1 Artificial Neural Networks

1.1 Introduction

An *artificial neural network* is simply a mathematical function. Specifically, a vector function, $\mathbf{F}$, of a vector argument $\mathbf{x}$ (often called the *input signal*),

$$\mathbf{y} = \mathbf{F}_{\mathbf{p}}(\mathbf{x}) \tag{1}$$

where the vector $\mathbf{y}$ is the return value (often called *response*), and the vector $\mathbf{p}$ is the set of internal parameters of the network. The parameters are tuned such that the network can perform some useful function like recognizing a pattern in a bitmap picture or approximating a solution to a differential equation. Tuning the parameters for a specific task is called *network training* or *learning* [1].

1.2 Applications

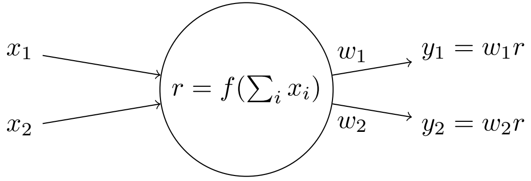
Neural networks have found numerous applications in many areas. In particlar they are used for pattern recognition. An example of pattern recognition is handwriting recognition. In this case the input to the network is the bitmap picture of a handwritten character: the vector $\mathbf{x}$ contains the RGB values of the bitmap's pixels. The response $\mathbf{y}$ of the network is the UTF code of the recognised character.

Another example is the traffic sign recognition network for vehicle control systems. In this case the input to the network is the bitmap picture from the vehicle's camera and the output is the code of the recognized traffic sign.

In physics neural networks are used as function approximants (for interpolation, regression, and numerical solution of differential equations) as well as in data processing and modelling of complex systems.

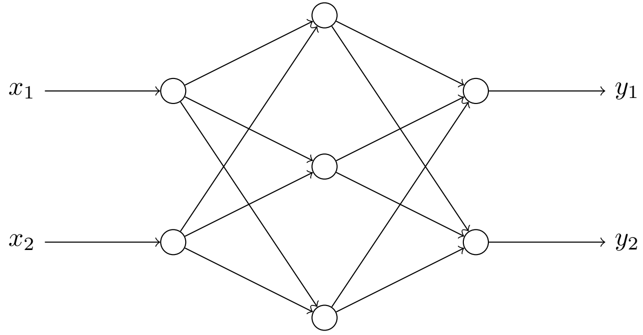
1.3 Graphical representation

Artificial neural networks are designed to resemble biological neural networks in the brains of animals, hence the name. Specifically, an artificial neural network is a collection of connected nodes called artificial neurons, typically represented graphically. A single artificial neuron is customarily pictured as



The neuron takes one or more input signals x_i, applies the neuron's *activation function*, f, to the sum of the input signals producing its response, r, and then sends one or more output signals, y_i, where the response is multiplied by weight-factors w_i.

Here is an example of a network that takes a 2-dimensional vector x as input and returns a 2-dimensional vector y as output,



The left column of neurons (which receive the input signal $\mathbf{x}$) is called the *input layer*. The right column of neurons which send out the response $\mathbf{y}$ is called the *output layer*. The middle column of neurons is called the *hidden layer*. Each arrow connecting neurons carries its own weight-factor. Networks with one or more hidden layers are often called *deep neural networks*. Networks where the signal flow is always from left to right are called *feedforward* networks.

The activation function $f_i(x)$ of the i'th neuron is often chosen as

$$f_i(x) = f\left(\frac{x - a_i}{b_i}\right) \tag{2}$$

where f is the common activation function and the shift a_i and scale b_i are the neuron's parameters. The network parameter vector $\mathbf{p}$ is then given as the collection of all weight-factors w_{ij}, shifts a_i, and scales b_i.

There exist many different types of networks with different numbers of neurons/layers and with different topological structures.

1.4 Training (learning)

Training is tuning the network's parameters $\mathbf{p}$ to better handle the given task. It typically involves minimization of certain *cost function* of network parameters, $C(\mathbf{p})$. The two major training paradigms are *supervised* and *unsupervised* learning.

Supervised learning uses a set of inputs paired with the desired outputs. Here the cost function is given by the difference between the network's output and the desired output. For example, in handwriting recognition the supervised learning can use a set of bitmaps with known handwritten characters with the cost function being the number of wrongly recognized characters.

Another example is the one-dimensional interpolation where the input data is pairs $\{x_i, y_i\}_{i=1\dots n}$ (the table to interpolate) and the cost function is the average squared deviation,

$$C(\mathbf{p}) = \frac{1}{n}\sum_{i=1}^{n}\left(F_{\mathbf{p}}(x_i) - y_i\right)^2 \ . \tag{3}$$

In unsupervised learning the input data is given together with the cost function but without the correct output. For example, in solving a differential equation the cost function could be the average mismatch between the left- and right-hand sides of the differentical equation.

References

[1] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61(arXiv:1404.7828):85–117, 2015.