

Tópicos

Avançados II

Aula 9

WHAT IS YOUR MOTIVATION WITH NEURAL NETWORKS?

AUGUSTO UCHÔA



Petran- Programa de Pós-graduação
em Engenharia de Transportes

TRILHA DE HOJE:

Decisões de modelagem: Classificação ou regressão?

Conhecer as diferentes formas de criar modelos neurais no Matlab

Conhecer os APP de modelagem neural do Matlab

Criar seu 1º modelo neural completo para classificação de padrões usando o APP nprtool

The MATLAB logo is a stylized 'M' shape composed of several overlapping, curved segments in shades of blue, orange, and yellow, set against a dark red background.

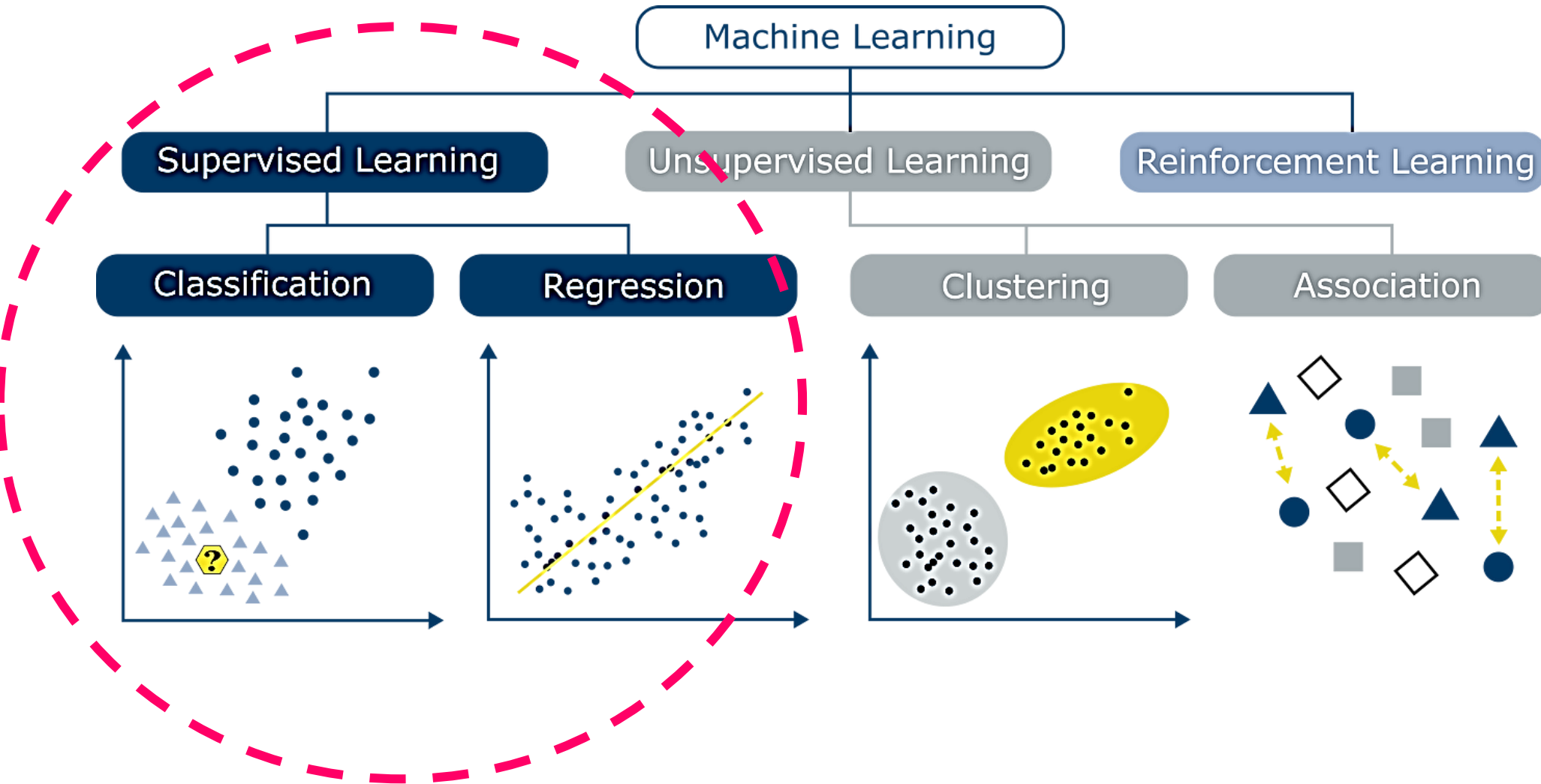
MATLAB

UMA ÚLTIMA QUESTÃO PRECISA SER RESPONDIDA!

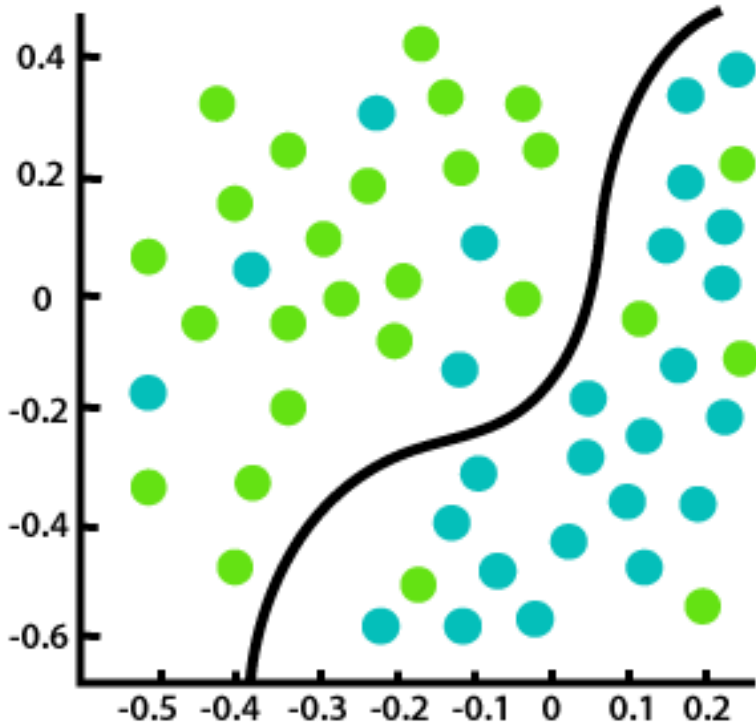


MACHINE LEARNING → 4 CLASSES DE PROBLEMAS

NEURAL NETWORKS (MLP) → 2 CLASSES DE PROBLEMAS

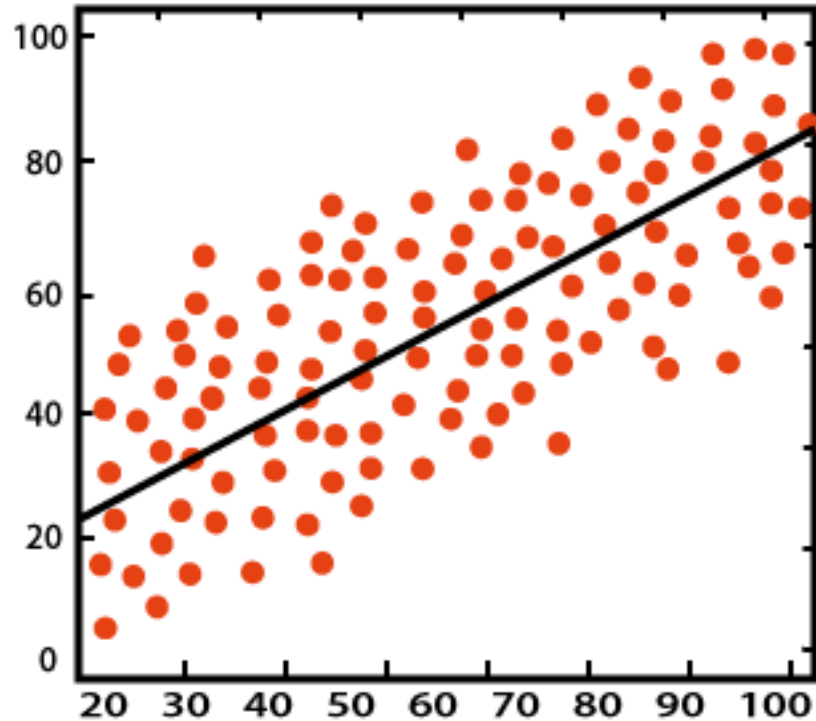


CLASSIFICAÇÃO & REGRESSÃO & CLUSTERIZAÇÃO



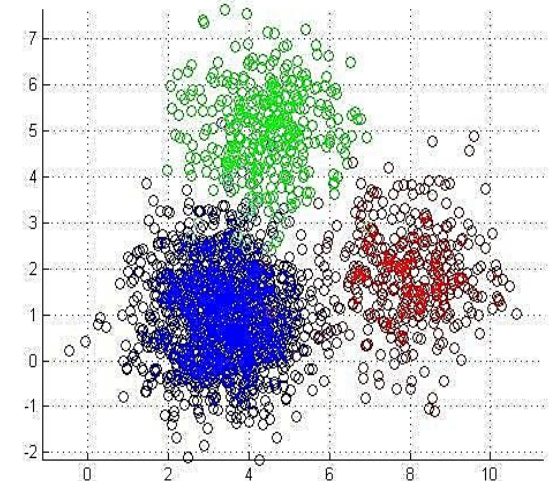
Classification

Dados com labels
discretos



Regression

Dados com labels
contínuos



Clusterization
Dados sem labels

EXISTEM 2 FORMAS DE CRIAR MODELOS NEURAIIS NO MATLAB

Menos flexível em termos de modelagem, mas bem simples de executar

4 APPS

nprtool : Neural Net Pattern Recognition (para classificação)

nftool : Neural Net Fitting (para regressão)

nctool : Neural Net Clustering (para mapas auto organizáveis)

ntstool : Neural Net Time Series (para regressão de séries temporais)

OU

CRIANDO CÓDIGOS

Você faz tudo que desejar e como desejar, contudo é necessário programar em m

Usando as funções e algoritmos já implementadas no Matlab por seus desenvolvedores

nprtool : NEURAL NET PATTERN RECOGNITION (PARA CLASSIFICAÇÃO)

The screenshot displays the nprtool software interface. The title bar reads "Neural Network Pattern Recognition". The main window is titled "NEURAL NETWORK PATTERN RECOGNITION" and features a toolbar with various icons for data import, training, performance analysis, and export. Below the toolbar, there are input fields for "Training data: 70%", "Validation data: 15", and "Test data: 15", along with a "Layer size: 10" dropdown. The main workspace is divided into two panes. The left pane, titled "Network", shows a diagram of a "Two-layer feedforward network with sigmoid hidden neurons and softmax output neurons, suitable for classification tasks." The diagram includes an "Input" layer, a "Hidden" layer with 10 neurons, and an "Output" layer with 6 neurons. Each layer is represented by a box containing weights (W) and biases (b), followed by a summation node (+) and an activation function block. The right pane, titled "Model Summary", contains the text "Train a neural network to classify predictors into a set of classes." The Windows taskbar at the bottom shows the search bar, several application icons, and system information including "Próxima do registro", "POR PTB2", and the date "27/10/2023".

É um aplicativo de modelagem neural que permite criar redes neurais probabilísticas. Ele é fixado em uma interface gráfica do usuário (GUI) que permite a criação de redes neurais personalizadas com base em parâmetros como número de camadas, número de neurônios por camada, função de ativação, etc. O nprtool também permite a importação e exportação de dados em vários formatos.

nftool: NEURAL NET FITTING (PARA REGRESSÃO)

Neural Network Fitting

NEURAL NETWORK FITTING

Training data: 70 %
Validation data: 15
Test data: 15

Layer size: 10

Train Stop

Training State

Performance

Error Histogram

Regression

Fit

Test

Test Plots

Export Plot to Figure

Generate Code

Export Model

DATA SPLIT BUILD TRAIN PLOTS TEST EXPORT

Network Training

Two-layer feedforward network with sigmoid hidden neurons and linear output neurons, suitable for regression tasks.

Input

Hidden

Output

Output

Model Summary

Train a neural network to map predictors to continuous responses.

Pesquisar

Lucros futuros

POR 08:30
PTBZ 27/10/2023

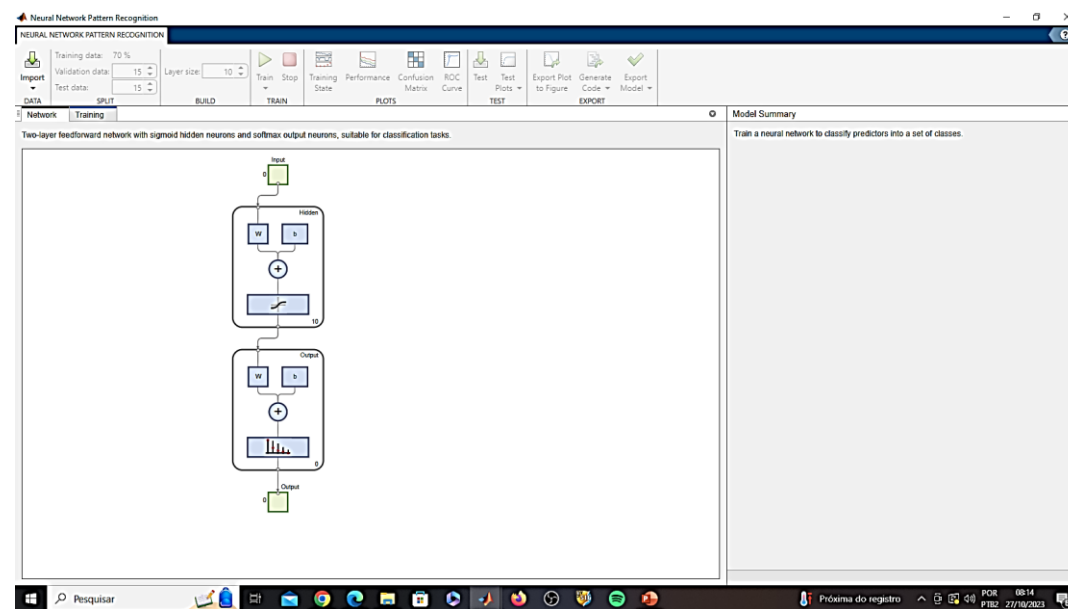
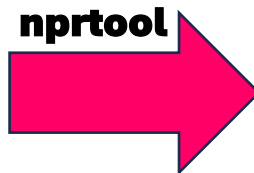
É um aplicativo de modelagem neural que permite criar redes neurais de alimentação direta (feedforward) e retroalimentação (feedback) com camadas ocultas. Ele é fixado em uma interface gráfica do usuário (GUI) que permite a criação de redes neurais personalizadas com base em parâmetros como número de camadas, número de neurônios por camada, função de ativação, etc. O nftool também permite a importação e exportação de dados em vários formatos.

NOSSO EXEMPLO É UM PROBLEMA DE CLASSIFICAÇÃO

(CSO: CLASSE DE SEGURANÇA OPERACIONAL → 0 OU 1)

	A	B	C	D	E	F	G	H
1	ID	CA	PM	RAB	TP	LP	RV	CSO
2	0	0,64	0,62	1,00	1,00	100,00	0,00	0,00
3	1	0,76	0,69	1,00	1,00	100,00	0,00	0,00
4	2	0,64	0,62	1,00	1,00	100,00	0,00	0,00
5	3	0,62	0,79	0,00	1,00	100,00	0,00	0,00
6	4	0,78	0,88	1,00	1,00	100,00	0,00	1,00
7	5	0,67	0,63	1,00	1,00	100,00	0,00	0,00
8	6	0,70	0,67	1,00	1,00	100,00	0,00	0,00
9	7	0,79	0,71	1,00	1,00	100,00	0,00	0,00
10	8	0,63	0,82	1,00	1,00	100,00	0,00	0,00
11	9	0,65	0,78	1,00	1,00	100,00	0,00	0,00
12	10	0,65	1,06	0,00	1,00	100,00	0,00	0,00
13	11	0,86	0,85	1,00	1,00	100,00	0,00	1,00
14	12	0,78	0,78	1,00	1,00	100,00	0,00	0,00
15	13	0,59	0,69	1,00	1,00	100,00	0,00	0,00
16	14	0,68	0,77	0,00	1,00	100,00	0,00	0,00
17	15	0,54	0,92	0,00	1,00	100,00	0,00	0,00
18	16	0,56	1,18	1,00	1,00	100,00	0,00	0,00

nprtool



Prompt inserido no Bing CHAT

“atue como especialista em matlab e machine learning com experiência de 10 anos de capacitação de iniciante e após ter carregado um arquivo excel no matlab com o comando: `dataset = xlsread('dadosexcel.xlsx', 1, 'B2:H1239')`, preciso realizar as seguintes tarefas e preciso de código no matlab para: criar uma ordem randômica das 1238 linhas, normalizar todos os valores usando a função mini-max entre 0 e 1, separar em 2 conjuntos de dados mutuamente exclusivos, o primeiro deve conter 70% dos dados e deve ser nomeado de treinamento, o segundo deve conter o restante dos dados e ser nomeado de teste. Agora cada um desses conjuntos devem ser divididos em dois subconjuntos, a saber: `entradastreino`, nele devem estar contidas os dados da coluna B até a coluna G, e `saidastreino`, deste deve conter os dados da coluna H. De forma análoga o conjunto nomeado de teste deve ser dividido em dois conjuntos, a saber: `entradasteste`, nele devem estar contidas os dados da coluna B até a coluna G, e `saidateste`, deste deve conter os dados da coluna H. Me forneça o código em m contendo comentários explicativos de cada comando”

CÓDIGO EM m CRIADO NO Bing CHAT

```
% Carrega os dados do arquivo Excel
```

```
dataset = xlsread('dadosexcel.xlsx', 1, 'B2:H1239');
```

```
% Cria uma ordem randômica das linhas
```

```
random_order = randperm(size(dataset, 1));
```

```
% Reordena as linhas do dataset
```

```
dataset = dataset(random_order, :);
```

```
% Normaliza os valores usando a função minimax entre 0 e 1
```

```
dataset_normalized = normalize(dataset, 'range');
```

```
% Cria um objeto cvpartition para dividir os dados em conjuntos de treinamento e teste
```

```
c = cvpartition(size(dataset_normalized, 1), 'HoldOut', 0.3);
```

```
% Separa os dados em conjuntos de treinamento e teste
```

```
training_set = dataset_normalized(training(c), :);
```

```
test_set = dataset_normalized(test(c), :);
```

```
% Divide o conjunto de treinamento em entradas e saídas
```

```
entradas_treinamento = training_set(:, 1:6);
```

```
saidas_treinamento = training_set(:, 7);
```

```
% Divide o conjunto de teste em entradas e saídas
```

```
entradas_teste = test_set(:, 1:6);
```

```
saidas_teste = test_set(:, 7);
```

DADOS CARREGADOS E PRÉ-PROCESSADOS

Na aula 8:

- Carregamos no Matlab os dados de um arquivo chamado “dadosexcel.xlsx”, que foi colocado na mesma pasta do script criado, o qual continha uma planilha 1, cujos dados a serem modelados ocupam o range → B2:H1239
- Criou-se um código com a ajuda do Bing CHAT para:
 - a) carregar esses dados como uma matriz (dataset);
 - b) embaralhar os dados, criando uma ordem randômica dos padrões (random_order);
 - c) Normalizar os dados através da função min-max, no intervalo entre 0 e 1; (dataset_normalized)
 - d) Separar os dados em subconjuntos mutuamente exclusivos, com 70% dos dados para treinamento e validação do modelo e 30% para testes (avaliação da qualidade do modelo (training_set e test-set));
 - e) Separar as entradas e saída em cada um desses subconjuntos (entradas-treinamento e saídas_treinamento; entradas_teste e saídas_teste);

The screenshot displays the MATLAB R2023a environment. The main editor window shows a script named 'primeiro.m' with the following code:

```
6  
7 % Reordena as linhas do dataset  
8 dataset = dataset(random_order, :);  
9  
10 % Normaliza os valores usando a função minmax entre 0 e 1  
11 dataset_normalized = normalize(dataset, 'range');  
12  
13 % Cria um objeto cvpartition para dividir os dados em conjuntos de treinamento e teste  
14 c = cvpartition(size(dataset_normalized, 1), 'HoldOut', 0.3);  
15  
16 % Separa os dados em conjuntos de treinamento e teste  
17 training_set = dataset_normalized(training(c), :);  
18 test_set = dataset_normalized(test(c), :);  
19  
20 % Divide o conjunto de treinamento em entradas e saídas  
21 entradas_treinamento = training_set(:, 1:6);  
22 saidas_treinamento = training_set(:, 7);  
23  
24 % Divide o conjunto de teste em entradas e saídas  
25 entradas_teste = test_set(:, 1:6);  
26 saidas_teste = test_set(:, 7);  
27  
28
```

The Command Window shows the execution of the script:

```
>> primeiro  
>> primeiro  
>> primeiro  
>> primeiro  
>> nprtool  
fx >>
```

The Workspace window on the right lists the following variables:

Name	Value
c	1x1 cvpartition
dataset	1238x7 double
dataset_normalized	1238x7 double
entradas_teste	371x6 double
entradas_treinamento	867x6 double
random_order	1x1238 double
saidas_teste	371x1 double
saidas_treinamento	867x1 double
test_set	371x7 double
training_set	867x7 double

ABRA O APP DE CLASSIFICAÇÃO

MATLAB R2023a - academic use

The image displays the MATLAB R2023a interface. The Command Window shows the command `nprtool` being entered. A red dashed circle highlights the Command Window and the `nprtool` text. A red arrow points from the `nprtool` text to the Command Window. The Neural Network Pattern Recognition (nprtool) app is open, showing a two-layer feedforward network diagram with an input layer, a hidden layer with 10 neurons, and an output layer. The app interface includes tabs for Network and Training, and a Model Summary section.

nprtool

Command Window
New to MATLAB: See resources for [Getting Started.](#)
fx >> nprtool

Neural Network Pattern Recognition

NEURAL NETWORK PATTERN RECOGNITION

Training data: 70 %
Validation data: 15
Test data: 15
Layer size: 10

Import
DATA SPLIT BUILD TRAIN TRAINING STATE PERFORMANCE CONFUSION MATRIX ROC CURVE TEST EXPORT

Network Training

Model Summary

Two-layer feedforward network with sigmoid hidden neurons and softmax output neurons, suitable for classification.

Train a neural network to classify predictors into a set of classes.

Input
Hidden
Output

W b
+

10

W b

Windows taskbar: Pesquisar, 30°C, Parc encoberto, POR PTB2, 08:58 27/10/2023

IMPORTE PARA O APP AS ENTRADAS E SAÍDAS PARA TREINAMENTO DA REDE

Neural Network Pattern Recognition

Import

Training data: 70 %
Validation data: 15 %
Test data: 15 %

Layer size: 7

Train Stop
Training State
Performance
Confusion Matrix
ROC Curve
Test Test Plots
Export Plot to Figure
Generate Code
Export Model

Network Training

Two-layer feedforward network with sigmoid hidden neurons and softmax output neurons, suitable for classification tasks.

Input 6

Hidden 7

Output 1

Import Data from Workspace

Select data for training the network.

Predictors: entradas_treinamento - [867x6 double] Browse

Responses: saidas_treinamento - [867x1 double] Browse

Observations in: Columns Rows

entradas_treinamento: double array of 867 observations with 6 features.
saidas_treinamento: double array of 867 observations with 1 classes.

Refresh OK Cancel

Import Data from Workspace

Select data for training the network.

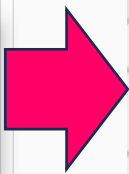
Predictors: entradas_treinamento - [867x6 double] Browse

Responses: saidas_treinamento - [867x1 double] Browse

Observations in: Columns Rows

entradas_treinamento: double array of 867 observations with 6 features.
saidas_treinamento: double array of 867 observations with 1 classes.

Refresh OK Cancel



SUAS PRIMEIRAS REDE MLP NO MATLAB

Arquitetura da rede com :

6 neurônios de entrada, uma camada intermediária e 1 neurônio na camada de saída. Mas quantos neurônios deveria ter a camada intermediária?

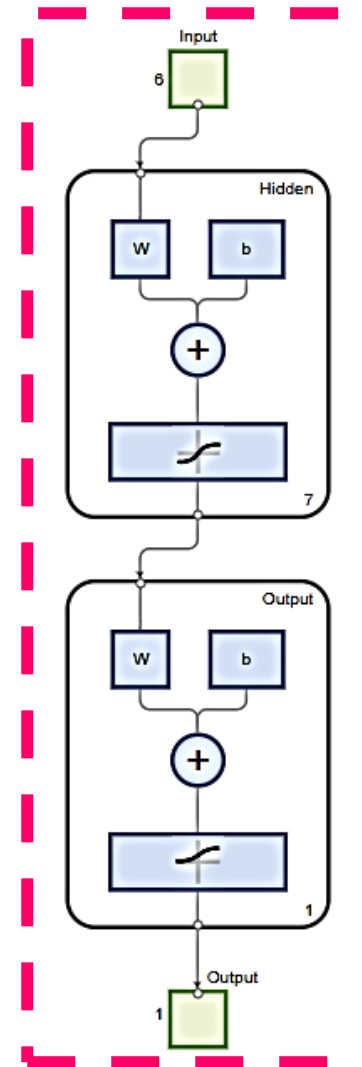
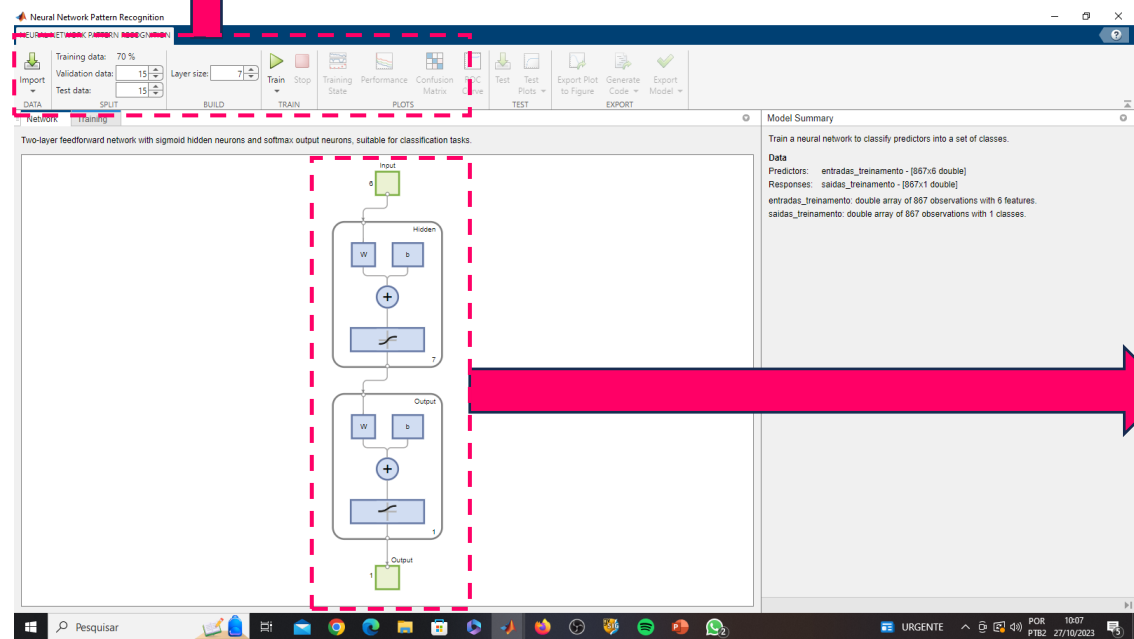
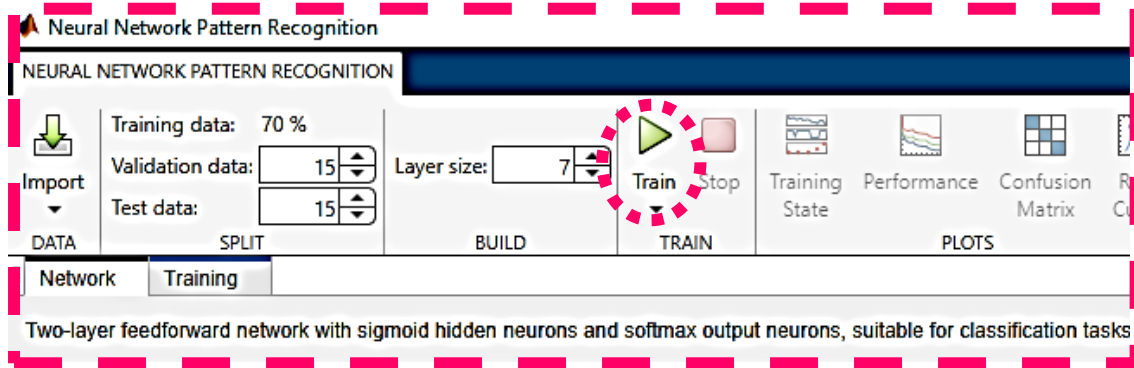
- Segundo HECHT E NIELSEN, 1989 → A única camada intermediária deve ter $2i + 1$ neurônios → **(2. 6) +1= 13 neurônios;**
- Conforme LIPPMANN, 1987 → a única camada intermediária deveria ter $s.(i+1)$ neurônios → **1.(6+1)= 7 neurônios;**

Assim, considerando que o número de neurônios na camada intermediária servirá para criar a curva de separação entre os dados e que esse fenômeno pode ser mais complexo, vamos testar os dois números sugeridos, embora eu tenha mais simpatia pelo 13.

As arquiteturas de suas primeiras redes neurais serão:

6-7-1 e 6-13-1

AJUSTE OS % E O NÚMERO DE NEURÔNIOS E TREINE SUA REDE



Training Results

Training finished: Met validation criterion ✔

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	118	1000
Elapsed Time	-	00:00:00	-
Performance	1.21	0.00511	0
Gradient	0.742	0.00269	1e-06
Validation Checks	0	6	6

Model Summary

Train a neural network to classify predictors into a set of classes.

Data
 Predictors: entradas_treinamento - [867x6 double]
 Responses: saidas_treinamento - [867x1 double]
 entradas_treinamento: double array of 867 observations with 6 features.
 saidas_treinamento: double array of 867 observations with 1 classes.

Algorithm
 Data division: Random
 Training algorithm: Scaled conjugate gradient
 Performance: Cross-entropy error

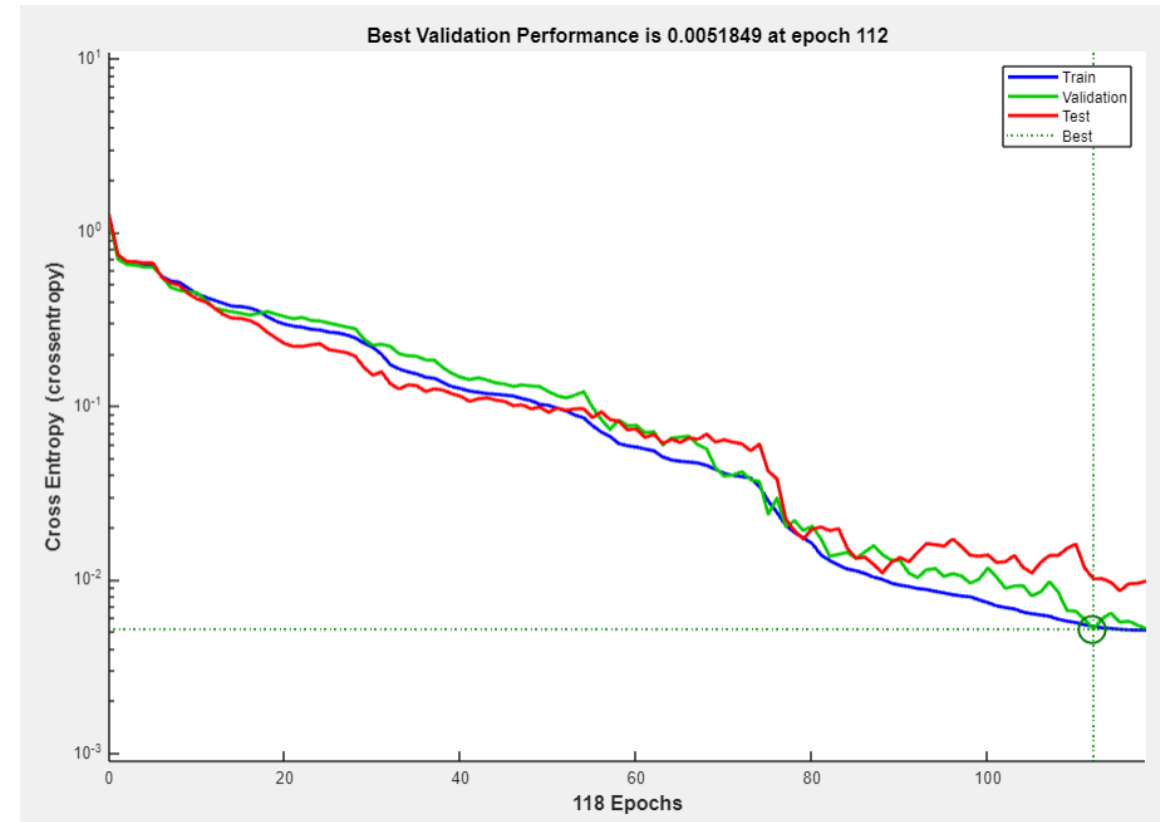
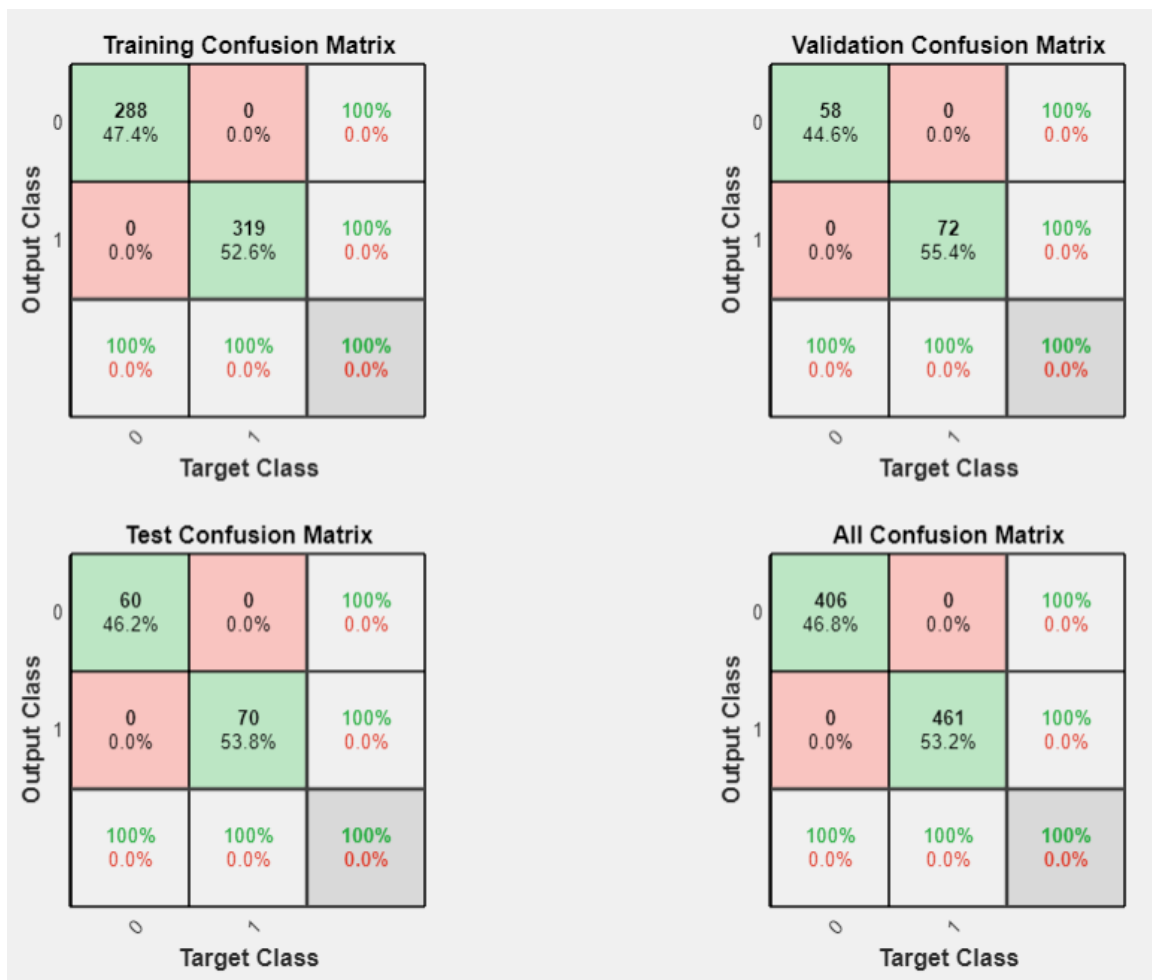
Training Results
 Training start time: 27-Oct-2023 10:13:34
 Layer size: 7

	Observations	Cross-entropy	Error
Training	607	0.0054	0
Validation	130	0.0052	0
Test	130	0.0101	0

RESULTADOS DO TREINAMENTO

PERFORMANCE DO TREINAMENTO E MATRIZ DE CONFUSÃO

- Altos números de classificações corretas nos quadrados verdes (diagonal)
- baixos números de classificações incorretas nos quadrados vermelhos (fora da diagonal)

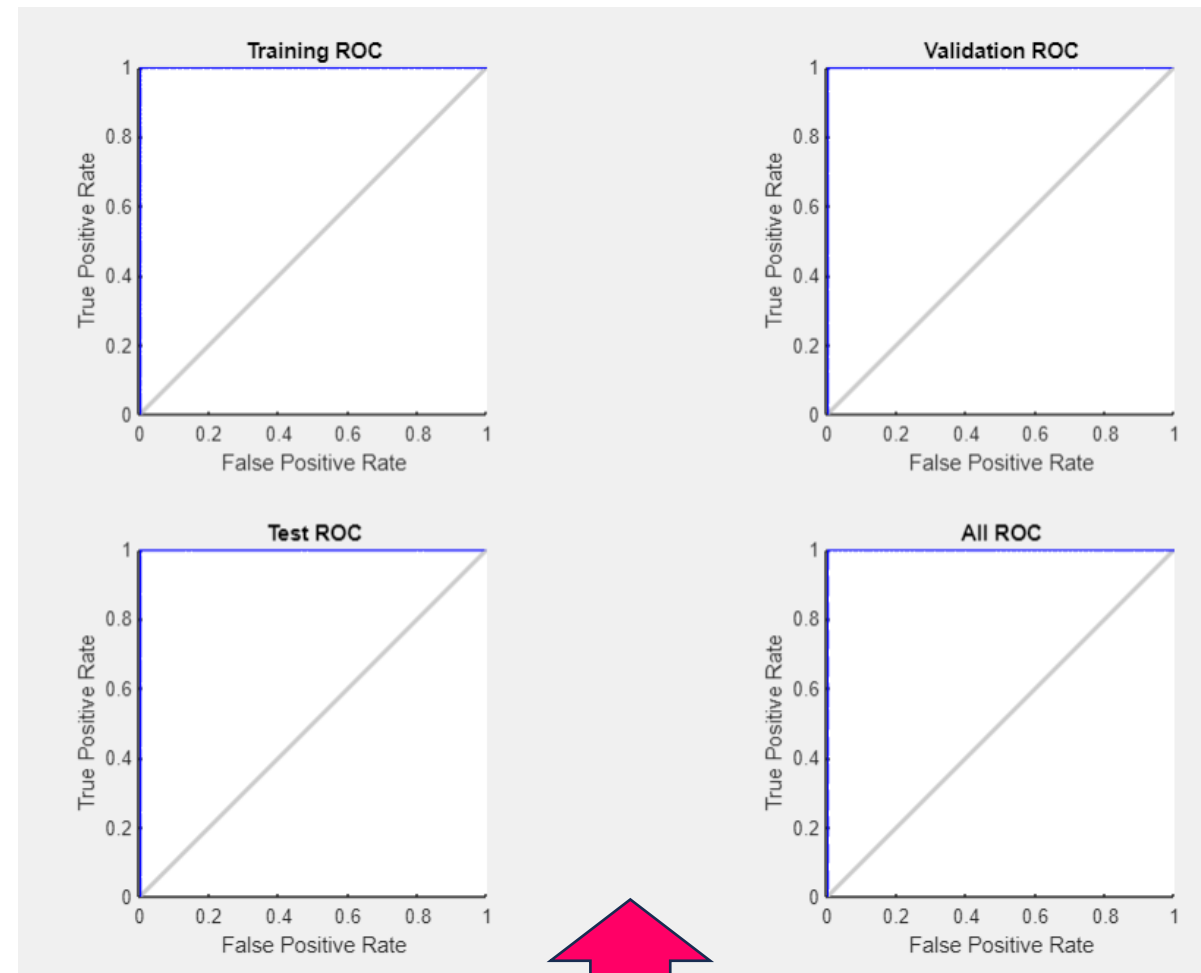


CURVA ROC (Receiver Operating Characteristic)

É uma ferramenta utilizada para avaliar a qualidade de um modelo de classificação binária. Ela é construída a partir da variação dos valores de corte utilizados para separar as classes do modelo. A curva ROC é composta por dois eixos: o eixo x representa a taxa de falsos positivos (1 - especificidade) e o eixo y representa a taxa de verdadeiros positivos (sensibilidade). O ponto (0,0) representa a situação em que o modelo classifica todos os exemplos como negativos, enquanto o ponto (1,1) representa a situação em que o modelo classifica todos os exemplos como positivos. O valor AUC (Area Under the Curve) é uma medida da qualidade do modelo, sendo que um valor próximo de 1 indica um modelo com alta qualidade.

Se você não estiver satisfeito com o desempenho da rede, pode fazer uma das seguintes opções:

- Treine a rede novamente.
- Aumente o número de neurônios ocultos.
- Use um conjunto de dados de treinamento maior.
- Se o desempenho no conjunto de treinamento for bom, mas o desempenho no conjunto de teste for ruim, isso pode indicar que o modelo está superajustado. Reduzir o número de neurônios pode reduzir o superajuste.
- Você também pode avaliar o desempenho da rede em um conjunto de testes adicional. Para carregar dados de teste adicionais para avaliar a rede, na seção Teste, clique em Teste. O Resumo do Modelo exibe os resultados do teste adicional. Você também pode gerar gráficos para analisar os resultados do teste adicional.

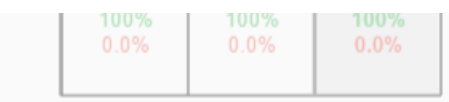


Um teste perfeito mostraria pontos no canto superior esquerdo, com 100% de sensibilidade e 100% de especificidade. Para este problema, a rede funciona muito bem.

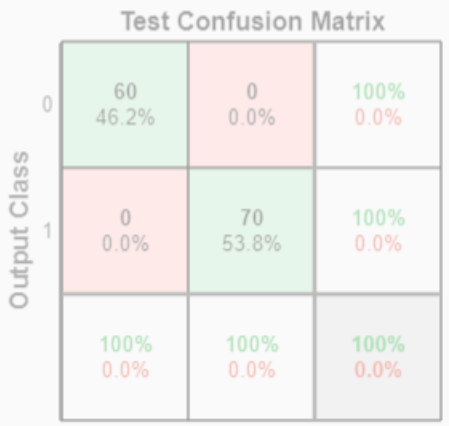
Import
 Training data: 70 %
 Validation data: 15
 Test data: 15
 Layer size: 7
 Train Stop
 Training State
 Performance
 Confusion Matrix
 ROC Curve
 Test Plots
 Export Plot to Figure
 Generate Code
 Export Model

Network Training Performance Plot Confusion Matrix ROC Curve Test Confusion Matrix Test ROC Curve

AFERINDO A QUALIDADE DO MODELO NEURAL

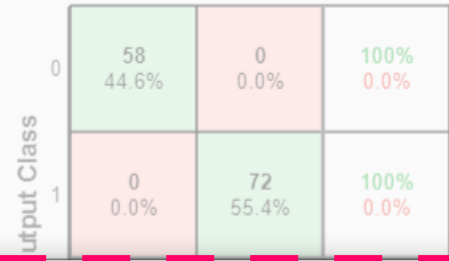


Target Class



Target Class

Validation Confusion Matrix



Test Network on Workspace Data

Select additional data for testing the network.

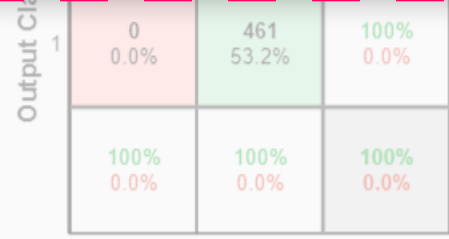
Predictors: entradas_teste - [371x6 double] Browse

Responses: saidas_teste - [371x1 double] Browse

Observations in: Columns Rows

entradas_teste: double array of 371 observations with 6 features.
saidas_teste: double array of 371 observations with 1 classes.

Refresh Test Cancel



Target Class

Model Summary

Train a neural network to classify predictors into a set of classes.

Data
 Predictors: entradas_treinamento - [867x6 double]
 Responses: saidas_treinamento - [867x1 double]
 entradas_treinamento: double array of 867 observations with 6 features.
 saidas_treinamento: double array of 867 observations with 1 classes.

Algorithm
 Data division: Random
 Training algorithm: Scaled conjugate gradient
 Performance: Cross-entropy error

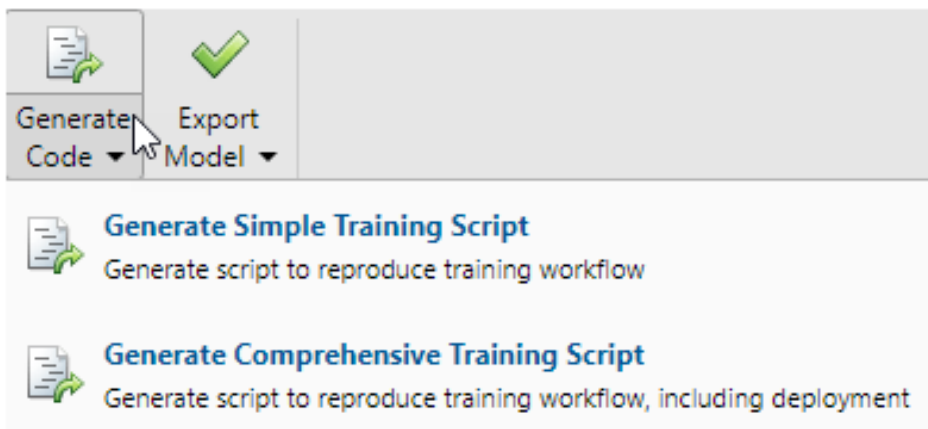
Training Results
 Training start time: 27-Oct-2023 10:13:34
 Layer size: 7

	Observations	Cross-entropy	Error
Training	607	0.0054	0
Validation	130	0.0052	0
Test	130	0.0101	0



Additional Test Results
 Predictors: entradas_teste - [6x371 double]
 Responses: saidas_teste - [1x371 double]
 entradas_teste: double array of 371 observations with 6 features.
 saidas_teste: double array of 371 observations with 1 classes.

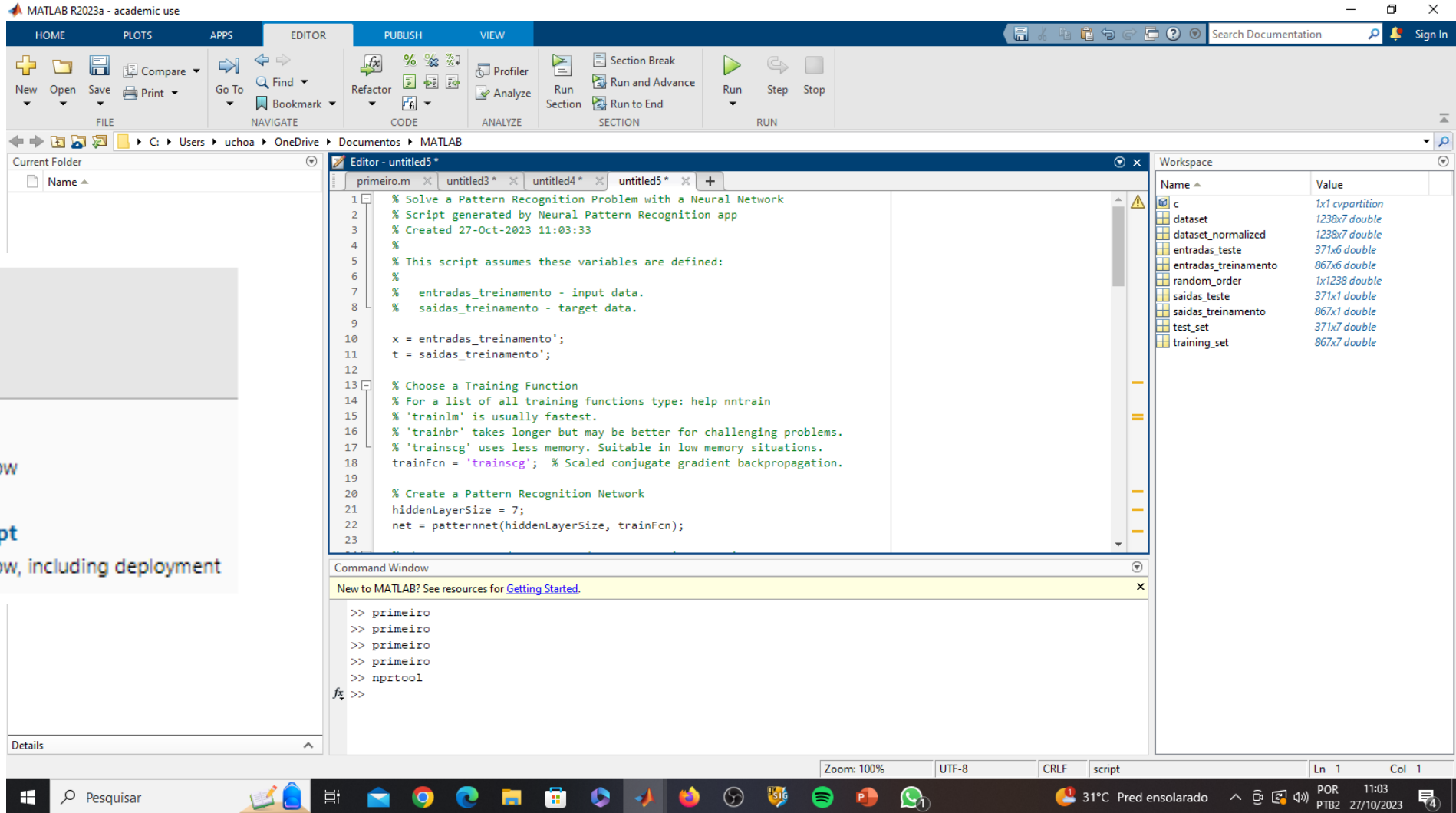
	Observations	Cross-entropy	Error
Additional test	371	0.0153	0.0081

GERAR O CÓDIGO



Generate Code ▾ Export Model ▾

-  **Generate Simple Training Script**
Generate script to reproduce training workflow
-  **Generate Comprehensive Training Script**
Generate script to reproduce training workflow, including deployment



MATLAB R2023a - academic use

HOME PLOTS APPS EDITOR PUBLISH VIEW

FILE NAVIGATE CODE ANALYZE SECTION RUN

Current Folder: C:\Users\uchoa\OneDrive\Documentos\MATLAB

Editor - untitled5*

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 27-Oct-2023 11:03:33
4 %
5 % This script assumes these variables are defined:
6 %
7 % entradas_treinamento - input data.
8 % saidas_treinamento - target data.
9 %
10 x = entradas_treinamento';
11 t = saidas_treinamento';
12 %
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.
19 %
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 7;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
```

Workspace

Name	Value
c	1x1 cvpartition
dataset	1238x7 double
dataset_normalized	1238x7 double
entradas_teste	371x6 double
entradas_treinamento	867x6 double
random_order	1x1238 double
saidas_teste	371x1 double
saidas_treinamento	867x1 double
test_set	371x7 double
training_set	867x7 double

Command Window

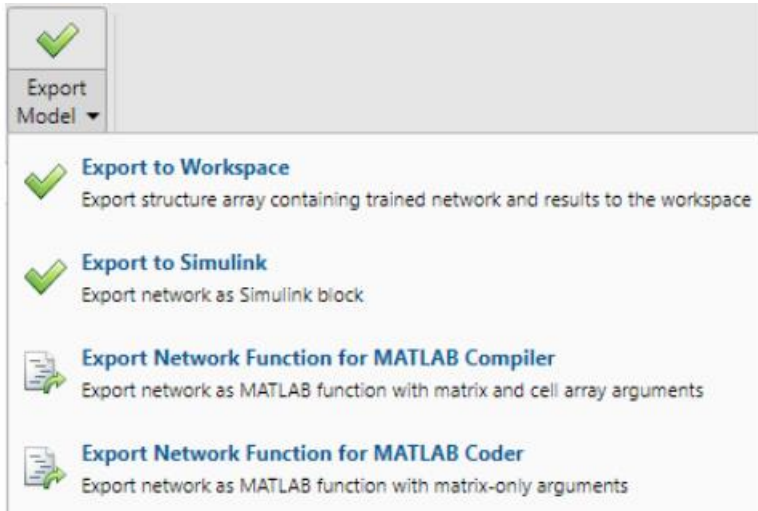
```
>> primeiro
>> primeiro
>> primeiro
>> primeiro
>> nprtool
fx >>
```

Zoom: 100% UTF-8 CRLF script Ln 1 Col 1

Gerar Código → Gerar Script de Treinamento Simples para criar código MATLAB para reproduzir os passos anteriores a partir da linha de comando.

Criar código MATLAB pode ser útil se você quiser aprender a usar a funcionalidade de linha de comando da caixa de ferramentas para personalizar o processo de treinamento. Em Reconhecimento de Padrões Usando Funções de Linha de Comando, você investigará os scripts gerados com mais detalhes.

EXPORTAR O MODELO NEURAL TREINADO



Você pode exportar sua rede treinada para o espaço de trabalho ou Simulink®. Você também pode implantar a rede com o MATLAB Compiler™ e outras ferramentas de geração de código MATLAB. Para exportar sua rede treinada e resultados, selecione Exportar Modelo > Exportar para Espaço de Trabalho.

A maneira mais fácil de aprender a usar a funcionalidade de linha de comando da caixa de ferramentas é gerar scripts a partir dos aplicativos e, em seguida, modificá-los para personalizar o treinamento da rede. Como exemplo, veja o script simples que foi criado na seção anterior usando o aplicativo Neural Net Pattern Recognition.

```
% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by Neural Pattern Recognition app
% Created 22-Mar-2021 16:50:20
%
% This script assumes these variables are defined:
%
%   glassInputs - input data.
%   glassTargets - target data.

x = glassInputs;
t = glassTargets;

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network
hiddenLayerSize = 10;
net = patternnet(hiddenLayerSize, trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);

% View the Network
view(net)

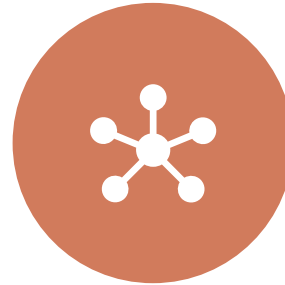
% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)
```



PONTOS CHAVE



Todas as decisões a tomar pelo modelador



Formas de Modelar redes neurais no MATLAB



APPs de Ccriação de modelos neurais no Matlab



Criação do 1º modelo neural completo com o APP nrptool

A SEGUIR, CENAS DO
PRÓXIMO
CAPÍTULO

**CRIAÇÃO DE
MODELO NEURAL
DE REGRESSÃO
COM O APP nftool**

