APPLICATION OF NEURAL NETWORKS IN VIDEO GAME SIMULATION

Biljana Tešić*,
Marko Marković,
Katarina Marković

University Singidunum,
Faculty of Health and
Business Studies,
Valjevo, Serbia

Abstract:

The subject of this paper is the development of 2D/3D video games using neural networks. The simulation method was used to design and program the video game Super Mario, which with its well-known functionality enables the application of artificial intelligence methods and neural networks. This paper aims to understand the relationship between the use of video games and their neural correlates, taking into account the whole range of cognitive factors that they include. The results of the research indicate the importance of using artificial neural networks in video games, bearing in mind that prediction is closely related to learning and that the existence of feedback allows game participants to evaluate their performance and increase the quality of future prediction of situations and moves. Despite the heterogeneity of the field of study, the research results indicate that it is possible to establish links between neural and cognitive aspects, especially in terms of attention, cognitive control, visual-spatial skills, cognitive load. However, many aspects could be improved. The lack of standardization in various aspects of video game-related research, such as participant characteristics, characteristics of each video game genre, and different goals, could contribute to disagreements with some related research.

Keywords:

Neural networks, video games, FeedForward, neuron.

INTRODUCTION

Video games are an increasingly popular activity in modern society, especially among young people, and are becoming increasingly popular not only as a research tool but also as a field of study. Many studies have focused on the neural and behavioural effects of video games, providing much of the brain correlation from video games in recent decades. There is a large amount of information, obtained through countless methods, providing neural correlates of video games.

Artificial Neural Networks (ANN - *Artificial Neural Networks*) are one of the most popular artificial intelligence techniques. In recent years, they have been applied in many areas and have become indispensable in solving increasingly complex problems that arise in the modern world. Starting from the fact that artificial neural networks are a family of statistical learning models based on biological neural networks, more

Correspondence:
Biljana Tešić

e-mail:
btesic@isingidunum.ac.rs

DOI: 10.15308/Sinteza-2022-20-28

precisely neurons, the basic idea of using artificial neural networks in this paper is that computer simulation enables learning concepts, pattern recognition and decision making in a human-like way.

For a neural network (NN - *Neural Networks*) to learn to recognize and classify concepts, there must be feedback. The connection between biological and artificial neurons can be seen in the example of the human brain because all people use feedback at all times. In this case, the brain of the player playing the video game for the first time observes the way the opponent moves and creates an image of what would be the simplest way to reach the goal, recording good and bad moves. The next time he plays, the brain remembers what he did wrong and corrects it, hoping to achieve better results. Feedback is used to compare the desired outcome with the outcome that occurred.

It is expected that the Super Mario video game will work on that principle, by improving the application by maintaining the quality of the game at the level of each player's decision, without disturbing the balance and basic principles of AI-Artificial Intelligence functionality. The reason for the improvement is that even more experienced players, who have mastered the mechanics of the game, will continue to be proportionately interested because simple mechanical behaviour will change and adapt, adding an element of surprise.

The paper uses a feedforward neural network. This network was the first, and also the simplest neural network. The flow of information is one-way, from input units, data passes through hidden units (if any) to output units. There are no cycles in the network, unlike recurrent neural networks.

2. VIDEO GAMES AND NEURAL NETWORKS

Artificial intelligence appeared in the video game back in 1992 in Wolfenstein 3D [1] by representing in certain video games supporting characters who come in contact with a human-controlled character. A non-player character (NPCs) with well-programmed artificial intelligence can follow you at your own pace as you run or walk [2]. Some video games determine the level of difficulty depending on whether the player is good or not so good, or adapt the video game depending on the player's style of play. By improving artificial intelligence and its implementation, video games are more interesting and customized depending on how the player plays or behaves in them. Also, artificial intelligence in video games can motivate a player and teach him perseverance so that he can be as good as possible.

Artificial neural networks are one form of implementation of artificial intelligence systems. They are made up of process elements that we call artificial neurons. The body of a neuron is called a node or unit. Each of the neurons has a local memory in which it remembers the data it processes. Patterns that neural networks recognize are presented numerically, so all real-world data, such as images, sounds, or text, must be translated. [3] In video games, they serve as a platform for learning how to communicate with the environment and solve complex problems as in real life. [4]

Reflections on the human brain have contributed to expanding the reach of technical ideas. Among the first works published in the field of artificial intelligence, McCulloch and Pitts published the first attempts to create artificial neural networks in the early 1940s [5]. After that period, more detailed and realistic models began to develop. Today, neural networks are one of the most popular and effective forms of training systems and deserve independent study. They are suitable for solving distinctly nonlinear problems [6]. They can learn certain non-dynamic properties of the system, and then take control of it.

Neural networks are often used to simulate an opposing player in various video games. The techniques used range from the use of evolutionary algorithms in combination with neural networks, through supportive learning and assigning grades, to the use of neural network techniques in combination with game theory. Namely, according to Karl Kapp, Lucas Blair and Rich Mesh [7], one of the definitions of video games is that they represent a system in which players participate in abstract challenges, defined by certain rules, whose feedback often results in some form of emotional reaction.

Mathematical model of the neurons on which the FeedForward network is based

Neural network architecture represents the specific connection of neurons into one whole. Each neuron consists of a cell body that contains a nucleus. A certain number of fibres are called dendrites and one long fibre called an axon branch from the body of the cell. A neuron makes connections with 103 to 104 other neurons, and their connections are called synapses [8]. The structure of the neural network differs in the number of layers. The first layer is input, and the last is output, while the layers in between are called hidden layers. There are usually three of them, but this mainly refers to smaller projects because the larger the number of neurons in the hidden layer, the more time it takes to overcome complex situations.

The first layer, ie. the input is the only layer that receives data from the external environment, which then further excites the layers of the hidden units, from where the relevant data is further passed to the third (output) layer. The final result was obtained at the output of the third layer. More complex neural networks have more hidden layers that are completely interconnected. This common design is called a feedforward network. In the example presented in the paper on the input layer, 2 neurons are used.

Layers communicate by connecting the output of each neuron from the previous layer to the inputs of all neurons in the next layer. So, each node has several inputs and one output. The strength of the connections by which neurons are connected is called the weight factor.

The learning of NN is reduced to learning from examples, which should be as many as possible so that the network can behave more precisely in later exploitation. The learning process leads to the correction of synaptic weights. When the patterns presented to the network no longer lead to a change in these coefficients, the peak of learning is considered to have been reached. There are three types of training:

- supervised training - the network presents input data and expected output data.
- evaluation training - the network is not presented with the expected output data, but after some time it is presented with the evaluation of previous work. One example is a net that learns to balance a rod. Whenever the rod falls, an evaluation of previous work is forwarded to the net, for example, in the form of an angular deviation of the rod from equilibrium.
- self-organization - networks are the only entrance.

The network with all inputs directly connected to the outputs is called a single-layer neural network or perceptron. Perceptrons are the best understood and most widely used of all the neural networks. The term perceptron was first used by Frank Rosenblatt in 1958. His idea was to make a functional description of how a real neuron works and then implement it as a software algorithm [9].

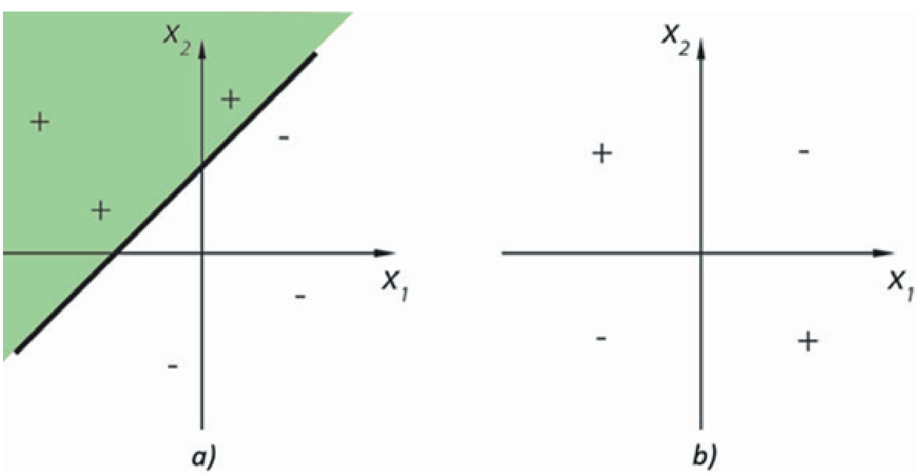
To best understand the use of perceptron, it is necessary to look at a practical example. If a set of points is given that has either a positive or negative value (each of them is represented by + or -), the perceptron can be trained to determine a line that will divide the set into positive and negative values. The set of input data for which it is possible to determine a line that divides it into two sets is called a linearly separable set (Graph 1 a)). Of course, there will always be sets that cannot be divided homogeneously (Graph 1 b)).

Neural networks consist of nodes connected by directional connections (Figure 2). The connection from unit i to unit j serves to spread the activation denoted by ai from i to j. Each connection has an associated and numerical weight wi, j. All units have one input a0 = 1 with the corresponding weight w0, j. Each unit j first calculates the weighted sum of its inputs [10,11]:

$$in_1 = \sum_{i=0}^{n} w_{ij} a_i .$$

Then the activation function f is applied to this sum to get the output:

$$a_1 = f(in_1) = f\left(\sum_{i=0}^{n} w_{ij} a_i\right).$$



Graph 1 – Division of the input set using perceptron. (Adapted from: Mitchell T., Machine Learning, McGraw Hill, Boston, 1997, p. 87)

4. SOLUTION DESCRIPTION

The programming language in which the video game was developed is C #, MS Visual Studio 2017, the design environment is Unity, and Unity Engine packages have been used for certain functions. The program consists of several parts, a separate program code is written for each instance, while in the end all instances are merged and form a single program unit. In the main instance-NeuralNetwork.cs, there is a general algorithm by which all special parts work.

The main goal is to find a way to keep the video game interesting no matter how much the player reports in it. All opponents have a simple pattern of behaviour on which it is easy to insert an element of surprise that adapts and changes with the skill level of the player. It can be equivalently applied to the more complicated ANN, but, as already mentioned, the video game Super Mario was simulated for easier visibility.

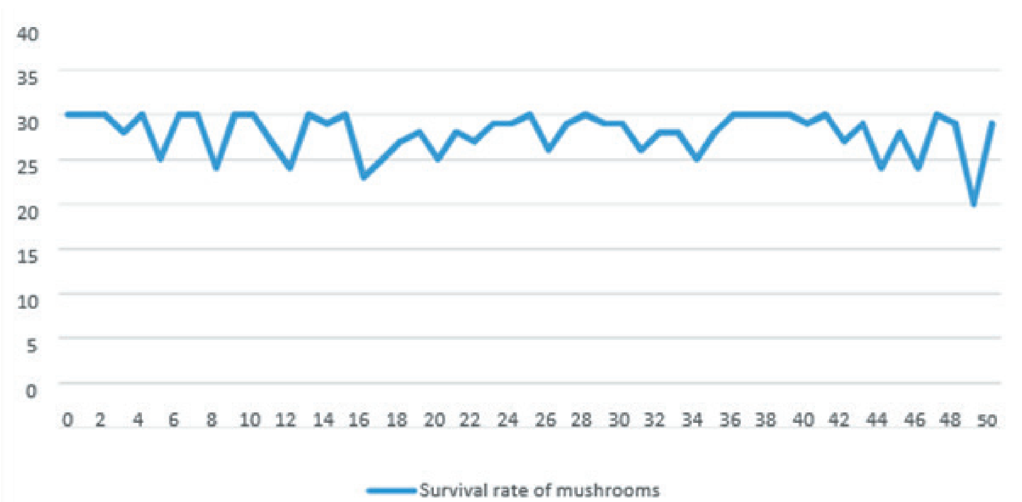
The main obstacle for Mario is the "mushroom", which is relatively easy to manage and is controlled by the neural network. During the first generation of starting the program, the "mushroom" moves only to the left, until it encounters the first obstacle, ie the wall ("end of the screen"). From generation to generation, the "mushroom" is moving more and more toward Mario intending to win over him. Of course, she can easily be eliminated with one jump from Mario. Namely, from generation to generation, "mushroom" (by using NN) "creates awareness" that she should be at a short distance from Mario to be able to avoid a potential attack, or manage to defeat Mario.

5. EXPERIMENTAL RESULTS

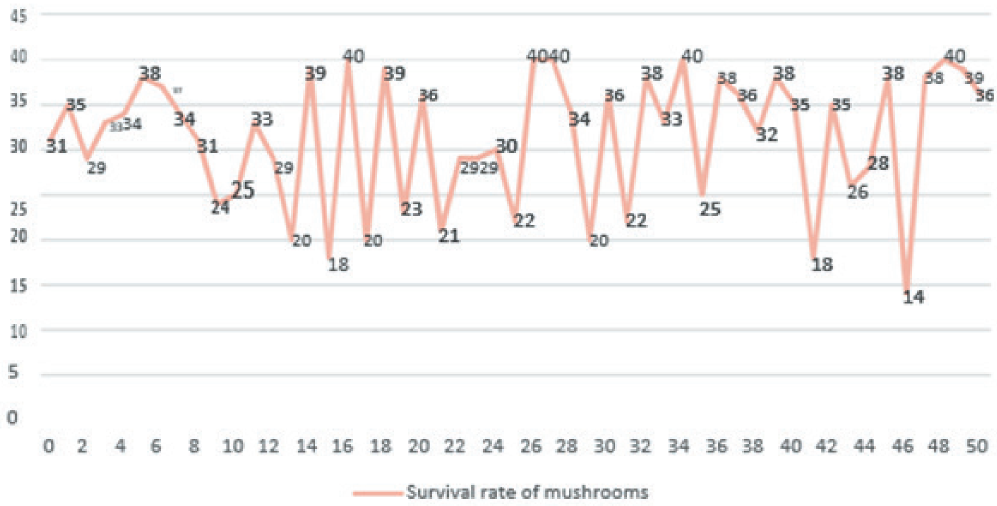
The video game was tested by experimental methods based on the opinions of the players and the observations of observers who, by participating in the game, made it possible to reach concrete conclusions. Starting from the fact that the experiences of the players and the way of playing changed from generation to generation, the goal was to establish the influence of ANN on the changes and improvements of the described video game.

The following graphics show the behaviour of all actors in this video game. Thus, we mean their change of behaviour from generation to generation (in this case 50 generations), which depends on the experience of the player and his abilities, but also on the fact that the "mushrooms" are getting smarter as time goes by. A neural network was used whose neurons have one input layer, and the goal is to define the relationship of the "mushroom" in the state of motion concerning Super Mario. More precisely, based on the distance formula: $D= [\sqrt{(x1^2 - x2^2)-(y1^2 - y2^2)}]$, which determines the shortest distance between two points in a two-dimensional plane, defines the distance in 2D space where x is the difference between x1 mushroom coordinates and x2 coordinates of Mario, and y is the difference of y1 coordinates of mushrooms and y2 coordinates of Mario.

Graph 2 shows that the elimination of "mushrooms" varies. We conclude that the game becomes more complicated over time because the "mushrooms" gain the ability to avoid attacks and eliminate Mario more efficiently and more easily. At the same time, the players' experience is a bit worse, which increases their speed of adaptation and observation at a certain level of the game.



Graph 2 – A survival rate of mushrooms, by generation, D - player 1



Graph 3 – Mushrooms death by generation, D - player 2

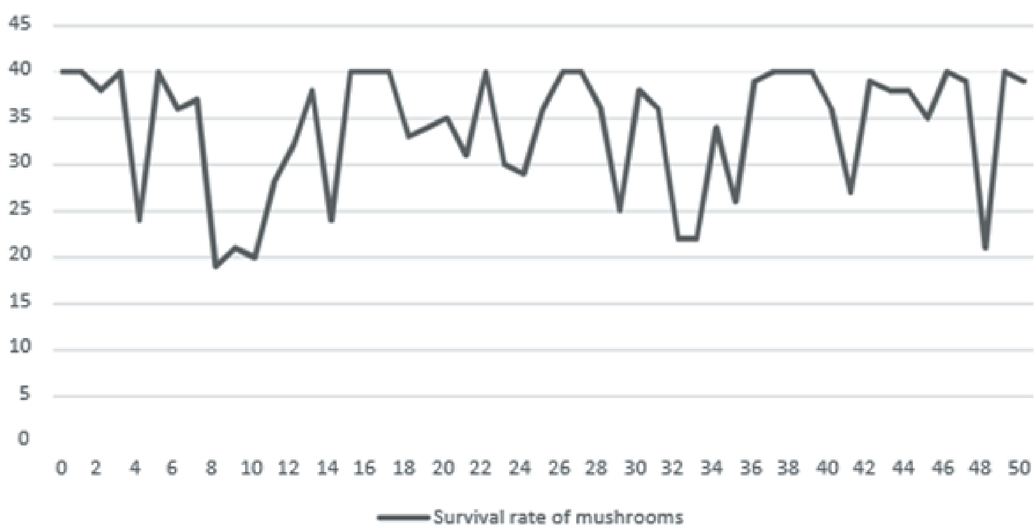
Graph 3 shows that the skill of the player is much higher, which additionally enabled him to learn effectively and to adapt to the game, which also applies to ANN from the aspect of improvement.

As in the previous case, the learning speed of the "mushrooms" is proportional to the learning speed of the player, with the fact that the player, in terms of skills and learning, is at an intermediate level (Graph 4).

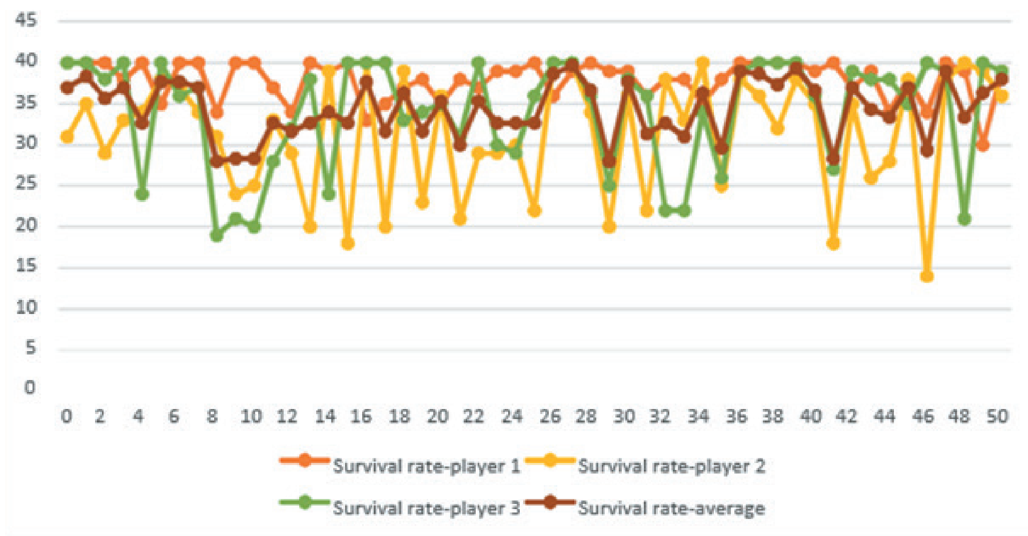
After testing the game with 3 different players, an average was made, because all three previous players have different skill levels. As can be seen in Graph 5, even though these are three players with different skills and experiences, the average line (brown line) shows both falls and rises in an almost uniform order. This proves that the process of "mushrooms" getting used to players, as well as the process of players getting used to "mushrooms" is a recurring cycle.

In the following graphs, a neural network is used whose neurons have two input layers, which means that the "mushrooms" move and observe Mario's position. Player distance is defined using the Distance formula.

With this approach, it can be seen that the "mushrooms" have acquired a higher degree of ability to perceive the environment around them, from generation to generation they become smarter, but this time at a higher level. Based on learning from previous situations, they react faster to potential attacks by Mario and eliminate their opponents more efficiently.



Graph 4 – Mushrooms death by generation, D - player 3



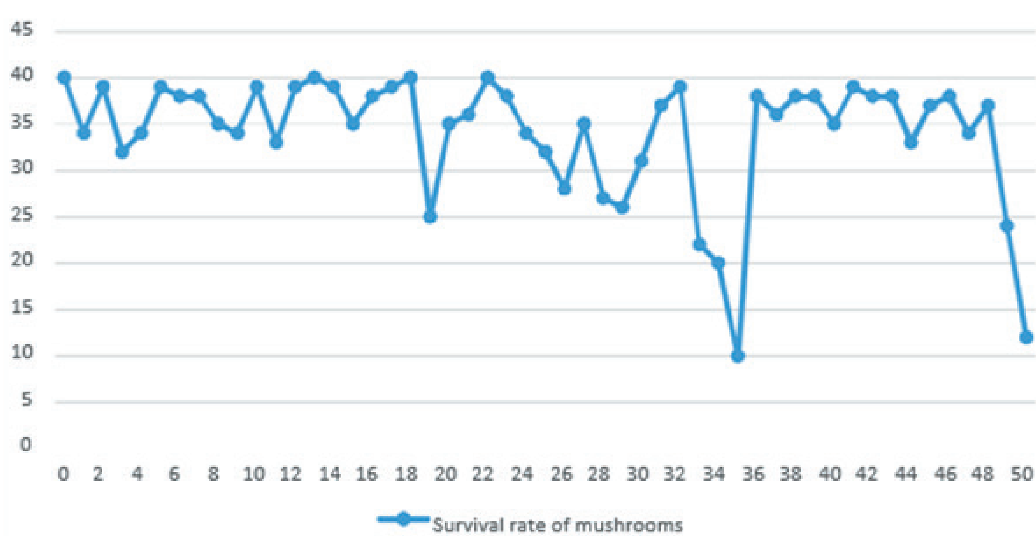
Graph 5 – Mushrooms death by generation, D - average

Graph 6 shows that in the beginning the player only gets acquainted with the game, so the result does not vary much, while after a certain number of attempts, the player's abilities to react increase, which leads to a better result. At the same time, there is an improvement in the ability of the "mushroom" to react.

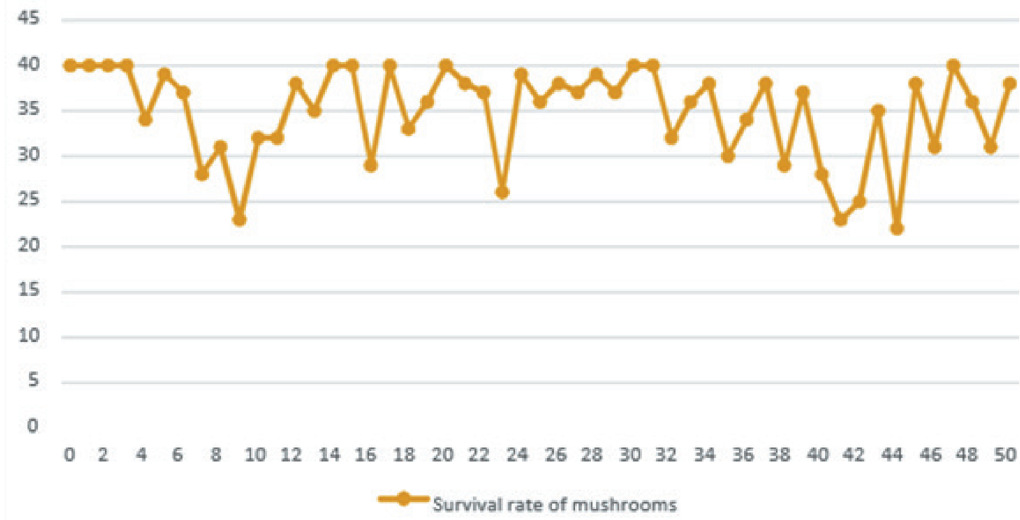
In the case of another player, at the beginning of the game, the result varies, and then during the game, it remains at a higher level, which leads to improving the responsiveness of the "mushroom" in avoiding attacks.

After testing the game with 2 different players, an average was made, which is marked in green on Graph 8.

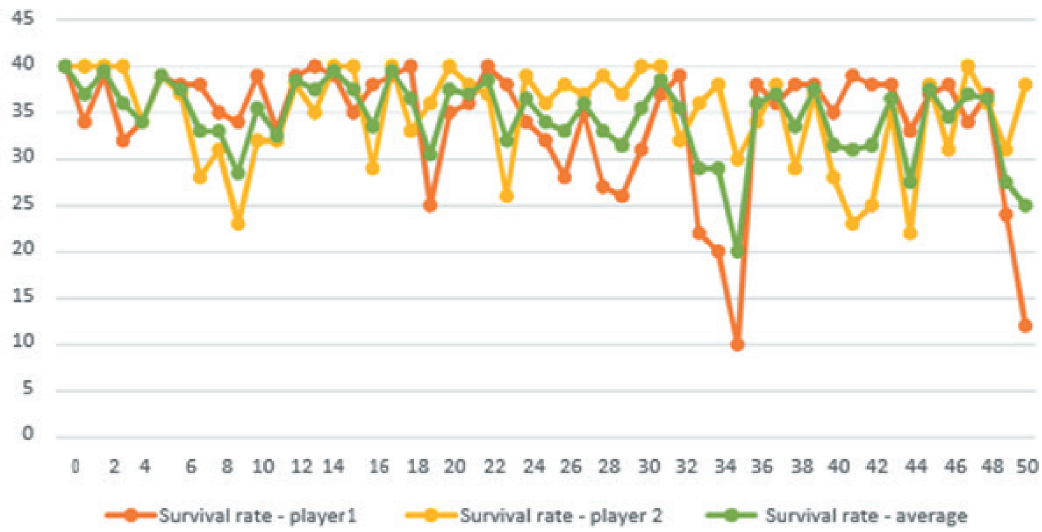
Graph 9 shows how the player gets used to the "mushrooms" in situations where they raise the level of difficulty of the game to adapt to the player. It is noticed that the player survives for a longer period, based on learning how the "mushrooms" behave and the way they move. A significant drop in the intensity of the defence, causes the behaviour of the mushroom to change in the next stage of the game. In this test, Mario behaves the same as in the first test, by trying to eliminate as many "mushrooms" as possible, but this time the goal is to stay in the game as long as possible. This, as well as the previous experiment, gives a clear picture of the interaction in the learning process on the relation "mushroom" - player and player - "mushroom".



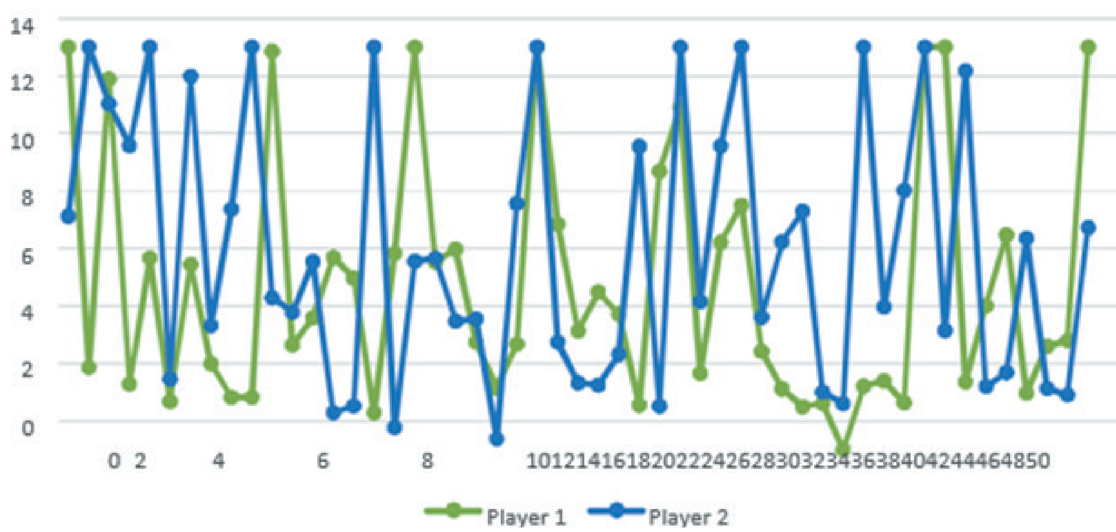
Graph 6 – Mushrooms death by generation, X-Y - player 1



Graph 7 – Mushrooms death by generation, X-Y - player 2



Graph 8 – Mushrooms death by generation, X-Y - average



Graph 9 – Mario death timers

6. CONCLUSION

Based on testing, algorithmic analysis that defines the behaviour of certain elements in the video game Super Mario, as well as based on experimental results, it is clear that neuron-based agents can overcome some of the shortcomings associated with classical AI techniques in video game design. In this thesis, artificial intelligence should be trained with the help of supervised, unsupervised and reinforced learning (machine learning methods) to beat human records in the game Super Mario. That is, it describes how the algorithm accurately recognizes and treats a particular game and its elements and what effects it has on players.

The paper points out the importance of applying an algorithm that uses supervised learning and assisted learning methods, which confirms the sustainable implementation of AI-controlled neural networks in the simulation of 2D and 3D video games. Supervised learning is effective in applying best practices to existing data, and applicable to the functionality of the Super Mario video game, given that players' behaviour, through multiple testing phases, is used to learn the NN algorithm. The algorithm can first be trained using pre-existing data, as far as the basic functions of the game are concerned. In Super Mario, this would be running, squatting, jumping, collecting coins and avoiding opponents.

As an additional confirmation of the objectives of the work, we can mention Ellon Musk's OpenAI which is based on ANN and uses the same approach to learning AI, which was used in the example of the video game shown in the paper. The basic rule is that there are no game rules, only monitoring (input/visual data that players would have).

It is clear that the current 3D video games are more complex than Super Mario, they have more variables, but equal freedom for creativity (all have rules that keep them within certain limits, but within them a high level of freedom). Such complex and diverse behaviours can hardly be described in lines of code, linear algorithms, behaviour trees, or similar methods. However, focusing on ANN design can influence the complexity of ANN, the speed of learning, and the behaviour of video game actors, as can be seen from the D and X-Y experiments, where players are seen to learn and adapt faster in the X-Y version.

Neural networks can be used in many different ways in video games, from agent control, environmental evolution to content generation. As stated in the paper, monitoring the process of neural network development, for example, can be a very useful process. Despite proving sustainable for agent design, there are still many unexplored directions of application and use of neural networks in video games, especially in 2D and 3D graphics environments. The potential is that neural networks can generate entire worlds or complex video games based on the preferences of human players.

7. FERENCES

[1] J. McCarthy, "Professor John McCarthy - Artificial Intelligence", Jmc.stanford.edu, 2022. [Online]. Available: http://jmc.stanford.edu/artificial-intelligence/index.html. [Accessed: 29- Mar- 2022].

[2] "Artificial Intelligence in Video Games", Medium, 2022. [Online]. Available: https://towardsdatascience.com/artificial-intelligence-in-video-games-3e2566d59c22. [Accessed: 29- Mar- 2022].

[3] "How AI changed (and is changing) game development", HPE, 2022. [Online]. Available: https://www.hpe.com/us/en/insights/articles/6-ways-ai-is-making-an-impact-on-video-games-1907.html. [Accessed: 29- Mar- 2022].

[4] 2022. [Online]. Available: https://wiki.pathmind.com/neural-network. [Accessed: 29- Mar- 2022].

[5] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity", The Bulletin of Mathematical Biophysics, vol. 5, no. 4, pp. 115-133, 1943. Available: 10.1007/bf02478259.

[6] D. Nguyen and B. Widrow, "Neural networks for self-learning control systems", IEEE Control Systems Magazine, vol. 10, no. 3, pp. 18-23, 1990. Available: 10.1109/37.55119.

[7] K. Kapp, L. Blair and R. Mesch, The gamification of learning and instruction field book. San Francisco, CA: Wiley & Sons, 2014.

[8] A. Jain, Jianchang Mao and K. Mohiuddin, "Artificial neural networks: a tutorial", Computer, vol. 29, no. 3, pp. 31-44, 1996. Available: 10.1109/2.485891.

[9] P. Picton, Neural networks. Basingstoke: Macmillan, 2000., pp. 29.

[10] S. Gallant, "Perceptron-based learning algorithms", IEEE Transactions on Neural Networks, vol. 1, no. 2, pp. 179-191, 1990. Available: 10.1109/72.80230.

[11] S. Dasgupta, A. T. Kalai, C. Monteleoni, "Analysis of Perceptron-Based Active Learning", Journal of Machine Learning Research, vol 10, pp. 281-299, 2009.