International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 3 Issue 2, February – 2019, Pages: 9-17

ANN for Tic-Tac-Toe Learning

Mohaned Abu Dalffa

Department of Information Technology,
Faculty of Engineering and Information Technology,
Al-Azhar University - Gaza, Palestine

Abstract: *Throughout this research, imposing the training of an Artificial Neural Network (ANN) to play tic-tac-toe bored game, by training the ANN to play the tic-tac-toe logic using the set of mathematical combination of the sequences that could be played by the system and using both the Gradient Descent Algorithm explicitly and the Elimination theory rules implicitly. And so on the system should be able to produce imunate amalgamations to solve every state within the game course to make better of results of winnings or getting draw.*

Keywords: Tic-Tac-Toe, neural network, ANN, prediction.

1. Introduction

Warren McCulloch and Walter Pitts [1] created a computational model for neural networks based on mathematics and algorithms called threshold logic. This model paved the way for neural network research to split into two approaches. One approach focused on biological processes in the brain while the other focused on the application of neural networks to artificial intelligence. This work led to work on nerve networks and their link to finite automata [2].
Artificial Neural Networks are computing algorithms that can solve complex problems imitating animal brain processes in a simplified manner [3].

Perceptron-type neural networks consist of artificial neurons or nodes, which are information processing units arranged in layers and interconnected by synaptic weights (connections). Neurons can filter and transmit information in a supervised fashion in order to build a predictive model that classifies data stored in memory.
A typical ANN model is a three-layered network of interconnected nodes: the input layer, the hidden layer, and the output layer.

The nodes between input and output layers can form one or more hidden layers. Every neuron in one layer has a link to every other neuron in the next layer, but neurons belonging to the same layer have no connections between them (Figure 1). The input layer receives information from the outside world, the hidden layer performs the information processing and the output layer produces the class label or predicts continuous values. The values from the input layer entering a hidden node are multiplied by weights, a set of predetermined numbers, and the products are then added to produce a single number. This number is passed as an argument to a nonlinear mathematical function, the activation function, which returns a number between 0 and 1[4].

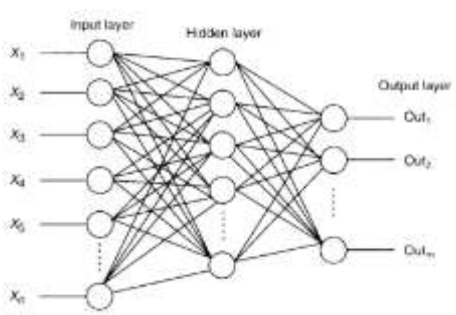


Figure 1. Neural network architecture.

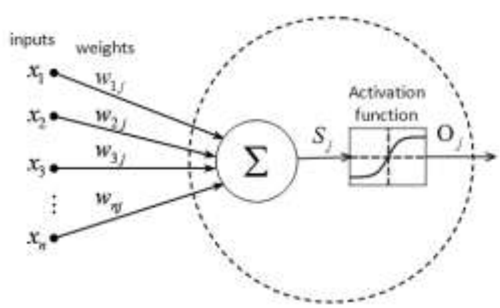


Figure 2. Neural network active node.

In Fig.2, the net sum of the weighted inputs entering a node j and the output activation function that converts a neuron's weighted input to its output activation (the most commonly used is the sigmoid function), are given by the equations respectively.

$$S_j = \sum_{i=1}^{n} x_i w_{ij} \quad \text{and} \quad O_j = \frac{1}{1+e^{S_j}}$$

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 3 Issue 2, February – 2019, Pages: 9-17

The neuron, and therefore the ANN, has two modes of operation, the training mode and the using mode. During the training phase, a data set with actual inputs and outputs will be used as examples to teach the system how to predict outputs. This supervised learning begins with random weights and, by using gradient descent search algorithms like Backpropagation, adjusts the weights to be applied to the task at hand. The difference between target output values and obtained values is used in the error function to drive learning [12]. The error function depends on the weights, which need to be modified in order to minimize the error. For a given training set $\{(x_1,t_1),(x_2,t_2),\cdots,(x_k,t_k)\}$ consisting of k ordered pairs of n inputs and m dimensional vectors (n-inputs, m-outputs), which are called the input and output patterns, the error for the output of each neuron can be defined by the equation:

$$E_j = \tfrac{1}{2}(O_j - t_j)^2$$

while the error function of the network that must be minimized is given by:

$$E_j = \tfrac{1}{2}\sum_{j=1}^{k}(O_j - t_j)^2$$

Where Oj is the output produced when the input pattern x j from the training set enters the network, and t j is the target value [13]. During the training mode, each weight is changed adding to its previous value the quantity

$$\Delta w_{ij} = -\gamma \frac{\partial E}{\partial w_{ij}}$$

Where γ is a constant that gives the learning rate. The higher the learning rate, the faster the convergent will be, but the searching path may trapped around the optimal solution and convergence become impossible. Once a set of good weights have been found, the neural network model can take another dataset with unknown output values and predict automatic the corresponding outputs.

2. TIC-TAC-TOE THEORY

Tic-tac-toe is played on a three-by-three grid (see figure 3). Each player takes turn to place a symbol on an open square. One player's symbol is "X" and the other's is "O". The game is over once a player has three signs in a row: horizontally, vertically, or diagonally (as shown in figure 4). The game can end with a draw result (as shown in figure 5), if there is no possibility of winning for both players.

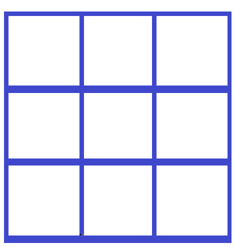


Figure 3: Empty board

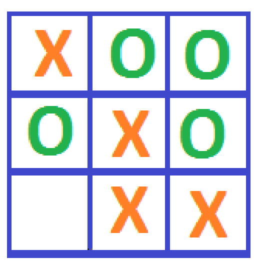


Figure 4: Palyer X win

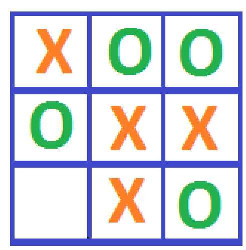


Figure 5: Draw game

The Tic-Tac-Toe game can be generalized to an (m, n, k) game in which two players take turns to put a symbol of their own color on an m × n board, with the goal of getting k of their own color in a row. Tic-tac-toe is specifically (3, 3, 3) game, where m =3, n=3 and k= 3 in this game [6]. If played correctly, the game will end in a draw, making tic-tac-toe a pointless game [8].

3. FUN HISTORICAL FACTS

Games played on three-in-a-row boards can be traced back to ancient Egypt [9], where such game boards have been found on roofing tiles dating from around 1300 BCE[10].

An early variation of tic-tac-toe was played in the Roman Empire, around the first century BC. It was called terni lapilli (three pebbles at a time) and instead of having any number of pieces, each player only had three, thus they had to move them around to empty spaces to keep playing [11]. The game's grid markings have been found chalked all over Rome. Another closely related

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 3 Issue 2, February – 2019, Pages: 9-17

ancient game is Three Men's Morris which is also played on a simple grid and requires three pieces in a row to finish,[12] and Picaria, a game of the Puebloans.

The different names of the game are more recent. The first print reference to "noughts and crosses", the British name, appeared in 1858, in an issue of Notes and Queries[13]. The first print reference to a game called "tick-tack-toe" occurred in 1884, but referred to "a children's game played on a slate, consisting in trying with the eyes shut to bring the pencil down on one of the numbers of a set, the number hit being scored". "Tic-tac-toe" may also derive from "tick-tack", the name of an old version of backgammon first described in 1558. The US renaming of "noughts and crosses" as "tic-tac-toe" occurred in the 20th century[14].

In 1952, OXO (or Noughts and Crosses), developed by British computer scientist Alexander S. Douglas for the EDSAC computer at the University of Cambridge, became one of the first known video games[15,16]. The computer player could play perfect games of tic-tac-toe against a human opponent.11

In 1975, tic-tac-toe was also used by MIT students to demonstrate the computational power of Tinkertoy elements. The Tinkertoy computer, made out of (almost) only Tinkertoys, is able to play tic-tac-toe perfectly [17]. It is currently on display at the Museum of Science, Boston.

4. Combinatory

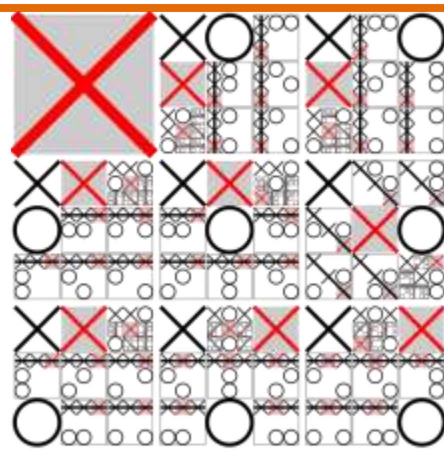
When considering only the state of the board, and after taking into account board symmetries (i.e. rotations and reflections), there are only 138 terminal board positions. A combinatorics study of the game shows that when "X" makes the first move every time, the game is won as follows[18]:

- 91 distinct positions are won by (X)
- 44 distinct positions are won by (O)
- 3 distinct positions are drawn (often called a "cat's game[19]")

5. Strategy

A player can play a perfect game of tic-tac-toe (to win or at least, draw) if each time it is his turn to play, he chooses the first available move from the following list, as used in Newell and Simon's 1972 tic-tac-toe program[20].

1- **Win**: If the player has two in a row, they can place a third to get three in a row.
2- **Block**: If the opponent has two in a row, the player must play the third themselves to block the opponent.
3- **Fork**: Create an opportunity where the player has two threats to win (two non-blocked lines of 2).
4- **Blocking an opponent's fork**: If there is only one possible fork for the opponent, the player should block it. Otherwise, the player should block any forks in any way that simultaneously allows them to create two in a row. Otherwise, the player should create a two in a row to force the opponent into defending, as long as it doesn't result in them creating a fork. For example, if "X" has two opposite corners and "O" has the center, "O" must not play a corner in order to win. (Playing a corner in this scenario creates a fork for "X" to win.)
5- **Center**: A player marks the center. (If it is the first move of the game, playing on a corner gives the second player more opportunities to make a mistake and may therefore be the better choice; however, it makes no difference between perfect players.)
6- **Opposite corner**: If the opponent is in the corner, the player plays the opposite corner.
7- **Empty corner**: The player plays in a corner square.
8- **Empty side**: The player plays in a middle square on any of the 4 sides.

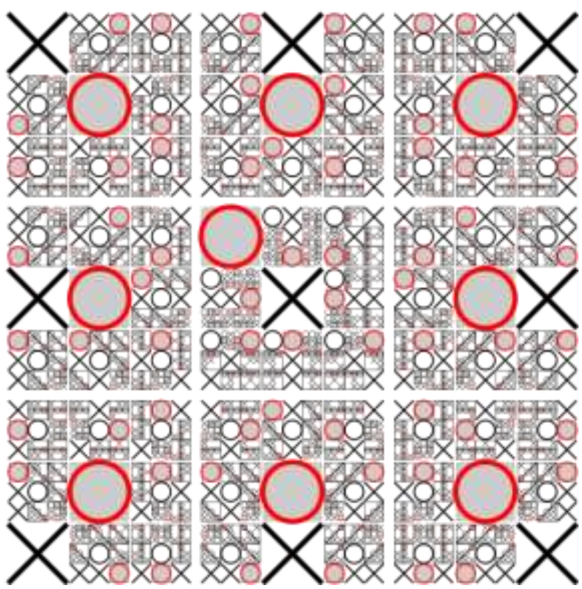


Optimal strategy for player X if starting in a corner. In each grid, the shaded red X denotes the optimal move, and the location of O's next move gives the next subgrid to examine. Note that only two sequences of moves by O (both starting with center, top-right, left-mid) lead to a draw, with the remaining sequences leading to wins from X.

The first player, who shall be designated "X", has 3 possible positions to mark during the first turn. Superficially, it might seem that there are 9 possible positions, corresponding to the 9 squares in the grid. However, by rotating the board, we will find that in the first turn, every corner mark is strategically equivalent to every other corner mark. The same is true of every edge (side middle) mark. For strategy purposes, there are therefore only three possible first marks: corner, edge, or center. Player X can win or force a draw from any of these starting marks; however, playing the corner gives the opponent the smallest choice of squares which must be played to avoid losing[21]. This might suggest that the corner is the best opening move for X, however another study [22] shows that if the players are not perfect, an opening move in the center is best for X.

The second player, who shall be designated "O", must respond to X's opening mark in such a way as to avoid the forced win. Player O must always respond to a corner opening with a center mark, and to a center opening with a corner mark. An edge opening must be answered either with a center mark, a corner mark next to the X, or an edge mark opposite the X. Any other responses will allow X to force the win. Once the opening is completed, O's task is to follow the above list of priorities in order to force the draw, or else to gain a win if X makes a weak play.

More detailedly, to guarantee a draw, O should adopt the following strategies:



Optimal strategy for player O. Player O can always force a win or draw by taking center. If it is taken by X, then O must take a corner

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 3 Issue 2, February – 2019, Pages: 9-17

If X plays corner opening move, O should take center, and then an edge, forcing X to block in the next move. This will stop any forks from happening. When both X and O are perfect players and X chooses to start by marking a corner, O takes the center, and X takes the corner opposite the original. In that case,

O is free to choose any edge as its second move. However, if X is not a perfect player and has played a corner and then an edge, O should not play the opposite edge as its second move, because then X is not forced to block in the next move and can fork.

If X plays edge opening move, O should take center or one of the corners adjacent to X, and then follow the above list of priorities, mainly paying attention to block forks.

If X plays center opening move, O should take corner, and then follow the above list of priorities, mainly paying attention to block forks.

When X plays corner first, and O is not a perfect player, the following may happen:

If O responds with a center mark (best move for them), a perfect X player will take the corner opposite the original. Then O should play an edge. However, if O plays a corner as its second move, a perfect X player will mark the remaining corner, blocking O's 3-in-a-row and making their own fork.

If O responds with a corner mark, X is guaranteed to win, by simply taking any of the other two corners and then the last, a fork. (since when X takes the third corner, O can only take the position between the two X's. Then X can take the only remaining corner to win).

Further details

Consider a board with the nine positions numbered as follows:

1	2	3
4	5	6
7	8	9

When X plays 1 as their opening move, then O should take 5.
Then X takes 9 (in this situation, O should not take 3 or 7, O should take 2, 4, 6 or 8):
X1 → O5 → X9 → O2 → X8 → O7 → X3 → O6 → X4, this game will be a draw.
or 6 (in this situation, O should not take 4 or 7, O should take 2, 3, 8 or 9. In fact, taking 9 is the best move, since a non-perfect player X may take 4, then O can take 7 to win).

- X1 → O5 → X6 → O2 → X8, then O should not take 3, or X can take 7 to win, and O should not take 4, or X can take 9 to win, O should take 7 or 9.
- X1 → O5 → X6 → O2 → X8 → O7 → X3 → O9 → X4, this game will be a draw.
- X1 → O5 → X6 → O2 → X8 → O9 → X4 (7) → O7 (4) → X3, this game will be a draw.
- X1 → O5 → X6 → O3 → X7 → O4 → X8 (9) → O9 (8) → X2, this game will be a draw.
- X1 → O5 → X6 → O8 → X2 → O3 → X7 → O4 → X9, this game will be a draw.
- X1 → O5 → X6 → O9, then X should not take 4, or O can take 7 to win, X should take 2, 3, 7 or 8.
- X1 → O5 → X6 → O9 → X2 → O3 → X7 → O4 → X8, this game will be a draw.
- X1 → O5 → X6 → O9 → X3 → O2 → X8 → O4 (7) → X7 (4), this game will be a draw.
- X1 → O5 → X6 → O9 → X7 → O4 → X2 (3) → O3 (2) → X8, this game will be a draw.
- X1 → O5 → X6 → O9 → X8 → O2 (3, 4, 7) → X4/7 (4/7, 2/3, 2/3) → O7/4 (7/4, 3/2, 3/2) → X3 (2, 7, 4), this game will be a draw.

If X is not a perfect player, X may take 2 or 3 as second move. Then this game will be a draw, X cannot win.

- X1 → O5 → X2 → O3 → X7 → O4 → X6 → O8 (9) → X9 (8), this game will be a draw.
- X1 → O5 → X3 → O2 → X8 → O4 (6) → X6 (4) → O9 (7) → X7 (9), this game will be a draw.

If X plays 1 opening move, and O is not a perfect player, the following may happen:

Although O takes the only good position (5) as first move, but O takes a bad position as second move:

- X1 → O5 → X9 → O3 → X7, then X can take 4 or 8 to win.
- X1 → O5 → X6 → O4 → X3, then X can take 2 or 9 to win.
- X1 → O5 → X6 → O7 → X3, then X can take 2 or 9 to win.

Although O takes good positions as the first two moves, but O takes a bad position as third move:

- X1 → O5 → X6 → O2 → X8 → O3 → X7, then X can take 4 or 9 to win.
- X1 → O5 → X6 → O2 → X8 → O4 → X9, then X can take 3 or 7 to win.

O takes a bad position as first move (except of 5, all other positions are bad):

- X1 → O3 → X7 → O4 → X9, then X can take 5 or 8 to win.
- X1 → O9 → X3 → O2 → X7, then X can take 4 or 5 to win.
- X1 → O2 → X5 → O9 → X7, then X can take 3 or 4 to win.
- X1 → O6 → X5 → O9 → X3, then X can take 2 or 7 to win

6. THE METHODOLOGY

The EasyNN-Plus program was used to develop the ANN model. The structure of the neural network was set as feed-forward, in which the output layer connects only to the previous layer. ANN training used 83% of the 958 cases (798) and 17% of cases (160) were selected for the validation set.

All cases were randomly selected by the EasyNN-Plus program. The parameters considered the input layer for the neural network training can be seen in the following adjacent figures, combined with other parameters describing the combinations of the tic-tac-toe game regression. A total of nine input parameters were used in the development of the ANN model (see Appendix). Several configurations of ANN were tested in other to find the best performing combination of number of hidden layers and nodes per layer. In all configurations, Eq. was used as activation function to smooth the output signal of each node:

$$s(x) = 1/(1 + e^{-x})$$

Where x is the sum of the weighted input of each previous node plus the bias of the node itself.

ANN results were evaluated based on the coefficient of determination of all strategies possible to the game mind, the mean bias error.

The accuracy of these ANNs was then compared to the accuracy of the original ANN (which includes all eight input parameters).

Technique and Description

Gradient Descent Formula was used in terms of measuring and evaluating each suitable move would and could be used:

Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent.

$$\begin{aligned}
&\text{repeat until convergence: } \{\\
&\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_0^{(i)}\\
&\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_1^{(i)}\\
&\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_2^{(i)}\\
&\cdots\\
&\}
\end{aligned}$$

Data Used in tic-tac-toe

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 3 Issue 2, February – 2019, Pages: 9-17

Table 1: Input and output attributes

SN.	Attribute Name	Type of Attribute
1	top-left-square: {x,o,b}	Input
2	top-middle-square: {x,o,b}	Input
3	top-right-square: {x,o,b}	Input
4	middle-left-square: {x,o,b}	Input
5	middle-middle-square: {x,o,b}	Input
6	middle-right-square: {x,o,b}	Input
7	bottom-left-square: {x,o,b}	Input
8	bottom-middle-square: {x,o,b}	Input
9	bottom-right-square: {x,o,b}	Input
10	Class: {positive, negative}	Output

The original dataset was normalized to be ready for Just NN environment. Dataset consists of 958 samples. It was divided into 83% of the total sample for training and 17% for the total samples into validation. That means the size of the training sample is equal 498 and the size of the validation samples is 160.

The ANN model consist of three layers: one layer for input (9 neuron), one Hidden Layer (3 neuron) and one output layer (one neuron) (as shown in Figure 6).

The ANN model was trained and validated. The number of cycles was 3580 and the error rate was equal 0.018237 and the accuracy was 99.38% (as shown Figure 7). The relative importance of the attribute was found as in Figure 8.

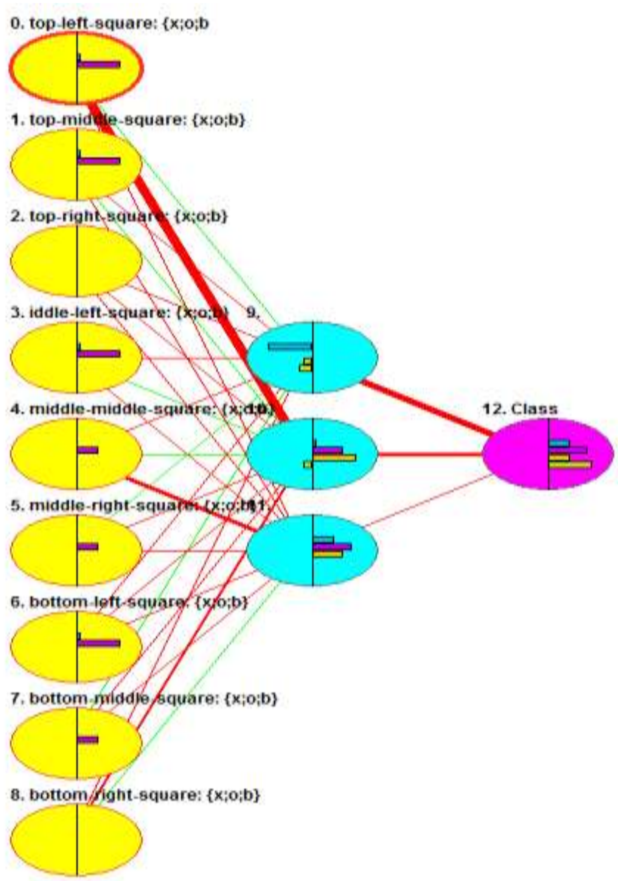


Figure 6: Neural Network Design

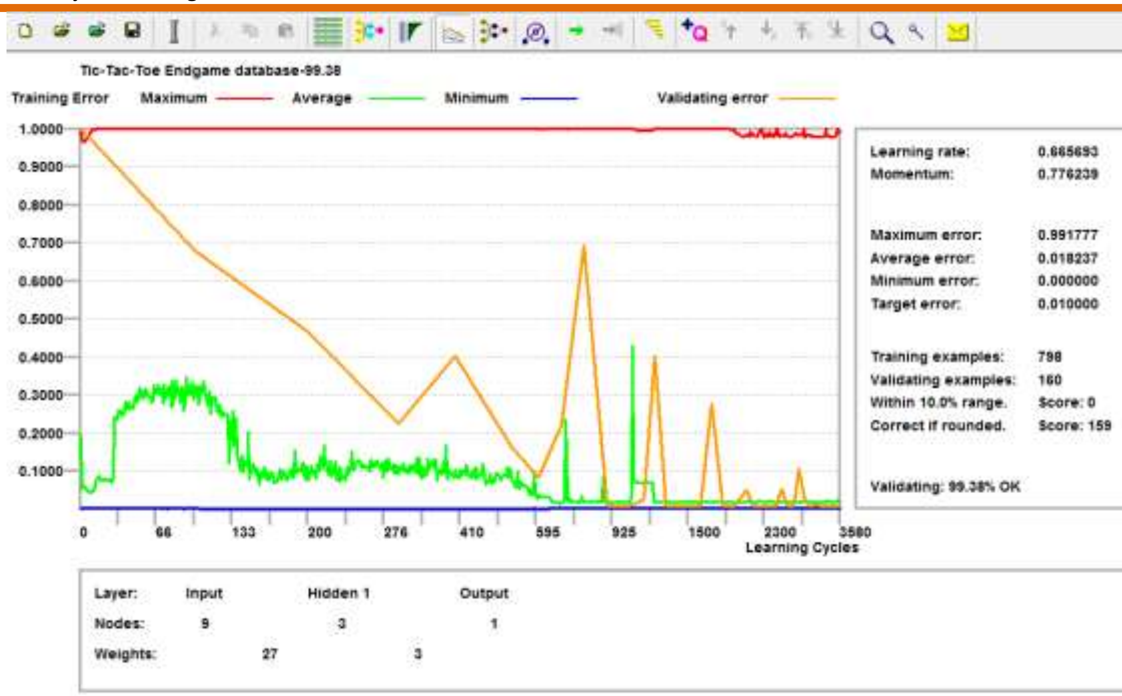


Figure 7: Neural Network model training and validation

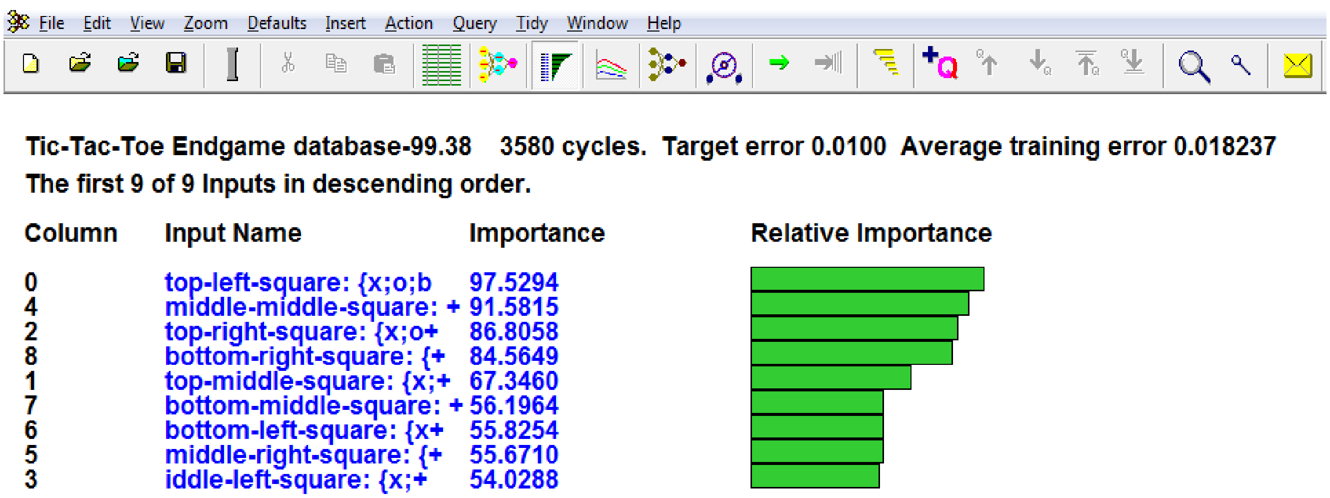


Figure 8: Neural Network relative importance of the attributes

7. CONCLUSION

In this research, we presented the training of an Artificial Neural Network (ANN) to play tic-tac-toe bored game, by training the ANN to play the tic-tac-toe logic using the set of mathematical combination of the sequences that could be played by the system and using both the Gradient Descent Algorithm explicitly and the Elimination theory rules implicitly. The accuracy of the ANN model we got was 99.38%.

REFERENCES

1. Abu-Naser, S. S. (2012). "Predicting learners performance using artificial neural networks in linear programming intelligent tutoring system." International Journal of Artificial Intelligence & Applications 3(2): 65.
2. Abu-Nasser, B. S. and S. S. Abu Naser (2018). "Rule-Based System for Watermelon Diseases and Treatment." International Journal of Academic Information Systems Research (IJAISR) 2(7): 1-7.
3. Abu-Nasser, B. S. and S. S. Abu-Naser (2018). "Cognitive System for Helping Farmers in Diagnosing Watermelon Diseases." International Journal of Academic Information Systems Research (IJAISR) 2(7): 1-7.
4. Abu-Saqer, M. M. and S. S. Abu-Naser (2019). "Developing an Expert System for Papaya Plant Disease Diagnosis." International Journal of Academic Engineering Research (IJAER) 3(4): 14-21.
5. Afana, M., et al. (2018). "Artificial Neural Network for Forecasting Car Mileage per Gallon in the City." International Journal of Advanced Science and Technology 124: 51-59.
6. Alajrami, E., et al. (2019). "Blood Donation Prediction using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 3(10): 1-7.
7. Alajrami, E., et al. (2020). "Handwritten Signature Verification using Deep Learning." International Journal of Academic Multidisciplinary Research (IJAMR) 3(12): 39-44.
8. Alajrami, M. A. and S. S. Abu-Naser (2020). "Type of Tomato Classification Using Deep Learning." International Journal of Academic Pedagogical Research (IJAPR) 3(12): 21-25.
9. Al-Daour, A. F., et al. (2020). "Banana Classification Using Deep Learning." International Journal of Academic Information Systems Research (IJAISR) 3(12): 6-11.
10. Alghoul, A., et al. (2018). "Email Classification Using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 2(11): 8-14.
11. Alkronz, E. S., et al. (2019). "Prediction of Whether Mushroom is Edible or Poisonous Using Back-propagation Neural Network." International Journal of Academic and Applied Research (IJAAR) 3(2): 1-8.
12. Al-Massri, R., et al. (2018). "Classification Prediction of SBRCTs Cancers Using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 2(11): 1-7.
13. Al-Mubayyed, O. M., et al. (2019). "Predicting Overall Car Performance Using Artificial Neural Network." International Journal of Academic and Applied Research (IJAAR) 3(1): 1-5.
14. Alqumboz, M. N. A. and S. S. Abu-Naser (2020). "Avocado Classification Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 3(12): 30-34.
15. Al-Shawwa, M. and S. S. Abu-Naser (2019). "Predicting Birth Weight Using Artificial Neural Network." International Journal of Academic Health and Medical Research (IJAHMR) 3(1): 9-14.
16. Al-Shawwa, M. and S. S. Abu-Naser (2019). "Predicting Effect of Oxygen Consumption of Thylakoid Membranes (Chloroplasts) from Spinach after Inhibition Using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 3(2): 15-20.
17. Al-Shawwa, M. O. and S. S. Abu-Naser (2020). "Classification of Apple Fruits by Deep Learning." International Journal of Academic Engineering Research (IJAER) 3(12): 1-7.
18. Alshawwa, I. A., et al. (2020). "Analyzing Types of Cherry Using Deep Learning. " International Journal of Academic Engineering Research (IJAER) 4 (1): 1-5.
19. Abu-Saqer, M. M., et al. (2020). "Type of Grapefruit Classification Using Deep Learning. " International Journal of Academic Information Systems Research (IJAISR) 4 (1): 1-5.
20. Al-Shawwa, M., et al. (2018). "Predicting Temperature and Humidity in the Surrounding Environment Using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 2(9): 1-6.
21. AlZamily, J. Y. and S. S. A. Naser (2020). "Lemon Classification Using Deep Learning." International Journal of Academic Pedagogical Research (IJAPR) 3(12): 16-20.
22. Ashqar, B. A. M. and S. S. Abu-Naser (2019). "Identifying Images of Invasive Hydrangea Using Pre-Trained Deep Convolutional Neural Networks." International Journal of Academic Engineering Research (IJAER) 3(3): 28-36.
23. Ashqar, B. A. M. and S. S. Abu-Naser (2019). "Image-Based Tomato Leaves Diseases Detection Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 2(12): 10-16.
24. Ashqar, B. A., et al. (2019). "Plant Seedlings Classification Using Deep Learning." International Journal of Academic Information Systems Research (IJAISR) 3(1): 7-14.
25. Barhoom, A. M., et al. (2019). "Predicting Titanic Survivors using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 3(9): 8-12.
26. Dalffa, M. A., et al. (2019). "Tic-Tac-Toe Learning Using Artificial Neural Networks." International Journal of Engineering and Information Systems (IJEAIS) 3(2): 9-19.
27. Dheir, I. M., et al. (2020). "Classifying Nuts Types Using Convolutional Neural Network." International Journal of Academic Information Systems Research (IJAISR) 3(12): 12-18.
28. El_Jerjawi, N. S. and S. S. Abu-Naser (2018). "Diabetes Prediction Using Artificial Neural Network." International Journal of Advanced Science and Technology 121: 55-64.
29. El-Kahlout, M. I. and S. S. Abu-Naser (2020). "Peach Type Classification Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 3(12): 35-40.
30. El-Khatib, M. J., et al. (2019). "Glass Classification Using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 3(2): 25-31.
31. El-Mashharawi, H. Q., et al. (2020). "Grape Type Classification Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 3(12): 41-45.
32. Elsharif, A. A., et al. (2020). "Potato Classification Using Deep Learning." International Journal of Academic Pedagogical Research (IJAPR) 3(12): 1-8.
33. Elzamly, A., et al. (2017). "Predicting Critical Cloud Computing Security Issues using Artificial Neural Network (ANNs) Algorithms in Banking Organizations." International Journal of Information Technology and Electrical Engineering 6(2): 40-45.
34. Heriz, H. H., et al. (2018). "English Alphabet Prediction Using Artificial Neural Networks." International Journal of Academic Pedagogical Research (IJAPR) 2(11): 8-14.
35. Jamala, M. N. and S. S. Abu-Naser (2018). "Predicting MPG for Automobile Using Artificial Neural Network Analysis." International Journal of Academic Information Systems Research (IJAISR) 2(10): 5-21.
36. Kashf, D. W. A., et al. (2018). "Predicting DNA Lung Cancer using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 2(10): 6-13.
37. Khalil, A. J., et al. (2019). "Energy Efficiency Predicting using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 3(9): 1-8.
38. Marouf, A. and S. S. Abu-Naser (2018). "Predicting Antibiotic Susceptibility Using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 2(10): 1-5.
39. Mettleq, A. S. A., et al. (2020). "Mango Classification Using Deep Learning." International Journal of Academic Engineering Research (IJAER) 3(12): 22-29.
40. Metwally, N. F., et al. (2018). "Diagnosis of Hepatitis Virus Using Artificial Neural Network." International Journal of Academic Pedagogical Research (IJAPR) 2(11): 1-7.
41. Musleh, M. M., et al. (2019). "Predicting Liver Patients using Artificial Neural Network." International Journal of Academic Information Systems Research (IJAISR) 3(10): 1-11.
42. Nabahin, A., et al. (2017). "Expert System for Hair Loss Diagnosis and Treatment." International Journal of Engineering and Information Systems (IJEAIS) 1(4): 160-169.
43. Nasser, I. M. and S. S. Abu-Naser (2019). "Artificial Neural Network for Predicting Animals Category." International Journal of Academic and Applied Research (IJAAR) 3(2): 18-24.
44. Nasser, I. M. and S. S. Abu-Naser (2019). "Lung Cancer Detection Using Artificial Neural Network." International Journal of Engineering and Information Systems (IJEAIS) 3(3): 17-23.
45. Nasser, I. M. and S. S. Abu-Naser (2019). "Predicting Books' Overall Rating Using Artificial Neural Network." International Journal of Academic Engineering Research (IJAER) 3(8): 11-17.
46. Nasser, I. M. and S. S. Abu-Naser (2019). "Predicting Tumor Category Using Artificial Neural Networks." International Journal of Academic Health and Medical Research (IJAHMR) 3(2): 1-7.
47. Nasser, I. M., et al. (2019). "A Proposed Artificial Neural Network for Predicting Movies Rates Category." International Journal of Academic Engineering Research (IJAER) 3(2): 21-25.
48. Nasser, I. M., et al. (2019). "Artificial Neural Network for Diagnose Autism Spectrum Disorder." International Journal of Academic Information Systems Research (IJAISR) 3(2): 27-32.
49. Nasser, I. M., et al. (2019). "Developing Artificial Neural Network for Predicting Mobile Phone Price Range." International Journal of Academic Information Systems Research (IJAISR) 3(2): 1-6.
50. Sadek, R. M., et al. (2019). "Parkinson's Disease Prediction Using Artificial Neural Network." International Journal of Academic Health and Medical Research (IJAHMR) 3(1): 1-8.
51. Salah, M., et al. (2018). "Predicting Medical Expenses Using Artificial Neural Network." International Journal of Engineering and Information Systems (IJEAIS) 2(20): 11-17.
52. Samy, M. and A. Naser (2012). "Predicting learners performance using artificial neural networks in linear programming intelligent tutoring systems." International journal of artificial intelligence & aplpications 3.
53. Zaqout, I., et al. (2015). "Predicting Student Performance Using Artificial Neural Network: in the Faculty of Engineering and Information Technology." International Journal of Hybrid Information Technology 8(2): 221-228.
54. Abu Naser, S. S. (2018). "TOP 10 NEURAL NETWORK PAPERS: RECOMMENDED READING–ARTIFICIAL INTELLIGENCE RESEARCH." word press 1(1).
55. Almadhoun, H. R. and S. S. Abu Naser (2018). "Banana Knowledge Based System Diagnosis and Treatment." International Journal of Academic Pedagogical Research (IJAPR) 2(7): 1-11.
56. Salman, F. and S. S. Abu-Naser (2019). "Rule based System for Safflower Disease Diagnosis and Treatment." International Journal of Academic Engineering Research (IJAER) 3(8): 1-10.
57. Salman, F. M. and S. S. Abu-Naser (2019). "Expert System for Castor Diseases and Diagnosis." International Journal of Engineering and Information Systems (IJEAIS) 3(3): 1-10.
58. Nassr, M. S. and S. S. Abu Naser (2018). "Knowledge Based System for Diagnosing Pineapple Diseases." International Journal of Academic Pedagogical Research (IJAPR) 2(7): 12-19.
59. Mettleq, A. S. A. and S. S. Abu-Naser (2019). "A Rule Based System for the Diagnosis of Coffee Diseases." International Journal of Academic Information Systems Research (IJAISR) 3(3): 1-8.
60. Musleh, M. M. and S. S. Abu-Naser (2018). "Rule Based System for Diagnosing and Treating Potatoes Problems." International Journal of Academic Engineering Research (IJAER) 2(8): 1-9.
61. Khalil, A. J., et al. (2019). "Apple Trees Knowledge Based System." International Journal of Academic Engineering Research (IJAER) 3(9): 1-7.
62. Elzamly, A., et al. (2015). "Classification of Software Risks with Discriminant Analysis Techniques in Software planning Development Process." International Journal of Advanced Science and Technology 81: 35-48.
63. Elzamly, A., et al. (2015). "Predicting Software Analysis Process Risks Using Linear Stepwise Discriminant Analysis: Statistical Methods." Int. J. Adv. Inf. Sci. Technol 38(38): 108-115.
64. Elqassas, R. and S. S. Abu-Naser (2018). "Expert System for the Diagnosis of Mango Diseases." International Journal of Academic Engineering Research (IJAER) 2(8): 10-18.
65. Elsharif, A. A. and S. S. Abu-Naser (2019). "An Expert System for Diagnosing Sugarcane Diseases." International Journal of Academic Engineering Research (IJAER) 3(3): 19-27.
66. El-Mashharawi, H. Q. and S. S. Abu-Naser (2019). "An Expert System for Sesame Diseases Diagnosis Using CLIPS." International Journal of Academic Engineering Research (IJAER) 3(4): 22-29.
67. Dheir, I. and S. S. Abu-Naser (2019). "Knowledge Based System for Diagnosing Guava Problems." International Journal of Academic Information Systems Research (IJAISR) 3(3): 9-15.
68. El Kahlout, M. I. and S. S. Abu-Naser (2019). "An Expert System for Citrus Diseases Diagnosis." International Journal of Academic Engineering Research (IJAER) 3(4): 1-7.
69. Alshawwa, I. A., et al. (2019). "An Expert System for Coconut Diseases Diagnosis." International Journal of Academic Engineering Research (IJAER) 3(4): 8-13.
70. Al-Shawwa, M. and S. S. Abu-Naser (2019). "Knowledge Based System for Apple Problems Using CLIPS." International Journal of Academic Engineering Research (IJAER) 3(3): 1-11.