Neural Networks

Lluís Talavera, 2022

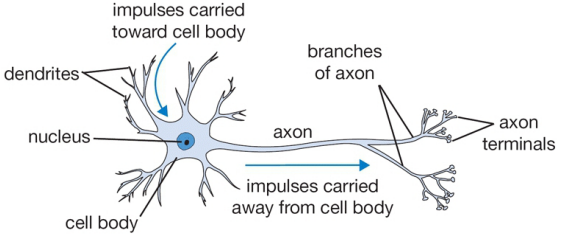
UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

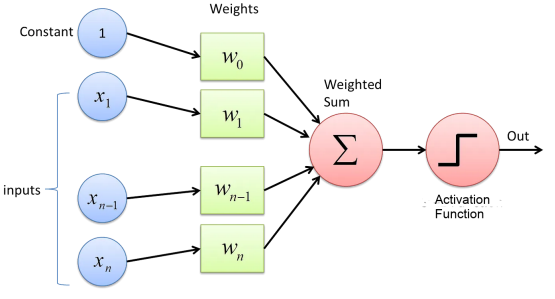
ETSEIB

What are neural networks?

Inspired by the human brain, in which billions of cells called neurons form complex webs of connections with one another, processing information as they fire signals back and forth.

"Neurons" are nodes in a network. Input nodes are multiplied by weights and the result adjusted with a constant value known as bias. The results are passed through an activation function to map the output (either classification or regression).



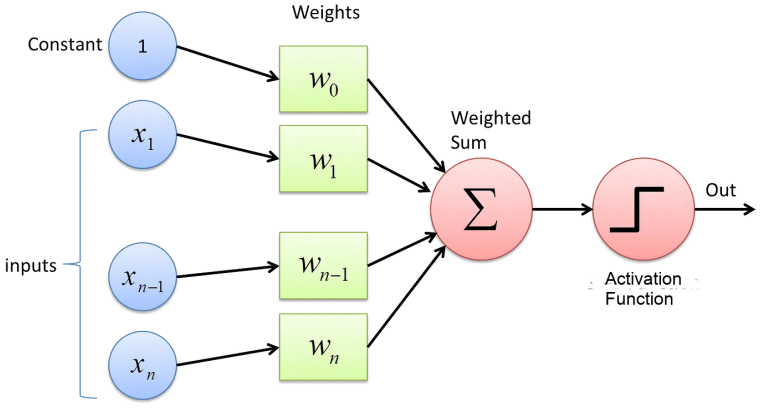


Perceptrón

Is a single unit network (one neuron).

Only learns hyperplanes (linearly separable concepts).

Very similar to logistic regression.

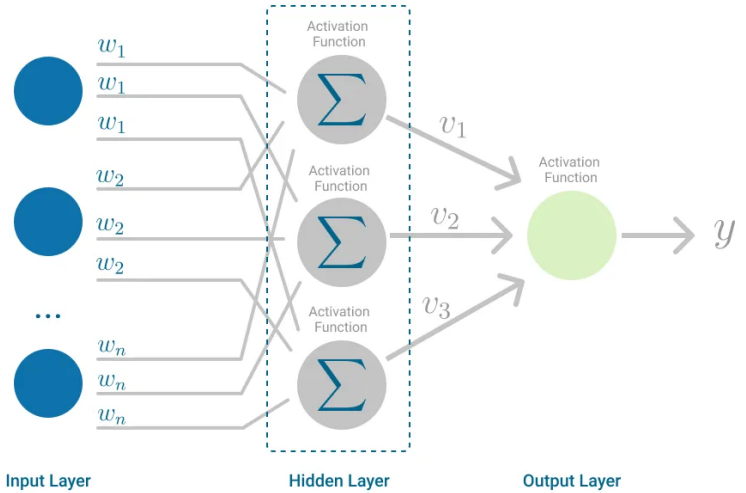


Multi-layer Perceptron (MLP)

Three layers: input, output and a hidden layer.

It is fully connected: each input node is linked to each node in the hidden layer.

It can learn non-linear decision boundaries.



Backpropagation

- Starting with the input layer, propagate data forward to the output layer. This step is the forward propagation.
- Based on the output, calculate the error (the difference between the predicted and known outcome). The error needs to be minimized.
- Backpropagate the error. Find its derivative with respect to each weight in the network, and update the model.

Repeat the three steps given above over multiple epochs (iterations over the data) to learn ideal weights.

In sklearn, error is defined as the Cross Entropy loss function and is often referred to as loss.

Hyperparameters

Number of layers. Terminology: we do not count the input layer. A 2-layer network is a network with one hidden layer and an output layer.

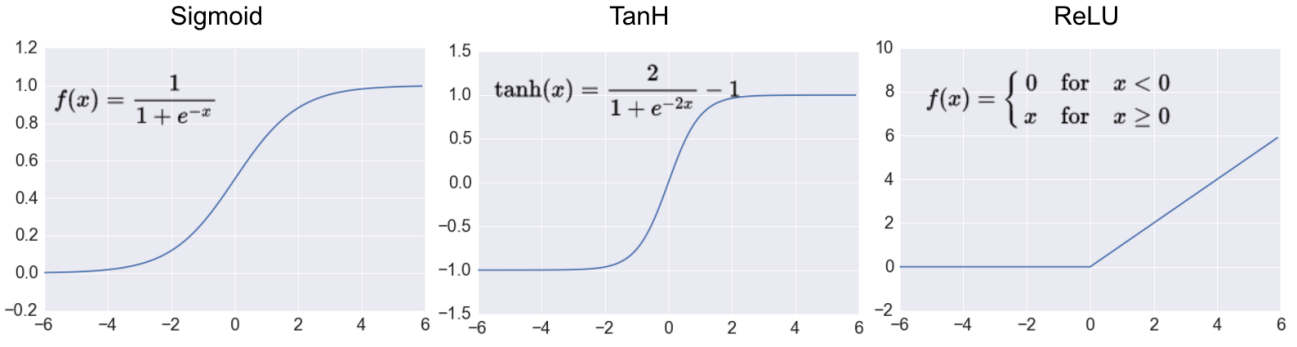
Number of nodes in each hidden layer.

The architecture of the network defines the functions that can be approximated.

But it is difficult to characterize for a particular network structure what functions. can be represented and what functions can not.

Hyperparameters

Activation function: Non-linear activation functions are needed because a linear combination of linear functions is still a linear function.



Hyperparameters

- **Learning rate**: step size at each iteration while moving toward the minimum of the loss function.
- **Alpha**: a regularization parameter, a penalty that constraints the size of the weights. Increasing/decreasing alpha encourages smaller/larger weights. Similar to the C parameter in logistic regression and SVM.
- **Early stopping**: if we set this to `True`, the algorithm splits the training data and creates an internal validation set to terminate training when the validation score is not improving.

(`sklearn` provides the attribute `loss_curve_`, that stores the loss metric computed at each step of learning, and `` `validation_scores_ `` that stores the scores over the validation set.)