McGILL UNIVERSITY

PROJECT

Something About Recurrent Neural Networks

Author:
Daniel Galeano

Supervisor:
Ioannis Psaromiligkos

A project submitted in partial fulfilment of the requirements for the degree of Master of Engineering

in the

Research Group Name
Department or School Name

October 26, 2015

Contents

List of Figures

List of Tables

List of Abbreviations

FNN	**F**eedforward **N**eural **N**etwork
RNN	**R**ecurrent **N**eural **N**etwork
BPTT	**B**ack **P**ropagation **T**hrough **T**time

List of Symbols

Chapter 1

Background Theory

1.1 Artificial Neural Networks

Artificial neural networks are parallel distributed processing systems that are composed by neurons, their most elementary building block. Neurons are processing units that generate an output according to an activation function $\sigma(x)$, which is usually a sigmoid function like in equations 1.1 and 1.2.

Logistic Regression:

$$\sigma(h) = \begin{cases} 1, h \geq 0 \\ 0, h < 0 \end{cases} \tag{1.1}$$

Hyperbolic tangent function:

$$\sigma(h) = tanh(h) \tag{1.2}$$

Neurons are interconnected by links, which have strengths that are coded by weights. A zero weight for example would indicate that absence of connection between two neurons. The link weights can also be tuned based on experience, an important property that makes neural networks adaptable and trainable.
Figure 1.1 shows a mathematical model of a neuron based on [Ke-Lin Du, 2006], with n input signals $x_1, x_2, ..., x_n$ and an output signal y. This model also shows the weights of each of the input links of this neuron as $w_1, w_2, ..., w_n$, a bias term w_0 and an activation function $\sigma(.)$. In this model x_0 is equals to 1 to include the bias term within the same vector form.
The output of this neuron is the aggregation of the input signals, shifted by the bias term and processed by the activation function; as shown in equation A.1.

$$y = \sigma(input + bias) = \sigma(W^T X) = \sigma(\sum_{i=0}^{n} w_i x_i) \tag{1.3}$$

1.1.1 Types of Neural Networks

The arrangement of synaptic links between neurons determines different types of neural networks. If the links are acyclic for example, the neural network is considered a FNN; while the presence of cyclic links indicates a RNN. The understanding FNNs is

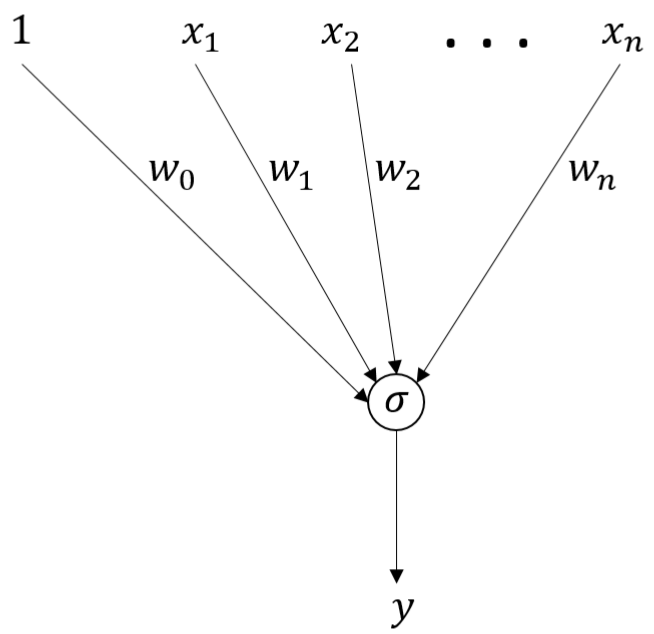

FIGURE 1.1: Mathematical Model of a Neuron.

essential to the understanding of RNNs, so a more detailed description of training and characteristics of FNNs is given in Appendix A.

1.2 Recurrent Neural Networks

Neural Networks with cyclic links are considered RNNs.

[TODO: Introduce general idea of RNNs]

1.3 Training RNNs

1.3.1 Backpropagation Through Time

BPTT is an adaption of the backpropagation method used in FNNs.

1.4 Challenges in RNNs

It has been theoretically proven that RNNs can be universal approximators of dynamic processes [KI and Y, 1993], which represents a powerful potential for modeling natural systems more accurately compared to existing methods such as FNNs. However, practical applications of RNNs have been stagnated due to theoretical and implementation challenges encountered in the this model. This section specifies some of these challenges.

1.4.1 Computational Cost

[TODO: Expand!]

1.4.2 Supervised Training

[TODO: Expand!]

Appendix A

Feedforward Neural Networks

[TODO: Introduce FFNs]

A.1 Terminology

Figure A.1 presents the terminology in a FNN with three input signals, a hidden layer with two neurons, and one output signal.

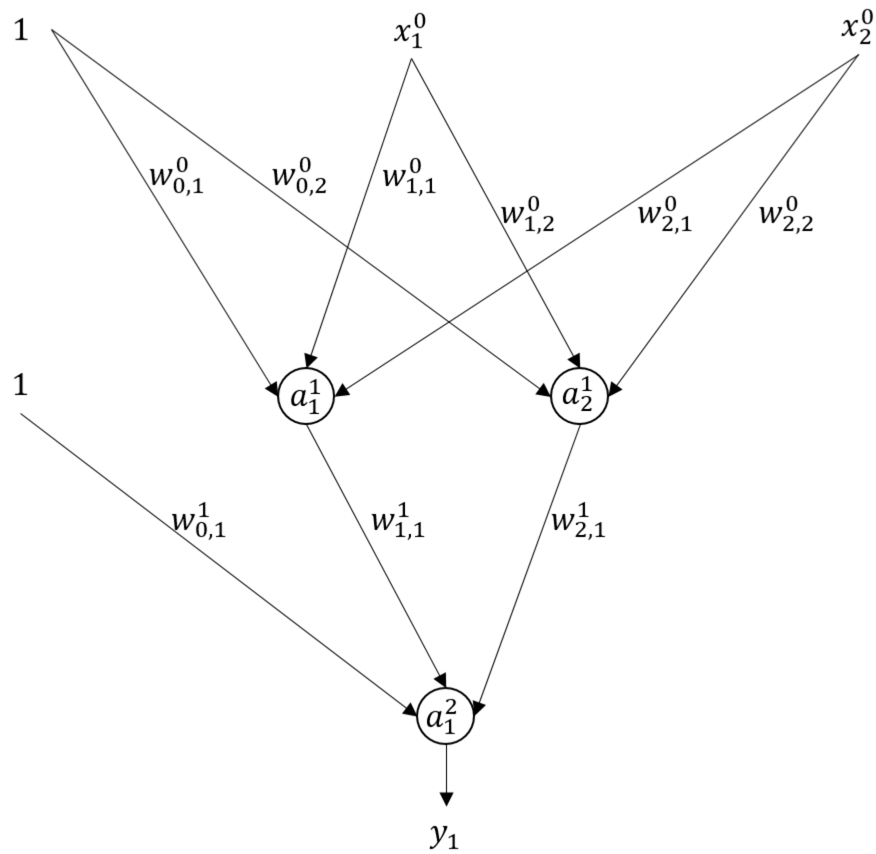

FIGURE A.1: Neural Networks Terminology

In this terminology figure w_{ij}^m represents the weight in the synaptic link that joins neuron a_i^m with neuron a_j^{m+1}. Also m indicates the layer in the neural network, where the input layer has $m = 0$, and the output layer $m = 2$. The two input signals are given by x_1^0, x_2^0, and the output signal by y_1. Even though it is not shown in the figure, the output of the hidden neurons a_1^1 and a_2^1 are x_1^1 and x_2^1 respectively.

A.2 Backpropagation Training in FNNs

Supervised training of FNNs can be done through the backpropagation method, which consists in propagating the output error with respect to the training data down all the internal layers in the network in order to adjust the weights while optimizing a model.

A.2.1 Mean Square Error

The weights in a neural network are initialized to random values, and then modified through gradient descent methods to reduce a given error function between k real (y) and predicted (y^*) output values.
The mean square is an example of an error function that can be used.

$$E = \sum_{i=1}^{k} E_i = \frac{1}{2} \sum_{i=1}^{k} (y_i^* - y_i)^2 \tag{A.1}$$

The optimization of the weights in a neural network requires the calculation of the gradient of the error function with respect to each of the weights.

[TODO: Add explanation of backpropagation training here]

Appendix B

Appendix B title

Bibliography

Ke-Lin Du, M.N.S. Swamy (2006). *Neural Networks in a Softcomputing Framework*.

KI, Funahashi and Nakamura Y (1993). “Approximation of dynamical systems by continuous time recurrent neural networks”. In: *Neural Networks*, 801–806.