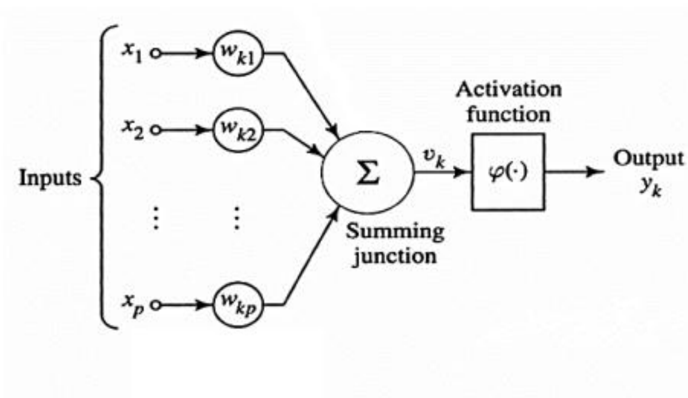
2-2-5-2. وحدات المعالجات (Processing Element) _ العصبونات

وحدات المعالجة أو العصبونات هي الوحدات التي تقوم بعملية معالجة المعلومات في الشبكة العصبية، وهي تشكل المكونات الأساسية التي تتألف منها كل طبقات الشبكة العصبية، وتتصل هذه الوحدات بطرائق مختلفة بوساطة الوصلات البينية لتعطي الشكل العام أو البنية المعمارية للشبكة العصبية الاصطناعية.

في إجراء الحسابات المسندة (Parallel Processing) تتبع عناصر المعالجة نظام المعالجة على التوازي، وهي في ذلك تشبه عمل العقل البشري، وتتألف أية وحدة معالجة أو عصبون من المكونات الأساسية الآتية:

1. معاملات الأوزان (Weighting Coefficients).
2. دالة الجمع (Summation Function).
3. دالة التحويل (Transfer Function) أو تابع التنشيط (Activation Function).
4. دالة المخرجات (Output Function).



الشكل (2-4): كيفية عمل العصبون الاصطناعي.

2-2-5-2-1. معاملات الأوزان (Weighting Coefficients)

يعتبر الوزن هو العنصر الرئيسي في الشبكات العصبية الاصطناعية، فهو يمثل الروابط المختلفة التي يتم عبرها نقل البيانات من طبقة إلى أخرى، ويعبر الوزن عن القوة والأهمية النسبية لكل مدخل بالنسبة لعنصر المعالجة، وتمثل الأوزان الوسيلة الأساسية لذاكرة الشبكة العصبية من خلال ضبط الأوزان ويرمز للوزن بين عنصري معالجة (i) و (j) بالرمز (wij).

2-2-5-2-2. دالة الجمع (Summation Function)

إن أول عملية تقوم بها وحدة المعالجة هي حساب مجموع المدخلات الموزونة القادمة إلى الوحدة باستخدام دالة الجمع، حيث تقوم هذه الدالة بحساب متوسط الأوزان لكل مدخلات وحدة المعالجة، ويتم ذلك بضرب كل قيمة مدخلة في وزنها المرافق ومن ثم إيجاد المجموع لكل نواتج الضرب.

ويعطى ذلك رياضياً وفق المعادلة (2-1):

$$S_i = \sum_{i=1}^{n} x_i * w_{ij} \quad \dots\dots\dots\dots\dots (1-2)$$

حيث:

- S_j : ناتج عملية الجمع لكل وحدة معالجة (j).
- x_i : القيمة المدخلة القادمة من الوحدة (i) والداخلة إلى الوحدة (j).
- w_{ij} : الوزن الذي يربط وحدة المعالجة (j) بالوحدة (i) الموجودة في الطبقة السابقة.

ويمكن أن تكتب المعادلة السابقة على الصيغة الآتية، المعادلة (2-2):

$$S_i = b_j + \sum_{i=1}^{n} x_i * w_{ij} \quad \dots\dots\dots\dots\dots (2-2)$$

حيث: b_j : يمثل معامل الانحياز (Bias)، ويعتبر أحد مكونات الدخل، ويأخذ دائماً القيمة الابتدائية $(x_0 = 1)$، وعمل الانحياز مشابه لعمل الأوزان لذلك يعامل معاملة أي وزن ويمكن أن يرمز له $(b_j = w_{oj})$، كما أن إضافة وحدة انحياز إلى وحدات الدخل يغير من شكل تابع التنشيط أو دالة التحويل.

كما يعبر نـها رياضيا بالصيغة التالية

$$net = \sum_{i=1}^{n} w_i \, x_i$$

2.1.1 Single Input Neuron:

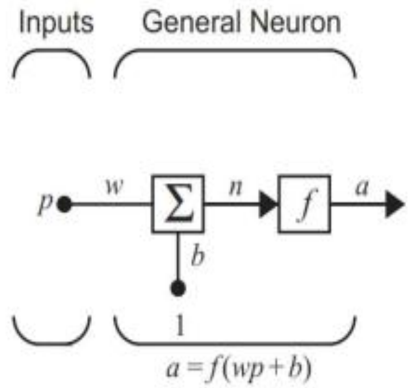


Figure 3: Single-Input Neuron

The neuron output is calculated as

$$a = f(wp + b)$$

Example 2.1:

Let $w = 3$, $p = 2$ and $b = -1.5$, what is the single-input neuron output ?

$$a = f\big(3 * 2 + (-1.5)\big) = f(4.5)$$

The actual output depends on the particular transfer function that is chosen.

2.1.3 Multiple Input Neuron:

Typically, a neuron has more than one input. A neuron with R inputs is shown in Figure 13. The individual inputs $\mathbf{p} = (p_1, p_2, p_3, \ldots, p_R)$ are each weighted by corresponding elements $w_{1,1}$, $w_{1,2}$, … , $w_{1,R}$ of the *weight matrix* $\mathbf{W}$.

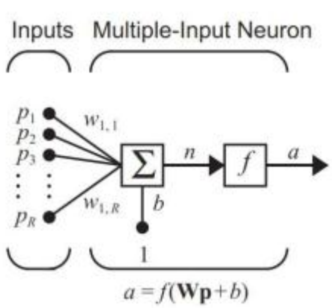


Figure 13: Multiple-Input Neuron

The neuron has a bias, which is summed with the weighted inputs to form the net input n :

$$n = w_{1,1}p_1 + w_{1,2}\, p_2 + \ldots + w_{1,R}\, p_R + b$$

This expression can be written in matrix form:

$$n = \mathbf{W}\mathbf{p} + b$$

where the matrix **W** for the single neuron case has only one row.

Example 3.1

- Input: $x_1 = 3,\ x_2 = 1\ ,\ x_3 = 2.$
- Weighting Coefficients: $w_1 = 0.2,\ w_2 = 0.4\ , w_3 = 0.4.$
- Summation Function: $y = 3 * 0.2 + 1 * 0.4 + 2 * 0.4 = 1.8.$

2.1.2 Transfer Functions (Activation Functions):

The transfer function in Figure 3 may be a linear or a nonlinear function of n. A particular transfer function is chosen to satisfy some specification of the problem that the neuron is attempting to solve.

There are variety of transfer functions some of them are listed below:

1. Threshold (Hard Limit) Transfer Function:

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

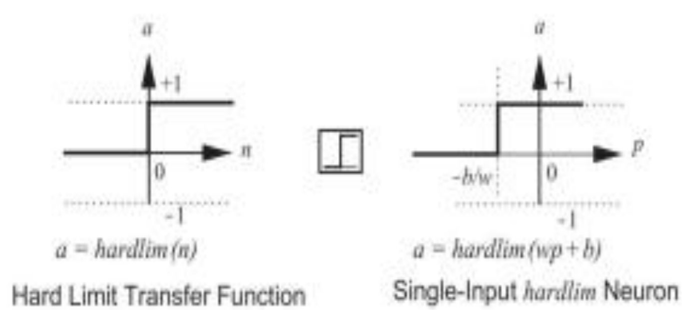


Figure 4: Hard Limit Transfer Function

This function will used to create neurons that classify inputs into two distinct categories.

2. Symmetric Hard Limit Transfer Function:

$$f(n) = \begin{cases} -1 & n < 0 \\ 1 & n \geq 0 \end{cases}$$

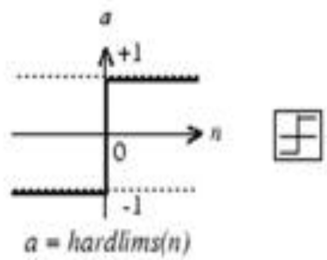


Figure 5: Symmetric Hard Limit Transfer Function

3. Linear Transfer Function:

$$f(n) = n$$

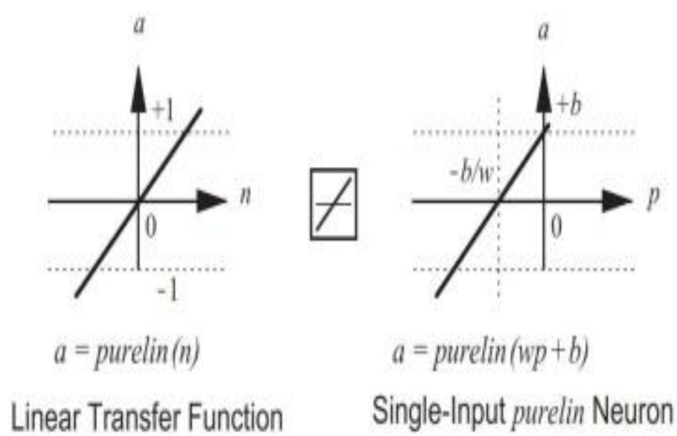


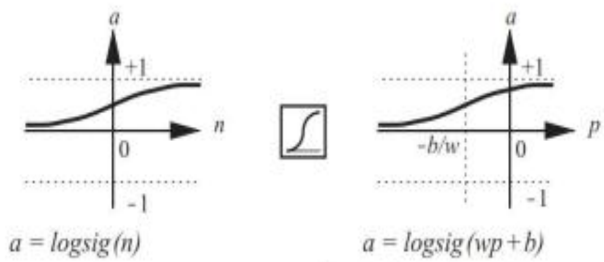
Figure 6: Linear Transfer Function

4. Logistic (Log-Sigmoid) Transfer Function:

Logistic function is a standard sigmoid function and is defined by

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

The derivative of f is defined by $f' = f\,(1-f)$



Log-Sigmoid Transfer Function   Single-Input *logsig* Neuron

Figure 7: Log-Sigmoid Transfer Function

The log-sigmoid transfer function is commonly used in multilayer networks that are trained using the backpropagation algorithm.

5. Hyperbolic Tangent Transfer Function:

The hyperbolic tangent is a sigmoid function and is defined by

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

The derivative of f is defined by $f' = (1-f^2)$

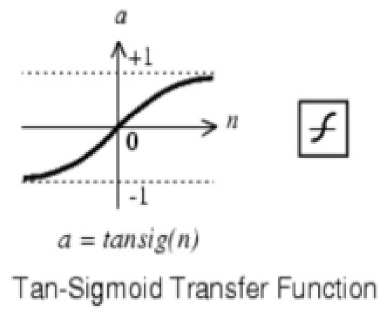


Figure 8: Hyperbolic Tangent Transfer Function

Since $\frac{\tanh\left(\frac{n}{2}\right)+1}{2} = \frac{1}{1+e^{-n}}$ then using the tanh function instead of the logistic one is equivalent. The tanh function has the advantage of being symmetrical with respect to the origin.

6. Radial Basis Transfer Function:

$$f(n) = e^{-n^2}$$

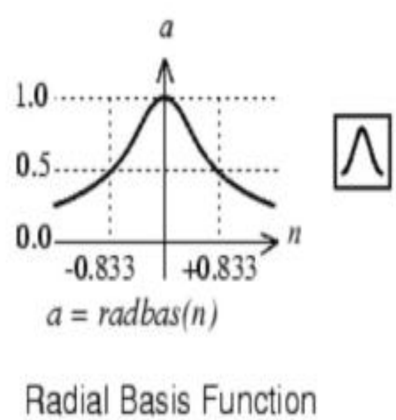


Figure 9: Radial Basis Transfer Function

7. Saturating Linear Transfer Function:

$$f(n) = \begin{cases} 0 & if \quad n < 0 \\ n & if \quad 0 \le n \le 1 \\ 1 & if \quad n > 1 \end{cases}$$

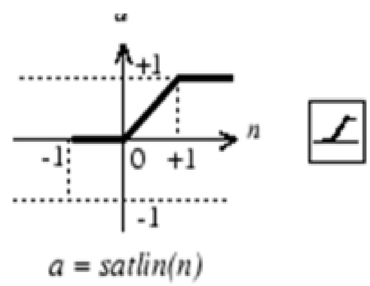


Figure 10: Saturating Transfer Function

8. Symmetric Saturating Linear Transfer Function:

$$f(n) = \begin{cases} -1 & if \quad n < -1 \\ n & if \quad -1 \le n \le 1 \\ 1 & if \quad n > 1 \end{cases}$$

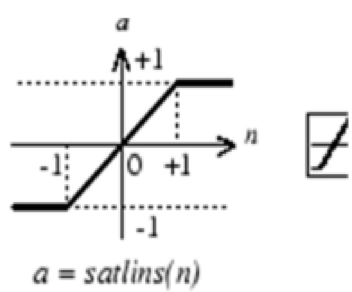


Figure 11: Symmetric Saturating Linear Transfer Function

9. Positive Linear Transfer Function:

$$f(n)=\begin{cases}0 & if \quad n<0\\ n & if \quad n\geq 0\end{cases}$$

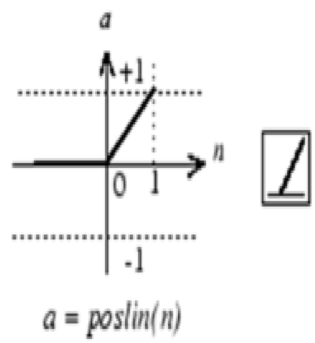
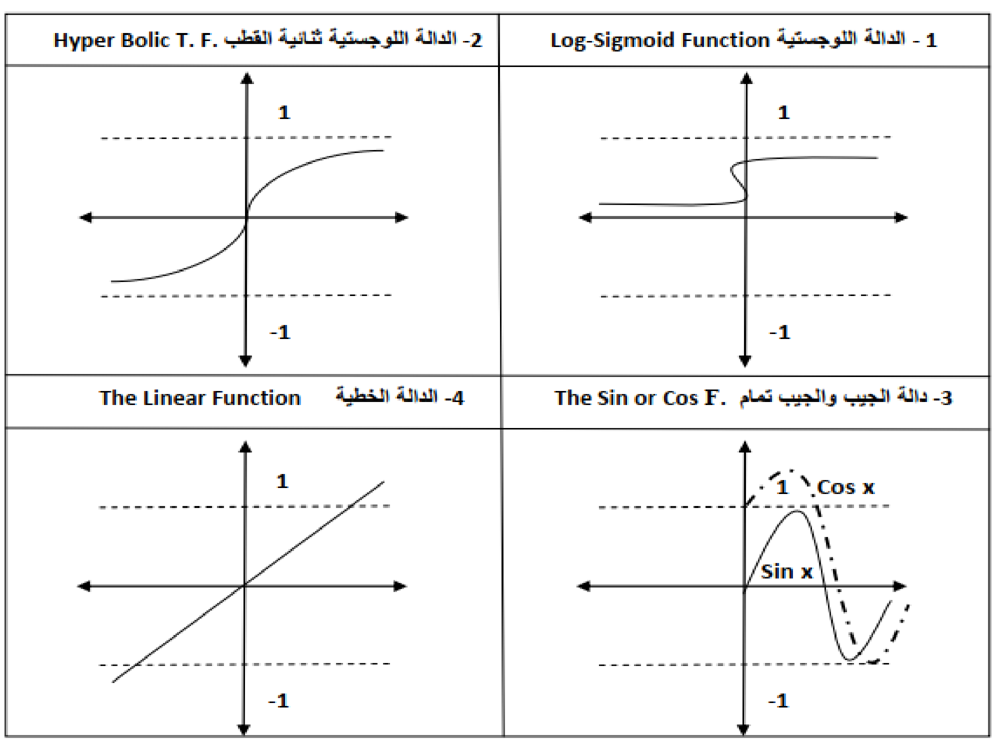
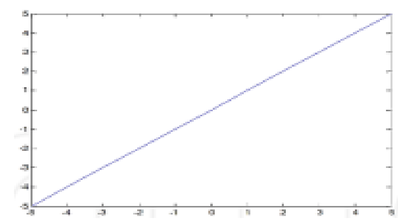
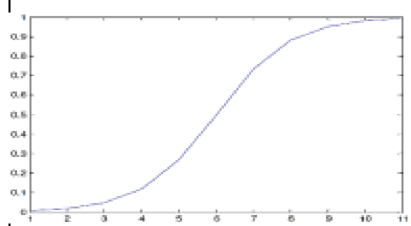
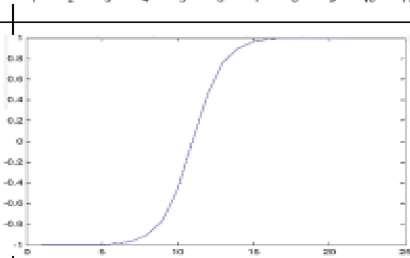


Figure 12: Positive Linear Transfer Function

الشكل (2-5) " يوضح اشكال دوال التفعيل "



الجدول (2−1): دوال التحويل ضمن بيئة برنامج (**Matlab**).

Function Name	**Graphical Illustration**	**Mathematical Form**
Linear (Purelin)		$f(x) = x$
Logistic Sigmoid (logsig)		$f(x) = \frac{1}{1 + e^{(-x)}}$
Hyperbolic Tangent Sigmoid (tansig)		$\mathrm{f(x)} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$