

# **INTRODUÇÃO AO ESTUDO DE REDES NEURAIS ARTIFICIAIS**

## **MÓDULO REDES NEURAIS NO MATLAB**

**Laboratório de Conexionismo e Ciências Cognitivas L3C  
Grupo SICRES  
INE - UFSC**



# Objetivo

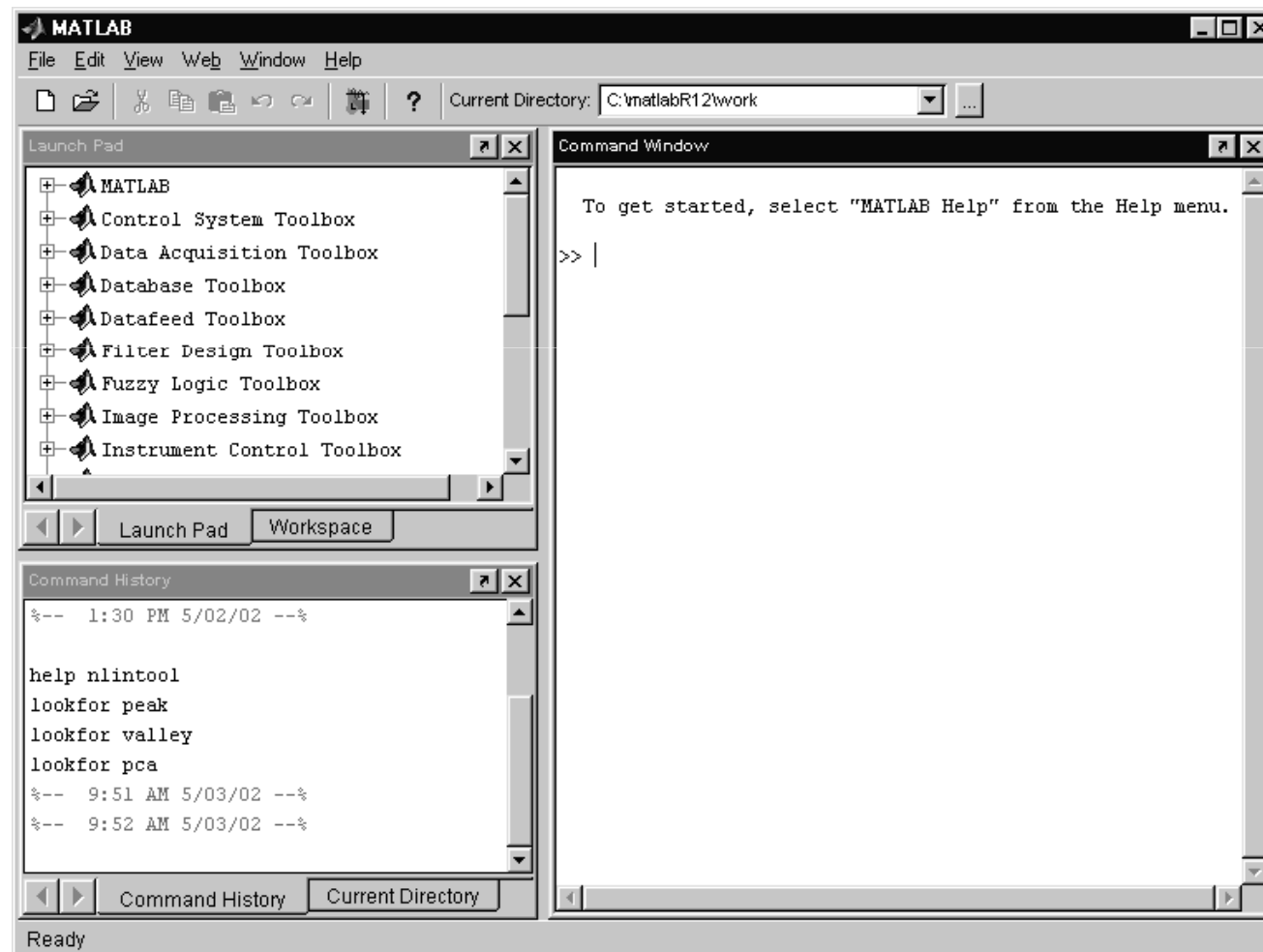
- Oferecer ao aluno uma introdução ao software MATLAB e ao toolbox de Redes Neurais, descrevendo comandos dicas e exemplos de utilização.



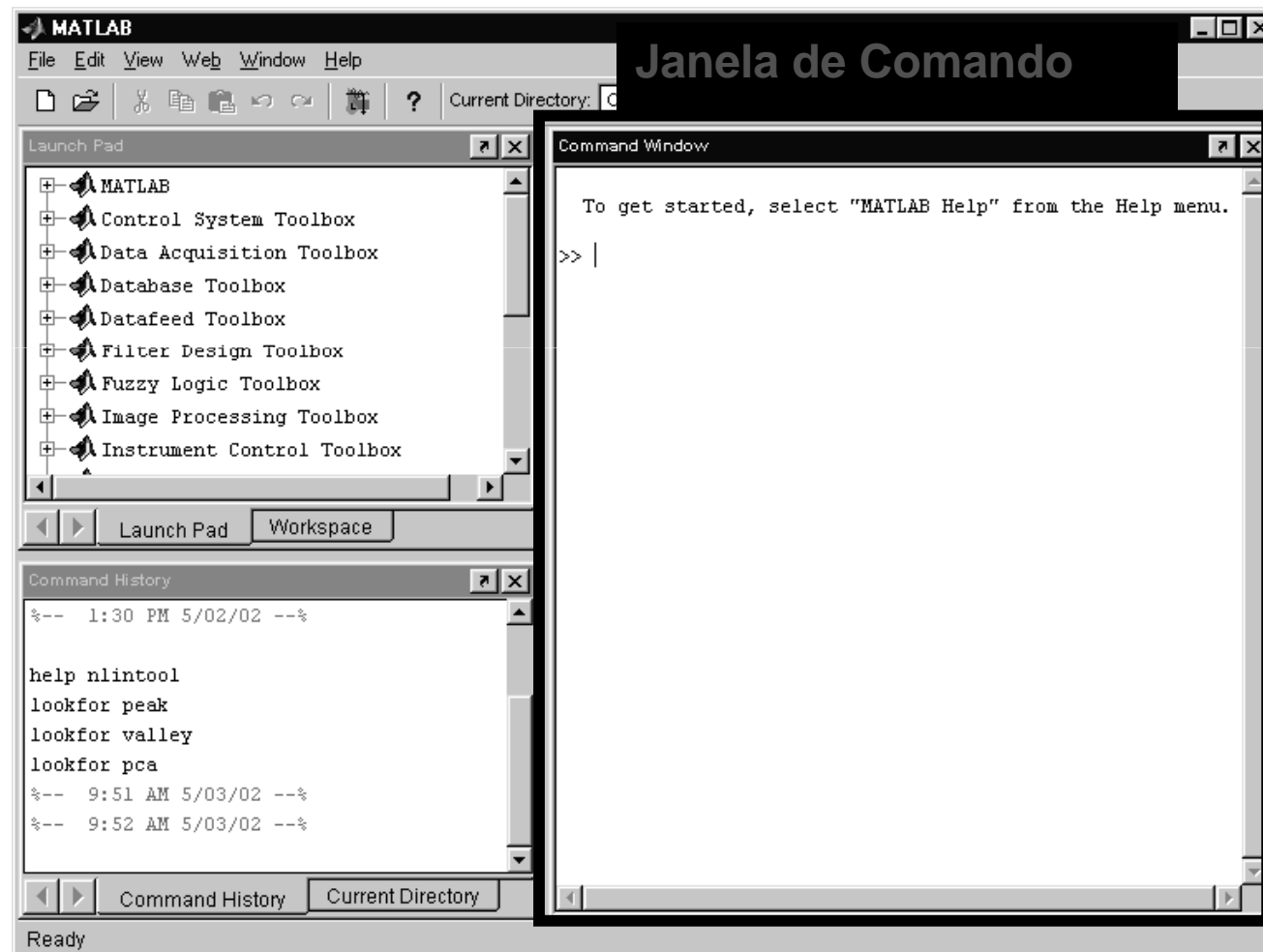
# **Ambiente MATLAB**

- MATLAB → MATrix LABoratory
- Programação baseada em Matrizes
- Vetores e escalares também podem ser considerados matrizes,  $1 \times N$ ,  $N \times 1$ ,  $1 \times 1$

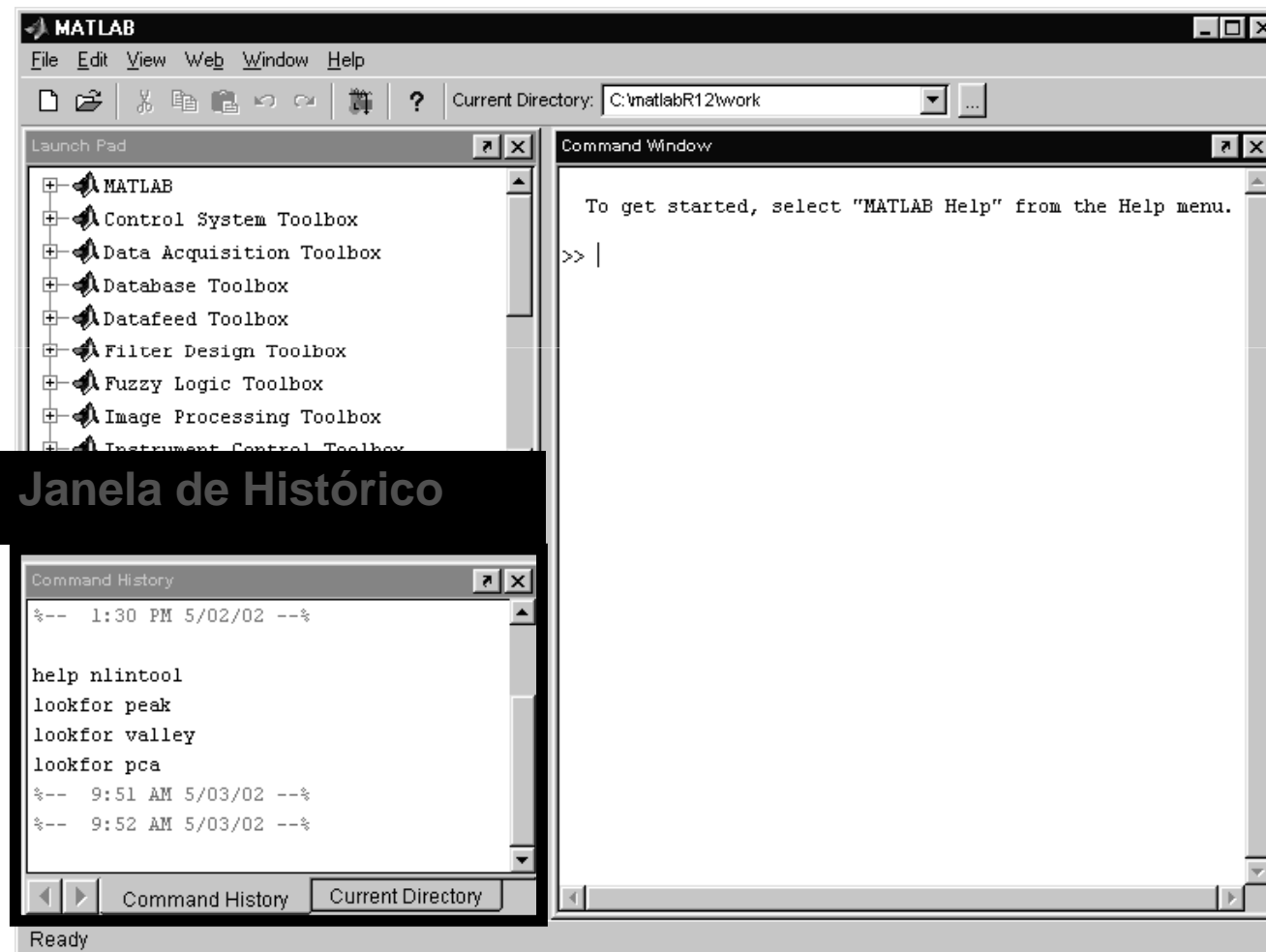
# Tela Principal do MATLAB



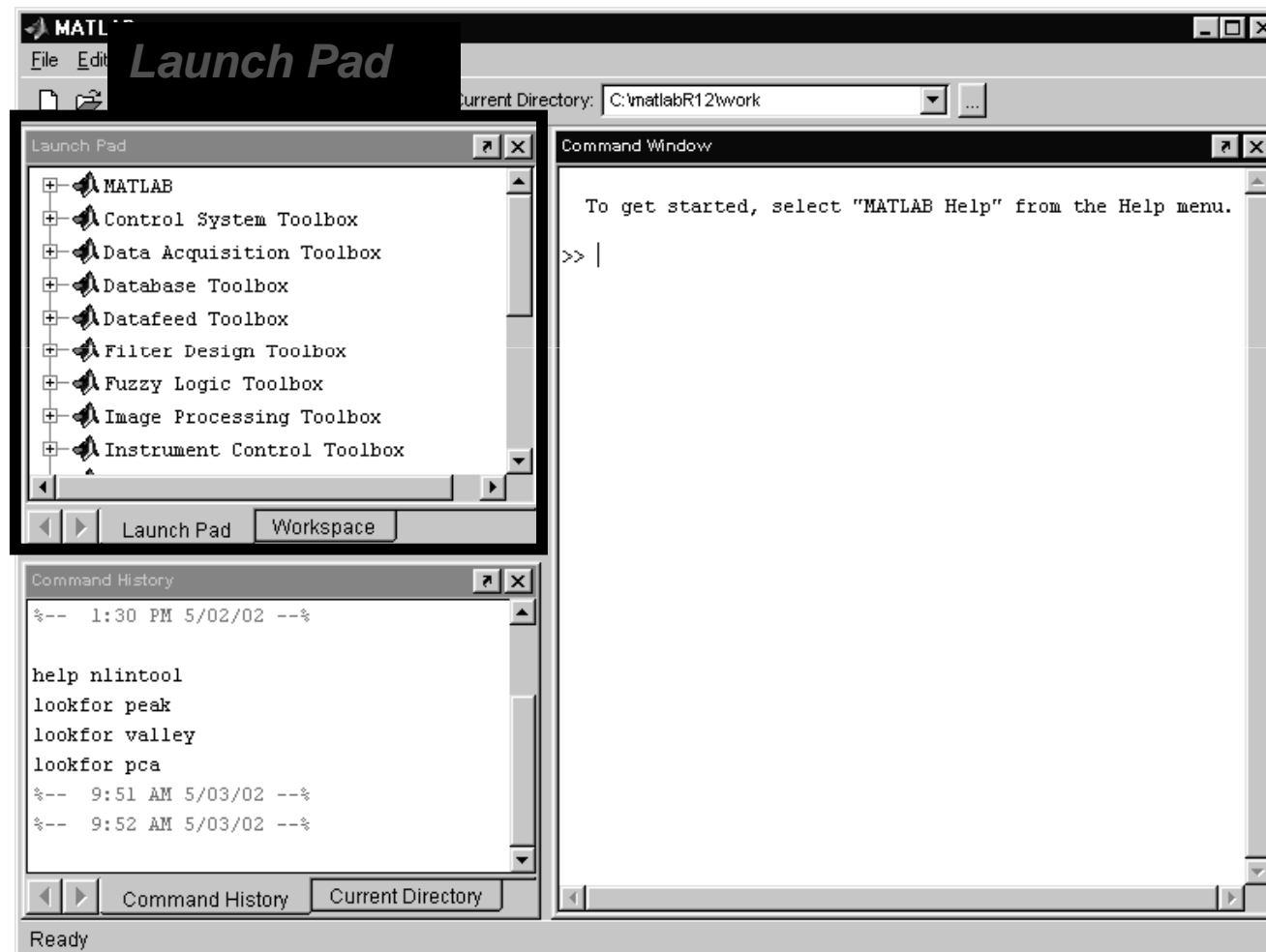
# Tela Principal do MATLAB



# Tela Principal do MATLAB



# Tela Principal do MATLAB





## **Definindo uma Matriz Explicitamente**

- Elementos de uma linha são separados por espaços ou vírgulas.
- O final de cada linha é indicado por um <ENTER> ou ponto-e-vírgula.
- A lista de elementos é delimitada por colchetes [ ].



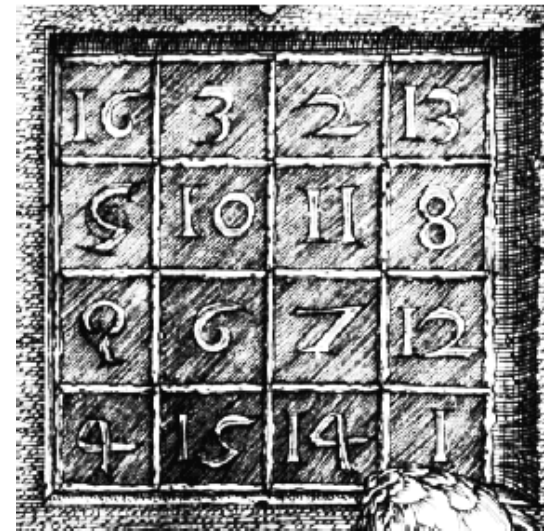
# Definindo uma Matriz Explicitamente

- Exemplo:

```
>> A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

A =

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1



# Operações em Matrizes

- ' : transposição de uma matriz

>> A'

A =

16	5	9	4
3	10	6	15
2	11	7	14
13	8	12	1

# Indexação de Matrizes

- Matriz Bidimensional:  $A(i,j)$  é o elemento da  $i$ -ésima linha,  $j$ -ésima coluna

$A =$	16	3	2	13
	5	10	<b>11</b>	8
	9	6	7	12
	4	15	14	1

$$A(2,3) = 11$$

# **Indexação de Matrizes**

- Soma dos Elementos da 2ª Linha:

```
>> A(2,1) + A(2,2) + A(2,3) + A(2,4)
```

```
ans =
```

```
34
```

# Redimensionamento Automático

```
>> X = A;
```

```
>> X(5,5) = 29
```

```
X =
```

16	3	2	13	0
5	10	11	8	0
9	6	7	12	0
4	15	14	1	0
0	0	0	0	29

# **O Operador ':' (dois pontos)**

```
>> 1:10
```

```
ans =
```

```
1 2 3 4 5 6 7 8 9 10
```

```
>> 1:3:27
```

```
ans =
```

```
1 4 7 10 13 16 19 22 25
```

# O Operador ':' e Matrizes

- $A(1:m, n)$ : primeiros  $m$  elementos da  $n$ -ésima coluna
- $A(:, n)$ : todos os elementos da  $n$ -ésima coluna
- $A(:, \text{end})$ : todos os elementos da última coluna
- $A(m, 1:n)$ : primeiros  $n$  elementos da  $m$ -ésima linha
- $A(m, :)$ : todos os elementos da  $m$ -ésima linha
- $A(\text{end}, :)$ : todos os elementos da última linha

# O Operador ':' e Matrizes

```
>> A(:,4)
```

```
ans =
```

```
13
```

```
8
```

```
12
```

```
1
```

OU

```
>> A(:,end)
```

```
ans =
```

```
13
```

```
8
```

```
12
```

```
1
```





# **Funções de Criação de Matrizes**

- `zeros`: matriz de zeros
- `ones`: matriz de uns
- `rand`: matriz de números aleatórios com distribuição uniforme
- `randn`: matriz de números aleatórios com distribuição normal (Gaussiana)

# Álgebra Linear para Matrizes

- $+$  : soma
- $-$  : subtração
- $/$  : divisão de matrizes
- $*$  : multiplicação de matrizes
- $^$  : potência de uma matriz
- $\text{inv}$  : inversa de uma matriz



# **Operadores Aritméticos para Matrizes**

- $+$  : soma
- $-$  : subtração
- $./$  : divisão elemento a elemento
- $.*$  : multiplicação elemento a elemento
- $.^$  : potência elemento a elemento



# **Redes Neurais no MATLAB**

- Duas formas de utilização:
  - Linhas de comando, e m-files
  - Interface gráfica (NNTool)



# **Redes Neurais no MATLAB**

- Duas formas de utilização:
  - Linhas de comando, e m-files
  - Interface gráfica (NNTool)



# ***Passos para a Criação de uma Rede Neural***

- Definir os padrões
- Inicializar a rede
- Definir os parâmetros de treinamento
- Treinar a rede
- Testar a rede

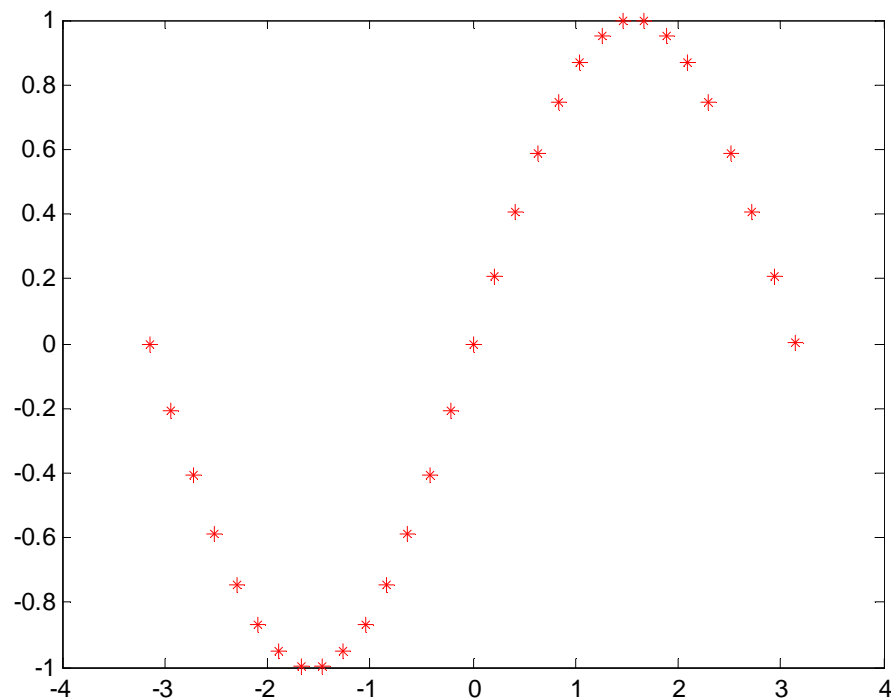


# ***Passos para a Criação de uma Rede Neural***

- Definir os padrões
- Inicializar a rede
- Definir os parâmetros de treinamento
- Treinar a rede
- Testar a rede

# Definindo os Padrões

- `>> angulo=-pi:pi/15:pi;`
- `>> seno=sin(angulo);`







# **Passos para a Criação de uma Rede Neural**

- Definir os padrões
- Inicializar a rede
- Definir os parâmetros de treinamento
- Treinar a rede
- Testar a rede

# Inicializando a Rede Neural

- `net=newff([minmax(angulo)], [5 1],... {'tansig', 'tansig'}, 'traingdm');`
  - `newff` - define uma rede MLP (feedforward)
  - `minmax` - pega os valores mínimos e máximos do padrão de entrada
  - `[5 1]` - 5 neurônios na camada hidden e 1 na camada de saída
  - `'tansig'` - função de saída dos neurônios
  - `'traingdm'` - treinamento batch gradiente descendente com momento

# Funções de Ativação

- purelin Linear
- logsig Sigmóide
- tansig Tangente hiperbólica
- satlin(s) Linear com saturação

# Algoritmos de Treinamento

- **traingd** → Gradient descent backpropagation
- **traingdm** → Gradient descent backpropagation com momentum
- **traingda** → Gradient descent backpropagation com taxa adaptativa
- **traingdx** → Gradient descent backpropagation com momentum e taxa adaptativa
- **trainlm** → Levenberg-Marquardt backpropagation (default)
- **trainrp** → Resilient backpropagation (Rprop)



# ***Passos para a Criação de uma Rede Neural***

- Definir os padrões
- Inicializar a rede
- Definir os parâmetros de treinamento
- Treinar a rede
- Testar a rede

# Definindo parâmetros de treinamento

- `net.trainParam.epochs = 3000;` Número máx. de épocas
- `net.trainParam.goal = 1e-3;` Erro final desejado
- `net.trainParam.lr = 0.1;` Taxa de aprendizado
- `net.trainParam.show = 25;` Atualização da tela (épocas)
- `net.trainParam.mc = 0.9;` Taxa de momento

# Definindo parâmetros de treinamento

- `net.trainParam.epochs = 100;` Número máx. de épocas
- `net.trainParam.goal = 1e-8;` Erro final desejado
- `net.trainParam.lr = 0.01;` Taxa de aprendizado
- `net.trainParam.show = 25;` Atualização da tela (épocas)
- `net.trainParam.mc = 0.9;` Taxa de momento
  
- `net.trainParam.lr_inc = 1.05;` Taxa de incremento da lr
- `net.trainParam.lr_dec = 0.7;` Taxa de decremento da lr
- `net.trainParam.max_perf_inc = 1.04;` Incremento máximo do erro



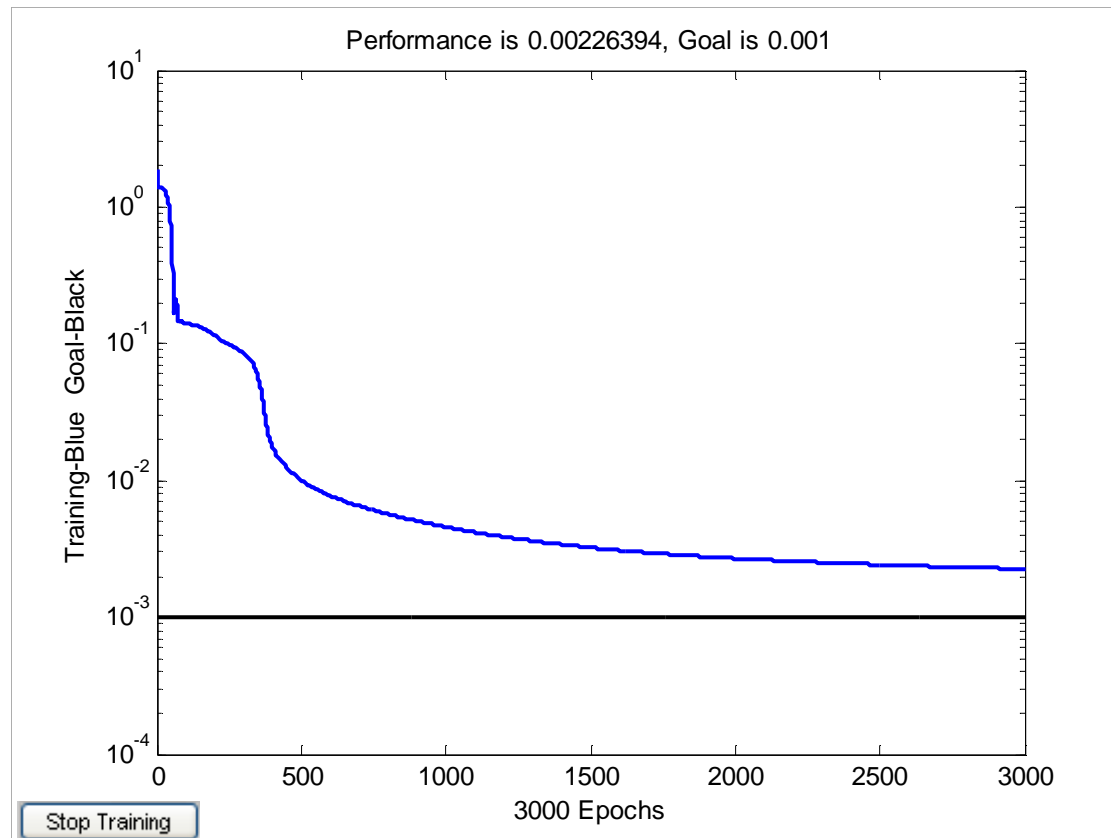
# **Passos para a Criação de uma Rede Neural**

- Definir os padrões
- Inicializar a rede
- Definir os parâmetros de treinamento
- Treinar a rede
- Testar a rede



# Treinando a Rede Neural

- `net = train(net, angulo, seno);`



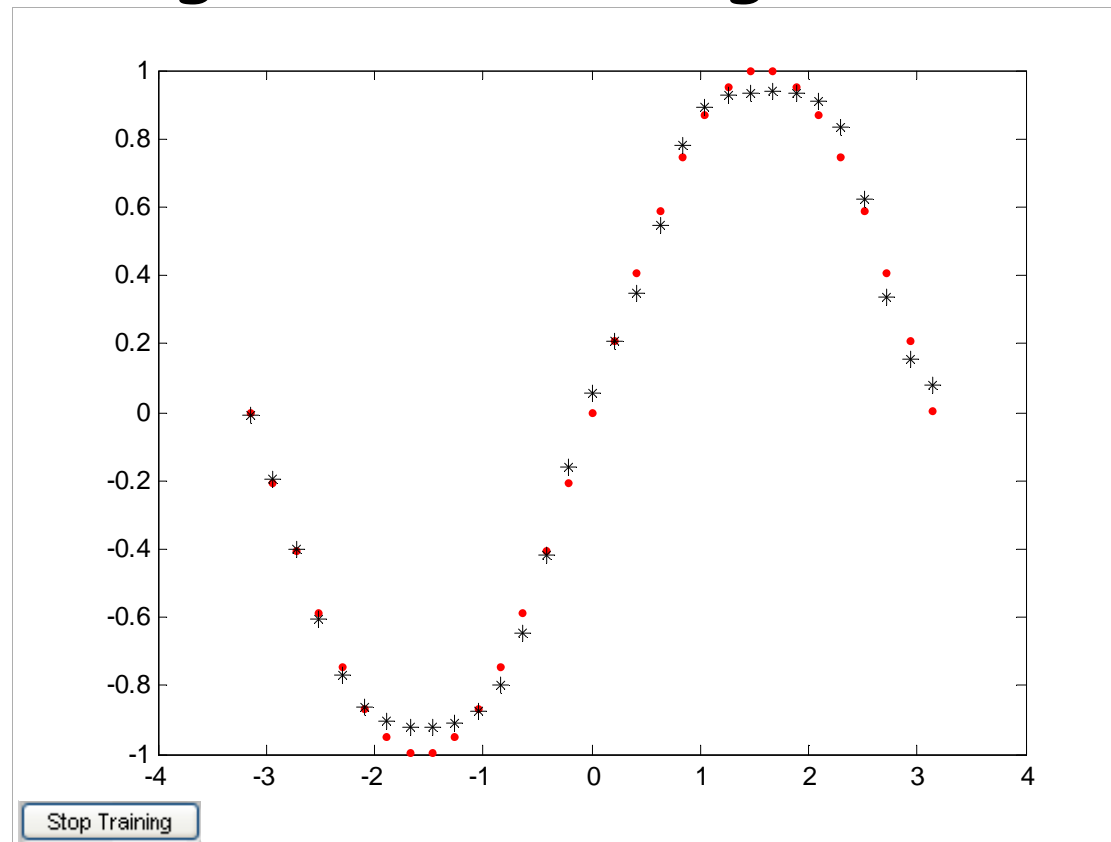


# **Passos para a Criação de uma Rede Neural**

- Definir os padrões
- Inicializar a rede
- Definir os parâmetros de treinamento
- Treinar a rede
- Testar a rede

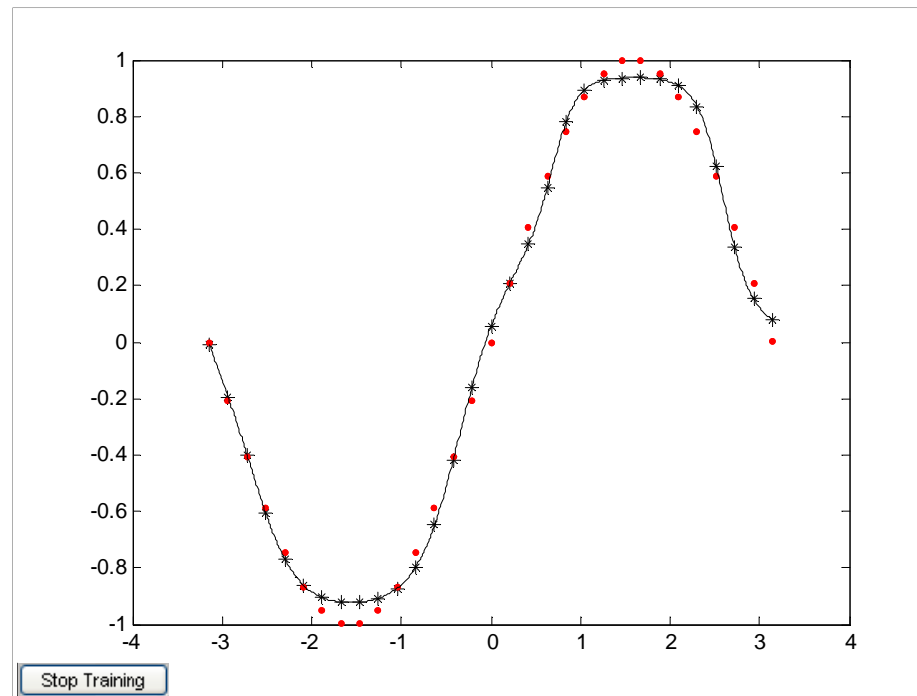
# Testando a Rede Neural

- `c = sim(net, angulo);`
- `plot(angulo, seno, 'r.', angulo, c, 'k*');`



# Testando a Rede Neural

- $A = -\pi : \pi / 306 : \pi;$
- $C = \text{sim}(\text{net}, A);$
- $\text{plot}(\text{angulo}, \text{seno}, 'r.', \text{angulo}, C, 'k-');$





# Validação

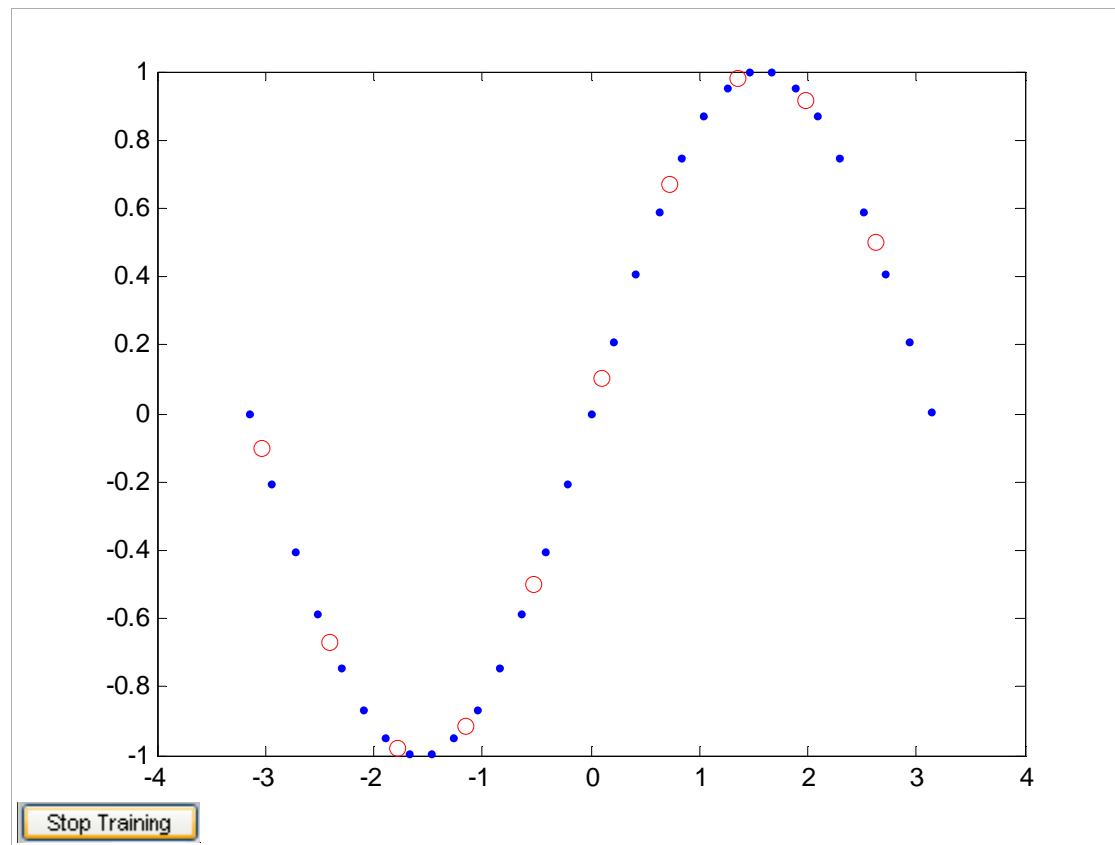
- Dividir os padrões disponíveis em três conjuntos:
  - treinamento (70%): matrizes  $P_{train}$ ,  $T_{train}$
  - teste (20%): matrizes  $P_{test}$ ,  $T_{test}$
  - validação (10%): matrizes  $P_{valid}$ ,  $T_{valid}$

# Validação

- Gera conjunto de treinamento:
  - `>> angulo=-pi:pi/15:pi;`
  - `>> seno=sin(angulo);`
- Gera conjunto de validação:
  - `>> val.P = -pi+pi/7:pi/6:pi;`
  - `>> val.T = sin(val.P);`
  - `>> plot(angulo, seno,'b.',val.P,val.T,'ro');`



# Validação



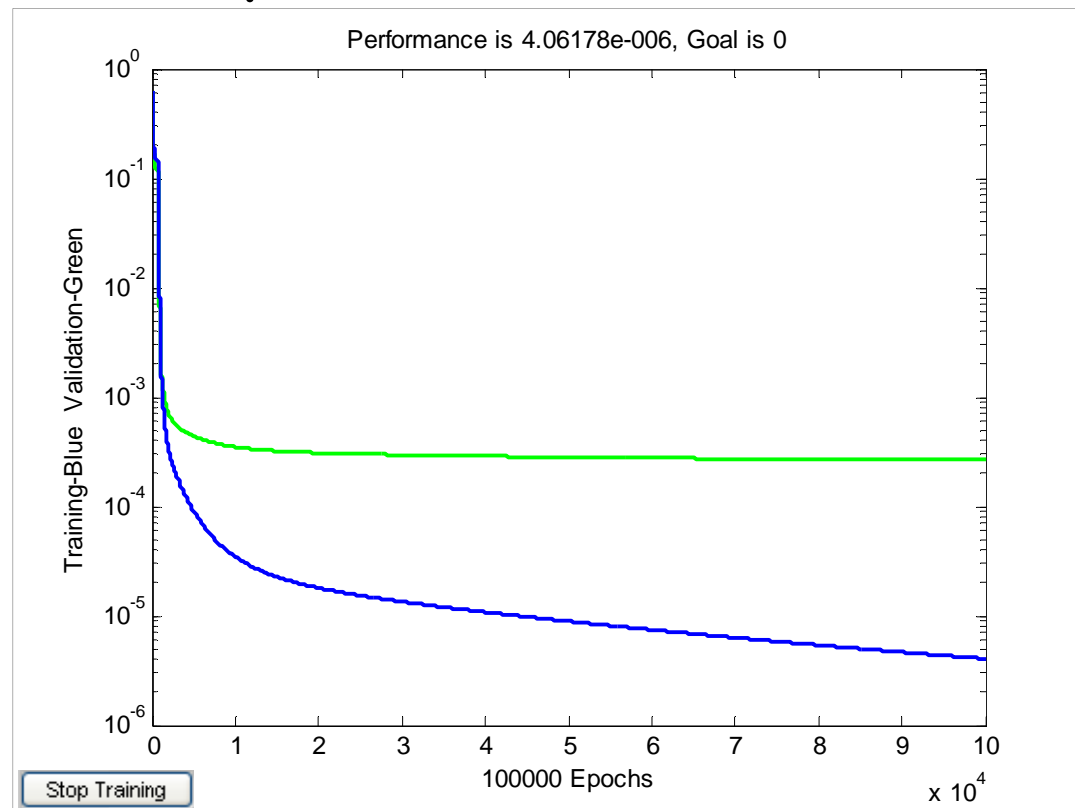
# Validação

- Treina a rede neural incluindo a validação como critério de parada
  - `>> net=newff([minmax(angulo)] ,[15,1],...  
{'tansig','tansig'},'traingdm');`
  - `>> net.trainParam.show = 25;`
  - `>> net.trainParam.epochs = 300000;`
  - `>> net.trainParam.max_fail = 50;`
  - `>> [net,tr]=train(net,p,t,[],[],val);`



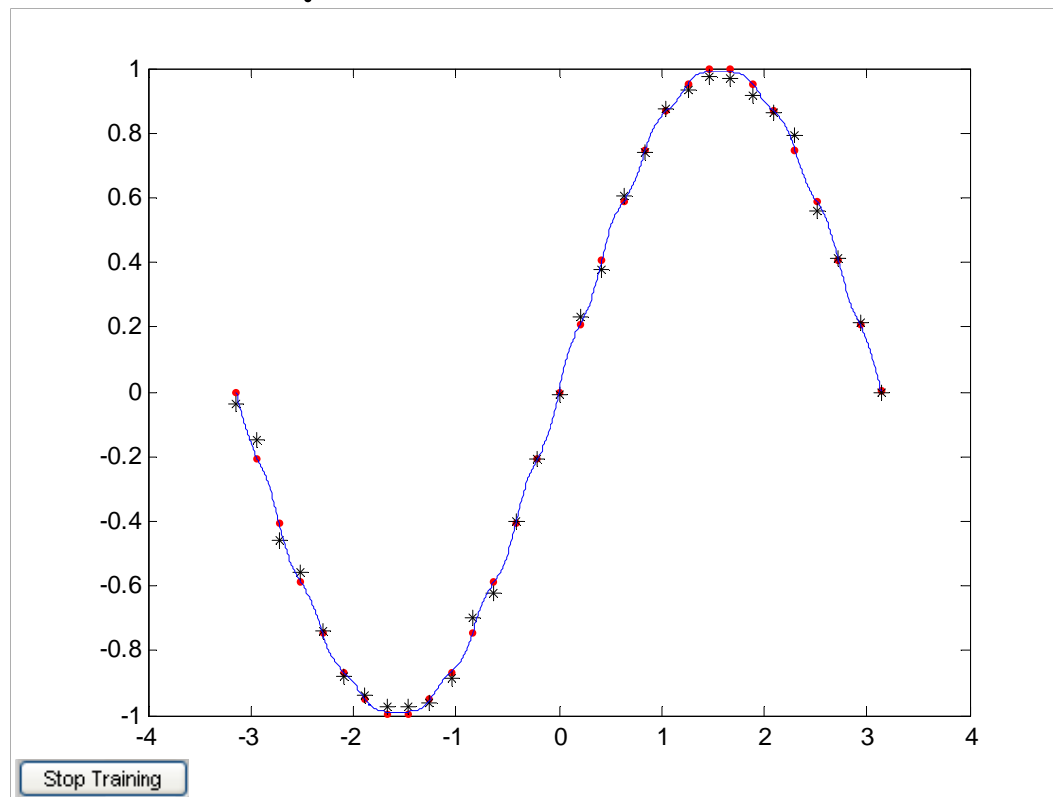
# Validação

- Treina a rede neural incluindo a validação como critério de parada



# Validação

- Treina a rede neural incluindo a validação como critério de parada

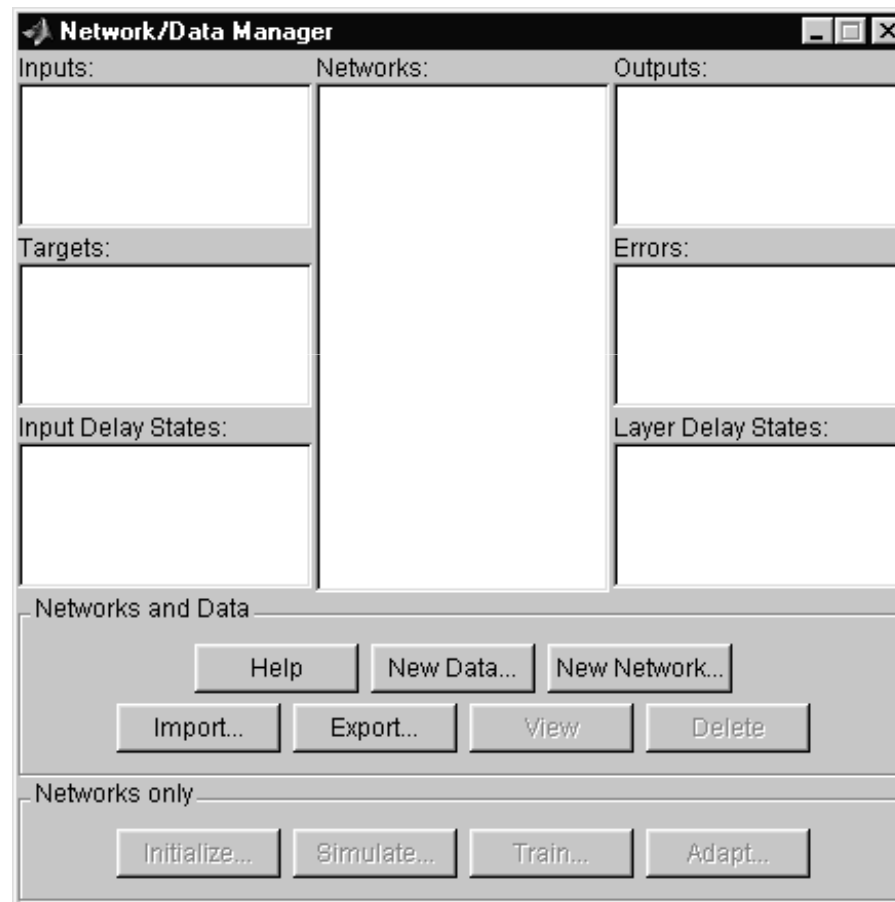




# **Redes Neurais no MATLAB**

- Duas formas de utilização:
  - Linhas de comando, e m-files
  - Interface gráfica (NNTool)

# Interface Gráfica NNTool





# **Passos para a Criação de uma Rede Neural**

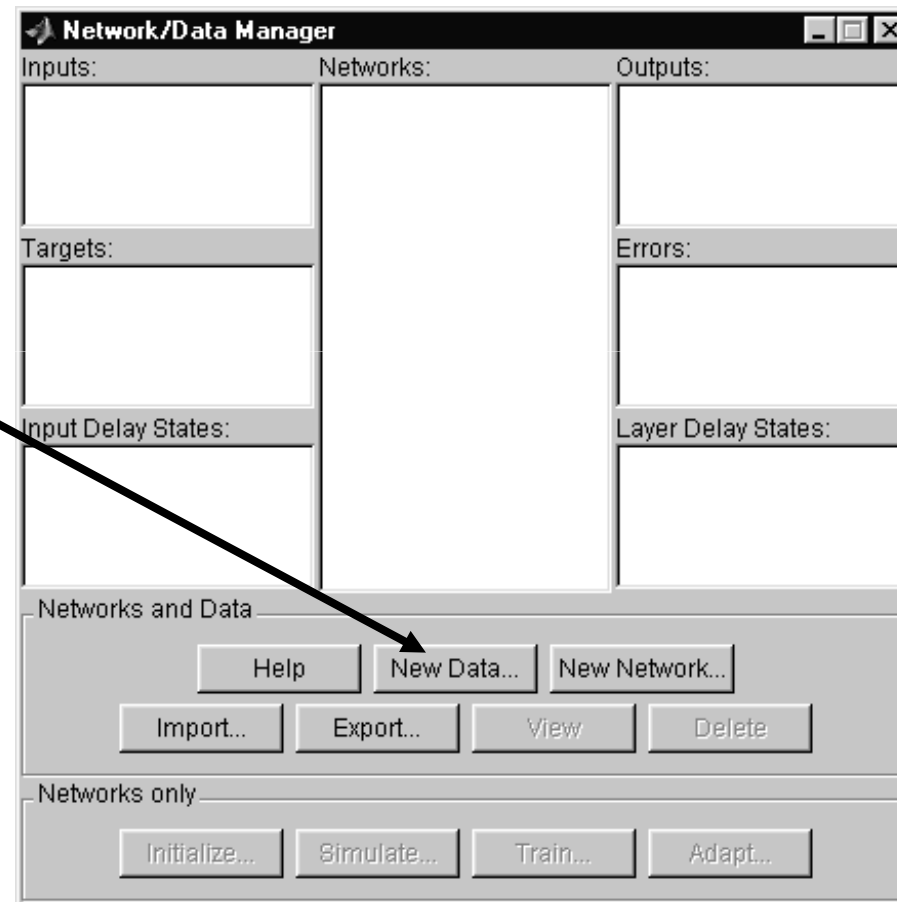
- Definir os padrões
- Criar a rede
- Inicializar a rede
- Definir os parâmetros de treinamento
- Treinar a rede
- Testar a rede



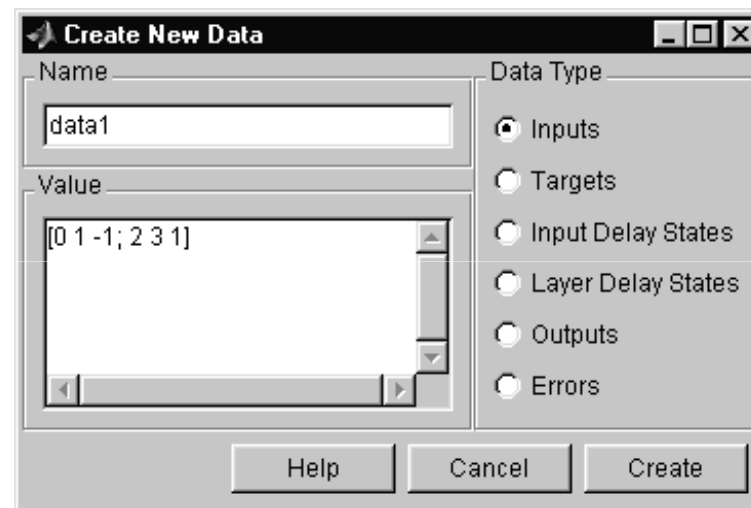
# **Passos para a Criação de uma Rede Neural**

- Definir os padrões
- Criar a rede
- Inicializar a rede
- Definir os parâmetros de treinamento
- Treinar a rede
- Testar a rede

# Definindo os Padrões



# Definindo os Padrões



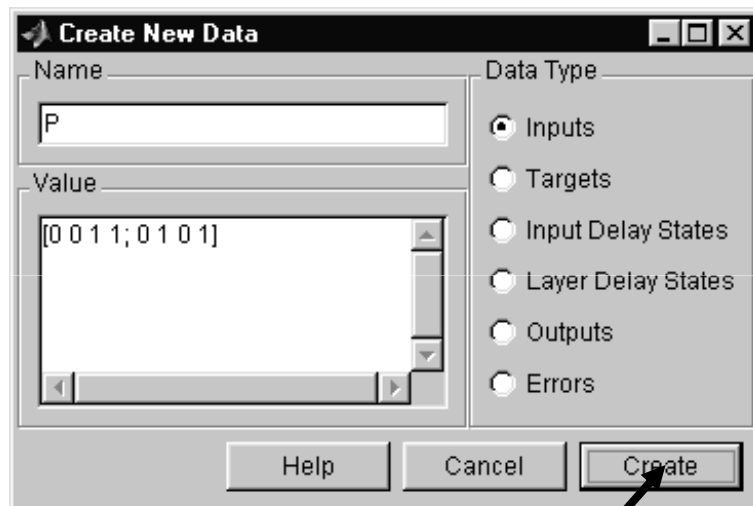
The image shows a 'Create New Data' dialog box with the following fields and options:

- Name:** A text field containing 'data1'.
- Value:** A text area containing the matrix  $\begin{bmatrix} 0 & 1 & -1 \\ 2 & 3 & 1 \end{bmatrix}$ .
- Data Type:** A list of radio button options:
  - ☒ Inputs
  - ☐ Targets
  - ☐ Input Delay States
  - ☐ Layer Delay States
  - ☐ Outputs
  - ☐ Errors

At the bottom of the dialog are three buttons: 'Help', 'Cancel', and 'Create'.



# Definindo os Padrões



**Create New Data**

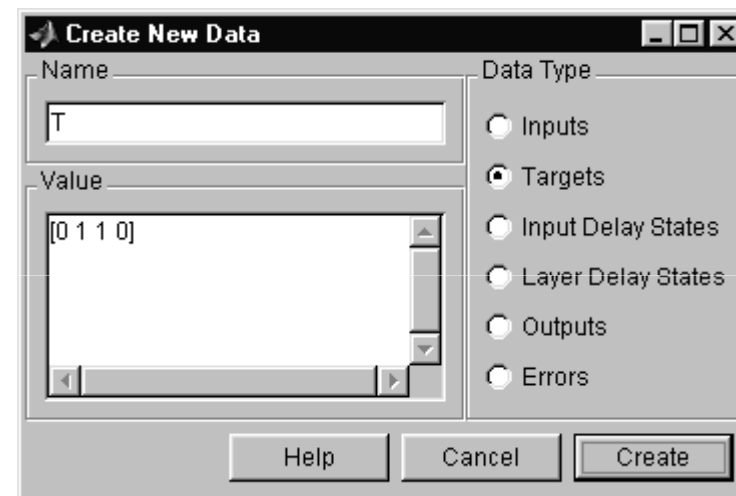
Name:

Value:

Data Type:

- ☒ Inputs
- ☐ Targets
- ☐ Input Delay States
- ☐ Layer Delay States
- ☐ Outputs
- ☐ Errors

Help Cancel **Create**



**Create New Data**

Name:

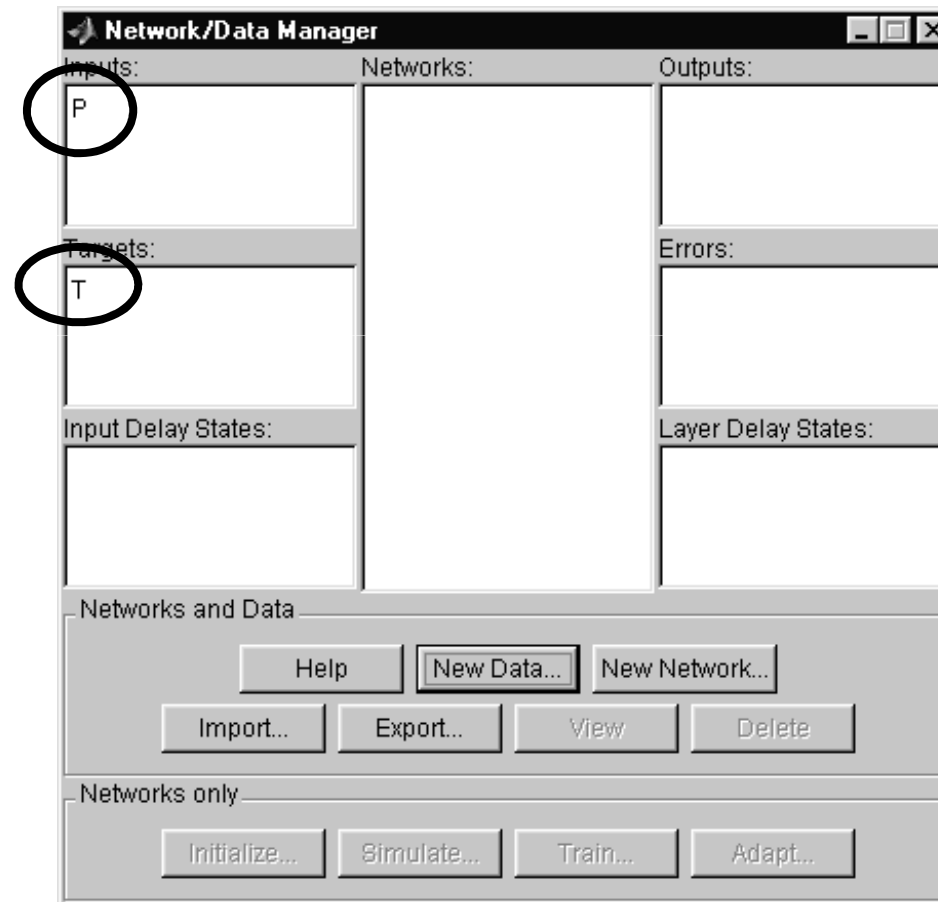
Value:

Data Type:

- ☐ Inputs
- ☒ Targets
- ☐ Input Delay States
- ☐ Layer Delay States
- ☐ Outputs
- ☐ Errors

Help Cancel **Create**

# Definindo os Padrões

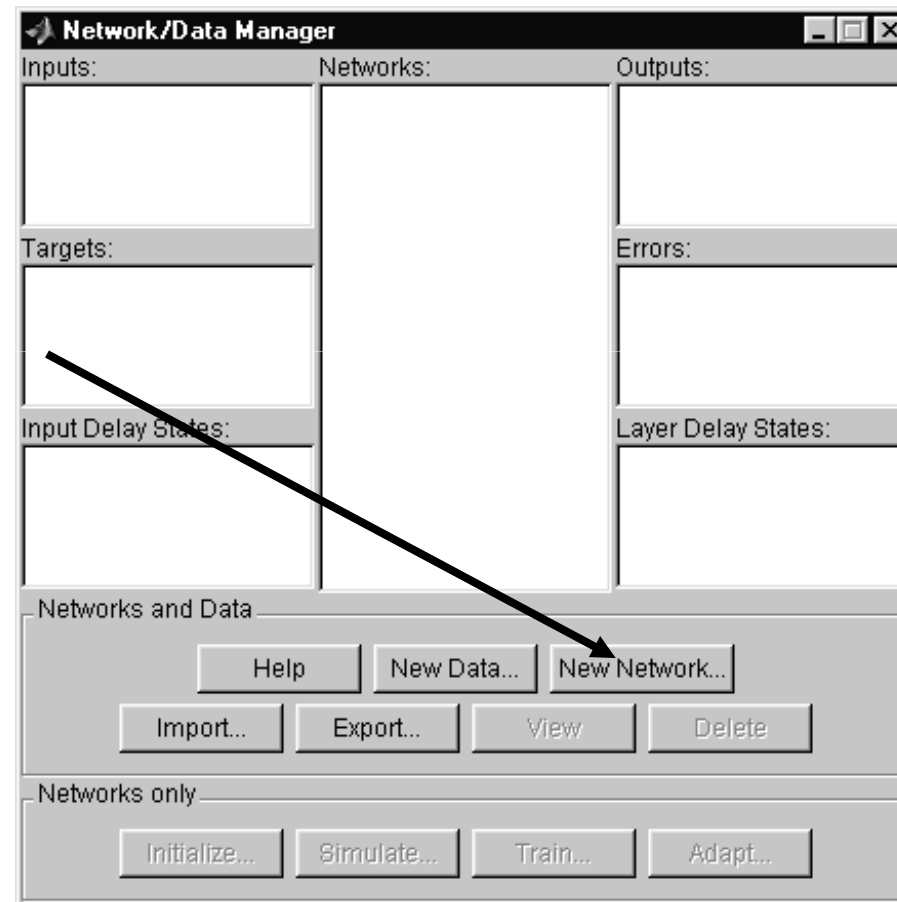




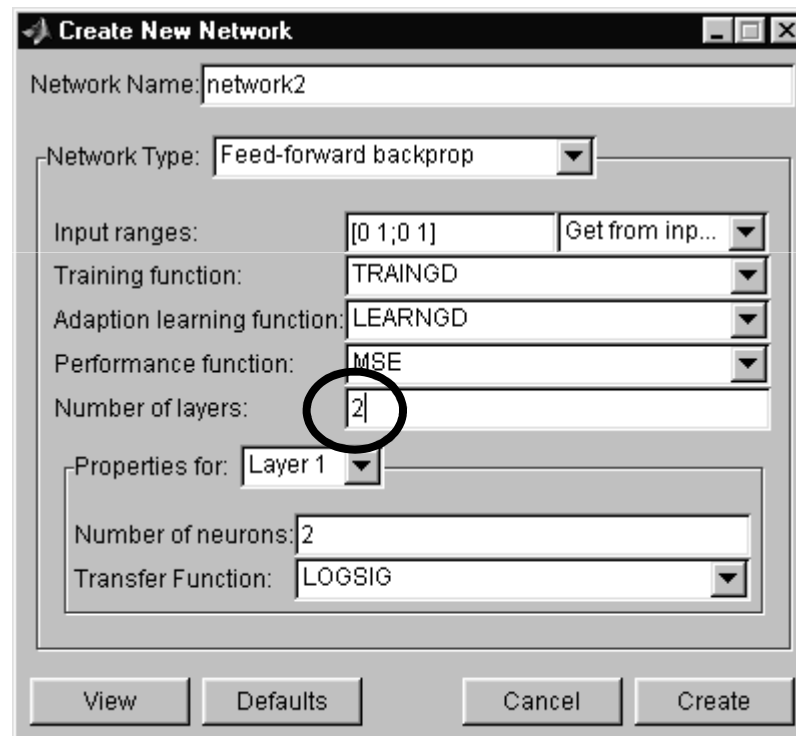
# **Passos para a Criação de uma Rede Neural**

- Definir os padrões
- Criar a rede
- Inicializar a rede
- Definir os parâmetros de treinamento
- Treinar a rede
- Testar a rede

# Criando a Rede Neural



# Criando a Rede Neural



The image shows the 'Create New Network' dialog box in MATLAB. The 'Network Name' field is set to 'network2'. The 'Network Type' is 'Feed-forward backprop'. The 'Input ranges' are '[0 1; 0 1]' with a 'Get from inp...' button. The 'Training function' is 'TRAINGD', the 'Adaption learning function' is 'LEARNGD', and the 'Performance function' is 'MSE'. The 'Number of layers' is '2', which is circled. The 'Properties for:' dropdown is set to 'Layer 1'. The 'Number of neurons' is '2' and the 'Transfer Function' is 'LOGSIG'. At the bottom are 'View', 'Defaults', 'Cancel', and 'Create' buttons.

**Create New Network**

Network Name: network2

Network Type: Feed-forward backprop

Input ranges: [0 1; 0 1] Get from inp...

Training function: TRAINGD

Adaption learning function: LEARNGD

Performance function: MSE

Number of layers: 2

Properties for: Layer 1

Number of neurons: 2

Transfer Function: LOGSIG

View Defaults Cancel Create

# Criando a Rede Neural

**Create New Network**

Network Name: network2

Network Type: Feed-forward backprop

Input ranges: [0 1;0 1] Get from inp...

Training function: TRAINGD

Adaption learning function: LEARNGD

Performance function: MSE

Number of layers: 2

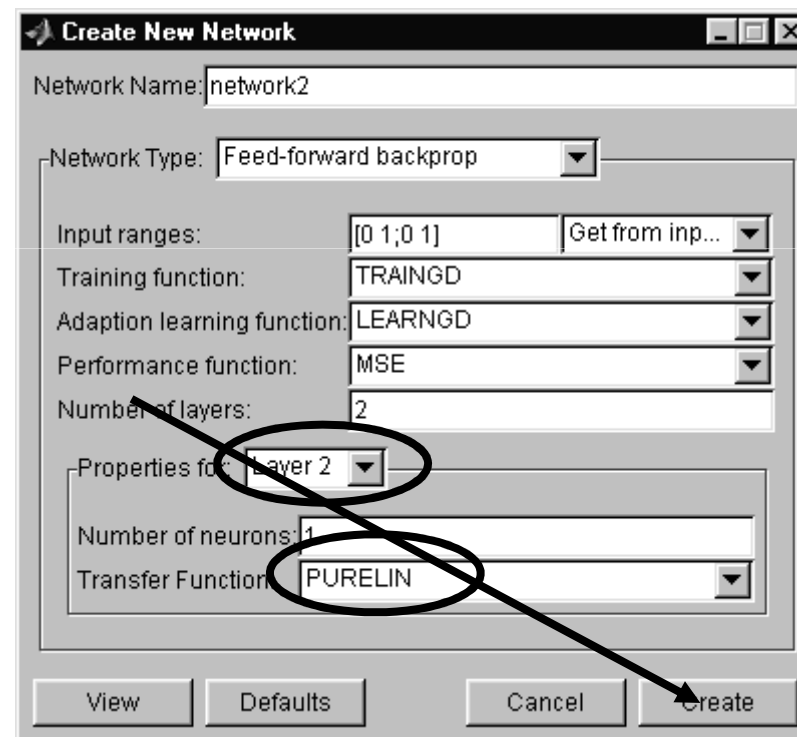
Properties for: Layer 1

Number of neurons: 2

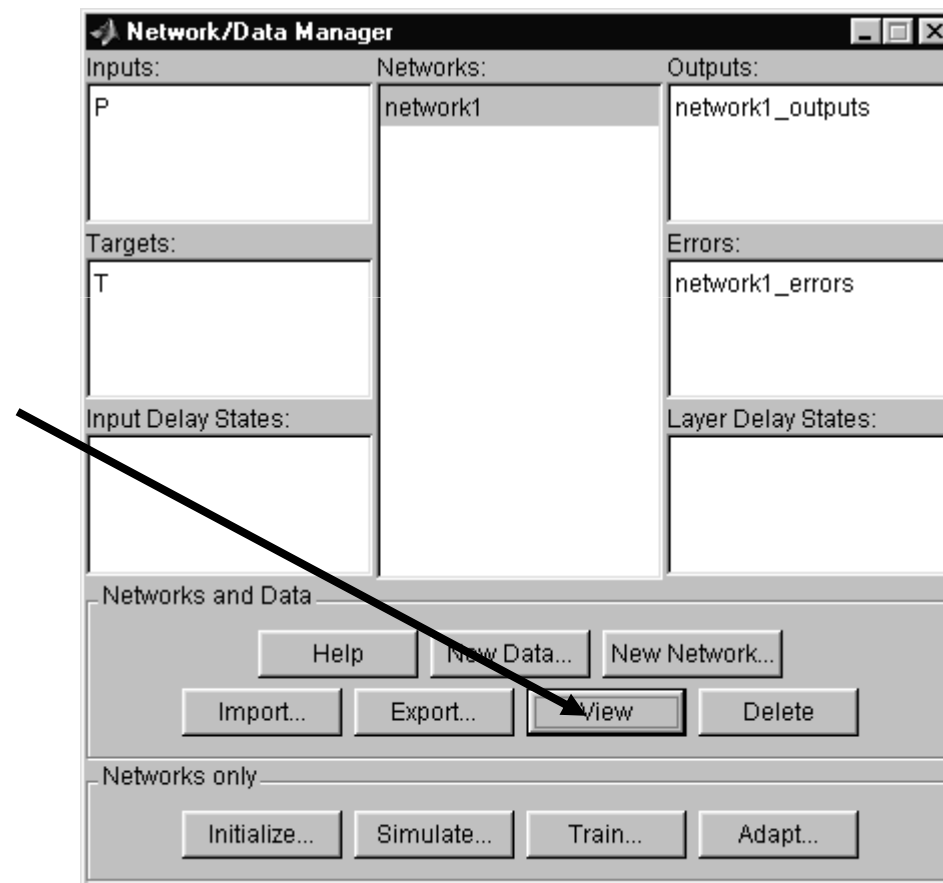
Transfer Function: LOGSIG

View Defaults Cancel Create

# Criando a Rede Neural

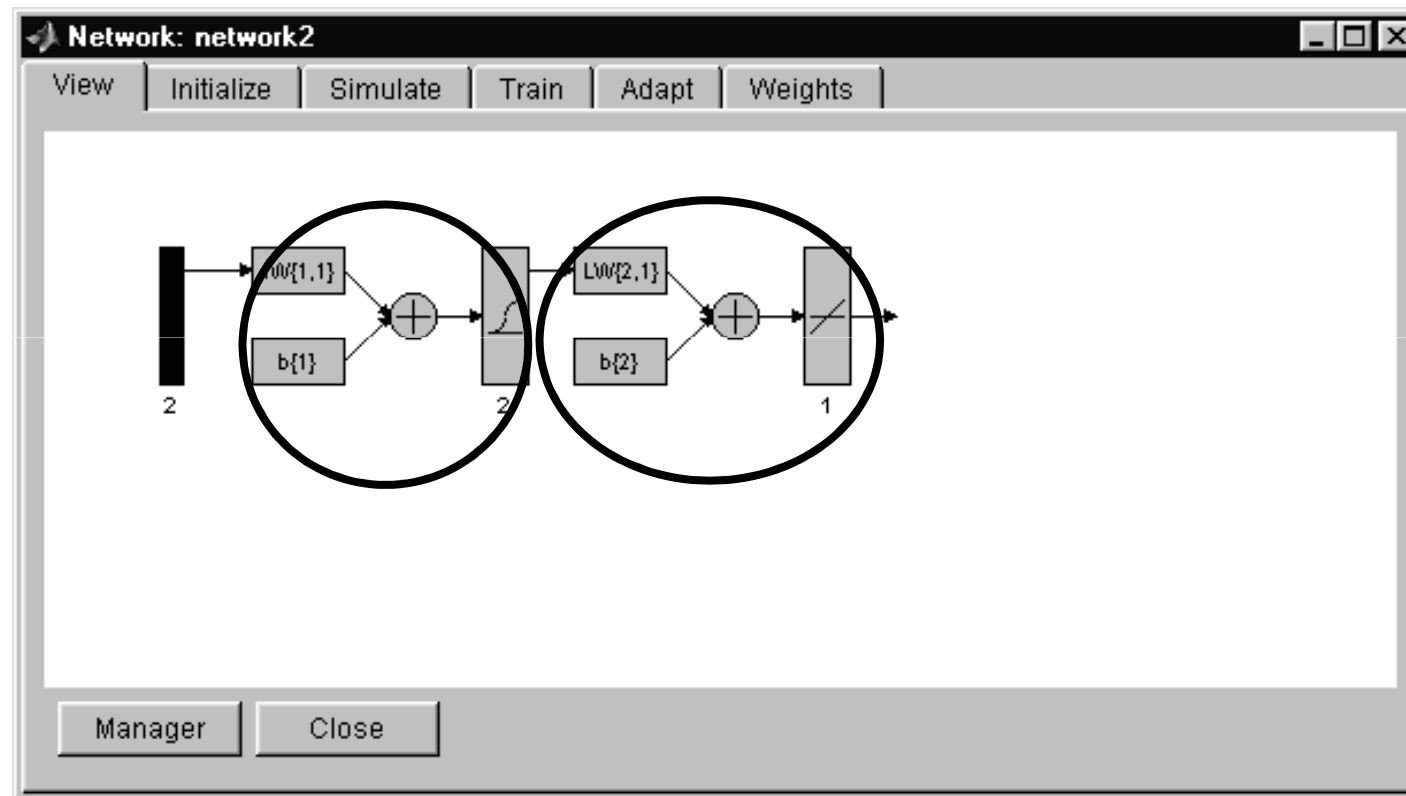


# Visualizando a Rede Neural





# Visualizando a Rede Neural

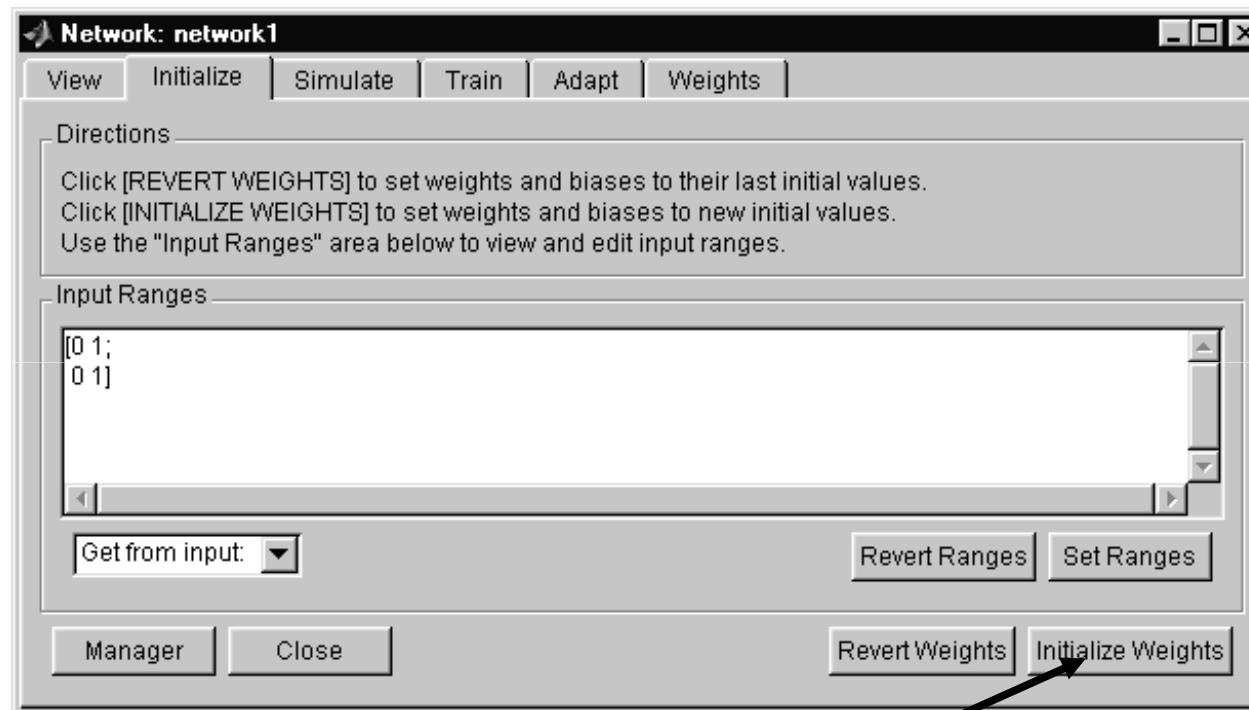




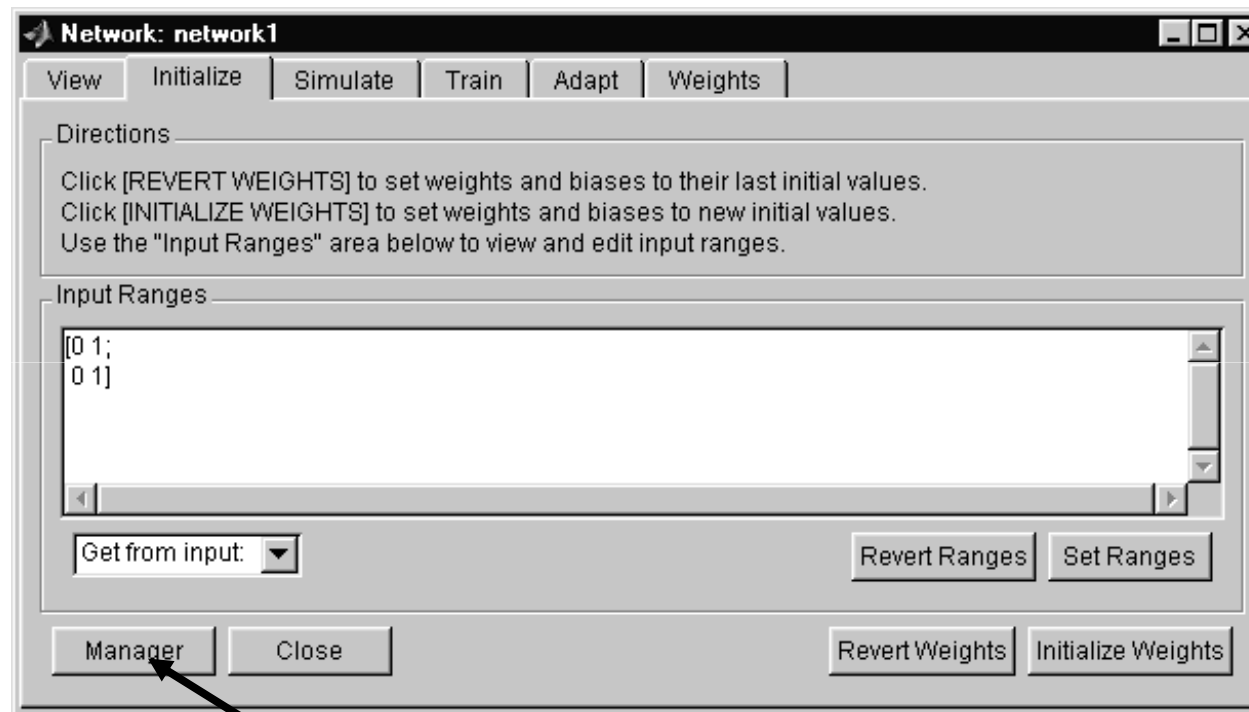
# **Passos para a Criação de uma Rede Neural**

- Definir os padrões
- Criar a rede
- Inicializar a rede
- Definir os parâmetros de treinamento
- Treinar a rede
- Testar a rede

# Inicializando a Rede Neural



# Inicializando a Rede Neural

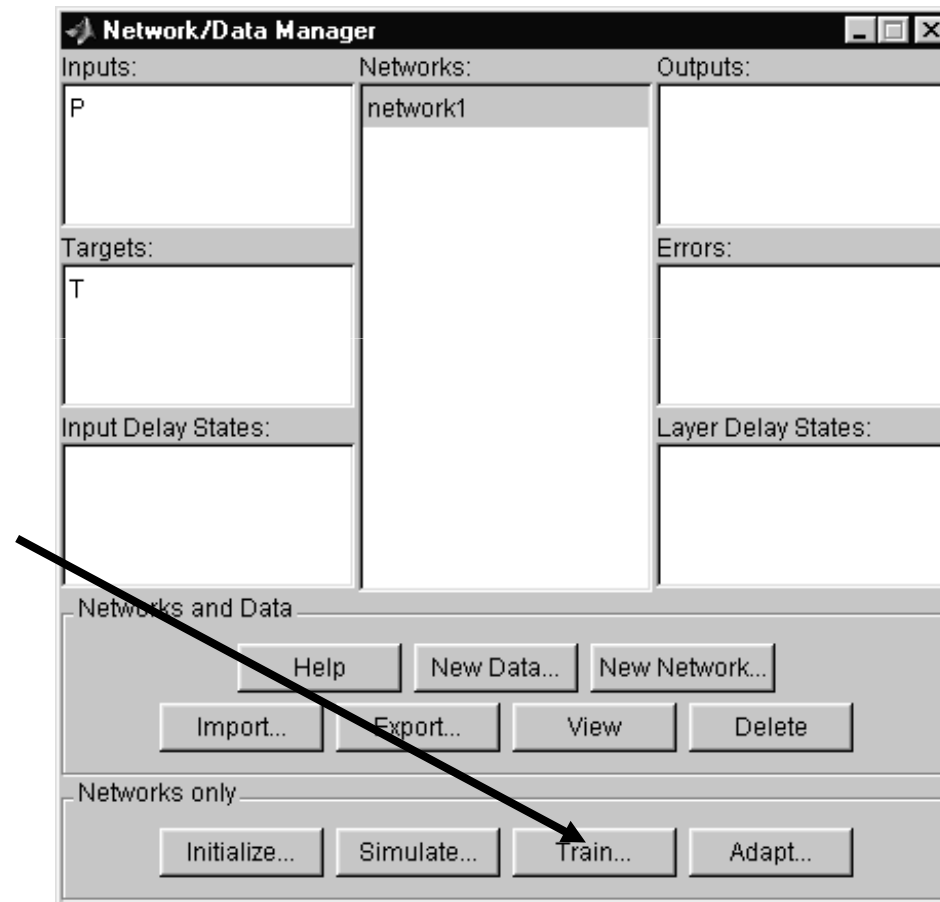




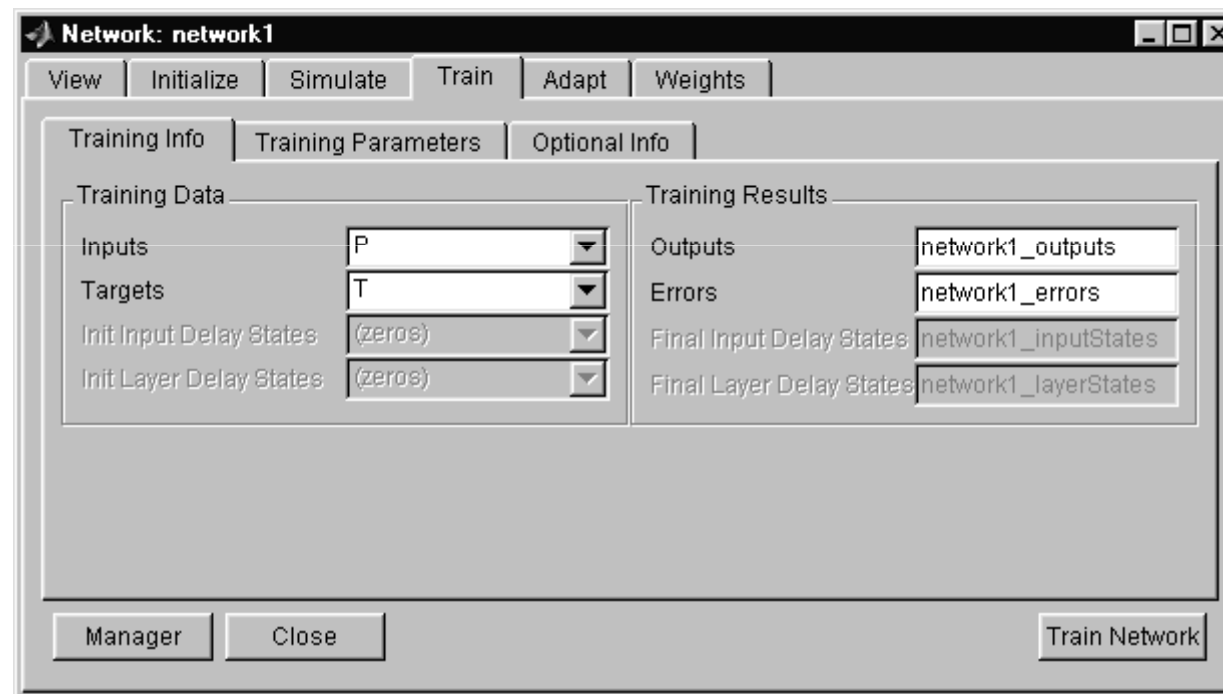
# **Passos para a Criação de uma Rede Neural**

- Definir os padrões
- Criar a rede
- Inicializar a rede
- Definir os parâmetros de treinamento
- Treinar a rede
- Testar a rede

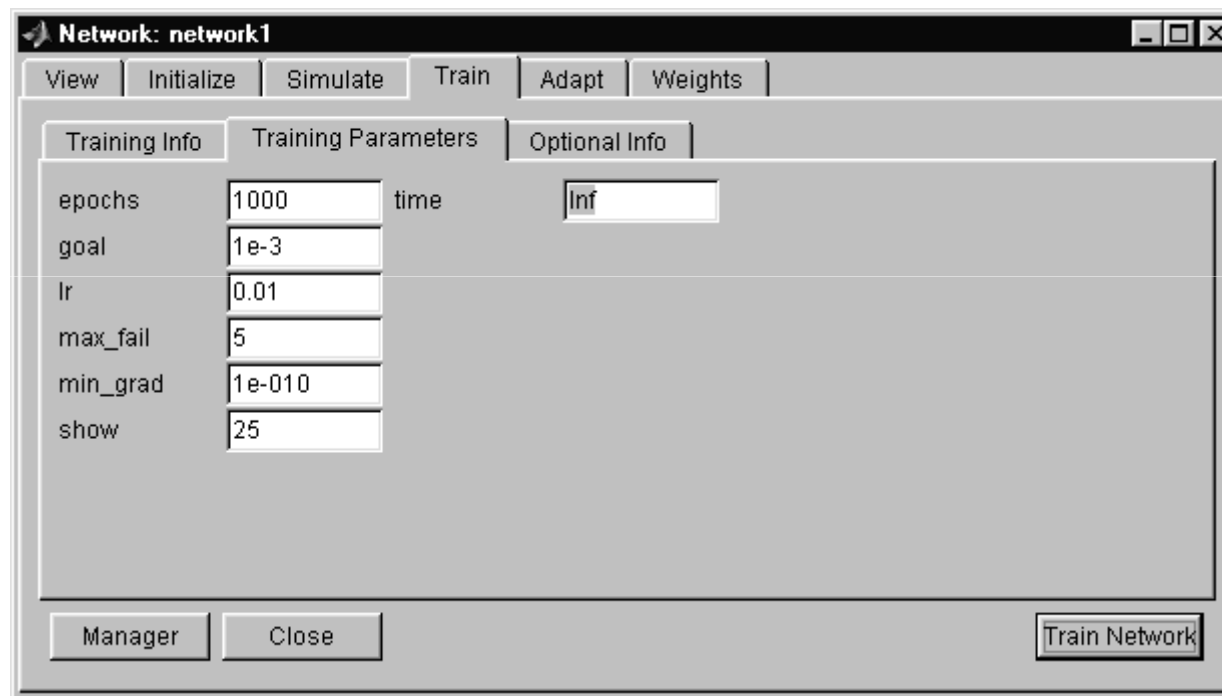
# Definindo parâmetros de treinamento



# Definindo parâmetros de treinamento



# Definindo parâmetros de treinamento



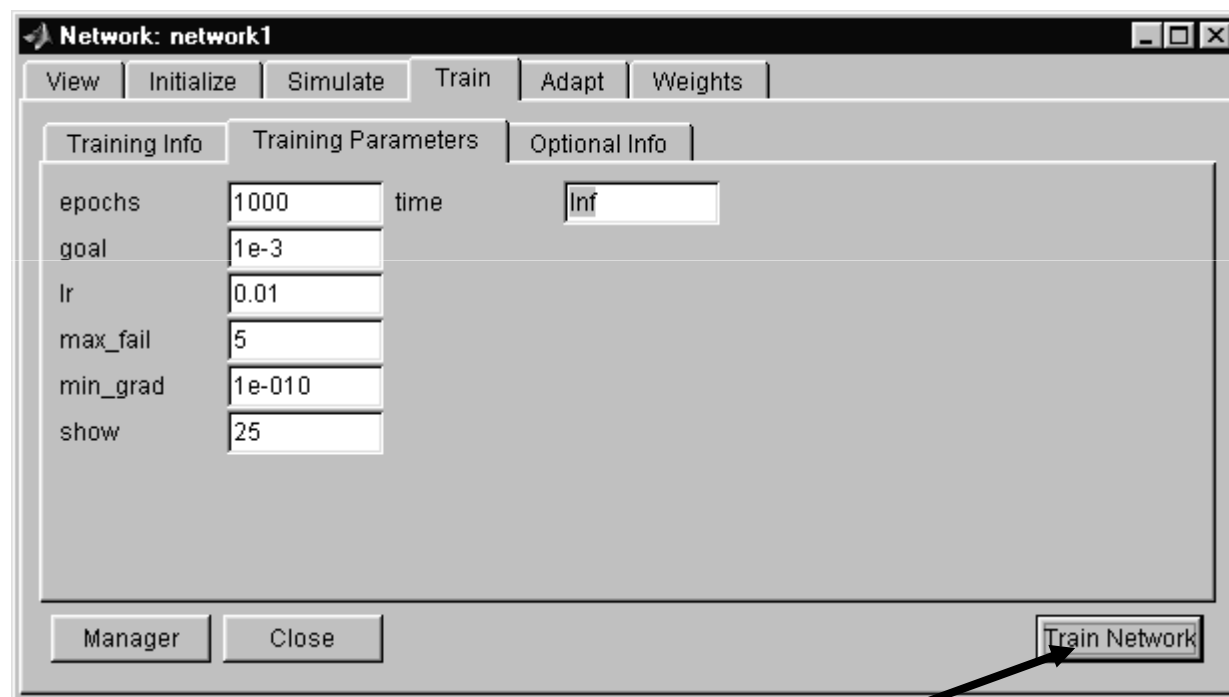




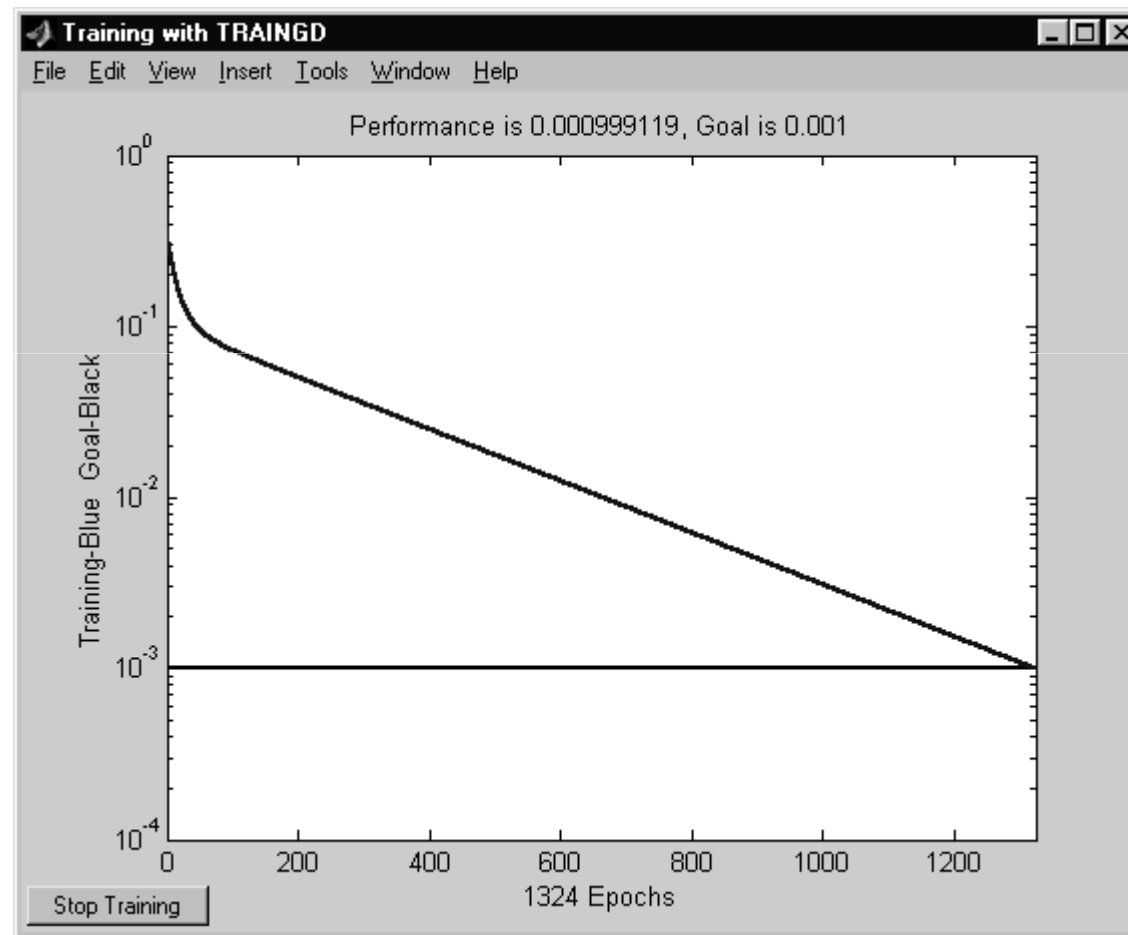
# **Passos para a Criação de uma Rede Neural**

- Definir os padrões
- Criar a rede
- Inicializar a rede
- Definir os parâmetros de treinamento
- Treinar a rede
- Testar a rede

# Treinando a Rede Neural



# Treinando a Rede Neural

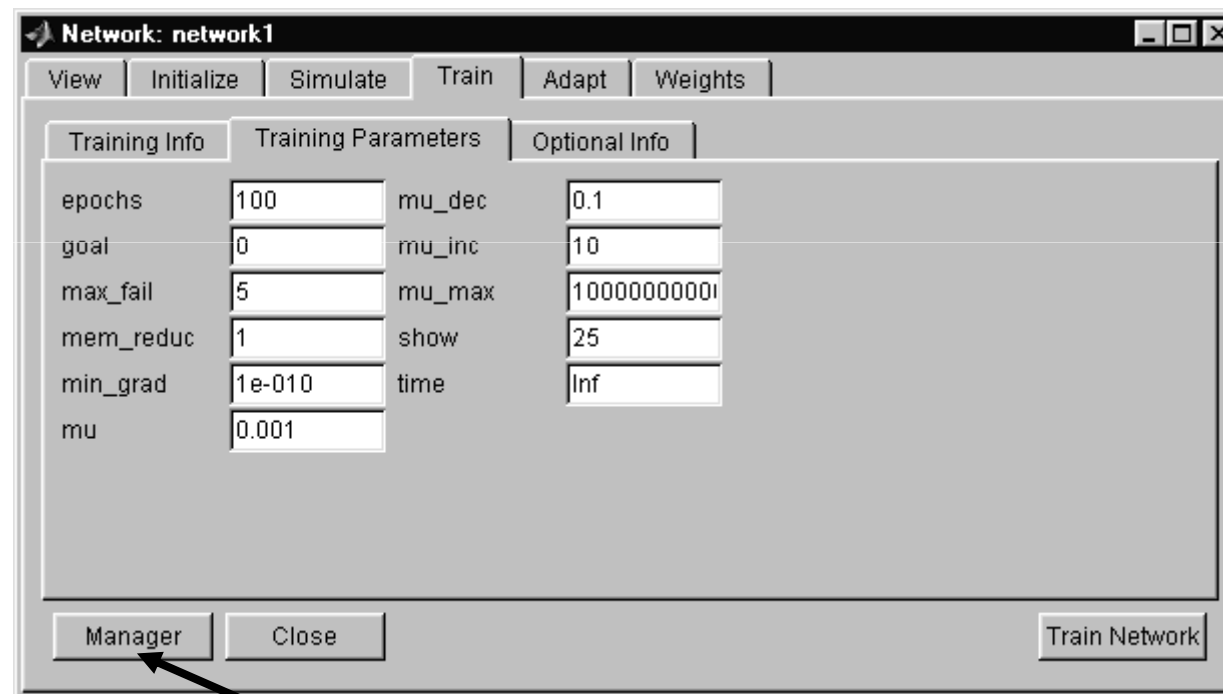




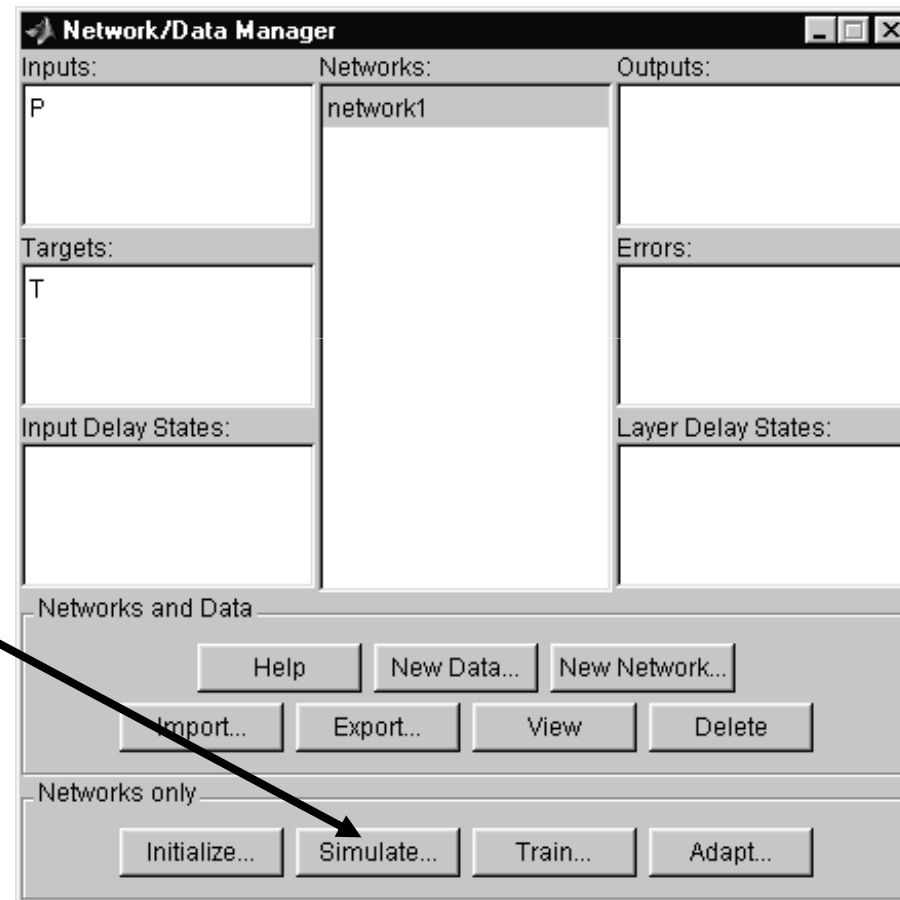
# **Passos para a Criação de uma Rede Neural**

- Definir os padrões
- Criar a rede
- Inicializar a rede
- Definir os parâmetros de treinamento
- Treinar a rede
- Testar a rede

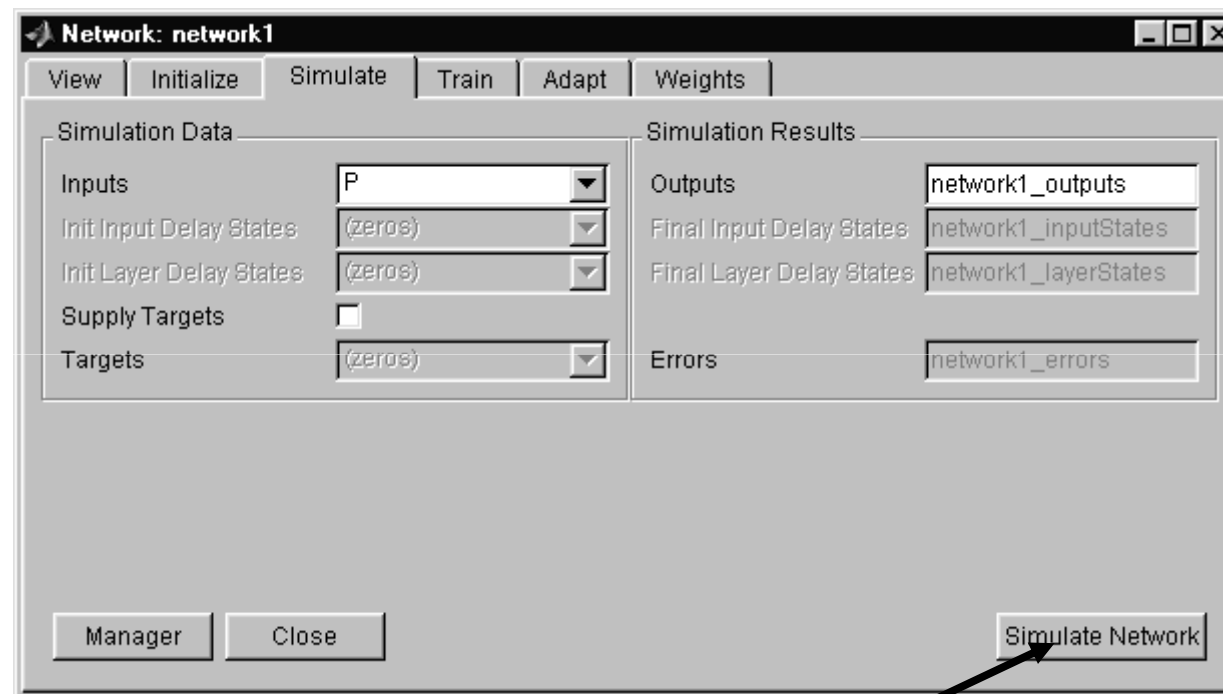
# Testando a Rede Neural



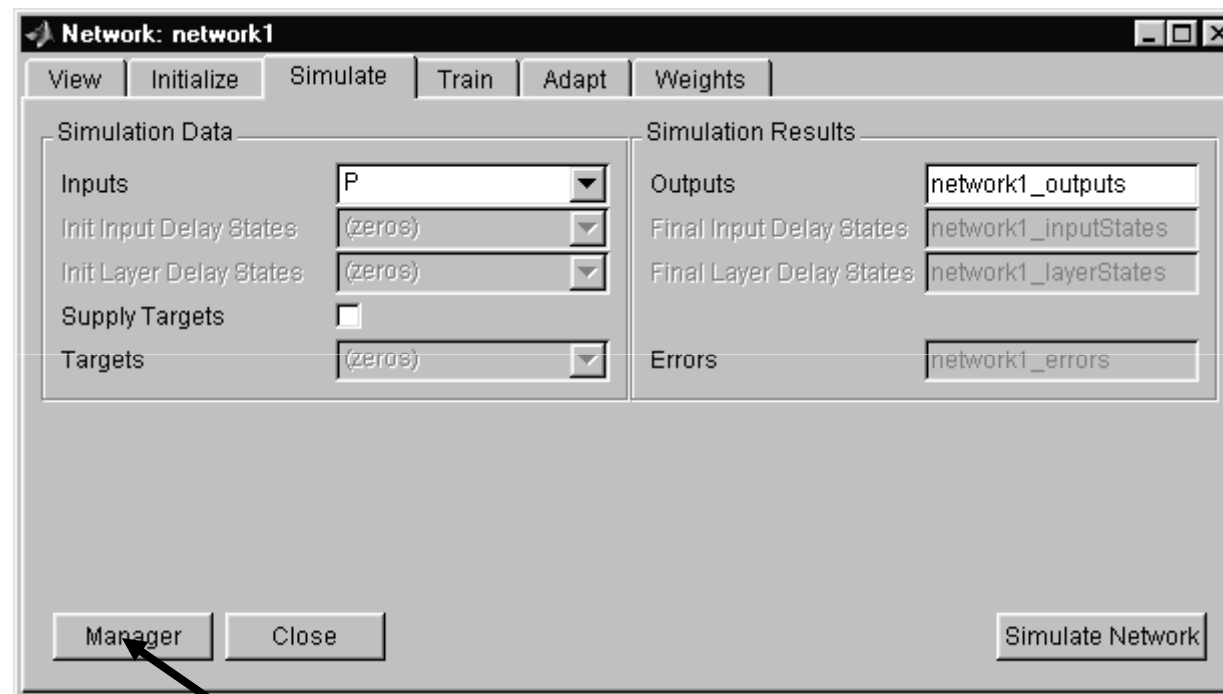
# Testando a Rede Neural



# Testando a Rede Neural

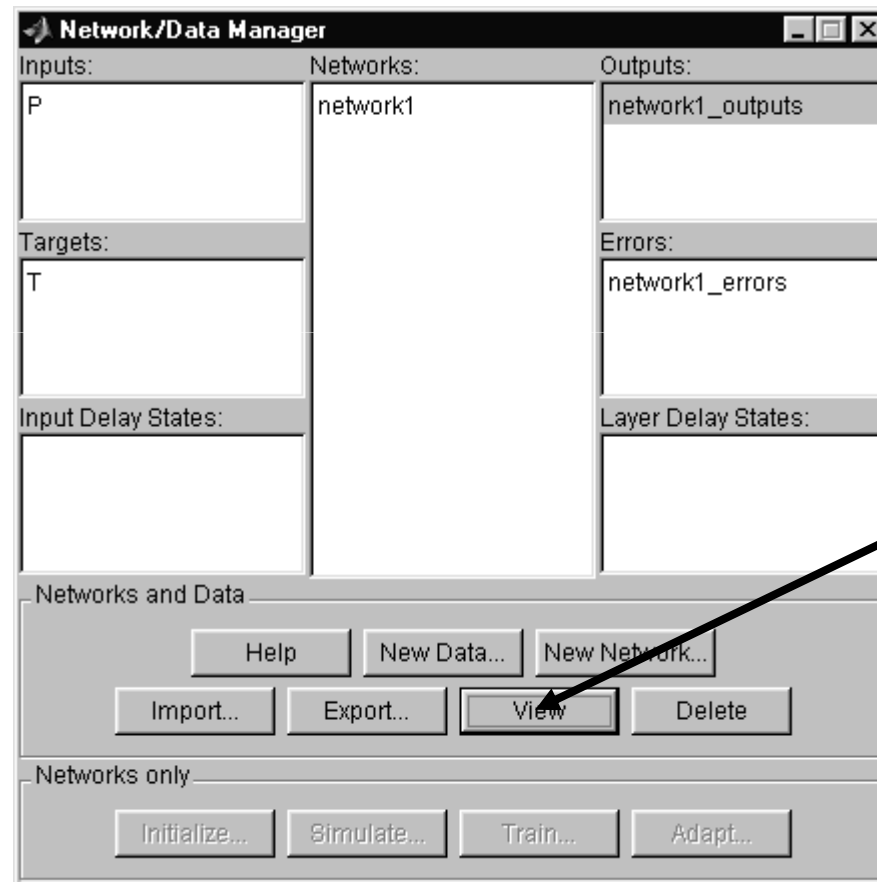


# Testando a Rede Neural

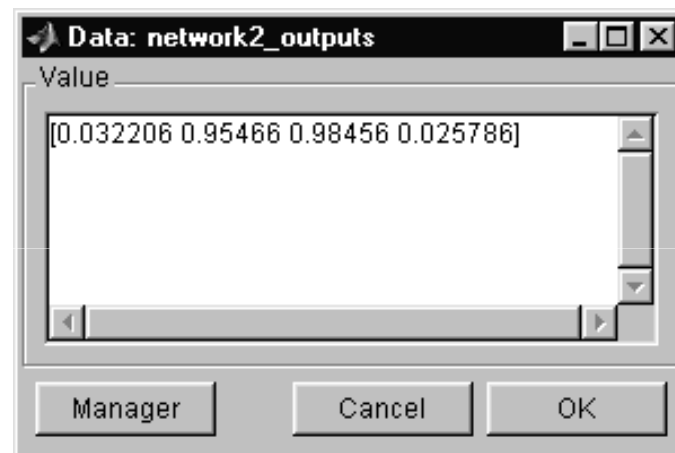




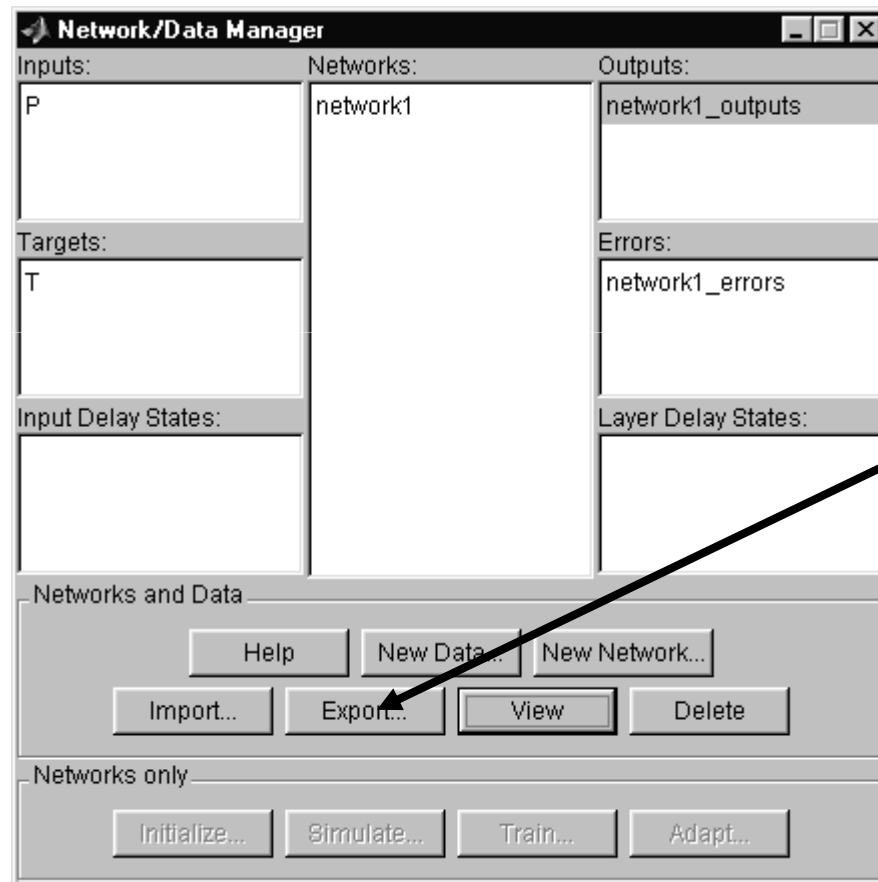
# Testando a Rede Neural



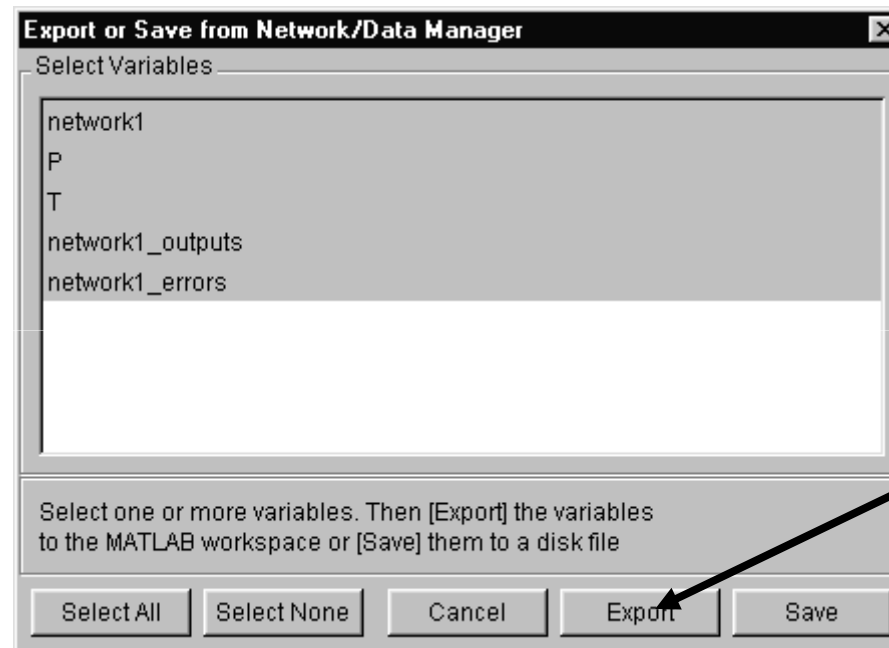
# Testando a Rede Neural



# Exportando os Dados



# Exportando os Dados





**FIM DO MÓDULO REDES  
NEURAIS NO MATLAB**