Neural Networks

Contents

Characteristics of Neural Networks

All Neural Networks **must have** the following characteristics

1. Activation Functions
2. Network Topology *(architecture)*
3. Training Algorithm *("Backpropagation Algorithm)*

Contd…

- **Activation function** – transforms ***a*** neuron's combined input signals into a single output signal to be broadcasted further in the network.

Contd…

- **Network topology** – describes…
    - the number of neurons in the model
    - the number of layers in the model and,
    - the manner in which they are connected.

Contd…

- **Training algorithm** – specifies how connection weights are set in order to **inhibit** or **excite** neurons in proportion to the input signal.
- *(if some inputs are not needed, their connection weights will be made zero, so that the input value x is not participating in the network…similar to association of neurons in brain n/w by weight adjusting based on relative importance.)*

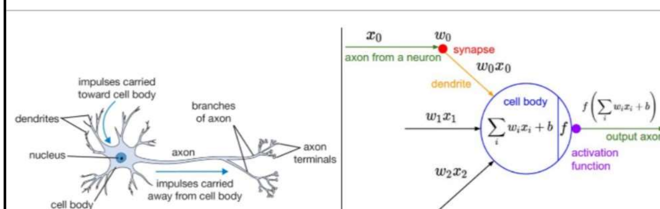
LEARN . GROW . EXCEL GOPEEKRISHNAN R

Activation Functions

- Activation functions decide whether a neuron **should be activated or not.** *(It is just a function that we use to get the o/p from a node)*
- **In the biological neuron:**
    - <u>an activation function</u>
        - ✓could be **imagined as** a process that involves summing the total input signal and determining whether it **meets the firing threshold.**
        - ✓**if it meets** the firing threshold, the **neuron passes on the signal**.
        - ✓**if it does not meet** the firing threshold, it **does nothing**.

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Contd…



A cartoon drawing of a biological neuron (left) and its mathematical model (right).

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Contd…

- We know that ***y = f(v)*** is the final output from a node in a network.
- **y-value** can be anything ranging from **-inf** to **+inf**.
- The node really **doesn't know** the **bounds of the value**.
- **So how do we decide whether the neuron should fire or not?**
- Activation functions **are used**
  1. ***to check*** the **y-value** produced by a neuron and,
  2. ***to decide*** whether outside connections should consider this neuron as "fired" or not.

Contd…

- There are a number of activation functions.
- Some of the commonly used activation functions are briefed here.
  a) threshold activation function
  b) unit – step activation function
  c) sigmoid activation function
  d) linear activation function
  e) saturated linear activation function
  f) gaussian activation function
  g) hyperbolic tangent activation function

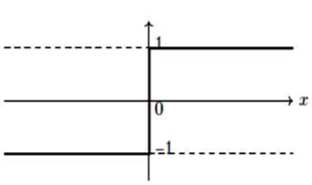
Contd…

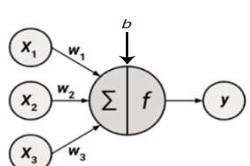
a) Threshold activation function

- The threshold activation function is defined by

$$f(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x \leq 0 \end{cases}$$

- The graph of this functions is shown below.





Reference: *a neuron*

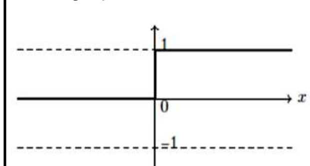
Contd…

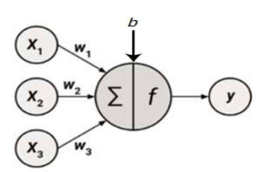
b) Unit – Step activation function

- The unit – step activation function is defined by

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

- The graph of this function is shown below.





Reference: *a neuron*

SAINTGITS

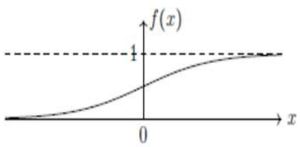
Contd…

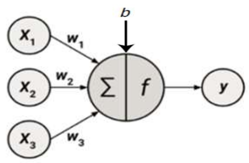
c) Sigmoid activation function

- The sigmoid activation function is defined by

$$f(x) = \frac{1}{1 + e^{-x}}$$

- The graph of this function is shown below.



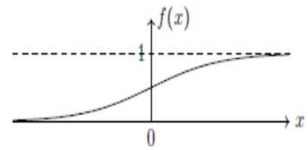


Reference: *a neuron*

SAINTGITS

Contd…

- The main reason why we use sigmoid function is because it exists between (0 to 1).
- Therefore, it **is especially used** for models **where we have to predict the probability as an output**.
- Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice.



SAINTGITS

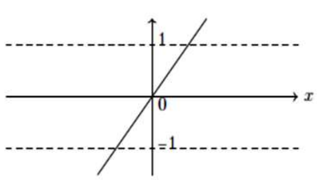
Contd…

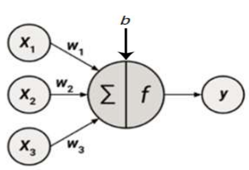
d) Linear activation function

- The linear activation function is defined by

f(x) = m.x

- The graph of this function is shown below.



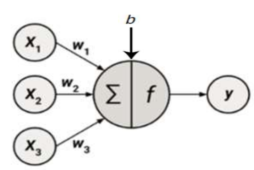


Reference: *a neuron*

SAINTGITS

Contd…

- It is clear that the linear function has the **equation similar to that** of a straight line.
- Here, the activation is seeming to be proportional (∵ **f(x) = mx**) to the input (which is $\sum_{i=1}^{n} w_i x_i$).



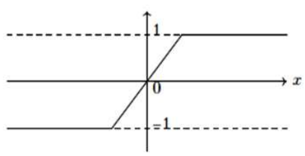
Contd…

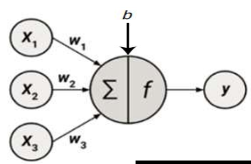
e) Saturated linear activation function

- The saturated linear activation function is defined by

$$f(x) = \begin{cases} 0 & \text{if } x < x_{\min} \\ mx + c & \text{if } x_{\min} \le x \le x_{\max} \\ 0 & \text{if } x > x_{\max} \end{cases}$$

- The graph of this function is shown below.





Reference: *a neuron*

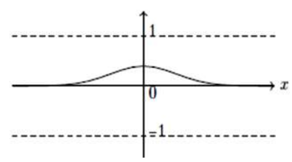
Contd…

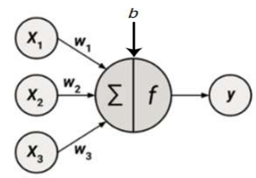
f) Gaussian activation function

- The gaussian activation function is defined by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- The graph of this function is given below.





Reference: *a neuron*

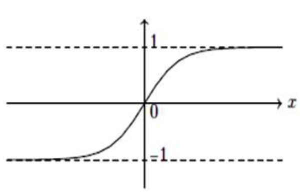
Contd…

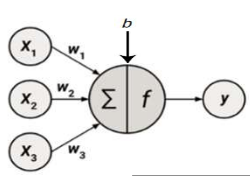
g) Hyperbolic tangent activation function (OR, tanh)

- The hyperbolic tangent activation function is given by

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- The graph of this function is given below.





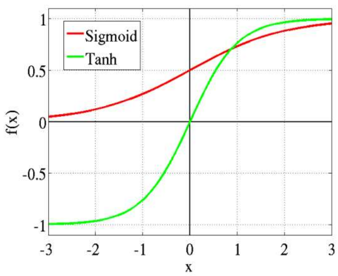
Reference: *a neuron*

Contd…

- **tanh** or hyperbolic tangent function is also like sigmoid function.
- But, **tanh is better** than sigmoid function.
- The range of tanh is from (-1 to +1).
- tanh is also **sigmoidal** ***(S – shaped).***

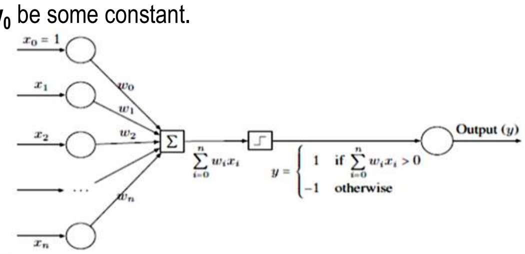
Contd…

- The following graph shows the two curves corresponding to sigmoid activation function (**red**) and hyperbolic tangent activation function (**green**).



Perceptron

- A **perceptron** is an artificial neuron in which the activation function is the threshold activation function.
- Consider an artificial neuron having $\mathbf{x_1, x_2, x_3, \ldots, x_n}$ as the input signals and $\mathbf{w_1, w_2, w_2, \ldots, w_n}$ as the associated weights.
- Let $\mathbf{w_0}$ be some constant.



Contd…

- The neuron **is called** **a perceptron** if the output of the neuron is given by the following function.

$$o(x_1, x_2, \ldots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + \cdots + w_n x_n > 0 \\ -1 & \text{if } w_0 + w_1 x_1 + \cdots + w_n x_n \le 0 \end{cases}$$

References

1. Machine Learning with R, Second Edition, Brett Lantz, PACKT Publishing.