Data Mining
Minería de Datos

Ivan Saavedra, Ph.D.
saavedrai@uninorte.edu.co

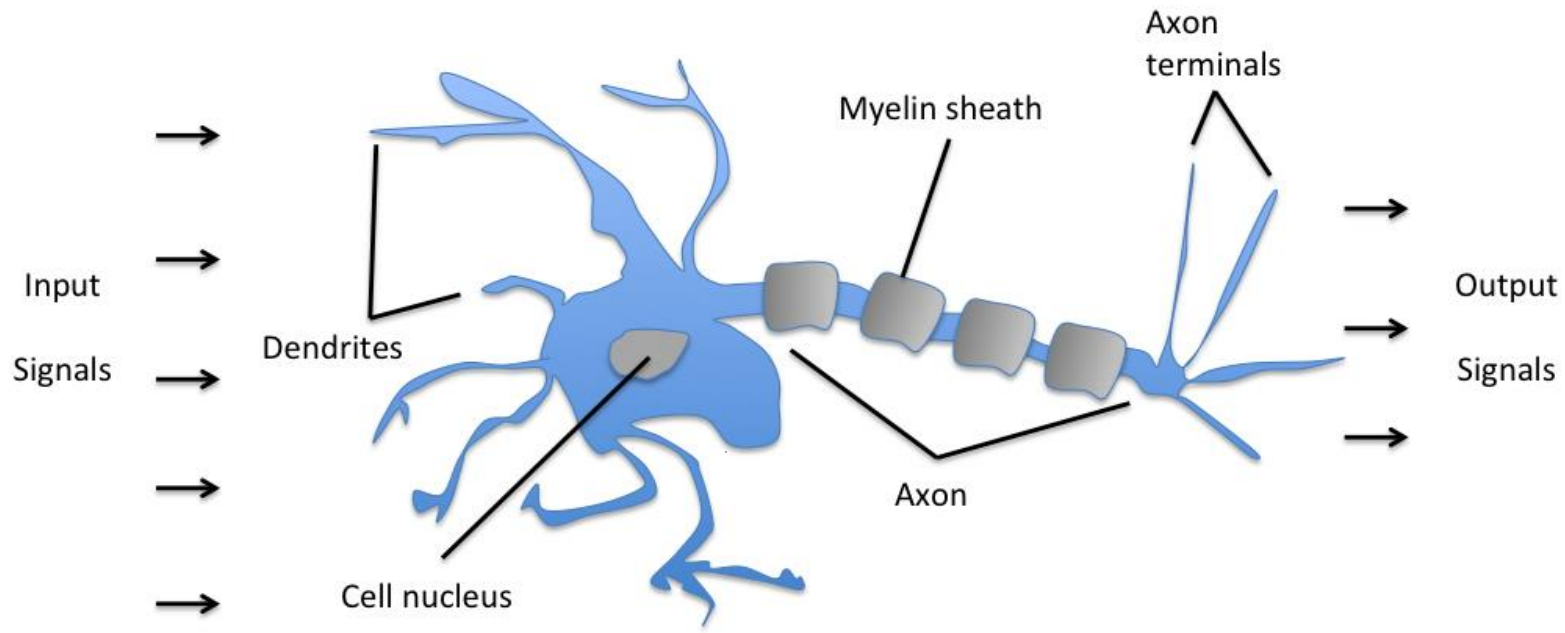
Universidad del Norte
División de Ingenierías
Dpto. Ingeniería de Sistemas

202030

Topics

- Background
- Statistical Concepts
- Markov Chain Monte Carlo
- Parametric Classification
- Non-Parametric Classification
- Clustering
- Decision Trees
- **Artificial Neural Networks**

Artificial Neurons and the McCulloch-Pitts Model



Schematic of a biological neuron.

Source: https://sebastianraschka.com/Articles/2015_singlelayer_neurons.html

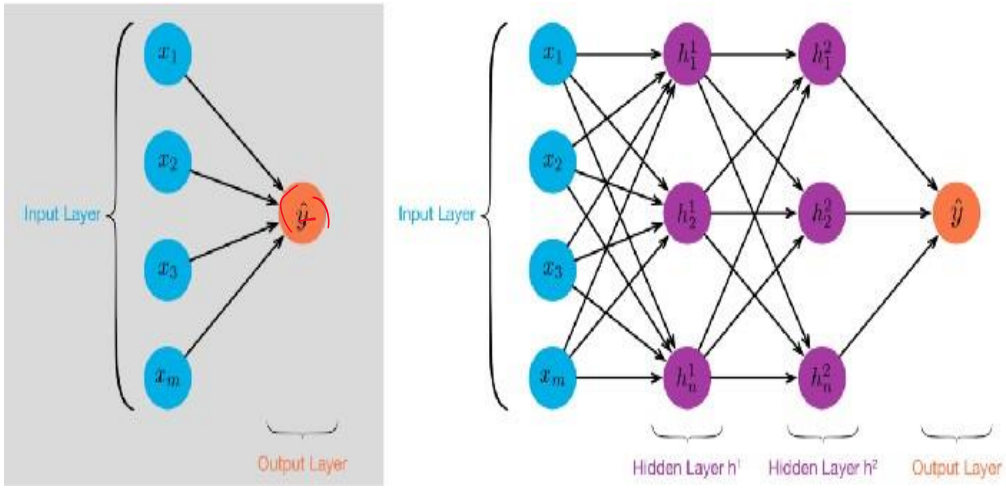
W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4):115–133, 1943.

Perceptron Model

- Developed at Cornell Aeronautical Laboratory, United States, in 1957 for machine-implemented image recognition
- Automatically learn the optimal weight coefficients that are then multiplied with the input features in order to make decision of whether a neuron fires or not.

1. Single-layered perceptron model
   - includes a feed-forward network
   - depends on a threshold transfer function
   - analyze only linearly separable objects with binary outcomes
   - doesn't have previous information
   - weights are allocated inconstantly
   - added weighted inputs are compare to a threshold
   - activated and delivered output as +1.
   - weights are updated to minimize errors.

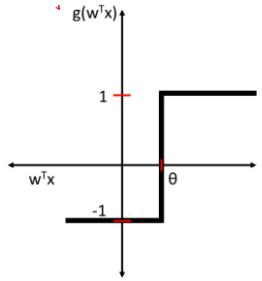
2. Multi-layered perceptron model
   - Similar to SLP but with more number of hidden layers
   - Also called Backpropagation algorithm
   - It is executed in two stages; the **forward stages** and the **backward stages**.
   - the activation function is no longer linear but non-linear



Source: https://towardsdatascience.com/multi-layer-neural-networks-with-sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f

Single Layer Perceptron Model

- We can pose this problem as a binary classification task
- Two classes: 1 (positive class) and -1 (negative class)
- An activation function $\phi(z)$ takes a linear combination of certain input values x and a corresponding weight vector w
- Where z is the so-called net input ($z=w_1x_1 + \dots + w_mx_m$):
- $$w = \begin{bmatrix} w_1 \\ \cdot \\ \cdot \\ \cdot \\ w_m \end{bmatrix}, x = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_m \end{bmatrix}, z = w^T x + b$$
- If the activation of a sample $x^{(i)}$ is greater than a define threshold θ, predict class 1 or class -1
- The activation function is a simple unit step function, also called the Heaviside step function
- $$\phi(z) = \begin{cases} 1 \; if \; z \geq \theta \\ -1 \; otherwise \end{cases}$$



Unit step function.

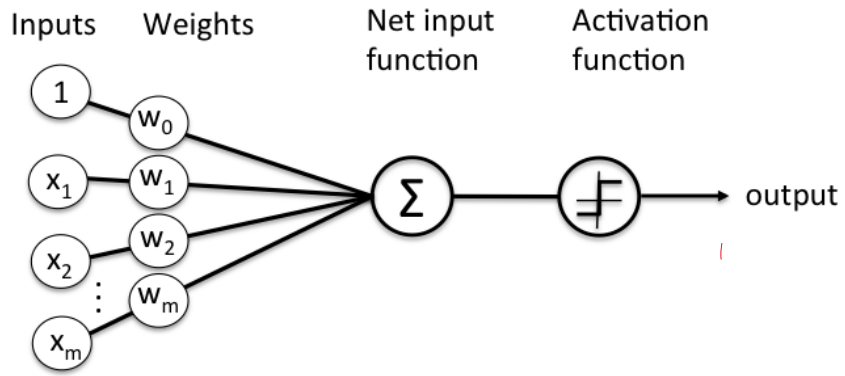
Single Layer Perceptron Model

- The idea is to mimic how a single neuron in the brain works: it either fires or it doesn't
- It can be summarized in the following steps:

    1. Initialize the weights to 0 or small random numbers
    2. For each training sample $x^{(i)}$ perform the following steps:
        1. Compute the output value $\hat{y}$
        2. Update the weights

- The output value is the class label predicted by the unit step function defined earlier
- The simultaneous update of each weight w_j in the weight vector w can be written as
    - $w_j := w_j + \Delta w_j$

- Δw_j is calculated by the perceptron rule
    - $\Delta w_j = \eta\left(y^{(i)} - \hat{y}^{(i)}\right)x_j^{(i)}$
    - Where η is the learning rate



Schematic of Rosenblatt's perceptron.

Supervised Learning – Classification – Artificial Neural networks

Supervised learner by

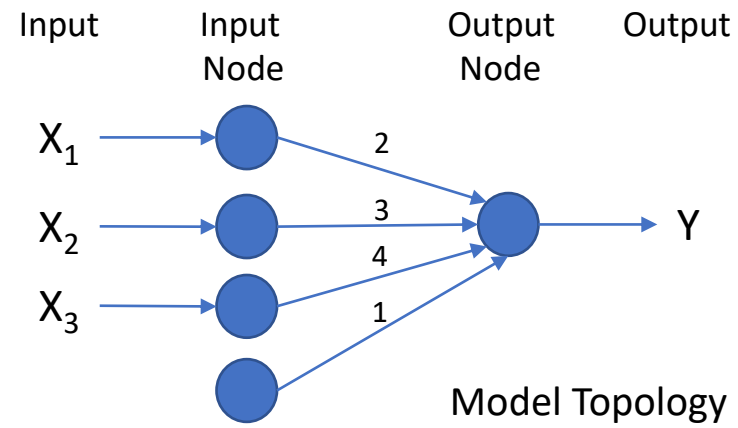
- Developing a functional relationship between input and output variables
- Mimics the architecture of the biological process of a neuron

Foundation

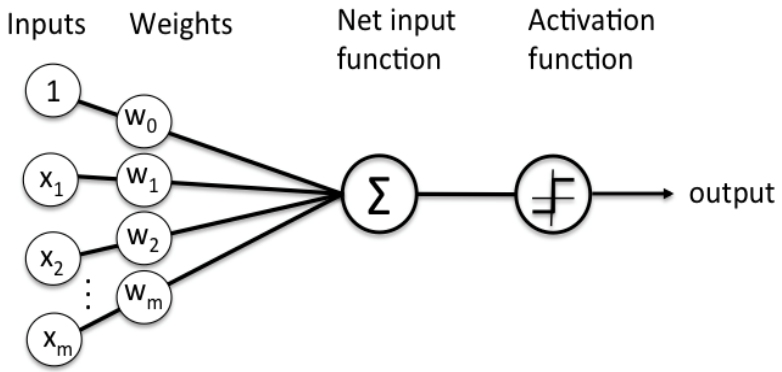
$$Y = 1 + 2X_1 + 3X_2 + 4X_3$$

Where,

Y = calculated output
X_1, X_2, X_3 = input attributes
1 is the intercept
2, 3, and 4 are the scaling factors or coefficients for the input attributes



Model Topology



Schematic of Rosenblatt's perception

Supervised Learning – Classification – Artificial Neural networks

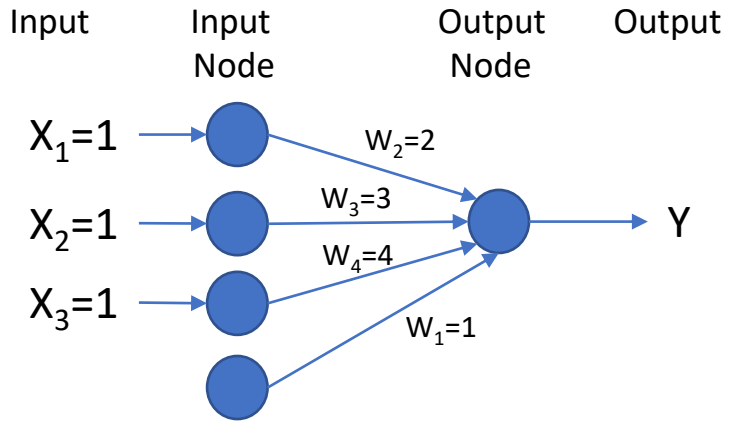
Example:

1. Determine the topology and activation function
   - Dataset with 3 numeric input attributes (X1, X2, X3) and one numeric output (Y)
   - A topology with two layers and a simple aggregation activation function is used
2. Initiation
   - Initial weights for links are 1,2,3, and 4
   - Training record with all inputs as 1 and the output Y=15
3. Calculating Error
   - Calculate the output using the model

     $\bar{y} = 1 + 1 * 2 + 1 * 3 + 1 * 4 = 10$

     $e = y - \bar{y} = 15 - 10 = 5$
4. Weight Adjustment
   - The error is passed back from the output node to all other nodes in the reverse direction
   - The weights of the links are adjusted by a fraction of the error
   - The fraction λ applied to the error is called the learning rate
   - The new weight of the link (w) is the sum of the old weight (w') and the product of the learning rate and proportion of the error (λ*e)

     $w = w' + \lambda * e$
   - Determine the initial value for λ



- The same training example can be repeated until the error rate is less than a threshold

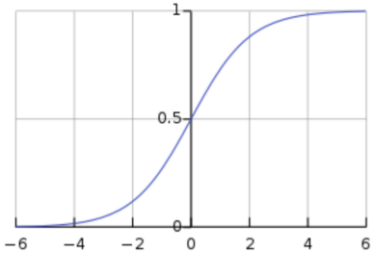
Supervised Learning – Classification – Artificial Neural networks

ANN,

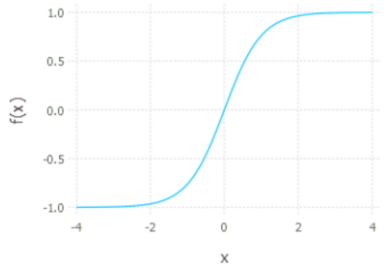
- Is typically used for modeling nonlinear, complicated relationships between input and output variables
- Creates more than one layer in the topology, called hidden layers

The activation function:

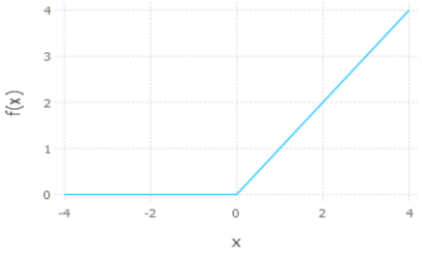
- Consist of a combination of an aggregation function (summarization), and a transfer function (sigmoid, normal bell curve, logistic hyperbolic, or linear functions) for linear and nonlinear transformations



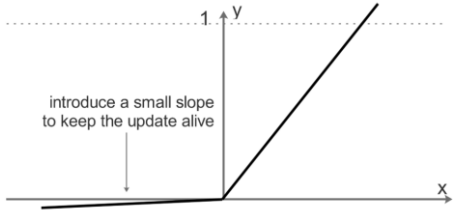
Sigmoid / Logistic



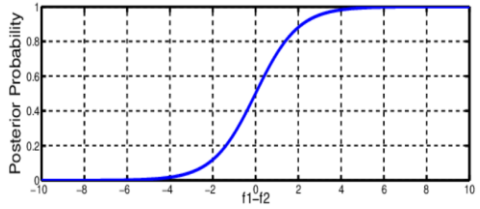
Tan-h / Hyperbolic tangent



ReLU (Rectified Linear Unit)



Leaky ReLU



Softmax

https://medium.com/@snaily16/what-why-and-which-activation-functions-b2bf748c0441

Bibliography

- **Book:** George C. Montgomery and George C. Runger. Applied Statistics and Probability for Engineers.
- **Book:** Vijay Kotu and Bala Deshpande. (2019). Data Science, Concepts and Practice. (Second Edition). Morgan Kaufmann.
- **Book:** Vijay Kotu and Bala Deshpande. Data Science. Concepts and Practice. Second Edition. 2019.
- **Book:** Sebastian Raschka. Python Machine Learning. Packt Publishing. 2015.
- **scikit-learn**. Scikit-learn: Machine Learning in Python, Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. Journal of Machine Learning Research, volume 12, 2011