Machine Learning for Neuroscience

Slides and notebooks: `https://github.com/PBarnaghi/ML4NS`
Notes by: *Nan Fletcher-Lloyd, Payam Barnaghi*

Lecture 6. Neural Networks

In lecture 6, we cover the basic concept behind the design and use of neural networks.

In machine learning, we are interested in modelling biologically inspired networks known as artificial neural networks to help us solve learning and decision-making problems. Our previous lectures have focused on the use of linear models; however, linearity implies the weaker assumption of monotonicity (any increase in a feature will always result in an increase or decrease of a model's output). Such an assumption is not always true in real-world applications.

The simplest kind of neural network is a single layer perceptron network, which consists of a single layer of output nodes to which the inputs are directly fed via a series of weights. The sum of the products of the inputs and their respective weights is calculated in each node and if the value is above a certain threshold, the neuron fires and takes the activated value. Otherwise, the neuron takes the deactivated value. Neurons with this kind of activation function are also called artificial neurons or linear threshold units.

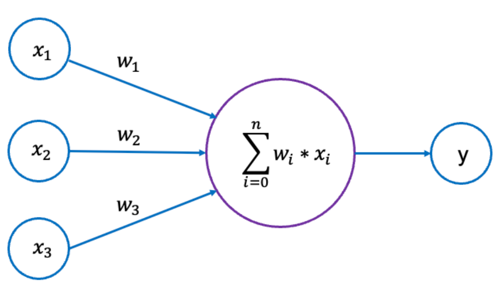

Figure 6.1. A simple illustration of an artificial neuron

Figure 6.1 illustrates a single artificial neuron (a type of single layer perceptron network that acts as a binary classifier).

In this lecture, we discuss how the limitations of linear models can be overcome by incorporating one or more hidden layers (with activation functions) into a neural network. We define network depth and width as the number of hidden layers and number of nodes in the hidden layers, respectively. We also introduce two commonly used activation functions by which to add nonlinearity to neural networks (the *sigmoid* function and rectified linear units or *ReLU*). Networks with multiple layers of computational units are known as multi-layer perceptrons or MLPs.

Figure 6.2 illustrates the architecture of a fully connected, feed-forward multilayer perceptron.

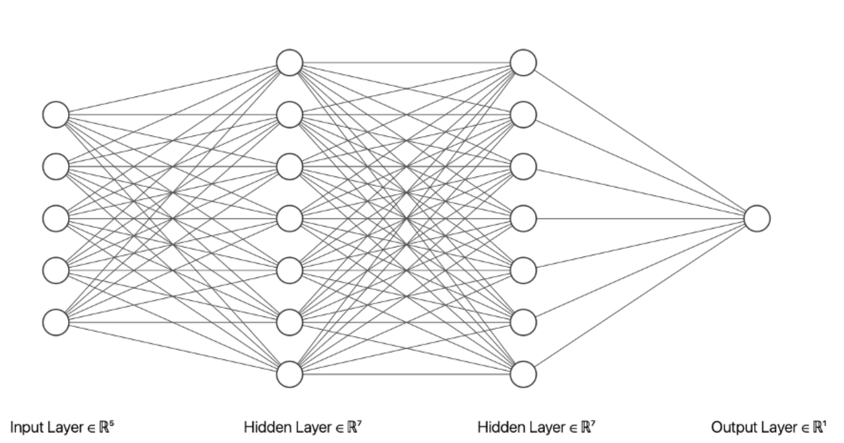

Figure 6.2. A sample multilayer perceptron architecture
Drawn using: https://alexlenail.me/NN-SVG/

MLPs are often interconnected in a feed-forward way, which is where the connections between nodes do not form a cycle as in recurrent neural networks (RNNs).

We further expand on how we train feed-forward neural networks using forward propagation and backpropagation. Finally, we consider how to design our neural networks, including choosing an appropriate number of hidden layers and choosing an appropriate learning rate for our neural networks. We also introduce and explain the objective of different adaptive learning rate optimisation algorithms.

The tutorial aims to build your familiarity with the machine learning framework *PyTorch* and will also demonstrate the basics of building and training simple neural networks.

A corresponding assessment will help you to evaluate your understanding of developing neural networks for machine learning and decision-making problems and demonstrate the skills you have learnt so far.