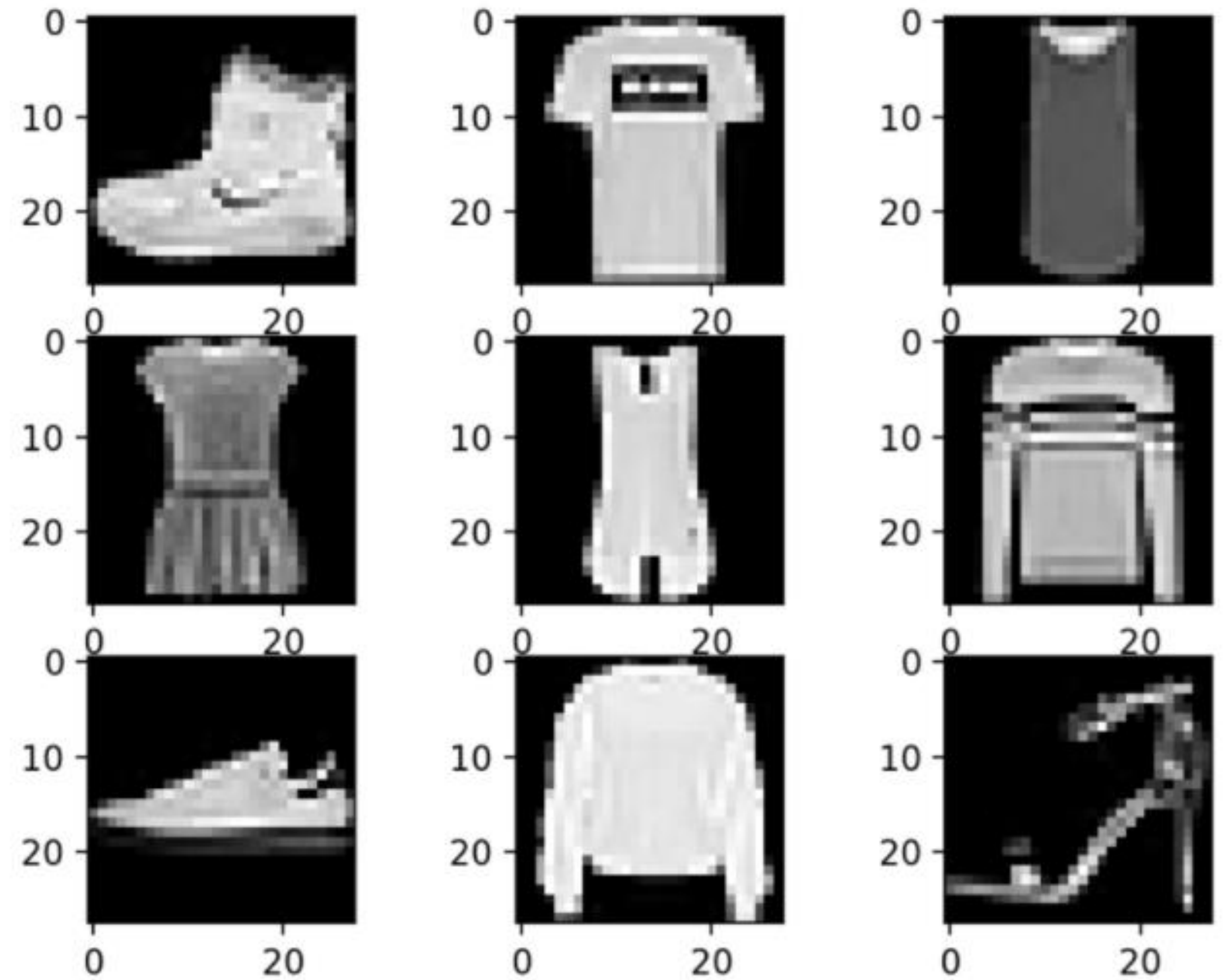


# PROJETO – FASE I

## FASHION MNIST



## DESCRIÇÃO DO CASO DE ESTUDO E OBJETIVOS DO PROBLEMA

Fashion-MNIST é um conjunto de dados de imagens de artigos da Zalando. Esta consiste num conjunto de treino de 60.000 exemplos e um conjunto de teste com 10.000 exemplos.

Cada exemplo é uma imagem em tons de cinza de 28x28, associada a um rótulo de 10 classes. A Zalando pretende que o Fashion-MNIST sirva como um substituto direto para o conjunto de dados MNIST original para benchmarking de algoritmos de machine learning.

Compartilha o mesmo tamanho de imagem e estrutura de divisões de treino e teste.

# DESCRIÇÃO DA IMPLEMENTAÇÃO DOS ALGORITMOS

Para a implementação deste dataset utilizámos a seguinte rede no MATLAB:

**Redes MLP** → Redes de múltipla camada – Redes MLP com Backpropagation

Em modelos MLP todos os neurônios são ligados aos neurônios da camada subsequente. Não há nem ligação com os neurônios laterais de mesma camada nem acontece a realimentação da rede.

A aprendizagem de rede MLP é conhecida como aprendizagem por experiência e dá-se por inúmeras iterações.

Os exemplos de treinamento são apresentados à camada de entrada e com base nos erros obtidos, é realizado ajustes nos pesos sinápticos com o objetivo de diminuir o erro nas próximas iterações.

## DEFINIÇÃO DAS ENTRADAS E SAÍDAS DA REDE

Os dados para o treino e de teste usámos o dataset Fashion-MNIST. Cada imagem tem um único item de roupa de tamanho 28x28, para o treino temos dois conjuntos em que o primeiro é uma matriz 784x60000 com todas as imagens (em grayscale) e outra matriz 10x60000 que corresponde às 10 classes, onde cada classe é representada por um valor de 0 a 9 que corresponde a um certo item de roupa.

Para a saída temos valores de 1 a 10 que corresponde a cada peça de roupa.

0-T-shirt/top    1 calças    2 Pullover    3 vestido    4 Casaco    5 sandálias    6-Camisa    7-sapatilhas    8-mala    9-bota tornozelo

# ANÁLISE DE RESULTADOS

Teste Padrão ~= 1. 20 min

Número de iterações: 1000

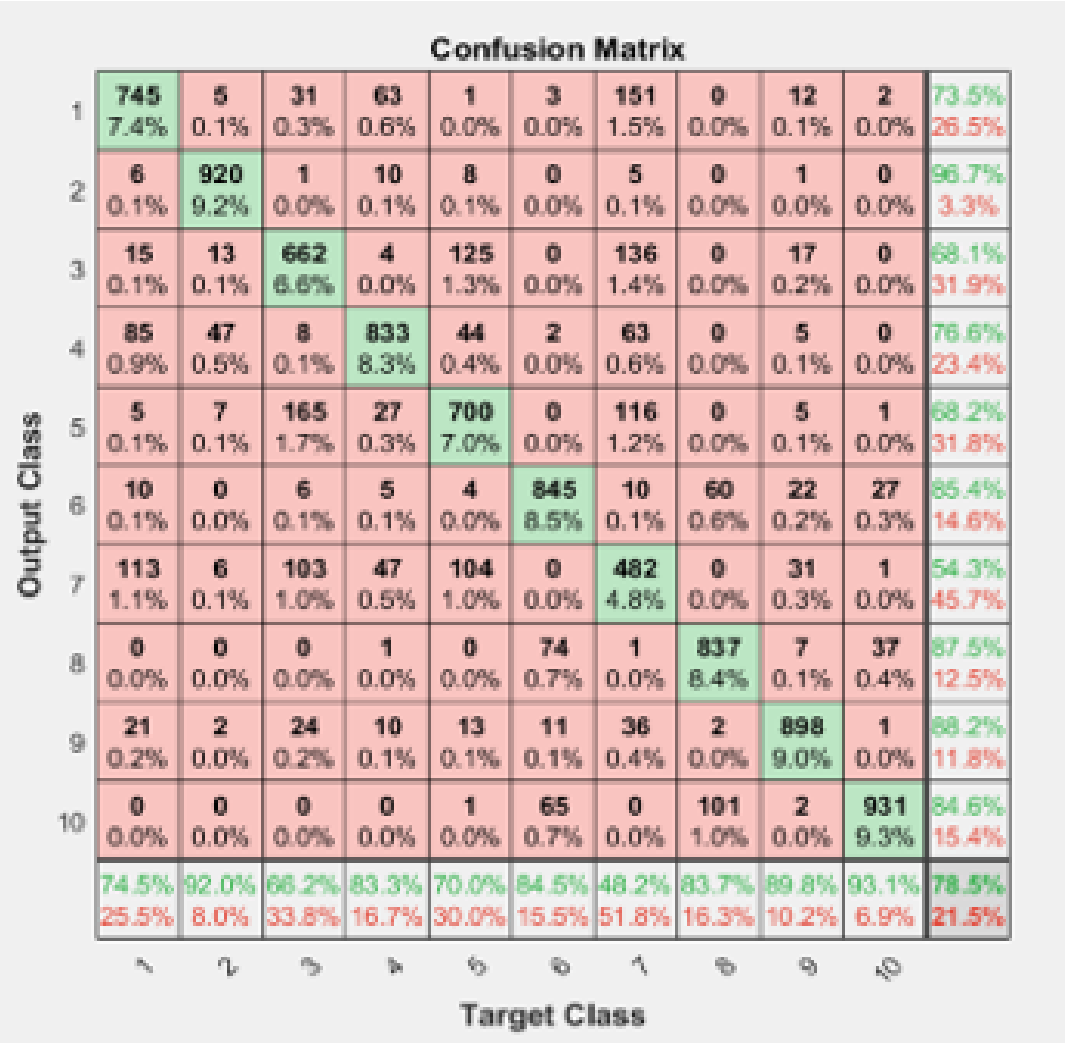
Coeficiente de aprendizagem: 0.07

Número de neurónios: 10

Função de treino: traingd

Tipo de rede: patternet

Resultados
78.5
77.5
78.2
78.0
78.5



Teste 1 - Alteração do Coeficiente de aprendizagem

~=1.40 min

Número de iterações: 1000

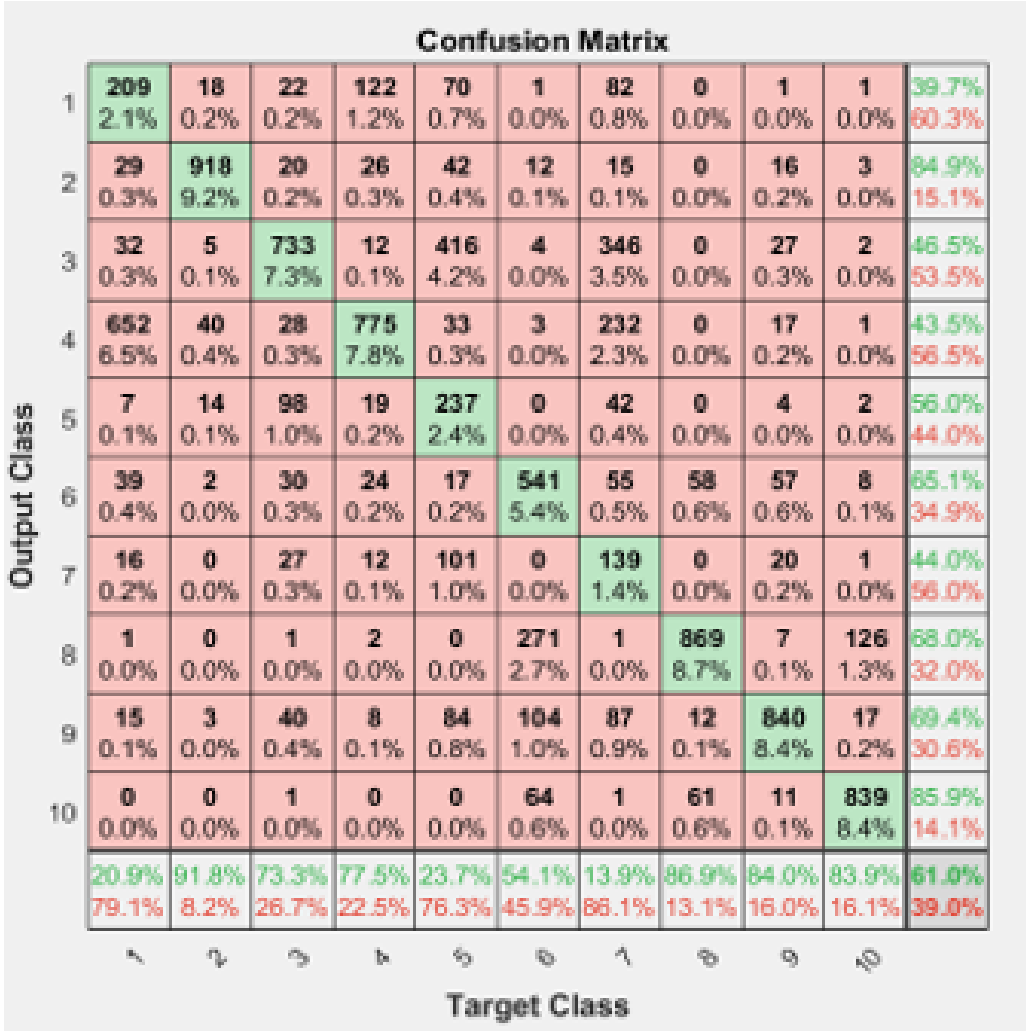
Coeficiente de aprendizagem: **0.01**

Número de neurónios: 10

função de treino: traingd

tipo de rede: patternet

Resultados %
61.0
56.8
60.6
63.7
64.8



Acuracy	0.95	0.99	0.94	0.96	0.94	0.97	0.91	0.97	0.98	0.98
Precisão	0.74	0.97	0.68	0.77	0.68	0.85	0.54	0.87	0.88	0.85
Sensibilidade	0.74	0.92	0.66	0.83	0.70	0.84	0.48	0.84	0.90	0.93
Especificidade	0.97	1.00	0.97	0.97	0.96	0.98	0.95	0.99	0.99	0.98

Acuracy	0.89	0.98	0.89	0.88	0.91	0.93	0.90	0.95	0.95	0.97
Precisão	0.40	0.85	0.46	0.44	0.56	0.65	0.44	0.68	0.69	0.86
Sensibilidade	0.21	0.92	0.73	0.78	0.24	0.54	0.14	0.87	0.84	0.84
Especificidade	0.96	0.98	0.91	0.89	0.98	0.97	0.98	0.95	0.96	0.98

Teste 2 - Alteração do Número de neurónios ~=2min

Número de iterações: 1000

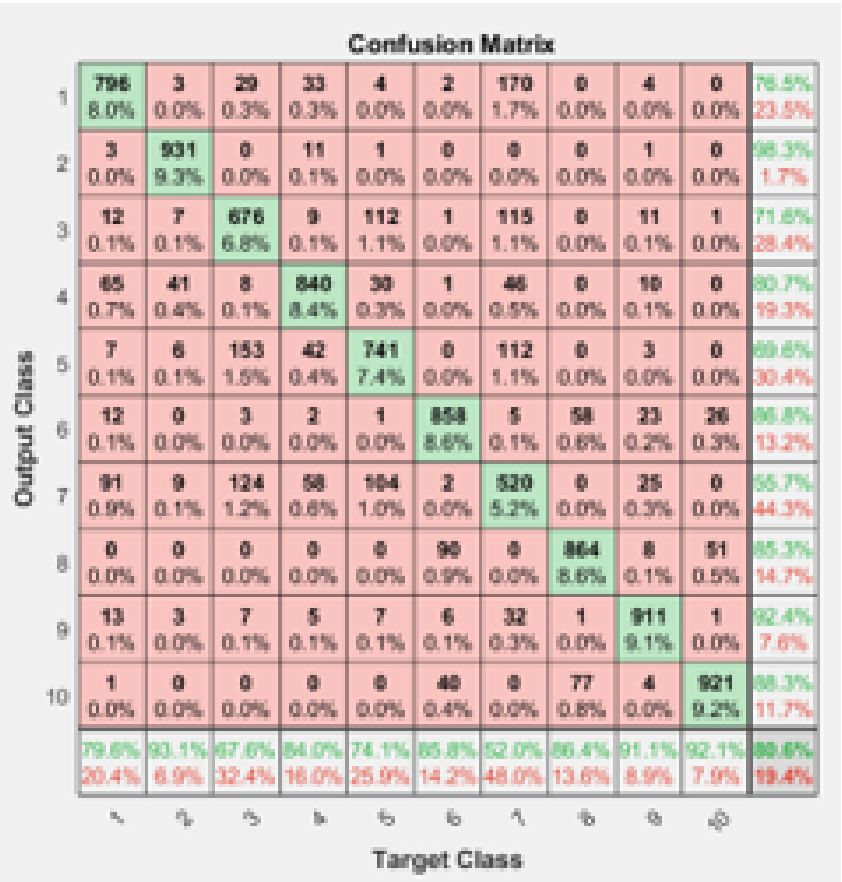
Coeficiente de aprendizagem: 0.07

Número de neurónios: 20

função de treino: traingd

tipo de rede: patternet

Resultados %
80.6
81.0
80.9
80.1
81.0



Teste 3 - Alteração do Número de neurónios ~= 2min

Número de iterações: 1000

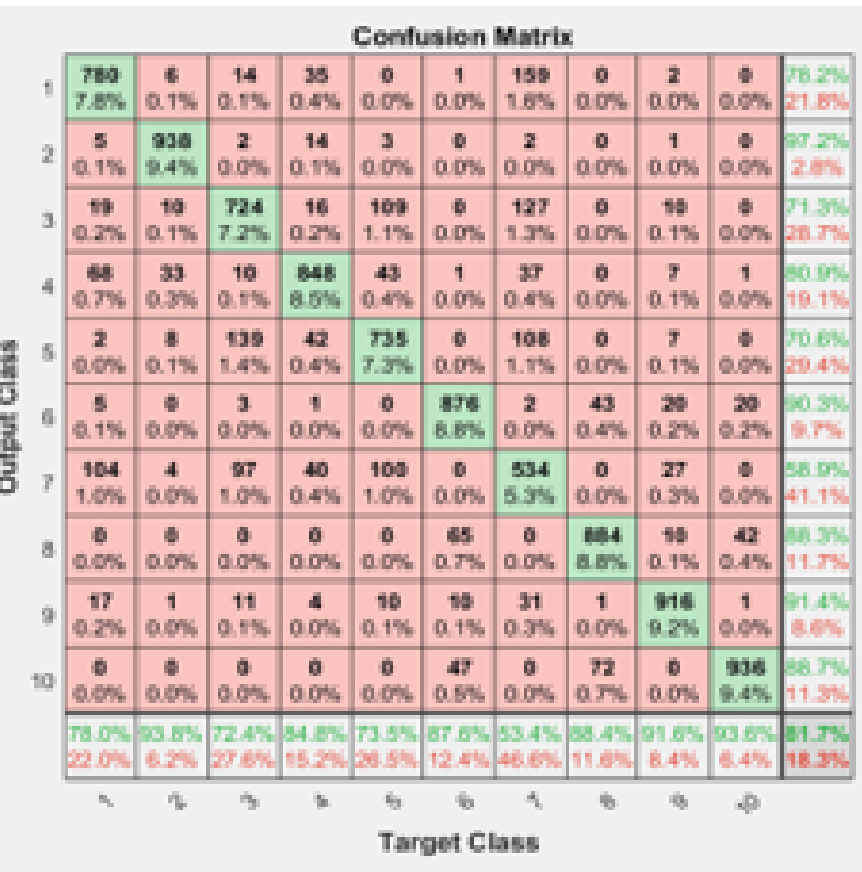
Coeficiente de aprendizagem: 0.07

Número de neurónios: 30

função de treino: traingd

tipo de rede: patternet

Resultados %
81.7
80.8
81.3
81.8
81.7



Acuracy	0.96	0.99	0.94	0.96	0.94	0.97	0.91	0.97	0.98	0.98
Precisão	0.76	0.98	0.72	0.81	0.70	0.87	0.56	0.85	0.92	0.88
Sensibilidade	0.80	0.93	0.68	0.84	0.74	0.86	0.52	0.86	0.91	0.92
Especificidade	0.97	1.00	0.97	0.98	0.96	0.99	0.95	0.98	0.99	0.99

Acuracy	0.96	0.99	0.94	0.96	0.94	0.98	0.92	0.98	0.98	0.98
Precisão	0.78	0.97	0.71	0.81	0.71	0.90	0.59	0.88	0.91	0.89
Sensibilidade	0.78	0.94	0.72	0.85	0.73	0.88	0.53	0.88	0.92	0.94
Especificidade	0.98	1.00	0.97	0.98	0.97	0.99	0.96	0.99	0.99	0.99



Teste 4 - Alteração do tipo de rede ~= 1min

Número de iterações: 1000

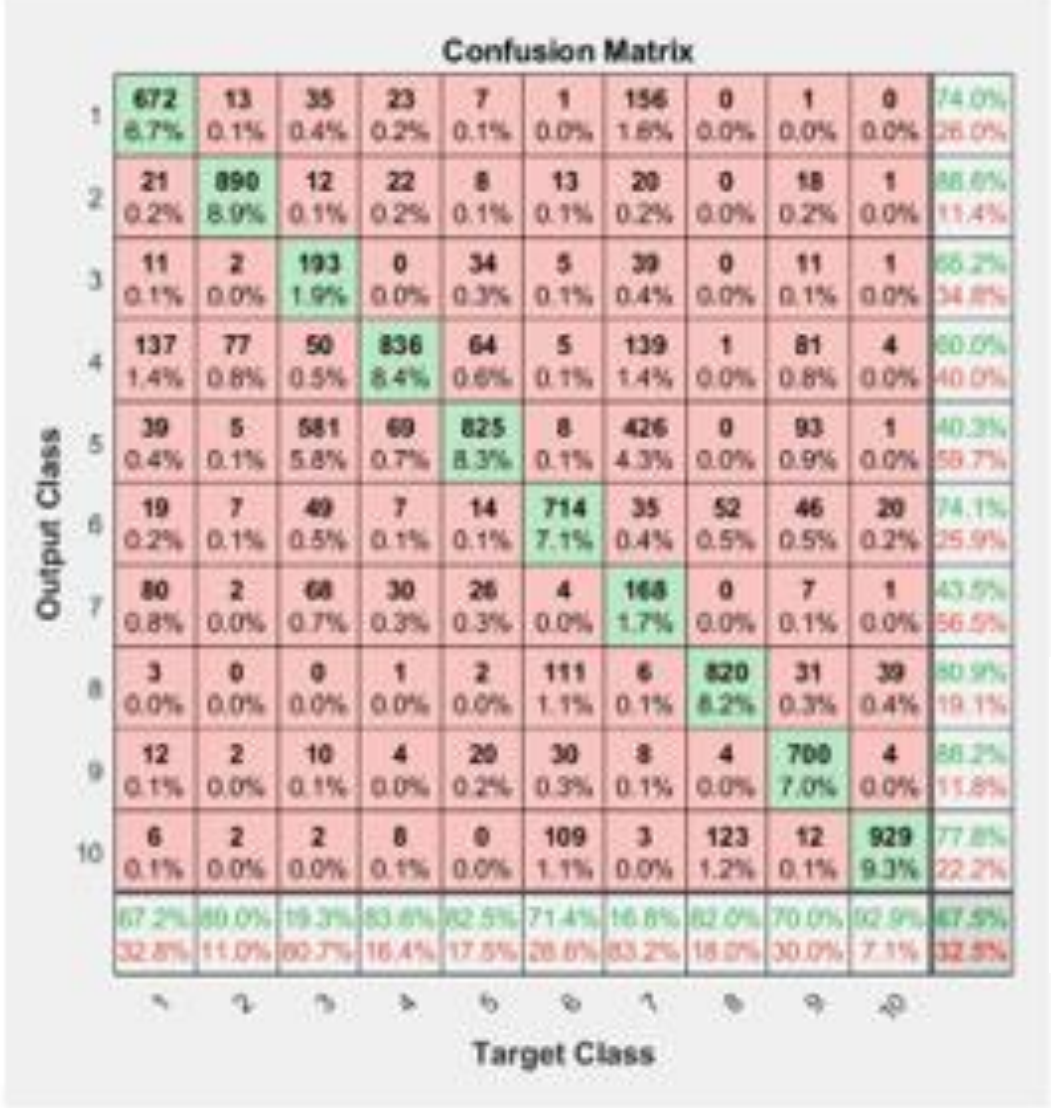
Coeficiente de aprendizagem: 0.07

Número de neurónios: 10

função de treino: traingd

tipo de rede: feedforwardnet

Resultados %
67.5
71.7
62.1
71.9
64.8



Acuracy	0.94	0.98	0.91	0.93	0.86	0.95	0.90	0.96	0.96	0.97
Precisão	0.74	0.89	0.65	0.60	0.40	0.74	0.44	0.81	0.88	0.78
Sensibilidade	0.67	0.89	0.19	0.84	0.82	0.71	0.17	0.82	0.70	0.93
Especificidade	0.97	0.99	0.99	0.94	0.86	0.97	0.98	0.98	0.99	0.97

Teste 5- Alteração do número de iterações ~=14 min

Número de iterações: **10000**

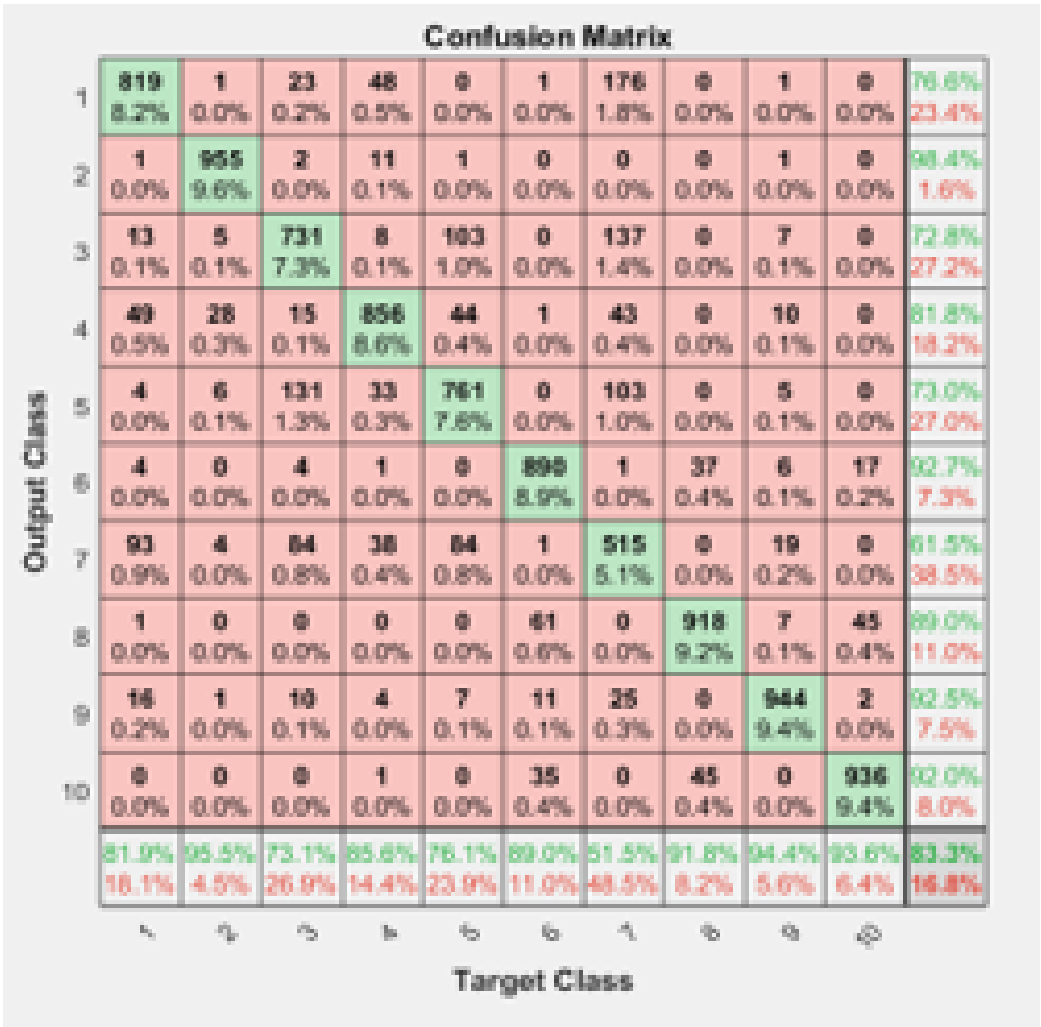
Coeficiente de aprendizagem: 0.07

Número de neurónios: 10

função de treino: traingd

tipo de rede: patternet

Resultados %
83.3
83.0



Acuracy	0.96	0.99	0.95	0.97	0.95	0.98	0.92	0.98	0.99	0.99
Precisão	0.77	0.98	0.73	0.82	0.73	0.93	0.61	0.89	0.93	0.92
Sensibilidade	0.82	0.95	0.73	0.86	0.76	0.89	0.52	0.92	0.94	0.94
Especificidade	0.97	1.00	0.97	0.98	0.97	0.99	0.96	0.99	0.99	0.99

Teste 6- Alteração do número de iterações ~=8 min

Número de iterações: 5000

Coeficiente de aprendizagem: 0.07

Número de neurónios: 10

função de treino: traingd

tipo de rede: patternet

Resultados %
82.6
82.9



Acuracy	0.96	0.99	0.95	0.97	0.95	0.98	0.91	0.98	0.99	0.98
Precisão	0.78	0.97	0.74	0.82	0.73	0.93	0.58	0.89	0.92	0.90
Sensibilidade	0.81	0.94	0.72	0.85	0.75	0.87	0.53	0.91	0.95	0.94
Especificidade	0.98	1.00	0.97	0.98	0.97	0.99	0.96	0.99	0.99	0.99

Teste 7- Alteração do tipo de rede

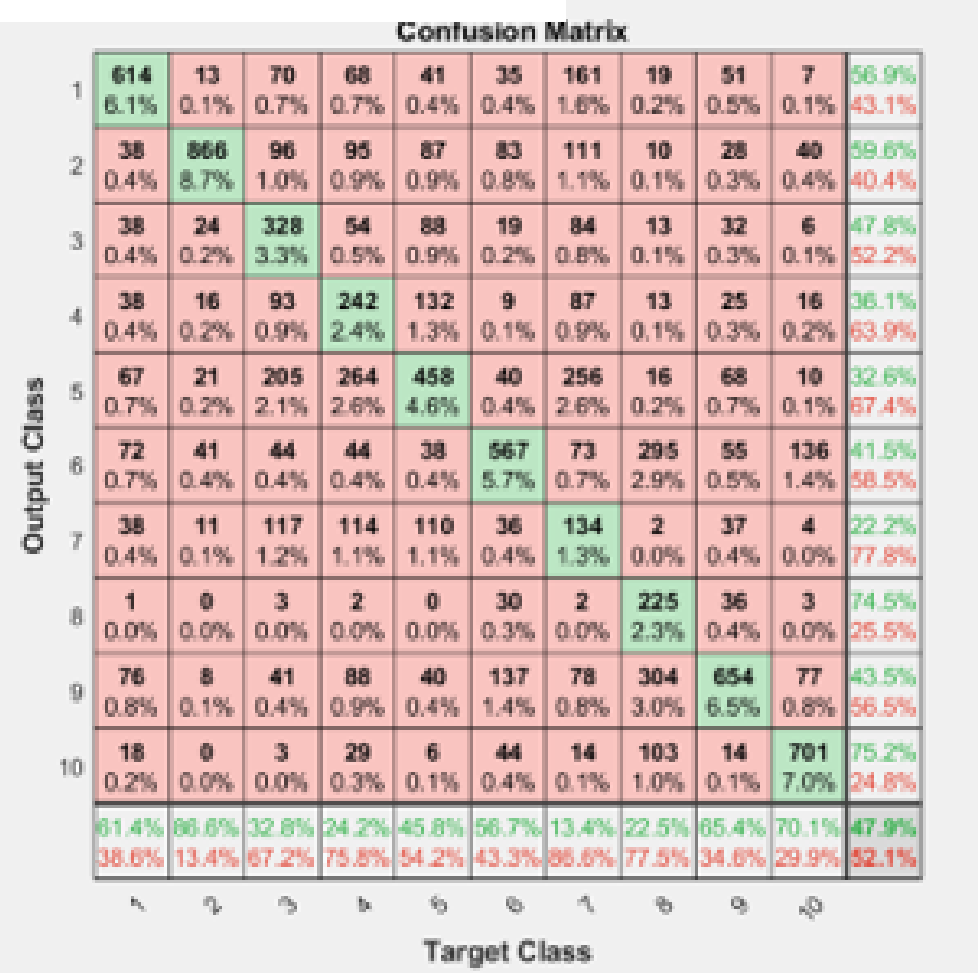
Número de iterações: 1000

Coeficiente de aprendizagem: 0.01

Número de neurónios: 10

função de treino: traingd

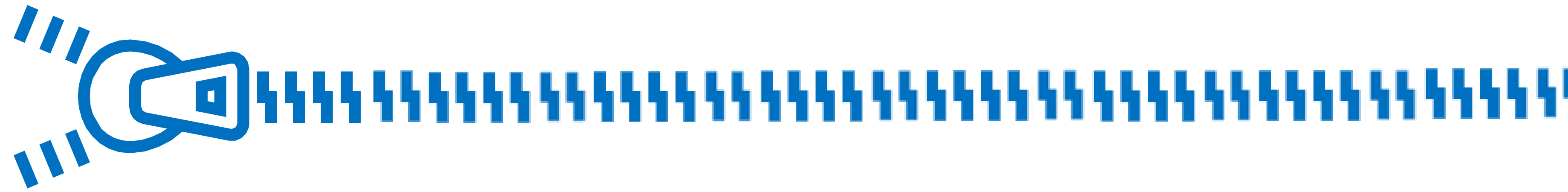
tipo de rede: newff



Acuracy	0.91	0.93	0.90	0.88	0.85	0.88	0.87	0.91	0.88	0.95
Precisão	0.57	0.60	0.82	0.36	0.33	0.42	0.22	0.75	0.44	0.75
Sensibilidade	0.61	0.87	0.85	0.24	0.46	0.57	0.13	0.23	0.65	0.70
Especificidade	0.95	0.93	0.96	0.95	0.89	0.91	0.95	0.99	0.91	0.97



# CONCLUSÕES



Em relação ao primeiro teste, podemos concluir que a alteração do coeficiente de aprendizagem de 0.07 para 0.01, piora a performance da rede, que passou de apresentar resultados de  $\sim 78\%$  para  $\sim 61\%$ . O coeficiente de aprendizagem é o parâmetro que controla o ajuste dos pesos, um coeficiente de aprendizagem demasiado baixo demora demasiado tempo a treinar, daí explicando os piores resultados neste teste, pois as iterações permaneceram iguais.

Em relação ao teste 2 e 3, em que foram alterados os números de neurónios, de 10, para 20 e para 30, respetivamente, a performance da rede melhorou, que passou de apresentar resultados de  $\sim 78\%$  para  $\sim 80.5\%$  e  $\sim 81.5\%$ .

Olhando para os resultados da accuracy, que foi uma média de 0.96% para a rede com 10 neurónios, 0.96% para a rede com 20 neurónios, e 0.963% para a rede com 30 neurónios, podemos dizer que praticamente já não se encontra a aprender, provavelmente a rede encontra-se em overfitting. (a accuracy é sobre o conjunto de teste e não do conjunto de validação, daí não termos a certeza)

No teste 4 foi alterado o tipo de rede para uma feedforwardnet, a performance da rede diminuiu de uma média de 78% para 67%, e o treino da rede foi significativamente mais rápido.

Nos testes 5 e 6 foi alterado o numero de iterações, no teste padrão, foram utilizados 1000, no teste 5, 10 000, e no teste 6, 5 000. Como é de esperar o teste 5 obteve os melhores resultados ( $\sim 83\%$ ), seguido do teste 6.