Neuronal network for Linux symbols

Gerardo Solano

March 2023

Abstract

This document contains the implementation of neuronal networks to recognize Linux symbols images

1 Concepts and background

First of all we need to introduce the concept of Neuronal Networks, according to Steeb(2005) "*Neural networks are models of the brain's cognitive process. The brain has a multiprocessor architecture that is highly interconnected. Neural networks have an incredible potential to advance the types of problems that are being solved by computers*".
This affirmation suggest what neuronal networks are the modeling cognitive process of the brain, which have a highly interconnected multiprocessor architecture. Also, mentioned what neuronal networks have an incredible potential to move forward in the resolution of problems what are approaching with computers.
Neural networks rely on the functioning of nerve cells in the brain, which are connected to each other to process information. By simulating this process in a mathematical model, machine learning algorithms can be designed that allow computers to process data more intelligently and perform tasks such as pattern recognition, image classification, text analysis, and much more.
The purpose of this project it's implement a prototype using open-source tools like Octave, mathematical models and programming languages like C++, all this process converge into the identification of symbols and patterns of the different Linux distributions.

2 Hopfield model

The Hopfield model is a type of artificial neural network that is commonly used for pattern recognition and classification tasks. It is based on the idea of creating a network of interconnected neurons that work together to recognize patterns and classify data. The neurons in the network are modeled after the neurons in the brain, and they are connected by synapses that transmit signals between them.

The basic idea behind the Hopfield model is to use the network to store a set of patterns that can later be used for recognition or classification tasks. To do this, the network is trained on a set of input patterns using a process known as Hebbian learning. During this process, the synapses in the network are adjusted based on the input patterns, in order to create stable patterns that can be later recognized.
Once the network has been trained on a set of patterns, it can be used to recognize or classify new patterns. To do this, the new pattern is presented to the network, and the neurons in the network work together to find the stored pattern that is closest to the new pattern. This is done by adjusting the synaptic weights in the network, which causes the neurons to activate in a way that corresponds to the stored pattern.
Configuring the Hopfield model involves setting up the model architecture, defining the training data, and specifying the parameters for the learning algorithm. Here are the basic steps:

1. Set up the model architecture: The Hopfield model is a single-layer network where all neurons are connected to each other. To set up the model, you need to create a matrix of synaptic weights that defines the connections between neurons.

2. Define the training data: The Hopfield model is an autoassociative memory, meaning it can learn to associate a pattern with itself. To train the model, you need to provide it with a set of input patterns. These patterns can be binary vectors, where each element is either -1 or 1.

3. Specify the learning algorithm: This model employs Hebbian learning, a learning rule that modifies the synaptic weights based on the correlation between the activations of the neurons in the input patterns. Several types of Hebbian learning exist, including the Storkey algorithm, which can enhance the stability and convergence of the Hopfield model.

4. Apply the model to new inputs: Once the model is trained, you can use it to retrieve a stored pattern from a partial or noisy input. This is done by iteratively updating the neuron activations until the network converges to a stable state.

This image shows a network of seven neurons that are connected to each other in a fully connected way, meaning each neuron is connected to all other neurons. The circles represent the neurons and the lines connecting them represent the synaptic connections, which are the connections between neurons that allow them to communicate with each other.
In addition to the circular neurons and connecting lines, there are also small boxes labeled with letters representing the synaptic weights, which determine the strength of the connections between neurons. The arrows pointing to the boxes indicate the direction of the connection and the numerical values inside the boxes represent the strength of the connection.
Example:

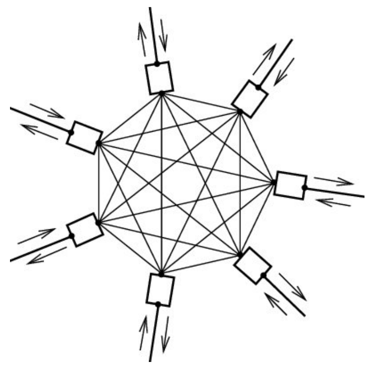

Figure 1: Graph emulating a neuronal network

3 Conclution

In conclusion, the implementation of neural networks for pattern recognition and classification tasks, such as identifying symbols and patterns of different Linux distributions, can greatly enhance the capabilities of computers. The Hopfield model, in particular, is a commonly used type of artificial neural network that relies on the interconnectivity of neurons to recognize patterns and classify data. The configuration of the Hopfield model involves setting up the model architecture, defining the training data, and specifying the parameters for the learning algorithm. By using open-source tools like Octave, mathematical models, and programming languages like C++, the implementation of a prototype for this purpose is possible. The potential applications of this technology are vast and can greatly benefit various fields, including image classification, text analysis, and more.

4 References

Steeb, W. H. (2010). The NonLinear Workbook (3rd ed.). World Scientific Publishing.