# M-net: A Convolutional Neural Network Based in Reinforcement Learning to Play Mario Bros

M-net: Una Red Neuronal Convolucional Basada en Aprendizaje por Refuerzo Utilizada Para Jugar Mario Bros

Uriel Calderon-Uribe[1], Salvador Calderon-Uribe[2], Esau Gonzalez-Villagomez[3], Jacob Gonzalez-Villagomez[3]

[1]Instituto Tecnológico Superior del Sur de Guanajuato, Educación Superior 2000, Benito Juárez, 38980 Uriangato, Guanajuato, México.
[2]Mechatronics / Engineering Faculty, Autonomous University of Queretaro, San Juan del Rio, Queretaro, Mexico.
[3]Departamento de Estudios multidisciplinarios, División de Ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato, Yuriria 38940, Guanajuato, México.
u.calderon@itsur.edu.mx1, scalderon02@alumnos.uaq.mx2, esau.gonzalez@ugto.mx3, jacob.gonzalez@ugto.mx3

## Abstract

Nowadays, Deep Reinforcement Learning (DRL) has made great progress since it was first proposed. Generally, DRL-based models take large amounts of data and make decisions according to the policy established by these models. This learning method updates the policy and maximizes the reward so that the agent can perform a task optimally. In this work, an agent is trained in order to learn how to play video games. Deep Q-learning (DQL) methods that explain the behavior and decision-making generated by the agent will be discussed. First, the agent is trained through a convolutional model to understand the environment it is learning. From this understanding, the agent makes decisions in order to maximize the reward generated by the model. Finally, the agent is put to the test in the Super Mario Bros environment in order to complete the level. The results of this work show how the agent is able to learn in the environment maximizing the reward.

***Keywords:*** Reinforcement learning, Convolutional neural network, Deep learning

## Introduction

Reinforcement learning (RL) in video games is a long-standing research area. The main goal is to train an agent in such a way that it can perform like a human (Shao, K. et al.). That is, it studies the complex interaction between agents and environments. Different video game environments represent a perfect problem for RL research. These environments provide enough information to test algorithms based on machine learning. This characteristic makes video games an environment for RL research.

Broadly speaking, the development of an intelligent agent involves perception and decision-making in the game environment. With these characteristics, there are different challenges to overcome. First, some environments provide a large state space, usually in strategy games. The second challenge is that learning an appropriate policy in an unknown environment is difficult. To solve these problems, data-driven methods such as convolutional neural networks (CNN) have emerged to overcome these challenges.

In recent years, RL has made great strides not only in the area of video games but also in the areas of computer vision and natural language processing (Torrado et al.). The addition of neural networks (including convolutional neural networks) has allowed new models to perform better in high-dimensional environments. In addition, advances have been made in generalization and scalability compared to traditional RL techniques (Alonso et al.). In the last decade, DRL has made great progress in video games, including Atari, ViZDoom, and Starcraft, among others. Different authors present advances in this field, for example; in (Emigh et al.) they survey the development of different DRL methods and test them on Alpha Go and Alpha Go Zero. In (Zhao et al.) different RL techniques are tested and compared with evolutionary and hybrid approaches. Arulkumara et al. describe the elements that make up DRL and discuss their applications and important mechanisms. The main contribution of this work is to give a detailed explanation of the process that exists when training an agent and performing a complex task.

The organization of this paper is described as follows. Section II describes the methods used to prepare and train the agent in the Mario Bros. environment. Section III describes the results and considerations of this work. Finally, in Section 4 a brief conclusion about the work is described.

## 2. Methodology

This section describes the methods used to train, validate, and measure the performance of the agent in the Super Mario Bros environment. Figure 1, shows the process used to validate the final model. From this figure, it can be observed that the process consists of four phases. In the first phase, a set of images is obtained from the environment. These images are used to inspect the action taken by the agent. In the second phase, the images are preprocessed to fit in the convolutional neural network model (CNN). Subsequently, the model is trained to predict the best decision according to the images. Finally, the model is evaluated to measure the performance in the Mario Bros environment.

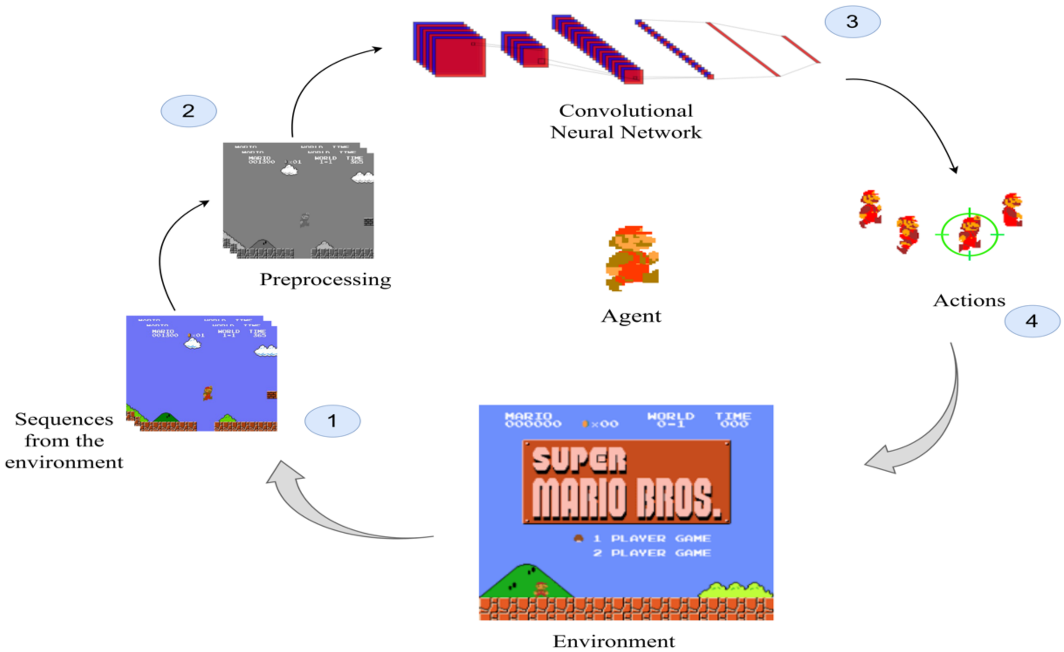


*Figure 1: Overall description to teach agent to play Mario Bros.*
*Fuente: Elaboración propia*

### 2.1 Image acquisition

In RL, acquiring information from the environment is an important step as it allows the agent to obtain data from the environment. In this case, the environment provides information that the agent uses to create its own set of rules or policies. These rules are based on maximizing the reward obtained through the interaction with the environment. Thus, it is important to model the environment correctly because it can generate inconsistencies when taking actions and receiving rewards. To avoid these problems, the OpenAI gymnasium library is used to generate the Mario Bros environment (Szita et al., R. Dechter et al). This library allows one to model different environments quickly and accurately, without the need to use many computational resources. Thus, level 1-1 of Mario Bros is used as an environment to train the agent. Figure 2 shows a set of images of this environment.

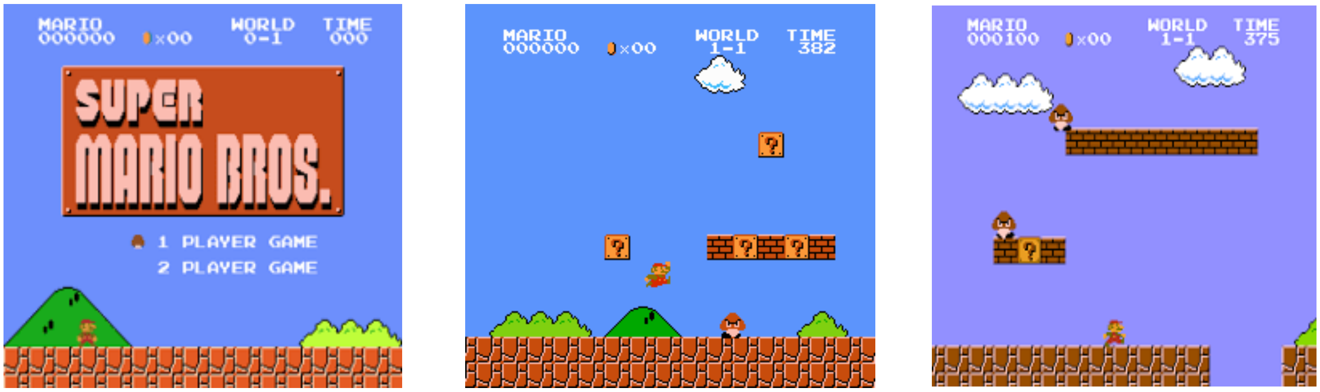


*Figure 2: Super Mario Bros environment*
*Fuente: Elaboración propia*

## 2.2 Image Preprocessing

In machine learning, image preprocessing is defined as setting the image properties to fit in the CNN (Jaderberg, M. et al. Matsuo et al.). The most common technique used in image preprocessing is grayscale transform. The grayscale transform consists of modifying the color channels (most commonly the RGB) to one channel (grayscale). Thus, the grayscale can be defined according to the equation (1).

$$I_g = 0.2126R + 0.7152G + 0.0722B \quad (1)$$

Where $R$ represents the matrix with the red channel in the image, $G$ represents the green channel in the image, $B$ represents the blue channel in the image, and $I_g$ represents the grayscale image. Figure 3 shows a set of images in grayscale format.

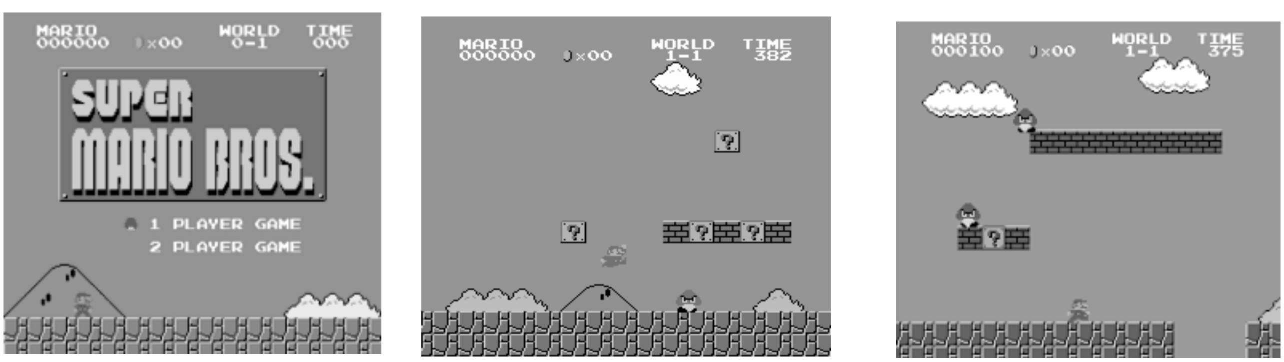


*Figure 3: Image preprocessing applied to the Super Mario Bros environment.*
*Fuente: Elaboración propia*

## 2.3 Convolutional neural network training

In the RL process, CNN are used as agents to predict the next state in a specific environment. These predictions are recollected according to exploitation and exploration. Exploitation allows to use of the action with a higher reward, while exploration is used to test different actions in the environment (Jaderberg et al. Osés et al.). Making these procedures allows to the CNN set the weights to converge in the optimal solution. The optimal solution is reached according to the loss function. However, the traditional loss function does not extract the relevant features therefore reaching an optimal solution is complex. For this reason, a modified loss function is used to correct these challenges. Equation 2, describes the loss function used to train the CNN model.

$$H(Q,P) = -\sum_i Q(x) log((P(x_i))) \quad (2)$$

Where Q(x) represents the predicted actions, P(x) represents the target (empirical) distribution. Once the loss function, a CNN needs to be constructed. Figure 4, shows the structure used to allow the agent to play Super Mario Bros.

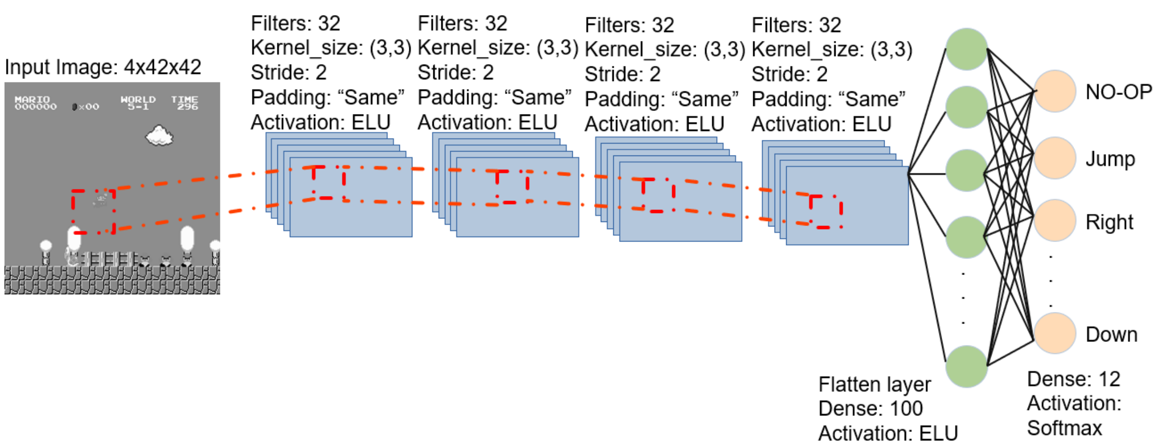


*Figure 4: Proposed model agent used to play Super Mario Bros.*
*Fuente: Elaboración propia.*

According to the figure, the model contains an input image with a size of 42x42 in a grayscale image. Subsequently the model counts with 5 convolutional layers, with 32 filters, a kernel size of (3, 3), strides of 2, and padding 1. Finally, two linear layers are used to represent the set of actions that allow the agent to succeed in the environment.

## 3. Results

To train the model, a computer with an AMD Ryzen 5600X and 16GB of RAM where used. Once trained, the length episode and the time of the train were used to measure the performance of the agent. Figure 5 shows the performance obtained in 50 minutes of training.

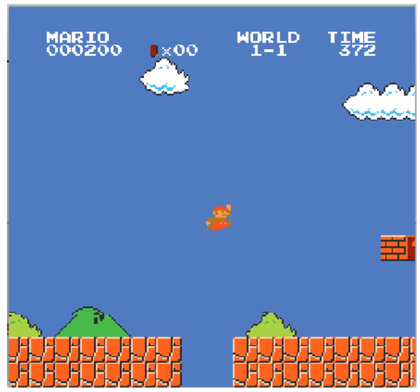


*Figure 5: Episode length reached by the agent in 50 minutes of training.*
*Fuente: Elaboración propia.*

Finally, Figure 6 shows the performance of the agent to avoid a chasm.

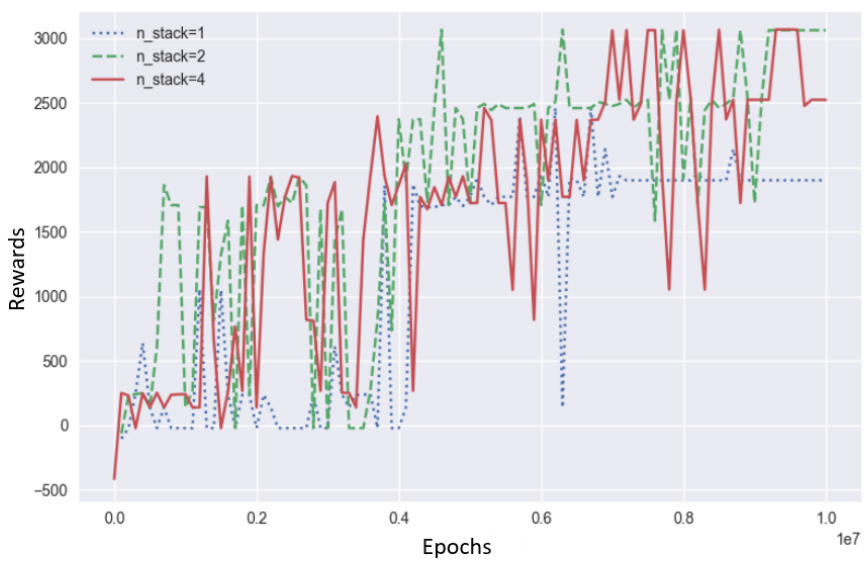


*Figure 6: The agent trained only from instrinsic rewards succesfully jumping over a chasm.*
*Fuente: Elaboración propia*

## 4. Conclusions

In this work, an agent based on reinforcement learning is trained to surpass the Super Mario Bros environment. The model is based on 4 phases; image acquisition, image preprocessing, CNN training, and predictions. In the first stage, a set of images are obtained from the environment. Subsequently, these images are preprocessed to fit in the CNN. In the third stage, the CNN model is trained according to the preprocessed images. Finally, the model predicts the best actions from a set of actions established by the environment. The results obtained in this project show that the agent can successfully overcome the challenges of the Super Mario Bros environment. Furthermore, these results underscore the robustness and versatility of CNNs in adapting and improving their performance as they face more complex situations. Future works are focused on training an agent to perform agriculture control.

## Bibliography

Alonso, E., Peter, M., Goumard, D., & Romoff, J. (2020). Deep reinforcement learning for navigation in AAA video games. arXiv preprint arXiv:2011.04764.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine, 34(6), 26-38.

Emigh, M. S., Kriminger, E. G., Brockmeier, A. J., Príncipe, J. C., & Pardalos, P. M. (2014). Reinforcement learning in video games using nearest neighbor interpolation and metric learning. IEEE Transactions on Computational Intelligence and AI in Games, 8(1), 56-66..

Jaderberg, M., Czarnecki, W. M., Dunning, I., Marris, L., Lever, G., Castaneda, A. G., ... & Graepel, T. (2019). Human-level performance in 3D multiplayer games with population-based reinforcement learning. Science, 364(6443), 859-865.

Matsuo, Y., LeCun, Y., Sahani, M., Precup, D., Silver, D., Sugiyama, M., ... & Morimoto, J. (2022). Deep learning, reinforcement learning, and world models. *Neural Networks*, *152*, 267-275.

Shao, K., Tang, Z., Zhu, Y., Li, N., & Zhao, D. (2019). A survey of deep reinforcement learning in video games. arXiv preprint arXiv:1912.10944.

Szita, I. (2012). Reinforcement learning in games. In Reinforcement Learning: State-of-the-art (pp. 539-577). Berlin, Heidelberg: Springer Berlin Heidelberg.

Osés Laza, A. (2017). Reinforcement learning in videogames (Bachelor's thesis, Universitat Politècnica de Catalunya).

R. Dechter, "Learning while searching in constraint-satisfaction-problems," pp. 178–183, 1986.

Torrado, R. R., Bontrager, P., Togelius, J., Liu, J., & Perez-Liebana, D. (2018, August). Deep reinforcement learning for general video game ai. In 2018 IEEE Conference on Computational Intelligence and Games (CIG) (pp. 1-8). IEEE.