

Toolbox de Redes Neurais

Matrix-Laboratory **Matlab**

Toolbox de Redes Neurais

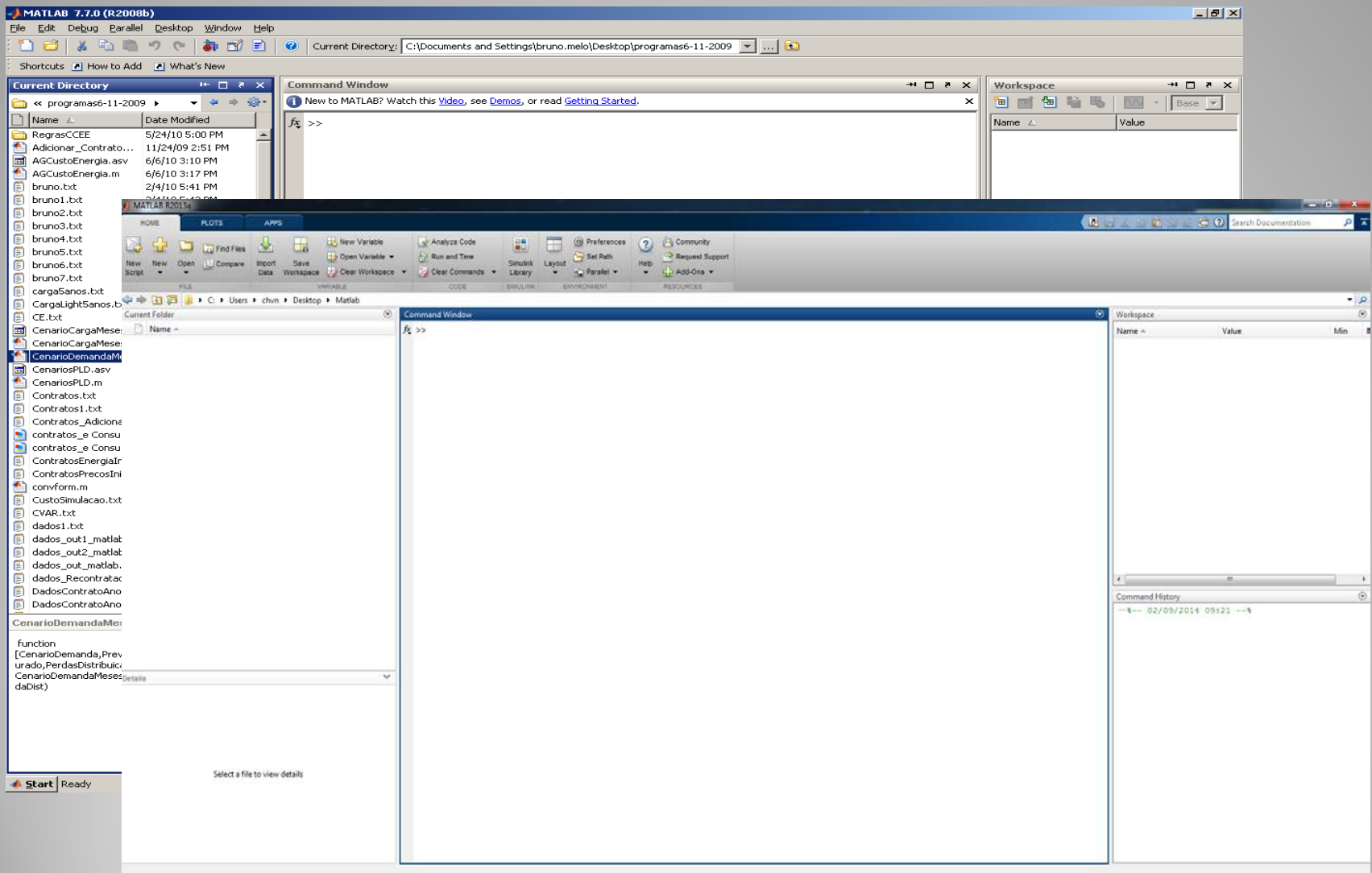
- Introdução ao Matlab
 - Linha de comando
 - NNTool
- Estudo de Casos
 - Análise de Crédito Bancário (Classificação).
 - Localização de Faltas (Inferência).
 - Sensação Térmica (Previsão).

Toolbox de Redes Neurais

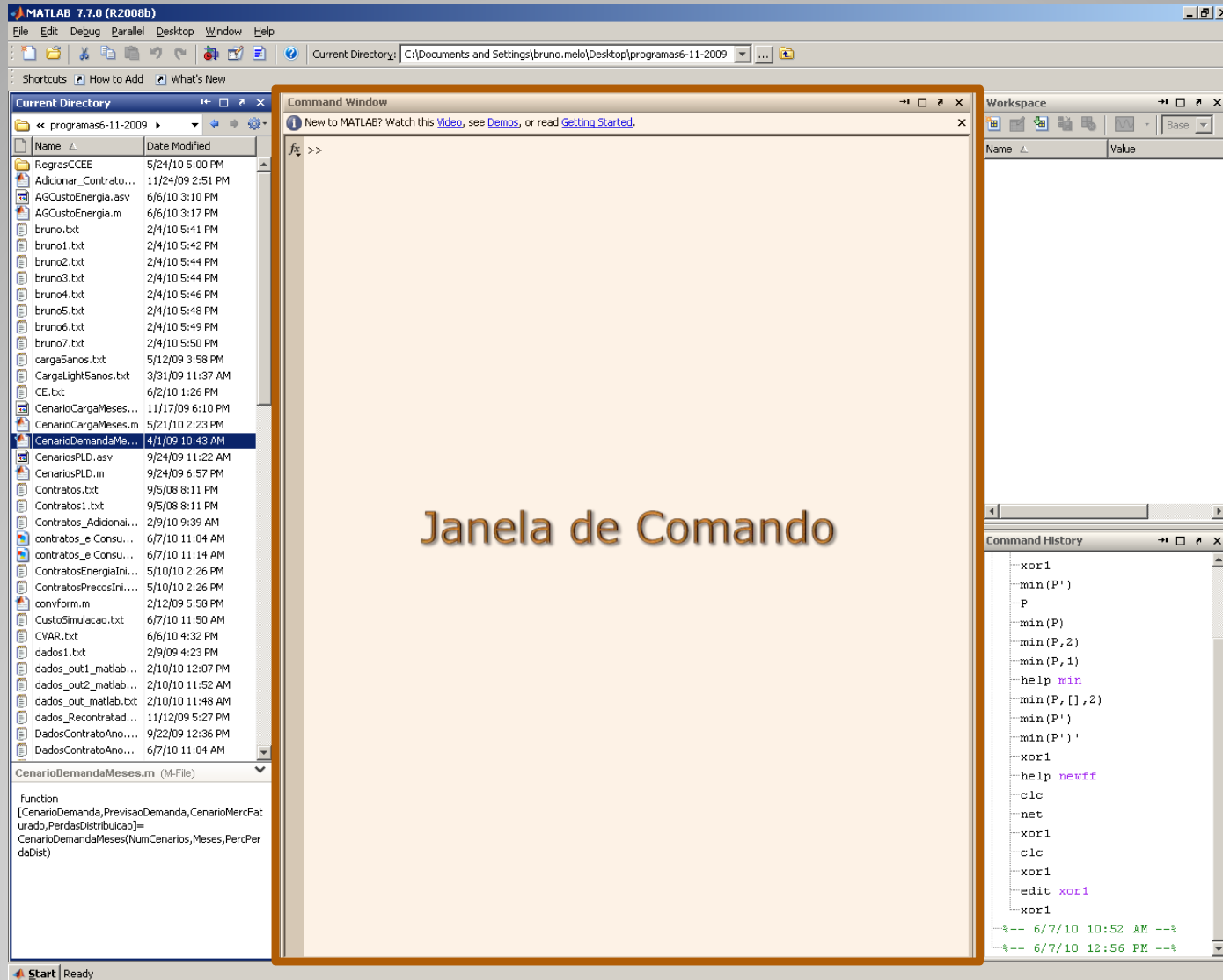
- **Introdução ao Matlab**
 - Linha de comando
 - NNTool
- **Estudo de Casos**
 - Análise de Crédito Bancário (Classificação).
 - Localização de Faltas (Inferência).
 - Sensação Térmica (Previsão).

- **MATLAB** → **MAT**rix **LAB**oratory
- Programação baseada em **Matrizes**
- **Vetores** e **escalares** são tratados como matrizes (**1xN**, **Nx1**, **NxN**)

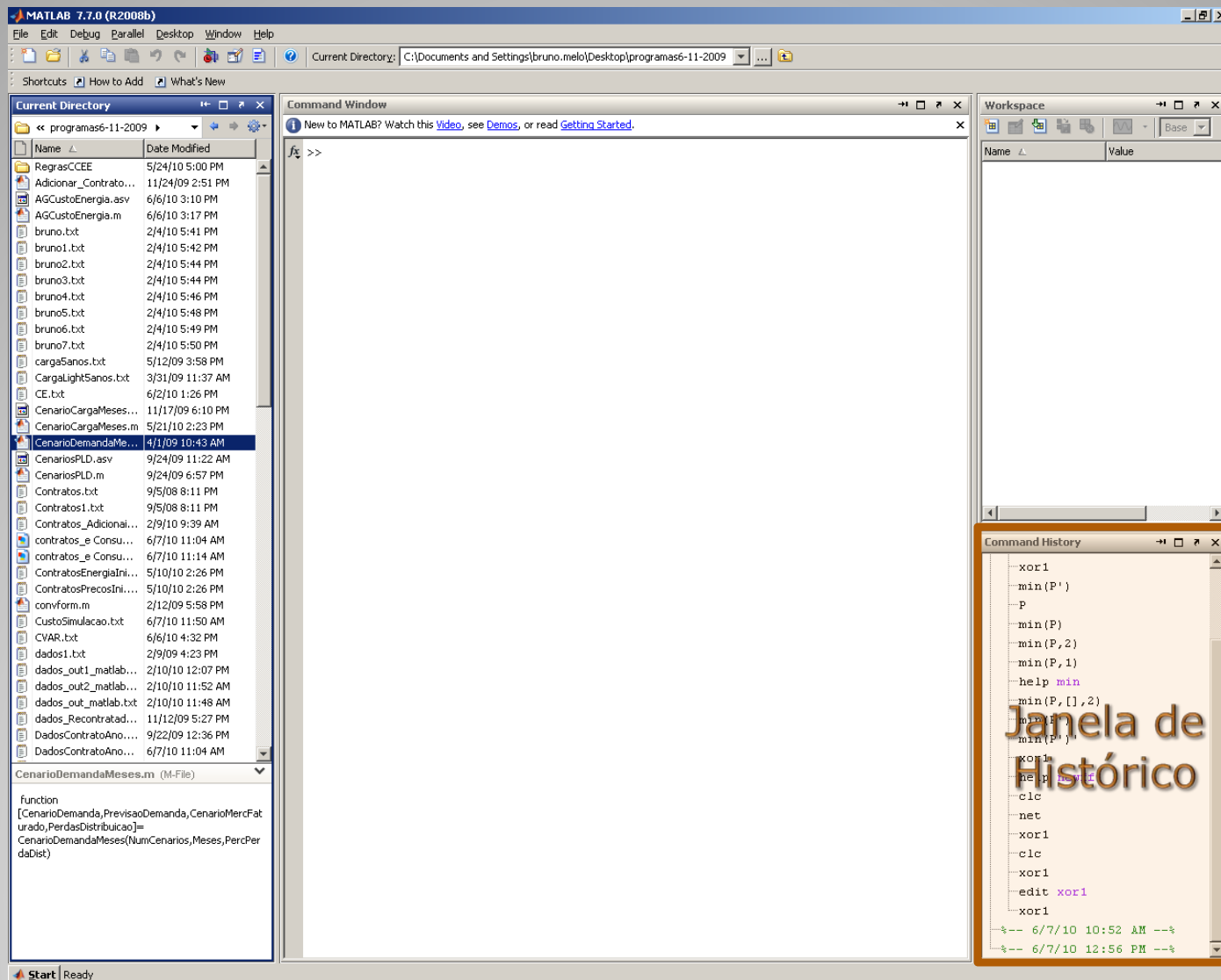
Introdução ao Ambiente



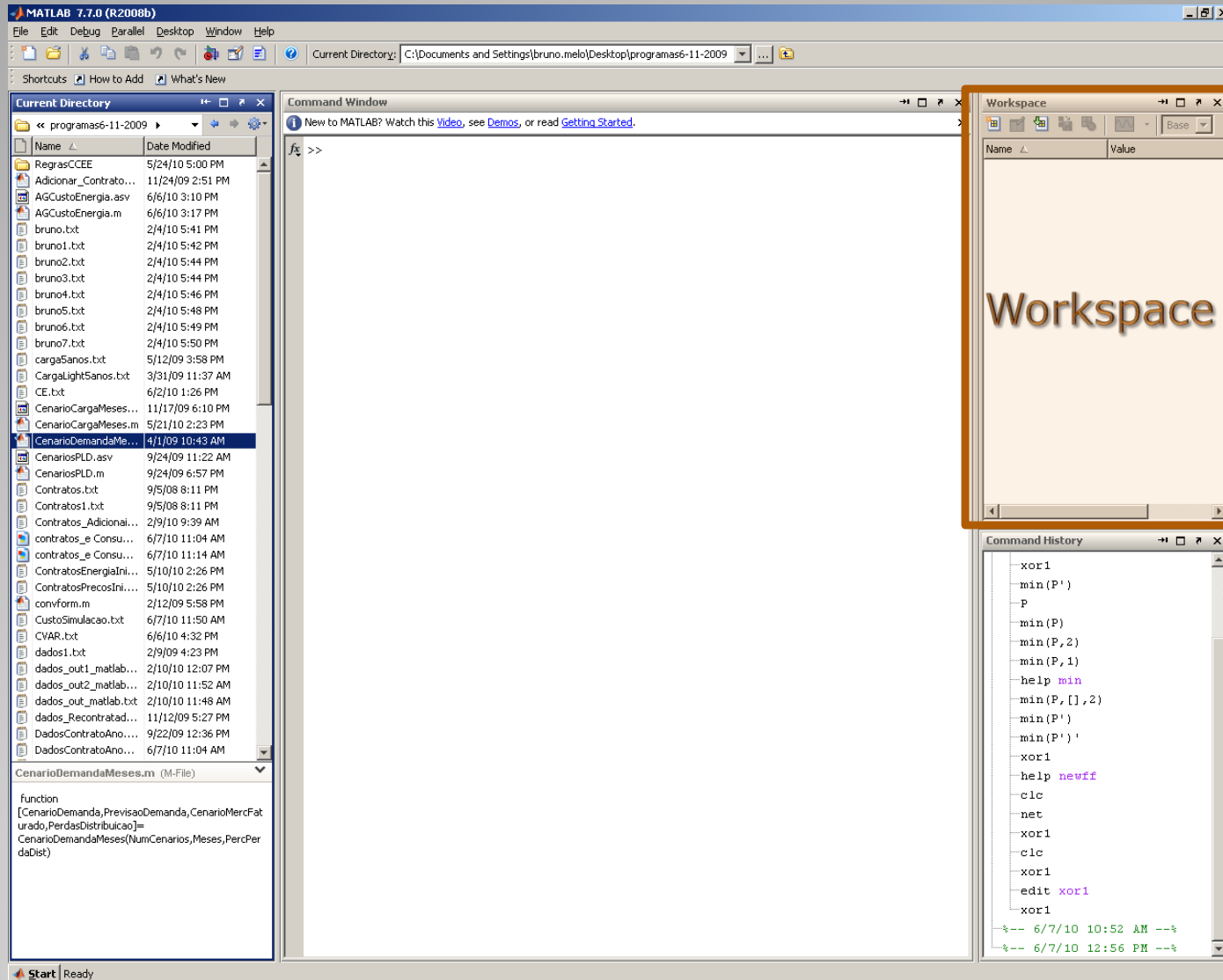
Tela Principal



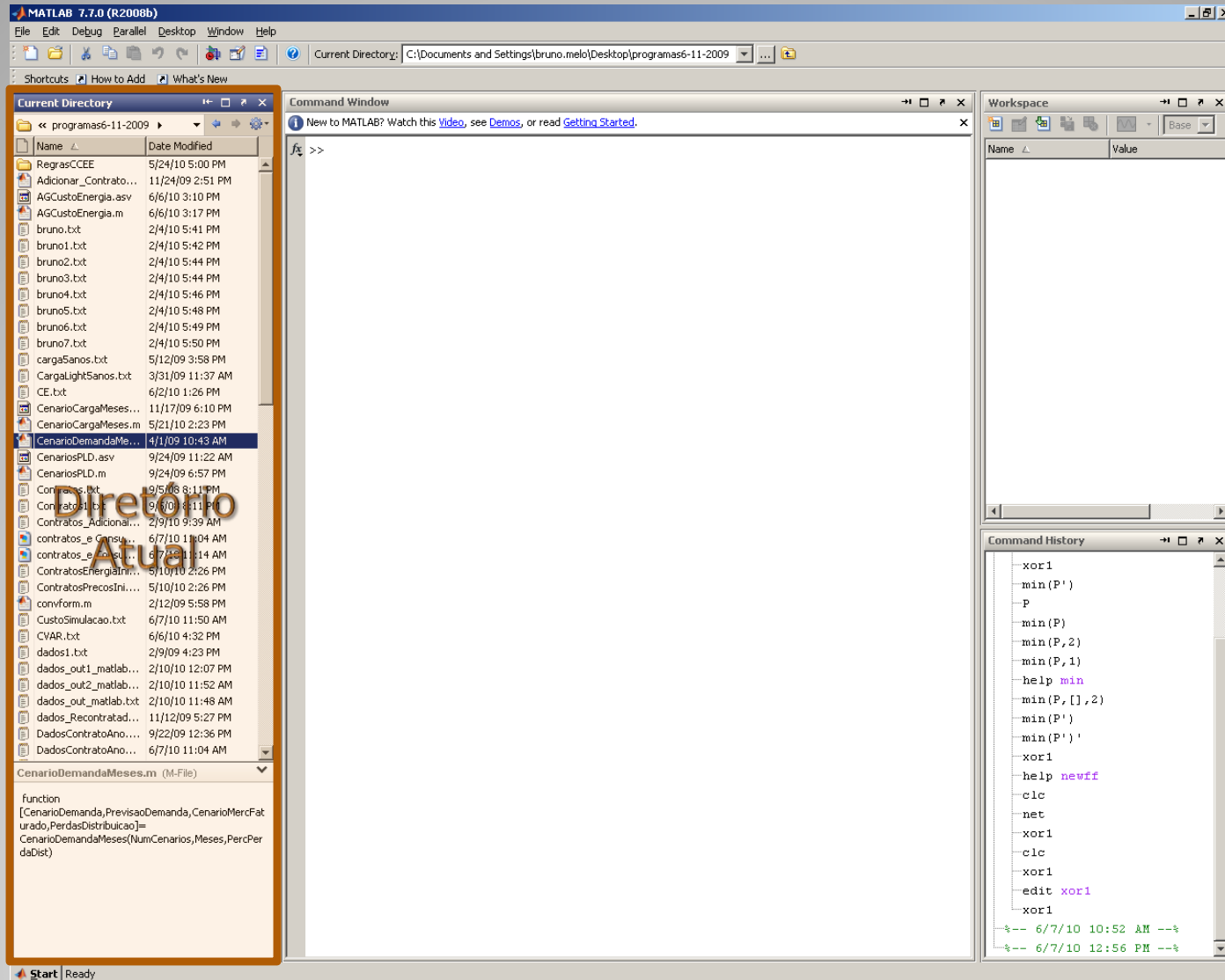
Tela Principal



Tela Principal

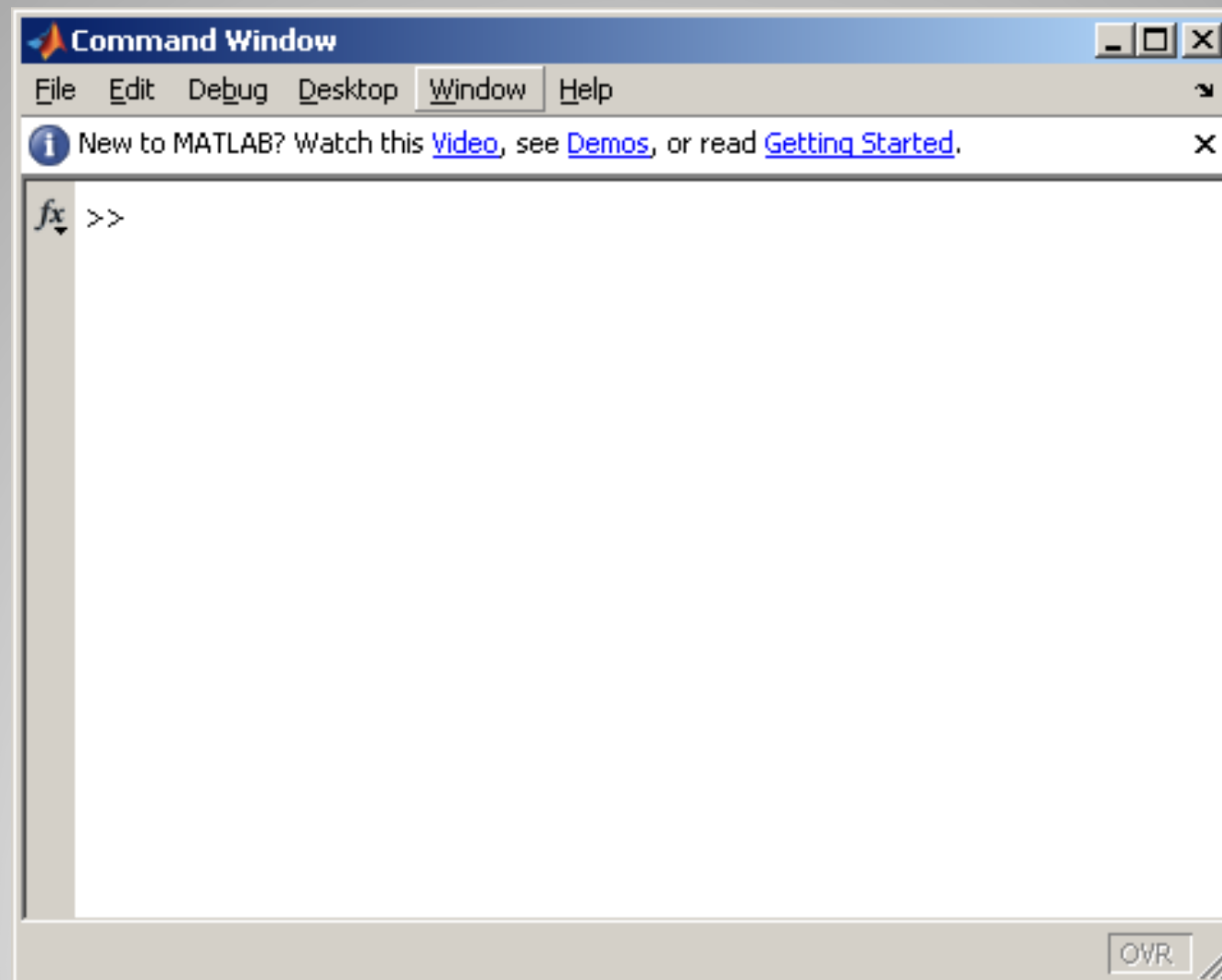


Tela Principal



Tela Principal





Janela de Comando

Quadrado Mágico

```
Command Window
File Edit Debug Desktop Window Help

>> help magic
MAGIC Magic square.
MAGIC(N) is an N-by-N matrix constructed from the integers
1 through N^2 with equal row, column, and diagonal sums.
Produces valid magic squares for all N > 0 except N = 2.

Reference page in Help browser
doc magic

>> m = magic(4)

m =

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

>> sum(m)

ans =

    34    34    34    34

>> sum(m,2)

ans =

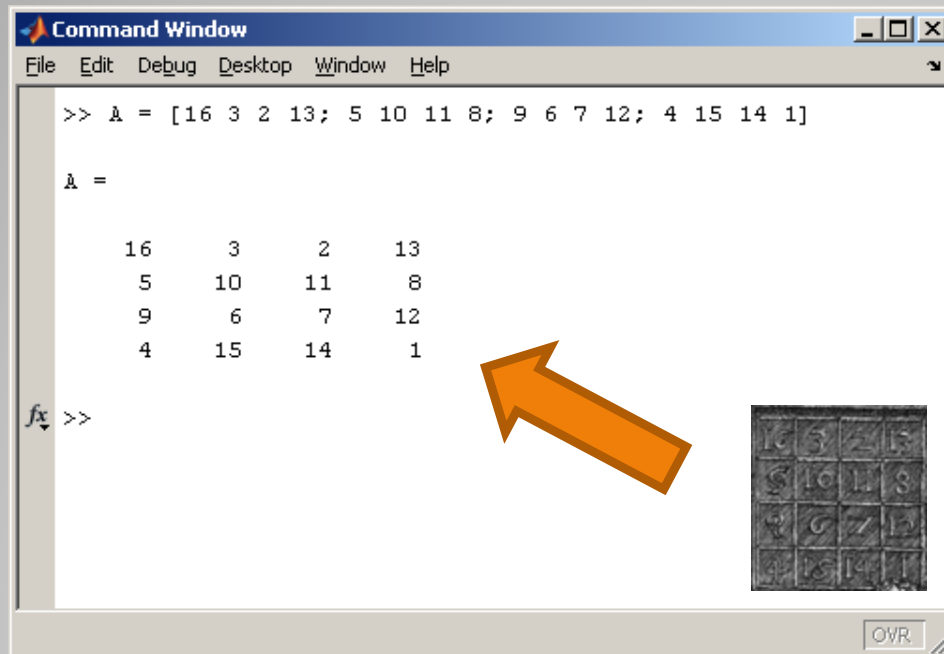
    34
    34
    34
    34

fx >>
```



Matriz $N \times N$ construída a partir de inteiros de 1 a N^2 em que as somas das colunas são iguais às somas das linhas.

Definindo uma Matriz Explicitamente



The image shows a MATLAB Command Window with the following content:

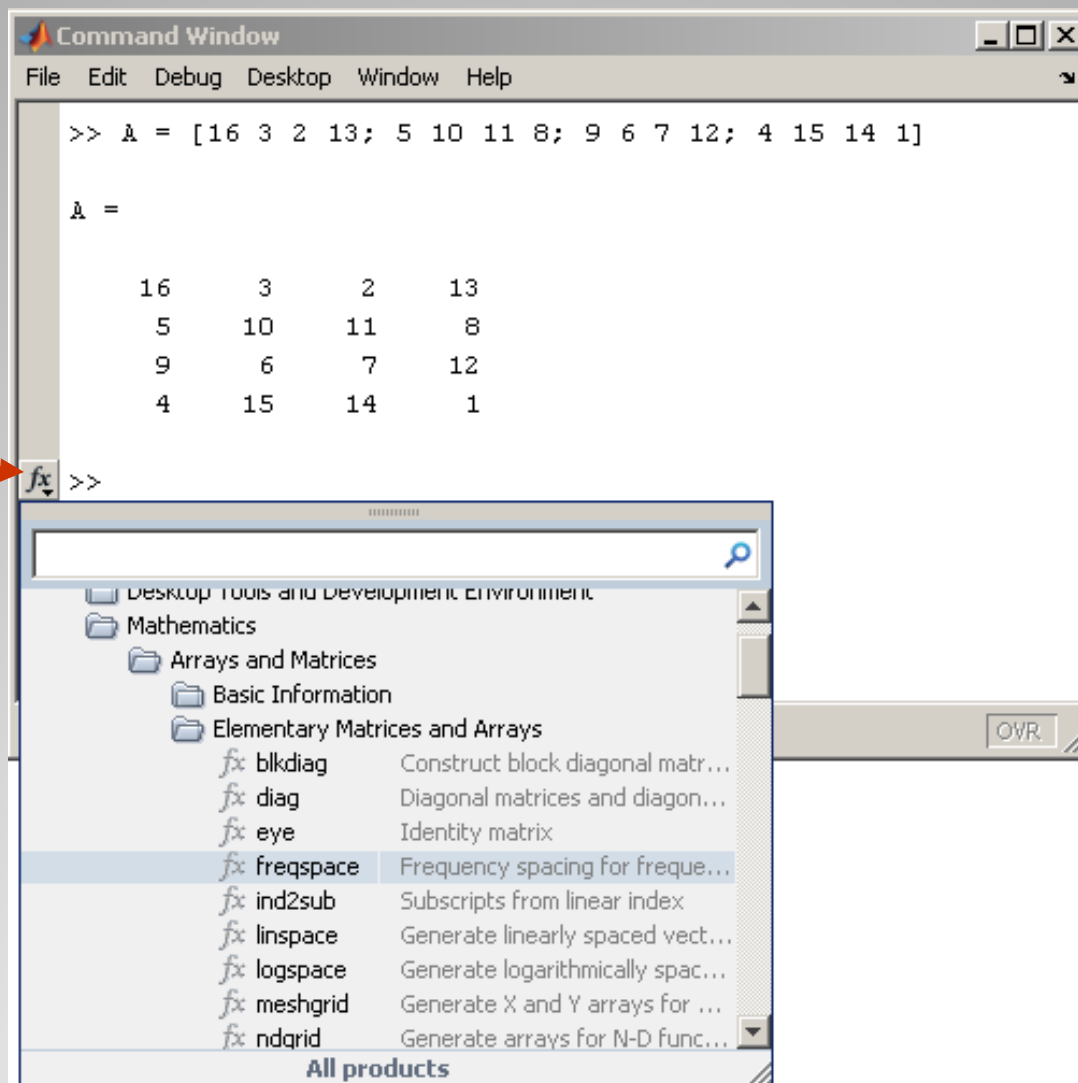
```
>> A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

The output displays the matrix A as a 4x4 grid of numbers:

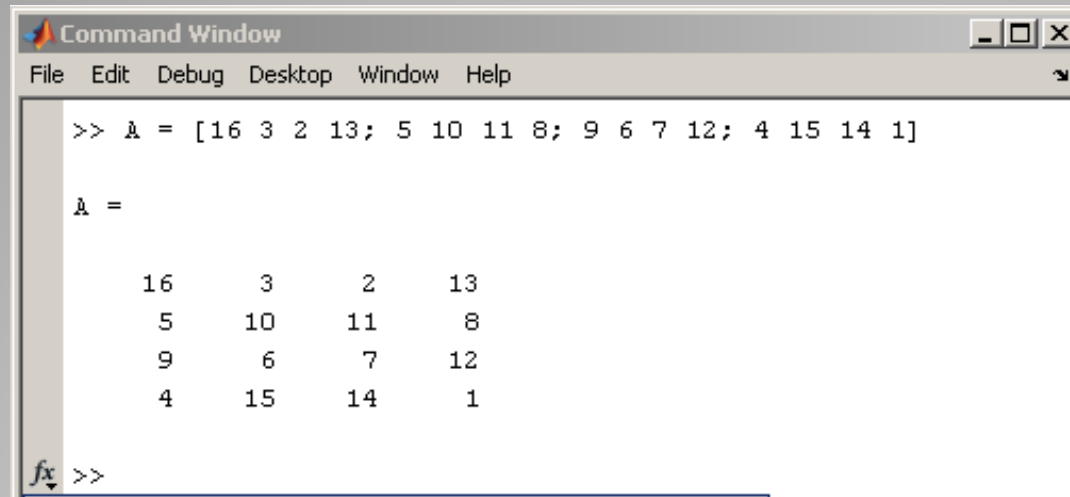
16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

A large orange arrow points from the text "ponto-e-vírgula" in the list to the semicolon in the matrix definition. A small 4x4 grid of numbers is also shown to the right of the arrow, representing the matrix A.

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



Janela de Comando



Command Window

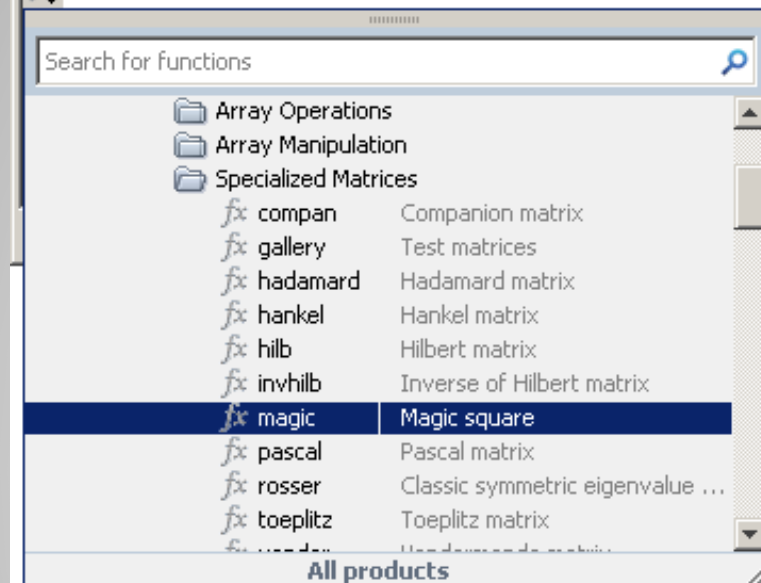
File Edit Debug Desktop Window Help

```
>> 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

fx >>



Search for functions

- Array Operations
- Array Manipulation
- Specialized Matrices
 - fx compan Companion matrix
 - fx gallery Test matrices
 - fx hadamard Hadamard matrix
 - fx hankel Hankel matrix
 - fx hilb Hilbert matrix
 - fx invhilb Inverse of Hilbert matrix
 - fx magic Magic square**
 - fx pascal Pascal matrix
 - fx rosser Classic symmetric eigenvalue ...
 - fx toeplitz Toeplitz matrix
 - fx vandermonde Vandermonde matrix

All products

magic

[More Help...](#)

Magic square

M = **magic**(**n**) returns an **n**-by-**n** matrix constructed from the integers 1 through **n**² with equal row and column sums. The order **n** must be a scalar greater than or equal to 3.

Janela de Comando

Toolbox de Redes Neurais

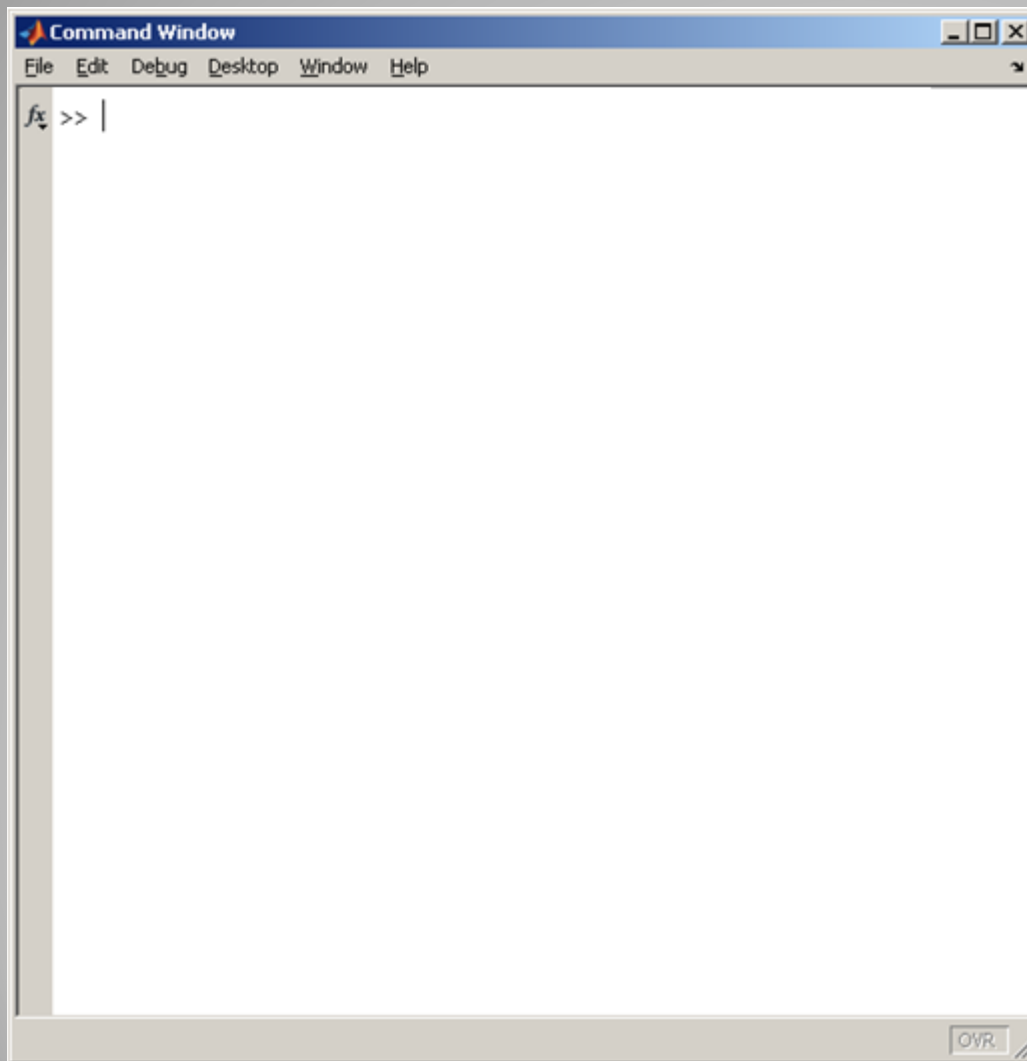
- Introdução ao Matlab
 - Linha de comando
 - NNTool
- Estudo de Casos
 - Análise de Crédito Bancário (Classificação).
 - Localização de Faltas (Inferência).
 - Sensação Térmica (Previsão).

- Definição do problema
- Inicialização da rede
- Definição dos parâmetros de treinamento
- Treinamento da rede
- Teste da rede

Passos p/ a criação de uma RN

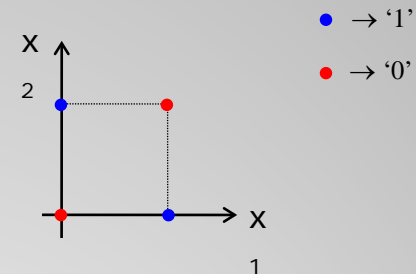
- Definição do problema
- Inicialização da rede
- Definição dos parâmetros de treinamento
- Treinamento da rede
- Teste da rede

Passos p/ a criação de uma RN



Definição do problema

O problema do OU-Exclusivo



X ₁	X ₂	valor
0	0	0
0	1	1
1	0	1
1	1	0

Passos p/ a criação de uma RN

- Definição do problema
- Inicialização da rede
- Definição dos parâmetros de treinamento
- Treinamento da rede
- Teste da rede

Passos p/ a criação de uma RN

```
Command Window
File Edit Debug Desktop Window Help

>> % Padrões de entrada:
>> P = [0 0 1 1; 0 1 0 1]

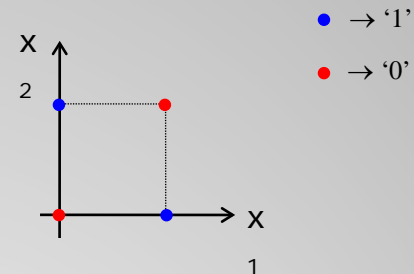
P =

    0     0     1     1
    0     1     0     1

fx >> |
```

Definição do problema

O problema do OU-Exclusivo



X ₁	X ₂	valor
0	0	0
0	1	1
1	0	1
1	1	0

Passos p/ a criação de uma RN

```
Command Window
File Edit Debug Desktop Window Help

>> % Padrões de entrada:
>> P = [0 0 1 1; 0 1 0 1]

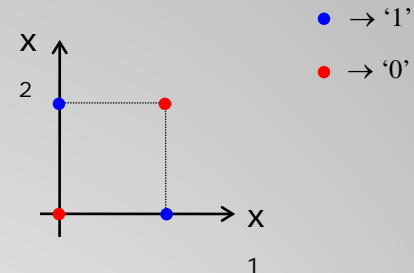
P =

    0     0     1     1
    0     1     0     1
```

linhas = # de variáveis
colunas = # de padrões

Definição do problema

O problema do OU-Exclusivo



X ₁	X ₂	valor
0	0	0
0	1	1
1	0	1
1	1	0

Passos p/ a criação de uma RN

```
Command Window
File Edit Debug Desktop Window Help

>> % Padrões de entrada:
>> P = [0 0 1 1; 0 1 0 1]

P =
    0     0     1     1
    0     1     0     1

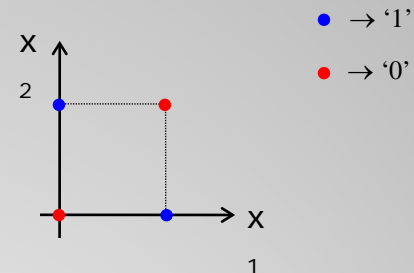
>> % Padrões de saída (alvos ou "targets"):
>> T = [0 1 1 0]

T =
    0     1     1     0

fx >> |
```

Definição do problema

O problema do OU-Exclusivo



X ₁	X ₂	valor
0	0	0
0	1	1
1	0	1
1	1	0

Passos p/ a criação de uma RN

- Definição do problema
- Definição dos padrões
- **Inicialização da rede**
- Definição dos parâmetros de treinamento
- Treinamento da rede
- Teste da rede

Passos p/ a criação de uma RN


```
Command Window
File Edit Debug Desktop Window Help

>> % Padrões de entrada:
>> P = [0 0 1 1; 0 1 0 1]

P =

     0     0     1     1
     0     1     0     1

>> % Padrões de saída (alvos ou "targets"):
>> T = [0 1 1 0]

T =

     0     1     1     0

>> % PERCEPTRON
>>
>> % opção 1:
>> % net1 = newff(P, T, 0);
>>
>> % opção 2:
>> % net1 = newff(P, T);
>>
>> % opção 3:
>> net = newp(P, T);
fx >>
```

Inicialização da rede

Perceptron:

```
net =
newp( P,      padrões entrada
      T      targets );
```

ou

```
net =
newff( P,      padrões entrada
      T      targets );
```

ou

```
net =
newff( P,      padrões entrada
      T,      targets
      N_hid   camadas escondidas );
```

ou

```
net =
feedforwardnet(P,T,[C1 C2],{FC1 FC2});
```

Passos p/ a criação de uma RN

Inicialização da rede

Perceptron:

```
net =  
newp( P,      padrões entrada  
      T      targets );
```

ou

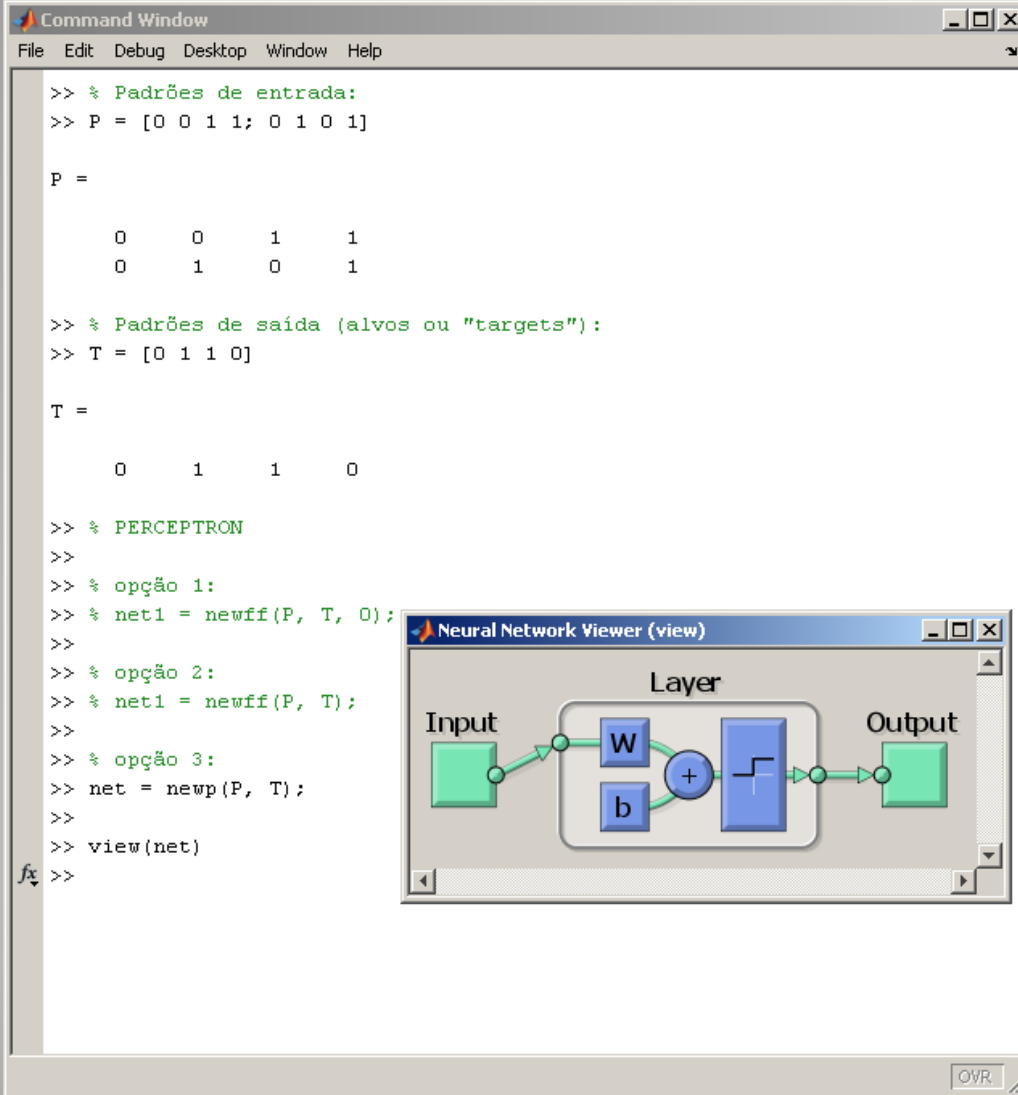
```
net =  
newff( P,      padrões entrada  
      T      targets );
```

ou

```
net =  
newff( P,      padrões entrada  
      T,      targets  
      N_hid   camadas escondidas );
```

ou

```
net =  
feedforwardnet(P,T,[C1 C2],{FC1 FC2});
```



The screenshot shows a MATLAB Command Window with the following code and output:

```
>> % Padrões de entrada:  
>> P = [0 0 1 1; 0 1 0 1]  
  
P =  
  
    0    0    1    1  
    0    1    0    1  
  
>> % Padrões de saída (alvos ou "targets"):  
>> T = [0 1 1 0]  
  
T =  
  
    0    1    1    0  
  
>> % PERCEPTRON  
>>  
>> % opção 1:  
>> % net1 = newff(P, T, 0);  
>>  
>> % opção 2:  
>> % net1 = newff(P, T);  
>>  
>> % opção 3:  
>> net = newp(P, T);  
>>  
>> view(net)  
>>
```

The Neural Network Viewer (view) window shows a diagram of a single-layer perceptron. It consists of an 'Input' block connected to a 'Layer' block, which is then connected to an 'Output' block. Inside the 'Layer' block, there are two input paths: one labeled 'W' (weights) and another labeled 'b' (bias). These paths lead to a summation node (a circle with a '+'). The output of the summation node passes through a transfer function block (a blue box with a step function icon). The final output is shown in the 'Output' block.

Passos p/ a criação de uma RN

```
Command Window
File Edit Debug Desktop Window Help

>> % Padrões de entrada:
>> P = [0 0 1 1; 0 1 0 1]

P =

     0     0     1     1
     0     1     0     1

>> % Padrões de saída (alvos ou "targets"):
>> T = [0 1 1 0]

T =

     0     1     1     0

>> % PERCEPTRON
>>
>> % opção 1:
>> % net1 = newff(P, T, 0);
>>
>> % opção 2:
>> % net1 = newff(P, T);
>>
>> % opção 3:
>> net = newp(P, T);
fx >>
```

Inicialização da rede

MLP:

```
net =
newff( P,           padrões entrada
       T,           targets
       {H1..Hn-1}, camadas escondidas
       {TF1..TFn}, func. transf.
       BTF,         alg. treinamento
       BLF           alg. aprendizado);
```

Passos p/ a criação de uma RN



```

Command Window
File Edit Debug Desktop Window Help

>> % Padrões de entrada:
>> P = [0 0 1 1; 0 1 0 1]

P =

     0     0     1     1
     0     1     0     1

>> % Padrões de saída (alvos ou "targets"):
>> T = [0 1 1 0]

T =

     0     1     1     0

>> % PERCEPTRON
>>
>> % opção 1:
>> % net1 = newff(P, T, 0);
>>
>> % opção 2:
>> % net1 = newff(P, T);
>>
>> % opção 3:
>> net = newp(P, T);
fx >>

```

Inicialização da rede

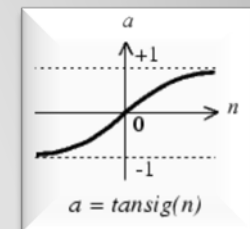
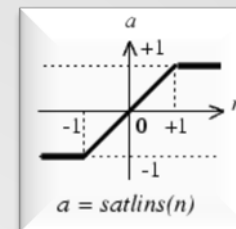
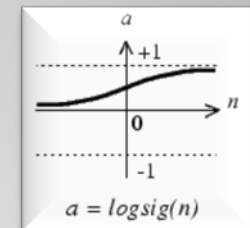
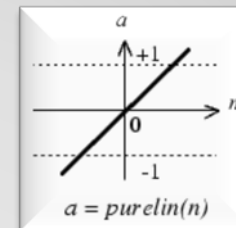
MLP:

```

net =
newff( P,           padrões entrada
       T,           targets
       {H1..Hn-1}, camadas escondidas
       {TF1..TFn}, func. transf.
       BTF,         alg. treinamento
       BLF,         alg. aprendizado);

```

Funções de Transferência:



Passos p/ a criação de uma RN



```

Command Window
File Edit Debug Desktop Window Help

>> % Padrões de entrada:
>> P = [0 0 1 1; 0 1 0 1]

P =

     0     0     1     1
     0     1     0     1

>> % Padrões de saída (alvos ou "targets"):
>> T = [0 1 1 0]

T =

     0     1     1     0

>> % PERCEPTRON
>>
>> % opção 1:
>> % net1 = newff(P, T, 0);
>>
>> % opção 2:
>> % net1 = newff(P, T);
>>
>> % opção 3:
>> net = newp(P, T);
fx >>

```

Inicialização da rede

MLP:

```

net =
newff( P,           padrões entrada
       T,           targets
       [H1..Hn-1], camadas escondidas
       {TF1..TFn}, func. transf.
       BTF,         alg. treinamento
       BLF           alg. aprendizado);

```

Algoritmos de Treinamento:

- **traingd** Gradient descent backpropagation
- **traingdm** Gradient descent backpropagation com momentum
- **trainгда** 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 p/ a criação de uma RN



```
Command Window
File Edit Debug Desktop Window Help

>> % Padrões de entrada:
>> P = [0 0 1 1; 0 1 0 1]

P =

     0     0     1     1
     0     1     0     1

>> % Padrões de saída (alvos ou "targets"):
>> T = [0 1 1 0]

T =

     0     1     1     0

>> % PERCEPTRON
>>
>> % opção 1:
>> % net1 = newff(P, T, 0);
>>
>> % opção 2:
>> % net1 = newff(P, T);
>>
>> % opção 3:
>> net = newp(P, T);
fx >>
```

Inicialização da rede

MLP:

```
net =
newff( P,          padrões entrada
       T,          targets
       {H1..Hn-1}, camadas escondidas
       {TF1..TFn}, func. transf.
       BTF,        alg. treinamento
       BLF          alg. aprendizado);
```

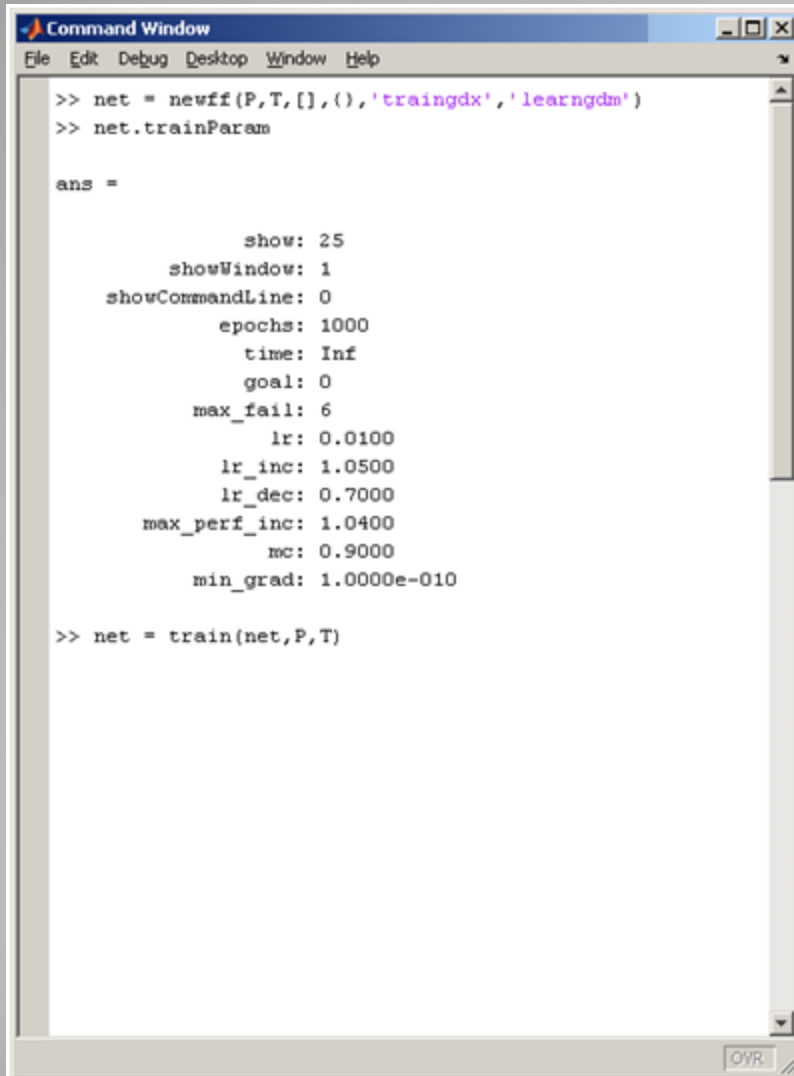
Algoritmos de Aprendizado:

- *learn*gd Gradient descent weight and bias learning function
- *learn*gdm Gradient descent with momentum weight and bias learning function

Passos p/ a criação de uma RN

- Definição do problema
- Definição dos padrões
- Inicialização da rede
- Definição dos parâmetros de treinamento
- Treinamento da rede
- Teste da rede

Passos p/ a criação de uma RN



```
>> net = newff(P,T,[],(),'traingdx','learngdm')
>> net.trainParam

ans =

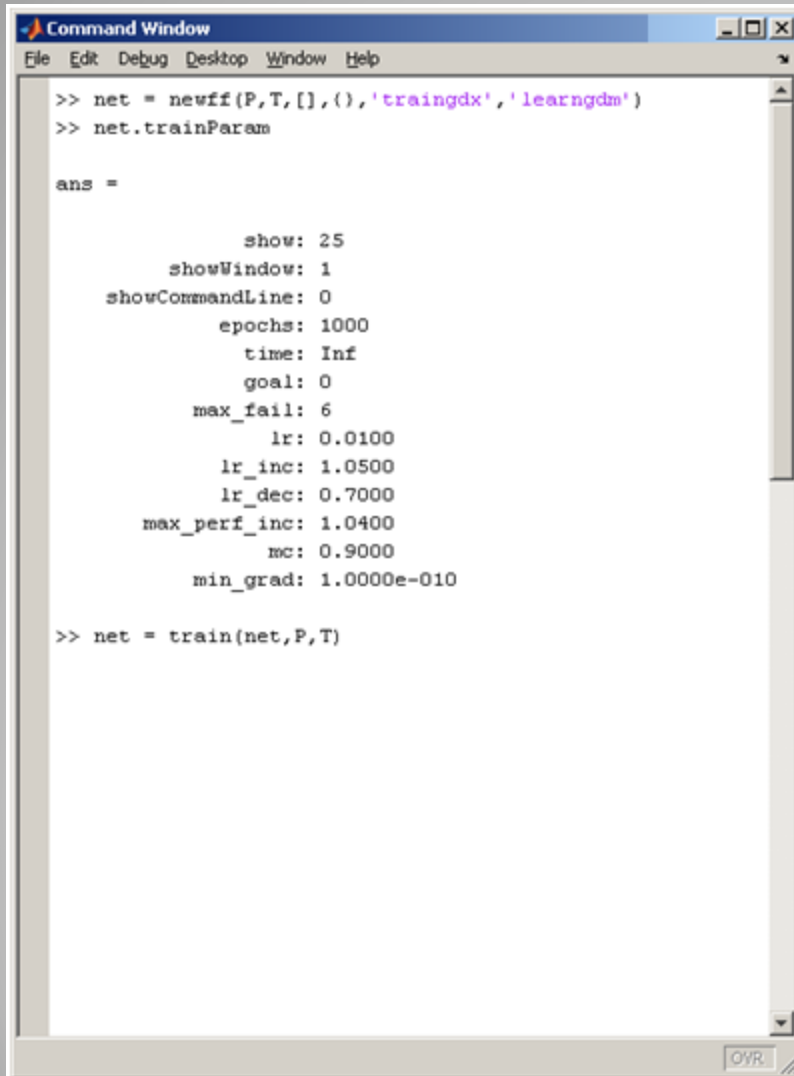
    show: 25
  showWindow: 1
showCommandLine: 0
    epochs: 1000
      time: Inf
       goal: 0
   max_fail: 6
        lr: 0.0100
     lr_inc: 1.0500
     lr_dec: 0.7000
max_perf_inc: 1.0400
         mc: 0.9000
   min_grad: 1.0000e-010

>> net = train(net,P,T)
```

Parâmetros de Treinamento

- net.trainParam.epochs -----> Número de epochs
- net.trainParam.goal -----> Erro final desejado
- net.trainParam.lr -----> Taxa de aprendizado
- net.trainParam.show -----> Atualização da tela (epochs)
- net.trainParam.mc -----> Taxa de momentum
- net.trainParam.lr_inc -----> Taxa de incremento da l.r.
- net.trainParam.lr_dec -----> Taxa de decremento da l.r.
- net.trainParam.max_perf_inc --> Incremento máximo do erro

Passos p/ a criação de uma RN



```
>> net = newff(P,T,[],(),'traingdx','learngdm')
>> net.trainParam

ans =

        show: 25
    showWindow: 1
showCommandLine: 0
        epochs: 1000
           time: Inf
          goal: 0
       max_fail: 6
           lr: 0.0100
        lr_inc: 1.0500
        lr_dec: 0.7000
    max_perf_inc: 1.0400
           mc: 0.9000
    min_grad: 1.0000e-010

>> net = train(net,P,T)
```

Parâmetros de Treinamento (early stopping)

Divide vectors into three sets using specified indices.

```
net.divideFcn = 'divideind';
net.divideParam.trainInd = indTreino;
net.divideParam.valInd = indValidacao;
net.divideParam.testInd = indTeste;
```

Divide vectors into three sets using random indices.

```
net.divideFcn = 'dividerand';
net.divideParam.trainRatio: 0.6000;
net.divideParam.valRatio: 0.2000;
net.divideParam.testRatio: 0.2000;
```

Divide vectors into three sets using blocks of indices.

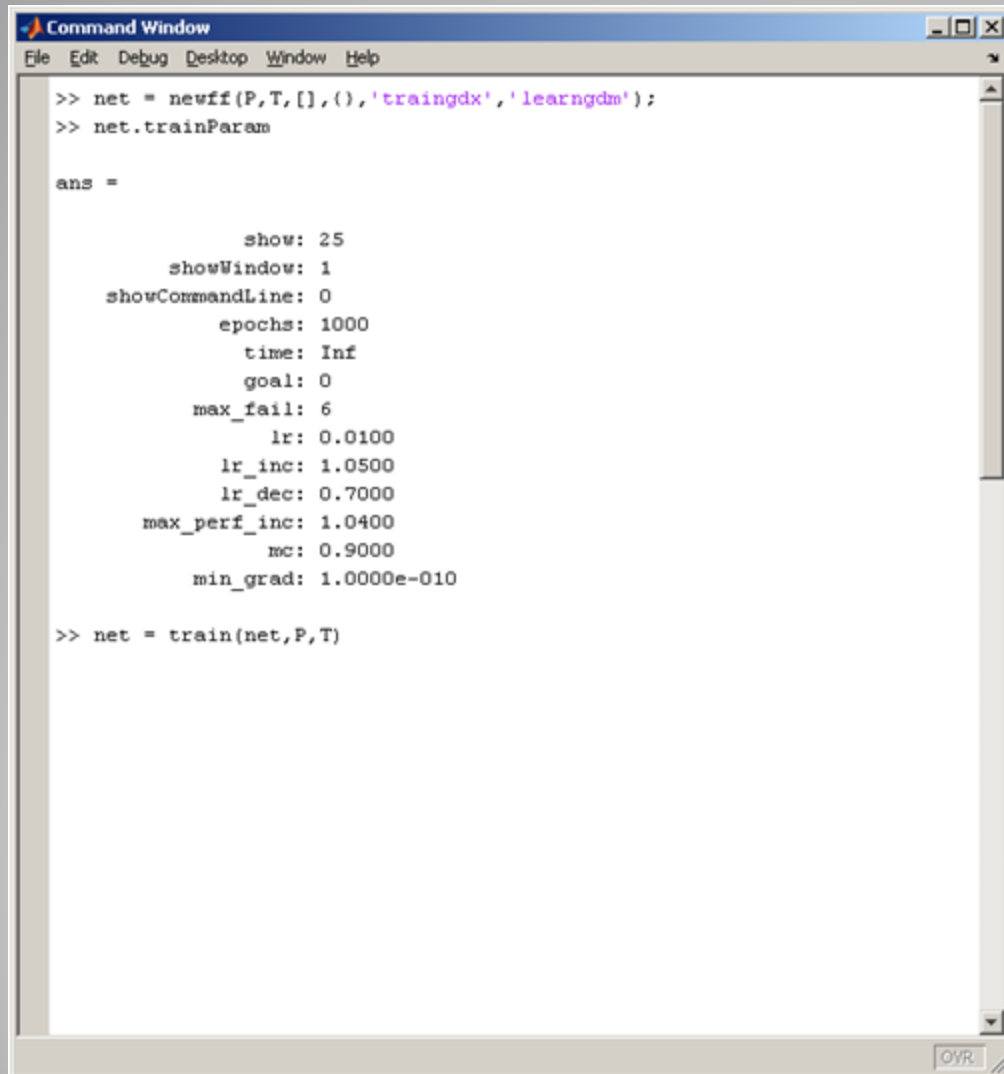
```
net.divideFcn = 'divideblock';
net.divideParam.trainRatio: 0.6000;
net.divideParam.valRatio: 0.2000;
net.divideParam.testRatio: 0.2000;
```

```
net.divideFcn = '';
```

Passos p/ a criação de uma RN

- Definição do problema
- Definição dos padrões
- Inicialização da rede
- Definição dos parâmetros de treinamento
- Treinamento da rede
- Teste da rede

Passos p/ a criação de uma RN



```
Command Window
File Edit Debug Desktop Window Help

>> net = newff(P,T,[],(),'traingdx','learngdm');
>> net.trainParam

ans =

        show: 25
    showWindow: 1
showCommandLine: 0
        epochs: 1000
           time: Inf
          goal: 0
       max_fail: 6
            lr: 0.0100
         lr_inc: 1.0500
         lr_dec: 0.7000
    max_perf_inc: 1.0400
             mc: 0.9000
       min_grad: 1.0000e-010

>> net = train(net,P,T)
```

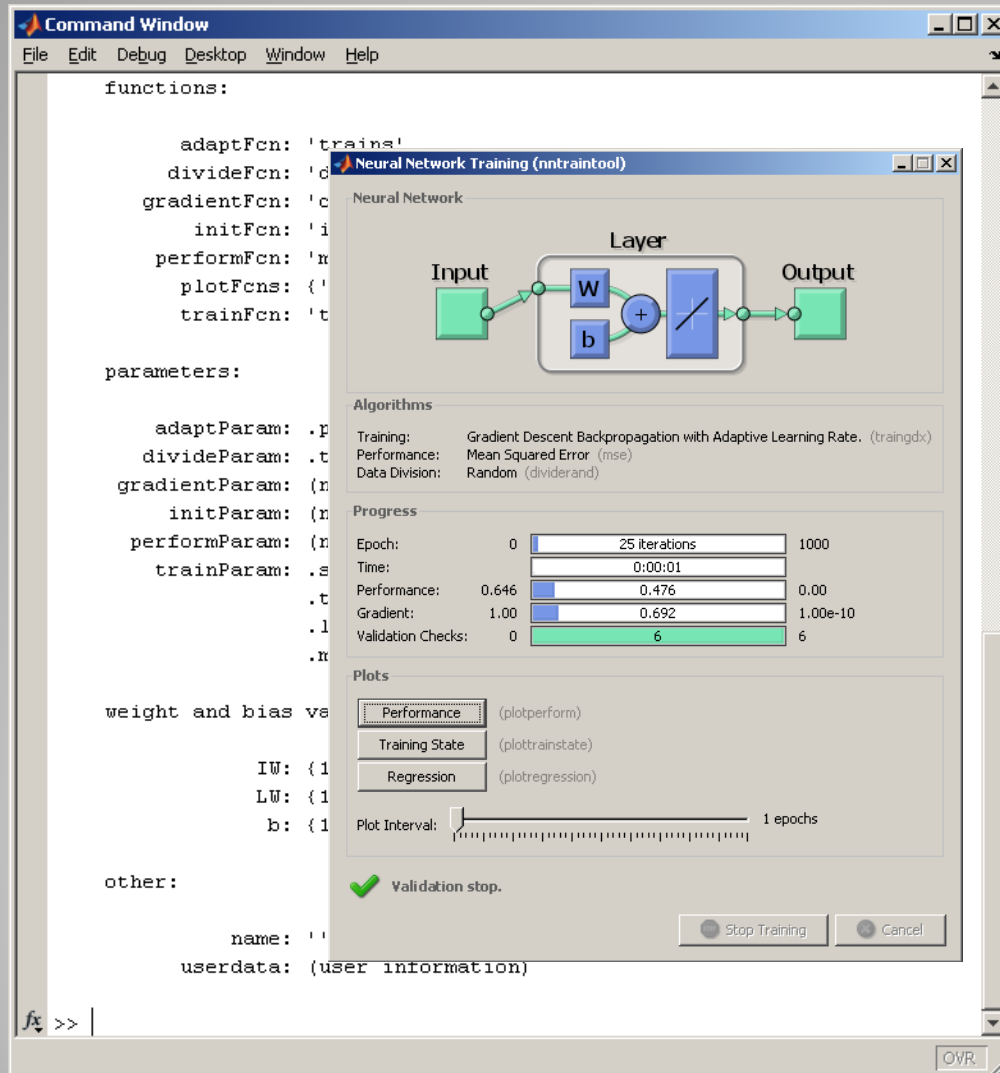
Treinando a rede

```
net = train(net, P, T);
```

Passos p/ a criação de uma RN

Treinando a rede

```
net = train(net, P, T);
```



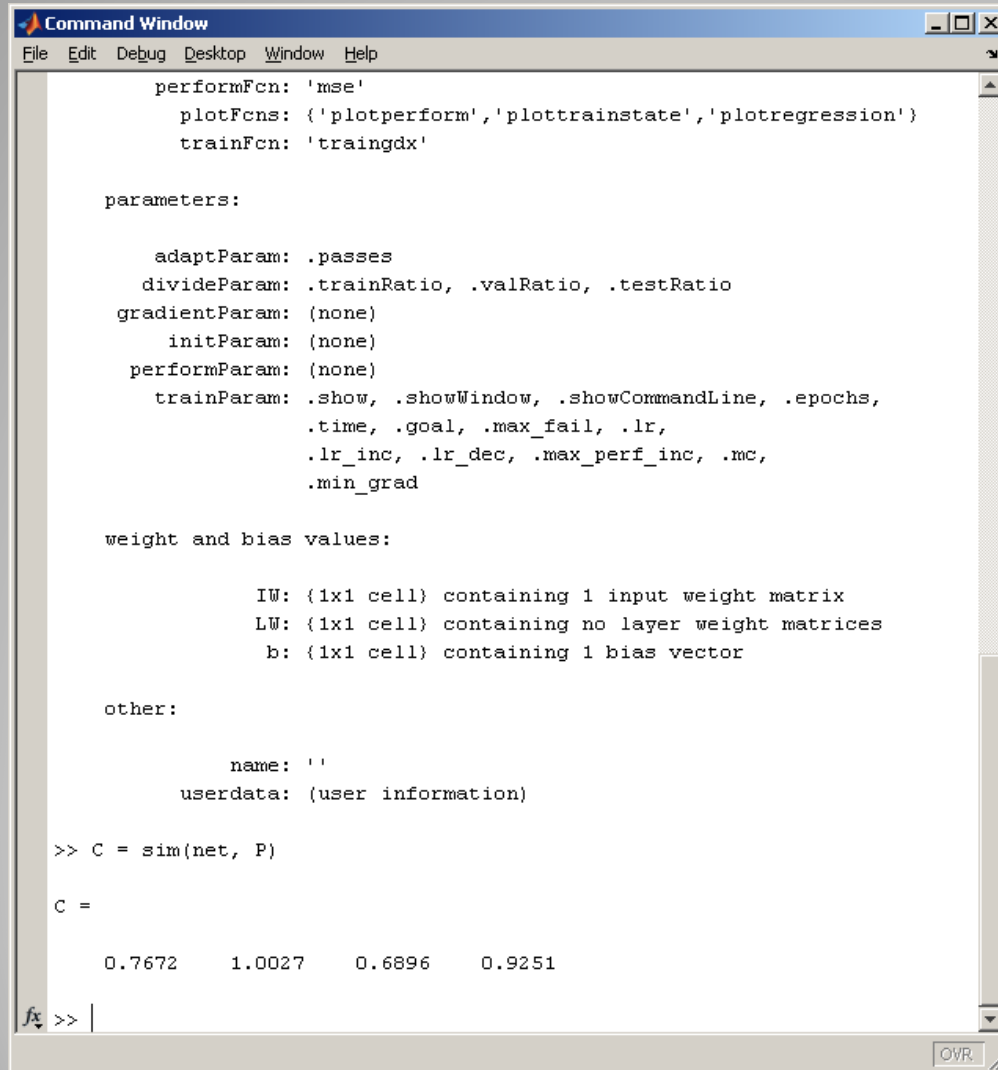
Passos p/ a criação de uma RN

- Definição do problema
- Definição dos padrões
- Inicialização da rede
- Definição dos parâmetros de treinamento
- Treinamento da rede
- Teste da rede

Passos p/ a criação de uma RN

Testando a rede

```
C = sim(net, P);
```



The image shows a MATLAB Command Window with a blue title bar and menu bar (File, Edit, Debug, Desktop, Window, Help). The window contains the following text:

```
performFcn: 'mse'
plotFcns: {'plotperform','plottrainstate','plotregression'}
trainFcn: 'traingdx'

parameters:

    adaptParam: .passes
    divideParam: .trainRatio, .valRatio, .testRatio
    gradientParam: (none)
    initParam: (none)
    performParam: (none)
    trainParam: .show, .showWindow, .showCommandLine, .epochs,
                .time, .goal, .max_fail, .lr,
                .lr_inc, .lr_dec, .max_perf_inc, .mc,
                .min_grad

weight and bias values:

    IW: {1x1 cell} containing 1 input weight matrix
    LW: {1x1 cell} containing no layer weight matrices
    b: {1x1 cell} containing 1 bias vector

other:

    name: ''
    userdata: (user information)

>> C = sim(net, P)

C =

    0.7672    1.0027    0.6896    0.9251

fx >> |
```

At the bottom right of the window, there is a small button labeled 'OVR'.

Passos p/ a criação de uma RN

```
Editor - G:\xor scripts\xor1.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base fx

1 P = [0 0 1 1; 0 1 0 1];
2 T = [0 1 1 0];
3
4 net = newff(P,T,[],(),'traingd');
5
6 net.trainParam.epochs = 1000;
7 net.trainParam.goal = 1e-3;
8 net.trainParam.lr = 0.01;
9 net.trainParam.show = 25;
10
11 net = train(net, P, T);
12
13 C = sim(net, P);
14
15 figure
16 subplot(1,2,1)
17 hold on
18 for i = 1:4,
19     if (T(i) > 0.5)
20         plot(P(1,i), P(2,i), 'b.')
```

```
21     else
22         plot(P(1,i), P(2,i), 'r.')
```

```
23     end
24 end
25 xlim([-1 2])
26 ylim([-1 2])
27 axis square
28 set(gca, 'xtick', [0 1])
29 set(gca, 'ytick', [0 1])
30 box on
31 title('Desejado')
```

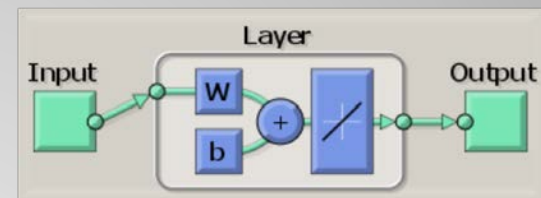
```
32
33 subplot(1,2,2)
34 hold on
35 for i = 1:4,
36     if (C(i) > 0.5)
37         plot(P(1,i), P(2,i), 'b.')
```

```
38     else
39         plot(P(1,i), P(2,i), 'r.')
```

```
40     end
41 end
42 xlim([-1 2])
43 ylim([-1 2])
44 axis square
45 set(gca, 'xtick', [0 1])
46 set(gca, 'ytick', [0 1])
47 box on
48 title('Obtido')
```

Perceptron básico

TESTE 1: Rede neural com apenas uma camada de processadores



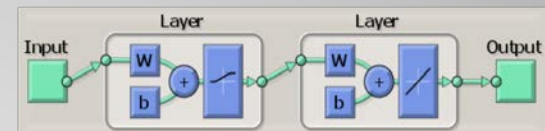
xor1.m

M-file desenvolvida para o XOR

```
Editor - G:\xor scripts\xor2.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1 P = [0 0 1 1; 0 1 0 1];
2 T = [0 1 1 0];
3
4 net = newff(P, T, [2], {'logsig'}, 'traingd');
5
6 net.trainParam.epochs = 5000;
7 net.trainParam.goal = 1e-3;
8 net.trainParam.lr = 0.01;
9 net.trainParam.show = 25;
10
11 net = train(net, P, T);
12
13 C = sim(net, P)
14
15 figure
16 subplot(1,2,1)
17 hold on
18 for i = 1:4,
19     if (T(i) > 0.5)
20         plot(P(1,i), P(2,i), 'b.')
21     else
22         plot(P(1,i), P(2,i), 'r.')
23     end
24 end
25 xlim([-1 2])
26 ylim([-1 2])
27 axis square
28 set(gca, 'xtick', [0 1])
29 set(gca, 'ytick', [0 1])
30 box on
31 title('Desejado')
32
33 subplot(1,2,2)
34 hold on
35 for i = 1:4,
36     if (C(i) > 0.5)
37         plot(P(1,i), P(2,i), 'b.')
38     else
39         plot(P(1,i), P(2,i), 'r.')
40     end
41 end
42 xlim([-1 2])
43 ylim([-1 2])
44 axis square
45 set(gca, 'xtick', [0 1])
46 set(gca, 'ytick', [0 1])
47 box on
48 title('Obtido')
```

Multi-layer Perceptron

TESTE 2: Rede neural com uma camada escondida de processadores (Multi-Layer Perceptron);



**TAXA de APRENDIZADO CONSTANTE,
sem MOMENTUM**



xor2.m

M-file desenvolvida para o XOR


```
Editor - G:\xor scripts\xor3.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base fx

1 P = [0 0 1 1; 0 1 0 1];
2 T = [0 1 1 0];
3
4 net = newff(P, T, 2, {'logsig'}, 'trainingdx');
5
6 net.trainParam.epochs = 5000;
7 net.trainParam.goal = 1e-3;
8 net.trainParam.lr = 0.01;
9 net.trainParam.show = 25;
10
11 net.trainParam.mc = 0.9;
12
13 net.trainParam.lr_inc = 1.05;
14 net.trainParam.lr_dec = 0.7;
15 net.trainParam.max_perf_inc = 1.04;
16
17 net = train(net, P, T);
18
19 C = sim(net, P)
20
21 figure
22 subplot(1,2,1)
23 hold on
24 for i = 1:4,
25     if (T(i) > 0.5)
26         plot(P(1,i), P(2,i), 'b.')
```

Multi-layer Perceptron

TESTE 3: Rede neural com uma camada escondida de processadores (Multi-Layer Perceptron);



TAXA ADAPTATIVA e MOMENTUM

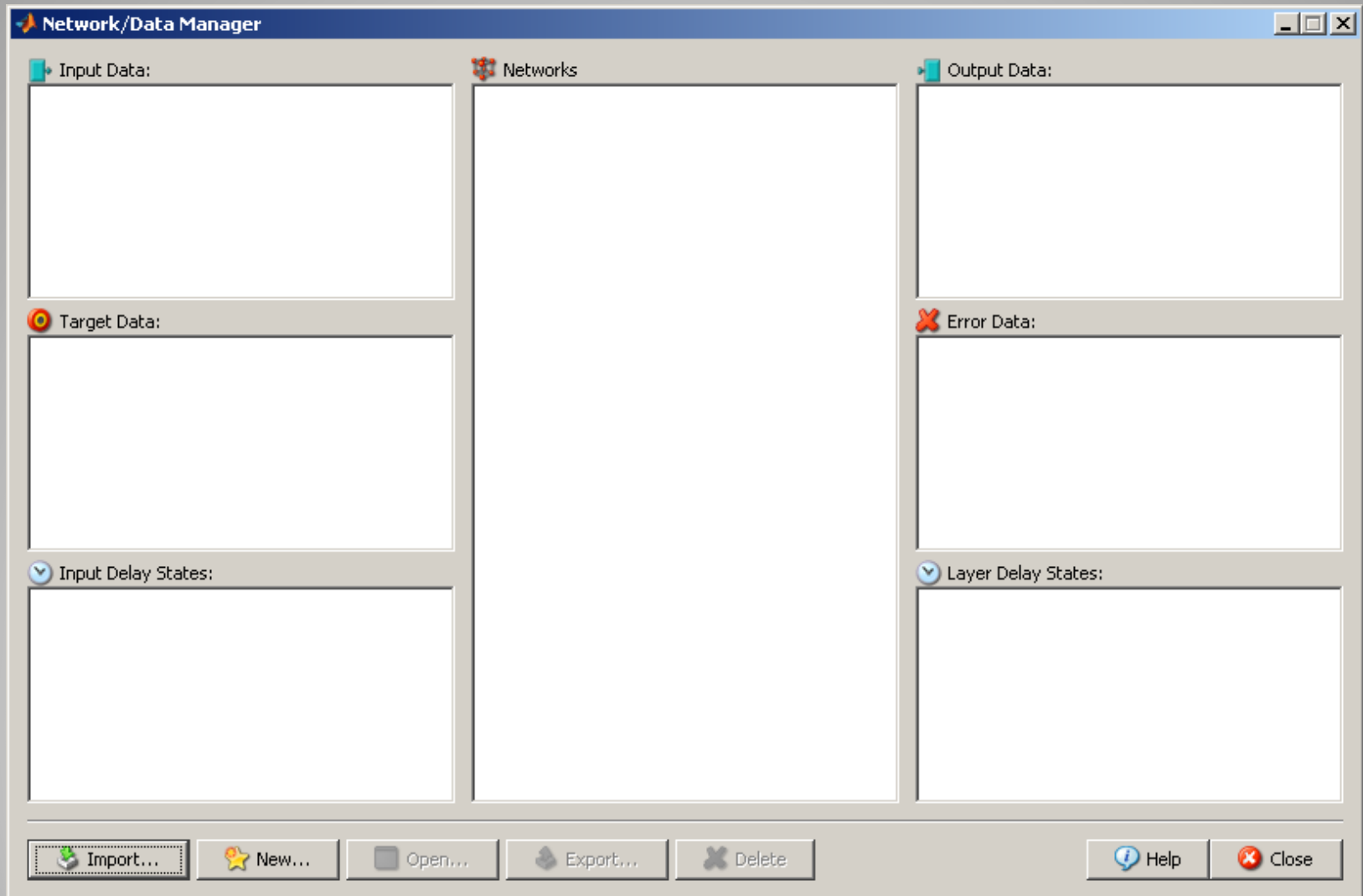


xor3.m

M-file desenvolvida para o XOR

Toolbox de Redes Neurais

- Introdução ao Matlab
 - Linha de comando
 - NNTool
- Estudo de Casos
 - Análise de Crédito Bancário (Classificação).
 - Localização de Faltas (Inferência).
 - Comportamento do Clima (Previsão).



Interface gráfica NNTool

- Definição do problema
- Inicialização da rede
- Definição dos parâmetros de treinamento
- Treinamento da rede
- Teste da rede

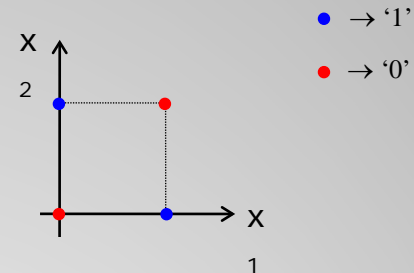
Passos p/ a criação de uma RN

- Definição do problema
- Inicialização da rede
- Definição dos parâmetros de treinamento
- Treinamento da rede
- Teste da rede

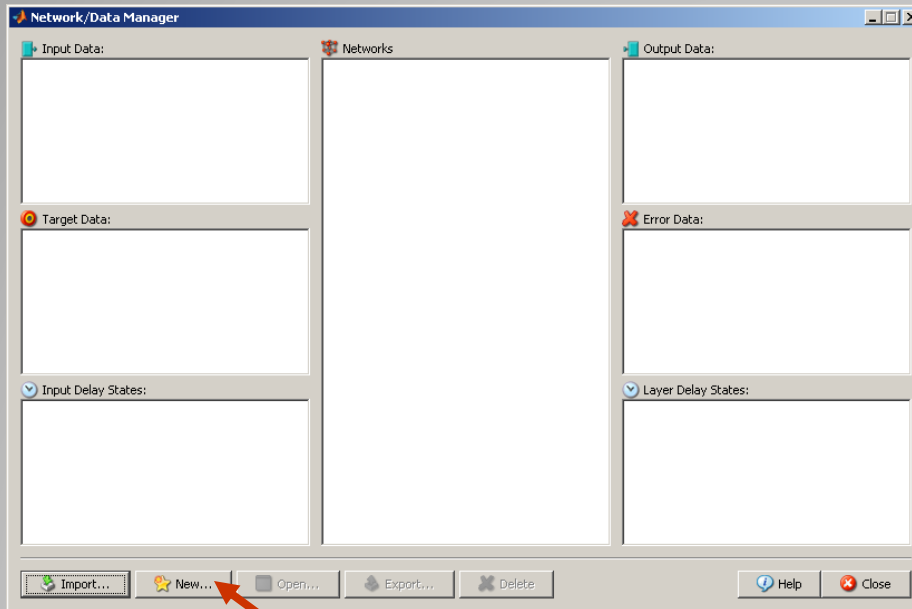
Passos p/ a criação de uma RN

Definição do problema

O problema do OU-Exclusivo



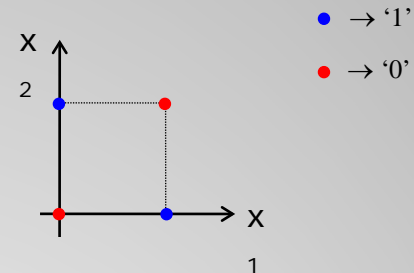
X_1	X_2	valor
0	0	0
0	1	1
1	0	1
1	1	0



Passos p/ a criação de uma RN

Definição do problema

O problema do OU-Exclusivo



X ₁	X ₂	valor
0	0	0
0	1	1
1	0	1
1	1	0

Create Network or Data

Network Data

Name: P

Value: [0 0 1 1; 0 1 0 1]

Data Type:

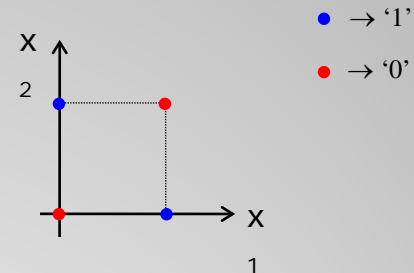
- ☒ Inputs
- ☐ Targets
- ☐ Input Delay States
- ☐ Layer Delay States
- ☐ Outputs
- ☐ Errors

Help Create Close

Passos p/ a criação de uma RN

Definição do problema

O problema do OU-Exclusivo



X ₁	X ₂	valor
0	0	0
0	1	1
1	0	1
1	1	0

Create Network or Data

Network Data

Name
[T]

Value
[0 1 1 0]

Data Type

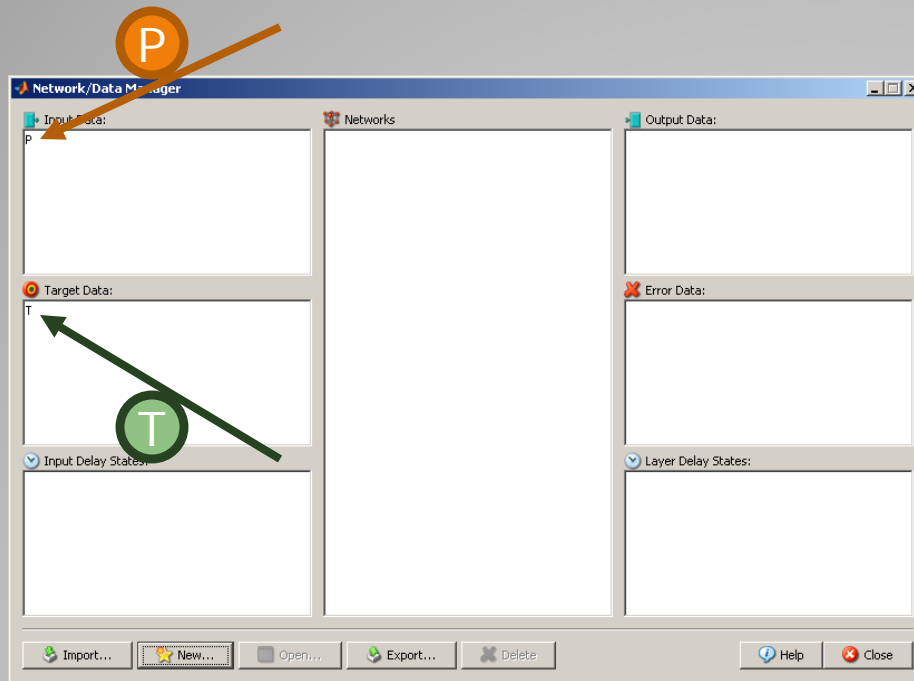
- ☐ Inputs
- ☒ Targets
- ☐ Input Delay States
- ☐ Layer Delay States
- ☐ Outputs
- ☐ Errors

Help Create Close

Diagram illustrating the steps for creating network data:

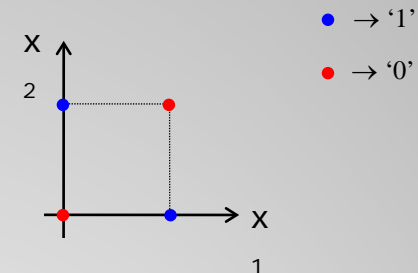
- Step 1: Click on the "Data" tab.
- Step 2: Select "Targets" under "Data Type".
- Step 3: Enter the target values [0 1 1 0] in the "Value" field.
- Step 4: Click the "Create" button.

Passos p/ a criação de uma RN



Definição do problema

O problema do OU-Exclusivo

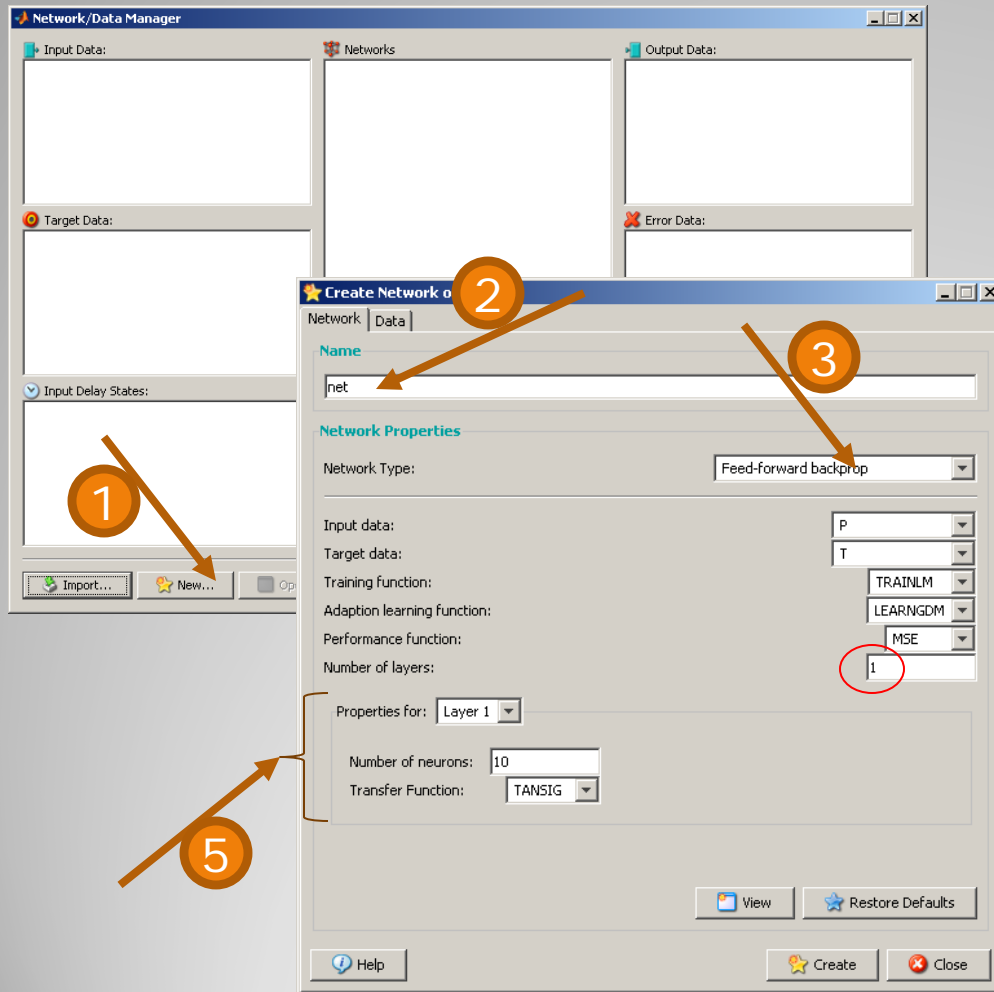
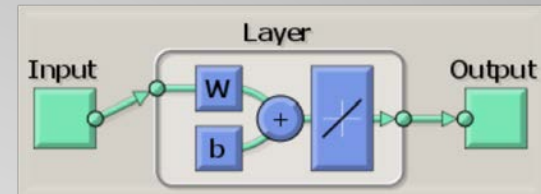


X_1	X_2	valor
0	0	0
0	1	1
1	0	1
1	1	0

Passos p/ a criação de uma RN

Criar a rede

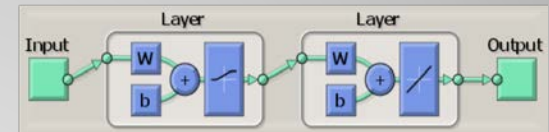
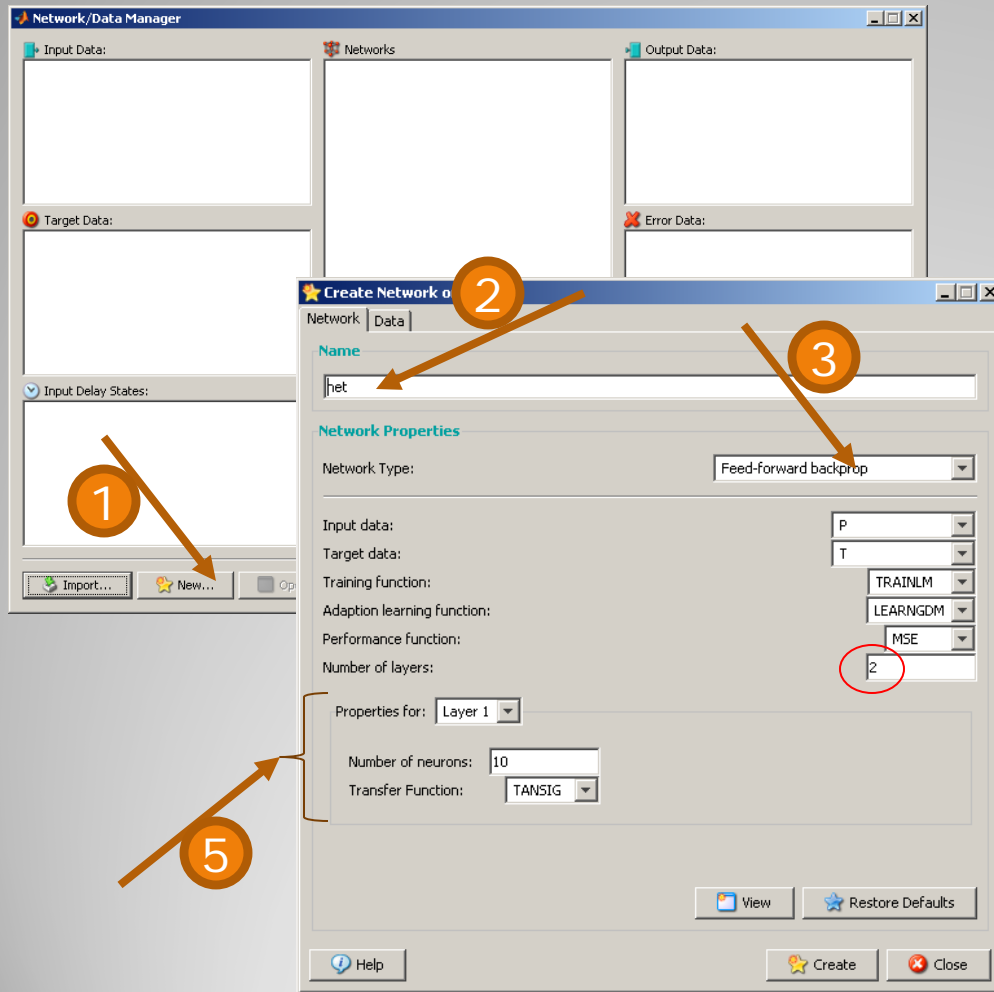
PERCEPTRON



Passos p/ a criação de uma RN

Criar a rede

Multi-layer PERCEPTRON

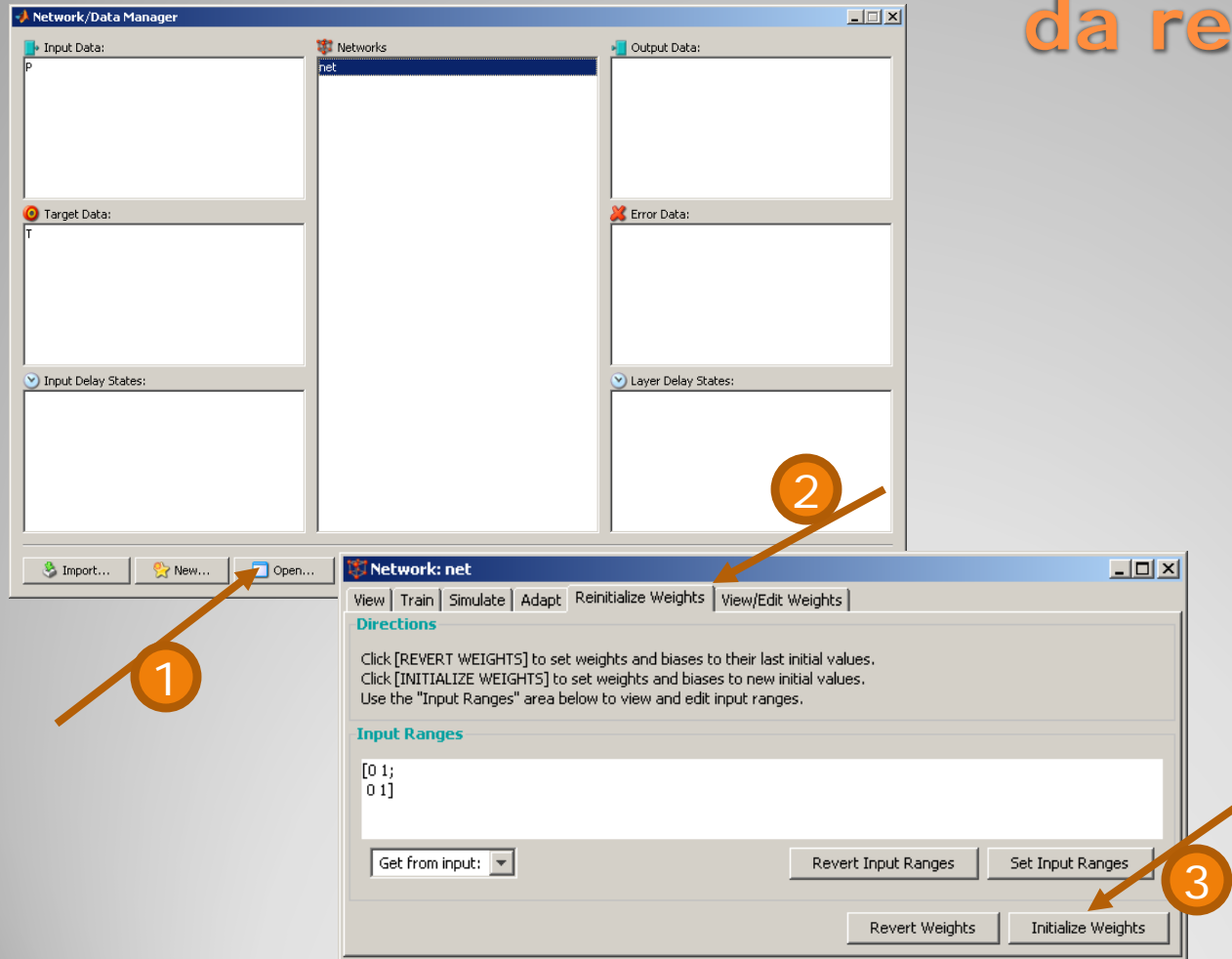


Passos p/ a criação de uma RN

- Definição do problema
- Inicialização da rede
- Definição dos parâmetros de treinamento
- Treinamento da rede
- Teste da rede

Passos p/ a criação de uma RN

Inicialização da rede

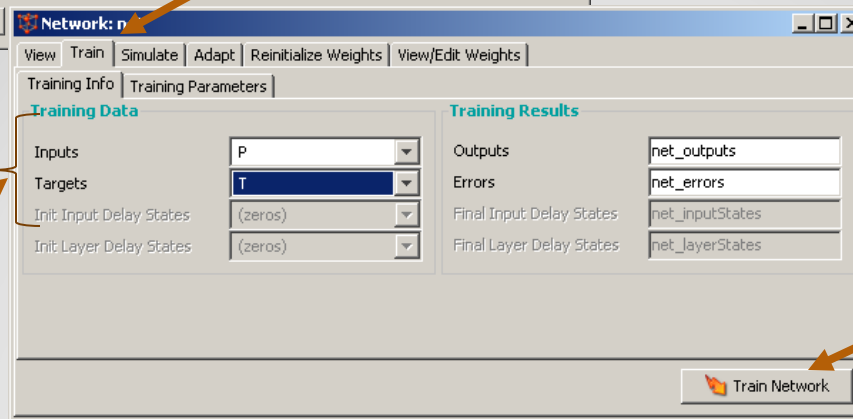
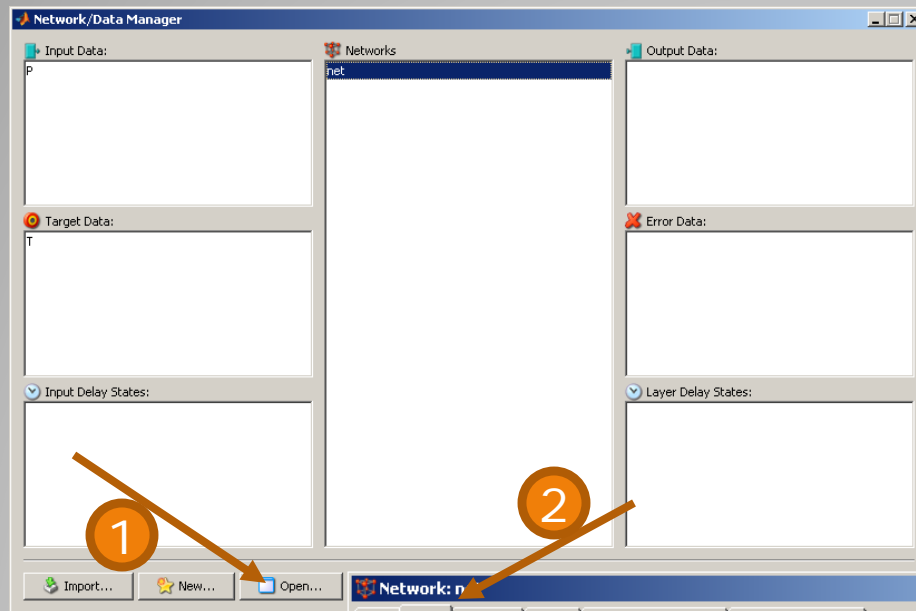


Passos p/ a criação de uma RN

- Definição do problema
- Inicialização da rede
- Definição dos parâmetros de treinamento
- Treinamento da rede
- Teste da rede

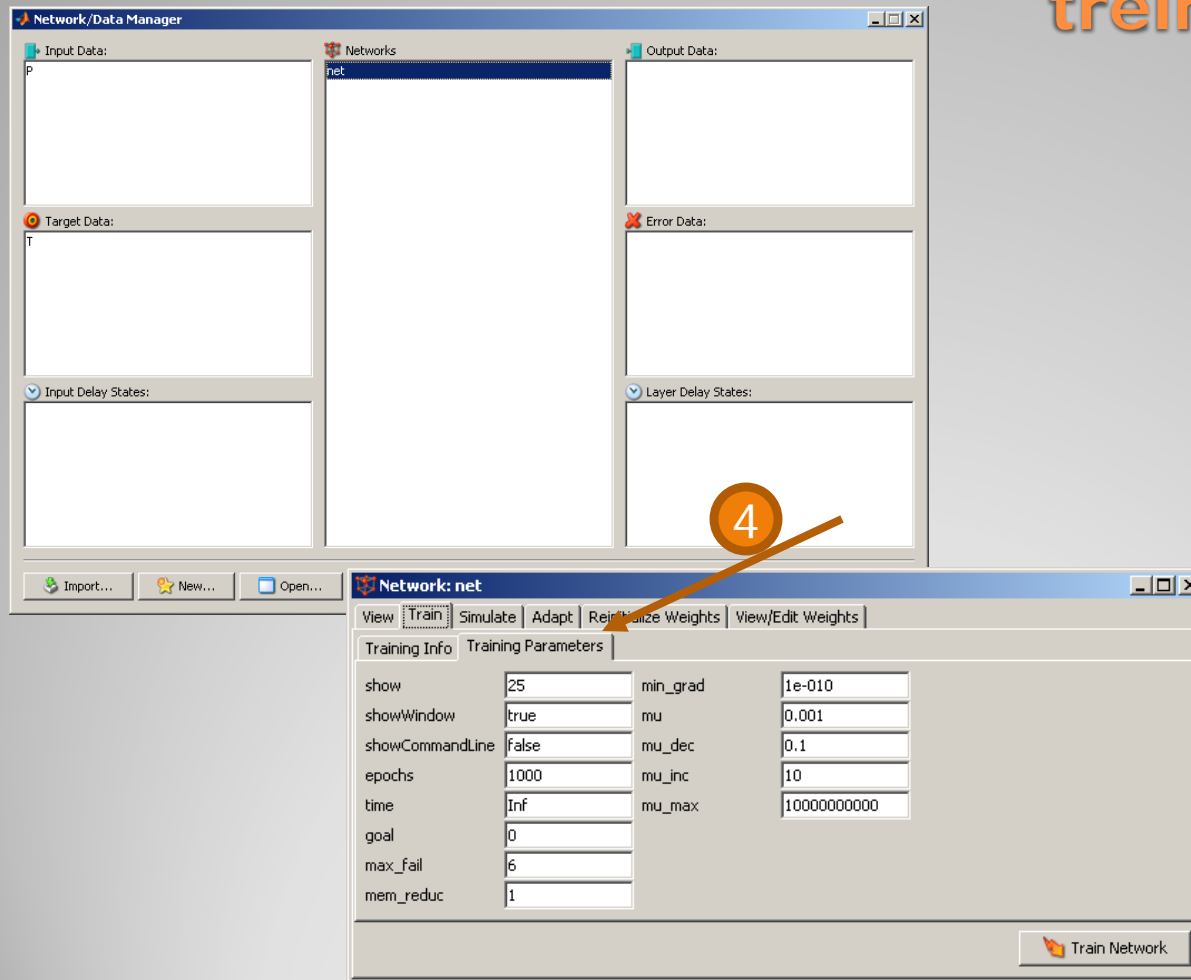
Passos p/ a criação de uma RN

Parâmetros de treinamento



Passos p/ a criação de uma RN

Parâmetros de treinamento



Passos p/ a criação de uma RN

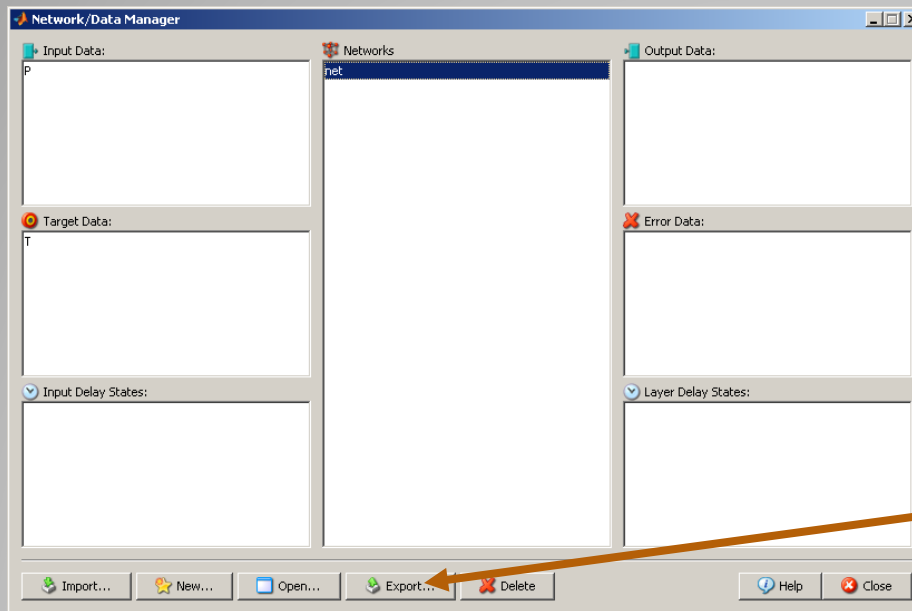
Parâmetros de treinamento (early stopping)

```
net.divideFcn = 'divideind';  
net.divideParam.trainInd = indTreino;  
net.divideParam.valInd = indValidacao;  
net.divideParam.testInd = indTeste;
```

```
net.divideFcn = 'dividerand';  
net.divideParam.trainRatio: 0.6000;  
net.divideParam.valRatio: 0.2000;  
net.divideParam.testRatio: 0.2000;
```

```
net.divideFcn = 'divideblock';  
net.divideParam.trainRatio: 0.6000;  
net.divideParam.valRatio: 0.2000;  
net.divideParam.testRatio: 0.2000;
```

```
net.divideFcn = '';
```

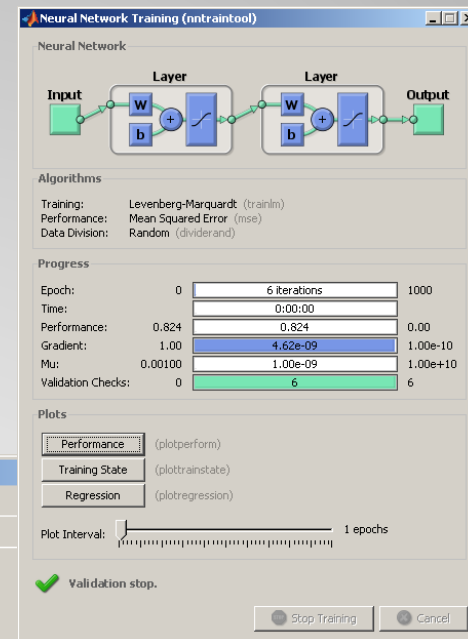
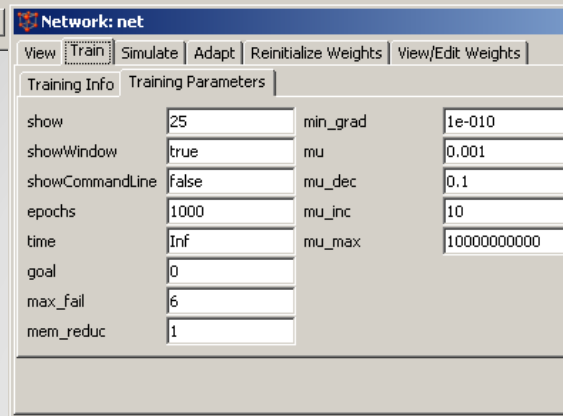
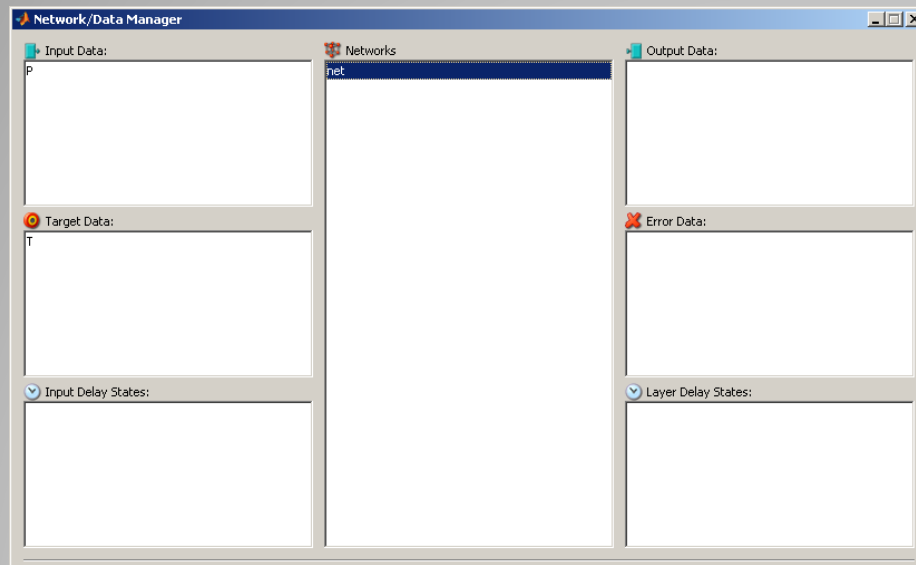


Passos p/ a criação de uma RN

- Definição do problema
- Inicialização da rede
- Definição dos parâmetros de treinamento
- Treinamento da rede
- Teste da rede

Passos p/ a criação de uma RN

Treinamento da rede



5

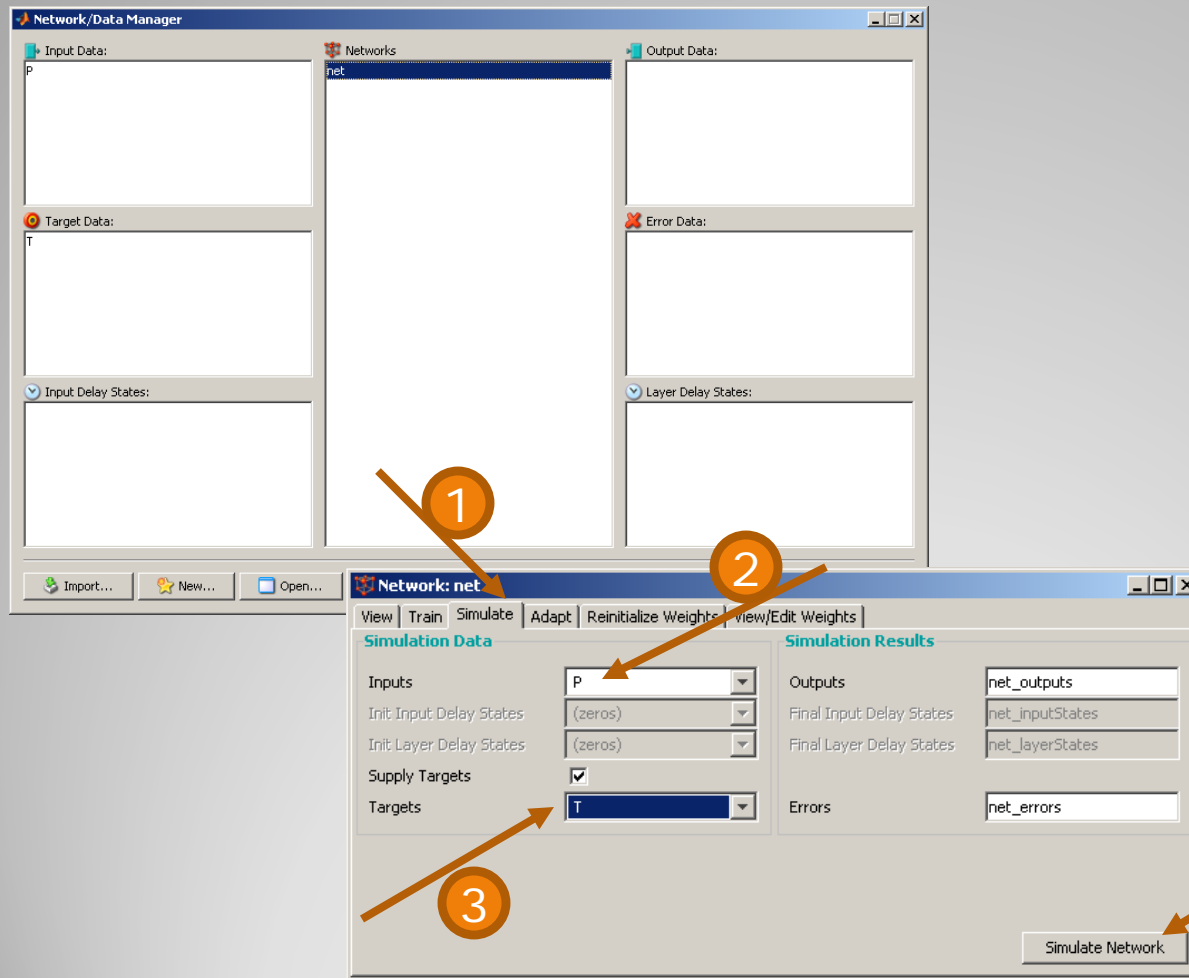
Train Network

Passos p/ a criação de uma RN

- Definição do problema
- Inicialização da rede
- Definição dos parâmetros de treinamento
- Treinamento da rede
- Teste da rede

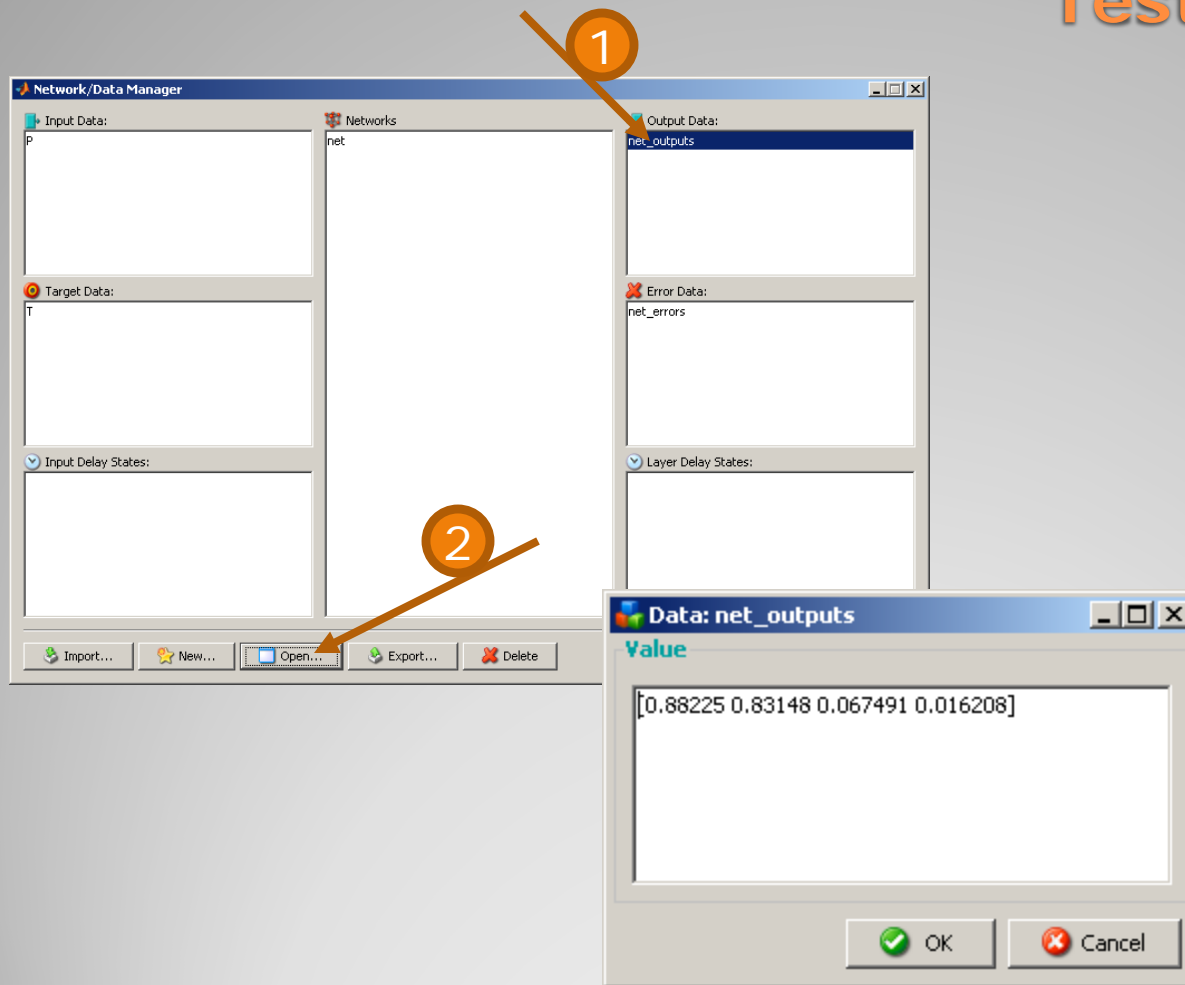
Passos p/ a criação de uma RN

Testando a rede



Passos p/ a criação de uma RN

Testando a rede



Passos p/ a criação de uma RN

Toolbox de Redes Neurais

- Introdução ao Matlab
 - Linha de comando
 - NNTool
- Estudo de Casos
 - Análise de Crédito Bancário (Classificação).
 - Localização de Faltas (Inferência).
 - Sensação Térmica (Previsão).

Análise de Crédito Bancário

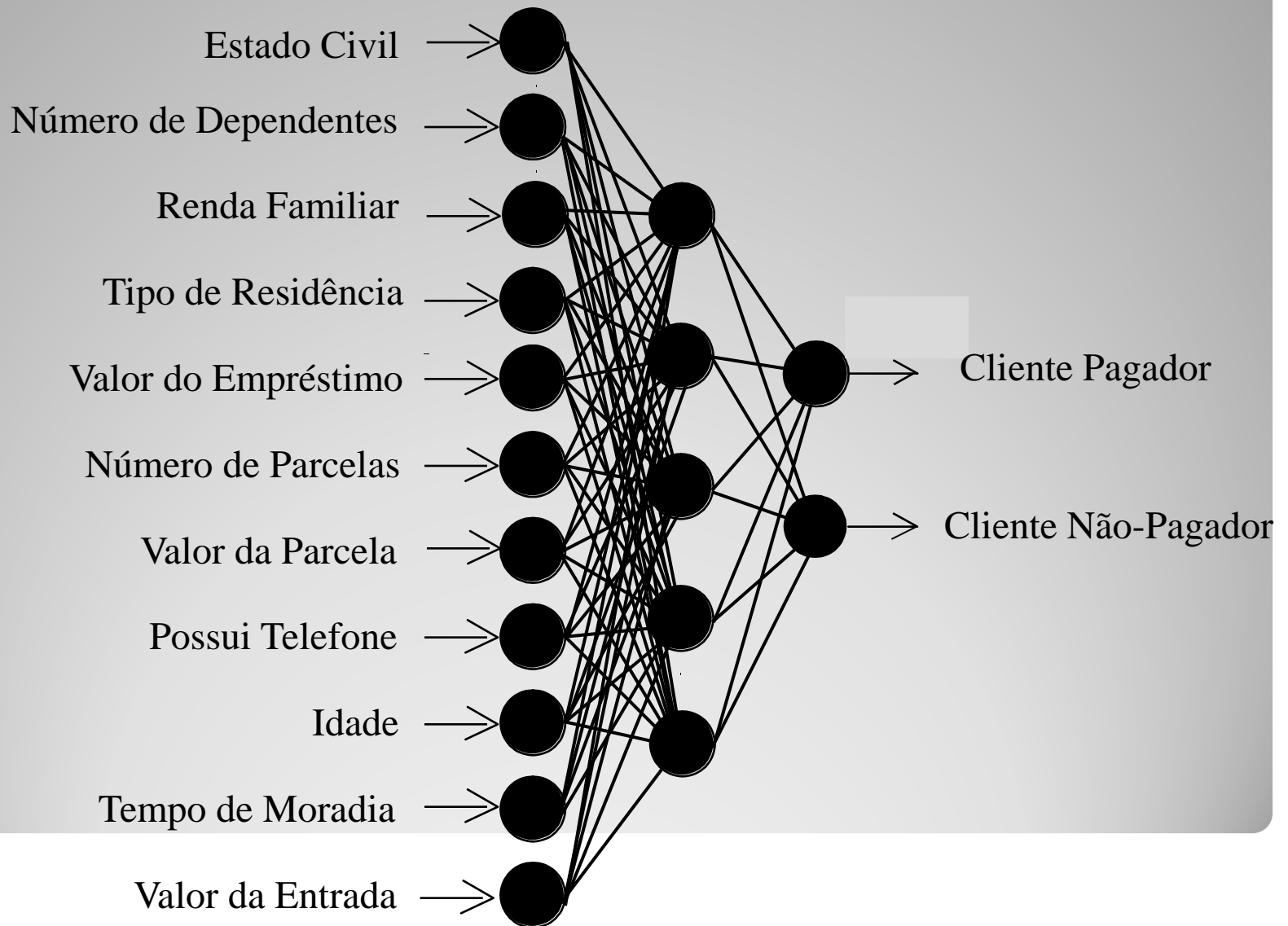
- **Base de Dados:** contém informações sobre 1500 clientes:
 - 715 pagadores; 785 não pagadores.
- **Atributos da Base de Dados:**
 - ESTC ⇒ Estado civil
 - NDEP ⇒ Número de dependentes
 - RENDA ⇒ Renda familiar
 - TIPOR ⇒ Tipo de Residência (0=alugada, 1=própria, 2=parentes)
 - VBEM ⇒ O valor do bem a ser adquirido
 - NPARC ⇒ O número de parcelas a serem pagas
 - VPARC ⇒ O valor de cada parcelas
 - TEL ⇒ Possui telefone (0=não e 1=sim)
 - IDADE ⇒ Quantos anos a pessoa tem
 - RESMS ⇒ Tempo de moradia (em mês)
 - ENTRADA ⇒ o valor da parcela paga no dia do empréstimo
- **Saída:**
 - CLASSE_1 ⇒ O cliente não pagou o empréstimo
 - CLASSE_2 ⇒ O cliente pagou o empréstimo

Análise de Crédito Bancário

- Amostra da Base de Dados:

ESTC	NDEP	RENDA	TIPOR	VBEM	NPARC	VPARC	TEL	IDADE	RESMS	ENTRADA	CLASSE_1	CLASSE_2
0	0	350	1	468	10	65	0	33	6	0	0	1
0	2	580	1	349	10	51	0	30	6	0	1	0
0	1	500	1	828	18	89	0	29	6	0	0	1
0	0	2000	1	309	6	66	0	29	6	0	1	0
1	0	411	1	390	12	55	1	40	0	0	1	0
0	0	500	1	614	10	81	0	59	6	0	1	0
0	2	500	1	347	4	101	0	44	6	0	1	0
0	0	800	1	359	6	76	0	51	6	0	1	0
0	0	300	1	309	6	66	0	30	6	0	0	1
0	0	554	1	429	10	60	0	46	6	0	1	0
0	0	332	1	593	4	100	0	44	215	198	1	0
1	0	342	1	453	10	59	0	52	6	0	1	0
0	2	632	1	365	6	78	0	49	6	0	1	0
0	0	600	1	587	10	89	0	44	60	0	1	0
0	0	800	1	495	15	59	0	50	6	0	0	1
1	0	480	1	460	3	135	0	59	6	100	1	0
0	0	1300	1	349	10	51	0	45	6	0	1	0
0	2	868	1	567	10	78	0	39	6	0	1	0
0	0	300	1	399	10	55	0	29	6	0	1	0

Análise de Crédito Bancário



Análise de Crédito Bancário

- Base de Dados:

Command Window

New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> A = importdata('treino01.txt');  
  
>> dados=A.data;  
  
>> padroes=dados(1:end,1:11);  
>> targets=dados(1:end,12);
```

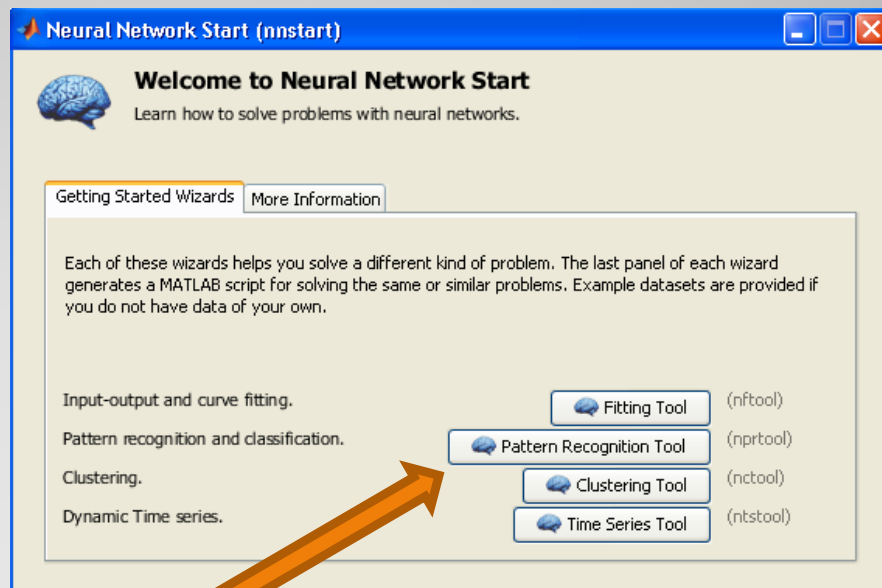
Workspace

Name	Value	Min
A	<1x1 struct>	
dados	<1500x12 double>	0
padroes	<1500x11 double>	0
targets	<1500x1 double>	0

Análise de Crédito Bancário

- Neural Network Start:

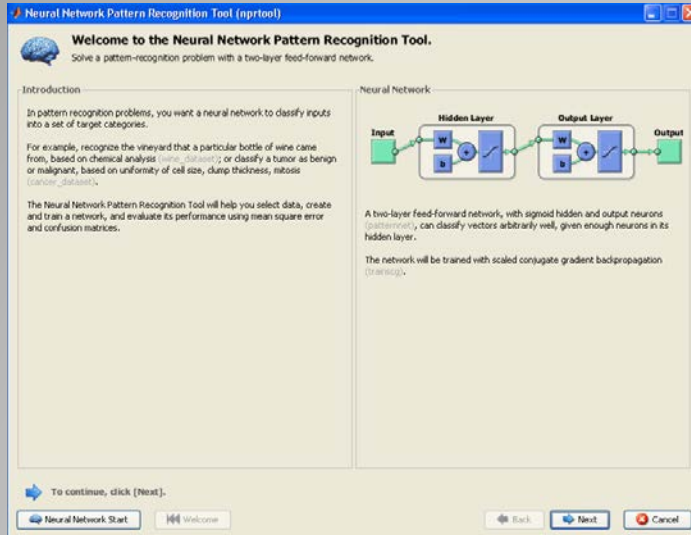
```
>> nnstart
```



Classificação



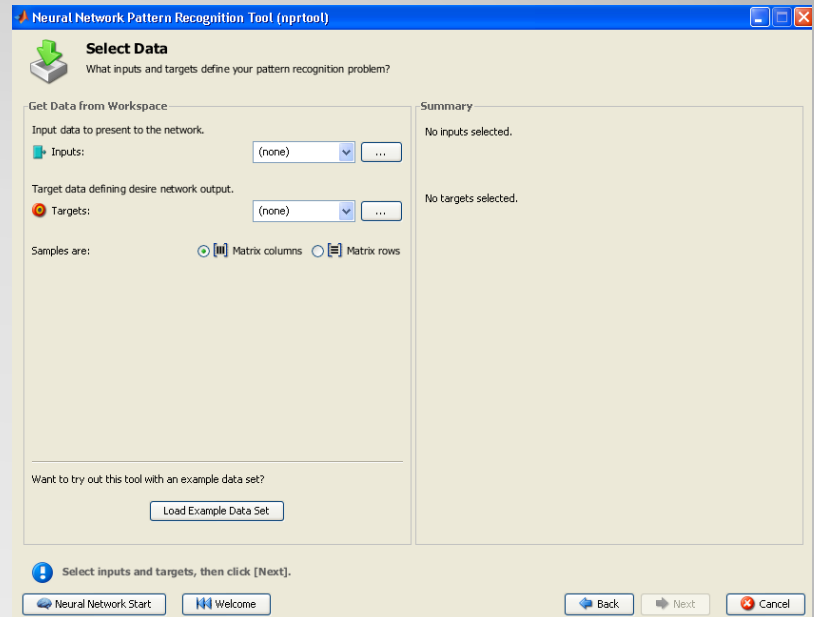
Análise de Crédito Bancário




Selecionar os dados



Informação



Análise de Crédito Bancário



Select Data
What inputs and targets define your pattern recognition problem?

Get Data from Workspace

Input data to present to the network.

Inputs: ...

Target data defining desired network output.

Targets: ...

Samples are: ☐ Matrix columns ☒ Matrix rows



Padrões




Targets




Organização dos dados






Análise de Crédito Bancário


Seleção conjuntos de treino, validação e teste

**Validation and Test Data**
Set aside some samples for validation and testing.


Select Percentages


 Randomly divide up the 1500 samples:


 Training:	70%	1050 samples
 Validation:	15% 	225 samples
 Testing:	15% 	225 samples





Explanation






 Three Kinds of Samples:

 Training:
These are presented to the network during training, and the network is adjusted according to its error.

 Validation:
These are used to measure network generalization, and to halt training when generalization stops improving.

 Testing:
These have no effect on training and so provide an independent measure of network performance during and after training.

 Change percentages if desired, then click [Next] to continue.



Análise de Crédito Bancário

Número de neurônios



Network Architecture

Set the dimensions of the self-organizing map's output layer.

Hidden Layer

Define a pattern recognition neural network. (patternnet)

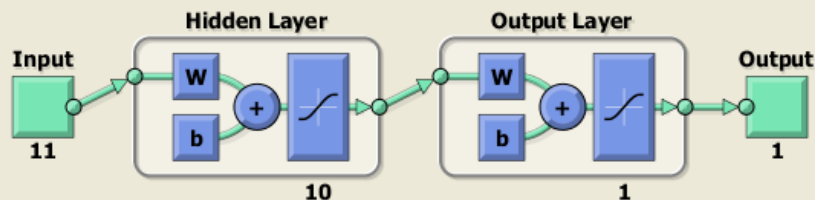
Number of Hidden Neurons:

Restore Defaults

Recommendation


Return to this panel and change the number of neurons if the network does not perform well after training.

Neural Network




Análise de Crédito Bancário

Treinamento da rede

**Train Network**
Train the network to classify the inputs according to the targets.







Train Network



Train using scaled conjugate gradient backpropagation. (trainscg)

 Train




Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Results

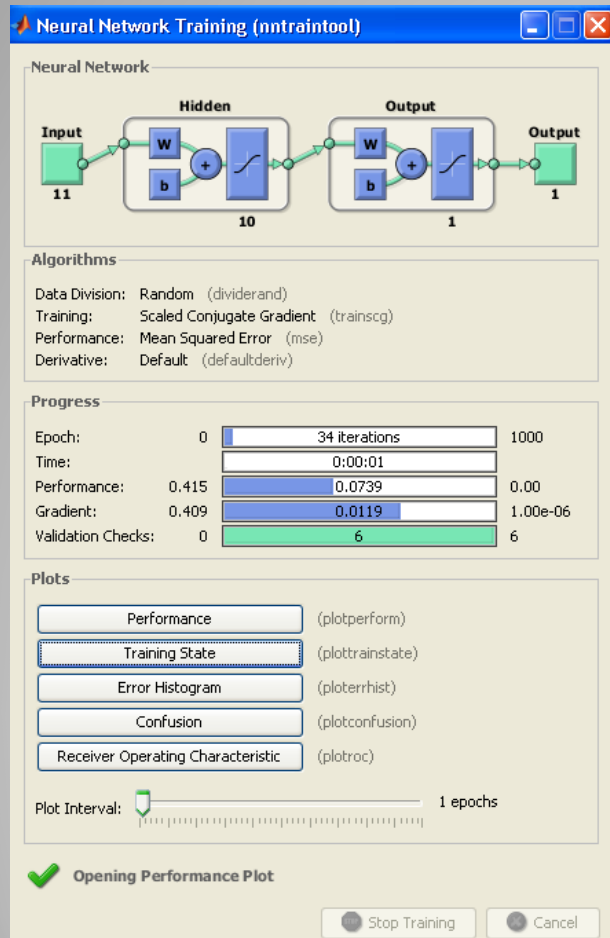
	 Samples	 MSE	 %E
 Training:	1050	-	-
 Validation:	225	-	-
 Testing:	225	-	-

 Plot Confusion  Plot ROC

Notes

-  Training multiple times will generate different results due to different initial conditions and sampling.
-  Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.
-  Percent Error indicates the fraction of samples which are misclassified. A value of 0 means no misclassifications, 100 indicates maximum misclassifications.

Análise de Crédito Bancário



Results

	Samples	MSE	%E icon"/> %E
Training:	1050	7.57541e-2	9.52380e-0
Validation:	225	9.19381e-2	11.55555e-0
Testing:	225	8.78837e-2	9.77777e-0

Plot Confusion Plot ROC

Matriz de Confusão

Comportamento do treinamento

Análise de Crédito Bancário

Classificação:

Treinamento

Validação

Teste

Training Confusion Matrix

Output Class	Target Class	
	0	1
0	531 50.6%	77 7.3%
1	23 2.2%	419 39.9%
	95.8% 4.2%	84.5% 15.5%

Validation Confusion Matrix

Output Class	Target Class	
	0	1
0	116 51.6%	22 9.8%
1	4 1.8%	83 36.9%
	96.7% 3.3%	79.0% 21.0%

Test Confusion Matrix

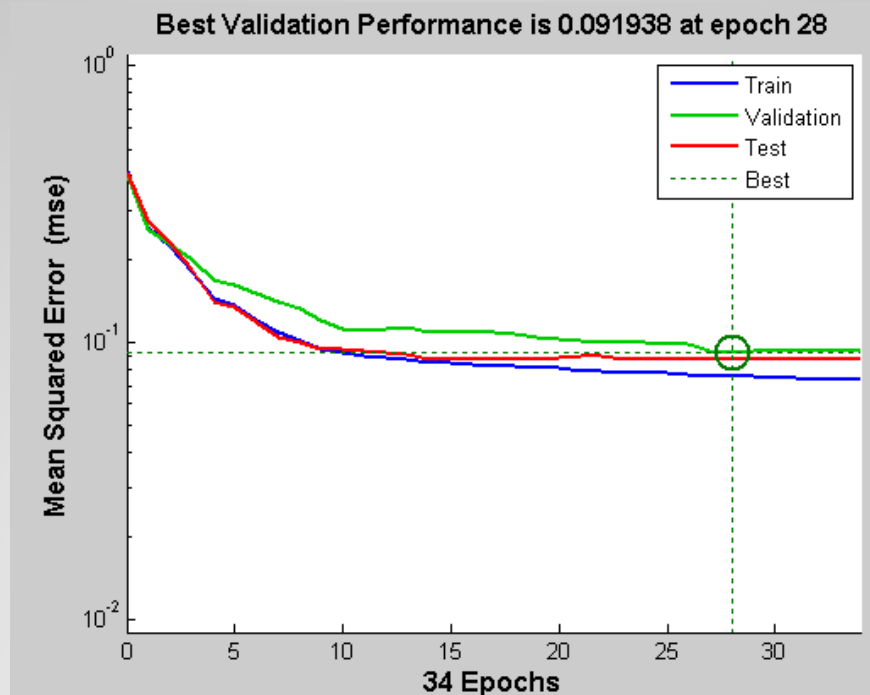
Output Class	Target Class	
	0	1
0	108 48.0%	19 8.4%
1	3 1.3%	95 42.2%
	97.3% 2.7%	83.3% 16.7%

All Confusion Matrix

Output Class	Target Class	
	0	1
0	755 50.3%	118 7.9%
1	30 2.0%	597 39.8%
	96.2% 3.8%	83.5% 16.5%

Análise de Crédito Bancário

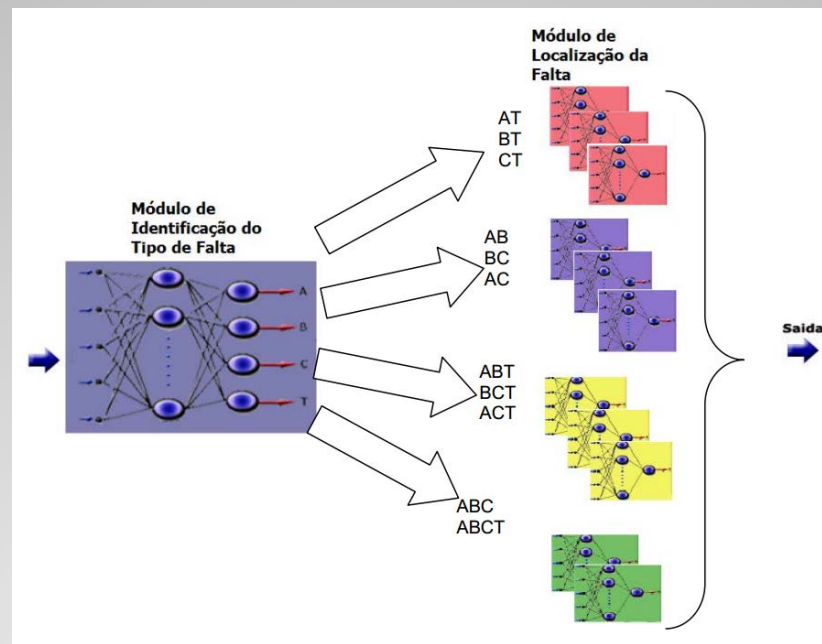
MSE



Toolbox de Redes Neurais

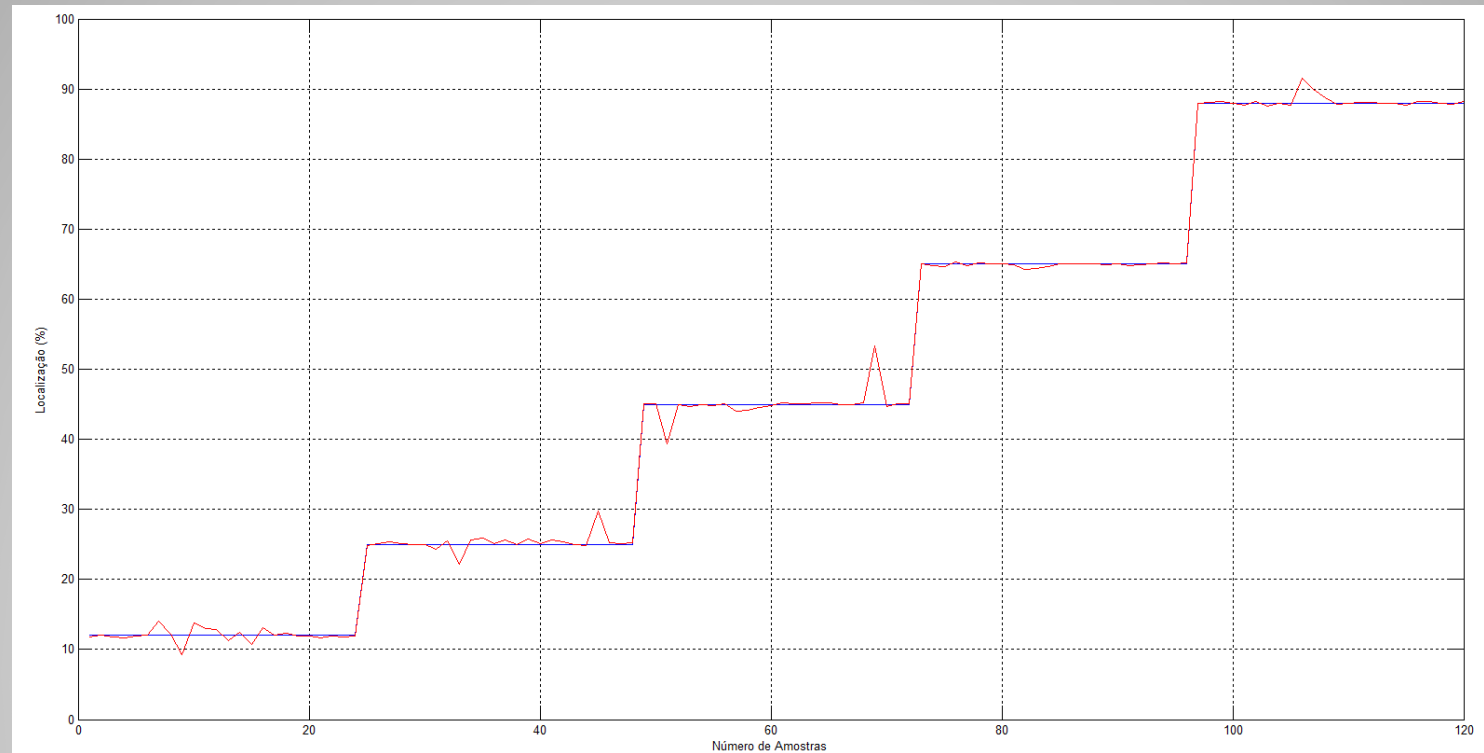
- Introdução ao Matlab
 - Linha de comando
 - NNTool
- Estudo de Casos
 - Análise de Crédito Bancário (Classificação).
 - Localização de Faltas (Inferência).
 - Sensação Térmica (Previsão).

Localização de Falhas



1 ciclo após a falta						2 ciclo após a falta						
Vmedio	Imedio	Zmedia	Pmedio	Vinst	linst	Vmedio	Imedio	Zmedia	Pmedio	Zinst	Vinst	linst
X	X					X	X					

Localização de Faltas



Toolbox de Redes Neurais

- Introdução ao Matlab
 - Linha de comando
 - NNTool
- Estudo de Casos
 - Análise de Crédito Bancário (Classificação).
 - Localização de Faltas (Inferência).
 - Sensação Térmica (Previsão).