




Machine learning and FPGAs

Graduate course on Dedicated Systems
Pierangelo Calanna
Department of Mathematics and
Computer Science



Neural computation

To understand how our brain actually works

Its network reminds the style of parallel computation with adaptive connections

Solve problems by using algorithms inspired by the brain

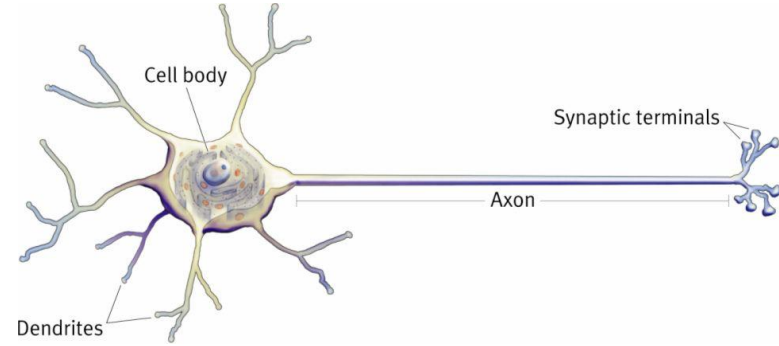
Cortical neuron

Axon that sends messages to other neurons.

Dendritic tree which receives messages from other neurons.

A connection between an axon and a dendritic tree is termed *synapse*.

Axon hillock is an event where an electric charge flow gets through a synapse and depolarizes the cell membrane, thus making the axon generate outgoing spikes.



Cortical neuron

In this way each neuron receives inputs from other neurons, whose outputs are weighted by a synaptic weight.

The weight is determined by transmitter chemicals. They differ in shape and bind to the post-synaptic neuron, creating holes in the membrane.

These holes determine specific ions to flow, making the synapses to adapt. They are very slow commuting devices compared to transistors, but have other advantages.

Computation is made possible thanks to 10^{11} neurons and 10^4 weights per neuron each of us has in the brain

Cortical neuron

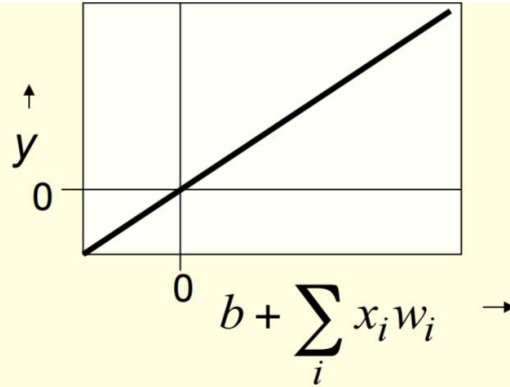
The cortex has a general purpose structure, with the ability to specialize to ad-hoc configurations for particular tasks, in response to experience

Rapid parallel computation happens when the net has learnt, while maintaining its inner flexibility.

Quite like an FPGA, where generic hardware elements are built and one defines which particular circuit configuration should be implemented, according to a specific netlist that usually is synthesized automatically

Linear models of artificial neurons

$$y = b + \sum_i x_i w_i$$



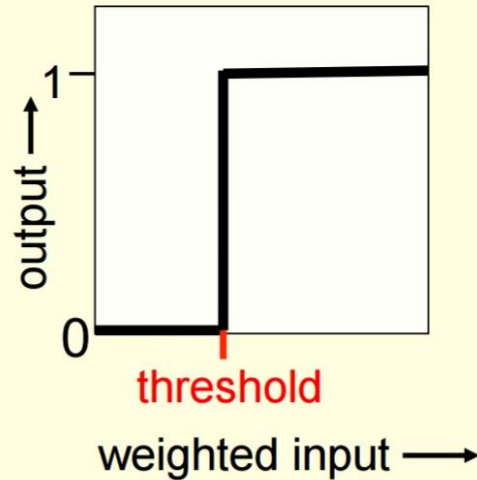
Simple but limited
idealized example.

The model represents the
transfer function of the
artificial neuron.

Binary threshold neurons

$$z = b + \sum_i x_i w_i$$

$$y = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



If the weighted sum of the inputs exceeds a given threshold, the neuron sends out a 1 or otherwise a 0.

Also called Heaviside step function. This model is also termed *activation function* of the artificial neuron.

Multilayer perceptron

Differentiable nonlinear activation function.

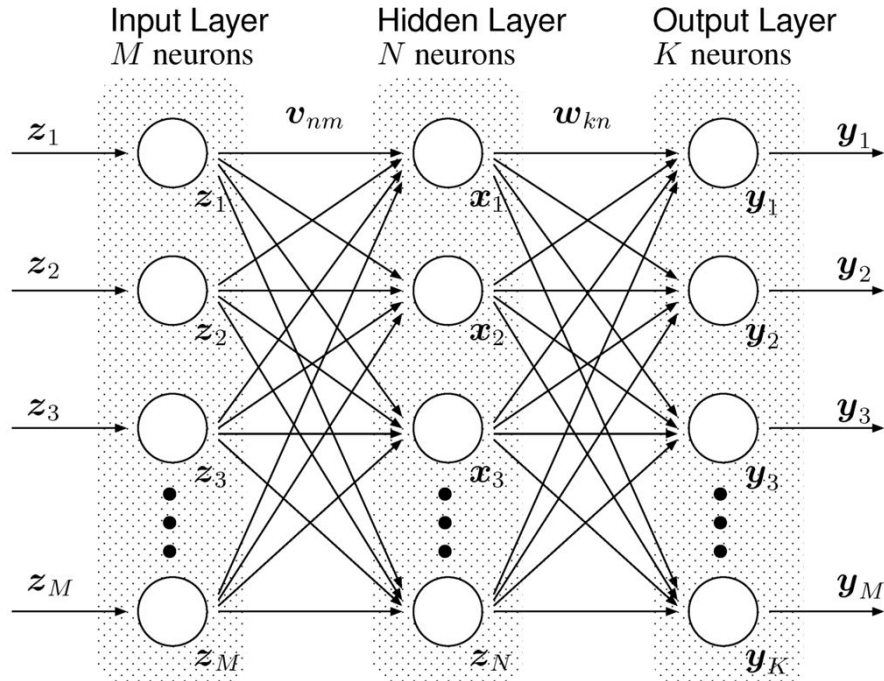
The purpose of the activation function is to introduce non-linearity into the network, allowing the model to exhibit a non-linear behavior.

Input,output, hidden layers.

An input vector it is propagated forward through the network, layer by layer.

The network exhibits a high degree of connectivity.

Multilayer perceptron



Neural networks

“A neural network is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use” (Neural Networks and Learning Machines, Simon O. Haykin)

Neural networks resemble the style of the human brain, which is very different from sequential computation.

Multilayer perceptron

First, the presence of a distributed form of nonlinearity and the high connectivity of the network make the theoretical analysis of a multilayer perceptron difficult to undertake.

Second, the use of hidden neurons makes the learning process harder to visualize.

Multilayer perceptron

Back-propagation algorithm: supervised learning technique

In the forward phase, the synaptic weights of the network are fixed and the input signal is propagated through the network, layer by layer, until it reaches the output.

In this phase, changes are confined to the activation potentials and outputs of the neurons in the network.

Multilayer perceptron

In the backward phase, an error signal is produced by comparing the output of the network with a desired response.

The resulting error signal is propagated through the network, again layer by layer, but this time the propagation is performed in the backward direction.

In this second phase, successive adjustments are made to the synaptic weights of the network.

Calculation of the adjustments for the output layer is straightforward, but it is much more challenging for the hidden layers.

Multilayer perceptron

The local gradient $\delta_j(n)$ depends on whether neuron j is an output node or a hidden node:

Output node: $\delta_j(n)$ equals the product of the derivative and the error signal, both of which are associated with neuron j

Hidden node: $\delta_j(n)$ equals the product of the associated derivative and the weighted sum of the s computed weights for the neurons in the next hidden or output layer that are connected to neuron j

$$\begin{pmatrix} \text{Weight} \\ \text{correction} \\ \Delta w_{ji}(n) \end{pmatrix} = \begin{pmatrix} \text{learning-} \\ \text{rate parameter} \\ \eta \end{pmatrix} \times \begin{pmatrix} \text{local} \\ \text{gradient} \\ \delta_j(n) \end{pmatrix} \times \begin{pmatrix} \text{input signal} \\ \text{of neuron } j, \\ y_i(n) \end{pmatrix}$$

Machine Learning

Instead of writing a program by hand for each specific task, we collect many examples that specify the correct output for a given input.

- A machine learning algorithm then takes these examples and produces a computational model that does the job.
- The model produced by the learning algorithm may look very different from a typical hand-written program. It may contain millions of numbers.
- If we do it right, the program works for new cases as well as the ones we trained it on.
- If data changes, the model can change too by training on the new data

ML problems

- Recognizing patterns:
 - Objects in real scenes – Facial identities or facial expressions – Spoken words
- Recognizing anomalies:
 - Unusual sequences of credit card transactions – Unusual patterns of sensor readings in a nuclear power plant
- Prediction:
 - Future stock exchange rates – Which movies will a person like?

ML problems

- Machine learning already the preferred approach to
 - Speech recognition, Natural language processing
 - Computer vision – Medical outcomes analysis – Robot control

Machine Learning

Training set: Training examples to pair the input with expected output.

Validation set: To estimate how well the model has been trained

Machine Learning

Problem Setting:

- Set of possible instances X
- Unknown target function $f : X \rightarrow Y$
- Set of function hypotheses H

Input: • Training examples $\{ \}$ of unknown target function f

Output: • Hypothesis $h \in H$ that best approximates target function f

Machine Learning

Overfitting:

Training data T: $\text{error}_T(h)$

Entire distribution D of data: $\text{error}_D(h)$

Hypothesis h overfits training set if there is another hypothesis h' such that $\text{error}_t(h) < \text{error}_t(h')$ and $\text{error}_d(h) > \text{error}_d(h')$

FPGA

A field programmable gate array is a semiconductor device on which the function can be defined after manufacturing.

An FPGA enables us to program product features and functions, adapt to new standards, and reconfigure hardware for specific applications even after the product has been installed in the field.

When to use an FPGA

Hardware/software system codesign

Small size

Low Cost

Heat dissipation

Rapid development and prototyping of custom hardware products

FPGA

Common FPGA Applications:

Aerospace and Defense

Medical Electronics

ASIC Prototyping

Audio

Automotive

Real-Time Video Engine

Encoders

Displays

Switches and Routers

Consumer Electronics

And many more..

FPGA

Demand for high-performance computing is a hot topic

- Smart watch with tens of GPUs in addition to CPUs
- Next-generation base stations will need around 500 cores
- Computation demand of advanced driver assistance systems requires about 40 cores

FPGA and AI

Intel recently revealed an FPGA accelerator that offers high computational power for developing AI-powered services

Intel sees FPGAs as the key to designing a new generation of products to address emerging customer workloads

FPGA and AI

FPGA, and GPUs are both good options to overcome problems where computational heterogeneity is key, competing with other solutions like supercomputing or HPC for acceleration

Each of them excels in one scope rather than another one, so it's likely that most of them will be used in a field or another.