Chapter 1

Machine Learning for Monte Carlo simulations

Computing techniques loosely based on mimicking the behavior of the human brain are becoming more and more important in a vast range of applications. Their utilization is not new, with studies based on simple Neural Netwoks [**?**, **?**, **?**] dating back at tleast tot he 80s; what is instead quite recent is the possibility to deploy efficient computing architectures, often specifically tailored to the tasks. At the same time, the capability to deploy larger and larger system has triggered theoretical studies, driving to more solid bases and to the definition of more complex and pecialized models.

In these chapter we will start with an introduction to the model most relevant for Monte Carlo simulations, followed by a selection of spplications. In the last part of the chapter, we will review the trong and weak poins about the utilization of Neural Networks applications for Monte Carlo simulations.

1.1 Introduction to Neural Networks

Neural Networks are a specific branch of the Artificial Intelligence (AI) domain in computer science. They get their inspiration from the fact that humans are able to fulfill complex tasks; hence, by replicating the low-level mechanisms of the human brain on computing systems, one can potentially construct high level algorithms with similar capabilities.

1.1.1 The human brain

Neglecting any functional description, the human brain can be described as an organ composed by neuron, glial cells, neural stem cells and blood vessels (Figure 1.1). With our current understanding, the neurons are the units performing

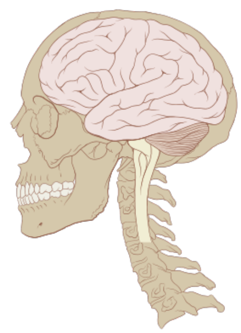

Figure 1.1: A pictorial view of the human brain (from Wikipedia).

basic "operations" within the human brain, and their aggregate response is generating the high-level behaviour typical of humans. A neuron, as sketched in Figure **??**, is composed of three main units: a number of dendrites, the soma (the cell body), and an axion; the total size largely varies between different types of neurons; the neurons used for cognitive functions (as those in the grey matter of the brain) are usually short, XX μm. Functionally, a neuron is able to generate an electric response on the axion (*output*), depending on the electrical potential present at the synapses (*inputs*) present on the dendrites, generating a quite low-level response mechanism. Neurons are *chained* by connections berween axions and dendrites, generating a mesh in which N neurons are connected via M synapses. The high-level response of the human brain to stimula is understood to come from the complexity of such mesh, with a standard human brain featuring 10^{11} neurons each with 7000 synapses, for a total of 10^{15} "connections".

In literature various models of the neuron behavior have been proposed [**?**, **?**, **?**], here we will focus on the simplest yet most simple to implement in computer systems [**?**] (see Figure 1.2): in this model, the *output* y signal at the axion is

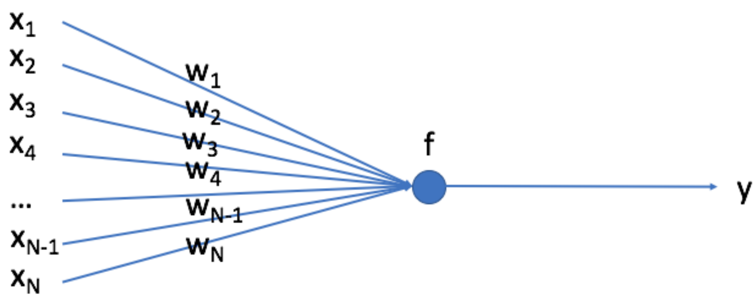

Figure 1.2: The artificial neuron.

assumed to be a function of the *inputs* x_i via

$$y = f(\sum_{i=1}^{N} w_i x_i) \tag{1.1}$$

where w_i are weights defined by chemical potentials at the synapses, and the function f wants to model the non linearity of response of biological neurons with the *inputs*; on top of this, the function f is needed in the mathematical model in order to allow the description of non linear phenomena [**?**]. The percepton [**?**], one of the first models used in literature for Neural Networks, uses a very similar model, with a simplified f function which is simply

$$f(\vec{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^{N} w_i x_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.2}$$

Today, two small modifications are typical when using Neural Netorks:

- the addition of a further synapse x_0 which is alwsys 1, as a bias to the system; its weigth is referred to as x_0 or b () as in]emphbias.
- the use of continous f non linear functions, as the logistic [**?**] or the hyperbolic [**?**] functions.

Neural networks are obtained by combining multiple neurons in *networks*, usually in a layered structure: one layer is used to map the inputs, a few/many layers are *hidden*, and a single layer used to to map the outputs. On top of that, more complex neurons an be used, for example including a "memory" cell, or presenting a recurrent behavior by reusing its output as one of the inputs. A full description of all the type of neurons and networks is beyond the scope of this chapter; in the following, the ones most relevalnt to Monte Carlo simulations will be presented wit more detail. For reference, still, a complete classification of currently relevant neural networks is shown in Figure 1.3.

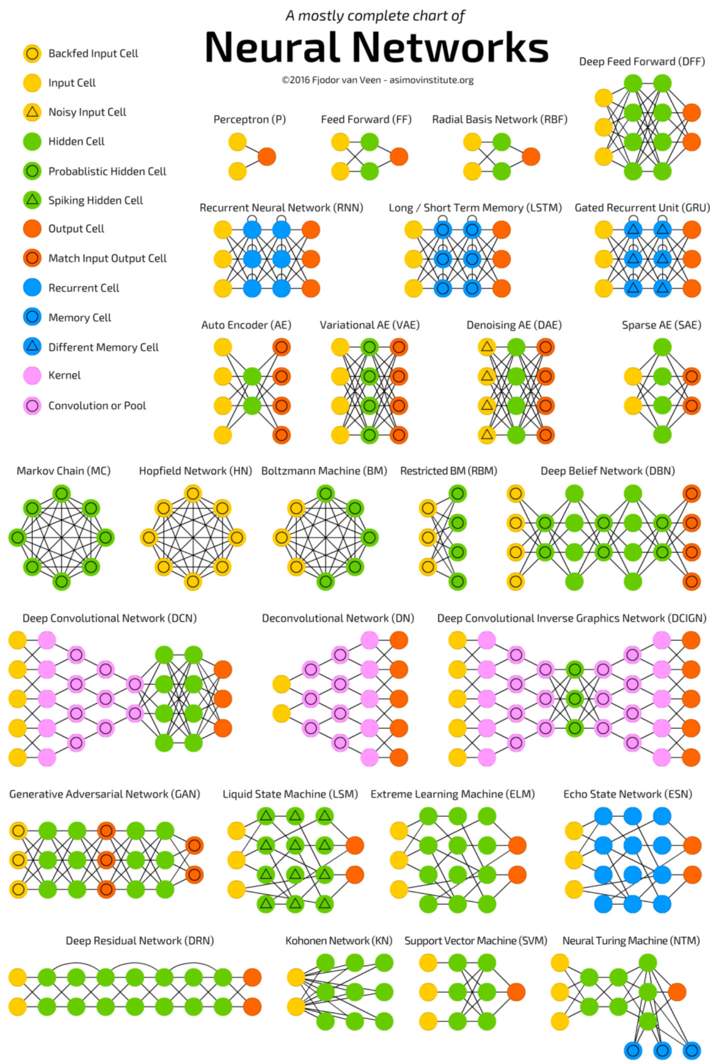

Figure 1.3: Typs of neurona nd neural networks currently relevant in literature (Copyright F. van Veen 2016).