ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

# XOR CLASSIFICATION

**Prof. Purushottam Das*, Prof. Ankur Singh Bist**
* Professor, Graphic Era Hill University, India
Professor, KIET, Ghaziabad, India

## ABSTRACT

Classification is an important issue in the field of computer science research. There are various methods for classification for complex problems. The level of accuracy depends on complexity of problem. Xor classification is an open problem and there are many approaches to solve it. In this paper we will discuss some basic outline and techniques proposed in this direction..

**KEYWORDS**: XOR problem.

## INTRODUCTION

### 1. The simple perceptron and the XOR problem

This approach introduces a new inequalities index, which was inspired from the simple perceptron's pattern in the artificial neural networks (ANN) scientific field and nominated by the authors Modulus Perceptron Inequalities Index (MPII). The structural similarities of this ANN pattern with a binary logic gate, in the Digital Electronics Theory (DET), lead to deal with the existence of operational similarities between these two models. The introduced index aroused while examining the potentials of the simple perceptron to solve the non-linear separable ―exclusive disjunction‖ (XOR) problem, by using techniques of the Theory of Numbers in Mathematics. Since the XOR architecture is considered by the DET an inequality detector, due to its ability to result to the same outcome in the same input values, the solution of the XOR problem for a simple perceptron authorizes the ANN and DET XOR models to be considered identical, in the extend that inequalities is regarded [1].

Simple perceptron (Figure 1) constitutes a solo neuron ANN pattern, consisting of n incoming synapses with weights wi that receive n-input signals si (si = 0 or 1, i=1, 2, 3,…, n). Inside the neuron a weighted summation takes place, due to relation (1) [2].

(1) $$S = \sum_{i}^{n} S_i W_i$$

When the sum of products is higher than a threshold value θ (S >= θ) the neuron activates the value 1, otherwise the neuron stays inactive [1]. The XOR operation [3], [4], concerns a logical function, which recommends a combination of the basic AND, OR and NOT logical operations (functions) as described at relation (2).

(2) XOR: A (XOR) B = {A (AND) NOTB} (OR) {NOTA (AND) B}

The XOR gate operator (Figure 1) receives two binary inputs (into A and B receptors) and results a binary output, as shown in Table 1. When the input signals (A, B) have the same value the outcome turns to zero, otherwise it turns to monad. The XOR truth table (Table 1) elects the XOR's attribute to indicate inequality (activates 1 when the inputs are unequal).

It can be considered that the zero result of the XOR gate operates as an equality index and, oppositional, the monad as an inequality index. In order to conform with the DET terms (where the zero/not conducting status of the XOR gate refers to an inactive state that cannot be decoded as a current flow in DET's applications) the equality and inequality logical detectors are considered the NOT XOR (XNOR) and the XOR logical gates in correspondence [6-7]. The XOR problem in the simple perceptron refers to the case that n=2. This problem [2] consists of the fact that there is no combination of the (real) weight values (w1, w2) for the input signal function of relation (3) that verifies the XOR Table 1. Moreover, relation (3) denotes that, for every possible pair of weights (w1, w2), the (0, 0), (1, 1) and (0, 1), (1, 0) vector pairs will never exist at opposite half plans (Figure 2), whose boundary is figured by line (3) [2].

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

**Table 1.** The XOR operation truth table

| input -A- | input -B- | output -O- |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

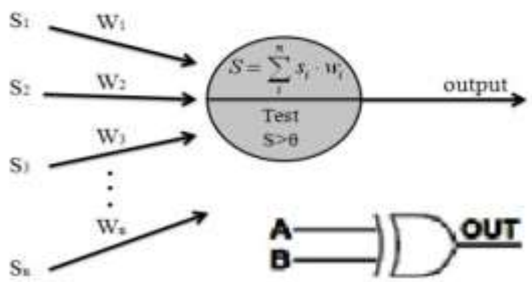



***Fig.1. The simple perceptron and XOR gate circuit structures***

(3) $f(s) = f(s_1, s_2) = s_1 \cdot w_1 + s_2 \cdot w_2 = \theta$

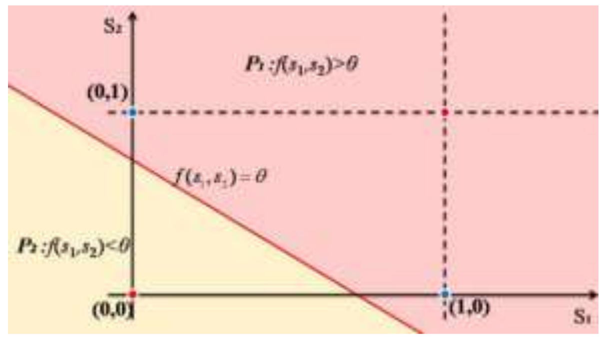
**Solving the XOR problem, by using techniques of the Theory of Numbers**

The XOR problem can be treated with success, in accordance with the complex-number [4] and quantum bit [5] solutions, by applying a threshold function with a modulus operant, whereas n is the dividend, m is the divisor and r is the residual. The n, k, m, r are integers (n, k, m, r Є Z) and the factor m is not zero, m≠0. The Euclidean Algorithm with the modulus operant is expressed [8]:

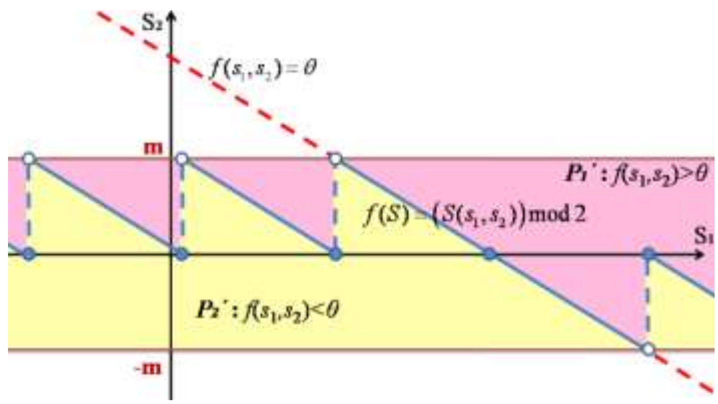
(4) $n = (k \cdot m + r) \bmod m = r \pmod m \Rightarrow n = r \pmod m$

In order to apply the modulus operant in the ANN simple perceptron model, the domain manifolds of n, k, and r parameters are expanded to real number manifolds. Parameter m is excluded from this expansion, since it is correlated with the discrete number of a neuron's inputs (synapses). Hence, it is defined that n, k, r parameters are real numbers and m is a non zero integer, so as relation (5) to be defined, where the divisor enumerates the number of the signal inputs (m=2). In this paper the symbolism ―f(S)=(S)mod2 denotes that the modulus operand is being applied after the S sum calculation and the ―f(S)≡S(mod2) that the parameter S stands for the calculation result after the modulus operant application. Function (5) can transform relation (3)'s half plans into two saw-shaped plan areas (P1΄ and P2΄), as shown at Figure 2. The modulus operant transforms the linear attitude of relation (1) into a periodical linear saw-shaped attitude, so to be possible for non-linear separable cases to be solved by simple perceptron models.

(5) $$S = \left[\sum_{1}^{2} S_i W_I\right] \bmod m \;\text{-->}\; (S(s_1,s_2)) \bmod 2$$




(a)




(b)

***Fig. 2. (a) The line s1w1+s2w2 = θ separates the plan into two half plans. (b) The saw-shaped transformation that the modulus operant causes to relation (3).***

By setting (*s1*, *s2*) = (0,0), (0,1), (1,0) and (1,1) and considering (*w1*, *w2*) = (1,1) a solution to the XOR problem is achieved. Although the XOR problem is not defined for more than 2 inputs, the modulus neuron operant can be generalized to an *n*-input perceptron, by dividing *S* with the inputs number of the neuron. This generalization is hence utilized for inequalities measurement and is displayed at relation (6), where *S* is defined in relation (1), the *modulus operant* is defined in relation (4), *v* denotes the neuron feature as a graph vertex and *deg*+(*v*) [9] declares the number of the input synapses of the neuron. The above generalization opines definitely at the equality case

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

and in analogy the XNOR consideration [6-7] at inequality.

(6) $f(S) = S \bmod(\deg+(v)),\ f(S) \in [0,n)$

The XOR gate architecture in the Digital Electronics Theory has structural and functional similarities to the simple perceptron model in ANN. By considering a threshold function with a modulus operator, as in relation (6), the simple ANN perceptron pattern can solve the non-linear separable XOR problem.

**2. Application of Functional Network to Solving Classification Problems**

Authors systematically discuss a numerical analysis method used for functional network, and apply two functional network models to solving XOR problem. The XOR problem that cannot be solved with two-layered neural network can be solved by two-layered functional network, which reveals a potent computational power of functional networks, and the performance of the proposed model was validated using classification problems [10].

CASTILLO et al. [11] present functional networks as an extension of artificial neural networks (ANNs). Castillo et al. [12] resumes all the functional network models that can be applied. Include time series, chaotic series, differential equations, regression problems, etc. these works show that functional networks can indeed have lots of applications.

***Elements of a Functional Network***

A functional network [11] consists of the following elements (seen Fig.1):

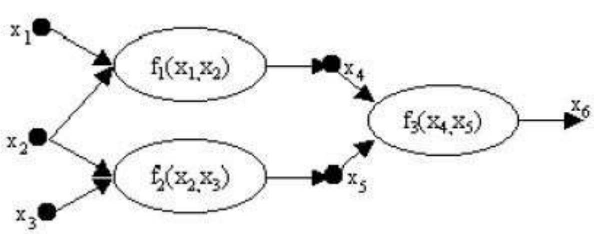



***Fig. 1 A functional network model***

1. A layer of input storing units. This layer contains the input data. Input units are also represented by small black circle with their corresponding names ($x_1$, $x_2$, $x_3$ in Fig.1).
2. A layer of output storing units. This layer contains the output data. Output units are also be represented by small circles with their corresponding names ($x_6$ in Fig.1).
3. One or several layers of processing units. These units evaluate asset of input values, coming from the previous layer (of intermediate or input units) and delivers a set of output values to the next layer (of intermediate or output units). To this end, each neuron has associated a neuron function, which can be multivariate, and can have as many arguments as inputs. Each component of a neural function is called a functional cell. For example, the functional networks in Fig.1 have 3 neurons $f_1$, $f_2$, $f_3$.
4. None, one or several layers of intermediate storing units.
5. These layers contain units store intermediate information produced by neuron units. Intermediate units are represented by small circle ($x_4$, $x_5$ In Fig.1).
6. A set directed links. They connect units in the input or intermediate layers to neuron units, and neuron units to intermediate or output units. Arrows represent connections, indicate the information flow direction.

Our definition is simple but rigorous: a functional network is a network in which the weights of the neurons are substituted by a set of functions. Some have the its advantages include the following issues:
(1) Unlike ANNs, functional networks can reproduce certain physical characteristics that lead to the corresponding network in a natural way. However, reproduction only takes place if an expression is used with a physical meaning inside the functions database, and authors do not dispose of that kind of information, this particular advantage does not apply in our case.
(2) The estimation of the network parameters can be obtained by resolving a linear system of equations. It is a fast and unique solution, and the global minimum of an error function.

Functional networks, as opposed to neural networks, learn the neural the sigmoid functions instead of link weights. In this paper, authors use functional network technique to solving classification problem. The XOR problem that cannot be solved with two-layered neural network can be solved by two-layered functional network, which reveals a potent computational power of functional neural nets. These models are of the properties such as easily trained and simply structured.

**3. Chaotic maps and pattern recognition – the XOR problem**

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Authors demonstrate that the chaotic properties of this map can be used to implement basic operations in Boolean logic. This observation leads naturally to the possibility of new computational models and implementations for conventional computational systems. Here Authors show that by considering the variation of the fractal dimension of its attractor, and using varying parameter values as inputs, the generalised Baker's map can be used as a natural exclusive OR (XOR) gate. We will show how the generalised Baker's map can be used to solve the exclusive OR (XOR) problem. This is a fundamental problem of pattern recognition, and involves telling at a single glance whether a point belongs to one of the two classes: class A or NOT class A (class B), where class A consists of two diagonally opposite corners of a unit square, and class B consists of the other two corners. The inability of a single-layer perception to solve this problemis considered to be a severe drawback for ANNs as a mechanism for nonlinear problem-solving.

The generalised Baker's map is a two-dimensional, three-parameter, nonlinear mapping, which is chaotic for virtually all parameter values. Authors use it here because it is one of the best-understood chaotic maps, and is particularly suited to rigorous analysis (see [13]). It also has the useful property that its Lyapunov dimension is monotonically increasing for a wide range of parameter values, and Authors shall utilise this when Authors develop the XOR gate. To the best of our knowledge, neither Baker's map, nor any other chaotic map, has been previously used to solve the XOR problem in this way.

**Pattern recognition and the XOR problem**

The pattern recognition problem consists of designing algorithms that automatically classify feature vectors associated with specific patterns as belonging to one of a finite number of classes. A benchmark problem in the design of pattern recognition systems is the Boolean exclusive OR (XOR) problem. The standard XOR problem is depicted in Fig. 1. Here the diagonally opposite corner-pairs of the unit square form two classes, A and B (or NOT A). From the figure, it is clear that it is not possible to draw a single straight line which will separate the two classes. This observation is crucial in explaining the inability of a single-layer perceptron to solve this problem (an overview of the perceptron is given in Appendix A).

This problem can be solved using multi-layer perceptrons (MLPs), or by using more elaborate single-layer ANNs such as the radial basis function neural network [13]. However, the inability of simple ANNs, such as the Adeline [14], to solve this problem, effectively ended research interest in the area of ANNs for over 20 years, which highlights the importance of the XOR problem in the design of pattern recognition systems.

**4. Introducing an Inequalities Index from the Simple Perceptron Pattern in ANN: an unemployment inequalities application in Regional Economics**

This article introduces a new inequalities index, which was inspired from the simple perceptron's pattern in the artificial neural networks (ANN) scientific field and nominated by the authors Modulus Perceptron Inequalities Index (MPII). The structural similarities of this ANN pattern with a binary logic gate, in the Digital Electronics Theory (DET), lead to deal with the existence of operational similarities between these two models. The introduced index aroused while examining the potentials of the simple perceptron to solve the non-linear separable ―exclusive disjunction (XOR) problem, by using techniques of the Theory of Numbers in Mathematics. Since the XOR architecture is considered by the DET an inequality detector, due to its ability to result to the same outcome in the same input values, the solution of the XOR problem for a simple perceptron authorizes the ANN and DET XOR models to be considered identical, in the extend that inequalities is regarded. The ANN simple perceptron pattern was proposed in 1962 by Rosenblatt and was the first and simplest one neuron ANN model. In 1969 Minsky and Papert presented the limitations of that model, regarding its inability to solve non-linear separable problems , such as the XOR. Thenceforth, the research about two level neuron patterns was abandoned . The next period, multi-layer ANN neuron patterns were introduced to overcome simple perceptron's limitations. The utility of the model was rearranged in modern research and some solutions of the XOR problem were proposed, by using computational techniques from Complex Number Analysis in Mathematics and from Quantum Computation. The XOR model's architecture is considered from DET an inequality detector, due to its ability to result the same numerical outcome when the input data consists of the same values. Solving the XOR problem for a simple perceptron render this ANN model to operate as an electronic XOR gate and to perform similarly, regarding the inequalities detection. These identical XOR patterns (simple perceptron and DET's logical gate) may also provide utility in Regional Economics inequalities research, a scientific sector that enumerates a considerable number of measures, such as the Theil index. At the following a solution of the XOR problem for the ANN simple perceptron is given, by using calculation techniques from the Theory of Numbers in

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Mathematics. Secondly, the proposed simple perceptron pattern is being formulated to an inequalities index (MPII) and it is compared to the Theil index, regarding their inequalities performance [16].

**5. A novel spiking perceptron that can solve XOR problem**

*A* novel spiking perceptron can solve XOR problem, while a classical spiking neuron usually needs a hidden layer to solve XOR problem.

Formally, a spiking perceptron receives a set of spikes with firing times and fires if the state variable crosses a threshold. In the existing literature, the output of the spiking perceptron is usually taken as the firing time, and the target of training is to learn a set of target firing times for a set of training samples. XOR classification problem is a well-known benchmark problem for testing the capacity of a certain kind of neural network. It has been shown that a hidden layer is usually included in the spiking perceptron for solving XOR problem. Authors recall that a classical feedforward perceptron without hidden layer, of which the output is the value of a Sigmoid function at the inner product of the weight vector and the input vector, can not solve XOR problem either. However, Authors note that a spiking perceptron obtains its output with much more effort than a classical feedforward perceptron, in that for a given input (a set of spikes), the spik -ing perceptron has to go through a sequence of temporary outputs by increasing the time parameter until the state variable crosses a given threshold. Therefore, it seems reasonable to expect the spiking perceptron to behave better than the classical feedforward perceptron.

A novel spiking neuron is proposed in this paper to give a positive answer to the above question. Authors introduce a new parameter for the output, i.e., the spiking intensity (the gradient of the state function) at the firing time. Now, the output of the neuron is not the ring time alone, but a linear combination of the firing time and the spike intensity. Authors show by numerical experiments that this new spiking perceptron is capable of solving XOR problem, and Authors believe that this novel spiking perceptron can be used as a building block to build up more efficient spiking neural networks [17].

## CONCLUSION

This paper tries to explain the network structures and methods for XOR problem. XOR is linear un-division operation, which cannot be treated by single-layer perceptron. With the analysis, several solutions are proposed in the paper to solve the problems of XOR. Basic outline of problem and Various solutions are discussed to understand the concerned problem.

## REFERENCES

1. D. Tsiotas and S. Polyzos, "Introducing an Inequalities Index from the Simple Perceptron Pattern in ANN: an unemployment inequalities application in Regional Economics", in : Journal of Engineering Science and Technology Review 4 (3) (291-296), 2011.
2. P. Argyrakis, "Neural Networks and Applications", Patra, Greece, Hellenic Open University Publications, 2001.
3. A. Rogers, J. Keating, R. Shorten, D. Heffernan, "Chaotic maps and pattern recognition – the XOR problem", Chaos, Solitons and Fractals, 14, 57–70, 2002.
4. T. Nitta, "Solving the XOR problem and the detection of symmetry using a single complex-valued neuron", Neural Networks, 16, 1101, 2003.
5. M. Maeda, "Qubit neuron according to quantum circuit for XOR problem", Applied Mathematics and Computation, 185, 1015, 2007.
6. B. Paton, "Fundamentals of Digital Electronics", Austin, National Instruments Corporation 1998.
7. J. Crowe, B. Gill, "Introduction to Digital Electronics", Oxford, Newnes Publications 2001.
8. T. Apostol, "Introduction to Analytic Number Theory", New York, Springer-Verlag Publications 1976.
9. R. Diestel, Graph Theory, Third Edition, Heidelberg, Springer-Verlag Publications 2005.
10. Yong-Quan Zhou, Deng-Xu He, and Zheng Nong, "Application of Functional Network to Solving Classification Problems", in Proceedings of World Academy Science, Engineering and technology, 2005.
11. Enrique Castillo, "Functional Networks". Neural Processing Letters, 1998.
12. Enrique Castillo, Angel Cobo, Jose Manuel Gutierrez, "Functional Networks with Applications". Kluwer Academic Publishers, 1999.
13. Bishop CM, "Neural networks for pattern recognition". Oxford: Clarendon Press; 1995.
14. Haykin S., "Neural networks – a comprehensive foundation". Englewood Cliffs, NJ: Prentice-Hall, 1999.
15. Farmer JD, Ott E, Yorke JA, "The dimension of chaotic attractors". Physica D, 7:153–80 1983.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

16. Rumelhart, D.E., Hinton, G.E., & Williams, R.J. Parallel distributed processing (Vol.1). Cambridge, MA: MIT Press, 1986a.
17. James Tin-Yau. "Moderating the outputs of support vector machine classifiers", in : IEEE Transactions On Neural Networks, 10(5): 1018-1031, 1999.