ARTIFICIAL NEURAL NETWORKS

Authors: G. H. Zerwes, *Universidade Federal do Espirito Santo, Vitória, Brazil*

Research Supervisor: L. Homri, *Arts et Métiers ParisTech, Metz, France*

1 Introduction

Artificial Neural Networks (ANN) is a versatile algorithm that is applied to a vast variety of problems, ranging from regression, classification and multi-class classification. It's main advantages are that it can handle well large amounts of data, as well as managing to achieve good results in non-linear problems.

Its working principal consists of a series of layers that perform mathematical transformations on the data to give the desired output. Each step, that is, each layer that the data goes through, it is multiplied by coefficients and suffers an activation to be transformed in the input of the next layer. The first layer of the networks is called the input layer, because that is where the data first enters the networks. The last layer is called the output layer, as it is where data exits the networks. Finally, all remaining layers are called hidden layers of the system.

Another key concept is that of neurons, that is to say, the nodes inside the network model. They are the singular units that compose each layer of a network. It is, in the neurons that the mathematical transformations above mentioned take place.

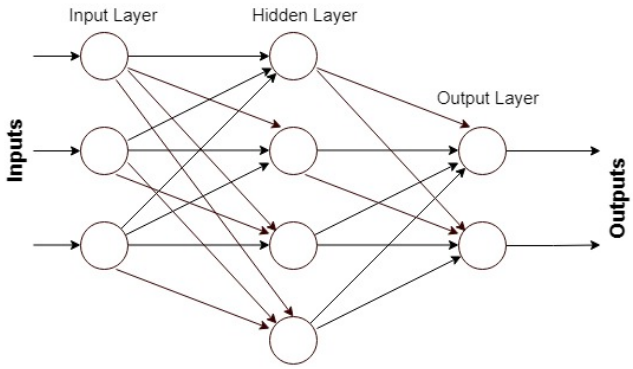

Figure 1: Example of ANN

An example of a feedforward neural network can be seen in figure 1. At the leftmost place, the input layer can be seen, and that is where data enters in the system and suffers the first set of transformations before it is passed forward to the hidden layer. After that there are the hidden layer, that are represented as a single layer in this image, but can be any number of intermediate layers between the input and output. Finally, the rightmost layer is the output. The network shown is said to be feedforward because data goes only one way through the network. Other, more complex types of neural networks exist in which data is fed back into previous layers as well.

2 Pseudo-code

To use this algorithm on your data, the pseudo-code below shows a procedure that should be followed.

```
#Step 1: Gather the data. The X data is the independent variable and should be already
    pre-treated. The Y array is the vector containing the labels for each class.

X = [x_value_1, x_value_2, ...]
Y = [y_value_1, y_value_2, ...]

#Step 2: Fit the model.
model = ANN.fit(X,Y)

#Step 3: Make new predictions.
y_prediction = model.predict(X)
```

3 Other recommendations

An important observation is that this algorithm is scale sensitive, that is to say, that the data should be scaled in the range [0,1], [-1,1] or to have zero mean and unitary variance. Furthermore, whenever a new prediction is to be made, the same transformation applied to the training data must be applied to the new data.

Another downside of this algorithm is the large number of parameters that can be changed, which makes it difficult to find the optimal network. In the particular case of the regularization parameter α, it is recommended to do a Grid Search, that is, an extensive search over the interval of $[10^{-1}, 10^{-7}]$

There are different algorithms to perform the optimization problem that lies in the training process of the network. For most cases, when there are large amount of data points, the recommended algorithm is ADAM, however, for smaller datasets, the L-BFGS can converge faster and with better results.

References

Bishop, C. M. (2006). *Pattern Recognition And Machine Learning*. Number 758. Springer.