. Liu, J. A. Starzyk, and Z. Zhu, "Optimized approximation algorithm in neural networks without overfitting," *IEEE Transactions on Neural Networks*, vol. 19, no. 6, pp. 983–995, 2008.
- [16] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Real-time learning capability of neural networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 863–878, 2006.
- [17] A. Bortoletti, C. di Fiore, S. Fanelli, and P. Zellini, "A new class of quasi-Newtonian methods for optimal learning in MLP-networks," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 263–273, 2003.
- [18] V. V. Phansalkar and P. S. Sastry, "Analysis of the backpropagation algorithm with momentum," *IEEE Transactions on Neural Networks*, vol. 5, no. 3, pp. 505–506, 1994.
- [19] A. Khashman, "A modified backpropagation learning algorithm with added emotional coefficients," *IEEE Transactions on Neural Networks*, vol. 19, no. 11, pp. 1896–1909, 2008.
- [20] A. Khashman, "Modeling cognitive and emotional processes: a novel neural network architecture," *Neural Networks*, vol. 23, no. 10, pp. 1155–1163, 2010.
- [21] A. Sierra, J. A. Macias, and F. Corbacho, "Evolution of functional link networks," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, pp. 54–65, 2001.

- [22] B. B. Misra and S. Dehuri, "Functional link artificial neural network for classification task in data mining," *Journal of Computer Science*, vol. 3, no. 12, pp. 948–955, 2007.
- [23] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*, PWS, Boston, Mass, USA, 1996.
- [24] E. Lotfi and M.-R. Akbarzadeh-T, "Adaptive brain emotional decayed learning for online prediction of geomagnetic activity indices," *Neurocomputing*, vol. 126, pp. 188–196, 2014.
- [25] E. Lotfi and M.-R. Akbarzadeh-T, "Brain emotional learning based pattern recognizer," *Cybernetics and Systems*, vol. 44, no. 5, pp. 402–421, 2013.
- [26] E. Lotfi and M.-R. Akbarzadeh-T, "Supervised brain emotional learning," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '12)*, pp. 1–6, Brisbane, Australia, June 2012.
- [27] J. E. LeDoux, "Emotion circuits in the brain," *Annual Review of Neuroscience*, vol. 23, pp. 155–184, 2000.
- [28] J. Le Doux, *The Emotional Brain*, Simon and Schuster, New York, NY, USA, 1996.
- [29] D. Goleman, *Emotional Intelligence; Why It Can Matter More Than IQ*, Bantam, New York, NY, USA, 2006.
- [30] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *1960 IRE WESCON Convention Record*, pp. 96–104, IRE, New York, NY, USA, 1960.
- [31] B. Widrow and S. D. Sterns, *Adaptive Signal Processing*, Prentice-Hall, New York, NY, USA, 1985.
- [32] S. Ferrari, B. Mehta, G. di Muro, A. M. J. VanDongen, and C. Henriquez, "Biologically realizable reward-modulated hebbian training for spiking neural networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN '08)*, pp. 1780–1786, Hong Kong, June 2008.
- [33] L.-X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 807–814, 1992.
- [34] W. Rudin, *Principles of Mathematical Analysis*, McGraw-Hill, New York, NY, USA, 1976.