Bulbasaur, Charmander or Squirtle: An Application of Artificial Neural Networks for Pattern Recognition

Alfa M. D. Ferreira
Institute of Technology - ITEC
Federal University of Pará - UFPA
Belém - PA, Brazil
alfamarine.ferreira@gmail.com

Lidia G. S. da Rocha
Institute of Technology - ITEC
Federal University of Pará - UFPA
Belém - PA, Brazil
lidiagianne@gmail.com

***Abstract*—This paper presents the use of an artificial neural network in pattern recognition. To validate its applicability, the determination of the starter pokémon based on the personality of the player was used as a theoretical problem. To solve the problem, a multi-layered neural network using a RPROP learning algorithm was developed.**

***Keywords*-Neural Network; Pattern Recognition; Pokémon**

I. Introduction

Created in 1996 by Satoshi Tajiri, Pokémon has become one of the most successful game based franchise on the planet. Beginning as a software made to be played on Game Boy, quickly the franchise has diversified into comics, television show, movies, card game, stickers, toys and ancillary products due to the audience appreciation of the story [1].

Based on Tajiri's hobby for collecting insects, Pokémon idealizes a world where wild creatures exist to be collected, trained and to battle with each other. The idea came in 1991, when the creator imagined an insect moving through the Game Boy game link or, as it is done in the game, pokémon being exchanged between players. After five years of development, the launch of Pokémon revived the technology of the Game Boy and later expanded to the rest of the planet, marking the culture of generations. In November 2000, it even happened a two-day conference entitled "Pikachu's Global Adventure", sponsored by the Japanese Studies Center of the University of Hawaii and focused on the discussion of this worldwide phenomenon and its impacts [2].

In terms of videogame, beyond the pokémon permutation aspect, the game has a battle mechanism based on the type of creature the player possesses, in which one type has both weaknesses and strengths over other types, making the battle mechanism more complex than just using the stronger pokémon [3]. Therefore, the types used will influence the journey that the player will make, becoming an essential strategic factor. This strategy starts early in the game, where the player needs to choose between three types of pokémon, being these types fire, water and grass. There is no accurate method for choosing the starter pokémon, leaving it in charge of the player's identification with one of the available types.

This paper will show a use of artificial neural networks to determine which starter pokémon the player should choose, based on his personality and based on the first generation of the game franchise.

II. The Pokémon World

The name "Pokémon" is derived from the abbreviation of "pocket monsters" and is used to designate the creatures present in this world [2].

Inside the Pokémon mythology, this world has locations (regions) which differ mainly by the variety of pokémon that they harbor. Each generation in the franchise of games is situated in one of these regions, possessing a set of different creatures. Within this set, there are the so-called starter pokémon, or starters, which are available for player selection early in their journey.

The first generation of games, located in the Kanto region, has as starters pokémon Bulbasaur, Charmander and Squirtle. Traditionally, the starters are fire, water or grass type, but the current total set of creatures varies by up to 18 types. This variaty of types will influence the battle mechanisms inside the game.

Bulbasaur, Charmander and Squirtle are, respectively, the starters of grass, fire and water types presented in the first generation of games. In addition to the type of power they present, the television series showed that they also differentiate themselves by personality, and this can be a criterion of choice for the player.

However, the correlation between the characteristics of the player and the pokémon's personality does not consist in an easily modeled mathematical function, leaving it to other techniques to infer this model.

To solve the problem of determining the starter pokémon, or the problem of classification of patterns to determine which pokémon is the most compatible with the player, it will be used a computational intelligence technique: artificial neural networks.

That technique was chosen because, among the existing computational intelligence techniques, it's one of the most successful in the context of pattern recognition, presenting a significantly compact and faster performance to evaluate models [4].

III. Artificial Neural Networks

Artificial neural networks, or simply neural networks, are defined as parallel distributed processors consisting of simple processing units, called neurons, whose operation is designed to model how the brain performs a particular task or function of interest [5]. This modeling similar to brain functioning is given by the ability of the neural network to acquire and retain knowledge, this acquisition being made from learning processes and retention made from synaptic weights [6].

The learning process is done through a learning algorithm and it will have the function of modifying the synaptic weights in order to make the network reach the desired goal.

Among the possibilities that neural networks provide from their parallel distributed structure and ability to learn, it has the ability to classify patterns from input-output values. These input-output sets will consist of training samples used in supervised learning.

The learning algorithm will use the training samples to modify the synaptic weights, bringing the network closer to the determined function. In the case of neural networks applied in pattern recognition, the learning process will allow the neural network to classify the inputs according to a set of pre-established classes.

Given this use of artificial neural networks, it will be used to solve the proposed problem of determining which starter pokémon will be suitable for a given input set.

IV. Neural Network Configuration

As seen in the previous topic, a neural network will be used to classify a given input set. In the present topic will be presented the configurations used for the construction of the neural network.

A. Structure of an artificial neural network

Neural networks are composed of nodes (neurons) connected by directed connections. These connections serve to propagate the input set across the network layers until an output is calculated. Each connection has a numerical weight (synaptic weight) associated with it, wich determines the strength and signal of the connection.

The neurons that aren't in the input layer, in turn, calculate the weighted sum of their inputs, which will be the junction of inputs added to a fictitious fixed value input wich is equal to 1. These inputs will be weighted by the values of the weights of the connections. In particular, the fictitious input will be connected by a weight called bias.

Once the weighted sum is calculated, an activation function is applied to this value in order to obtain the output of the neuron. This activation function is usually non-linear and differentiable.

B. Connection between neurons

The connection between the neurons can be made in two distinct ways.

The first way is through connections that have only one direction, in which each node receives the input of nodes "above" them and releases its outputs to nodes "below". Networks with such a form of connection are known as feedforward networks.

In the second way, the network feeds its outputs back to its own inputs. The network response to a particular input will depend on both the current input and the previous inputs. This type of network is called a recurring network.

This work will use a feedforward network.

C. Layers of neural networks

Feedforward networks are usually arranged in the form of layers, in which each unit receives input only from units in the layer immediately prior to them. In relation to the amount of layers a network has, we can classify them into two types: single-layer neural networks and multi-layered neural networks.

Single-layer neural networks, or perceptrons, are characterized by having their inputs directly connected to the outputs, these outputs being the only network processing layer. Such networks can solve only linearly separable problems.

In order to solve non-linearly separable problems, the multi-layered neural networks were created. This type of network, also known as multilayer perceptron, has more than one processing layer in addition to the output layer.

D. Learning algorithm

A property of prime importance for the operation of a neural network is the ability to learn from its environment and to improve performance through this learning process. Ideally, a network becomes more educated about its environment after each iteration of the learning process, which will adjust the synaptic weights and bias levels until a certain parameter of evaluation is reached.

Learning algorithms have been responsible for solving several difficult problems on the part of neural networks, which can be of different types, each offering specific advantages. The learning algorithms differ from each other by how the adjustment is made of the synaptic weights of a neuron.

The most popular learning algorithm for multi-layer perceptrons training is the error backpropagation algorithm. This algorithm is based on the error correction learning rule, which makes the adjustment of the synaptic weights of a neuron in a proportional way to the product of the error signal by the synapse input signal.

The backpropagation algorithm has two steps through the different layers of the network, the first step being propagation, in which the input is applied to the sensory nodes of the network and its effect propagates through the layers of the network to generate an output, and the second step is backpropagation, in which the synaptic weights are adjusted according to an error correction rule which is propagated behind the network.

The error correction rule used is the gradient-descent in which the new synaptic weights will be given by the difference between the current values and the partial derivative of the mean square error by the synaptic weight.

Many variations of the backpropagation algorithm were proposed to improve its performance. The learning algorithm used in this work was Resilient Backpropagation, also known as RPROP, a method derived from the original backpropagation algorithm.

RPROP is defined as a local adaptive learning algorithm. It differs from the original backpropagation algorithm because it considers only the gradient-descent signal in the error calculation. This feature makes your convergence faster [7].

Along with the gradient signal, a η factor is considered in the calculation of the new values of synaptic weights. For an iteration, if there is a signal change between the current gradient-descent and the previous iteration, the η value will be $\eta^- = 0,5$. If there is no change of signal, the value of η will be $\eta^+ = 1,2$.

V. Neural Network Implementation

To solve the problem proposed, the language R, in its version 3.5.1, was used with the help of the *neuralnet* library, created by Stefan Fritsch [8].

R is a programming language and freely available software environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques. The R language is widely used among statisticians and data miners for developing statistical software and data analysis.

The mode of implementation of the neural network will be described in the following subsections.

A. *Characteristics of the neural network implemented*

As mentioned in the previous topic, the neural network proposed to solve the problem will be multi-layered with RPROP learning algorithm. In order to generate the training data set, an empirical method of output pattern definition was used based on the answers given to a set of questions that had as objective to acquire the characteristics of the inputs.

A set of eight questions was made, each containing three possible answers. The set of questions and answers had subjective nature, making the process of data acquisition in the form of a personality quiz. A graphic interface in brazilian portuguese is being made to implement the quiz. The Fig. 1 shows one of the questions used to obtain the input data, already in the proposed design of the graphic interface.

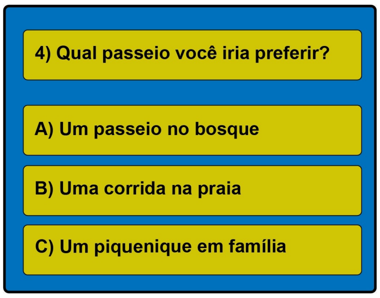

Figure 1. Example of question used to obtain input data.

From the input values, given in the form of a line vector for each example, the network should generate an output between three possible ones, these being the three first generation starter pokémon (Bulbasaur, Charmander and Squirtle) or, numerically, a vector with three columns such that only one of the components of the vector will assume the value 1 while the others will be null. The first column of the vector will correspond to the bulb output, the second column will correspond to the charm output, and the third column will correspond to the squir output. Fig. 2 shows one of the possible outputs obtained from the network, also in the proposed design of the graphic interface.

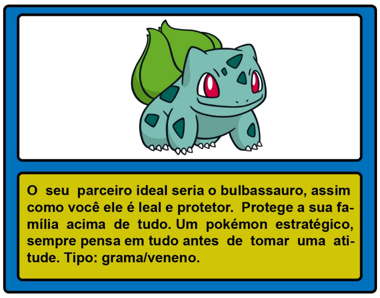

Figure 2. Example of output.

For the training of the network a set with 200 input-output values was used, and for the network test it was used a set with 76 input values.

B. *Neuralnet library*

The *neuralnet* library has as main function *neuralnet*() and it aims to create the neural network itself.

The *neuralnet()* function has as input the relevant parameters for the network, such as the estimated formula of the problem, the dataset, the amount of hidden layers and the number of neurons that each layer contains, the threshold value, the maximum steps for the training of the neural network, etc. In this work, it will be created a 8-5-3 multilayer perceptron with the *neuralnet()* fuction. The threshold value used was $1 \cdot 10^{-3}$ and the maximum steps value used was $1 \cdot 10^{5}$.

VI. Results

The implemented neural network had as outputs an image of the network architecture used, as shown in Fig. 3, and the confusion matrices for the general output, as shown in Fig. 4.

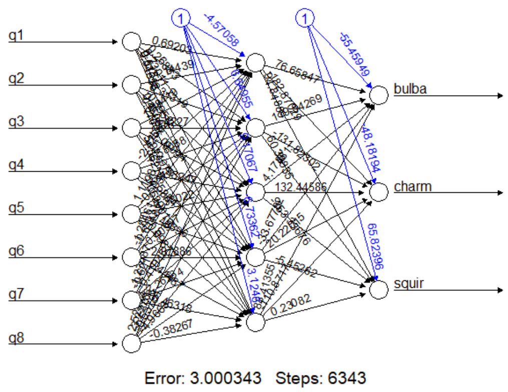

Figure 3. Neural network architecture.

```
                resultado.rede
resultado.esperado   0   1
                 0 150   2
                 1   3  73

          bulba.rede
bulba.esperado  0  1
             0 20  0
             1  3 53

          charm.rede
charm.esperado  0  1
             0 63  2
             1  0 11

          squir.rede
squir.esperado  0  1
             0 67  0
             1  0  9
```

Figure 4. Obtained confusion matrices.

These results were obtained after several runs of the neural network algorithm. To choose the best neural network configuration obtained, the error function was used as metric. The error function is given by the sum of squared errors.

The best neural network obtained had an error $E = 3.000343$ during the training phase, as it is verified in Figure 3.

To validate the neural network, confusion matrices were used as performance metric. In the test phase, as indicated by the confusion matrices, the accuracy values presented in Table I were obtained.

TABLE I. Neural Network Accuracy

	General output	Bulbasaur	Charmander	Squirtle
Negatives	98,68%	100%	96,92%	100%
Positives	96,05%	94,64%	100%	100%

VII. Conclusion

The implemented neural network was able to solve the proposed problem and, although the error in the training phase was relatively large, the success rate was satisfactory, with an average accuracy of 98.29%.

It was verified that the neural network implementation was simple, provided by the use of the R language and the *neuralnet* library. However, due to the amount of intermediate layers and the required threshold, the network showed a slow performance. It is advisable for future work using a more balanced number of neurons and intermediate layers to achieve better network performance. It is also expected for future work the implementation of networks for the other generations of the Pokémon franchise.

Acknowledgement

The authors would like to thank the Faculty of Electrical and Biomedical Engineering (FEEB) of the Federal University of Pará (UFPA) for the support in this work.

References

[1] J. Tobin, *Pikachu's global adventure: The rise and fall of Pokémon*. Durham, USA: Duke University Press Books, 2004.

[2] J. Bainbridge, *'It is a Pokémon world': The Pokémon franchise and the environment*, International Journal of Cultural Studies. Thousand Oaks, USA: SAGE Journals, 2013, pp. 1-16.

[3] Eurogamer, *Pokémon Go Type chart, Type effectiveness and weakness explained: Calculate the damage for any matchup with our Type Chart*, 2018. Available at: https://www.eurogamer.net/articles/2018-01-15-pokemon-go-type-chart-effectiveness-weaknesses. Accessed: 2018 - 07 - 17.

[4] C. Bishop, *Pattern recognition and machine learning*, 1rd ed. New York, USA: Springer, 2006.

[5] S. Haykin, *Redes neurais: Princípios e práticas*, 2rd ed. Porto Alegre, Brazil: Bookman, 2001.

[6] S. Russel and P. Norvig, *Inteligência artificial*, 3rd ed. Rio de Janeiro, Brazil: Elsevier, 2013.

[7] M. Riedmiller and H. Braun, *A direct adaptive method for faster backpropagation learning: The RPROP algorithm*, Proceedings of the IEEE International Conference on Neural Networks (ICNN). San Francisco, USA: IEEE, 1993, pp. 586-591.

[8] S. Fritsch, F. Guenther, M. Suling and S. Mueller, *Package 'neuralnet'*, 2016. Available at:https://cran.r-project.org/web/packages/neuralnet/index.html. Accessed: 2018 - 07 - 17.