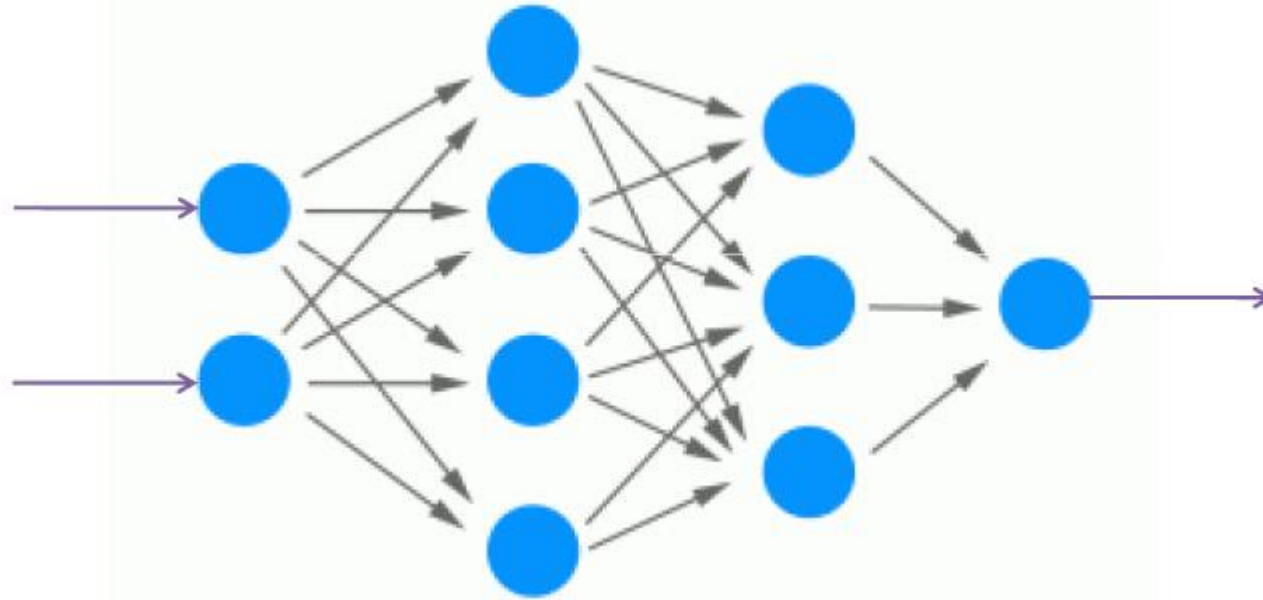
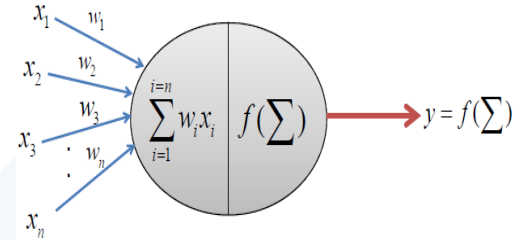
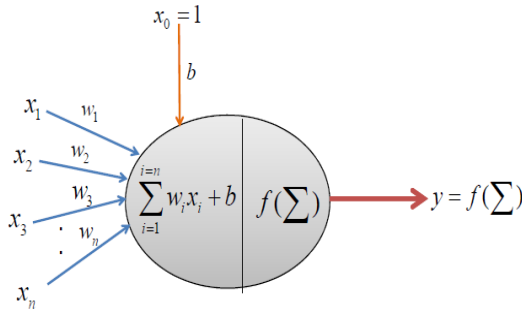


## المتحكمات العصبونية والعائمة المحاضرة /1/ - عملي



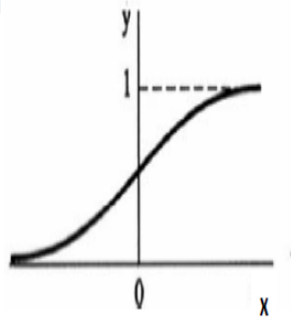
## مقدمة عن الشبكات العصبونية:

- الشبكات العصبونية الصناعية هي نماذج حسابية تحاكي عمل الشبكة العصبونية في دماغ الانسان.
- يدعى عنصر المعالجة فيها بالعصبون Neuron .
- ينجز كل عصبون مهمة حسابية بسيطة.



- اتصال هذه العصبونات ببعضها يحدد سلوك الشبكة الكلي.

## تابع Log-Sigmoid



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

يقع في المجال  $[0,1]$

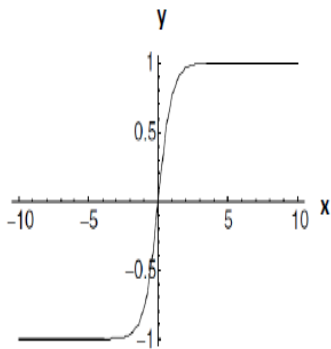
## تابع الحد الفاصل Hard-Limiting



$$y = 0 \text{ if } x < 0$$

$$y = 1 \text{ if } x \geq 0$$

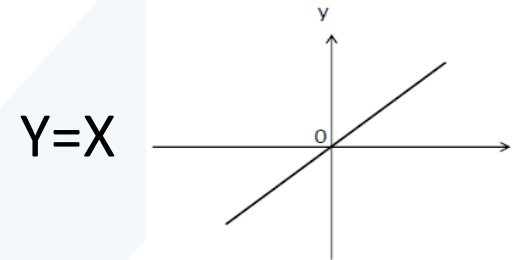
## تابع الظل القطعي Hyperbolic Tangent



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

يقع في المجال  $[-1,1]$

## التابع الخطي Linear



$$Y = X$$

## تصميم الشبكة - مثال 1 (بوابة XOR)

لتصميم شبكة عصبونية ذات تغذية أمامية تقوم بوظيفة محددة يجب اتباع الخطوات الثلاث التالية:

1. إنشاء الشبكة.

2. تدريب الشبكة على بيانات دخل معلومة.

3. اختبار خرج الشبكة عند بيانات دخل جديدة.

### XOR

- لشرح الخطوات سنبدأ بأبسط مثال، بوابة XOR المنطقية التي تتألف من مدخلين ومخرج وحيد، وهي تمثل الطرح الثنائي بالقيمة المطلقة.

I <sub>1</sub>	I <sub>2</sub>	Out
0	0	0
0	1	1
1	0	1
1	1	0

في البداية وقبل إنشاء الشبكة يجب تعريف المداخل والمخارج الخاصة بها:

```
p=[0 0 1 1 ; 0 1 0 1];  
T=[0 1 1 0];
```

ثم ننشئ الشبكة باستخدام التابع newff

```
net = newff( minmax(p) , [4 1] , {'logsig','purelin'} , 'trainlm' );
```

```
net = newff( minmax(p) , [4 1] , {'logsig','purelin'} , 'trainlm' );
```

- التابع newff له عدة بارامترات، الأول minmax(p) يمثل أدنى وأعلى قيمة في كل دخل.
- البارامتر التالي [4 1] يمثل عدد طبقات الشبكة (المخفية والخرج)، وعدد العصبونات في كل طبقة منها (طبقة الدخل لا تقوم بعمليات معالجة)، يخصص الرقم الأخير دوماً لعدد عصبونات طبقة الخرج، في مثالنا الحالي لدينا طبقة مخفية عدد عصبوناتها 4 وطبقة الخرج عدد عصبوناتها 1.
- البارامتر التالي {'logsig','purelin'} يمثل تابع التفعيل لعصبونات كل طبقة من الطبقات، يجب أن يكون عدد التوابع بنفس عدد الطبقات بكل تأكيد.
- البارامتر الأخير 'trainlm' يمثل نمط تدريب الشبكة.

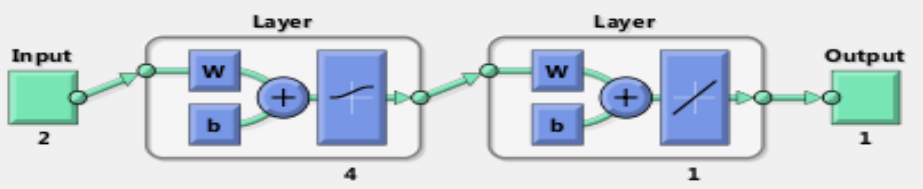
```
net.trainParam.epochs=100;  
net.trainParam.show=10;  
net.trainParam.goal=1e-6;  
net = train (net,p,T);
```

- يمكن الدخول لبارامترات تدريب الشبكة وتعديلها مثل عدد مرات التدريب epochs وعدد مرات عرض النواتج show والهدف المرغوب الذي سيتوقف التدريب عند الوصول إليه goal وغيرها من البارامترات.

- بعد تحديد كامل الخصائص يمكن تدريب الشبكة باستخدام التابع train بإدخال ثلاث بارامترات، اسم الشبكة وبيانات الدخل وبيانات الخرج المرغوب.

Neural Network Training (nntraintool)

**Neural Network**



**Algorithms**

Training: Levenberg-Marquardt (trainlm)  
 Performance: Mean Squared Error (mse)  
 Calculations: MATLAB

**Progress**

Epoch:	0	4 iterations	100
Time:		0:00:00	
Performance:	0.217	3.55e-18	0.00
Gradient:	0.897	1.96e-09	1.00e-07
Mu:	0.00100	1.00e-07	1.00e+10
Validation Checks:	0	0	6

**Plots**

Performance (plotperform)  
 Training State (plottrainstate)  
 Regression (plotregression)

Plot Interval:  epochs

Minimum gradient reached.

$$\text{net} = \text{train}(\text{net}, p, T);$$

يمثل الشكل المبين جانباً أداة تدريب الشبكات العصبونية في برنامج ماتلاب ، وتظهر بنية الشبكة وعدد عصبونات طبقاتها وخوارزمية تدريبها وكامل تفاصيلها .



$y = \text{sim}(\text{net}, p)$

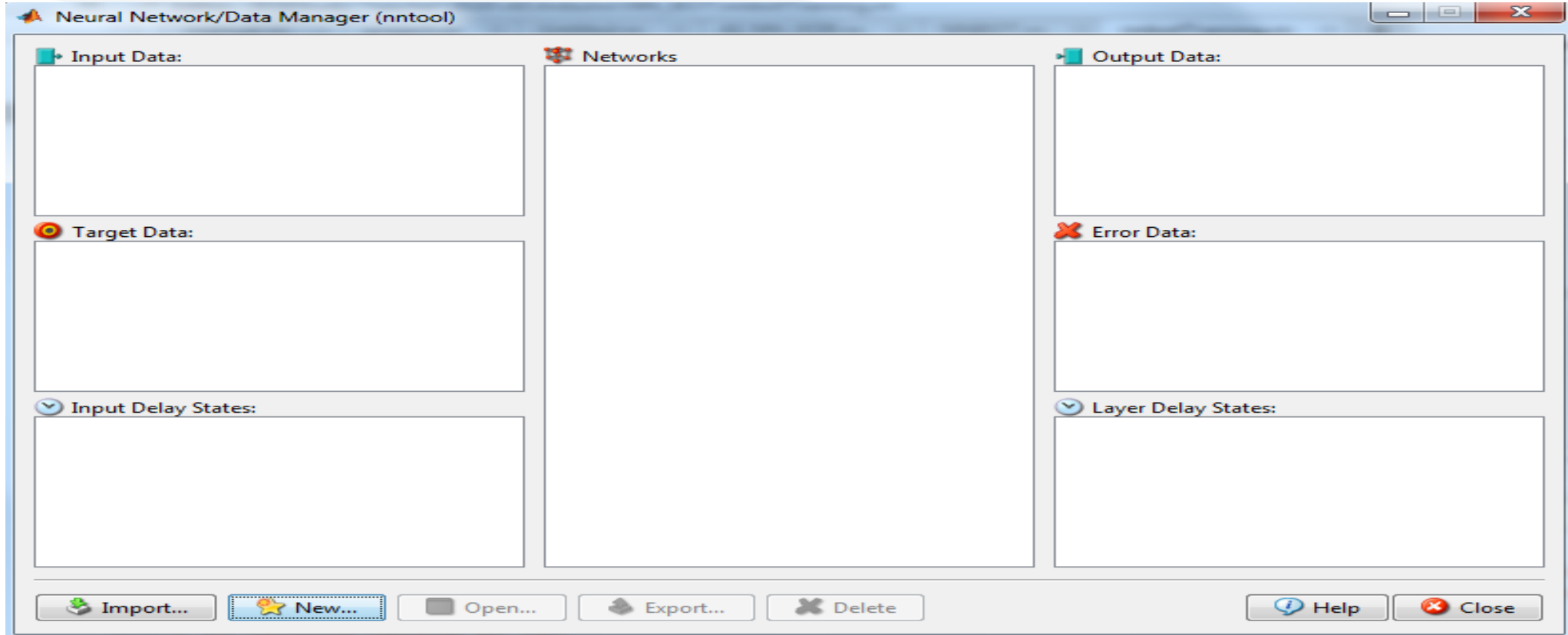
باستخدام التابع `sim` وبإدخال بارامترين، اسم الشبكة المدربة والبيانات التي نريد اختبار الشبكة عندها، نلاحظ النتيجة التالية لخرج الشبكة:

$y =$       0.0000    1.0000    1.0000    0.0000-

لكن في الأمثلة العملية يجب الاختبار عند قيم لم يتم التدريب عليها مسبقاً.

يمكن استدعاؤها بالتعلمية :

>> nntool

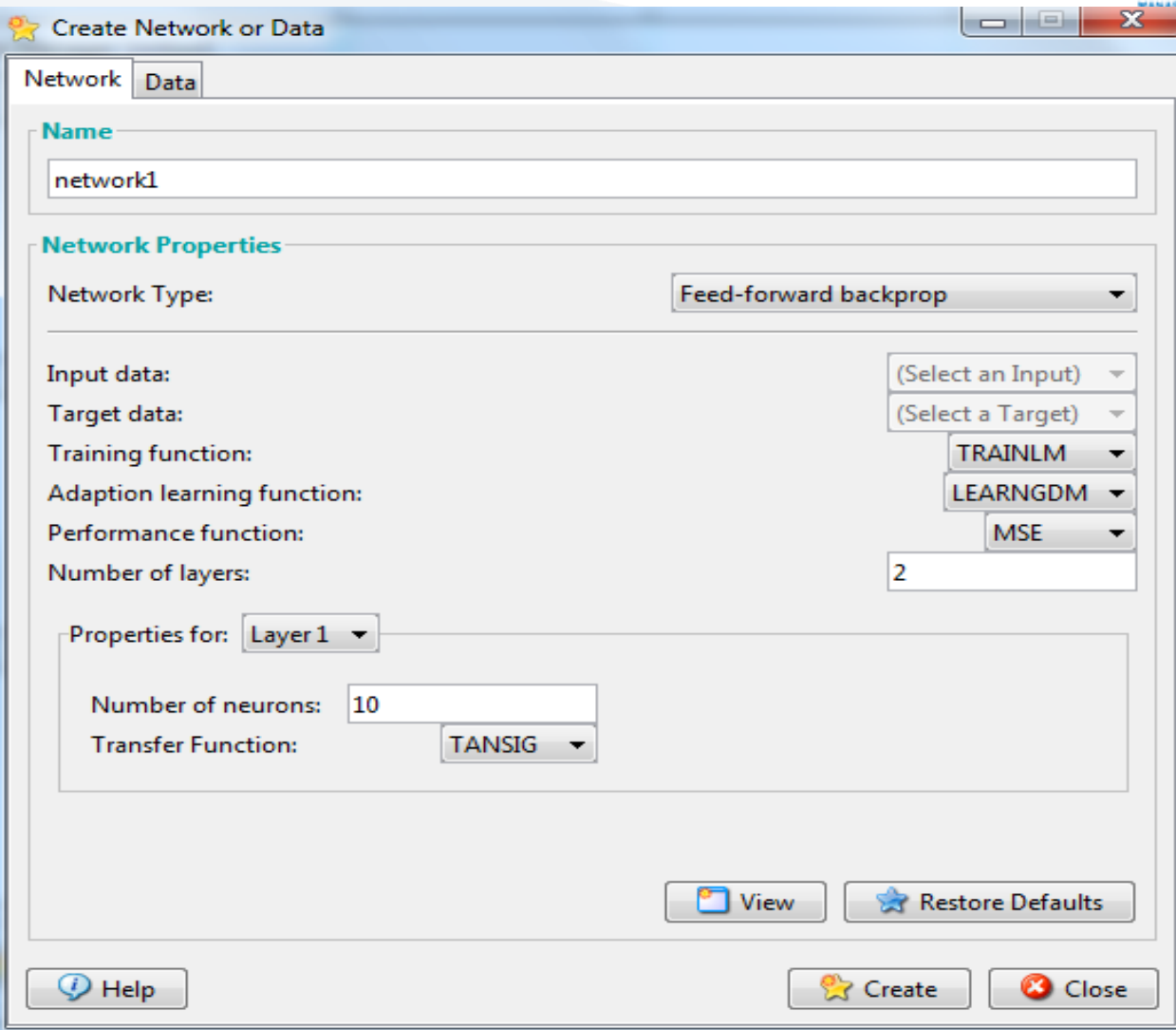


من هنا يتم استيراد الشبكة أو بيانات الدخل والخرج (Import)، أو يمكن إنشاء شبكة جديدة (New)

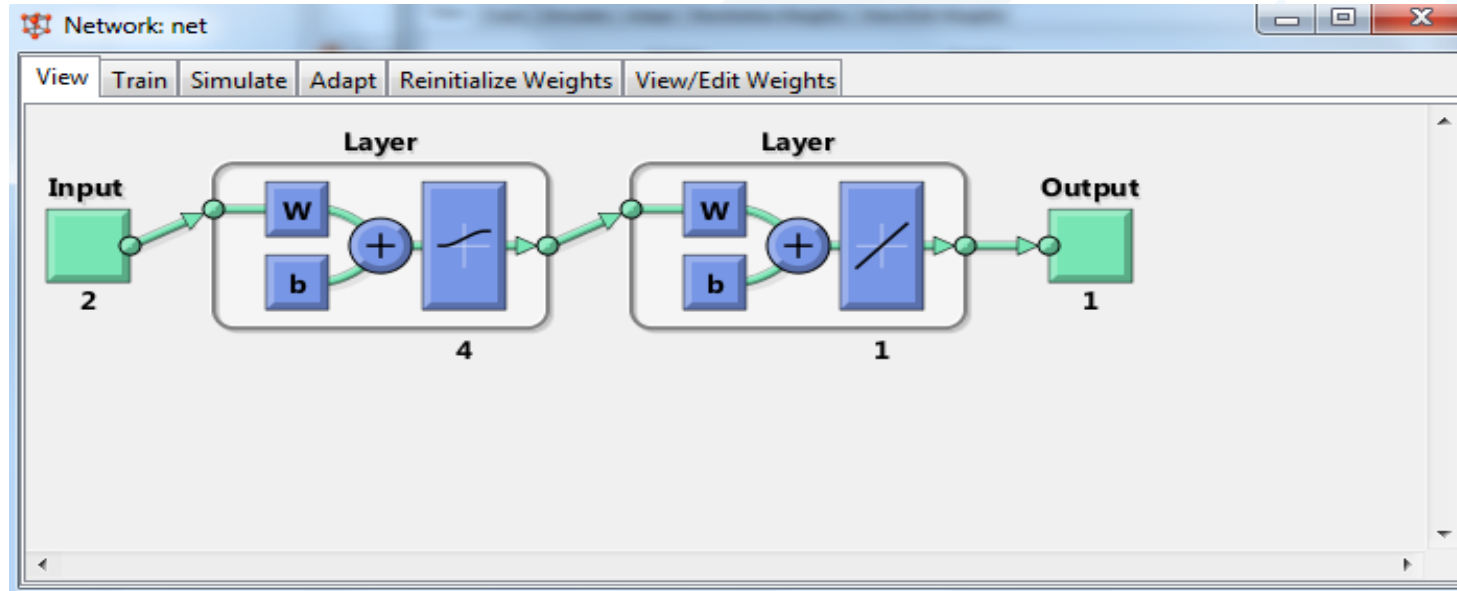
## واجهة ماتلاب للشبكات العصبونية:

عند إنشاء شبكة جديدة (New) عندها تظهر النافذة التالية:

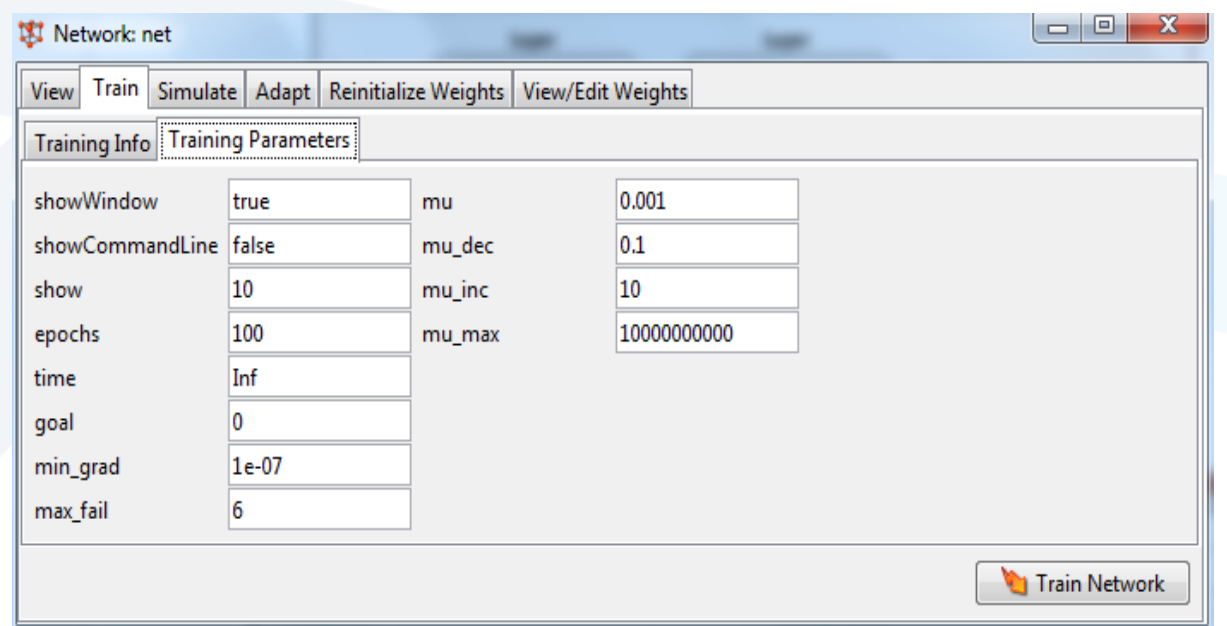
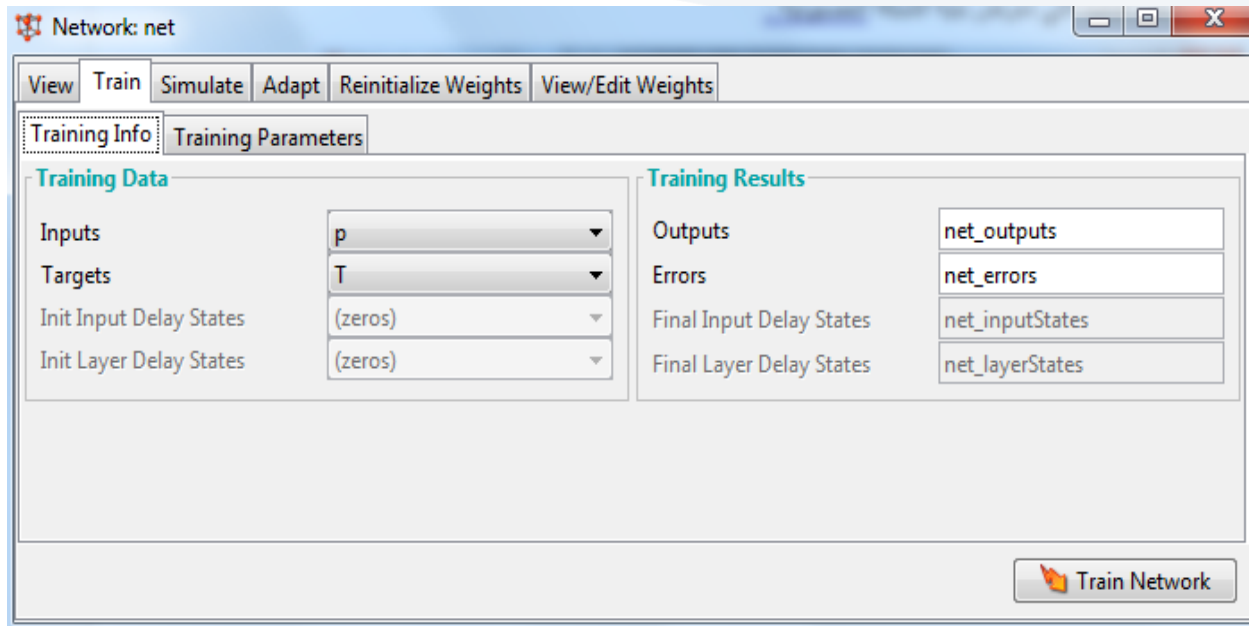
في هذه النافذة، وفي القائمة الأولى Network يمكن تحديد كامل تفاصيل الشبكة من اسمها ونوعها وعدد طبقاتها وعدد عصبونات كل طبقة إضافة لتابع تدريب كل طبقة وتابع التدريب الكلي وتابع الخطأ المستخدم. في القائمة الثانية (Data) يمكن إدخال بيانات الدخل والخرج للشبكة.



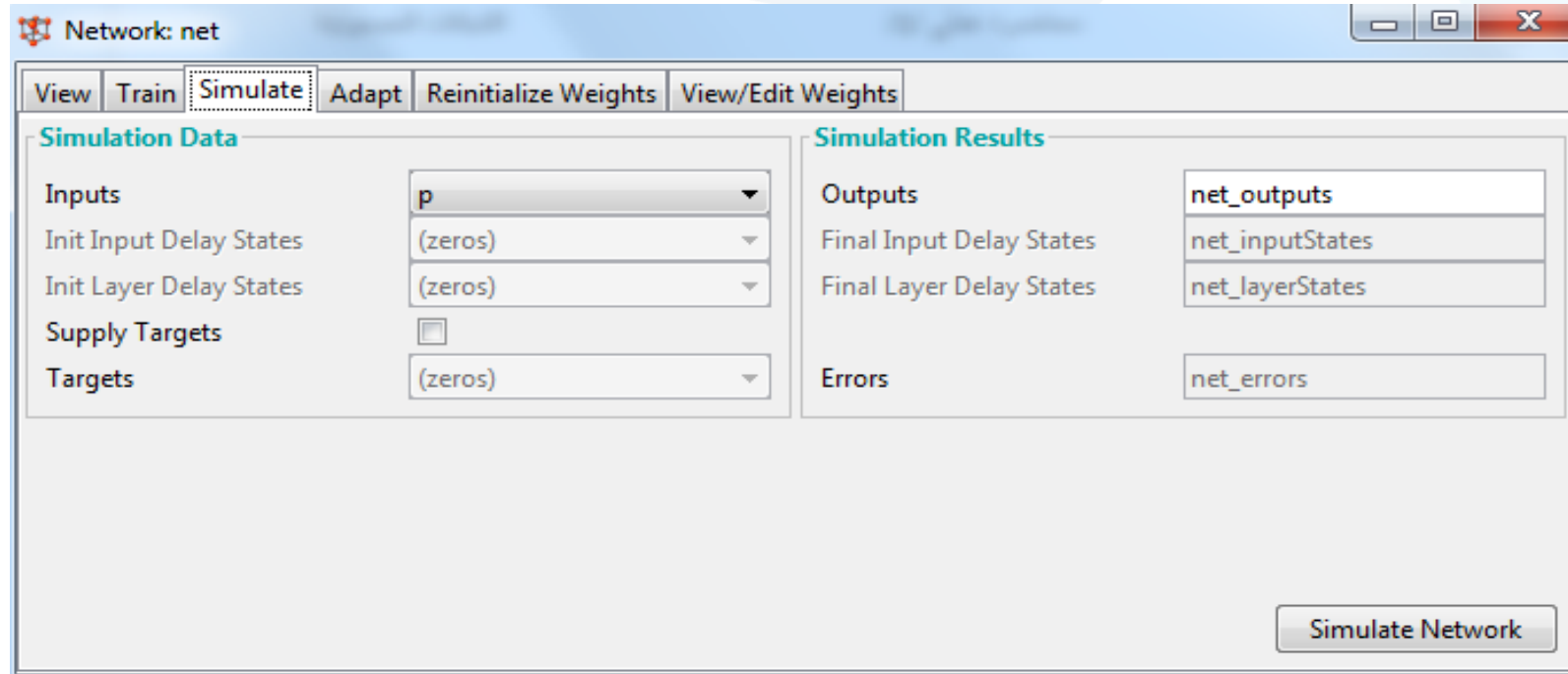
بعد تشكيل الشبكة (يمكن بالطبع تصديرها إلى فضاء عمل ماتلاب) أو التعامل معها من داخل الواجهة كتدريبها واختبارها، وذلك بتحديد الشبكة من النافذة الوسطى (سواء أكانت مستوردة أو منشأة بواسطة الواجهة) واختيار Open.  
عندها ستظهر النافذة التالية التي تعرض بنية الشبكة العصبونية:



نقوم اختيار Train لتدريب الشبكة وفق بيانات دخل وخرج محدد، وبيارمترات محددة (عدد المرات والهدف وغيرها)، ثم نضغط (Train Network):



الخطوة الأخيرة هي اختبار الشبكة عند بيانات الاختبار، وذلك باختيار (Simulate) وتحديد بيانات الدخل الاختباري ثم اختبار الشبكة، ثم الضغط على (Simulate Network) ستصدر النتائج إلى فضاء عمل ماتلاب ليتم التعامل معها لاحقاً كما يقتضي التطبيق.





GOOD LUCK ..

GOOD LUCK ..