

Redes Neurais

Relatório do Trabalho Prático – Tema 1

João Oliveira – a21260748
Ana Videira – a21250074
Turma P4

Índice

<i>Introdução</i>	3
<i>Procedimento</i>	3
Tabela 1 - Tabela de funções de Matlab utilizadas	3
<i>Resultados</i>	4
<i>Resultados de Formas_1</i>	4
Treino de Formas_1	4
Algoritmos de Treino de Formas_1	4
F1: Perceptron training rule	4
Tabela 2 – Resultados Obtidos por perceptron training rule em folhas_1.....	4
F1: Gradient Descent	5
Tabela 3 – Resultados Obtidos por gradient descent em folhas_1	5
F1: Stochastic Approximation to Gradient Descent.....	5
Tabela 4 – Resultados Obtidos por stochastic approximation to gradient descent em folhas_1	5
Conclusão para Formas_1	5
<i>Resultados de Formas_2</i>	6
Treino de Formas_2	6
Algoritmos de Treino de Formas_2	6
F2: Perceptron training rule	6
Tabela 5 – Resultados Obtidos por perceptron training rule em folhas_2.....	6
F2: Gradient Descent	6
Tabela 6 – Resultados Obtidos por gradient descent em folhas_2	6
F2: Stochastic Approximation to Gradient Descent.....	7
Tabela 7 – Resultados Obtidos por stochastic approximation to gradient descent em folhas_2	7
Conclusão para Formas_2	7
<i>Resultados de Formas_3</i>	8
Formas_3 – Usando a RN treinada por Formas_2.....	8
Figura 1 - Resultados de Formas_3 com a rede guardada anteriormente	8
Treino para Formas_3	8
Algoritmos de Treino para Formas_3	9
F3: Perceptron training rule	9
Tabela 8 – Resultados Obtidos por perceptron training rule em folhas_3.....	9
F3: Gradient Descent	9
Tabela 9 – Resultados Obtidos por gradient descent em folhas_3	9
F3: Stochastic Approximation to Gradient Descent.....	9
Tabela 10 – Resultados Obtidos por stochastic approximation to gradient descent em folhas_3	10
Conclusão para Formas_3	10
<i>Interface Gráfica</i>	10
Figura 2 - Janela inicial	10
Figura 3 - Opções de configuração de rede neuronal	11
Figura 4 - Output do programa	11
Figura 5 - Exemplo do funcionamento do programa	12
<i>Conclusão</i>	13

Introdução

Com este trabalho prático explora-se a capacidade de uma rede neuronal aprender a classificar formas distintas: círculo, quadrado, triângulo e estrela. Para a realização e exploração deste tema é utilizado o ambiente Matlab com recurso a Excel para tratamento de resultados.

Foram fornecidas três pastas com imagens de várias formas diferentes: Formas_1 (4 imagens), Formas_2 (840 imagens) e Formas_3 (200 imagens). Primeiramente é treinada a rede apenas com as imagens presentes na pasta Formas_1.

Seguidamente é utilizado o conteúdo das pastas Formas_2, a rede é treinada. A rede com melhor desempenho é gravada.

Por fim são utilizadas as imagens e Formas_3 para verificar a classificação da rede neuronal gravada anteriormente e treina-se novamente com Formas_3.

Procedimento

De forma a identificar a forma presente em cada uma das imagens, primeiramente as imagens são tratadas: é atribuído um id a cada uma e são convertidas em matrizes binárias de 32 por 32, onde o valor de 0 corresponde a pixels pretos e o de 2 pixels brancos. Estas matrizes são seguidamente agregadas todas num único vetor.

Assim, o input da rede será os id's das imagens e a matriz vetorizada transposta e o output será ao tipo de figura.

Para conseguir esclarecer o tipo de figura de forma rápida, recorre-se a diferentes funções de Matlab de modo a extrair características relevantes.

Funções	Descrição
<i>ConvexArea</i>	número de pixels contidos em ' <i>ConvexImage</i> ', retornados como escalar
<i>Eccentricity</i>	excentricidade da elipse, retornada como escalar. A excentricidade é o rácio da distancia entre os focos da elipse e o seu maior comprimento de eixos.
<i>Extent</i>	rácio de pixels na região para os pixels totais no retângulo envolvente, retornado como escalar, ou seja, uma área a dividir pela área do retângulo envolvente
<i>EquivDiameter</i>	diâmetro do círculo com a mesma área que a região, retornado como escalar
<i>FilledArea</i>	número de pixels contidos em ' <i>FilledImage</i> ', retornados como escalar
<i>MajorAxisLength</i>	comprimento, em pixels, do maior eixo da elipse, retornado como escalar.
<i>MinorAxisLength</i>	comprimento, em pixels, do menor eixo da elipse, retornado como escalar.
<i>Orientation</i>	ângulo entre o eixo dos xx e o maior eixo da elipse. O valor é dado em graus, variando entre -90 e 90°
<i>Perimeter</i>	distância à volta da fronteira, retornado como escalar
<i>Solidity</i>	proporção de pixels contidos no lado convexo que também fazem parte da região, retornados como escalar.

Tabela 1 - Tabela de funções de Matlab utilizadas

Resultados

De acordo com enunciado do trabalho, é solicitado que se registre os resultados da melhor rede utilizada. Uma das formas de verificar se uma rede cumpre os objetivos propostos, é observando a sua taxa de acerto: Se existe uma boa taxa de acerto, sabe-se que os tipos de figuras foram bem identificados.

Resultados de Formas_1

Treino de Formas_1 : Nesta fase inicial não foram definidos valores para o treino , validação e teste.

Algoritmos de Treino de Formas_1

F1: Perceptron training rule

Com a utilização do algoritmo de *treino perceptron training rule*, a melhor taxa de acerto conseguida (100%) foi obtida com o recurso á função de ativação Step.

A duração do tos testes está exposta pela coluna Tempo na tabela de resultados obtidos com este algoritmo de treino sendo que, o melhor tempo corresponde também á função de ativação responsável pela melhor taxa de acerto. A coluna Epoch fornece-nos o número de épocas bastante variável pois são valores atribuídos automaticamente.

Pasta	Inputs	Camadas	Funções de Ativação		Algoritmo de Treino	Rácio (em %)			Quantidade de Imagens Tratadas	Acerto (%)	Epoch	Tempo (min,seg)
			Função 1	Função 2		Treino	Validação	Teste				
Formas_1	Imagem em binário	10	Step	Step	Perceptron Training Rule	----	----	----	4	100%	0	0.02
Formas_1	Imagem em binário	10	Linear	Linear	Perceptron Training Rule	----	----	----	4	0%	3	0.06
Formas_1	Imagem em binário	10	Sigmoide	Sigmoide	Perceptron Training Rule	----	----	----	4	0%	6	0.14
Formas_1	Imagem em binário	10	TanH	TanH	Perceptron Training Rule	----	----	----	4	0%	6	0.11
Formas_1	Imagem em binário	10	Sinal	Sinal	Perceptron Training Rule	----	----	----	4	0%	0	0

Tabela 2 - Resultados Obtidos por perceptron training rule em folhas_1

F1: Gradient Descent

De acordo com os resultados apresentados na tabela 3, utilizando o algoritmo de treino gradient descent, a função de ativação que fornece uma melhor taxa de acerto é a função tanH.

Imagens	Inputs	Camadas	Funções de Ativação		Algoritmo	Rácio (em %)			Quantidade de Imagens Tratadas	Acerto (%)	Epoch	Tempo (min,seg)
			Função 1	Função 2		Treino	Validação	Teste				
Formas_1	Imagem em binário	10	Step	Step	Gradient Descent	-----	-----	-----	4	0%	0	0
Formas_1	Imagem em binário	10	Linear	Linear	Gradient Descent	-----	-----	-----	4	0%	11	0
Formas_1	Imagem em binário	10	Sigmoide	Sigmoide	Gradient Descent	-----	-----	-----	4	0%	6	0
Formas_1	Imagem em binário	10	TanH	TanH	Gradient Descent	-----	-----	-----	4	100%	6	0
Formas_1	Imagem em binário	10	Sinal	Sinal	Gradient Descent	-----	-----	-----	4	0%	0	0

Tabela 3 - Resultados Obtidos por gradient descent em folhas_1

F1: Stochastic Approximation to Gradient Descent

Por aplicação do algoritmo *stochastic approximation to gradient descent* a maior taxa de acerto registada é o resultado da utilização da função de ativação Sinal. É também devido á aplicação deste algoritmo que resulta a variação do número de épocas e do tempo de duração dos testes.

Imagens	Inputs	Camadas	Funções de Ativação		Algoritmo	Rácio (em %)			Quantidade de Imagens Tratadas	Acerto (%)	Epoch	Tempo (min,seg)
			Função 1	Função 2		Treino	Validação	Teste				
Formas_1	Imagem em binário	10	Step	Step	Stochastic Approximation to Gradient Descent	-----	-----	-----	4	0%	0	0
Formas_1	Imagem em binário	10	Linear	Linear	Stochastic Approximation to Gradient Descent	-----	-----	-----	4	0%	6	0.8
Formas_1	Imagem em binário	10	Sigmoide	Sigmoide	Stochastic Approximation to Gradient Descent	-----	-----	-----	4	0%	9	1.19
Formas_1	Imagem em binário	10	Sinal	Sinal	Stochastic Approximation to Gradient Descent	-----	-----	-----	4	100%	6	0.78

Tabela 4 - Resultados Obtidos por stochastic approximation to gradient descent em folhas_1

Conclusão para Formas_1

Através da análise das imagens em Folhas_1 e da aplicação dos 3 diferente algoritmos , podemos concluir que o número de imagens utilizadas é insuficiente para que a rede aprenda corretamente.

Resultados de Formas_2

Treino de Formas_2 : Não foram definidos valores para o treino , validação e teste.

Algoritmos de Treino de Formas_2

F2: Perceptron training rule

Por aplicação do algoritmo *perceptron training rule* obtêm-se os resultados apresentados pela tabela 5:

- » A função TanH (Tangente Hiperbólica) apresenta uma maior taxa de acerto;
- » O número de épocas apresenta-se muito variável, sendo que o atinge o seu maior valor com a função de ativação Sigmoide;
- » Em relação á duração dos testes, é também a função sigmoide que apresentas maiores valores.

Imagens	Inputs	Camadas	Funções de Ativação		Algoritmo	Rácio (em %)			Quantidade de Imagens Tratadas	Acerto (%)	Epoch	Tempo (min.seg)
			Função 1	Função 2		Treino	Validação	Teste				
Formas_2	Imagem em binário	10	Step	Step	Perceptron Training Rule	70	15	15	804	12.4%	0	0
Formas_2	Imagem em binário	10	Linear	Linear	Perceptron Training Rule	70	15	15	804	99.17%	10	2.12
Formas_2	Imagem em binário	10	Sigmoide	Sigmoide	Perceptron Training Rule	70	15	15	804	76.03%	410	109.7
Formas_2	Imagem em binário	10	TanH	TanH	Perceptron Training Rule	70	15	15	804	100%	19	4.16
Formas_2	Imagem em binário	10	Sinal	Sinal	Perceptron Training Rule	70	15	15	804	26.45%	0	0

Tabela 5 - Resultados Obtidos por perceptron training rule em folhas_2

F2: Gradient Descent

Por aplicação do algoritmo *gradient descent* obtêm-se os resultados apresentados pela tabela 6:

- » A função de ativação TanH (Tangente Hiperbólica) é a que apresenta uma maior taxa de acerto.
- » É também com esta função de ativação que se conseguem a quantidade de épocas mais elevada.
- » Todos estes foram rápidos, com duração de apenas segundos.

Imagens	Inputs	Camadas	Funções de Ativação		Algoritmo	Rácio (em %)			Quantidade de Imagens Tratadas	Acerto (%)	Epoch	Tempo (min.seg)
			Função 1	Função 2		Treino	Validação	Teste				
Formas_2	Imagem em binário	10	Step	Step	Gradient Descent	70	15	15	804	19.83%	0	0
Formas_2	Imagem em binário	10	Linear	Linear	Gradient Descent	70	15	15	804	88.43%	1000	0.05
Formas_2	Imagem em binário	10	Sigmoide	Sigmoide	Gradient Descent	70	15	15	804	62.81%	1000	0.06
Formas_2	Imagem em binário	10	TanH	TanH	Gradient Descent	70	15	15	804	99.17%	1000	0.05
Formas_2	Imagem em binário	10	Sinal	Sinal	Gradient Descent	70	15	15	804	14.05%	0	0

Tabela 6 - Resultados Obtidos por gradient descent em folhas_2

F2: Stochastic Approximation to Gradient Descent

Por aplicação do algoritmo *stochastic approximation gradient descent* obtêm-se os resultados apresentados pela tabela 7:

- » A maior taxa de acerto acontecer com a função de ativação Linear;
- » É também com a função Linear que acontecem os testes com uma maior quantidade de épocas;
- » Com a aplicação deste algoritmo observamos que a duração dos testes aumentou significativamente, em alguns casos supera os 60 minutos;

Imagens	Inputs	Camadas	Funções de Ativação		Algoritmo	Rácio (em %)			Quantidade de Imagens Tratadas	Acerto (%)	Epoch	Tempo (min.seg)
			Função 1	Função 2		Treino	Validação	Teste				
Formas_2	Imagem em binário	10	Step	Step	Stochastic Approximation to Gradient Descent	70	15	15	804	3.31%	0	0
Formas_2	Imagem em binário	10	Linear	Linear	Stochastic Approximation to Gradient Descent	70	15	15	804	99.17%	136	63.91
Formas_2	Imagem em binário	10	Sigmoide	Sigmoide	Stochastic Approximation to Gradient Descent	70	15	15	804	22.31%	5	2.26
Formas_2	Imagem em binário	10	TanH	TanH	Stochastic Approximation to Gradient Descent	70	15	15	804	97.52%	56	28.57
Formas_2	Imagem em binário	10	Sinal	Sinal	Stochastic Approximation to Gradient Descent	70	15	15	804	26.45%	0	0

Tabela 7 - Resultados Obtidos por *stochastic approximation to gradient descent* em folhas_2

Conclusão para Formas_2

Podemos concluir que a melhor rede é a resultante da aplicação do algoritmo *perceptron training rule* por ter a maior taxa de acerto (100%) relativamente a outros algoritmos explorados. Essa é a rede guardada.

Relativamente ao número de épocas e à duração do tempo de testes nos diferentes algoritmos podemos concluir que são altamente variáveis. Deste modo pode-se apenas ressaltar a aplicação do algoritmo *stochastic approximation gradient* fez disparar o tempo dos testes, superando até os 60 minutos de duração,

Resultados de Formas_3

Formas_3 - Usando a RN treinada por Formas_2

Considerando os testes efetuados anteriormente concluiu-se que a melhor rede neuronal é formada pela utilização do algoritmo Perceptron training rule com a função de ativação TanH (tangente Hiperbólica).

Como forma de verificar a veracidade da afirmação anterior, usou-se essa mesma rede para classificar as imagens em Formas_3. Assim, desta atividade resultou uma taxa de acerto de 76% o que podemos considerar uma boa taxa de acerto e consequentemente uma boa rede neuronal.

The screenshot shows the NNTrainTool GUI. In the 'Input' section, the file name is 'Formas_3', activation functions are 'TanH', the training algorithm is 'Perceptron Training Rule', and there are 10 layers. The split ratio is set to 70% for training, 15% for validation, and 15% for testing. The 'Output' section shows a success rate of 76% for 200 images. The 'Treinar Rede' button is highlighted.

Figura 1 - Resultados de Formas_3 com a rede guardada anteriormente

Treino para Formas_3

Nesta fase, a rede é treinada com os parâmetros seguintes: *Treino: 70%* , *Validação: 15%* ; *Teste: 15%* .

Algoritmos de Treino para Formas_3

F3: Perceptron training rule

Por aplicação do algoritmo *perceptron training rule* obtêm-se os resultados apresentados pela tabela 8:

- » A taxa de acerto é 100% para as funções de ativação Linear, Sigmoid e tanH (Tangente Hiperbólica);
- » Tanto a quantidade de épocas como a duração dos testes é superior com a função sigmoide;

Imagens	Inputs	Camadas	Funções de Ativação		Algoritmo	Rácio (em %)			Quantidade de Imagens Tratadas	Acerto	Epoch	Tempo
			Função 1	Função 2		Treino	Validação	Teste				
Formas_3	Imagem em binário	10	Step	Step	Perceptron Training Rule	----	----	----	200	23.33%	0	0
Formas_3	Imagem em binário	10	Linear	Linear	Perceptron Training Rule	----	----	----	200	100%	8	0.57
Formas_3	Imagem em binário	10	Sigmoid	Sigmoid	Perceptron Training Rule	----	----	----	200	100%	128	14.74
Formas_3	Imagem em binário	10	TanH	TanH	Perceptron Training Rule	----	----	----	200	100%	16	1.41
Formas_3	Imagem em binário	10	Sinal	Sinal	Perceptron Training Rule	----	----	----	200	56.67%	0	0

Tabela 8 - Resultados Obtidos por perceptron training rule em folhas_3

F3: Gradient Descent

Por aplicação do algoritmo *gradient descent* obtêm-se os resultados apresentados pela tabela 9:

- » As funções de ativação Linear, Sigmoid e TanH (Tangente Hiperbólica) possuem uma taxa de acerto de 100%;
- » Também com essas funções de ativação conseguimos 1000 épocas com tempos de duração de testes muito pequenos.

Imagens	Inputs	Camadas	Funções de Ativação		Algoritmo	Rácio (em %)			Quantidade de Imagens Tratadas	Acerto (%)	Epoch	Tempo (min,seg)
			Função 1	Função 2		Treino	Validação	Teste				
Formas_3	Imagem em binário	10	Step	Step	Gradient Descent	----	----	----	200	10%	0	0
Formas_3	Imagem em binário	10	Linear	Linear	Gradient Descent	----	----	----	200	100%	1000	0.03
Formas_3	Imagem em binário	10	Sigmoid	Sigmoid	Gradient Descent	----	----	----	200	100%	1000	0.03
Formas_3	Imagem em binário	10	TanH	TanH	Gradient Descent	----	----	----	200	100%	1000	0.03
Formas_3	Imagem em binário	10	Sinal	Sinal	Gradient Descent	----	----	----	200	26.67%	0	0

Tabela 9 - Resultados Obtidos por gradient descent em folhas_3

F3: Stochastic Approximation to Gradient Descent

Por aplicação do algoritmo *stochastic approximation gradient descent* obtêm-se os resultados apresentados pela tabela 10:

- » As funções de ativação Linear e TanH (Tangente Hiperbólica) atingem uma taxa de acerto de 100%;

» Com a função de ativação Linear resultam também o maior número de épocas e a maior duração dos testes.

Imagens	Inputs	Camadas	Funções de Ativação		Algoritmo	Rácio (em %)			Quantidade de Imagens Tratadas	Acerto (%)	Epoch	Tempo (min.sec)
			Função 1	Função 2		Treino	Validação	Teste				
Formas_3	Imagem em binário	10	Step	Step	Stochastic Approximation to Gradient Descent	----	----	----	200	36.67%	0	0
Formas_3	Imagem em binário	10	Linear	Linear	Stochastic Approximation to Gradient Descent	----	----	----	200	100%	119	44.55
Formas_3	Imagem em binário	10	Sigmoide	Sigmoide	Stochastic Approximation to Gradient Descent	----	----	----	200	43.33%	20	7.73
Formas_3	Imagem em binário	10	TanH	TanH	Stochastic Approximation to Gradient Descent	----	----	----	200	100%	25	11.17
Formas_3	Imagem em binário	10	Sinal	Sinal	Stochastic Approximation to Gradient Descent	----	----	----	200	40%	0	0

Tabela 10 - Resultados Obtidos por stochastic approximation to gradient descent em folhas_3

Conclusão para Formas_3

Utilizando as imagens em Formas_3 (200 imagens) testa-se verdadeiramente a fidelidade da rede obtida e gravadas dos treino anteriores efetuados. Através da rede gravada anteriormente, foram registadas taxas de acerto si

Interface Gráfica

Ao iniciar o programa, abre-se uma janela com o seguinte aspeto:

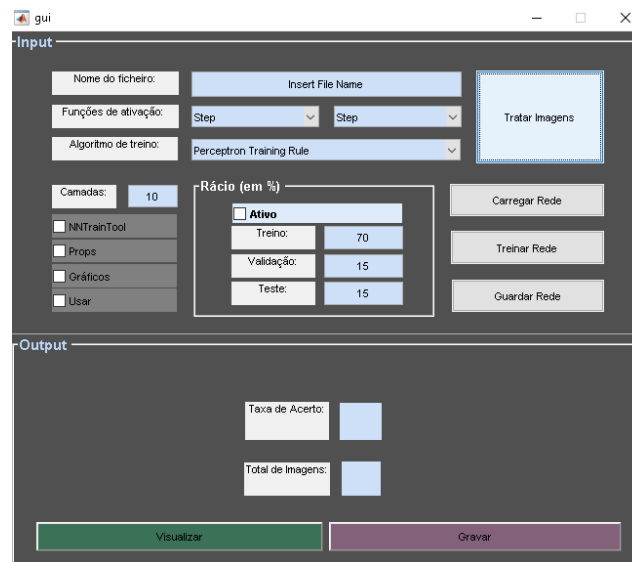


Figura 2 - Janela inicial

Nesta janela estão presentes todas as opções para a configuração do treino ou uso de uma rede neuronal.

Na parte superior da janela estão presentes as opções de input: Campos onde se podem inserir o nome da pasta a trabalhar, escolher as funções de ativação desejadas entre outros apresentados na imagem.

O botão “Tratar Imagens” recolhe as imagens da pasta seleccionada e transforma-as numa matriz binária pronta a ser utilizada. O botão “Carregar Rede” carrega a última rede guardada para memória de forma a que fique pronta para ser usada ou treinada. Finalmente, os botões “Treinar Rede” e “Guardar Rede”: o primeiro treina a rede com as características inseridas anteriormente e o segundo guarda essa mesma rede.

The 'Input' window contains the following elements:

- Nome do ficheiro:** A text input field with the placeholder 'Insert File Name'.
- Funções de ativação:** Two dropdown menus, both currently set to 'Step'.
- Algoritmo de treino:** A dropdown menu set to 'Perceptron Training Rule'.
- Camadas:** A numeric input field set to '10'.
- Checkboxes:** A vertical list of checkboxes labeled 'NNTrainTool', 'Props', 'Gráficos', and 'Usar', all of which are currently unchecked.
- Rácio (em %):** A section containing a checkbox labeled 'Ativo' (unchecked) and a table for data distribution:

Treino:	70
Validação:	15
Teste:	15
- Buttons:** A vertical stack of buttons on the right side labeled 'Tratar Imagens', 'Carregar Rede', 'Treinar Rede', and 'Guardar Rede'.

Figura 3 - Opções de configuração de rede neuronal

Na parte superior da janela estão presentes as opções de output: é apresentada a taxa de acerto da rede neuronal e o total de imagens tratadas. Possui ainda 2 botões, “Visualizar”, que abre a tabela de dados das redes gravadas, e “Gravar” que grava esses dados.

The 'Output' window contains the following elements:

- Taxa de Acerto:** A label next to a light blue rectangular display area.
- Total de Imagens:** A label next to another light blue rectangular display area.
- Buttons:** Two large buttons at the bottom, 'Visualizar' (green) and 'Gravar' (purple).

Figura 4 - Output do programa

A imagem seguinte expõe um exemplo de treino de uma rede neuronal.

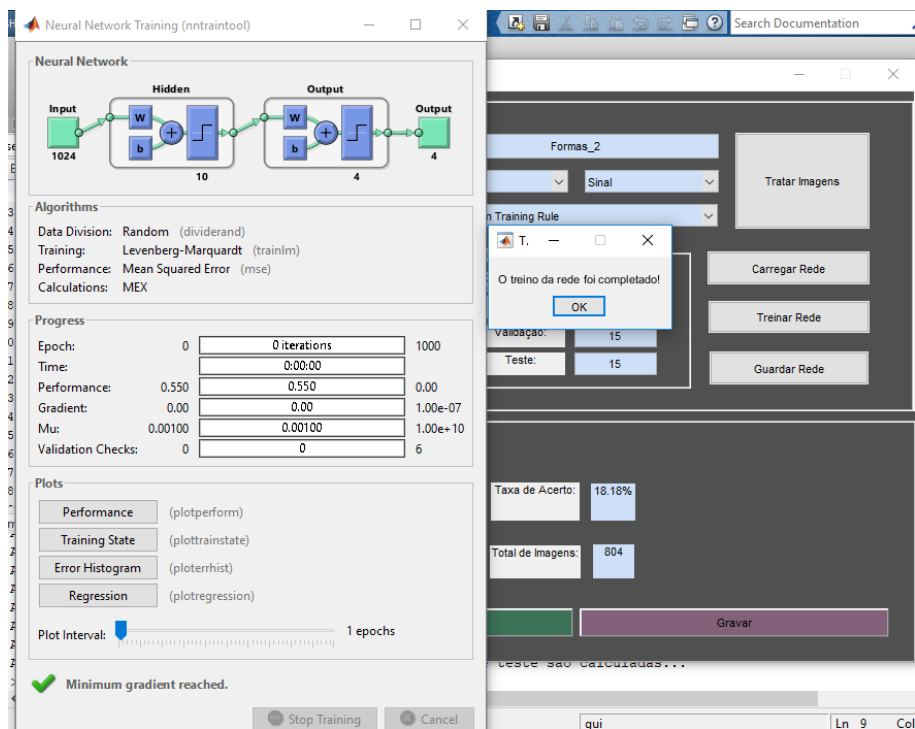


Figura 5 - Exemplo do funcionamento do programa

Conclusão

Uma boa rede neuronal está associada ao cumprimento dos objetivos propostos para a mesma. Uma das formas de se poder verificar o cumprimento desses objetivos é através da percentagem da taxa de acerto sendo que, uma alta percentagem é equivalente a uma boa rede neuronal.

Través do treino das imagens em Formas_1 podemos concluir que apenas 4 imagens é uma quantidade inferior á necessária para treinar uma rede. Com um número tão reduzido de imagens não é possível treinar uma rede eficiente.

Em relação às imagens em Formas_2 , o treino e testes destas, com recurso aos 3 diferente algoritmos e 4 diferentes funções de ativação , revela um par algoritmo – função de ativação relevante: *perceptron training rule* e função de ativação TanH (tangente Hiperbólica). É com este par : *perceptron training rule* e função de ativação TanH que se consegue a melhor taxa de acerto (100%) .

É também ainda de ressaltar que a utilização do algoritmo *stochastic approximation gradient* ,com as imagens em Formas_2, fez disparar o tempo dos testes, superando até os 60 minutos de duração.

Encontrada a melhor rede neuronal com Formas_2, utilizaram-se a imagens em Formas_3 para verificar a fidelidade da rede guardada. Através da mesma rede, mas com recurso a Formas_3 foi conseguido uma taxa de acerto de 76% , uma taxa de acerto significativamente alta.