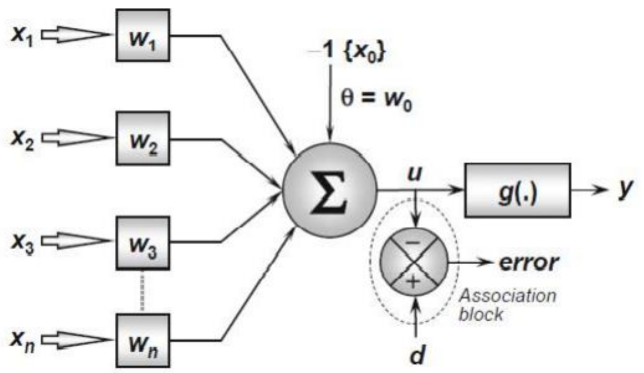
The ADALINE

د. ياسر خضرا

Adaptive Linear Neuron

- رغم كفاءة الشبكات العصبونية السابقة التي مرّت معنا إلا أنها لم تكن تحل جميع المشاكل وهذا أدى إلى تطوير الشبكة العصبونية بطريقة الAdaline حيث تم تصميمها وتطويرها في الستينيات وكانت من قبل مهندسي كهرباء، الذين استخدموها لحل بعض المشاكل التي واجهتهم، وقد أثبتو بالتجربة العملية أنها فعالة وقابلة للعمل.
- يكون لها شكل الشبكة وبنية العصبون نفسه، ويكمن الفرق بين الشبكة العصبونية بطريقة الAdaline والشبكات العصبونية الأخرى في كون هذه الطريقة تحوي تابع اختبار بعد تابع التفعيل.

شكل العصبون بالAdaline:



ونلاحظ إضافة قسم ال **error**.

ويشترك ال **Adaline** مع ال **perceptron** في كون قيم الخرج ثنائية وبالتالي هما طريقتان جيدتان للتصنيف الثنائي.

من أهم صفات طريقة ال Adaline:

- **تابع التفعيل:**

تستخدم توابع التفعيل الثنائية، ومن أشهر توابع التفعيل الثنائية المستخدمة هما تابع **Step** و **Bipolar** .

- **Supervised**

و تعني بأن الدخل و الخرج معلومين و يتم استخدامهم لتدريب الشبكة.

قاعدة التعلم Learning rule:

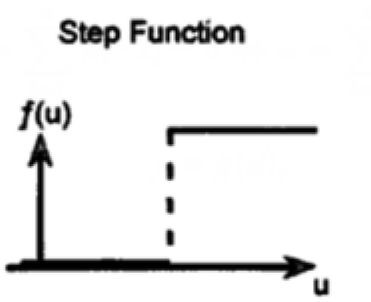
وتستخدم قاعدة دلتا المعرفة بالشكل:

$$u=\sum_{i=1}^{n} w_i . x_i-\theta \Leftrightarrow u=\sum_{i=0}^{n} w_i . x_i$$

$$y=g(u)$$

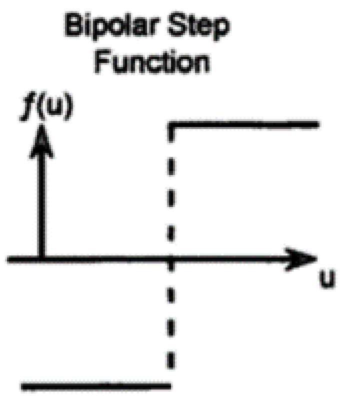
من المهم معرفة تابع التفعيل المستخدم قبل البدء بالتدريب وذلك لمعرفة كيفية وصف المشكلة بشكل يناسب التابع:

لو استخدمنا تابع ال **Step**



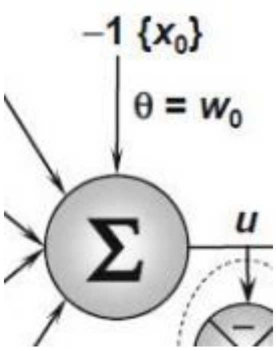
فإن القيم ستكون محصورة بين ال 0 وال 1

أما لو استخدمنا تابع ال **Bipolar**



فإن القيم ستكون بين ال -1 و ال 1

وبالنسبة للقاعدة العامة فقد اعتبرنا أن الانزياح هو أحد المداخل (x_0) وبالتالي تابع الجمع سيبدأ من الصفر (بعض المراجع يكون الانزياح موجود على تابع الجمع وبالتالي المجموع يبدأ من ال 1)



- ملاحظة مهمة:
- P: هو عدد عينات التدريب x^1

n : هو عدد الواصفات x_1

عدد عينات التدريب يعني كم مثال للاختبار و كل مثال اختبار يحوي n واصفة.

عملية تدريب ال Adaline:

- دائماً في خرج كل عملية تدريب يوجد نسبة خطأ يرمز له E ويعني الفرق بين القيمة المطلوبة والقيمة الناتجة، وكلما نتج خطأ فإن الشبكة بحاجة للتدريب.
- في طريقة ال perceptron لا يقبل الخطأ ولو بنسبة قليلة ويمكن اعتبار أن الخطأ موجود و لكن بقيمة 0. أما في الأدالاين فإن الخطأ مقبول ولكن بنسبة صغيرة قدر الإمكان وتعطى معادلة الخطأ المقبولة بالشكل:

$$E(w^*) \le E(w), for\ \forall w \in R^{n+1}$$

إن التابع E تابع ل W وذلك لإن الدخل ثابت والخرج ثابت والذي يتغير هو w وبالتالي فإن الخطأ يقع في الأوزان.

$E(w*)$: قيمة الخطأ المثالية المطلوبة والمتوقعة

$E(w)$: قيمة الخطأ الناتج بكل دورة

و بالتالي الخطأ يعتبر مقبول عندما تكون قيمة الخطأ أصغر أو تساوي القيمة المطلوبة.

- ملاحظة: لا نستمر بتصغير قيمة الخطأ حتى نصل لأصغر قيمة قد يصل إليها التابع، حيث أن الاستمرار بتصغير القيمة يمكن أن يستنزف الكثير من الجهد وقيمة الخطأ تنقص بمقدار قليل وغير مقبول وبهذه الحالة نتوقف.

- يعرف $E(w)$ بالعلاقة:

$$E(w) = \frac{1}{2}\sum_{k=1}^{p}\left(d^{(k)} - u\right)^2$$

$$E(w) = \frac{1}{2}\sum_{k=1}^{p}\left(d^{(k)} - \left(\sum_{i=1}^{n} w_i . x_i - \theta\right)\right)^2$$

$$E(w) = \frac{1}{2}\sum_{k=1}^{p}\left(d^{(k)} - (w^T . x^{(k)} - \theta)\right)^2$$

d_k: هي القيمة المتوقعة (المحسوبة).

u: هو التابع المعطى بالعلاقة:

$$\left(\sum_{i=0}^{n} w_i . x_i - \theta\right)$$

ولكن: $\sum_{i=0}^{n} w_i . x_i$ هي عملية ضرب مصفوفات حيث يمكن التعبير عنها بالشكل $W^T . x_i^{(k)}$

معلومة رياضية:
الرياضيات المستمرة هي تفاضل
الرياضيات المتقطعة هي فرق

والرياضيات المستخدمة في الشبكات العصبونية هي الرياضيات المتقطعة، لأنه كمثال لا نستطيع تحقيق التكامل بالبرمجة، إنما نقوم بالتقطيع والجمع.

- نقوم باشتقاق تابع الخطأ بالنسبة للأوزان:

$$\nabla E(w) = \frac{\partial E(w)}{\partial w}$$

ونسمي هذا التفاضل لتابع الخطأ بالتدرج.

وتعني العلاقة بأننا نحسب تأثر تغير تابع الخطأ بتغير الوزن w والذي نسعى لتصغيره قدر الإمكان.

- وباشتقاق العلاقة:

$$E(w) = \frac{1}{2} \sum_{k=1}^{p} \left(d^{(k)} - (w^T . x^{(k)} - \theta)\right)^2$$

ينتج التابع:

$$\nabla E(w) = \sum_{k=1}^{p} \left(d^{(k)} - \left(w^T . x^{(k)} - \theta\right)\right) . (-x^{(k)})$$

$$\nabla E(w) = \sum_{k=1}^{p} (d^{(k)} - u) . x^{(k)}$$

ولكن:

$$\Delta w = -\eta . \nabla E(w) \Rightarrow \Delta w = \eta \sum_{k=1}^{p} (d^{(k)} - u) . x^{(k)}$$

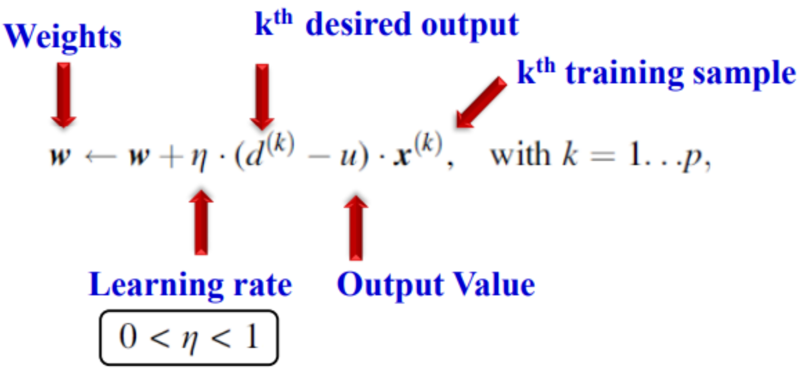
وهو الشكل النهائي للعلاقة، وبالتالي فإن معادلة تحديث الأوزان هي من الشكل:

$$w^{current} = w^{previos} + \eta . (d^{(k)} - u) x^{(k)}, \qquad with\ k = 1 \dots p$$

حيث أن η يسمى عامل التعلم وهو يحدد سرعة العملية ويكون عادة محصور بين ال 0 و 1

فلو كان صغير جداً فإن القفزة ستكون صغيرة ولو كان كبير القفزة ستكون كبيرة، ولذلك علينا اختيار قيمة مناسبة له.

Delta learning rule

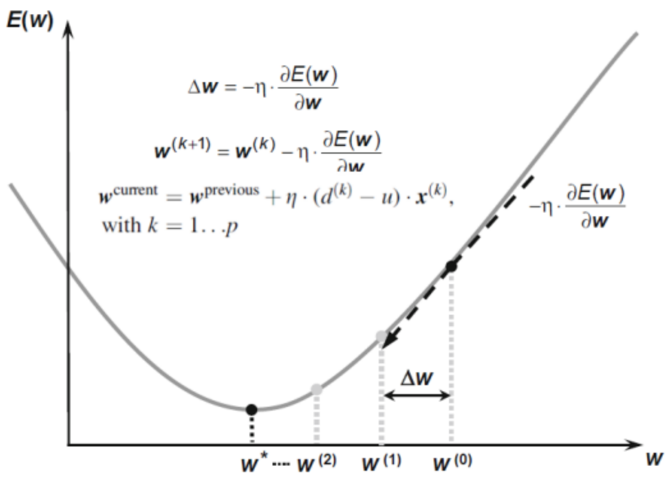


W : مصفوفة الاوزان

$x^{(k)}$: مصفوفة الدخل

U : القيمة الناتجة من التابع

η : عامل التعلم



كما هو موضح بالصورة فإن عملية التدريب تستمر حتى الوصول للقيمة الصغرى وتختلف سرعتها بحسب عامل التعلم كما تكلمنا سابقاً، وتتوقف عندما ينحدر التدرج للصفر ونقبل الخطأ بنسبة معينة حسب حدود الخطأ المحددة

$$\left|\bar{E}(w^{current}) - \bar{E}\left(w^{previos}\right)\right| \leq \varepsilon$$

خطوات تطبيق الAdaline بالقاعدة دلتا:

- الحصول على مجموعات التدريب
- الحصول على القيم المطلوبة وربط كل قيمة مع عينة التدريب الموافقة
- تعيين مصفوفة الأوزان W بقيم عشوائية صغيرة
- تعيين قيمة عامل التعلم η و تعيين الحد لقيمة الخطأ 3
- تعيين عدد حقب التدريب epoch
- تكرار: $< 6.1 > \bar{E}w^{previos} \leftarrow \bar{E}(w)$

وتعني اسناد القيمة الجديدة للخطأ الناتجة عن حقبة التدريب لقيمة الخطأ السابقة

- ثم نقوم بالمرور على جميع عينات التدريب وتعديل الأوزان

$$< 6.2 > For\ all\ training\ samples \left\{x^{(k)}, d^{(k)}\right\}, do:$$

$$< 6.2.1 > u \leftarrow w^T . x^{(k)};$$
$$< 6.2.2 > w \leftarrow w + \eta\left(d^{(k)} - u\right). x^{(k)};$$

نزيد عداد الحقبة:

$$< 6.3 > epoch \leftarrow epoch + 1$$

بعدها نقوم بحساب تابع الخطأ بالأوزان الجديدة واسنادها للخطأ الحالي

$$< 6.4 > \bar{E}^{current} \leftarrow \bar{E}(w)$$

بالعلاقة:

$$\bar{E}(w) = \frac{1}{p}\sum_{k=1}^{p}\left(d^{(k)} - u\right)^2$$

و هي ما تسمى MSE (Mean Squared Error) وتكون بالطريقة التالية:

Begin {MSE Algorithm}

<1> Obtain the number of training samples {p};

<2> Initialize variable $\bar{E}$ with zero value {$\bar{E} \leftarrow 0$};

<3> For all training inputs {$\boldsymbol{x}^{(k)}$, $d^{(k)}$}, do:

<3.1> $u \leftarrow \boldsymbol{w}^T \cdot \boldsymbol{x}^{(k)}$;

<3.1> $\bar{E} \leftarrow \bar{E} + (d^{(k)} - u)^2$;

<4> $\bar{E} \leftarrow \dfrac{\bar{E}}{p}$;

Begin {MSE Algorithm}

والهدف منها هو إيجاد قيمة الخطا الجديدة وحساب الفرق بينها وبين القيمة القديمة للخطأ وذلك لمعرفة أن كان هناك داعي للاستمرار (قيمة الخطأ غير مقبولة) أو التوقف (قيمة الخطأ لا يمكن أن تكون أصغر أو قيمة نقصان الخطأ صغيرة) ونلاحظ أن الفرق بالقيمة المطلقة حيث أننا لا نهتم بالقيمة وإنما بالفرق بين قيمة الخطأ السابق وقيمة الخطأ الجديد.

$$Until: \left|\bar{E}\left(w^{current}\right) - \bar{E}\left(w^{previos}\right)\right| \le \varepsilon$$

وبالتالي الشكل النهائي للخوارزمية:

Begin {ADALINE Algorithm – Training Phase}

<1> Obtain the training sample set $\{\boldsymbol{x}^{(k)}\}$;
<2> Associate the desired output $\{d^{(k)}\}$ for each obtained sample;
<3> Initialize the vector $\boldsymbol{w}$ with small random values;
<4> Specify the learning rate $\{\eta\}$ and the required precision $\{\varepsilon\}$;
<5> Initialize the epoch counter $\{epoch \leftarrow 0\}$;
<6> Repeat:
<6.1> $\bar{E}^{previous} \leftarrow \bar{E}(\boldsymbol{w})$;
<6.2> For all training samples $\{\boldsymbol{x}^{(k)}, d^{(k)}\}$, do:
<6.2.1> $u \leftarrow \boldsymbol{w}^T \cdot \boldsymbol{x}^{(k)}$;
<6.2.2> $\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta \cdot (d^{(k)} - u) \cdot \boldsymbol{x}^{(k)}$;
<6.3> $epoch \leftarrow epoch + 1$;
<6.4> $\bar{E}^{current} \leftarrow \bar{E}(\boldsymbol{w})$;
Until: $\left|\bar{E}^{current} - \bar{E}^{previous}\right| \leq \varepsilon$

End {ADALINE Algorithm – Training Phase}

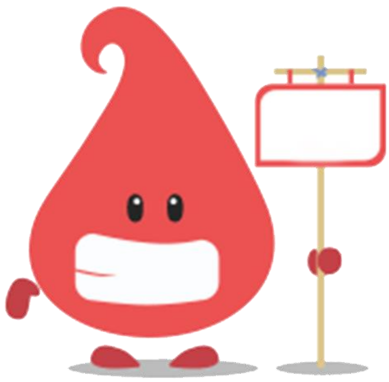
وبهذه الطريقة استطعنا إيجاد الأوزان التي تحقق أصغر نسبة خطأ.

مثال: شبكة للفرز الثنائي:

Begin {ADALINE Algorithm – Training Phase}

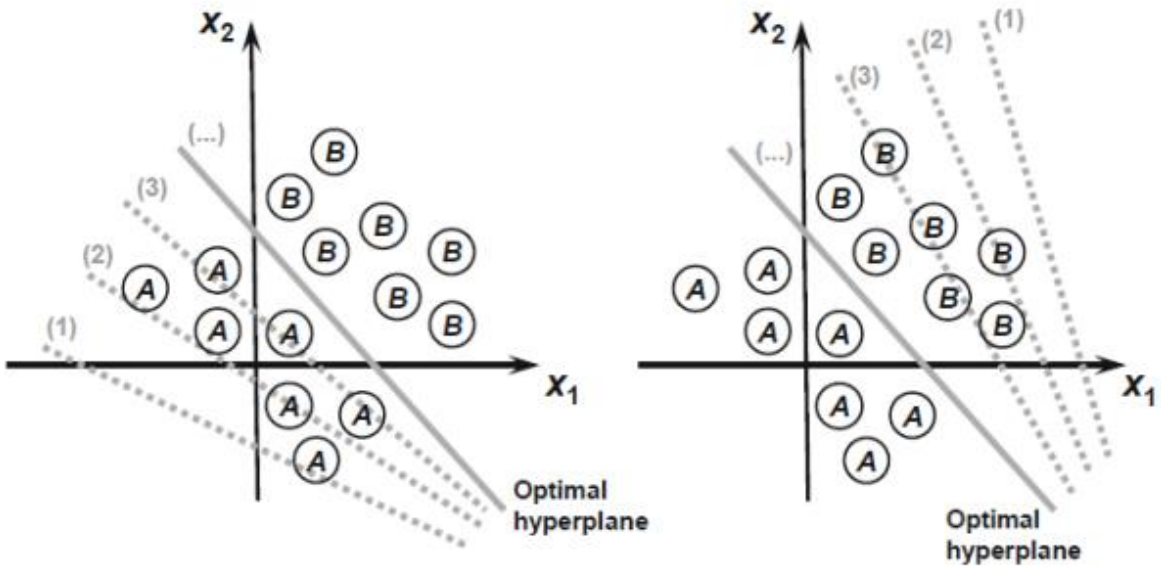
<1> Obtain the training sample set $\{\boldsymbol{x}^{(k)}\}$;
<2> Associate the desired output $\{d^{(k)}\}$ for each obtained sample;
<3> Initialize the vector $\boldsymbol{w}$ with small random values;
<4> Specify the learning rate $\{\eta\}$ and the required precision $\{\varepsilon\}$;
<5> Initialize the epoch counter $\{epoch \leftarrow 0\}$;
<6> Repeat:
<6.1> $\bar{E}^{previous} \leftarrow \bar{E}(\boldsymbol{w})$;
<6.2> For all training samples $\{\boldsymbol{x}^{(k)}, d^{(k)}\}$, do:
<6.2.1> $u \leftarrow \boldsymbol{w}^T \cdot \boldsymbol{x}^{(k)}$;
<6.2.2> $\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta \cdot (d^{(k)} - u) \cdot \boldsymbol{x}^{(k)}$;
<6.3> $epoch \leftarrow epoch + 1$;
<6.4> $\bar{E}^{current} \leftarrow \bar{E}(\boldsymbol{w})$;
Until: $\left|\bar{E}^{current} - \bar{E}^{previous}\right| \leq \varepsilon$

End {ADALINE Algorithm – Training Phase}



بعد تطبيق الخوارزمية وإيجاد الأوزان المناسبة وإنشاء الشبكة العصبونية سينتج لدينا خط القرار

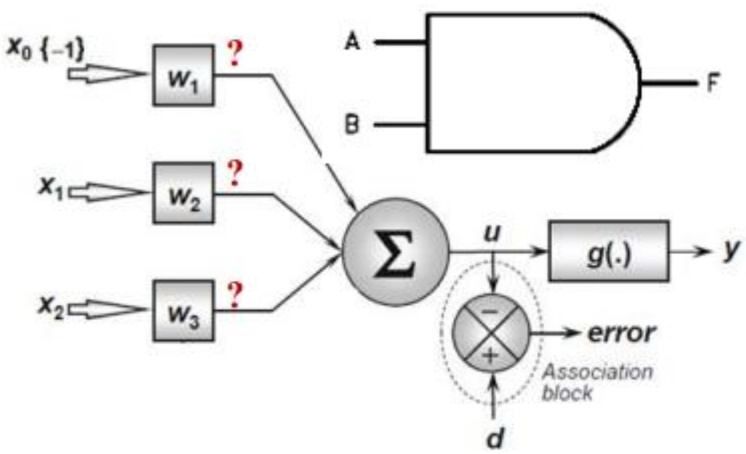
Decision Boundary وهو الذي يفصل الصنفين كما هو واضح بالشكل:



- الشكلين يوضح الخط الفاصل للأصناف ونلاحظ بالشكل اليميني أن الخط يتدرج من الأعلى إلى الأسفل حتى يصل للمنطقة المناسبة والأفضل
- وبالشكل اليساري فإن الخط يتدرج من الأسفل للأعلى حتى يصل لنفس المنطقة
- في كلتا الحالتين فإن الخط توقف في نفس المكان وبالتالي لا يهم فيما إذا كانت أصغر أو أكبر من القيمة المطلوبة وإنما المهم أن تكون قريبة منها.

دارة AND باستخدام ال ADALINE

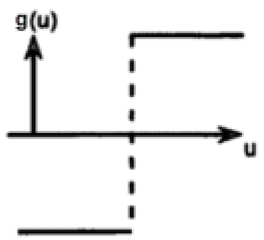
لدينا شكل الدارة :



ولدينا جدول الحقيقة المعبر عن دارة AND:

	x_0	x_1	x_2	d	
$x^{(1)}$	-1	-1	-1	-1	$d^{(1)}$
$x^{(2)}$	-1	-1	1	-1	$d^{(2)}$
$x^{(3)}$	-1	1	-1	-1	$d^{(3)}$
$x^{(4)}$	-1	1	1	1	$d^{(4)}$

سنستخدم تابع Bipolar كتابع تفعيل :



- سنستخدم قاعدة دلتا، بالبداية سنحصل على عينات التدريب والقيم المطلوبة من كل عينة وهو جدول الحقيقة
- سنقوم بربط كل عينة تدريب مع القيمة المطلوبة:

$d^{(1)}$	-1
$d^{(2)}$	-1
$d^{(3)}$	-1
$d^{(4)}$	1

	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	
x_0	-1	-1	-1	-1	$d^{(1)}$
x_1	-1	-1	1	1	$d^{(2)}$
x_2	-1	1	-1	1	$d^{(3)}$

- نقوم بتهيئة الأوزان بقيم عشوائية و لتكن $W^T = [1.0\ , 1.0\ , 1.0\ ]$
- نهيئ عامل التعلم، مقدار الفرق في قيمة الخطأ الناتج والذي سيعبرعن شرط التوقف، وعداد الحقبة، ولتكن بالقيم التالية:

$$\eta = 0.5\ , \varepsilon = 0.001, epoch = 0$$

- الآن سنقوم بإنشاء حلقة وتعديل الأوزان لتصغير نسبة الخطأ قدر الإمكان، سنحسب الخطأ الناتج بالبداية بالأوزان الأبتدائية

$$\bar{E}(w) = \frac{1}{p}\sum_{k=1}^{p}\left(d^{(k)} - u\right)^2 = \frac{1}{4}\sum_{k=1}^{4}\left(d^{(k)} - u\right)^2$$

وبالتالي:

$$u^{(1)} = w_0 * x_0^{(1)} + w_1 * x_1^{(1)} + w_2 * x_2^{(1)} = (1.0 * (-1) + 1.0 * (-1) + 1.0 * (-1)) = -3$$
$$u^{(2)} = w_0 * x_0^{(2)} + w_1 * x_1^{(2)} + w_2 * x_2^{(2)} = (1.0 * (-1) + 1.0 * (-1) + 1.0 * (1)) = -1$$
$$u^{(3)} = w_0 * x_0^{(3)} + w_1 * x_1^{(3)} + w_2 * x_2^{(3)} = (1.0 * (-1) + 1.0 * (1) + 1.0 * (-1)) = -1$$
$$u^{(4)} = w_0 * x_0^{(4)} + w_1 * x_1^{(4)} + w_2 * x_2^{(4)} = (1.0 * (-1) + 1.0 * (1) + 1.0 * (1)) = 1$$

بالتعويض:

$$\bar{E}(w) = 0.25\left(\left((-1)-(-3)\right)^2 + \left((-1)-(-1)\right)^2 + \left((-1)-(-1)\right)^2 + \left((1)-(1)\right)^2\right)=1$$

وبالتالي قيمة الخطأ الحالية هي 1 وسنقوم بحساب u وبعدها تعديل الأوزان ومن ثم حساب قيمة الخطأ القديمة واختبار إذا كانت مناسبة وعندها التوقف أو مازال بالإمكان تصغيرها وحينها سنستمر بالتكرار.

الحقبة الأولى:

في كل مرة نتبع القاعدة الخاصة بتعديل الأوزان وهي:

$$w \leftarrow w + \eta.\left(d^{(k)} - u\right).x^{(k)}, \quad with\ k = 1 \dots p$$

حيث

$$U^n = W^T.X^n$$

حيث(.) تدل على ضرب المصفوفات

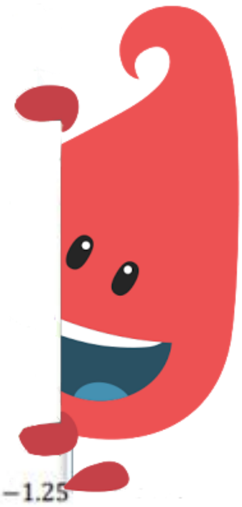
و بالتعويض بالقاعدة:

$$u^1 = w_0 * x_0^{(1)} + w_1 * x_1^{(1)} + w_2 * x_2^{(1)}$$

وبعد حساب u الجديدة نقوم بتعديل الأوزان

$$\bar{E}^{previous}(w) = \bar{E}(w) = 1$$

$x^{(1)}$	$u^{(1)} = w_0 * x_0^{(1)} + w_1 * x_1^{(1)} + w_2 * x_2^{(1)}$
-1	$= (1.0*(-1)+1.0*(-1)+1.0*(-1)) = -3$
	$w_0 = 1.0 + (0.5)(-1-(-3))(-1) = 0.0$
-1	$w_1 = 1.0 + (0.5)(-1-(-3))(-1) = 0.0$
-1	$w_2 = 1.0 + (0.5)(-1-(-3))(-1) = 0.0$
$x^{(2)}$	$u^{(2)} = w_0 * x_0^{(2)} + w_1 * x_1^{(2)} + w_2 * x_2^{(2)}$
-1	$= (0.0*(-1)+0.0*(-1)+0.0*(1)) = 0$
	$w_0 = 0.0 + (0.5)(-1-(0))(-1) = 0.5$
-1	$w_1 = 0.0 + (0.5)(-1-(0))(-1) = 0.5$
1	$w_2 = 0.0 + (0.5)(-1-(0))(1) = -0.5$
$x^{(3)}$	$u^{(3)} = w_0 * x_0^{(3)} + w_1 * x_1^{(3)} + w_2 * x_2^{(3)}$
-1	$= (0.5*(-1)+0.5*(1)+(-0.5)*(-1)) = 0.5$
	$w_0 = 0.5 + (0.5)(-1-(0.5))(-1) = 1.25$
1	$w_1 = 0.5 + (0.5)(-1-(0.5))(1) = -0.25$
-1	$w_2 = -0.5 + (0.5)(-1-(0.5))(-1) = 0.25$
$x^{(4)}$	$u^{(4)} = w_0 * x_0^{(4)} + w_1 * x_1^{(4)} + w_2 * x_2^{(4)}$
-1	$= (1.25*(-1)+(-0.25)*(1)+(0.25)*(1)) = -1.25$
	$w_0 = 1.25 + (0.5)(1-(-1.25))(-1) = 0.125$
1	$w_1 = -0.25 + (0.5)(1-(-1.25))(1) = 0.875$
1	$w_2 = 0.25 + (0.5)(1-(-1.25))(1) = 1.375$



$$w_0 = 1.0 + (0.5) - 1 - (-3)(-1) = 0.0$$
$$w_1 = 1.0 + (0.5) - 1 - (-3)(-1) = 0.0$$
$$w_2 = 1.0 + (0.5) - 1 - (-3)(-1) = 0.0$$

وينتج لدينا:

$epoch = 1$

$$
\begin{aligned}
u^{(1)} &= -2.375 \\
u^{(2)} &= 0.375 \\
u^{(3)} &= -0.625 \\
u^{(4)} &= 2.125 \\
\bar{E}(w) &= 1.29688 \\
\bar{E}^{current}(w) &= \bar{E}(w) = 1.29688 \\
\left|\bar{E}^{current}(w) - \bar{E}^{previous}(w)\right| &= 0.296875 > \varepsilon
\end{aligned}
$$

فقيمة الخطأ غير مقبولة وبالتالي سنستمر بتعديل الأوزان عدة دورات أخرى.

و بعد 17 حقبة تدريب سينتج لدينا الأوزان التالية:

Number of Training Iterations 17

$$W^T = [0.500042\,, 0.999978\,, 1.50002\,]$$

ويمكن تقريبها لتصبح:

$$W^T \approx [0.5\,, 1.0\,, 1.5\,]$$

وهي أوزان جيدة نستطيع بناء شبكة عصبونية لدارة **AND** بناء عليها.

> - ملاحظة: التكرار لا يكون للنهاية ! وإنما بعدد epoch معين وذلك لتفادي اللانهائية والتوقف عند حد معين حتى من دون الوصول للحل المطلوب، وفي هذه الحالة تكون هناك مشكلة سواء الأوزان أو القيم المطلوبة أو قيم الخطأ المتوقع والخ...

و يمكن التجريب على كود مكتوب بلغة بايثون للفهم أكثر (حاول تغيير قيم عامل التعلم والأوزان وملاحظة التغيير في عدد الحقب التي احتاجها التدريب للوصول للأوزان الأفضل)، ويمكن الممارسة عليه لتعلم استخدام مكتبة **numpy** التي تساعدنا في تدريب العصبونات.

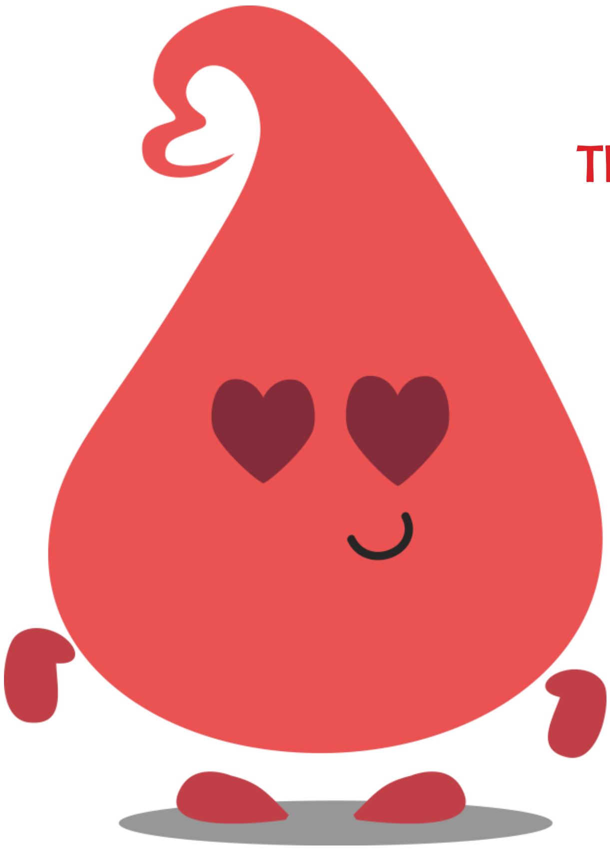
الكود:

رابط موقع لقراءة كود ال **QR**:

https://4qrcode.com/scan-qr-code.php

الشبكات العصبونية

إن هدفنا من تعلم تدريب العصبون ليس الحصول على عصبون واحد فقط، وإنما انشاء شبكات عصبونية متكاملة قادرة على القيام بمهمات صعبة ومعقدة فالعصبون هو الوحدة الأساسية للشبكة العصبونية، وسنتعلم بالمحاضرة القادمة كيفية إنشاء الشبكات العصبونية وبناء الأساس لها ودراسة النظريات والطرق الأفضل لتدريبها واستعمالها في حل المشاكل.



The key to artificial intelligence has always been the representation.

- Jeff Hawkins-

The end