



# الشبكات العصبية الاصطناعية NEURAL NETWORKS

د. عادل عبدالنور  
قسم الهندسة الكهربائية  
جامعة الملك سعود



- من المفارقات التي تدعوا للافهاش أن سرعة الحاسوب تفوق سرعة الخلية العصبية بـ 10 مليار مرة ( $10^7$ )
- مع هذا ، بإمكان الشخص العادي التعرف على وجه مألوف في عشرٍ من الثانية مستعملاً خلايا عصبية لا تزيد سرعتها على  $1/1000$  من الثانية.
- بتأمل هذه الأرقام نستنتج أن الحد الأقصى للخطوات التي تتبعها الخلايا لا تزيد عن 100 أي ( $1000/10$ ) بأي حال من الأحوال.



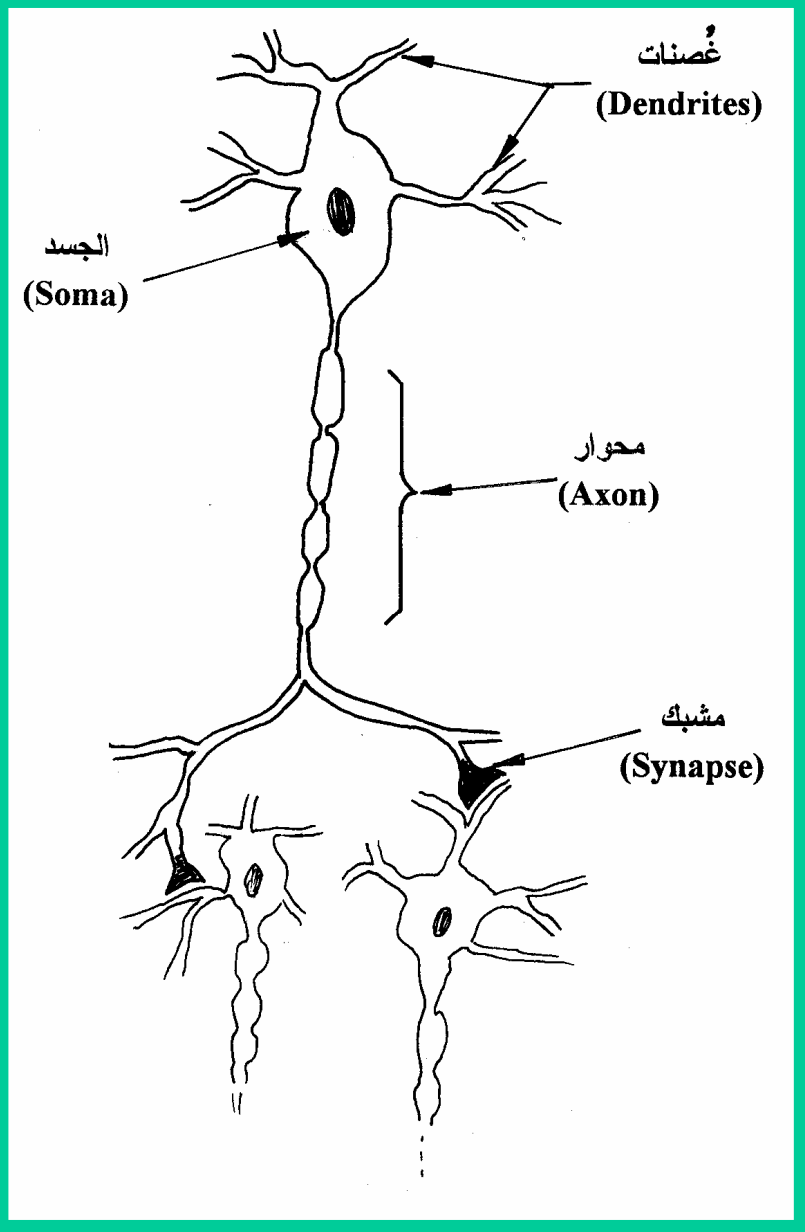
■ فكيف تتمكن خلايا بطيئة (مقارنة بالحاسوب) من الوصول إلى حلول بسرعة عالية ؟

■ لم يتوصل العلماء إلى تفسير مقنع ومنطقي سواء أن يكون سرّ قوة هذه الشبكات في طريقة معالجتها للبيانات (Information Processing)

■ فهذه الخلايا العصبية تعالج بياناتها بالتوازي (Parallel) مما يكسبها سرعة فائقة

■ لقد كانت هذه النتيجة كافية لإغراء الكثير من الباحثين بمحاولة " تقليد " الشبكات العصبية البشرية باستعمال الحاسوب





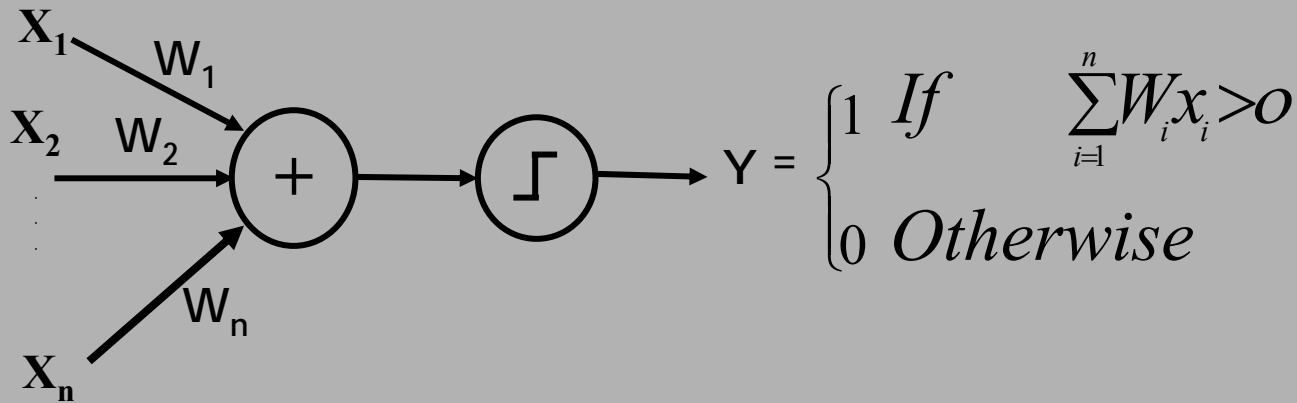
- رغم أن المعلومات عن الخلايا العصبية غير مكتملة وما زلنا نجهل الكثير عنها ، إلا أن القليل مما نعرفه كاف لإقناعنا بتعقيدها الشديد
- فدماع الإنسان العادي يحتوي على ألف مليار ( $10^{12}$ ) خلية عصبية
- لكل واحدة منها ما يقارب 10000 مشبك مع الخلايا الأخرى
- فمجرد محاولة تخيل هذه الشبكة العجيبة ليس بالأمر السهل



# الخلية العصبية الاصطناعية (Artificial Neuron)

- اتضح أنه بالإمكان تبسيط مكونات الخلية والاكتفاء مبدئياً ببعض وظائفها واستعمال عدد صغير منها ومن ثمّ تمثيلها "رياضياً" للحصول على خلايا عصبية اصطناعية
- يوضح الشكل التالي أبسط تمثيل للخلية العصبية الاصطناعية وهو عبارة عن وحدة معالجة (Processing Unit) أطلق عليها اسم المدرك (Perceptron)





■ باستعمال هذا التمثيل المبسط ، يمكن لوحدة المدرك أن  
تتمذج معظم العمليات المنطقية كعمليات:

- "أو" (OR)
- "و" (AND)
- "لا - و" (NAND)
- "لا - أو" (NOR)



■ للوصول إلى ذلك تحتاج وحدة المدرك إلى تدريب

■ التدريب هو إعطاء الوحدة بعض المدخلات وبعض المخرجات المرغوبة لتتمكن من تعديل الأوزان حسب القاعدة التالية:

$$W_i \text{ new} = W_i \text{ old} + a (O_{\text{desired}} - O) x_i$$

■ حيث :

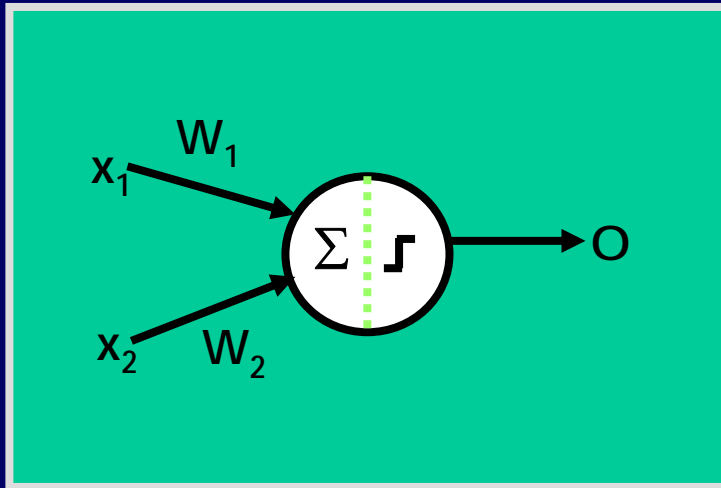
$W_i$  = الأوزان  
 $O$  = المخرج  
 $x_i$  = المدخل  
 $a$  = سرعة التعلم





■ كمثال على ذلك، لنفترض أننا نودّ تدريب وحدة المدرك  
نمذجة العملية المنطقية "أو" (OR)

■ لنبدأ عشوائياً باختيار الأوزان لتكون صفراً في البداية  
وتكون قيمة سرعة التعلم 1



$x_1$	$x_2$	$O_{\text{desired}}$
0	0	0
0	1	1
1	0	1
1	1	1



■ في هذه حالة يمكن كتابة المعادلة على الشكل التالي:

$$W1 \text{ new} = W1 \text{ old} + (O \text{ desired} - O) \times 1$$

$$W2 \text{ new} = W2 \text{ old} + (O \text{ desired} - O) \times 2$$

■ لتدريب الوحدة ، تطبق المعادلة على كل المدخلات مع تكرار العملية حتى تستقر الأوزان عند قيم محددة لا تتغير بعدها

■ ويوضح الجدول التالي خطوات التدريب



$x_1$	$x_2$	$W_{1old}$	$W_{2old}$	$O_{desired}$	$O$	error	$W_{1new}$	$W_{2new}$
0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	1	0	1
1	0	0	1	1	0	1	1	1
1	1	1	1	1	1	0	1	1
0	0	1	1	0	0	0	1	1
0	1	1	1	1	1	0	1	1

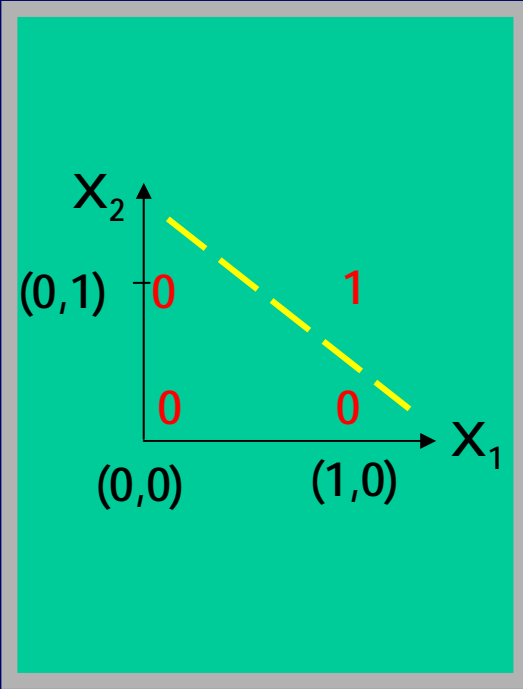
من خلال الجدول نلاحظ أن الأوزان النهائية هي :  $W_2 = 1 = W_1$



■ لقد أظهرت وحدة المدرك نجاعة في حلّ هذه المشكلات البسيطة والتي تُعنى بالتصنيف (Classification) و الفصل (Separation)

■ فعملية "أو" هي عبارة عن وضع حدّ فاصل Boundary بين مجموعتين مفصولتين أصلاً

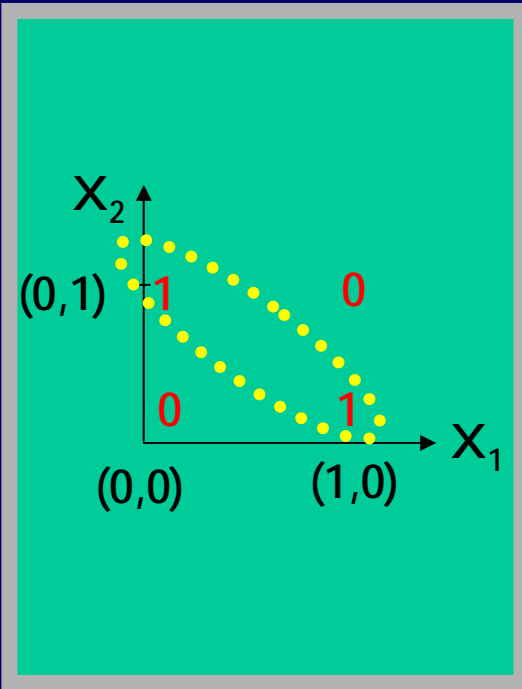
■ وعملية التدريب لا تتعدى البحث عن هذا الخط الفاصل عن طريق اختيار أوزان الترابط



■ هذه الوحدة غير قادرة على نمذجة  
عملية "أو المقصورة" XOR

■ فهذه العملية غير قابلة للفصل

■ الخطي !!!!!

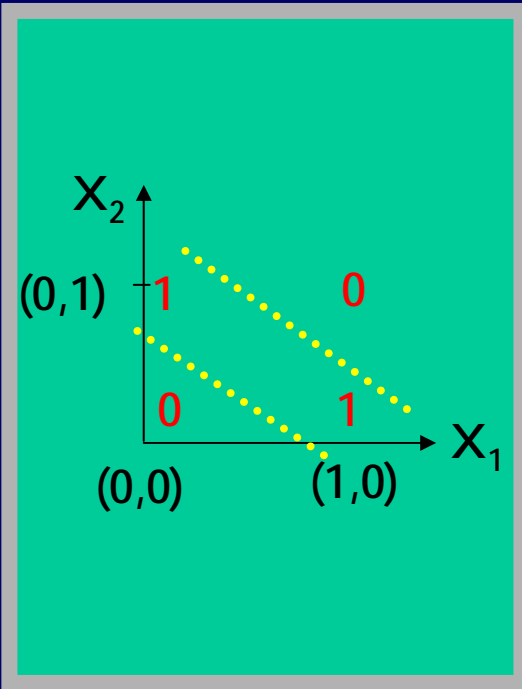


خيبة امل ؟؟؟؟



■ مواصلة البحث في الميدان توصلت إلى أنه بإمكان فصل عملية "أو المقصورة" بطريقة لا خطية

■ وبذلك ظهرت فكرة استعمال **أكثر** من خلية عصبية واحدة لتكوين شبكة متكاملة من هذه الخلايا



# بُنية الشبكات العصبية الاصطناعية

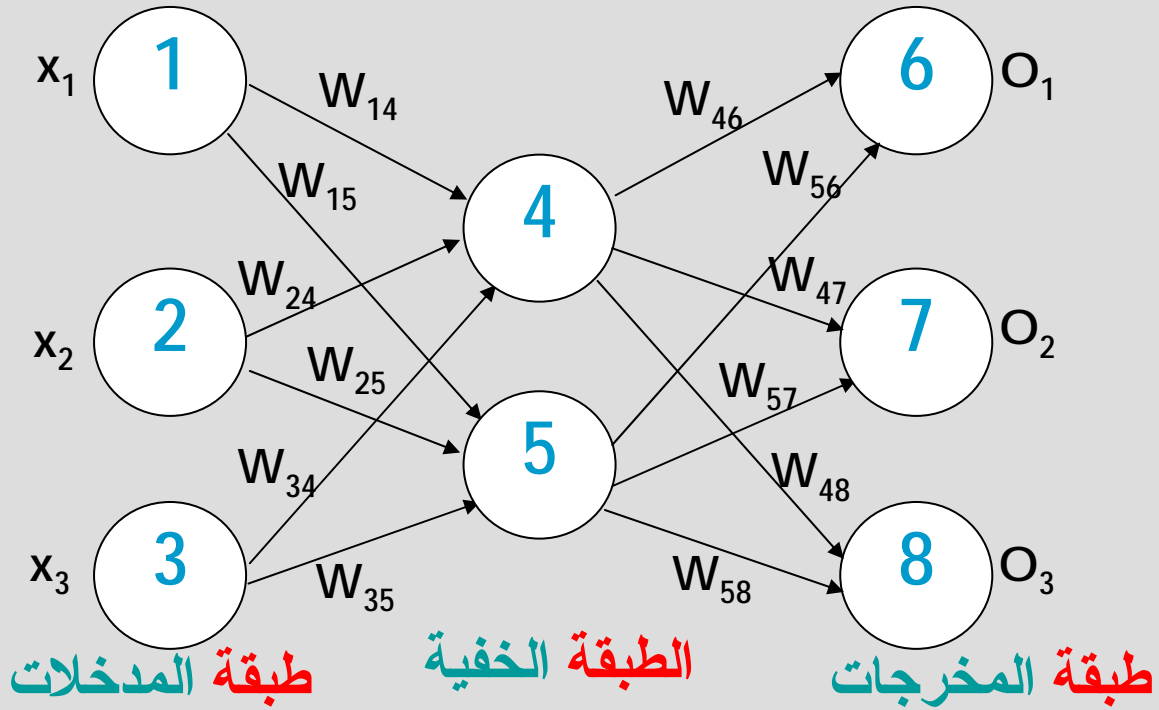
- بعد اكتشاف عدم نجاعة استعمال الخلية الواحدة ، ظهرت فكرة استعمال شبكة كاملة من الخلايا العصبية الاصطناعية
- تم اقتراح أنواع متعددة من هذه الشبكات ، أهمها الشبكات ذات التغذية الأمامية (Feed forward networks)
- في هذه البنية يسمح للإشارة بالتنقل فقط إلى الأمام من المدخلات إلى المخرجات
- فمخرجات أي من الطبقات لا تؤثر إلا في الطبقة التي تليها كما لا يوجد أي ترابط بين خلايا الطبقة الواحدة



- معظم الشبكات التي تتبع هذا النمط تتكون من طبقة المدخلات (Input Layer) وطبقة المخرجات (Output Layer) وهما الطبقتان الوحيدتان اللتان لهما اتصال بالمحيط خارج الشبكة
- إضافة إلى هاتان الطبقتان تضم الشبكة – على الأقل – طبقة خفية (Hidden Layer) وسميت كذلك لأنها لا تتصل بالمحيط خارج الشبكة ومرتبطة فقط بالطبقة التي تسبقها والتي تليها
- وتتكون كل من هذه الطبقات الثلاثة من عدد من الخلايا العصبية والتي يرمز لها بدائرة للتبسيط







مثال على الشبكات ذات التغذية الأمامية



- إذا ما أخذنا كل خلية من الشبكة على حدة ، فإن العمليات التي تقوم بها هذه الخلية هي كالتالي تقوم بها الوحدة المدركة والتي تطرقنا إليها سابقاً مع اختلاف بسيط ولكنه مهم
- ففي الخلية المدركة كانت المعالجة تتم عن طريق دالة "العتبة" (Threshold) وهذه الدالة ثنائية القيمة وهذا لا يسمح إلا بنمذجة العمليات الخطية
- لهذا، حتى تتمكن الشبكات العصبية من نمذجة الأنظمة والعمليات اللاخطية والمعقدة فلا بد من إيجاد دالة أو دالات أخرى غير دالة العتبة



# الشبكات العصبية المراقبة

## Supervised Neural Networks

- تسمى هذه الشبكات بالشبكات العصبية المراقبة لأنها تحتاج أثناء عملية التدريب إلى مراقب ليبين لها المخرج المطلوب لكل من المدخلات
- مع مقارنة المخرج الفعلي مع المخرج المطلوب تحاول الشبكات تحديث الأوزان (Weights) لتصل في النهاية إلى النتيجة الصحيحة والتي لا تحتاج بعدها لا إلى التدريب ولا إلى هذا الإشراف الخارجي



■ وتتم هذه العملية باستعمال العديد من الطرق والخوارزميات  
أهمها طريقة "انتشار الخطأ الارتدادي"  
(Error Back Propagation)

■ تبدأ العملية بإيجاد الخطأ بين المخرج المطلوب والفعلي  
وترجع بهذا الخطأ ارتدادياً من الطبقة الأخيرة إلى الطبقات  
الخفية ثم أخيراً إلى طبقة المدخلات

■ وفي أثناء هذا الارتداد يتم تغيير الأوزان في الاتجاه الذي يدفع  
بالخطأ إلى النقصان ومن ثمة في اتجاه الصفر



- وتستعمل هذه الطريقة التدريبية مع الشبكات ذات التغذية الأمامية (Feedforward Networks)
- حتى لا يقع أي التباس نحتاج إلى توضيح أن وصف:

- "التغذية الأمامية" يرجع إلى بنية الشبكة
- و "انتشار الخطأ الارتدادي" يرجع إلى طريقة التدريب التي تستعمل مع هذه الشبكة



# دالات التنشيط Activation Functions

- حتى تتمكن الشبكات المراقبة من نمذجة المشكلات اللاخطية والمعقدة لابد من تطوير الطريقة التي تعالج بها الخلية البيانات
- فعوضاً عن دالة العتبة والتي يكون مخرجها واحداً أو صفراً تحتاج خلايا هذه الشبكات إلى دالة أكثر ملاءمة
- وأهم المواصفات التي يجب توفرها في هذه الدالة هي:
  - أولاً: أن لا تكون ثنائية القيمة لتكون قادرة على النمذجة اللاخطية
  - ثانياً: أن تكون قابلة للتفاضل (Differentiation) وهذا شرط أساسي تفرضه طريقة الانتشار الارتدادي للتدريب



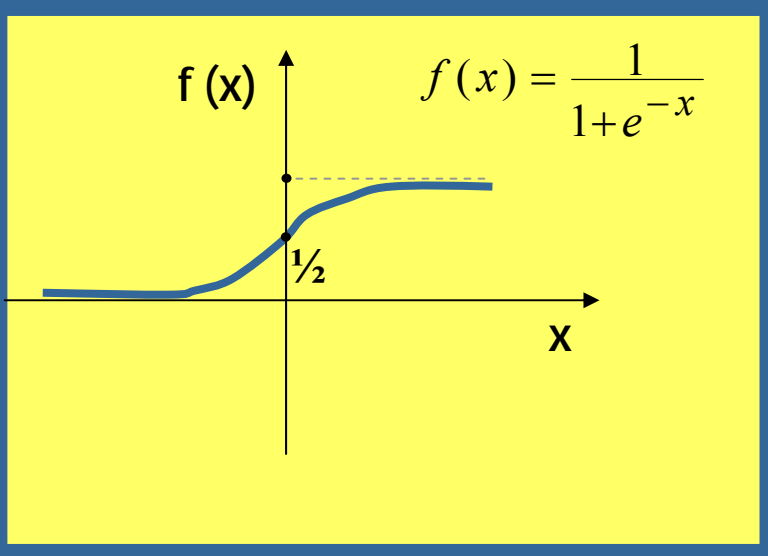
■ ولأهمية هذه الدالة في معالجة البيانات ، سميت بالدالة التنشيطية أو الدالة التفعيلية (Activation Function) لأنها تمثل (رياضيا) وصفاً لفعل الخلية (فعل منع أو تحريض)

■ وتوجد أنواع مختلفة من دالات التنشيط منها:

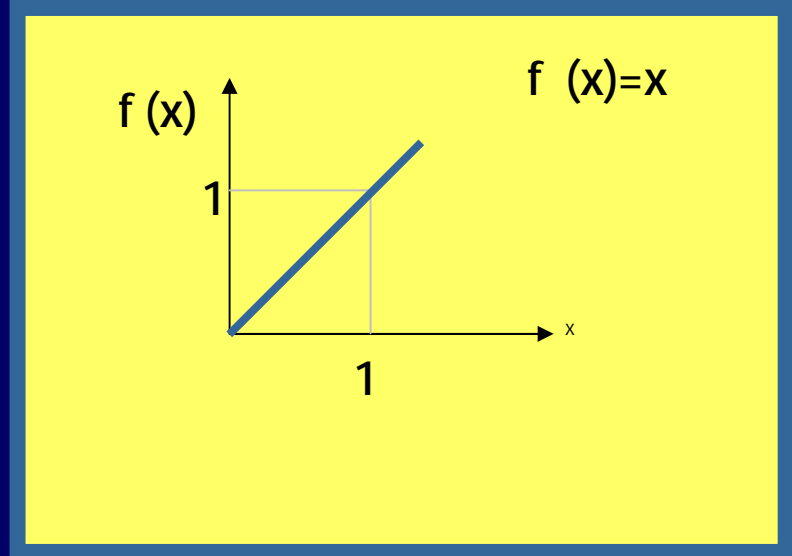
■ **الدالة الخطية** وتستعمل عادة في خلايا طبقة المخرجات

■ **الدالة السجماوية** وتستعمل في الطبقات الأخرى





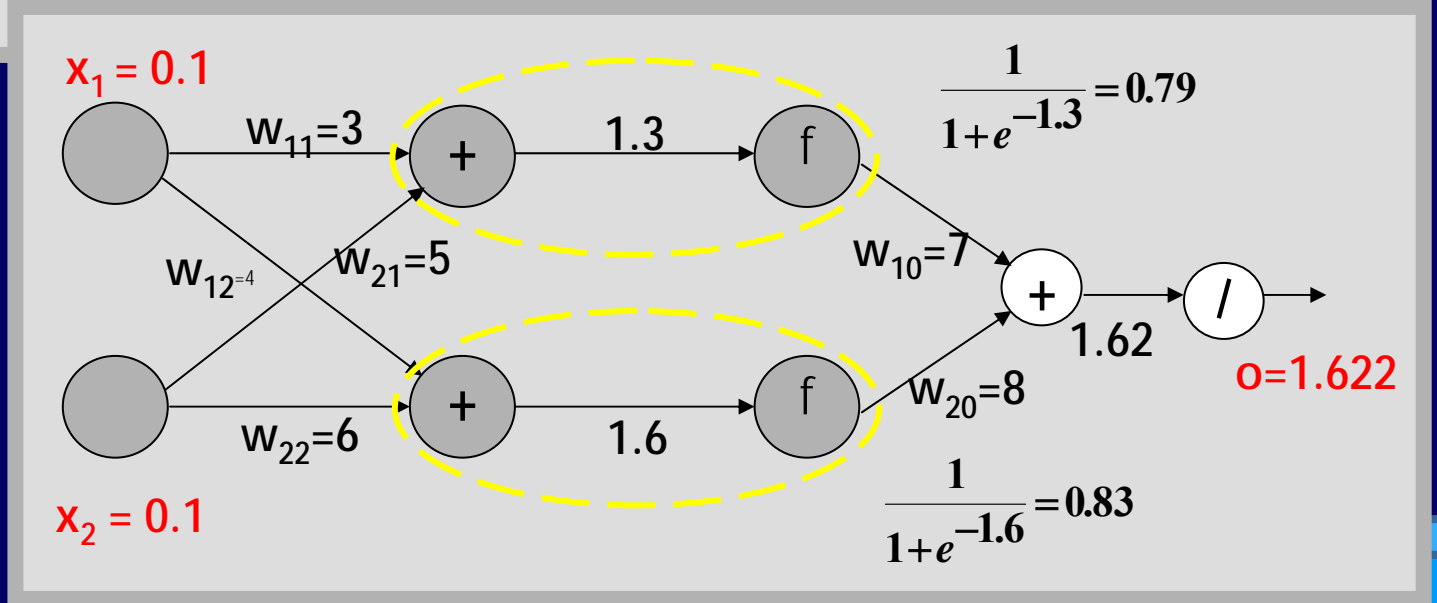
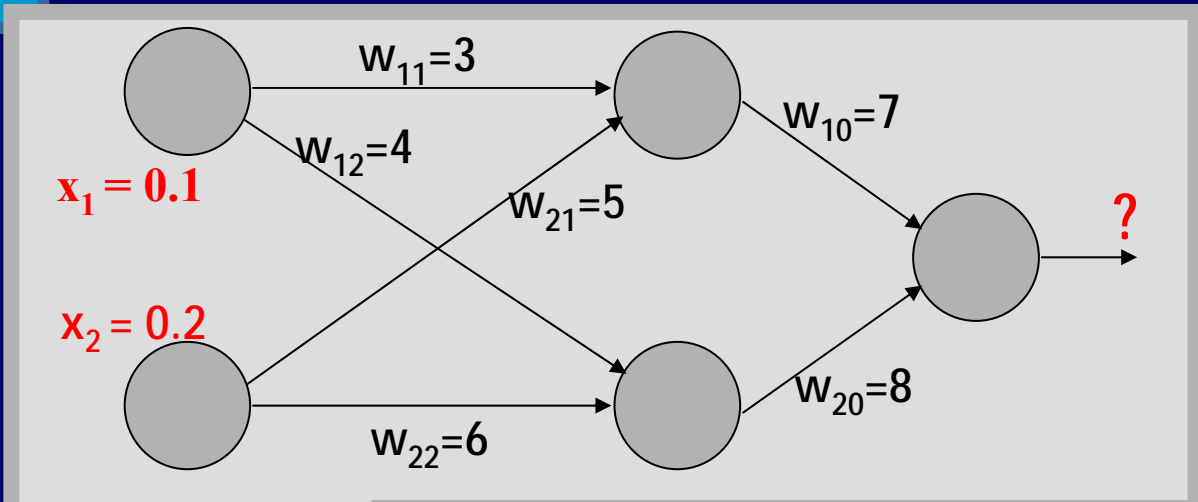
الدالة السجماوية



الدالة الخطية







# خوارزمية التدريب (Training Algorithm)

- هناك عدد كبير من الخوارزميات أكثرها استعمالاً هي خوارزميات الانتشار الارتدادي (Back propagation)
- هذه الخوارزميات قد تبدو معقدة كما سنرى لكنها في الواقع أسهل بكثير (على مستوى الفهم والبرمجة) مما تبدو.
- هي في الأصل اشتقاقاً من العملية المعروفة والمسماة "متوسط المربعات الأدنى" (Least Mean Squares) وبالتالي فهي تسعى لتقليل مربع الخطأ عند التدريب



■ سنفترض أن الخلايا العصبية الاصطناعية تستعمل الدالة السجماوية كدالة تنشيط لأنه من السهل جداً اشتقاقها كما هو موضح في المعادلة التالية:

$$\frac{df(x)}{dx} = f(x)[1 - f(x)] \quad f(x) = \frac{1}{1 + e^{-x}}$$

■ وتتمتع عملية الانتشار الارتدادي بميزات عدة أهمها:

- ✱ ضمان الحد الأدنى لمتوسط مربع الخطأ
- ✱ قابليتها للتعامل مع البيانات المشوشة
- ✱ وقدرتها على التعامل مع الأنظمة والدالات اللاخطية وغير القابلة للفصل الخطي



وتتكون هذه العملية من ست خطوات أساسية وهي:

- إعطاء أوزان عشوائية للترابط بين خلايا الشبكة
- مدّ الشبكة بإحدى المدخلات المعدة للتدريب
- تطبيق عملية الانتشار الأمامي لتحديد مخرجات الشبكة
- مقارنة المخرجات الفعلية مع المخرجات المطلوبة وتحديد قيمة الخطأ
- التراجع بالخطأ عبر الشبكة وتصحيح الأوزان في الاتجاه الذي يضمن تصغير قيمة الخطأ ومن هنا جاءت تسمية **الانتشار الارتدادي**
- تصغير إجمالي الخطأ لكل المدخلات المستعملة في التدريب

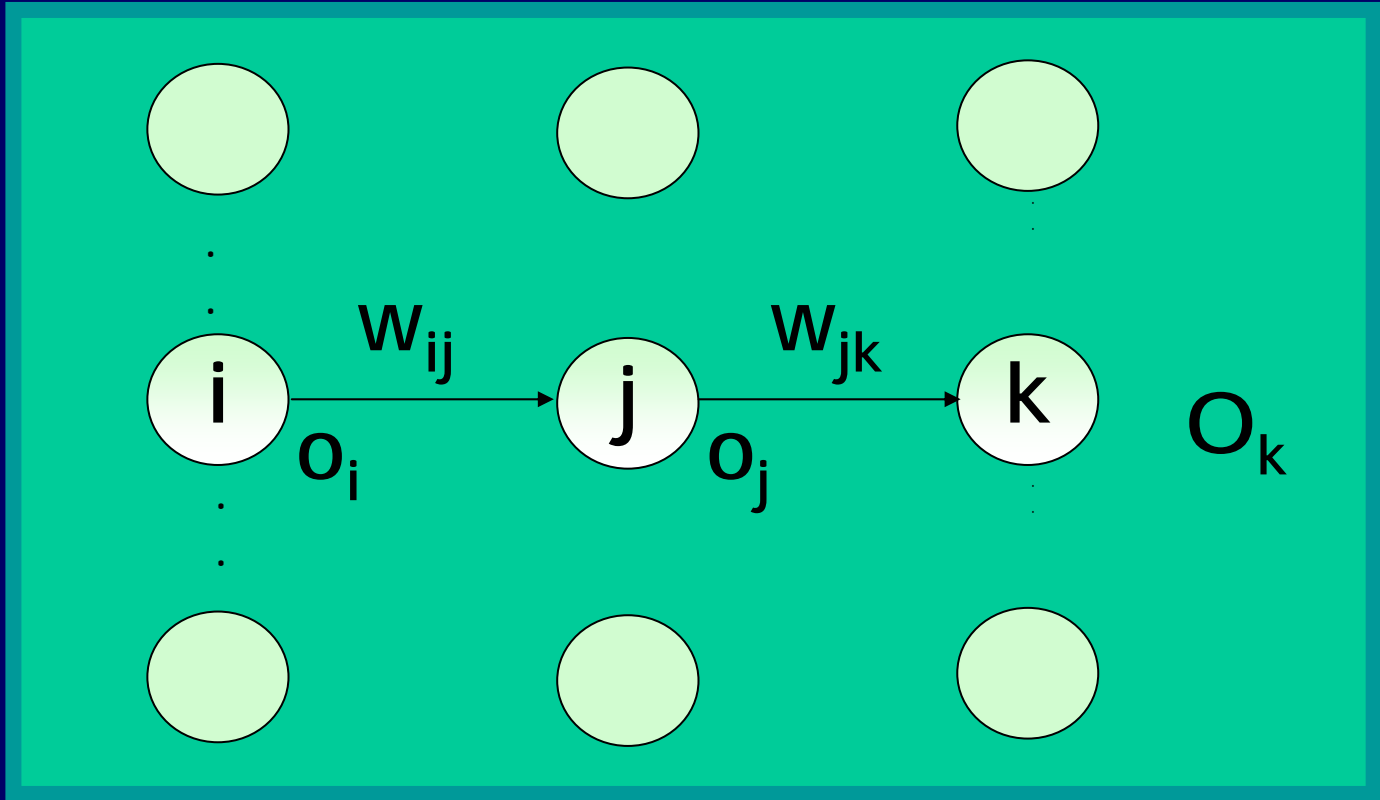




$t$ (Target) :	المخرج النهائي المطلوب
$O$ (Output) :	مخرج أي خلية عصبية
$f$ (Activ. Func.) :	دالة التنشيط
$\eta$ (Learning Rate) :	سرعة التعلم
$W$ (Weight) :	الوزن أو قوة الترابط
$\delta$ (Error signal) :	الخطأ

بعض هذه الرموز ستحمل رموزاً سفلية دليليه (Subscripts) حسب الطبقة التي تنتمي إليها الخلية العصبية .





الرموز المستعملة في تدريب شبكة ذات ثلاث طبقات



## الخطوة الأولى :

■ تبدأ هذه الخطوة عند طبقة المخرجات وفيها نحدد مجموع الإشارات التي تدخل على كل من خلاياها:

$$N_k = \sum_j W_{jk} O_j$$

■ بعد مرور هذه القيمة على الخلية والممثلة في الدالة التنشيطية السجماوية يكون مخرج كل خلية في طبقة المخرجات:

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



## الخطوة الثانية:

• في هذه الخطوة يتم تحديد مقدار الخطأ ( $\delta$ ) على النحو التالي:

$$\delta_k = (t_k - O_k) f' (N_k)$$

• و يمكن تبسيط المعادلة لتصبح:

$$\delta_k = (t_k - O_k) O_k (1 - O_k)$$

• وبالوصول على هذه القيمة يمكن أن نصح كل الأوزان التي تربط الطبقة الخفية بطبقة المخرجات وبالسرعة المطلوبة حسب سرعة التعلم  $\eta$ . على النحو التالي:

$$W_{jk} \leftarrow W_{jk} + \eta \delta_k O_j$$





## الخطوة الثالثة :

- وصلنا للخلايا العصبية في الطبقة الخفية لنعيد تقريباً نفس العمليات السابقة وأولها تحديد قيمة الخطأ في هذه الطبقة:

$$\delta_j = O_j (1 - O_j) \sum_k W_{jk} \delta_k$$

- بالتالي يتم تحديد الأوزان الجديدة بين طبقة المدخلات والطبقة الخفية باستعمال:

$$W_{ij} \leftarrow W_{ij} + \eta \delta_j O_i$$

- ويتم تطبيق هذه الخطوات على كل المدخلات المعدة للتدريب ولمرات عديدة حتى تصل الشبكة إلى أقل نسبة خطأ ممكنة وعندها وتصبح الشبكة جاهزة للاستعمال



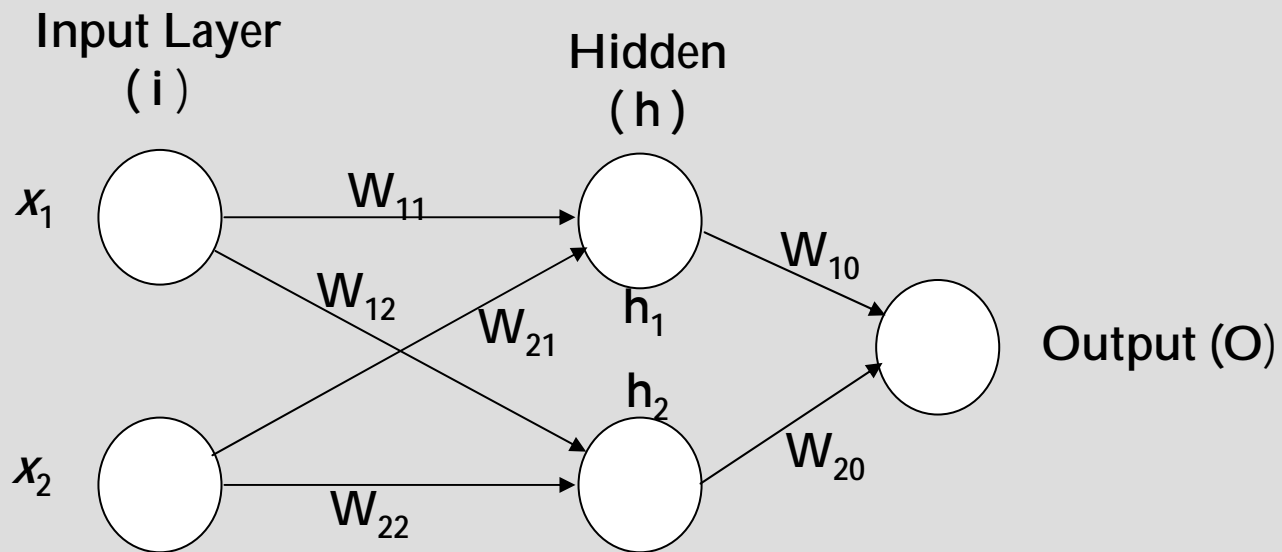
# مثال على تدريب الشبكات

يكون الانطباع الأولي عن عملية تدريب الشبكات بأنها عملية صعبة ومعقدة

لهذا سنحاول من خلال مثال مبسط توضيح أنها ليست بالصعوبة التي يظنها الكثير

سنختار سرعة التدريب  $\eta$  لتكون 1 فقط لتسهيل العمليات الحسابية






## الشبكة المعدة للتدريب



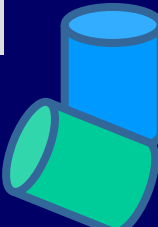
## المدخلات والمخرجات المستعملة للتدريب

$x_1$	$x_2$	Target (t)
0	0	0
0	1	1
1	0	1
1	1	1

في البداية سنفترض أوزاناً  
عشوائية ونبدأ باستعمال الصف  
الأول من جدول المدخلات  
والمخرجات كما هو مبين في  
الجدول



$x_1$	$x_2$	t	$W_{11}$	$W_{12}$	$W_{21}$	$W_{22}$	$W_{10}$	$W_{20}$
0	0	0	1	0	0	1	1	1



بعد ذلك يتم تمرير المدخلات على الشبكة للوصول إلى المخرجات وهنا سنستعمل المصطلحات التالية:

$$h_{i1} =$$

$$h_{i2} =$$

$$h_{o1} =$$

$$h_{o2} =$$

$$N =$$

$$O =$$



وبالتالي نحصل على القيم التالية:

$$\begin{aligned} \mathbf{h}_{i1} &= \mathbf{W}_{11} \mathbf{x}_1 + \mathbf{W}_{21} \mathbf{x}_2 \\ &= (1) (0) + (0) (0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} \mathbf{h}_{i2} &= \mathbf{W}_{12} \mathbf{x}_1 + \mathbf{W}_{22} \mathbf{x}_2 \\ &= (0) (0) + (1) (0) \\ &= 0 \end{aligned}$$



$$\begin{aligned} h_{O1} &= \frac{1}{1 + e^{-h_{i1}}} \\ &= \frac{1}{1 + e^{-0}} = 0.5 \end{aligned}$$

$$\begin{aligned} h_{O2} &= \frac{1}{1 + e^{-h_{i2}}} \\ &= \frac{1}{1 + e^{-0}} = 0.5 \end{aligned}$$



- يمكننا الآن الحصول على مجموع الإشارات التي تدخل على خلية طبقة المخرجات:

$$\begin{aligned} N &= W_{10} h_{O1} + W_{20} h_{O2} \\ &= (1) (0.5) + (1) (0.5) \\ &= 1 \end{aligned}$$

- وبهذا يكون المخرج الفعلي للشبكة:

$$\begin{aligned} O &= \frac{1}{1 + e^{-N}} = \frac{1}{1 + e^{-1}} \\ &= 0.73106 \end{aligned}$$





■ هذه النتيجة الأولية بعيدة عن النتيجة المطلوبة ولذلك نحتاج  
لتعديل الأوزان للاقتراب من الهدف

■ لتحديد مقدار الخطأ في النتيجة ، سنستعمل:

$$\begin{aligned}\delta_O &= (t - O)O(1 - O) \\ &= (0 - 0.73106) (0.73106) (1 - 0.73106) \\ &= -0.14373\end{aligned}$$

■ باستعمال هذه القيمة يمكننا الآن تعديل الأوزان التي تربط بين  
الطبقة الخفية وطبقة المخرجات باستعمال المعادلة التالية:



$$\begin{aligned}W_{10} &\leftarrow W_{10} + \eta \delta_{O1} h_{O1} \\ &= 1 + (1) (-0.14373) (0.5) \\ &= 0.92813\end{aligned}$$

$$\begin{aligned}W_{20} &\leftarrow W_{20} + \eta \delta_{O2} h_{O2} \\ &= 1 + (1) (-0.14373) (0.5) \\ &= 0.92813\end{aligned}$$

عند هذه النقطة نكون قد تراجعنا من مستوى طبقة  
المخرجات إلى الطبقة الخفية



- وسنواصل على نفس النهج في اتجاه طبقة المدخلات .  
يمكننا الآن تحديد قيمة لخطأ:

$$\begin{aligned}\delta_{h_1} &= h_{O1}(1-h_{O1})W_{10}\delta_O \\ &= (0.5)(1-0.5)(.92813)(-0.14373) \\ &= -0.03335\end{aligned}$$

$$\begin{aligned}\delta_{h_2} &= h_{O2}(1-h_{O2})W_{20}\delta_O \\ &= (0.5)(1-0.5)(0.92813)(-0.14373) \\ &= -0.03335\end{aligned}$$



■ ويمكن الآن تعديل الأوزان الباقية كالتالي:

$$\begin{aligned}W_{11} &= W_{11} + \eta \delta_{h_1} x_1 \\ &= 1 + (1) (-0.03335) (0) \\ &= 1\end{aligned}$$

$$\begin{aligned}W_{12} &= W_{12} + \eta \delta_{h_2} x_1 \\ &= 0 + (1) (-0.03335) (0) \\ &= 0\end{aligned}$$

$$\begin{aligned}W_{21} &= W_{21} + \eta \delta_{h_1} x_2 \\ &= 0 + (1) (-0.03335) (0) \\ &= 0\end{aligned}$$

$$\begin{aligned}W_{22} &= W_{22} + \eta \delta_{h_2} x_2 \\ &= 1 + (1) (-0.03335) (0) \\ &= 1\end{aligned}$$

■ نلاحظ هنا أن الأوزان لم تتغير وهذا طبيعي لأن المدخلات كلها تساوي صفر. لكن سيتغير الوضع مع المدخلات الأخرى.



## نتائج المرور الأول في عملية التدريب

$x_1$	$x_2$	$t$	$W_{11}$	$W_{12}$	$W_{21}$	$W_{22}$	$W_{10}$	$W_{20}$
0	0	0	1	0	0	1	0.92813	0.92813

- سنأخذ الآن الصف الثاني من المعطيات ونعيد تدريب الشبكة بنفس الطريقة السابقة وبتابع نفس الخطوات
- إذا البيانات التي سنستعملها في التدريب هي:

$$x_1 = 0 , x_2 = 1 , t = 1$$



- باستعمال هذه القيم والأوزان التي تحصلنا عليها في المرحلة السابقة تكون نتائج التدريب كالتالي:

### نتائج المرور الثاني في التدريب

$x_1$	$x_2$	t	$W_{11}$	$W_{12}$	$W_{21}$	$W_{22}$	$W_{10}$	$W_{20}$
0	1	1	1	0	0.01054	1.00838	0.9503	0.96056



■ تقتضي عملية التدريب أن نعيد نفس الخطوات مرات عديدة  
لنحصل على أقل قيمة للخطأ

■ يوضح الجدول التالي الأوزان بعد إعادة العمليات السابقة  
قراءة الألف مرة وكما نرى في الجدول أصبحت النتائج  
الفعلية قريبة جداً من النتائج المطلوبة

### جدول الأوزان النهائية

$W_{11}$	$W_{12}$	$W_{21}$	$W_{22}$	$W_{10}$	$W_{20}$
-3.5402	4.0244	-3.5248	4.5814	-11.9103	4.6940



## مقارنة بين النتائج الفعلية والمطلوبة بعد إتمام عملية التدريب

$x_1$	$x_2$	Target (t)	Output (O)
0	0	0	0.0264
0	1	1	0.9867
1	0	1	0.9863
1	1	1	0.9908

من خلال المثال السابق ، نرى أن صعوبة التدريب لا تكمن في فهمه ولكن في الجهود الذي يتطلبه خصوصاً مع تكرار العمليات أحياناً **لآلاف المرات** ولهذا السبب تتم هذه العملية عادة باستعمال الحاسوب





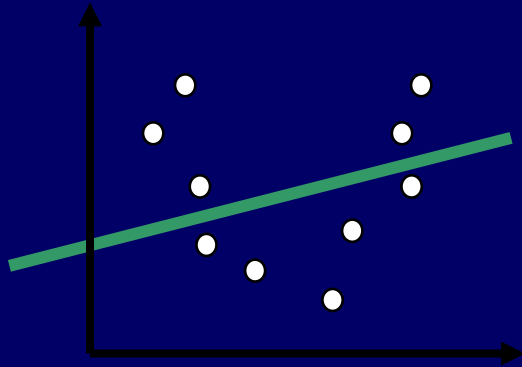
# قضايا إضافية في تدريب الشبكات العصبية:

عند التعامل مع هذه الشبكات يجب الانتباه إلى بعض القضايا المهمة خصوصاً عند التصميم والتدريب وعدم الانتباه لأي من هذه القضايا يؤدي إلى شبكات عصبية ناقصة أو عديمة الفعالية. ومن هذه القضايا:

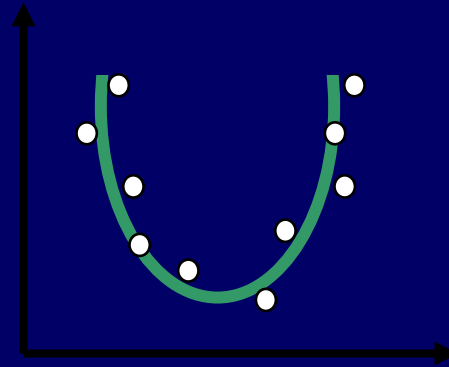
- التدريب المفرط (Overfitting)
- التدريب الناقص (Underfitting)
- اختيار حجم الشبكة
- معايرة البيانات (Normalization)
- سرعة التعلم (Learning Rate)



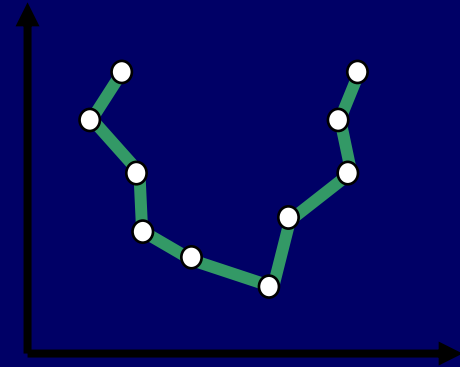
# التدريب المفرط والتدريب الناقص



تدريب ناقص



تدريب سليم



تدريب مفرط



■ لتلافي مشكلات التدريب، هناك عدد كبير من الطرق والتعليمات،  
أهمها التوقف المبكر (Early Stopping) للتدريب.

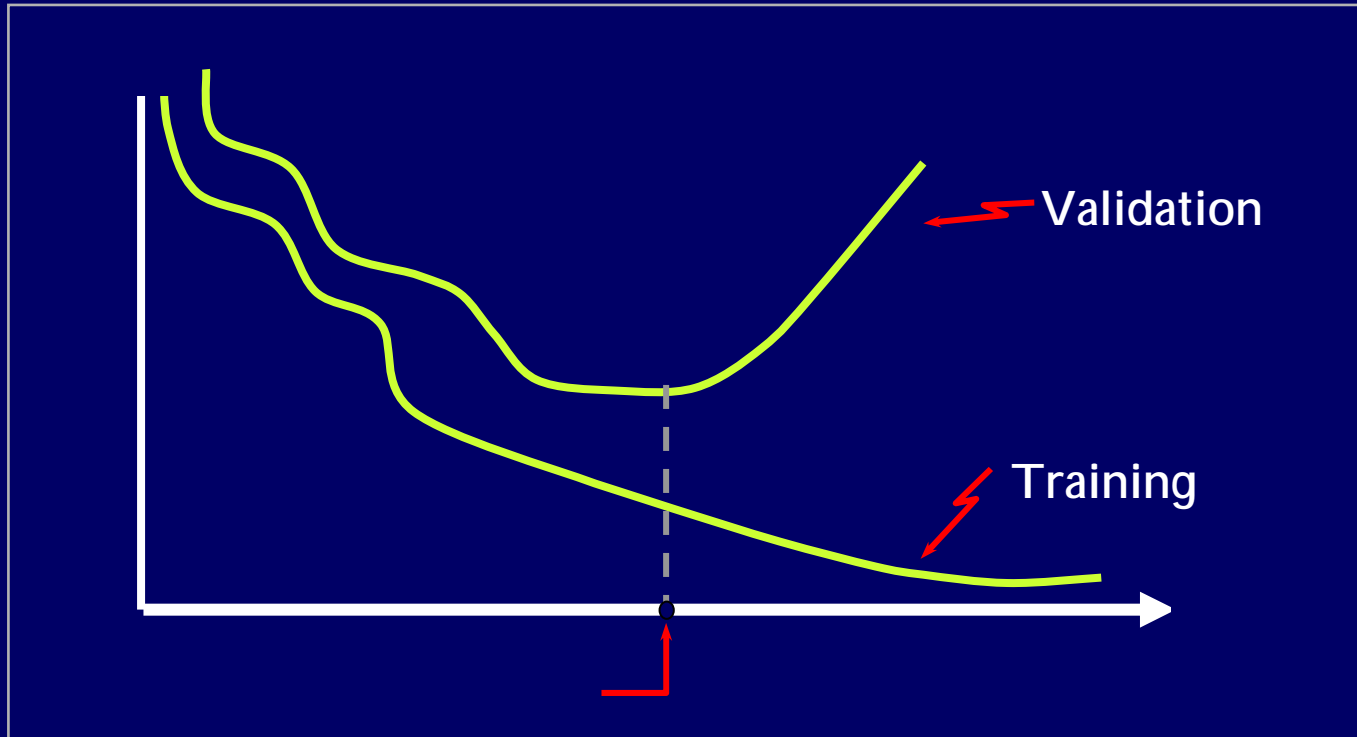
■ وفي هذه الطريقة، تُقسم البيانات إلى ثلاثة أقسام:

● ثلث للتدريب (Training)

● وثلث للتدقيق (Validation)

● وثلث آخر للاختبار (Testing)





رسم خطأ التدريب وخطأ التدقيق لتحديد الوقت المناسب لإنهاء التدريب



# اختيار الحجم المناسب للشبكة:

- يعتبر اختيار الحجم المناسب للشبكة أصعب المشكلات على الإطلاق في تصميم الشبكات العصبية الاصطناعية
- فبالإضافة إلى الاختيارات الكثيرة المتوفرة لدالة التنشيط لكل خلية، هناك إشكالية اختيار العدد المناسب للطبقات في الشبكة وعدد الخلايا في كل من هذه الطبقات
- وكل هذه الاختيارات يجب أن تتم قبل البداية في التدريب وبدون شك، فإن الاختيار غير الموفق لحجم الشبكة يؤدي إلى نتائج غير مقبولة
- مازالت أكثر الطرق استعمالاً في اختيار حجم الشبكة هي أقدم الطرق وأسهلها وهي طريقة المحاولة والخطأ (Trial and Error)



- على المصمم أن يجرب عدداً من الشبكات ويختار أحسنها على أن يكون هذا التجريب منهجياً نوعاً ما حتى لا يستغرق وقتاً طويلاً
- بإمكان المصمم أن يبدأ بشبكة بسيطة ويزيد في حجمها شيئاً فشيئاً بإضافة خلايا أو طبقات إلى أن يصل إلى نتائج مقبولة
- كما يمكنه أن يبدأ بشبكة معقدة ويعمل على تبسيطها شيئاً فشيئاً إلى أن يصل إلى شبكة مقبولة من ناحية التعقيد والأداء
- قد يتبادر إلى الذهن أن هذه الطريقة تستغرق وقتاً طويلاً – قد يكون الحال كذلك – لكن في معظم الأحيان وبتوفر البرمجيات الكثيرة كبرنامج الـ **MATLAB** يمكن الوصول إلى نتائج ممتازة في وقت معقول



# معايرة البيانات Normalization

✱ في حالات كثيرة ، تكون البيانات المعدة لتدريب الشبكة ذات قيم عالية . يُنصح دائماً بمعايرة مدخلات ومخرجات الشبكة

✱ ولهذا المعايير فوائد كثيرة أهمها:

✱ أولاً: أن كل المدخلات تكون قيمها متقاربة وبالتالي لا يطغى مدخل على آخر

✱ ثانياً: أن القيم المعيرة (من الأحسن بين 1 و -1) تجعل التدريب سريعاً

✱ وللقيام بمعايرة البيانات يمكن أن تُقسم بيانات متغير ما على أكبر قيمة فيه وبالتالي تصبح القيمة القصوى بعد المعايرة مساوية لواحد



# قضايا في خوارزمية التدريب :

- من الاختيارات التي يجب على المصمم تحديدها أثناء التدريب سرعة التعلم Learning Rate وهذا المتغير هو الذي يحدد سرعة تحديث الأوزان والوصول إلى الأوزان النهائية
- إذا كان هذا المتغير صغيراً يكون التحديث بطيئاً وبالتالي تستغرق عملية التدريب وقتاً طويلاً
- لكن إذا كان هذا المتغير كبيراً قد تتذبذب الأوزان وتبتعد عن الأوزان المطلوبة شيئاً فشيئاً وتصل عملية التدريب إلى حالة عدم استقرار





- فتحديد سرعة التعلم ، إذا ، تحتاج إلى شيء من الاهتمام
- نظرياً، يمكن للمصمم أن يرسم مقدار الخطأ حسب قيم الأوزان أثناء التدريب ويحصل بذلك على مساحة تسمى مساحة الخطأ
- فإذا كانت المساحة منبسطة بإمكانه تكبير سرعة التعلم وإذا ما كانت متغيرة (صعوداً أو نزولاً) فعليه تصغير سرعة التعلم
- عملياً، يمكن للمصمم أن يختار سرعة التعلم عشوائياً على أن تكون أقل من 1 ويغيرها شيئاً فشيئاً ليصل إلى اختيار مناسب يجمع بين سرعة التدريب والمحافظة على استقراره



## خلاصة

- من خلال ما استعرضناه من بعض القضايا الكثيرة في عملية التدريب وتصميم الشبكات العصبية نرى أن الميدان واسع جداً
- إلا أن هذا الاتساع والتشعب لا يستهوي إلا المتخصصين والباحثين في هذا الميدان ونظرياته
- أما البقية وهم المستعملون (وقد يكونوا الأغلبية) فيكفيهم القليل مما تم استعراضه (حتى بدون القضايا الإضافية) لحل عدد كبير من المشكلات الهندسية في النمذجة والتوقع والاستقراء ومسائل أخرى مماثلة تُظهر فيها الشبكات العصبية الاصطناعية أداءً ملفتاً





# The End

