Photonic Artificial Neural Networks:
All-Optical Activation Function

Davide Bazzanella

Department of Physics, University of Trento

January 17, 2018

Contents

Introduction

WHAT and WHY? BACKUP project, why silicon photonics (CMOS compatibility)

The formalization of the von Neumann architecture (1945) and the development of modern computers in the second half the 20th century allowed a great improvement in many scientific and technological areas. The technological progress led to [...]

Now, on the verge of the era of artificial intelligence (AI), many breakthroughs in complex of AIs are happening. This is mainly due to the immense amount of data fed to AI algorithms. For this reason however, the algebraic calculus which is carried out so efficiently by computers is not enough anymore.

Under the

Chapter 1

Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational systems which elaborate information likewise biological neural networks (animal brains).

ANNs can be either simulated on computers or built on specific hardware designed ad hoc. Both options can be modeled by

For simplicity in this chapter I will consider only logical models of artificial neural networks, which are therefore closer to simulated ANN than hardware ANN.

1.1 Neural Networks

A neural network is a collection of processing elements, or nodes, interconnected in an arbitrary topology. From its input nodes, the network accepts information, which will propagate into the inner nodes through the interconnections and will be elaborated at each node. At the end of the network, there will be a number of output nodes, with the task of reading a portion of the inner nodes. The inner nodes are also called hidden, because they are not meant to be accessible to the external world. A generic scheme of such network is shown in Figure 1.1 on this page.

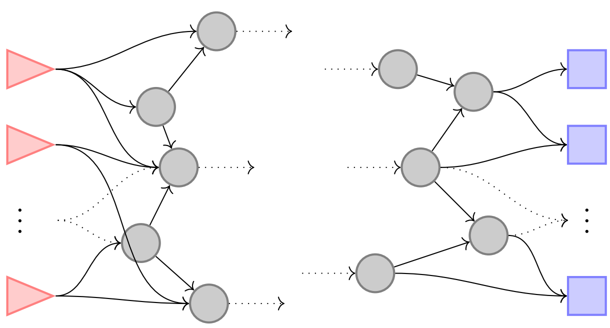

Figure 1.1: Generic scheme of a neural network. Triangles (red) are input nodes, circles (grey) are inner nodes, and squares (blue) are output nodes. Interconnections among nodes are represented by arrows: continuous when both elements are drawn, and dotted otherwise; straight between inner nodes, and curved otherwise.

Each node can operate in the same way of the others or in a completely different manner, depending on the type of neural network. The operation of nodes resembles that of animal neurons: the various input gets collected and elaborated together to obtain an output, which will become one of the many inputs for subsequent neurons. Specifically, the most used model for neurons is divided into two parts, as shown in Figure 1.2 on the current page. The first part is a weighted sum of the inputs, while the second part is given by the so called activation function.

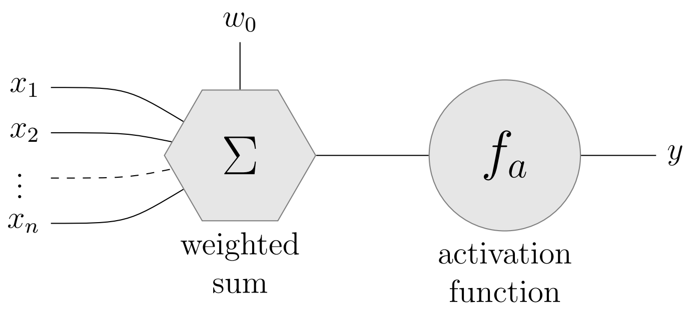

Figure 1.2: Generic node representation. x-values are inputs, y-values are outputs, w_0 is the bias.

The node is described mathematically by Equation 1.1.

$$y = f_a\left(w_0 + \sum_{i=1}^{n} w_i x_i\right) \tag{1.1}$$

Each node accepts values at its inputs and produces an output accordingly. However, the output depends also on the node's parameters: the weights and the bias, which are usually changed outside the operation phase of the neural network, as I will explain in Section 1.2 later on page 10.

Moreover it is mandatory for the activation function $f_a(\cdot)$ to be nonlinear, because otherwise a collection of nodes will result in just a weighted sum of its inputs. Two examples of nonlinear function are shown below in Figure 1.3.

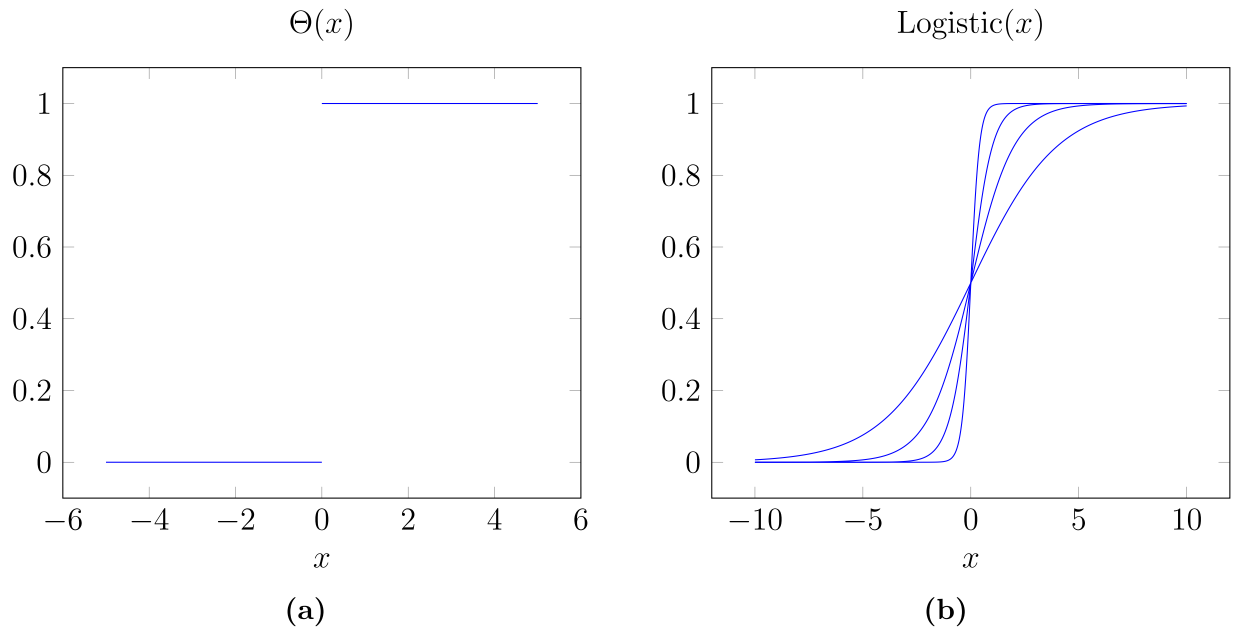

Figure 1.3: Examples of activation function: (1.3a) is the well-known step function, or Heaviside Θ, (1.3b) depicts a few functions from the family of the Logistic functions.

1.1.1 Feedforward NN

The first and most simple type of neural network is called Feedforward. In this kind of neural network, nodes are divided into groups called *layers*. A layer is a collection of nodes that accepts inputs from a preceding group and generate as many outputs as the number of nodes in the layer. Each layer of a Feedforward neural network is connected in series with the others, except of input layer at the beginning and the output layer at the end. As for the single nodes, the inner layer are called hidden, because usually not accessible.

The information travels from the input to the output and gets elaborated from each hidden layer: there are no connection between nodes of the same layer, nor loops or feedback between layers. Depending on the topology of the network, there might be more or less layers, each composed by the same or a different number of nodes. Moreover the connection between the layers might be complete, in that case the layer is said to be *fully connected*, or sparse as in the case of convolutional layers.

Perceptron

The most naive topology of a Feedforward neural network is given by the so called *Perceptron*. The Perceptron dates back to the 1957, when the homonym *Perceptron algorithm* was software implemented by Frank Rosenblatt on a computer (IBM 704) and only subsequently in hardware as the *Mark 1 perceptron*[1], [2]. The graph of such network is shown in Figure 1.4 below.

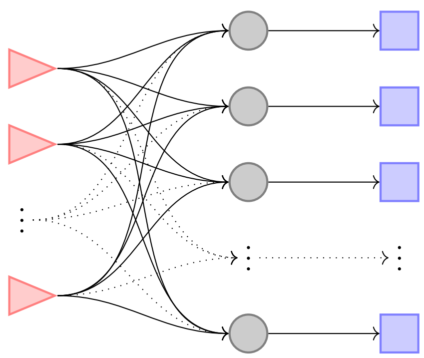

Figure 1.4: Perceptron type neural network: in this representation the perceptron has n inputs and m outputs as well as a hidden layer with m nodes.The colors, shape and styles are the same as in Figure 1.1 on page 7.

By adding more than one hidden perceptron layer to a neural network, one obtain the so called *Multi-Layer Perceptron* (MLP), shown in Figure 1.5 on the next page. The shape of the hidden layer is represented by a rectangular matrix of size $n \times m$, where n is the number of nodes in each layer and m is the number of layers. In principle other shapes are possible, i.e. each layer could have a different number of nodes. However, since such network can still be reproduced by a appropriate sized rectangular matrix of nodes, from now on I will assume the simpler rectangular form, except where otherwise specified.

Other Feedforward NNs

Autoencoder

Probabilistic

Time delay

Convolutional

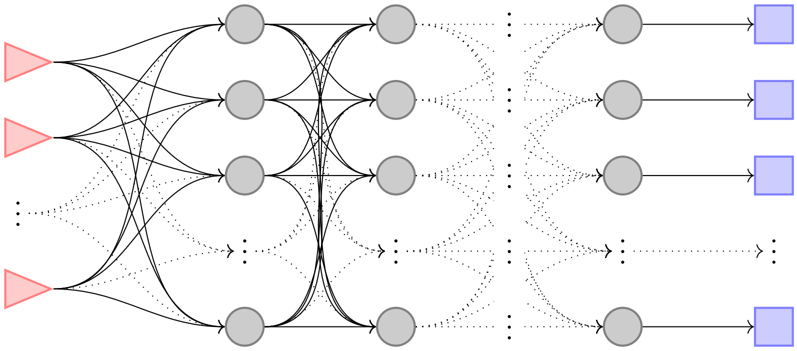

Figure 1.5: Multi-Layer Perceptron (MLP).

1.1.2 Other Types of NNs

Recurrent NN

Reservoir NN

Modular NN

Spiking NN

1.2 Working Principles of ANNs

1.2.1 Learning Process

1.3 Real-Life Examples

Chapter 2

Photonics applied to ANNs

How do I intend to create a hardware photonic ANN node?

2.1 Weighted sum of inputs

This has already been demonstrated and integrated widely, so it will not be the focus of this work.

2.2 Nonlinear activation function

On the other hand, a photonic nonlinear activation function has not yet been found. This is where the focus of my work will be.

2.2.1 Simulations

Chapter 3

Samples, setup and measurements

3.1 The samples

I could say that in the time this work was done there would have not been enough time to design and produce an ad hoc device. The aim of this thesis is to produce a proof of concept, to answer the question of feasibility.

The samples on which this work is based have been provided by the IRIS project. Specifically I had a few different structures available: from single rings resonators to the full matrix, each accessible via grating couplers. My choice was a system of intermediate complexity: a simple waveguide, coupled to eight drop channels by a single or a couple of ring resonators each. Moreover, these samples were provided with thermo-electric pads to heat the rings and effectively tune their resonance. The final choice was to study the *mini-matrix* in which the coupling mechanism was provided by single ring resonators, because it was considered simpler.

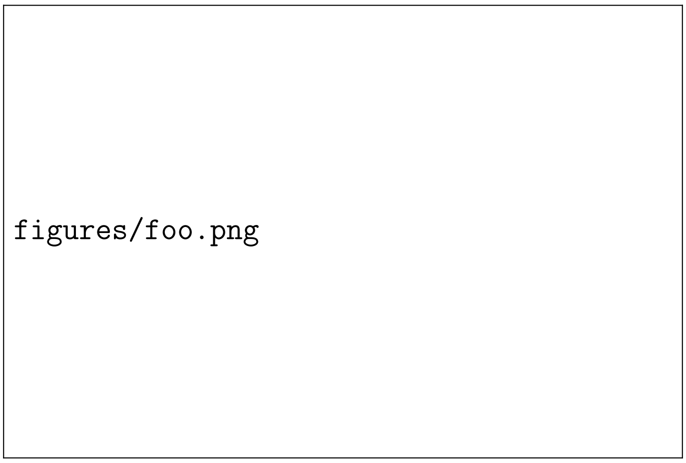

Figure 3.1: image/scheme of the minimatrix

3.2 The setup

3.3 Results

Conclusion

Bibliography

[1] R. Frank, “The perceptron a perceiving and recognizing automaton”, *tech. rep., Technical Report 85-460-1*, 1957.

[2] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in . . . ”, *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958, ISSN: 1939-1471(Electronic);0033-295X(Print). DOI: 10.1037/h0042519. arXiv: arXiv:1112.6209. [Online]. Available: http://psycnet.apa.org/journals/rev/65/6/386.pdf%7B%5C%%7D5Cnpapers://c53d1644-cd41-40df-912d-ee195b4a4c2b/Paper/p15420.

Aknowledgements