# IMPLEMENTATION OF GENETIC ALGORITHM FOR NEURAL NETWORK OPTIMIZATION IN THE CASE STUDY OF THE TRON GAME

**Muhammad Darmadi[1], Irwan Budiman[2], Muliadi[3], Andi Farmadi[4], Triando H. Saragih [5]**
[1,2,3,4,5] Computer Science Study Program FMIPA ULM
Jalan A. Yani Km 36 Banjarbaru, South Kalimantan
e-mail: j1f114076@mhs.ulm.ac.id

**Abstract**

Tron is played in an arena composed of grids and often both players are placed at different starting points, each player basically playing the game by aiming straight, turning left or turning right until one or both of them hit a wall or laser object. This study aims to examine how good genetic algorithms are in optimizing neural networks for artificial intelligence. As well as to find out what the winning percentage is for each researched artificial intelligence. The results obtained are that N5 is faster in obtaining optimal results, which only requires 9 generations but has the lowest percentage. So it can be concluded that the faster finding optimal results does not guarantee that artificial intelligence will be better.

**keywords:** Games, Genetics, Neural networks, Artificial intelligence, Tron.

## 1. INTRODUCTION

Game is taken from the English language which has a basic meaning or the meaning of "Game". The game in this case means "intellectual agility" (intellectual playability). The game can also be understood as an arena for the decisions and actions of the players. There are targets and missions for players to achieve. Games can be classified into several types, namely based on the platform used, genre, and dimensions of the game itself. [2]

The game Tron was adapted into the 1982 film of the same name. Tron is played in an arena composed of grids and often both players are placed at different starting points, each player basically playing the game by aiming straight, turning left or turning right until one or both of them hit a wall or laser object. [5].

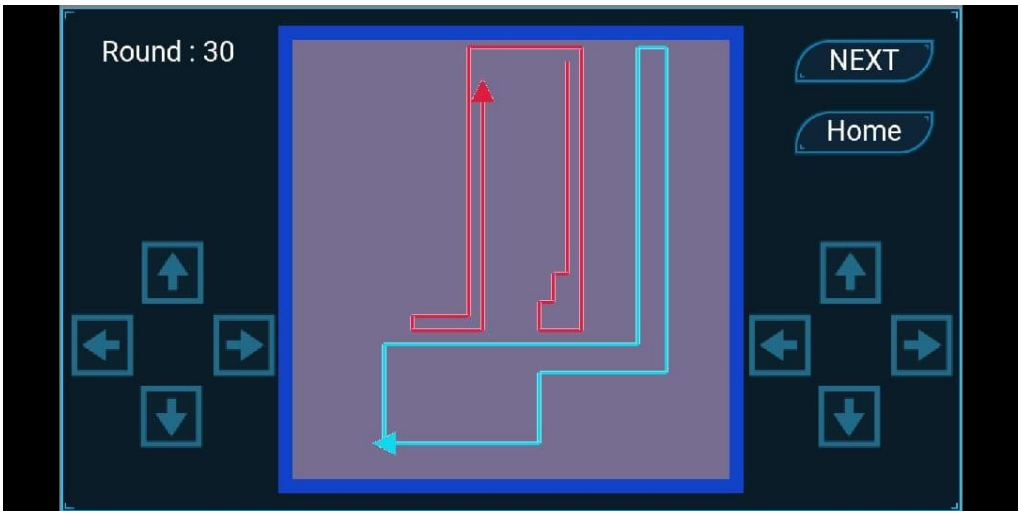


Figure 1 Tron Game

An artificial neural network is a network of a group of processing units modeled on the basis of a human neural network. An artificial neural network is an adaptive system that can change its structure to solve problems based on external and internal information [9].

The neurons in a neural network are arranged in groups, called layers. Basically ANN consists of 3 layers (layers), namely the input layer, process, and output layer. The input layer contains input data variables, the process layer contains object recognition steps and the output layer contains the results of the introduction of an object. In general, there are 2 types of artificial neural networks that are often used, namely: Single-Layer Neural Network and Multilayer Neural Network. [9].

In Single-Layer Neural Network, the input layer is connected directly to the output layer with no hidden layer. The disadvantage of this type is that it can only be used in simple cases and cannot be used for complex cases.

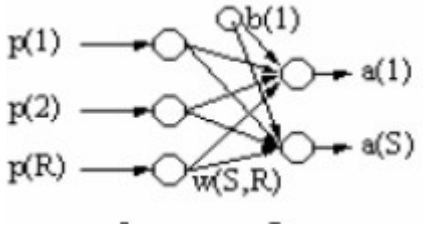


Figure 2 Single Layer Neural Network Schematic

In a Multilayer Neural Network, there is a hidden layer that lies between the input layer and the output layer

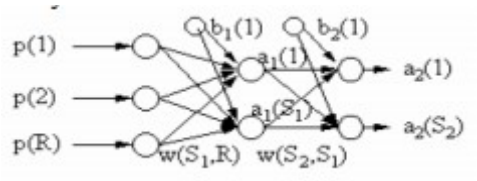


Figure 3 Schematic of Multi Layer Neural Network

The general form of the neural network schema itself is as follows:

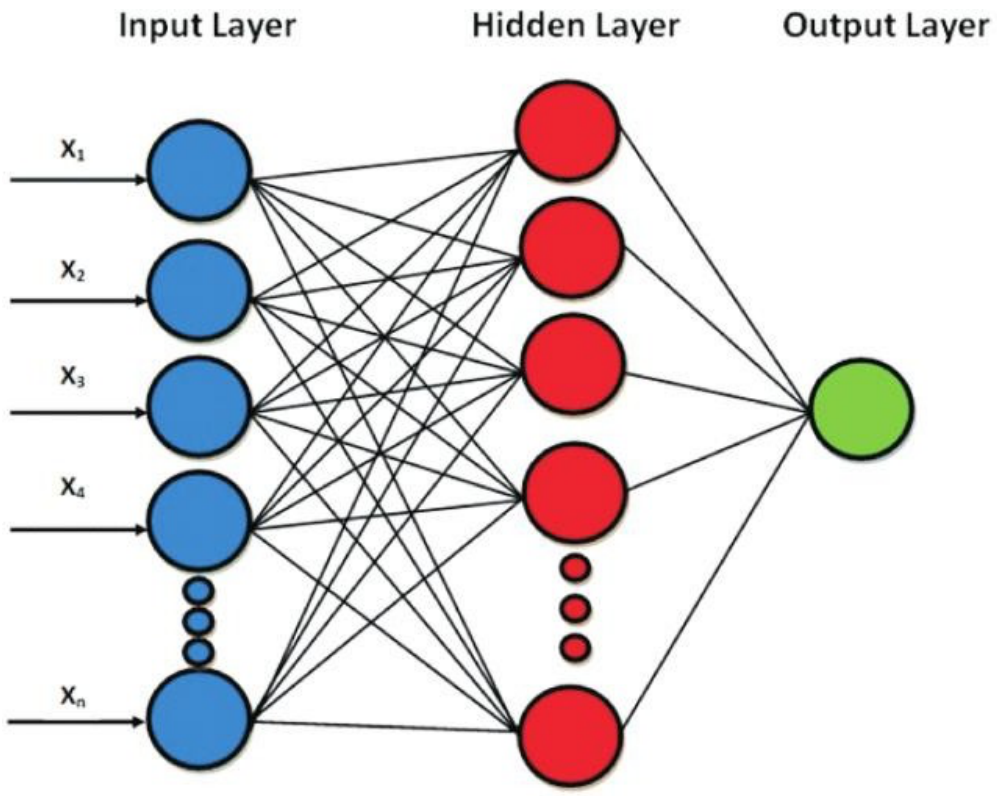


Figure 4 General Schematic of Neural Network

There are main parts in the picture above, namely::

a. The input layer is the units in the input layer called input units which are in charge of receiving input patterns from the outside that describe a problem with 2 nodes.
b. The hidden layer is the units in the hidden layer called hidden units whose output values cannot be observed directly. User-defined number of nodes
c. The output layer, the units in the output layer are called output units, which is an artificial neural network solution to a problem. Consists of 1 node that is the sum of all male and female variables as predicted values

[10]

Genetic algorithm is a stochastic search algorithm based on biological evolution. The basic principle of the Genetic Algorithm is selection, crossing, and individual mutation with the aim of producing a better generation than before [7] as shown in the following figure.

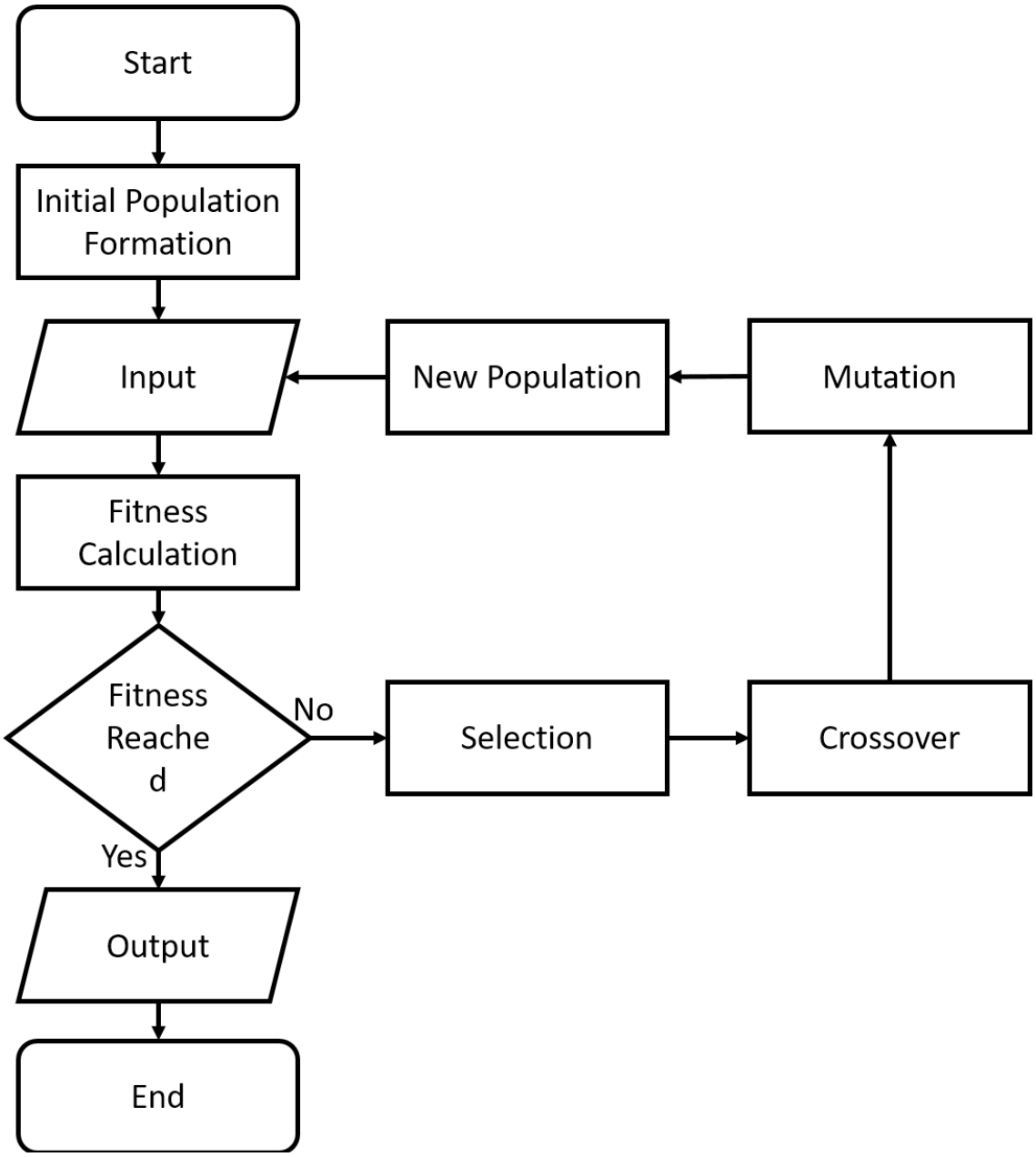


Figure 5 Genetic Algorithm Schematic

The genetic algorithm process itself includes:

a. Formation of Chromosomes and Populations, Chromosomes consist of several genes. Chromosomes are used to represent a possible solution of the problem to be solved using a genetic algorithm.
b. Calculating the Fitness Function, The fitness function of a chromosome is the value of the suitability of the chromosome to the problem. The higher the fitness value, the more optimal the solution should be
c. Selection, selection is a way of choosing a superior population that will be used as a parent for crossing over so as to produce a new population

d. Crossover, interbreeding between two individual populations to get a new population
e. Mutation, the process of modifying one or more genes in an individual to generate a population.
f. Termination Condition, The genetic algorithm stops if there is a globally optimal chromosome or solution, the fitness value has reached convergence, or the iteration has reached the maximum number specified by the user.

## 2. RESEARCH METHODOLOGY

The research methodology carried out in this study is as follows:

a. *Initiation*, Performing the design of the tron game and calculation scheme, using a genetically optimized neural network algorithm in a short description. Made using simple doodles to calculate predetermined results for further implementation.
b. *Pre-Production*, At this stage, game design is carried out, game design in the form of explanations of gameplay, characters, system flow, interface display, asset and sound creation. Prepare all the things needed, assets will use free assets or make your own.
c. *Production*, this stage focuses on translating game designs that have been prepared in the Pre-Production stage into source code lines, asset creation and integration between source code and assets that have been created. The game will be made using the Construct 3 application and will be implemented into the android platform, application and web as needed later.
d. *Optimization,* training data. At this stage, neural network optimization is carried out using genetics based on a predetermined design. There will be repeated optimizations carried out to get the optimal value to be tested against players later.
e. *Testing*, At this stage testing is carried out by testing the algorithm that has been made to find out neurons and the level of victory against players

## 3. RESULTS AND DISCUSSION

### 3.1. Initiation

The game tron is built by involving one player against the computer. The computer is equipped with a neural network calculation that has been optimized in advance using a genetic algorithm. The game will be played in 40 rounds. there are four AIs to be tested and each AI will get 10 rounds against the player. later the results will be calculated what the percentage of AI wins on players.

### *3.2. Pre-production*

The next process is the creation of a Game Design Document for the Tron game which focuses on the genre, target player platform, gameplay, and algorithm calculations. The scheme for the number of hidden layers to be tested is as follows:

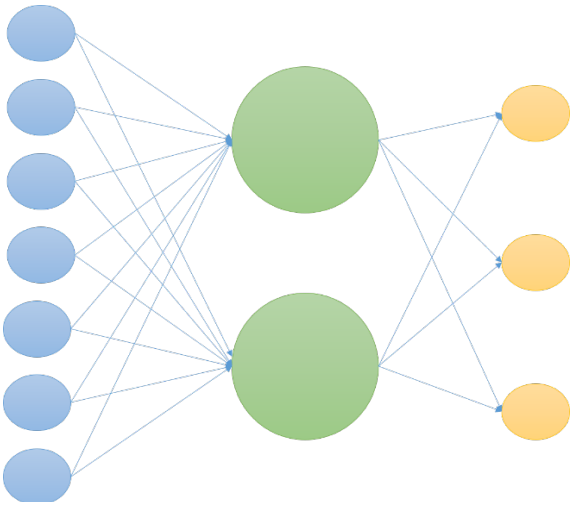

Figure 6 neural network schema with 2 hidden layers

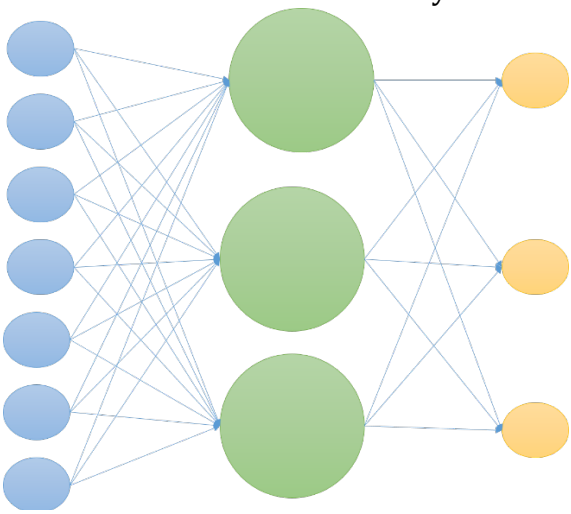

Figure 7 neural network schema with 3 hidden layers

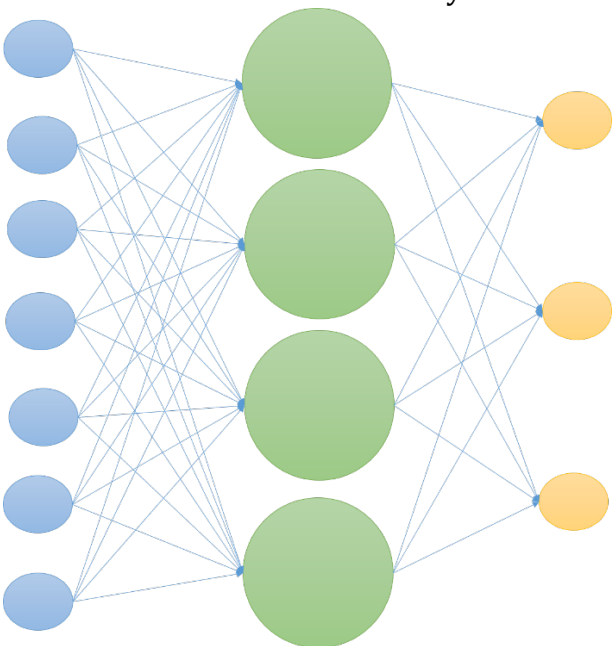

Figure 8 neural network schema with 4 hidden layers

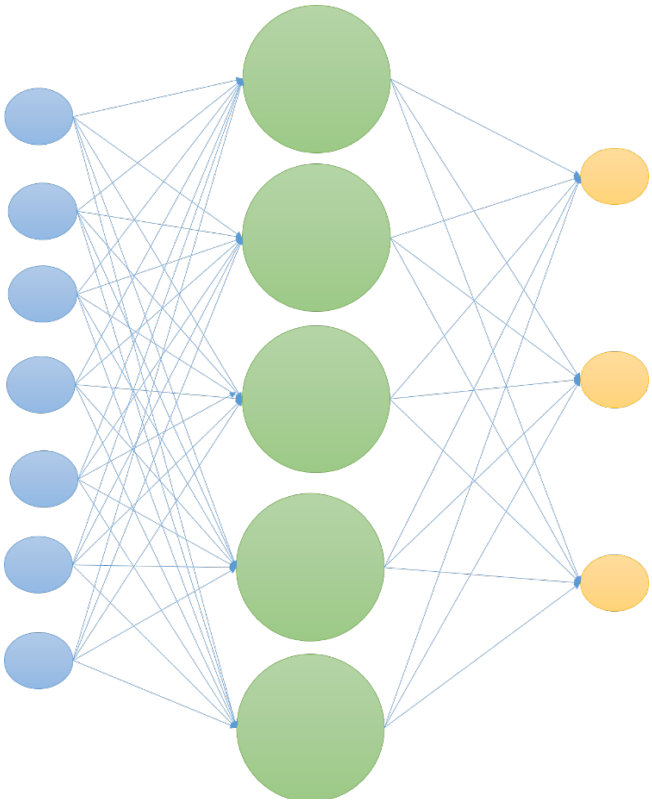

Figure 9 neural network schema with 5 hidden layers

.

### 3.2.1. Genetic Algorithm Calculation

a. New population formation

The formation of the population will consist of 10 populations in each generation, each population will have a number of chromosomes whose number depends on the hidden layer used. The chromosome value at the beginning of the population will be randomized between values -1 to 1, based on the scheme for the number of hidden layers to be tested, it is obtained that the number of hidden layers is 20 chromosomes for hidden layer 2, 30 chromosomes for hidden layer 3, 40 chromosomes for hidden layer 4, and 50 chromosomes for hidden layer 5

**b. Determination of fitness value**

Fitness value is determined based on the distance the computer can move until it loses, and the condition of losing itself if it hits a wall or is hit by a laser. Moving one grid increases the fitness value by 1 until the computer loses.

**c. Selection**

Selection is the determination of which population will be the parent to be crossed over and form a new population, in this case the 4 populations that have the highest fitness value will become the parent. Then it will be chosen randomly from the 4 best populations to be crossed and will form a child.

**d. CrossOver**

Is the process of exchanging chromosomes from 2 previously selected parents, previously randomized values 0 to 1 if less than the crossover rate the chromosomes will be exchanged, otherwise if it is greater than the crossover rate it will not be exchanged. The result of this crossover will be called child.

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Parent 1 | -0,41 | 0,99 | 0,89 | -0,1 | -0,7 | -0,73 | 0,87 | 0,49 | 0,25 | -0,39 | 0,72 | 0,19 | 0,58 | -0,53 | 0,58 | -0,1 | 0,31 | 0,69 | 0,43 | -0,52 |
| Array | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| Parent 2 | 0,83 | 0,07 | 0,9 | -0,03 | 0,66 | -1 | -0,63 | -0,06 | 0,39 | 0,03 | 0,21 | -0,55 | -0,16 | -0,38 | 0,69 | -0,94 | -0,16 | 0,8 | -0,32 | -0,4 |

*Figure* 10. parent example obtained before crossover

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cross Over Rate | 0,81 | 0,77 | ***0,1*** | 0,72 | 0,94 | 0,39 | 0,42 | 0,87 | 0,82 | ***0,15*** | 0,79 | 0,76 | ***0,27*** | 0,65 | 0,68 | 0,84 | 0,87 | ***0,28*** | 0,93 | 0,99 |

*Figure* 11. random crossover

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Child 1 | -0,41 | 0,99 | ***0,9*** | -0,1 | -0,7 | -0,73 | 0,87 | 0,49 | 0,25 | ***0,03*** | 0,72 | 0,19 | ***-0,16*** | -0,53 | 0,58 | -0,1 | 0,31 | ***0,8*** | 0,43 | -0,52 |
| Array | 0 | 1 | ***2*** | 3 | 4 | 5 | 6 | 7 | 8 | ***9*** | 10 | 11 | ***12*** | 13 | 14 | 15 | 16 | ***17*** | 18 | 19 |
| Child 2 | 0,83 | 0,07 | ***0,98*** | -0,03 | 0,66 | -1 | -0,63 | -0,06 | 0,39 | ***-0,39*** | 0,21 | -0,55 | ***0,58*** | -0,38 | 0,69 | -0,94 | -0,16 | ***0,69*** | -0,32 | -0,4 |

*Figure* 12. Result crossover

**e. Mutation**

Mutations are changes in chromosomes in children, mutations also have a mutation rate that will determine whether a chromosome will mutate or remain. If the mutation rate value is below 0.3 it will mutate into a new number between -1 to 1

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Child 1 | -0,41 | 0,99 | 0,9 | -0,1 | -0,7 | -0,73 | 0,87 | 0,49 | 0,25 | 0,03 | 0,72 | 0,19 | -0,16 | -0,53 | 0,58 | -0,1 | 0,31 | 0,8 | 0,43 | -0,52 |
| Mutate Rate | 0,76 | ***0,16*** | 0,91 | 0,5 | ***0,01*** | 0,63 | 0,76 | ***0,09*** | 0,54 | ***0,29*** | 0,47 | 0,97 | ***0,29*** | 0,62 | 0,97 | ***0,06*** | 0,38 | 0,8 | ***0,17*** | ***0,17*** |
| Child 1 mutate | -0,41 | ***0,56*** | 0,9 | -0,1 | ***0,83*** | -0,73 | 0,87 | ***-0,01*** | 0,25 | ***0,25*** | 0,72 | 0,19 | ***-0,9*** | -0,53 | 0,58 | ***0,02*** | 0,31 | 0,8 | ***-0,42*** | ***0,17*** |

*Figure* 13. mutated child result

**f. Formation of a New Generation**

This new generation will consist of 4 best parents who have been previously recovered and then 4 children resulting from crossbreeding and mutations then 2 more will return randomly so that it will get 10 populations in the new generation.

**g. Stop Condition**

The stopping condition is a condition that will determine the iteration of the formation of a new population. That is, if the fitness value of a population reaches the fitness target. The fitness target determined by the author is 70% of the total grid. If the grid used is 30x30, in other words, the number of tiles is 900. Then 70% of 900 is 630. If any grid reaches that fitness, the loop will stop and the population with the fitness value will be stored as an AI to use against players

### 3.2.2. Training Results

The procedure for conducting this training is that the test is carried out with an arena with an area of 30x30 grids, each population starts a random position to make it more varied, the calculation process is the same as that described in the genetic algorithm process, the process will stop when the fitness reaches 70% of the total grid. After the training, the following results were obtained:.

a. *Training Result for N2*

*After doing training for N2, the optimal fitness value is obtained in the 30th generation with a fitness value of 855 and the following chromosomes in the population are,*

*[[[-0.97]], [[-0.92]], [[-0.86]], [[0.84]], [[0.7]], [[0.31]], [[0.65]], [[0.92]], [[-0.45]], [[-0.1]], [[0.7]], [[-0.74]], [[0.65]], [[-0.56]], [[-0.41]], [[-0.11]], [[-0.18]], [[-1]], [[0.85]], [[-0.18]]]*

b. *Training Result for N3*

*After doing training for N3, the optimal fitness value is obtained in the 55th generation with a fitness value of 639 and the following chromosomes in the population are,*

*[[[0.48]], [[0.6]], [[0.59]], [[-0.99]], [[0.03]], [[-0.38]], [[0.39]], [[-0.5]], [[-0.04]], [[-0.59]], [[-0.96]], [[-0.93]], [[-0.95]], [[-0.66]], [[-0.8]], [[0.83]], [[0.1]], [[0.1]], [[0.42]], [[0.82]], [[-0.59]], [[0.57]], [[-0.76]], [[-0.36]], [[0.3]], [[-0.27]], [[0.01]], [[0.84]], [[-0.42]], [[0.29]]]*

c. *Training Result for N4*

*After doing training for N4 gets the optimal fitness value in the 26th generation with a fitness value of 689 and the following chromosomes in the population are,*

*[[[-0.15]], [[-0.17]], [[0.06]], [[-0.17]], [[-0.24]], [[-0.97]], [[0.92]], [[-0.26]], [[-0.64]], [[0.95]], [[-0.65]], [[-0.05]], [[0.74]], [[-0.97]], [[0.75]], [[-0.41]], [[0.87]], [[-0.46]], [[0.4]], [[-0.67]], [[0.59]], [[-0.36]], [[-0.14]], [[0.77]], [[-0.63]], [[0.31]], [[-0.88]], [[0.78]], [[0.98]], [[0.17]], [[0.06]], [[0.46]], [[-0.08]], [[0.24]], [[-0.23]], [[-0.22]], [[-0.99]], [[-0.35]], [[-0.71]], [[-0.55]]]*

d. *Training Result for N5*

*After doing training for N5, the optimal fitness value is obtained in the 9th generation with a fitness value of 758 and the following chromosomes in the population are,*

*[[[0.98]], [[-0.77]], [[-0.82]], [[0.74]], [[-0.39]], [[0.62]], [[0.49]], [[-0.14]], [[-0.32]], [[0.27]], [[-0.57]], [[-0.23]], [[-0.52]], [[-0.29]], [[-0.07]], [[0.72]], [[0.99]], [[0.6]], [[0.81]], [[0.69]], [[0.64]], [[0.96]], [[0.2]], [[-0.44]], [[0.58]], [[0.26]], [[0.61]], [[0.05]], [[0.73]], [[-0.15]], [[-0.86]], [[0.84]], [[-0.12]], [[0.24]], [[-0.99]], [[-0.26]], [[0.3]], [[-0.6]], [[-0.43]], [[0.63]], [[-0.06]], [[-0.82]], [[-0.82]], [[-0.21]], [[0.8]], [[-0.85]], [[0.32]], [[-0.75]], [[0.88]], [[0]]]*

### 3.3. Testing

The testing procedure for testing: The application being played can be played via the android application or via the web, Each participant will play 40 rounds consisting of 10 rounds each of the AI, The order of use of the AI is randomized and determined from the start, After finishing playing all the rounds then the results of the number of wins, draws and draws will appear against the computer

In Testing, testing was carried out on 15 people who had never played the game tron, while the data obtained were as follows in the form of percentages:.

Tabel 1. Hasil Akurasi pada semua *dataset*

| ***Kode Player*** | ***Player*** | ***AI*** | ***Player Win*** | ***Draw*** | ***Computer Win*** |
|---|---|---|---|---|---|
| **P1** | **Muhammad Darmadi** | N2 | 70 | 0 | 30 |
| | | N3 | 40 | 0 | 60 |
| | | N4 | 50 | 0 | 50 |
| | | N5 | 80 | 10 | 10 |
| **P2** | **Riza Susanto** | N2 | 20 | 0 | 80 |
| | | N3 | 20 | 10 | 70 |
| | | N4 | 30 | 0 | 70 |
| | | N5 | 70 | 10 | 20 |
| **P3** | **Fitri Nadi** | N2 | 20 | 0 | 80 |
| | | N3 | 30 | 0 | 70 |
| | | N4 | 20 | 10 | 70 |
| | | N5 | 50 | 10 | 40 |
| **P4** | **Eko Prayogo** | N2 | 0 | 0 | 100 |
| | | N3 | 40 | 0 | 60 |
| | | N4 | 30 | 20 | 50 |
| | | N5 | 90 | 0 | 10 |
| **P5** | **Fitria Agustina** | N2 | 30 | 10 | 60 |
| | | N3 | 30 | 10 | 60 |
| | | N4 | 40 | 10 | 50 |
| | | N5 | 30 | 0 | 70 |
| **P6** | **Akhmad Kharis** | N2 | 10 | 10 | 80 |
| | | N3 | 30 | 20 | 50 |
| | | N4 | 0 | 0 | 100 |
| | | N5 | 30 | 10 | 60 |
| **P7** | **Rifqi idztihar** | N2 | 50 | 0 | 50 |
| | | N3 | 20 | 10 | 70 |
| | | N4 | 30 | 0 | 70 |
| | | N5 | 50 | 10 | 40 |
| **P8** | **Faisal Murtadho** | N2 | 40 | 0 | 60 |
| | | N3 | 20 | 10 | 70 |
| | | N4 | 40 | 0 | 60 |
| | | N5 | 60 | 0 | 40 |

| | | | | | |
|---|---|---|---|---|---|
| **P9** | **Iswan Riadi** | N2 | 40 | 0 | 60 |
| | | N3 | 60 | 0 | 40 |
| | | N4 | 80 | 0 | 20 |
| | | N5 | 60 | 10 | 30 |
| **P10** | **Zainal Ilmi** | N2 | 50 | 0 | 50 |
| | | N3 | 60 | 10 | 30 |
| | | N4 | 90 | 0 | 10 |
| | | N5 | 90 | 0 | 10 |
| **P11** | **Salsabila Anjani** | N2 | 10 | 0 | 90 |
| | | N3 | 30 | 0 | 70 |
| | | N4 | 0 | 10 | 90 |
| | | N5 | 60 | 10 | 30 |
| **P12** | **Muhammad Reza Ramadhana** | N2 | 20 | 0 | 80 |
| | | N3 | 10 | 0 | 90 |
| | | N4 | 30 | 10 | 60 |
| | | N5 | 50 | 0 | 50 |
| **P13** | **Tri Cahyo Swasono** | N2 | 50 | 0 | 50 |
| | | N3 | 70 | 0 | 30 |
| | | N4 | 60 | 20 | 20 |
| | | N5 | 100 | 0 | 0 |
| **P14** | **Ardiansyah Sukma Wijaya** | N2 | 70 | 10 | 20 |
| | | N3 | 70 | 0 | 30 |
| | | N4 | 80 | 0 | 20 |
| | | N5 | 60 | 10 | 30 |
| **P15** | **Muhamma Ihsanul Qamil** | N2 | 10 | 0 | 90 |
| | | N3 | 30 | 10 | 60 |
| | | N4 | 30 | 10 | 60 |
| | | N5 | 60 | 0 | 40 |

Based on the results of trials conducted on 15 people who have never played the Tron game, the results obtained are games with Hidden Layer 2 Amount winning with a percentage of 65.33%, losing 32.67% and a draw 2%. with the number of hidden layer 3 wins with a percentage of 57.33%, loses 37.33% and draws 5.33%. with Total Hidden layer 4 wins with a percentage of 53.33% losing 40.67% and a draw 6%. And finally with the number of hidden layers 5, winning with a percentage of 32%, losing 62.67% and drawing 5.33%

## 4. CONCLUSION

From the research that has been done, it can be concluded as follows: For neural network optimization using genetics on N5 is the fastest with only 9 generations but also has the lowest win rate than others, For the percentage of wins N2 has the highest percentage of wins than the others, namely 65.33% followed by N3 at 57.33%, N4 at 53.33% and N5 having the lowest percentage of only 32%. So it

can be concluded that the faster getting the optimal value in training does not guarantee the level of intelligence of an artificial intelligence.

**BIBLIOGRAPHY**

[1] Abdi Prawiram, Mahardhika. 2013. **Analysis of the influence of storytelling on the game of time passage-Prince Diponegoro as a historical education medium**. ISSN : 2089-9033

**[2]** Asmiatun & Putri. 2017. **Learn to Make 2D and 3D Games Using Unity.** Yogyakarta : Deepublish.

[3] Karim, Syafei. 2017. **Changes in the behavior of Non-Player Character (NPC) in the Arabic Hunter Game using the Perceptron Neural Network**. Scientific Journal of Information Systems Technology 3 (2) 34-41. ISSN 2502-3357.

[4] Kusnadi, Adhi dan Pratama, Jansen. 2017. **Implementation of Genetic Algorithms and Neural Networks in Noodle Production Forecasting Applications (Case Study: Omega Mie Jaya).** ULTIMATICS, Vol. IX, No. 1 | Juni 2017, ISSN 2085-4552..

[5] Lanctot, Marc dkk. 2013. **Monte Carlo Tree Search for Simultaneous Move Games: A Case Study in the Game of Tron**.

[6] Masri, Ari Tri Mukti. 2014. **Shortest Path Search In Snake Game Using A* Algorithm.** Scientific Journal of SISFOTENIKA, Vol 4 No.1 January 2014, ISSN 2087-7897.

[7] Pranata, John dkk. 2017. **Strategy Game Application Using Genetic Algorithm Method for Troop Making**. Journal Infra Vol 4, No 1 (2016) page. pp.199 - pp.204

[8] Rulaningtyas, Riries. 2007. **Application of Neural Network Optimization by Genetic Algorithm on the Detection of Ischemic Stroke Brain Disorders as a Learning Media for Young Doctors.** Journal of Physics and Its Applications Vol 3, No. 1 January 2007, 070111|1-4

[9] Noviando, Satrio Edo dkk. 2016. **Study on the Application of ANN (Artificial Neural Network) to Eliminate Harmonics in the Computer Center Building.** Let's FTEKNIK Volume 3 No. October 2, 2016

[10] Sudarsono, Aji. 2016. **Artificial Neural Network to Predict Population Growth Rate Using Bacpropagation Method (Case Study In Bengkulu City**). Journal of Media Infotama Vol. 12 No. 1, February 2016. ISSN 1858 – 2680