Chapter 3

Activation Functions

4th Class

INTELLIGENT APPLICATIONS

التطبيقات الذكية

أستاذة المادة الدراسة الصباحية : د. ايناس محمد حسين

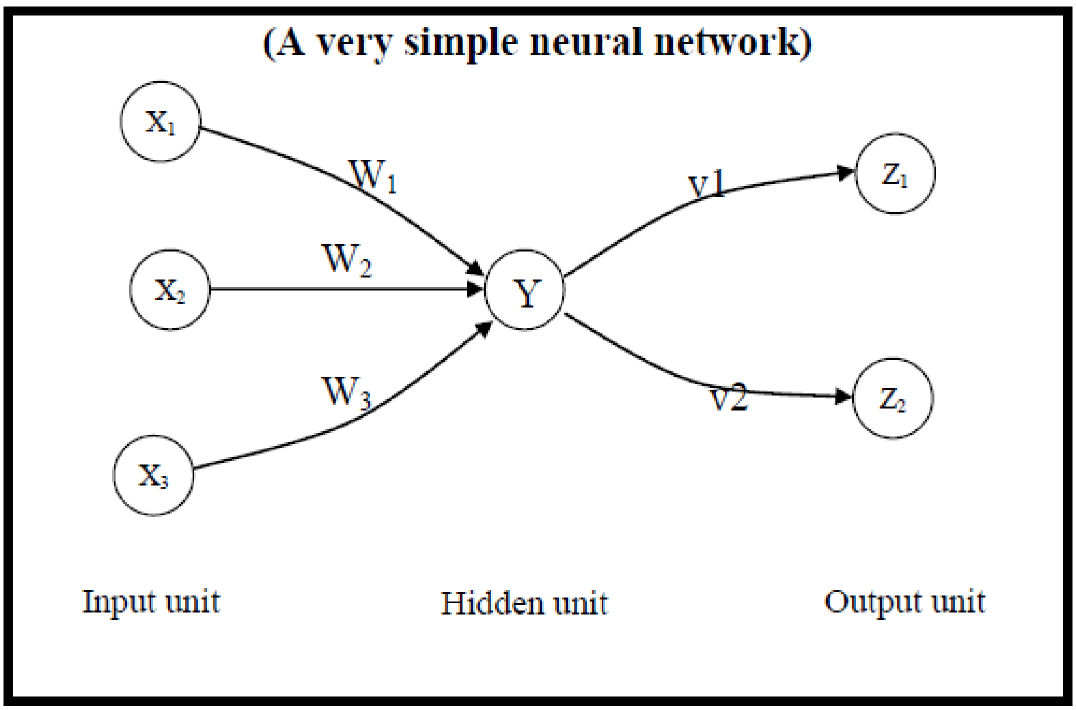
أستاذة المادة الدراسة المسائية : م.م. ايمان حسين رحيم

Activation Functions

3.1 Introduction
3.2 Basic Activation Functions
3.3 Examples:
3.4 The Bias

3.1 Introduction

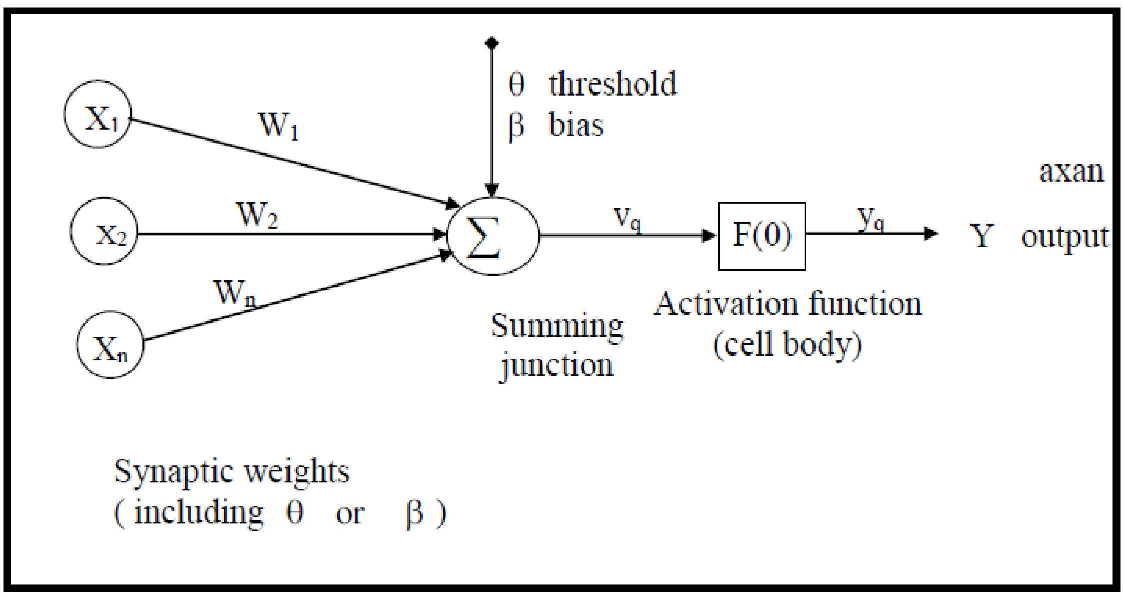
Typical Architecture of NN, Neural nets are often classified as single layer or multilayer. In determining the number of layers, the input units are not counted as a layer, because they perform no computation. Equivalently, the number of layers in the net can be defined to be the number of layers of weighted interconnects links between the slabs of neurons. This view is motivated by the fact that the weights in a net contain extremely important information. The net shown bellow has two layers of weights:



3.2 Basic Activation Functions

The activation function (Sometimes called a transfers function) shown in figure below can be a linear or nonlinear function. There are many different types of activation functions. Selection of one type over another depends on the particular problem that the neuron (or neural network) is to solve. The most common types of activation function are:-

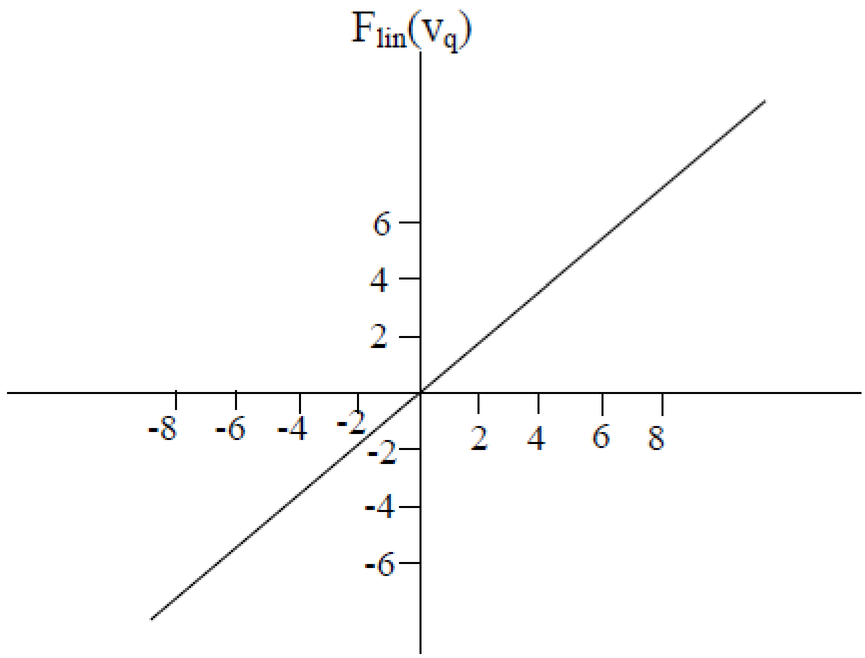
$$V_q = \sum_{v=0}^{n} W_{qj} X_j$$



Alternate nonlinear model of an ANN

1- <u>The first type</u> is ***the linear*** (or ***identity***) function. Ramp

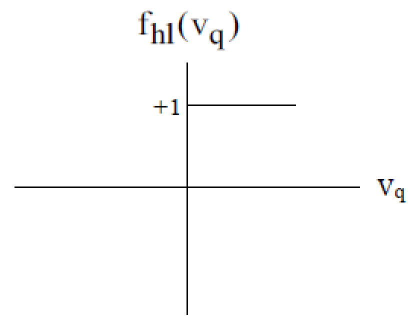
$$y_q f_{lin}(v_q) = v_q$$



2-<u>The second type of activation function</u> is a ***hard limiter***; this is a binary (or bipolar) function that hard-limits the input to the function to either a **0 or a 1 for the binary type**, and a **-1 or 1 for the bipolar type**. The <u>binary hard</u> limiter is sometimes called the <u>threshold function</u>, and the <u>bipolar hard</u> limiter is referred to as the <u>symmetric hard limiter.</u>

a- The o/p of the binary hard limiter:-

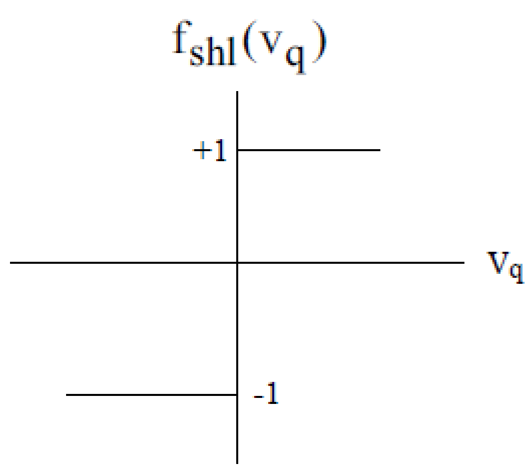
$$y_q = f_{hl}(v_q) = \begin{cases} 0 & \text{if} \quad v_q < 0 \\ 1 & \text{if} \quad v_q >= 0 \end{cases}$$



b-The o/p for the symmetric hard limiter (shl):-

$$y_q = f_{shl}(v_q) = \begin{cases} -1 & \text{if} \quad v_q < 0 \\ 0 & \text{if} \quad v_q = 0 \\ 1 & \text{if} \quad v_q > 0 \end{cases}$$

double side تسمى ايضا

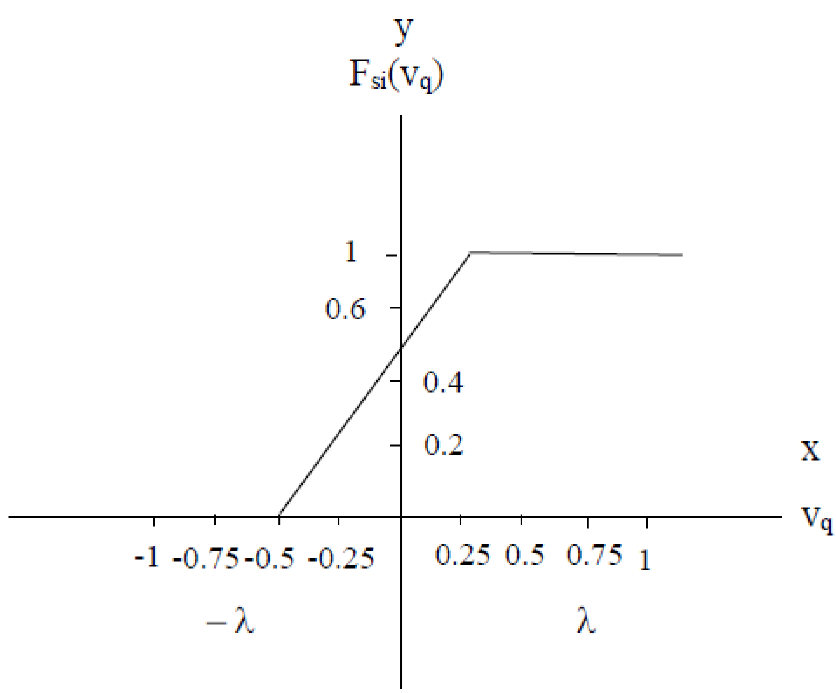


3-<u>The third type of basic activation function</u> is the ***saturating linear*** function or threshold logic Unite (tLu).
This type of function can have either a binary or bipolar range for the saturation limits of the output. The bipolar saturating linear function will be referred to as the symmetric saturating linear function.

a- The o/p for the ***saturating linear*** function (binary o/p):-

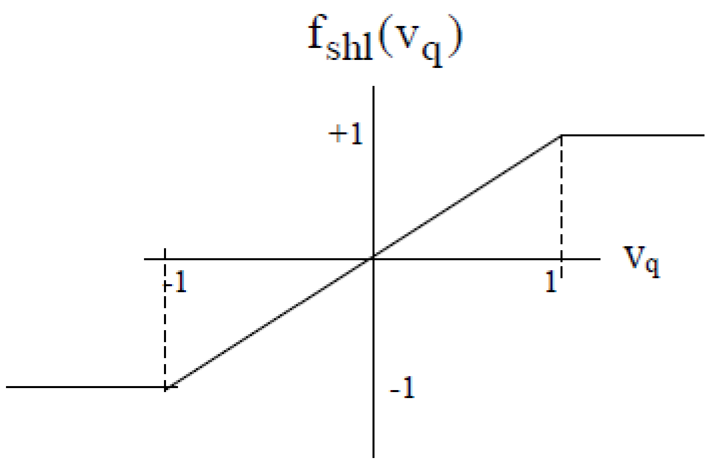
$$y_q = f_{sl}(v_q) = \begin{cases} 0 & \text{if} \quad v_q < -1/2 \\ v_q + 1/2 & \text{if} \quad -1/2 <= v_q <= 1/2 \\ 1 & \text{if} \quad v_q > 1/2 \end{cases}$$

or $$y = \begin{cases} \lambda & \text{if} \quad x > \lambda \\ x & \text{if} \quad -\lambda <= x <= \lambda \\ -\lambda & \text{if} \quad < -\lambda \end{cases}$$



b- The o/p for the ***symmetric saturating linear*** function:-

$$y_q = f_{ssl}(v_q) = \begin{cases} -1 & \text{if} \quad v_q < -1 \\ v_q & \text{if} \quad -1 <= v_q < = 1 \\ 1 & \text{if} \quad v_q > 1 \end{cases}$$

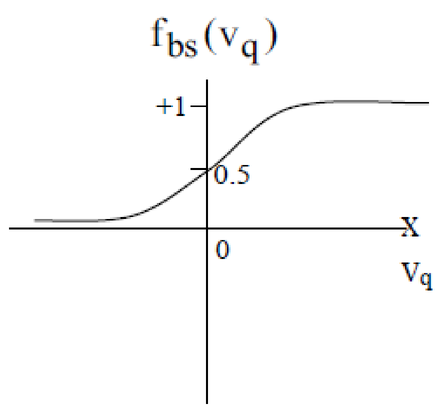


4-The fourth type is ***sigmoid***. Modern NN's use the sigmoid nonlinearity which is also known as **logistic, semi linear, or squashing function.**

محصورة بين 0 و 1 وبها مرونة

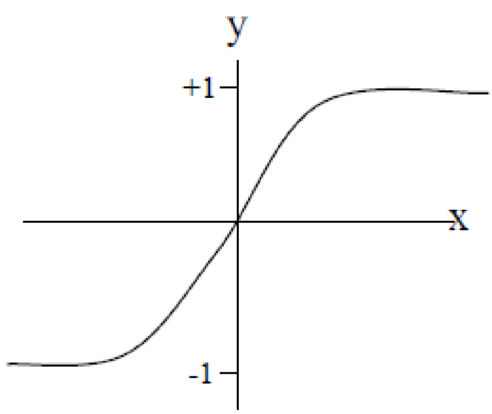
$$y_q = f_{bs}(v_q) = \frac{1}{1 + e^{-\propto v_q}}$$

$$y = \frac{1}{1 + e^{-x}}$$



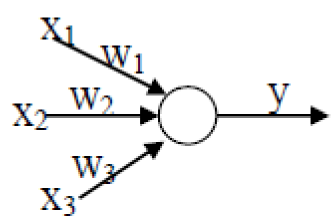
5-*Hyperbollc tangent* function is similar to sigmoid in shape but symmetric about the origin. (tan h)

$$y = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$



3.3 Examples:

Ex.1 find y for the following neuron if: $x_1=0.5$, $x_2=1$, $x_3= - 0.7$
$w_1=0$, $w_2=-0.3$, $w_3=0.6$



Sol

$$net = = X_1W_1 + X_2W_2 + X_3W_3$$

$$=0.5*0+1*-0.3+(-0.7*0.6)= -0.72$$

1- if f is linear

$$y = -0.72$$

2- if f is hard limiter (on-off)

$$y = -1$$

3-if f is sigmoid

$$y = \frac{1}{1+e^{-(-0.72)}} = 0.32$$

4-if f is tan h

$$y = \frac{e^{-0.72} - e^{0.72}}{e^{-0.72} + e^{+0.72}} = -0.6169$$

5-if f is (TLU) with b=0.6, a=3 then y=-3

$$f(y) = \begin{cases} a & y > b \\ ky & -b < y < b \\ -a & y < -b \end{cases} \quad \Bigg| \quad f(y) = \begin{cases} a & y > b \\ ky & 0 < y < b \end{cases}$$

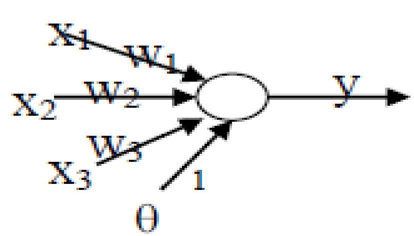
<u>Ex2:-</u> (H.W)

Find y for the following neuron if

$x_1 = 0.5,\ x_2 = 1,\ x_3 = -0.7$

$w_1 = 0,\ w_2 = -0.3,\ w_3 = 0.6$

$\theta = 1$



<u>Sol</u>

$$Net = \sum W_i X_i + \theta$$

$$= -0.72 + 1 = 0.28$$

1- if f is linear

$$y = 0.28$$

2- if f is hard limiter

$$y = 1$$

3-if f is sigmoid

$$y = \frac{1}{1 + e^{-0.28}} = 0.569$$

4-if f is tan sh

$$y = \frac{e^{0.28} - e^{-0.28}}{e^{0.28} + e^{0.28}} = 0.272$$

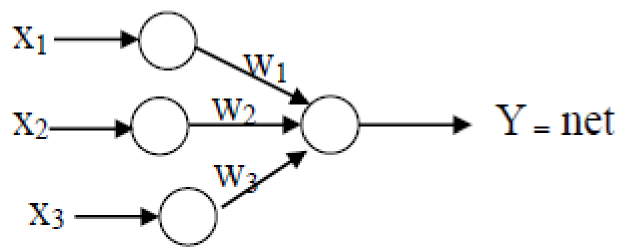
5-if f is TLU with b=0.6, +a=3

$y=0.28$ $\qquad$ $y \leftarrow -b < y < b$

Ex.3:

The output of a simulated neural using a sigmoid function is 0.5 find the value of threshold when the input $x_1 = 1$, $x_2 = 1.5$, $x_3 = 2.5$. and have initial weights value = 0.2.

<u>Sol</u>



$$Output = F\ (net + \theta\ )$$

$$F(net) = \frac{1}{1+e^{-net}}$$

$$Net \quad = \sum W_i X_i$$

$$= X_1 W_1 + X_2 W_2 + X_3 W_3$$

$$=(1*0.2)+(1.5*0.2)+(2.5*0.2) = 0.2 + 0.30 + 0.50 \ = 1$$

$$0.5 = \frac{1}{1+e^{-(1+\theta)}}$$

$$0.5\ (1+e^{-(1+\theta)}) = 1$$

$$0.5 + 0.5\ e^{-(1+\theta)} = 1$$

$$0.5\ e^{-(1+\theta)} = 0.5$$

$$e^{-(1+\theta)} = 1$$

$$-(1+\theta) = \ln\ 1 \ \Rightarrow\ -1\ -\theta\ = \theta\ \Rightarrow\ -\theta = 1 \Rightarrow\ \therefore\ \theta = -1$$

3.4 The Bias

قيمة ثابتة تضاف لتحسين التعلم

Some networks employ a bias unit as part of every layer except the output layer. This units have a constant activation value of 1 or -1, it's weight might be adjusted during learning. The bias unit provides a constant term in the weighted sum which results in an improvement on the convergence properties of the network.

A bias acts exactly as a weight on a connection from a unit whose activation is always 1. Increasing the bias increases the net input to the unit. If a bias is included, the activation function is typically taken to be:

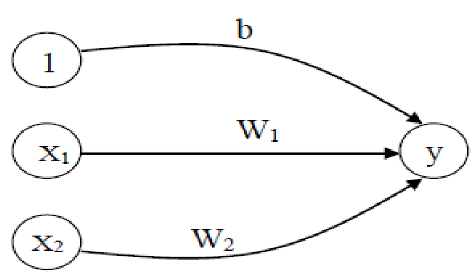
Same authors do not use a bias weight, but instead use a fixed threshold θ for the activation function.

$$f(net)\begin{cases} 1 & \text{if } net >= 0; \\ -1 & \text{if } net < 0; \end{cases}$$

Where

$$net = b + \sum_i X_i W_i$$

Figure: - single –layer NN for logic function



Input unit output unit

$$f(net)\begin{cases} 1 & \text{if} \quad net >= \ 0; \\ -1 & \text{if} \quad net < \ 0; \end{cases}$$

Where

$$net = b + \sum_i X_i W_i$$

However, this is essentially equivalent to the use of an adjustable bias.