

# Chapter V

## Simulation of the Action Potential in the Neuron's Membrane in Artificial Neural Networks

**Juan Ramón Rabuñal Dopico**  
*University of Coruña, Spain*

**Javier Pereira Loureiro**  
*University of Coruña, Spain*

**Mónica Miguélez Rico**  
*University of Coruña, Spain*

### ABSTRACT

*In this chapter, we state an evolution of the Recurrent ANN (RANN) to enforce the persistence of activations within the neurons to create activation contexts that generate correct outputs through time. In this new focus we want to file more information in the neuron's connections. To do this, the connection's representation goes from the unique values up to a function that generates the neuron's output. The training process to this type of ANN has to calculate the gradient that identifies the function. To train this RANN we developed a GA based system that finds the best gradient set to solve each problem.*

### INTRODUCTION

Due to the limitation of the classical ANN models (Freeman, 1993) to manage time problems, over the year 1985 began the development of recurrent models (Pearlmutter, 1990) capable to solve efficiently this kind of problems. But this situation didn't change until the arrival of the Recurrent Backpropagation algorithm. Before

this moment, the more wide used RANN were Hopfield networks and Boltzman machines that weren't effective to treat dynamic problems. The powerful of this new type of RANN is based on the increment of the number of connections and the whole recursivity of the network. These characteristics, however, increment the complexity of the training algorithms and the time to finish the convergence process. These problems have slow

down the use of the RANN to solve static and dynamic problems.

However, the chances of RANN are very big compared to the powerful of feedforward ANN. For the dynamic or static pattern matching, the RANN developed until now offer a better performance and a better learning skill.

Most of the studies that have already been done about RANN, have been center in the development of new architectures (partial recurrent or with context layers, whole recurrent, etc.) and to optimize the learning algorithms to achieve reasonable computer times. All of these studies don't reflect changes in the architecture of the process elements (PE) or artificial neurons, that continue having an input function, an activation function and an output function.

The PE architecture has been modified, basing our study in biological evidences, to increment the RANN powerful. These modifications try to emulate the biological neuron activation that is generated by the action potential.

The aim of this work is to develop a PE model with activation output much more similar to the biological neurons one.

## **BACKGROUND**

### **Artificial Neural Networks**

An Artificial Neural Network (ANN) (Lippmann, 1987; Haykin, 1999) is an information-processing system that is based on generalizations of human cognition or neural biology and they are electronic or computational models based on the neural structure of the brain. The brain basically learns from experience. An Artificial Neural Network consists on various layers of parallel processing elements or neurons. One or more hidden layers may exist between the input and the output layer. The neurons in the hidden layer(s) are connected to the neurons of a neighboring layer by weighting factor that can be adjusted during the training

process. The ANN's are organized according to training methods for specific applications.

There are two types of ANN's, the first one with only feed forward connections is called feed forward ANN, and the second one with arbitrary connections without any direction, are often called Recurrent ANN (RANN). The most common type of ANN consists on different layers, with some neurons on each of them and connected with feed-forward connections and trained with the back propagation algorithm (Johansson et al., 1992).

The numbers of neurons contained in the input and output layers are determined by the number of input and output variables of a given problem. The number of neurons of a hidden layer is an important consideration when solving problems using multilayer feed-forward networks. If there are fewer neurons within a hidden layer, there may not be enough opportunity for the neural network capture the intricate relationships between the inputs and the computed output values. Too many hidden layer neurons not only require a large computational time for accurate training, may also result in overtraining situation (Brion et al., 1999). A neural network is said to be "overtrained" when the ANN focuses on the characteristics of individual data points rather than just capturing the general patterns in the entire training set. The optimal number of neurons in a hidden layer can be estimated as two-thirds of the sum of the number of input and output neurons.

An ANN has a remarkable ability to derive meaning from complicated or imprecise data. The ANN can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. "Training" of an ANN model is a procedure by which the ANN repeatedly processes a set of test data (input-output data pairs), changing the values of its weights according to a predetermined algorithm in order to improve its performance. Backpropagation is the most popular algorithm for training feed-forward ANN's (Lippman,

18 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: [www.igi-global.com/chapter/simulation-action-potential-neuron-membrane/4973](http://www.igi-global.com/chapter/simulation-action-potential-neuron-membrane/4973)

## Related Content

---

### Load Balancing for the Dynamic Distributed Double Guided Genetic Algorithm for MAX-CSPs

Sadok Bouamama, Khaled Ghedira and Nisrine Zaier (2010). *International Journal of Artificial Life Research* (pp. 68-86).

[www.irma-international.org/article/load-balancing-dynamic-distributed-double/49684](http://www.irma-international.org/article/load-balancing-dynamic-distributed-double/49684)

### A Hardware Immune System for MC8051 IP Core

Xin Wang, Wenjian Luo, Zhifang Li and Xufa Wang (2009). *Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies* (pp. 322-339).

[www.irma-international.org/chapter/hardware-immune-system-mc8051-core/19651](http://www.irma-international.org/chapter/hardware-immune-system-mc8051-core/19651)

### Mathematical Model to Assess the Relative Effectiveness of Rift Valley Fever Countermeasures

Holly Gaff, Colleen Burgess, Jacqueline Jackson, Tianchan Niu, Yiannis Papelis and David Hartley (2011). *International Journal of Artificial Life Research* (pp. 1-18).

[www.irma-international.org/article/mathematical-model-assess-relative-effectiveness/54746](http://www.irma-international.org/article/mathematical-model-assess-relative-effectiveness/54746)

### Design of Globally Robust Control for Biologically-Inspired Noisy Recurrent Neural Networks

Ziqian Liu (2011). *System and Circuit Design for Biologically-Inspired Intelligent Learning* (pp. 116-135).

[www.irma-international.org/chapter/design-globally-robust-control-biologically/48893](http://www.irma-international.org/chapter/design-globally-robust-control-biologically/48893)

### Automatic Generation of Synsets for Wordnet of Hindi Language

Priyank Pandey, Manju Khari, Raghavendra Kumar and Dac-Nhuong Le (2018). *International Journal of Natural Computing Research* (pp. 31-47).

[www.irma-international.org/article/automatic-generation-of-synsets-for-wordnet-of-hindi-language/209449](http://www.irma-international.org/article/automatic-generation-of-synsets-for-wordnet-of-hindi-language/209449)