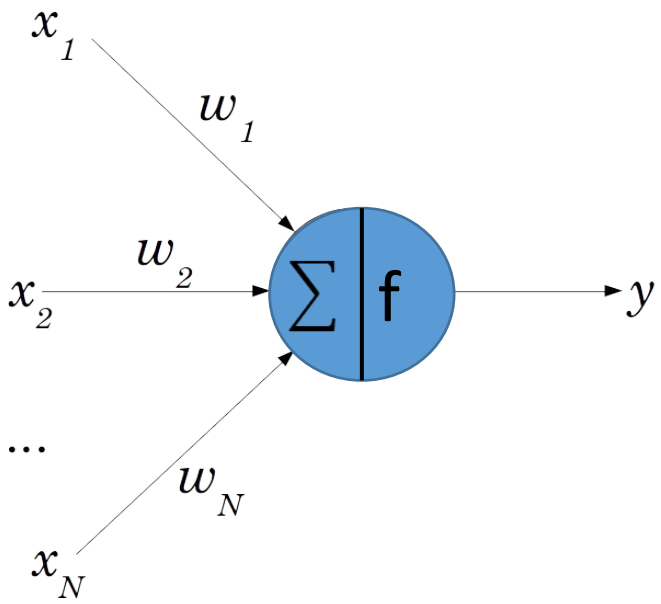
Introduction to Artificial Neural Network

Perceptron: **Artificial neuron**



$$y = f(\sum \boldsymbol{x_j} * \boldsymbol{w_j})$$

$$f = sigmoid()$$

Training

1. Propagation of the inputs towards the output
2. Compute the loss

Least square error: $J = (y\text{-}y_{correct})^2$

3. Minimize the loss to find the best prediction

Gradient Descent:

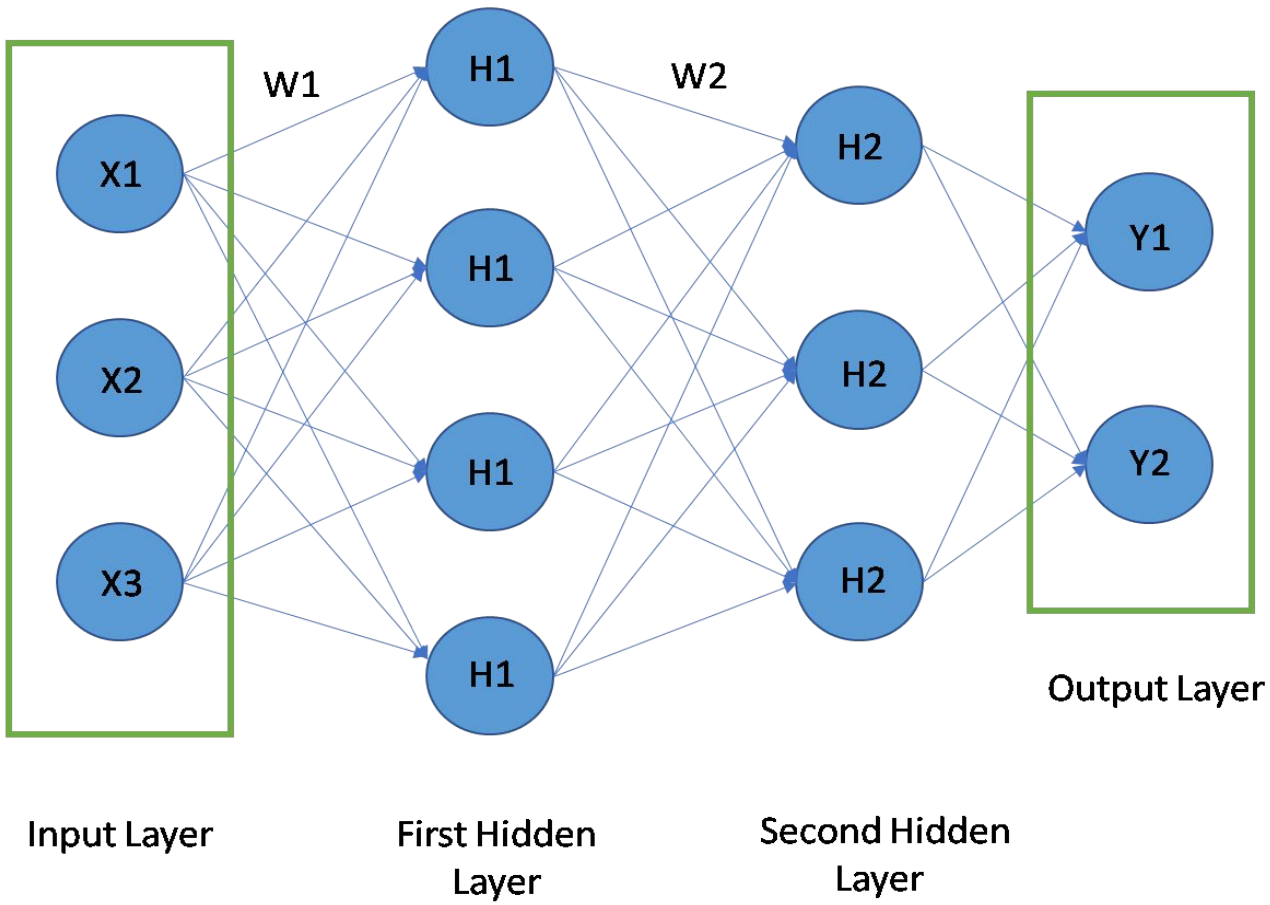
Repeat $\left\{ w_j = w_j + \alpha \frac{\partial}{\partial w_j} J(w_j) \right\}$

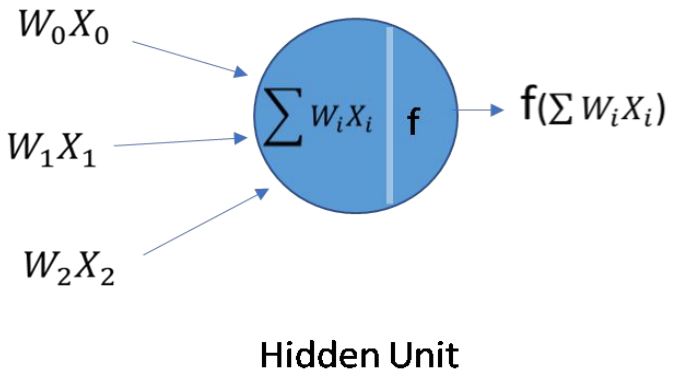
Testing

1. Performs same operations 1 and 2 of the training

However for complex problems you need a muti-layers of perceptrons

Neural Networks





Operations:

1. Propagation of the inputs towards the output
2. Compute the loss (Least squares error)
3. **Apply backpropagation algorithm**
4. Minimize the loss to find the best prediction (Gradient Descent)

Example: MNIST dataset

- Find the corresponding number of a **handwritten** digit