Co-financed by the Connecting Europe Facility of the European Union

Neural Network and Tree Models

Trustworthy AI - Lecturer: Emanuela Raffinetti; Python instructor: Alex Gramegna
E-mail: emanuela.raffinetti@unipv.it; alex.gramegna01@universitadipavia.it

Background on neural network models

- Neural networks have been developed in the field of machine learning, imitating the neurophysiology of the human brain.
- Neural networks are information processing systems, composed of a large number of highly interconnected elements (neurons), working in union to solve specific problems.
- There are two main kinds of neural networks: supervised and non-supervised (self-organising).
- Here we consider supervised neural networks.

Architecture of a neural network - I

- The most common structure of a supervised network is composed of a graph, where the nodes (*neurons*) are placed on more levels (*layers*) and are interconnected from a layer to the other in a single direction.

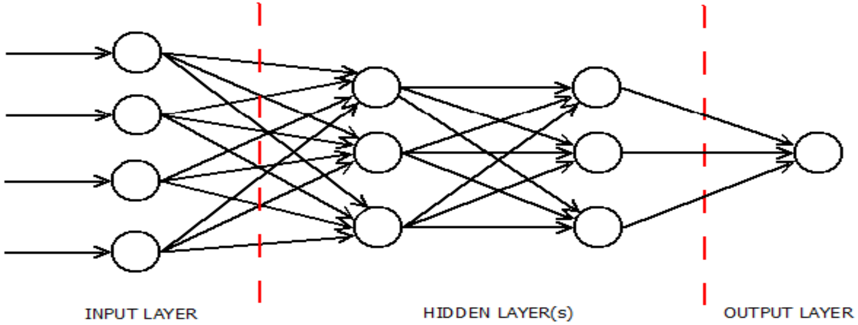

Figure: Structure of a supervised neural network

Co-financed by the Connecting Europe Facility of the European Union

Architecture of a neural network - II

- The neurons of a neural network are organized in three types of layers: (i) input, (ii) output, (iii) hidden.
- (i) The **input** layer receives information from the external environment; each neuron in it usually corresponds to an explanatory variable.
- (ii) The **output** layer furnishes the final results, which are sent by the network to the outside of the system. Each of its neurons corresponds to a response variable.
- (iii) The **hidden** layers contain intermediate computational neurons, whose role is to increment the model fit.

How does a neural network work? - I

- A generic neuron j, with a threshold θ_j, receives n input signals $x = [x_1, x_2, x_3 \ldots\ldots x_n]$ from the units it is connected to in the previous layer.
- Each signal has an importance weight: $w_j = [w_{1j}, w_{2j}, w_{3j} \ldots\ldots w_{nj}]$
- The same neuron elaborates the input signals, according to their importance weights and the threshold value, through a function called *combination function*, that produces a value called *potential* or *net input*.
- The potential of a neuron j is defined by the following linear combination:

$$P_j = \sum_{i=1}^{n} (x_i w_{ij} - \theta_j)$$

How does a neural network work? - II

- Consider now the way according to which every neuron sends signals in output.
- The output of the j-th neuron, y_j, derives from the application of a function, called *activation function*, to the potential P_j:

$$y_j = f(x, w_j) = f(P_j) = f\left(\sum_{i=0}^{n} x_i w_{ij}\right)$$

How does a neural network work? - III

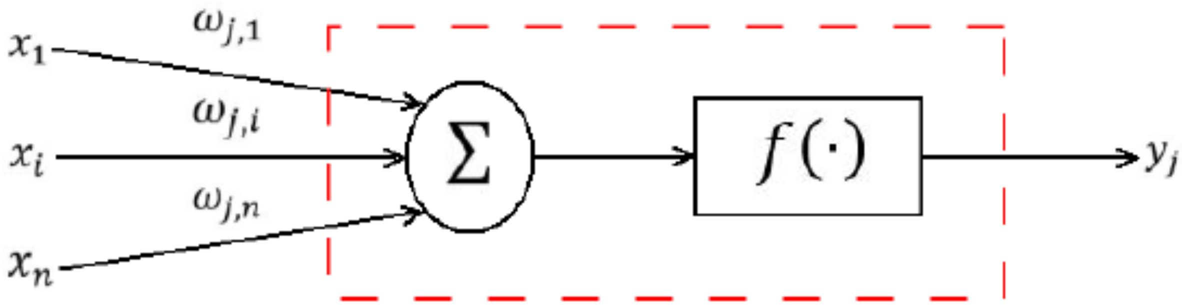

Figure: Signal elaboration in a neuron

Single-layer perceptrons

- Neural networks with a single set of weights are known as **single layer perceptrons**.
- They are constituted by n input units $(x_1, x_2, x_3.....x_n)$ connected to a layer of p output units $(y_1, y_2, y_3.....y_p)$ through a system of weights that can be represented in matrix form, as follows:

$$\begin{matrix} w_{11} & \dots & w_{1j} & \dots & w_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ w_{i1} & \dots & w_{ij} & \dots & w_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ w_{n1} & \dots & w_{nj} & \dots & w_{np} \end{matrix}$$

w_{ij} represents the weight of the connection between the i-th neuron of the input layer and the j-th neuron of the output layer

Multi-layer perceptrons - I

- Neural networks with more than one set of weights are called **multi-layer perceptrons**.
- Consider the simplest case, with n neurons in the input layer, h in the (only) hidden layer and p in the output layer.
- The network can be represented by a double system of weights: the weights $w_{ik}(i = 1, ...n; k = 1, ...h)$ that connect the nodes of the input layer with the hidden ones and the weights $z_{kj}(k = 1, ...h, j = 1, ...p)$ that connect the nodes of the hidden layer with the output nodes.

Multi-layer perceptrons - II

- The output of a generic neuron j of the output layer is therefore:

$$y_j = g\left(\sum_k l_k z_{kj}\right) = g\left(\sum_k z_{kj} f\left(\sum_i x_i w_{ik}\right)\right)$$

- A deep learning network is a neural network with many layers, described by many compound functions.

Estimates and predictions

- The weight parameters of a neural networks are typically unknown, and should be estimated from the data.
- However, the final aim is not to estimate some parameters, but to predict future values.
- The performance of a neural network can be measured by splitting the data into a training and a validation set.
- The best architecture will then be chosen by maximising the obtained performance measure.

Training of a neural network - I

- Once decided the structure of a neural network, it is necessary to *train* it.
- This process is based on a training set which is composed of an input vector and a target vector, and is called *supervised learning*.

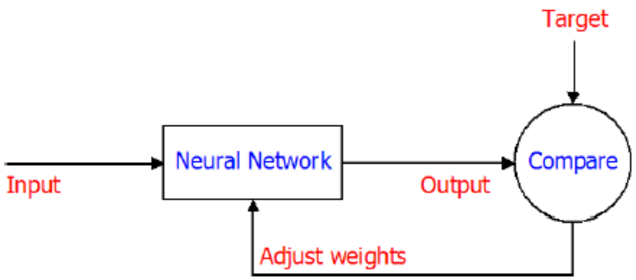

Figure: Supervised learning in a neural network

Co-financed by the Connecting Europe Facility of the European Union

What are tree models?

- Predictive data mining models
- Classify observations in groups; score constantly within each group
- **Regression trees** when target Y is continuous
- **Classification trees** when target Y is categorical

How do they work?

- A tree model is defined by a recursive procedure
- n statistical units are progressively divided in subgroups according to a splitting rule that maximizes purity (homogeneity) of the Y values within each obtained group.

What is the splitting criterion to be maximised?

For a regression tree the splitting criteria to be maximized is:

$$\Phi_{(s,t)} = I_v(t) - \sum_{r=1}^{s} I_v(t_r) p_r$$

$$I_v(m) = \frac{\sum_{l=1}^{n_m} (y_{lm} - \hat{y}_m)^2}{n_m}$$

Misclassification and Gini

Misclassification:

$$I_M(m) = \frac{\sum_{l=1}^{n_m} 1(y_{lm}, y_k)}{n_m}$$

where k is the class to which the observation belongs.

Gini:

$$I_G(m) = 1 - \sum_{k=1}^{K} \pi_k^2$$

Tree model estimate

- For each response observation y_i, a tree model yields an estimate which is the mean of the target variable in the group containing the observation i.

$$\hat{y}_i = \frac{\sum_{l=1}^{n_m} y_{lm}}{n_m}$$

- For classification trees such an estimate is a class probability:

$$\hat{\pi}_i = \frac{\sum_{l=1}^{n_m} y_{lm}}{n_m}$$

Example

Figure: Decision Tree

Chaid Trees

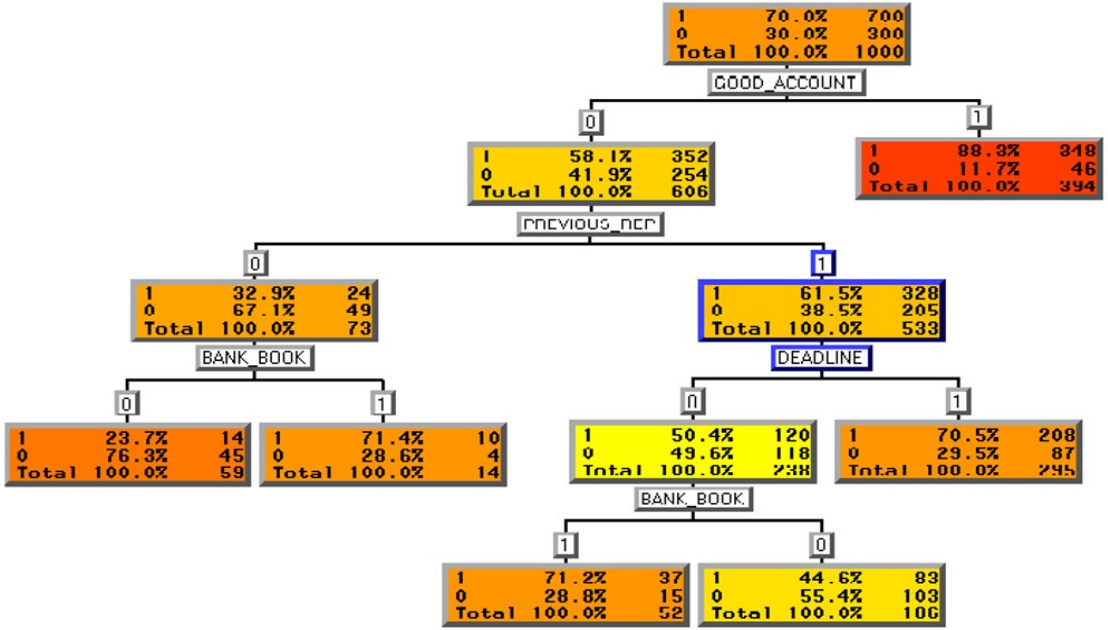

Figure: Decision Tree

Co-financed by the Connecting Europe Facility of the European Union

The obtained tree rules

IF GOOD_ACCOUNT = 1	N : 394 1 : 11.7% 0 : 88.3%
IF BANK_BOOK = 0 AND PREVIOUS_REP = 0 AND GOOD_ACCOUNT = 0	N : 59 1 : 76.3% 0 : 23.7%
IF BANK_BOOK = 1 AND PREVIOUS_REP = 0 AND GOOD_ACCOUNT = 0	N : 14 1 : 28.6% 0 : 71.4%
IF DEADLINE = 1 AND PREVIOUS_REP = 1 AND GOOD_ACCOUNT = 0	N : 295 1 : 29.5% 0 : 70.5%
IF BANK_BOOK = 1 AND DEADLINE = 0 AND PREVIOUS_REP = 1 AND GOOD_ACCOUNT = 0	N : 52 1 : 28.8% 0 : 71.2%
IF BANK_BOOK = 0 AND DEADLINE = 0 AND PREVIOUS_REP = 1 AND GOOD_ACCOUNT = 0	N : 186 1 : 55.4% 0 : 44.6%

Table: Decision Tree Output

CART trees

- Based on a pruning strategy
- First the tree is grown at its maximum level. Then pruning proceeds backward, at each step eliminating the node that maximizes:

$$Ra(T) = R(T) + aN(T)$$

where for a final partition T:
R(T) = total misclassification error
N(T) = number of leaves
a = penalty term (default a = 1)

Co-financed by the Connecting Europe Facility of the European Union

Cross-validation criteria

- Data sampled into training and test subsets (eg. 70%-30%)
- Models are selected and fit on the training sample.
- Predictions for the test sample are then compared with actual values

What is Random Forest?

- An ensemble classifier using many decision tree models;
- Can be used for both classification and regression;
- Accuracy and variable importance information is provided with the results.

The Algorithm (Flow-Chart)

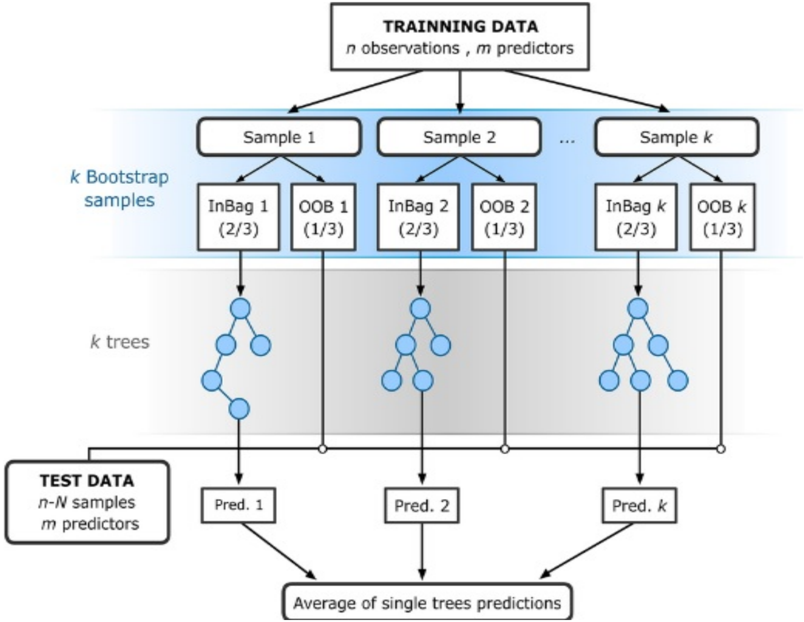

Figure: Flow Chart

Co-financed by the Connecting Europe Facility of the European Union

Advantages and Disadvantages

- It produces a highly accurate classifier and learning is fast.
- It can handle thousands of input variables without variable deletion.
- It offers an experimental method for detecting variable importance.
- When used for regression, it cannot predict beyond the range in the training data.
- It may over-fit data sets that are particularly noisy in nature.

Co-financed by the Connecting Europe Facility of the European Union

Reference

- Mitchell T.M.: Machine Learning 4° edition, McGraw Hill (1997), available at: https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf