

Conhecimento e Raciocínio

TURMA P1

Relatório de Trabalho Prático

REDES NEURONAIS

Elaborado por:

Ana Filipa Costa Farinha Alves

Ricardo Diamantino Oliveira Belinha

COIMBRA, julho de 2018

Conhecimento e Raciocínio

TURMA P1

Relatório de Trabalho Prático

REDES NEURONAIS

Elaborado por:

Ana Filipa Costa Farinha Alves, n.º 21190240

Ricardo Diamantino Oliveira Belinha, n.º 21250197

Professora Orientadora:

Anabela Simões

COIMBRA, julho de 2018

ÍNDICE

Índice de Figuras	3
Índice de Tabelas.....	4
1. INTRODUÇÃO	5
2. REDES NEURONAIIS	5
2.1. Introdução ao trabalho efetuado	10
2.2. Espécie	11
2.3. Descrição de procedimento	12
2.4. Características	13
2.5. Subespécie.....	13
3. RESULTADOS	14
3.1. Folhas_1: dados de treino.....	15
3.1.1. Perceptron training rule	15
3.1.2. Gradient Descent	15
3.1.3. Stochastic Approximation to Gradient Descent	16
3.1.4. Conclusão para Folhas_1	17
3.2. Folhas_2: dados de validação.....	18
3.2.1. Perceptron training rule	18
3.2.2. Gradient Descent	19
3.2.3. Stochastic Approximation to Gradient Descent	19
3.2.4. Conclusão para Folhas_2	20
3.3. Comparação de taxas de acerto	20
3.4. Folhas_3: dados de teste.....	21
3.5. Folhas personalizadas.....	21
3.6. Quantidade de neurónios variável	23
4. INTERFACE GRÁFICA	24
4.1. Autores	24

4.2.	Relatório	24
4.3.	Programa	25
4.4.	Exemplos de interface no decorrer de testes	27
5.	CONCLUSÃO	30
6.	BIBLIOGRAFIA.....	31
	ANEXOS.....	32
	Anexo A	33
	ID's correspondentes às espécies	33
	Anexo B	34
	ID's correspondentes às subespécies.....	34

ÍNDICE DE FIGURAS

Figura 1 Perceptrão ou neurónio artificial (Santos, s.d.).....	5
Figura 2 Função de ativação step (Marques).....	6
Figura 3 Função de ativação linear (Marques).....	7
Figura 4 Função de ativação sigmoide (Marques).	7
Figura 5 Função de ativação tangente hiperbólica (Marques).....	7
Figura 6 Função de ativação sinal (Marques).	7
Figura 7 Rede neuronal	8
Figura 8 Rede neuronal: input, hidden e output layers.....	8
Figura 9 Treino, validação e teste (Moura, 2016).	9
Figura 10 Esquema de funcionamento de uma rede neuronal, com treino supervisionado (Santos, s.d.).....	10
Figura 11 Rede neuronal com treino não supervisionado (Cortez, 2015).	10
Figura 12 Espécies de folhas, onde é evidente a maioria de folhas da espécie Quercus.	11
Figura 13 Imagem de folha de exemplo.....	12
Figura 14 Matriz com vetor de imagens nas linhas e ID de espécie nas colunas.....	12
Figura 15 Rede neuronal de espécies.	12
Figura 16 Folhas personalizadas.	22
Figura 17 Interface - janela inicial.	24
Figura 18 Interface - autores.	24
Figura 19 Interface - inserir imagem e seu tratamento.....	25
Figura 20 Interface - escolha de funções de ativação.....	25
Figura 21 Interface - algoritmos de treino da rede.	25
Figura 22 Interface - percentagem atribuída a treino, validação e teste.	25
Figura 23 Interface - output.....	26
Figura 24 Resultado de treino de rede utilizando a função tangente hiperbólica.....	27
Figura 25 Teste efetuado para a pasta Folhas_2.....	27
Figura 26 Histograma de erros para um dos testes efetuados.	28
Figura 27 Matriz de confusão para espécies fornecida pelo Matlab, com taxa de acerto de 60,96%.	28
Figura 28 Matriz de confusão para espécies fornecida pelo Matlab, com taxa de acerto de 22,92%.	29

ÍNDICE DE TABELAS

Tabela 1 Funções de Matlab utilizadas para extrair características (Matlab, s.d.).....	13
Tabela 2 Subespécies de Quercus e sua quantidade de exemplares.	14
Tabela 3 Resultados obtidos para perceptron training rule – Folhas_1.....	15
Tabela 4 Resultados obtidos gradient descet.....	16
Tabela 5 Resultados obtidos para Stochastic Approximation to Gradient Descent.	17
Tabela 6 Resultados obtidos para perceptron training rule – Folhas_2.....	18
Tabela 7 Resultados obtidos para Gradient Descent – Folhas_2.	19
Tabela 8 Resultados obtidos para Stochastic Approximation to Gradient Descent – Folhas_2..	19
Tabela 9 Resumo das melhores taxas de acerto nos testes realizados.....	21
Tabela 10 Resultados de Folhas_3.	21
Tabela 11 Resultados de testes efetuados a Folhas personalizadas.....	22
Tabela 12 Testes com diferente quantidade de neurónios.....	23
Tabela 13 Resultado das redes gravadas.	26

1. INTRODUÇÃO

O trabalho apresentado resulta duma proposta da unidade curricular de Conhecimento e Raciocínio, do 2º ano da Licenciatura em Engenharia Informática.

Com este pretende-se que se aprofundem os conhecimentos sobre redes neuronais. O objetivo consiste na implementação de uma rede neuronal capaz de classificar corretamente um conjunto de folhas pertencentes a 34 espécies diferentes, com 98 subespécies dado que, uma das espécies fornecidas (*Phildelphus*), não possui subespécie associada.

Na consecução deste projeto foi utilizado Matlab, através das ferramentas guide para desenvolvimento de interface gráfica e através do programa em si, além de se ter recorrido ao Excel. Baseou-se grandemente no dataset da iris, publicado por Fisher, o qual contém 3 classes, cada qual com 50 instâncias referentes ao tipo de flor iris. O dataset de Fisher possui ma das classes facilmente separável das restantes duas, e a saída prevista da rede corresponde à classe da flor iris (setosa, versicolour ou virginica). Tem quatro entradas na rede: o comprimento e largura da sépala e o comprimento e largura da pétala.

2. REDES NEURONAIS

Uma rede neuronal é constituída por neurónios, onde cada um desses neurónios corresponde a uma unidade de processamento de informação. À informação que entra na rede, (atributos ou características do problema) chama-se input; à informação resultante do processamento, dá-se o nome de output (aquilo que se pretende atingir). Nas redes neuronais, o processamento de informação está distribuído por um grande número de unidades interligadas. Cada neurónio é estimulado por uma ou mais ligações vindas de outros neurónios, chamadas sinapses e este sinal é propagado ao longo do sistema indo, por sua vez, estimular outros neurónios (Menezes, s.d.). Nas redes neuronais artificiais os neurónios têm o nome de perceptrões (Figura 1).

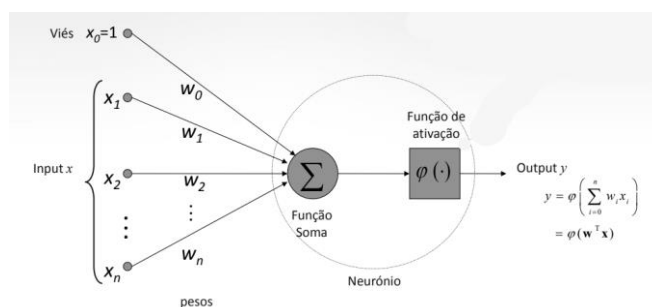


Figura 1 Perceptrão ou neurónio artificial (Santos, s.d.).

A saída do perceptron, resulta da sua função ou algoritmo de treino, do peso atribuído e da função de ativação. Os algoritmos de treino podem ser:

1. *Perceptron Training Rule* — os coeficientes w são inicializados com valores aleatórios.
2. *Gradient Descent* — inicializa os coeficientes w aleatoriamente, aplica-os a todos os exemplos de treino à unidade, calcula o erro gerado para cada coeficiente w , e volta a recalcular o erro até este ser suficientemente baixo.
3. *Stochastic Approximation to Gradient Descent* — o valor de correção de cada coeficiente w_i é calculado logo após a apresentação de um só exemplo, em vez de se somarem os erros de todos os exemplos como no *Gradient Descent*.

Ao processo de ajustamento de pesos dá-se o nome de época: uma época é quando um dataset inteiro passa pela rede neuronal para a frente e para trás uma só vez (Sharma, 2017). À medida que o número de épocas aumenta, o número de vezes que os pesos w são ajustados e alterados também aumenta. Assim, quanto mais épocas existirem no treino de uma rede, mais vezes os pesos foram ajustados.

Resumidamente, em qualquer um destes algoritmos de treino (Marques):

- Ao perceptron ou unidade são aplicadas entradas que constituem exemplos de treino.
- A cada um corresponde um valor alvo (*target*) que se compara com um resultado: a saída apresentada pelo perceptron ou unidade linear
- Se este resultado gerar uma classificação errada, reajustam-se os coeficientes sinápticos ou pesos, w
- Terminado o treino, em presença de exemplos diferentes dos utilizados, o perceptron ou unidade deverá responder corretamente.

As funções de treino podem ser (Marques):

- Step — Figura 2.

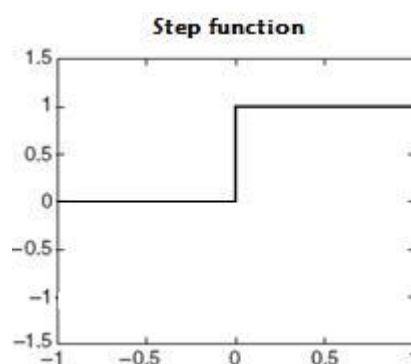


Figura 2 Função de ativação step (Marques).

- Linear — Figura 3.

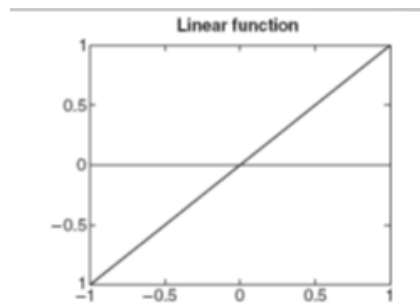


Figura 3 Função de ativação linear (Marques).

- Sigmoide — Figura 4.

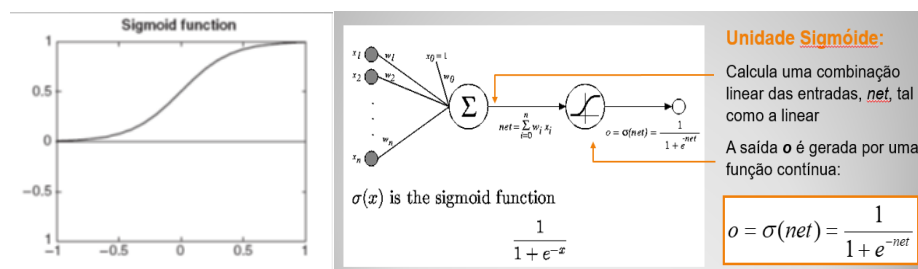


Figura 4 Função de ativação sigmoide (Marques).

- Tangente hiperbólica — Figura 5. Substitui a função sigmoide por tangente hiperbólica.

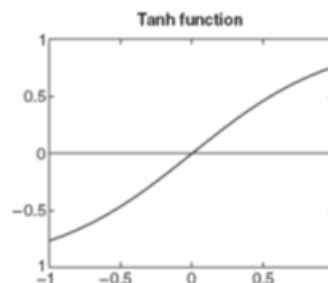


Figura 5 Função de ativação tangente hiperbólica (Marques).

- Sinal — Figura 6.

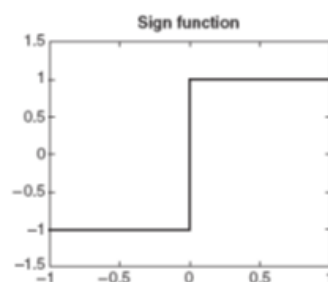


Figura 6 Função de ativação sinal (Marques).

Um conjunto de perceptrões forma uma rede neuronal, sendo o perceptrão multicamada uma rede *feed-forward* (Figura 7). Segundo Santos (s.d.) a rede *feed-forward* consiste num grafo acíclico, direcionado (geralmente completamente ligado), em que as entradas de cada camada são as saídas da camada anterior.



Figura 7 Rede neuronal

Os perceptrões e as unidades lineares apenas podem representar superfícies de decisão lineares, dando resultados como “positivo” e “negativo” (Marques). As redes neuronais multinível, treinadas pelo *BackPropagation Algorithm*, ajustam os pesos das redes da frente para trás, permitem treinar redes neuronais em multicamada, pelo que podem representar superfícies de decisão de forma muito variadas.

A rede neuronal é, então, composta por unidades organizadas em camadas, geralmente 2 ou 3, sendo as primeiras unidades as de input, que não constituem uma camada (Figura 8). As unidades internas compõem a(s) camada(s) interna(s) — ou *hidden layers* —, e as unidades de saída são as *output layer*.

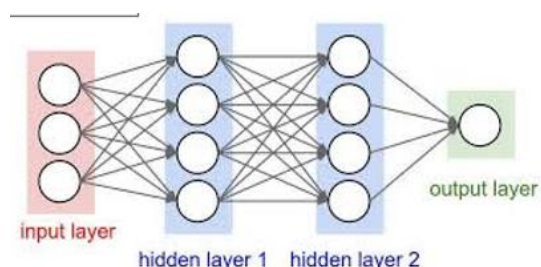


Figura 8 Rede neuronal: input, hidden e output layers.

A utilidade das redes neuronais está diretamente relacionada com a sua capacidade de aprender a classificar, i.e., distinguir imagens, caracteres, sons, etc. sendo que, a cada classificação corresponde uma saída ou combinação de saídas distintas (Marques). Para que exista esta utilidade a elas inerente, são necessárias três fases, feitas de forma automática pelo Matlab mediante a utilização de funções específicas. Por *default*, o programa atribui-lhes valores aleatórios:

- Aprendizagem – geralmente, é um valor de 70%, quando definido pelo utilizador.
- Validação – habitualmente, cerca de 15%.
- Teste – restantes 15%.

É necessário conhecer estas métricas de percentagens para se saber o quão bom o sistema é, tendo, assim, um termo de comparação. Quando se predefinem os valores do treino, validação e teste, tem-se um processo de *hold-out validation*, onde se criam amostras, sendo geralmente utilizado para se ter mais dados e se conseguir comprovar que as amostras criadas têm significância estatística para representar a população (Moura, 2016).

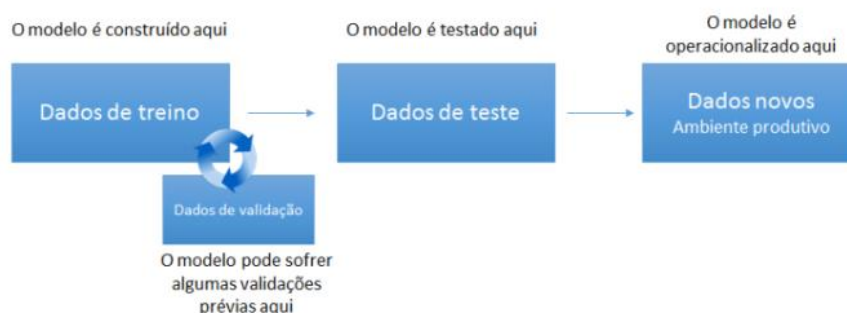


Figura 9 Treino, validação e teste (Moura, 2016).

De acordo com (Moura, 2016), se um modelo for testado com os dados utilizados no treino, ou seja, os mesmos com que foi construído, o seu comportamento não pode ser generalizado, por se desconhecer qual o seu comportamento com dados nunca vistos e, para isso, são utilizados os dados de teste (subcapítulo Folhas_3: dados de teste). Enquanto se treina a rede, de acordo com o mesmo autor (Moura, 2016), podem ser divididos os dados do treino em vários conjuntos de dados: os dados de validação, utilizados para realizar as validações iniciais durante a aprendizagem da rede (subcapítulo Folhas_2: dados de validação). Ao longo de todo o processo, treino, validação e teste, deve-se manter a homogeneidade de dados, aplicáveis ao problema concreto que se tem: se se trabalha com imagens de folhas no início, deve-se trabalhar ao longo de todo o processo com folhas (Moura, 2016).

A aprendizagem pode ser supervisionada, não supervisionada ou por *reinforcement learning* (i.e. aproximadamente “por recompensa”) (Marques). É feita por alteração dos coeficientes sinápticos (pesos, w_n) através de um algoritmo chamado *backpropagation*. O treino supervisionado faz-se por aplicação de exemplos previamente classificados (i.e. em que as entradas e as saídas pretendidas são previamente conhecidas) (Figura 10). Segundo Cortez (2015) o objetivo é ajustar os parâmetros da rede, de forma a encontrar uma função que efetue o mapeamento entre os pares, de entrada e saída, fornecidos, ajustando os parâmetros da rede, de forma a encontrar uma função que efetue o mapeamento entre os pares, de entrada e saída, fornecidos (Cortez, 2015).

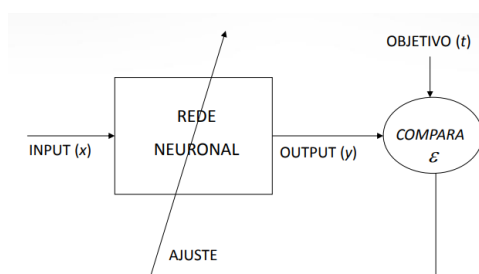


Figura 10 Esquema de funcionamento de uma rede neuronal, com treino supervisionado (Santos, s.d.).

Segundo este mesmo autor, Cortez (2015), existem dois tipos de treino supervisionado: o offline e o online. No treino offline os dados de conjunto de treino não alteram, e uma vez obtida a solução para a rede, esta deve permanecer fixa Cortez (2015). Se surgirem novos dados adicionados ao conjunto de treino, um novo treino envolvendo também os dados anteriores deve ser realizado para se evitar interferências no treino anterior (Cortez, 2015). No treino não supervisionado a aprendizagem é feita através das descobertas dos padrões nos dados de entrada. Nem sempre, no final de um treino, a rede fica capaz de resolver os problemas corretamente e nesta situação a aprendizagem não foi bem conseguida.

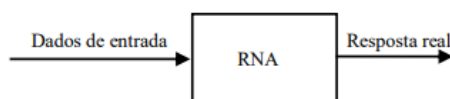


Figura 11 Rede neuronal com treino não supervisionado (Cortez, 2015).

2.1. Introdução ao trabalho efetuado

Foram fornecidas três pastas com imagens de folhas a preto e branco:

- Folhas 1: com 99 imagens;
- Folhas 2: com 1564 imagens;
- Folhas 3: com 20 imagens.

Ao mesmo tempo, foi também fornecida uma tabela onde, a cada id das imagens, correspondia uma espécie e subespécie. No enunciado do trabalho é solicitado para registar os resultados da melhor rede, pelo que lhe é um tipo de treino supervisionado. Inicialmente, treinou-se a rede apenas com os dados da Folhas_1 e, posteriormente, adicionaram-se os dados da Folhas_2 para validação do treino. Numa fase posterior, verificam-se os resultados, utilizando as imagens contidas em Folhas_3 e outras que não foram previamente fornecidas, incluindo, folhas a cores.

2.2. Espécie

Da Figura 12 conclui-se facilmente, que existe uma maioria de folhas da espécie *Quercus*, pelo que, à priori, a rede tenderá a reconhecer as folhas desta espécie mais facilmente.

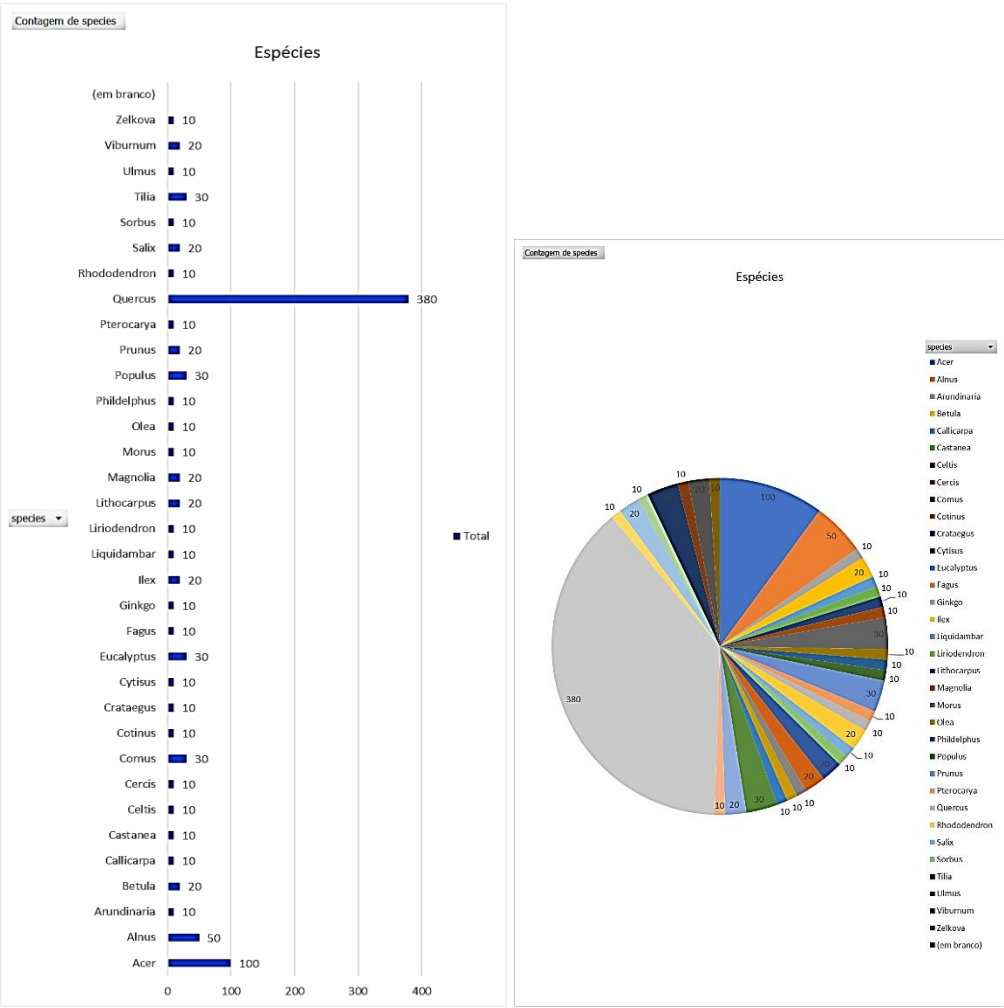


Figura 12 Espécies de folhas, onde é evidente a maioria de folhas da espécie *Quercus*.

2.3. Descrição de procedimento

Para identificar a espécie é necessário processar a informação de entrada. Face a isto, procedeu-se a atribuir um id específico a cada espécie (Anexo A). De seguida, converteram-se as imagens em formato .jpg (Figura 13) para matrizes binárias, onde o valor 0 corresponde ao pixéis pretos e os brancos ao valor 1.



Figura 13 Imagem de folha de exemplo.

Como as imagens de folhas são de grandes dimensões, reduziram-se para matrizes binárias de 32 por 32. Agregaram-se todas estas matrizes num só vetor que serve, então, de entrada para a rede: os ID's de espécie serão os inputs da rede e a saída da rede será a espécie (Figura 14). De seguida, faz-se a transposta da matriz de imagens vetorizadas para que possam entrar como input da rede (Figura 15).

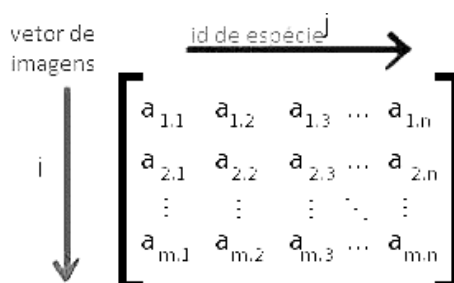


Figura 14 Matriz com vetor de imagens nas linhas e ID de espécie nas colunas.

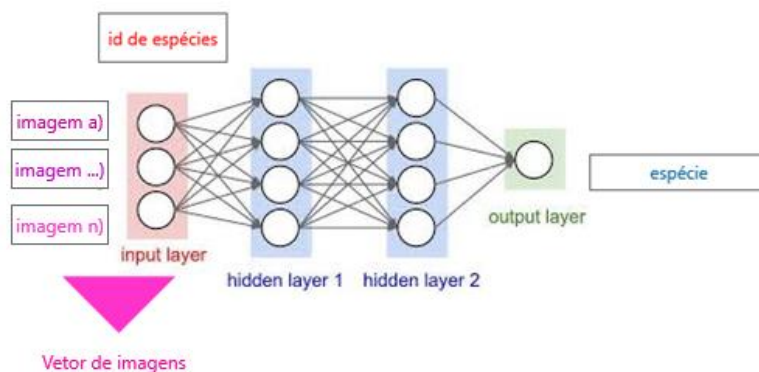


Figura 15 Rede neuronal de espécies.

2.4. Características

A utilização de extração de características permite que a rede encontre a espécie e subespécie a que pertence de modo rápido. As características utilizadas para extrair dados das folhas foram as presentes na Tabela 1.

Tabela 1 Funções de Matlab utilizadas para extrair características (Matlab, s.d.).

Funções	Descrição
<i>ConvexArea</i>	número de pixéis 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 pixéis na região para os pixéis 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 pixéis contidos em ' <i>FilledImage</i> ', retornados como escalar
<i>MajorAxisLength</i>	comprimento, em pixéis, do maior eixo da elipse, retornado como escalar.
<i>MinorAxisLength</i>	comprimento, em pixéis, 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 pixéis contidos no lado convexo que também fazem parte da região, retornados como escalar.

2.5. Subespécie

Para obtenção da subespécie, utilizou-se o mesmo procedimento, criando-se uma nova rede neuronal. Desta vez, entra na rede um vetor com as imagens em binário e como targets os ID's das imagens e o ID respetivo à sua subespécie (Anexo B). As imagens são processadas do mesmo modo, obtendo-se, assim, a subespécie correspondente à imagem(ns).

3.1. Folhas_1: dados de treino

Nesta fase não foram definidos valores para treino, validação e teste. Por esta razão, o Matlab atribuiu-lhe percentagens aleatórias.

3.1.1. Perceptron training rule

Utilizando o algoritmo de perceptron training rule, obtêm-se os resultados apresentados na Tabela 3. Desta se conclui que:

- As **funções de ativação** com maior taxa de acerto para as espécies são as que possuem a função step numa das camadas (taxa de acerto de 47%, 40% e novamente, 40%). A função sigmoide também obteve bons resultados, tendo-se registado 33% e 40%. Utilizando a função sigmoide, registou-se num dos testes efetuados, uma taxa de acerto de 47%. Para as subespécies, e como seria de esperar, as taxas de acerto foram bastante inferiores em relação às das espécies: a maioria registou taxa de acerto igual a 0. Contudo, nas que registaram valores superiores a 0, tinham a função sigmoide em comum na função de ativação.
- A **duração do teste**, dada na coluna Tempo, registou tempos muito bons, inclusive para as redes com melhores taxas de acerto.
- O **número de épocas**, dada na coluna Epoch, varia bastante, sendo este valor atribuído automaticamente pelo Matlab.

Tabela 3 Resultados obtidos para perceptron training rule – Folhas_1.

Nome do Ficheiro	Inputs	Camadas escondidas	Funções de ativação		Topologia da rede	Rácio (em)			Quantidade de imagens tratadas	Espécies			Subespécies		
			1ª Função	2ª Função		Treino	Validação	Teste		Taxa de acerto (%)	Epoch	Tempo (minutos)	Taxa de acerto (%)	Epoch	Tempo (minutos)
Folhas_1	Imagem em binário	10	Step	Step	Perceptron Training Rule	-----	-----	-----	99	47	5	1,05	0	5	3,01
Folhas_1	Imagem em binário	10	Step	Linear	Perceptron Training Rule	-----	-----	-----	99	40	5	1,11	0	5	3,08
Folhas_1	Imagem em binário	10	Step	Sigmoide	Perceptron Training Rule	-----	-----	-----	99	27	8	1,79	7	7	4,17
Folhas_1	Imagem em binário	10	Step	TanH	Perceptron Training Rule	-----	-----	-----	99	40	9	2,14	0	18	12,06
Folhas_1	Imagem em binário	10	Step	Sinal	Perceptron Training Rule	-----	-----	-----	99	7	0	0,00	0	0	0,00
Folhas_1	Imagem em binário	10	Linear	Linear	Perceptron Training Rule	-----	-----	-----	99	13	9	2,50	0	9	6,14
Folhas_1	Imagem em binário	10	Linear	Sigmoide	Perceptron Training Rule	-----	-----	-----	99	33	14	5,29	0	18	17,47
Folhas_1	Imagem em binário	10	Linear	TanH	Perceptron Training Rule	-----	-----	-----	99	0	31	9,84	0	30	23,88
Folhas_1	Imagem em binário	10	Linear	Sinal	Perceptron Training Rule	-----	-----	-----	99	0	0	0,00	0	0	0,00
Folhas_1	Imagem em binário	10	Sigmoide	Sigmoide	Perceptron Training Rule	-----	-----	-----	99	40	8	2,51	0	9	6,68
Folhas_1	Imagem em binário	10	Sigmoide	TanH	Perceptron Training Rule	-----	-----	-----	99	27	12	3,81	0	14	11,27
Folhas_1	Imagem em binário	10	Sigmoide	Sinal	Perceptron Training Rule	-----	-----	-----	99	0	0	0,00	7	0	0,00
Folhas_1	Imagem em binário	10	TanH	TanH	Perceptron Training Rule	-----	-----	-----	99	47	12	3,84	0	9	7,34
Folhas_1	Imagem em binário	10	TanH	Sinal	Perceptron Training Rule	-----	-----	-----	99	0	0	0,00	0	0	0,00
Folhas_1	Imagem em binário	10	Sinal	Sinal	Perceptron Training Rule	-----	-----	-----	99	27	4	0,95	0	5	3,52

3.1.2. Gradient Descent

Utilizando o algoritmo de *Gradient Descent*, obtêm-se os resultados apresentados na Tabela 4. Desta se conclui que:

- Nada se pode concluir em relação às **funções de ativação** para este algoritmo, uma vez que não apresentam funções comuns de certa forma recorrentes nos resultados. Poder-se-ia atribuir a taxa de acerto na espécie de 7% e 20% à presença da função sinal, contudo, só em metade dos casos, é que apresenta valores superiores a 0%. Utilizando este algoritmo, as taxas de acerto são francamente baixas tanto para as espécies como para subespécies.
- A **duração do teste**, registou, mais uma vez, tempos muito bons, inferiores a 1 minuto tanto para a rede de espécies como para subespécies.
- O **número de épocas** foi extremamente elevado, registando 1000 iterações tanto na rede de espécies como de subespécies.

Tabela 4 Resultados obtidos gradient descent.

Nome do Ficheiro	Inputs	Camadas escondidas	Funções de ativação		Topologia da rede	Rácio (em)			Quantidade de imagens tratadas	Espécies			Subespécies		
			1ª Função	2ª Função		Treino	Validação	Teste		Taxa de acerto (%)	Epoch	Tempo (minutos)	Taxa de acerto (%)	Epoch	Tempo (minutos)
Folhas_1	Imagem em binário	10	Step	Step	Gradient Descent	-----	-----	-----	99	0	1000	0,02	0	1000	0,03
Folhas_1	Imagem em binário	10	Step	Linear	Gradient Descent	-----	-----	-----	99	0	1000	0,02	0	1000	0,03
Folhas_1	Imagem em binário	10	Step	Sigmoide	Gradient Descent	-----	-----	-----	99	0	1000	0,02	0	1000	0,03
Folhas_1	Imagem em binário	10	Step	TanH	Gradient Descent	-----	-----	-----	99	0	1000	0,02	0	1000	0,03
Folhas_1	Imagem em binário	10	Step	Sinal	Gradient Descent	-----	-----	-----	99	0	0	0,00	0	0	0,00
Folhas_1	Imagem em binário	10	Linear	Linear	Gradient Descent	-----	-----	-----	99	0	1000	0,02	0	1000	0,03
Folhas_1	Imagem em binário	10	Linear	Sigmoide	Gradient Descent	-----	-----	-----	99	0	1000	0,02	0	1000	0,03
Folhas_1	Imagem em binário	10	Linear	TanH	Gradient Descent	-----	-----	-----	99	7	1000	0,02	0	1000	0,03
Folhas_1	Imagem em binário	10	Linear	Sinal	Gradient Descent	-----	-----	-----	99	7	0	0,00	7	0	0,00
Folhas_1	Imagem em binário	10	Sigmoide	Sigmoide	Gradient Descent	-----	-----	-----	99	0	1000	0,02	0	1000	0,03
Folhas_1	Imagem em binário	10	Sigmoide	TanH	Gradient Descent	-----	-----	-----	99	0	1000	0,02	0	1000	0,03
Folhas_1	Imagem em binário	10	Sigmoide	Sinal	Gradient Descent	-----	-----	-----	99	20	0	0,00	0	0	0,00
Folhas_1	Imagem em binário	10	TanH	TanH	Gradient Descent	-----	-----	-----	99	0	1000	0,02	0	1000	0,03
Folhas_1	Imagem em binário	10	TanH	Sinal	Gradient Descent	-----	-----	-----	99	0	0	0,00	7	0	0,00
Folhas_1	Imagem em binário	10	Sinal	Sinal	Gradient Descent	-----	-----	-----	99	7	1000	0,02	0	1000	0,03

3.1.3. Stochastic Approximation to Gradient Descent

Utilizando o algoritmo de *Stochastic Approximation to Gradient Descent*, obtêm-se os resultados apresentados na Tabela 5. Desta se conclui que:

- As **funções de ativação** mais indicadas são a sigmoide, com valores de taxas de acerto de 60 e 47%. A função step, aliada à linear, também registou um valor bom de taxa de acerto para a espécie, de 53%. Apesar dos bons valores verificados na rede de espécie, não se traduziram igualmente em resultados positivos para a rede das subespécies, onde a taxa de acerto foi sempre 0, excetuando quando foi utilizada a função de tangente hiperbólica e sinal, com taxa de acerto para subespécie de apenas, 7%.
- A **duração do teste**, registou tempos elevados, comparativamente às anteriores: tendo a rede de subespécie registado um tempo de 70 minutos: 1h 10m.

- O **número de épocas** também neste algoritmo é bastante elevado, registando-se um teste com 90 iterações no treino da rede de espécie e outro de 103, no treino da rede de subespécie.

Tabela 5 Resultados obtidos para *Stochastic Approximation to Gradient Descent*.

Nome do Ficheiro	Inputs	Camadas escondidas	Funções de ativação		Topologia da rede	Rácio (em)			Quantidade de imagens tratadas	Espécies			Subespécies		
			1ª Função	2ª Função		Treino	Validação	Teste		Taxa de acerto (%)	Epoch	Tempo (minutos)	Taxa de acerto (%)	Epoch	Tempo (minutos)
Folhas_1	Imagem em binário	10	Step	Step	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	27	35	19,10	0	22	14,81
Folhas_1	Imagem em binário	10	Step	Linear	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	53	31	13,90	0	22	12,46
Folhas_1	Imagem em binário	10	Step	Sigmoide	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	0	3	1,58	0	3	1,99
Folhas_1	Imagem em binário	10	Step	TanH	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	13	3	1,43	0	3	2,05
Folhas_1	Imagem em binário	10	Step	Sinal	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	0	0	0,00	0	0	0,00
Folhas_1	Imagem em binário	10	Linear	Linear	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	27	90	42,16	0	103	70,16
Folhas_1	Imagem em binário	10	Linear	Sigmoide	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	0	9	4,45	0	3	2,17
Folhas_1	Imagem em binário	10	Linear	TanH	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	0	7	3,48	0	9	5,99
Folhas_1	Imagem em binário	10	Linear	Sinal	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	0	0	0,00	0	0	0,00
Folhas_1	Imagem em binário	10	Sigmoide	Sigmoide	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	60	26	14,16	0	19	12,75
Folhas_1	Imagem em binário	10	Sigmoide	TanH	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	47	7	4,16	0	11	7,87
Folhas_1	Imagem em binário	10	Sigmoide	Sinal	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	7	0	0,00	0	0	0,00
Folhas_1	Imagem em binário	10	TanH	TanH	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	20	23	11,96	0	31	20,30
Folhas_1	Imagem em binário	10	TanH	Sinal	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	0	0	0,00	7	0	0,00
Folhas_1	Imagem em binário	10	Sinal	Sinal	Stochastic Approximation to Gradient Descent	-----	-----	-----	99	20	42	21,19	0	43	26,65

3.1.4. Conclusão para Folhas_1

- O algoritmo que regista maior **taxa de acerto** na rede de espécie é o *Perceptron Training Rule*. O algoritmo de *Stochastic Approximation to Gradient Descent* também revelou valores muito bons de taxa de acerto, sendo, até superiores aos do *Perceptron Training Rule*. O *Gradient Descent* revelou-se bastante mau nos testes realizados.
- Em termos de **número de épocas**, o *Perceptron Training Rule* mais uma vez se revelou bastante bom. O *Stochastic Approximation to Gradient Descent* também apresenta valores aceitáveis para os resultados obtidos.
- Relativamente à **duração** do treino de redes, tanto o *Perceptron Training Rule* como o *Gradient Descent* apresentam bons tempos; o *Stochastic Approximation to Gradient Descent* é mais moroso.
- É de relevar que nestes testes realizados, não foram definidas as percentagens para treino, teste e validação, pelo que as comparações dos algoritmos previamente realizadas poderão ser enganadoras.

3.2. Folhas_2: dados de validação

Nos testes realizados para a pasta de Folhas_2 houve a predefinição da percentagem de treino, validação e teste (Figura 9):

- Treino: 70%.
- Validação: 15%.
- Teste: 15%.

Ao contrário dos testes anteriormente realizados, onde as percentagens foram atribuídas de forma aleatória pelo programa, através da sua predefinição, neste capítulo, conta-se perceber quais as funções às quais correspondem melhores taxas de acerto.

3.2.1. Perceptron training rule

Utilizando o algoritmo de *perceptron training rule*, obtêm-se os resultados apresentados na Tabela 6. Desta se conclui que:

- As **funções de ativação** com maior taxa de acerto para as espécies são as que possuem a função linear ou sigmoide numa das camadas (taxas que rondam os 52%). Para as subespécies, e novamente, as taxas de acerto foram baixas, sendo a maior, de 24%, para a qual foi utilizada a função tangente hiperbólica.
- Os testes com maior **duração** correspondem àqueles com funções não lineares, sigmoide e tangente hiperbólica (acima de 38 minutos, chegando aos 100 minutos, isto é, 1 hora e 40 minutos).
- Em relação ao **número de épocas**, o teste com função linear e tangente hiperbólica apresentou o maior valor: 47.

Tabela 6 Resultados obtidos para perceptron training rule – Folhas_2.

Nome do Ficheiro	Rede Carregada	Inputs	Camadas escondidas	Funções de ativação		Topologia da rede	Rácio (em)			Quantidade de imagens tratadas	Espécies			Subespécies		
				1ª Função	2ª Função		Treino	Validação	Teste		Taxa de acerto (%)	Epoch	Tempo (minutos,segundos)	Taxa de acerto (%)	Epoch	Tempo (minutos,segundos)
Folhas_2	Off	Imagem em binário	10	Step	Step	Perceptron Training Rule	70	15	15	1564	12	0	0,01	1	0	0,00
Folhas_2	Off	Imagem em binário	10	Step	Linear	Perceptron Training Rule	70	15	15	1564	34	5	9,55	3	5	24,52
Folhas_2	Off	Imagem em binário	10	Step	Sigmoide	Perceptron Training Rule	70	15	15	1564	17	22	38,16	3	6	28,40
Folhas_2	Off	Imagem em binário	10	Step	TanH	Perceptron Training Rule	70	15	15	1564	35	15	26,63	6	12	65,47
Folhas_2	Off	Imagem em binário	10	Step	Sinal	Perceptron Training Rule	70	15	15	1564	8	0	0,00	1	0	0,00
Folhas_2	Off	Imagem em binário	10	Step	Nenhuma	Perceptron Training Rule	70	15	15	1564	38	5	8,41	3	6	31,52
Folhas_2	Off	Imagem em binário	10	Linear	Linear	Perceptron Training Rule	70	15	15	1564	55	12	22,44	9	14	74,10
Folhas_2	Off	Imagem em binário	10	Linear	Sigmoide	Perceptron Training Rule	70	15	15	1564	40	14	53,84	3	20	143,91
Folhas_2	Off	Imagem em binário	10	Linear	TanH	Perceptron Training Rule	70	15	15	1564	55	47	100,40	1	40	260,05
Folhas_2	Off	Imagem em binário	10	Linear	Sinal	Perceptron Training Rule	70	15	15	1564	3	0	0,00	3	0	0,00
Folhas_2	Off	Imagem em binário	10	Linear	Nenhuma	Perceptron Training Rule	70	15	15	1564	52	10	21,48	13	11	65,77
Folhas_2	Off	Imagem em binário	10	Sigmoide	Sigmoide	Perceptron Training Rule	70	15	15	1564	8	10	18,69	0	7	36,48
Folhas_2	Off	Imagem em binário	10	Sigmoide	TanH	Perceptron Training Rule	70	15	15	1564	51	10	17,97	13	11	55,42
Folhas_2	Off	Imagem em binário	10	Sigmoide	Sinal	Perceptron Training Rule	70	15	15	1564	12	0	0,01	1	0	0,00
Folhas_2	Off	Imagem em binário	10	Sigmoide	Nenhuma	Perceptron Training Rule	70	15	15	1564	44	8	15,65	10	10	51,94
Folhas_2	Off	Imagem em binário	10	TanH	TanH	Perceptron Training Rule	70	15	15	1564	53	10	20,46	24	12	61,71
Folhas_2	Off	Imagem em binário	10	TanH	TanH	Perceptron Training Rule	70	15	15	1564	51	9	16,47	1	10	51,68
Folhas_2	Off	Imagem em binário	10	TanH	Sinal	Perceptron Training Rule	70	15	15	1564	9	0	0,00	1	0	0,00
Folhas_2	Off	Imagem em binário	10	TanH	Sinal	Perceptron Training Rule	70	15	15	1564	4	0	0,00	0	0	0,00
Folhas_2	Off	Imagem em binário	10	TanH	Nenhuma	Perceptron Training Rule	70	15	15	1564	54	8	14,78	9	13	67,37
Folhas_2	Off	Imagem em binário	10	Sinal	Sinal	Perceptron Training Rule	70	15	15	1564	7	0	0,00	0	0	0,00
Folhas_2	Off	Imagem em binário	10	Sinal	Nenhuma	Perceptron Training Rule	70	15	15	1564	36	5	8,17	3	6	27,70
Folhas_2	Off	Imagem em binário	10	Sinal	Nenhuma	Perceptron Training Rule	70	15	15	1564	36	5	8,44	1	5	23,23

3.2.2. Gradient Descent

Utilizando o algoritmo de *Gradient Descent*, obtêm-se os resultados apresentados na Tabela 7.

7. Desta se conclui que:

- As **funções de ativação** com maior taxa de acerto para as espécies são as que possuem a função linear (taxas muito baixas, de 10 e 15%). Para as subespécies, foram de 0, 1 e 3%.
- Os testes foram muito velozes, todos com **duração** inferior a 1 minuto.
- A quantidade de **épocas** foi novamente muito elevada, com vários registos de 1000.

Tabela 7 Resultados obtidos para Gradient Descent – Folhas_2.

Nome do Ficheiro	Rede Carregada	Inputs	Camadas escondidas	Funções de ativação		Topologia da rede	Rácio (em)			Quantidade de Imagens tratadas	Espécies			Subespécies		
				1ª Função	2ª Função		Treino	Validação	Teste		Taxa de acerto (%)	Epoch	Tempo (minutos,segundos)	Taxa de acerto (%)	Epoch	Tempo (minutos,segundos)
Folhas_2	Off	Imagem em binário	10	Step	Step	Gradient Descent	70	15	15	1564	8	0	0,00	0	0	0,00
Folhas_2	Off	Imagem em binário	10	Step	Linear	Gradient Descent	70	15	15	1564	4	1000	0,05	0	1000	0,07
Folhas_2	Off	Imagem em binário	10	Step	Sigmoide	Gradient Descent	70	15	15	1564	3	1000	0,05	1	1000	0,07
Folhas_2	Off	Imagem em binário	10	Step	TanH	Gradient Descent	70	15	15	1564	3	1000	0,05	3	1000	0,07
Folhas_2	Off	Imagem em binário	10	Step	Sinal	Gradient Descent	70	15	15	1564	7	0	0,00	1	0	0,00
Folhas_2	Off	Imagem em binário	10	Step	Nenhuma	Gradient Descent	70	15	15	1564	2	1000	0,05	0	1000	0,07
Folhas_2	Off	Imagem em binário	10	Linear	Linear	Gradient Descent	70	15	15	1564	10	1000	0,06	1	1000	0,07
Folhas_2	Off	Imagem em binário	10	Linear	Sigmoide	Gradient Descent	70	15	15	1564	8	1000	0,05	2	1000	0,07
Folhas_2	Off	Imagem em binário	10	Linear	TanH	Gradient Descent	70	15	15	1564	3	1000	0,05	1	1000	0,07
Folhas_2	Off	Imagem em binário	10	Linear	Sinal	Gradient Descent	70	15	15	1564	15	0	0,00	1	0	0,00
Folhas_2	Off	Imagem em binário	10	Linear	Nenhuma	Gradient Descent	70	15	15	1564	10	1000	0,05	1	1000	0,07
Folhas_2	Off	Imagem em binário	10	Sigmoide	Sigmoide	Gradient Descent	70	15	15	1564	1	1000	0,06	1	1000	0,08
Folhas_2	Off	Imagem em binário	10	Sigmoide	TanH	Gradient Descent	70	15	15	1564	1	1000	0,05	0	1000	0,07
Folhas_2	Off	Imagem em binário	10	Sigmoide	Sinal	Gradient Descent	70	15	15	1564	3	0	0,00	1	0	0,00
Folhas_2	Off	Imagem em binário	10	Sigmoide	Nenhuma	Gradient Descent	70	15	15	1564	3	1000	0,05	1	1000	0,07
Folhas_2	Off	Imagem em binário	10	TanH	TanH	Gradient Descent	70	15	15	1564	0	1000	0,06	1	1000	0,08
Folhas_2	Off	Imagem em binário	10	TanH	Sinal	Gradient Descent	70	15	15	1564	13	0	0,00	2	0	0,00
Folhas_2	Off	Imagem em binário	10	TanH	Nenhuma	Gradient Descent	70	15	15	1564	9	1000	0,05	1	1000	0,07
Folhas_2	Off	Imagem em binário	10	Sinal	Sinal	Gradient Descent	70	15	15	1564	8	0	0,00	1	0	0,00
Folhas_2	Off	Imagem em binário	10	Sinal	Nenhuma	Gradient Descent	70	15	15	1564	1	1000	0,05	1	1000	0,07

3.2.3. Stochastic Approximation to Gradient Descent

Utilizando o algoritmo de *Stochastic Approximation to Gradient Descent*, obtêm-se os resultados apresentados na Tabela 8. Desta se conclui que:

- As **funções de ativação** com maior taxa de acerto para as espécies são as que possuem a função linear (taxas de 45 e 52%). Para as subespécies, foram de 10 e 11%.
- Em termos de **duração**, obtiveram-se tempos superiores a 1 hora nalguns casos.
- A quantidade de épocas variou bastante, entre as 0 e as 155.

Tabela 8 Resultados obtidos para Stochastic Approximation to Gradient Descent – Folhas_2.

Nome do Ficheiro	Rede Carregada	Inputs	Camadas escondidas	Funções de ativação		Topologia da rede	Rácio (em)			Quantidade de Imagens tratadas	Espécies			Subespécies		
				1ª Função	2ª Função		Treino	Validação	Teste		Taxa de acerto (%)	Epoch	Tempo (minutos,segundos)	Taxa de acerto (%)	Epoch	Tempo (minutos,segundos)
Folhas_2	Off	Imagem em binário	10	Step	Step	Stochastic Approximation	70	15	15	1564	8	0	0,00	1	0	0,00
Folhas_2	Off	Imagem em binário	10	Step	Linear	Stochastic Approximation	70	15	15	1564	38	35	18,23	1	42	26,35
Folhas_2	Off	Imagem em binário	10	Step	Sigmoide	Stochastic Approximation	70	15	15	1564	12	3	1,61	0	4	2,75
Folhas_2	Off	Imagem em binário	10	Step	TanH	Stochastic Approximation	70	15	15	1564	37	4	2,08	2	13	7,92
Folhas_2	Off	Imagem em binário	10	Step	Sinal	Stochastic Approximation	70	15	15	1564	12	0	0,01	1	0	0,00
Folhas_2	Off	Imagem em binário	10	Step	Nenhuma	Stochastic Approximation	70	15	15	1564	45	38	21,04	2	24	15,79
Folhas_2	Off	Imagem em binário	10	Linear	Linear	Stochastic Approximation	70	15	15	1564	1	123	40,62	11	124	78,28
Folhas_2	Off	Imagem em binário	10	Linear	Sigmoide	Stochastic Approximation	70	15	15	1564	0	12	7,04	1	5	3,46
Folhas_2	Off	Imagem em binário	10	Linear	TanH	Stochastic Approximation	70	15	15	1564	1	7	3,83	2	11	7,42
Folhas_2	Off	Imagem em binário	10	Linear	Sinal	Stochastic Approximation	70	15	15	1564	9	0	0,01	2	0	0,00
Folhas_2	Off	Imagem em binário	10	Linear	Nenhuma	Stochastic Approximation	70	15	15	1564	52	155	86,26	10	128	85,86
Folhas_2	Off	Imagem em binário	10	Sigmoide	Sigmoide	Stochastic Approximation	70	15	15	1564	12	7	3,70	1	5	3,71
Folhas_2	Off	Imagem em binário	10	Sigmoide	TanH	Stochastic Approximation	70	15	15	1564	38	34	20,76	1	15	10,96
Folhas_2	Off	Imagem em binário	10	Sigmoide	Sinal	Stochastic Approximation	70	15	15	1564	2	0	0,00	2	0	0,00
Folhas_2	Off	Imagem em binário	10	Sigmoide	Nenhuma	Stochastic Approximation	70	15	15	1564	42	17	9,81	0	16	11,21
Folhas_2	Off	Imagem em binário	10	TanH	TanH	Stochastic Approximation	70	15	15	1564	3	32	17,06	1	26	17,08
Folhas_2	Off	Imagem em binário	10	TanH	Sinal	Stochastic Approximation	70	15	15	1564	11	0	0,00	2	0	0,00
Folhas_2	Off	Imagem em binário	10	TanH	Nenhuma	Stochastic Approximation	70	15	15	1564	29	16	8,57	4	84	53,01
Folhas_2	Off	Imagem em binário	10	Sinal	Sinal	Stochastic Approximation	70	15	15	1564	7	0	0,00	1	0	0,00
Folhas_2	Off	Imagem em binário	10	Sinal	Nenhuma	Stochastic Approximation	70	15	15	1564	45	38	22,93	2	31	22,31

3.2.4. Conclusão para Folhas_2

- O algoritmo que regista maior **taxa de acerto** na rede de espécie é novamente, o *Perceptron Training Rule*. O algoritmo de *Stochastic Approximation to Gradient Descent* também revelou valores bons de taxa de acerto, apesar de não serem com tanta frequência como o do *Perceptron Training Rule*. O *Gradient Descent* revelou-se, mais uma vez, mau nos testes realizados.
- Relativamente ao **número de épocas**, o *Perceptron Training Rule* realizou ajustes em quantidade inferior, isto é, o máximo atingido foi de 47, apesar de obter números positivos em praticamente todos os testes. O *Stochastic Approximation to Gradient Descent* apresentou valores muito variantes, do 0 ao 155. Novamente, o *Gradient Descent* tem valores muito elevados neste ponto.
- Quanto à **duração**, tanto o *Perceptron Training Rule* como o *Stochastic Approximation to Gradient Descent* tempos semelhantes, isto é, elevados; o *Gradient Descent* teve ótimos tempos.

3.3. Comparação de taxas de acerto

Pela análise realizada, conclui-se facilmente, que:

- Apesar de o algoritmo *Gradient Descent* apresentar ótimos tempos e um elevado número de épocas, não lhe corresponde boas taxas de acerto, pelo que não se recomenda a sua utilização.
- Ao *Perceptron Training Rule* correspondem boas taxas de acerto, atingindo uma taxa de 54% para a espécie, e à qual está associada uma de 9% para a subespécie. O número de épocas variou entre o 0 e 47. Pelos testes realizados não é possível extrapolar se a maior número de épocas ou a maior duração corresponde uma maior taxa de acerto.
- Em relação ao *Stochastic Approximation to Gradient Descent*, as suas taxas de acerto variaram dos 0 aos 53% para espécie e dos 0 aos 11% para as subespécies.
- A uma boa taxa de acerto na rede da espécie não corresponde, necessariamente, uma boa taxa de acerto à subespécie. Registaram-se casos em que houve taxa de acerto na espécie

de 60% e 0 na subespécie. Da mesma maneira, a elevada duração ou elevado número de épocas também são inconclusivas em relação a esta taxa (Tabela 9).

Tabela 9 Resumo das melhores taxas de acerto nos testes realizados.

Nome do Ficheiro	Rede Carregada	Inputs	Camadas escondidas	Funções		Topologia da rede	Rácio			Quantidade de imagens tratadas	Espécies			Subespécies		
				1ª Função	2ª Função		Treino	Validação	Teste		Taxa de acerto (%)	Epoch	Tempo (minutos,segundos)	Taxa de acerto (%)	Epoch	Tempo (minutos,segundos)
Folhas_1	Off	Imagem em binário	10	Step	Step	Perceptron Training Rule	-----	-----	-----	99	47	5	1,05	0	5	3,01
Folhas_1	Off	Imagem em binário	10	Step	Linear	Perceptron Training Rule	-----	-----	-----	99	40	5	1,11	0	5	3,08
Folhas_1	Off	Imagem em binário	10	Step	TanH	Perceptron Training Rule	-----	-----	-----	99	40	9	2,14	0	18	12,06
Folhas_1	Off	Imagem em binário	10	Sigmoide	Sigmoide	Perceptron Training Rule	-----	-----	-----	99	40	8	2,51	0	9	6,68
Folhas_1	Off	Imagem em binário	10	TanH	TanH	Perceptron Training Rule	-----	-----	-----	99	47	12	3,84	0	9	7,34
Folhas_1	Off	Imagem em binário	10	Step	Linear	Stochastic Approximation	-----	-----	-----	99	53	31	13,90	0	22	12,46
Folhas_1	Off	Imagem em binário	10	Sigmoide	Sigmoide	Stochastic Approximation	-----	-----	-----	99	60	26	14,16	0	19	12,75
Folhas_1	Off	Imagem em binário	10	Sigmoide	TanH	Stochastic Approximation	-----	-----	-----	99	47	7	4,16	0	11	7,87
Folhas_2	Off	Imagem em binário	10	Linear	Linear	Perceptron Training Rule	70	15	15	1564	55	12	22,44	9	14	74,10
Folhas_2	Off	Imagem em binário	10	Linear	Sigmoide	Perceptron Training Rule	70	15	15	1564	40	14	53,84	3	20	143,91
Folhas_2	Off	Imagem em binário	10	Linear	TanH	Perceptron Training Rule	70	15	15	1564	55	47	100,40	1	40	260,05
Folhas_2	Off	Imagem em binário	10	Linear	Nenhuma	Perceptron Training Rule	70	15	15	1564	52	10	21,48	13	11	65,77
Folhas_2	Off	Imagem em binário	10	Sigmoide	TanH	Perceptron Training Rule	70	15	15	1564	51	10	17,97	13	11	55,42
Folhas_2	Off	Imagem em binário	10	Sigmoide	Nenhuma	Perceptron Training Rule	70	15	15	1564	44	8	15,65	10	10	51,94
Folhas_2	Off	Imagem em binário	10	TanH	TanH	Perceptron Training Rule	70	15	15	1564	53	10	20,46	24	12	61,71
Folhas_2	Off	Imagem em binário	10	TanH	TanH	Perceptron Training Rule	70	15	15	1564	51	9	16,47	1	10	51,68
Folhas_2	Off	Imagem em binário	10	TanH	Nenhuma	Perceptron Training Rule	70	15	15	1564	54	8	14,78	9	13	67,37
Folhas_2	Off	Imagem em binário	10	Step	Nenhuma	Stochastic Approximation	70	15	15	1564	45	38	21,04	2	24	15,79
Folhas_2	Off	Imagem em binário	10	Linear	Nenhuma	Stochastic Approximation	70	15	15	1564	52	155	86,26	10	128	85,86
Folhas_2	Off	Imagem em binário	10	Sigmoide	Nenhuma	Stochastic Approximation	70	15	15	1564	42	17	9,81	0	16	11,21
Folhas_2	Off	Imagem em binário	10	Sinal	Nenhuma	Stochastic Approximation	70	15	15	1564	45	38	22,93	2	31	22,31

3.4. Folhas_3: dados de teste

Com a pasta de Folhas_3, que contém 20 imagens, testa-se verdadeiramente a melhor rede obtida e gravada dos treinos anteriormente efetuados (Tabela 10). Com a rede gravada, foi registada uma percentagem de acerto de 21% para as espécies e de 0% para as subespécies. Após esta rede ter sido novamente treinada conseguiu-se melhorar as taxas de acerto da espécie, além da subespécie ter deixado de ser nula.

Tabela 10 Resultados de Folhas_3.

Nome do Ficheiro	Rede Carregada	Inputs	Camadas escondidas	Funções de ativação		Topologia da rede	Rácio (em)			Quantidade de imagens tratadas	Espécies			Subespécies	
				1ª Função	2ª Função		Treino	Validação	Teste		Taxa de acerto (%)	Epoch	Tempo (minutos,segundos)	Taxa de acerto (%)	Epoch
Folhas_3	On	Imagem em binário	-----	-----	-----	-----	-----	-----	-----	20	21	-----	-----	0	-----

3.5. Folhas personalizadas

O trabalho desenvolvido permite o reconhecimento de folhas que não tenham sido previamente inseridas, isto é, que não estejam nem em Folhas_1 nem Folhas_2. Para isso, testou-

se com os exemplos da Figura 16, onde além de serem imagens a cores, possuem diferentes tamanhos, utilizando a extração de características.

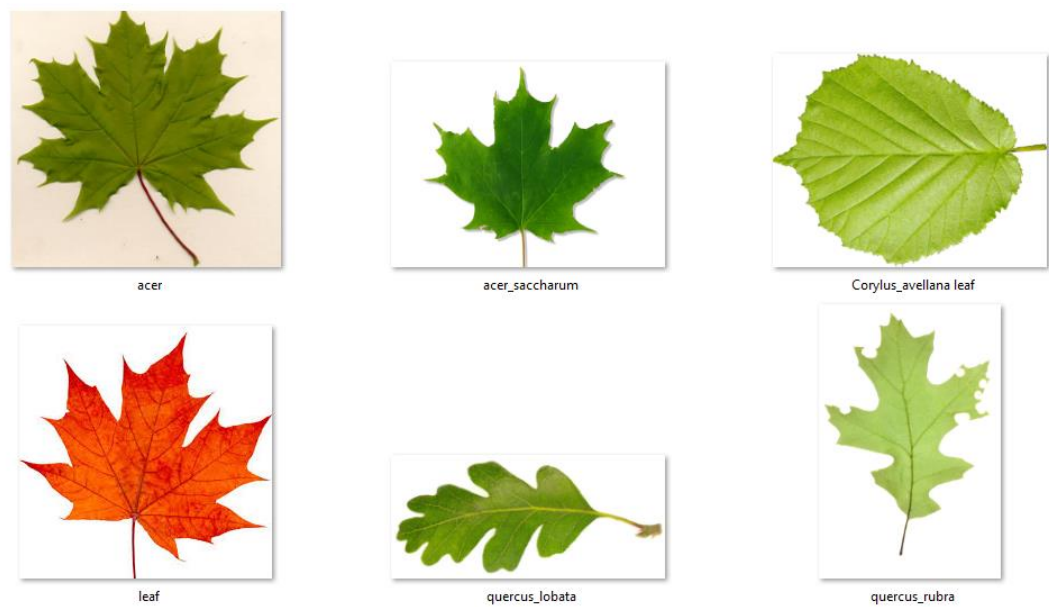


Figura 16 Folhas personalizadas.

Estas folhas deverão ser colocadas numa pasta específica, a Folhas_Personalizadas, de maneira a que o programa desenvolvido e a rede consigam identificar a espécie e subespécie (Tabela 11).

Tabela 11 Resultados de testes efetuados a Folhas personalizadas.

Nome da Imagem	Inputs	Espécie	Subespécie
acer.JPG	Características da imagem	Quercus	Palustris
acer_saccharum.jpg	Características da imagem	Quercus	Simonii
Corylus_avellana leaf.jpg	Características da imagem	Quercus	Simonii
quercus_lobata.jpg	Características da imagem	Quercus	Tulipifera
quercus_rubra.jpg	Características da imagem	Quercus	Tulipifera
leaf.jpg	Características da imagem	Acer	Simonii
leaf.jpg	Características da imagem	Rhododendron	Intergra
acer_saccharum.jpg	Características da imagem	Quercus	Simonii
acer_saccharum.jpg	Características da imagem	Eucalyptus	Chinensis
acer_saccharum.jpg	Características da imagem	Quercus	Simonii
leaf.jpg	Características da imagem	Acer	Simonii
acer.JPG	Características da imagem	Olea	Vulcanica
quercus_lobata.jpg	Características da imagem	Tilia	Saccharinum
quercus_lobata.jpg	Características da imagem	Quercus	Tulipifera

3.6. Quantidade de neurónios variável

Um dos objetivos a atingir no desenvolvimento deste trabalho prático consistia em testar a rede com diferente número de neurónios (Tabela 12). As redes com melhor taxa de acerto corresponderam às com 8 e 10 neurónios, pelo que, das duas, se optou por utilizar com 10 neurónios.

Tabela 12 Testes com diferente quantidade de neurónios.

Nome do Ficheiro	Rede Carregada	Inputs	Camadas escondidas	Funções de ativação		Topologia da rede	Rácio (em)			Quantidade de imagens tratadas	Espécies			Subespécies		
				1ª Função	2ª Função		Treino	Validação	Teste		Taxa de acerto (%)	Epoch	Tempo (minutos,segundos)	Taxa de acerto (%)	Epoch	Tempo (minutos,segundos)
Folhas_3	On	Imagem em binário	----	----	----	----	----	----	----	20	21	----	----	0	----	----
Folhas_1	On	Imagem em binário	----	----	----	----	----	----	----	99	54	----	----	34	----	----
Folhas_2	Off	Características da imagem	10	Step	Step	Perceptron Training Rule	70	15	15	1564	11	0	0,01	1	0	0,00
Folhas_2	On	Imagem em binário	----	----	----	----	----	----	----	1564	56	----	----	47	----	----
Folhas_3	On	Imagem em binário	----	----	----	----	----	----	----	20	21	----	----	0	----	----
Folhas_2	Off	Imagem em binário	10	TanH	TanH	Perceptron Training Rule	70	15	15	1564	61	7	12,78	19	50	267,86
Folhas_3	On	Imagem em binário	----	----	----	----	----	----	----	20	26	----	----	16	----	----
Folhas_2	Off	Imagem em binário	8	TanH	TanH	Perceptron Training Rule	70	15	15	1564	62	14	16,23	14	12	41,63
Folhas_2	Off	Imagem em binário	5	TanH	TanH	Perceptron Training Rule	70	15	15	1564	42	16	6,66	5	12	16,20
Folhas_2	Off	Imagem em binário	2	TanH	TanH	Perceptron Training Rule	70	15	15	1564	12	16	1,07	0	15	3,55
Folhas_2	Off	Imagem em binário	15	TanH	TanH	Perceptron Training Rule	70	15	15	1564	50	27	133,55	17	10	143,69
Folhas_2	On	Imagem em binário	10	TanH	TanH	Perceptron Training Rule	70	15	15	1564	59	6	11,44	20	7	38,62
Folhas_2	On	Imagem em binário	----	----	----	----	----	----	----	1564	60	----	----	22	----	----
Folhas_3	On	Imagem em binário	----	----	----	----	----	----	----	20	26	----	----	16	----	----

4. INTERFACE GRÁFICA

Através da utilização do *guide* do Matlab criou-se a interface gráfica do programa. Nesta, o utilizador começará por escolher se pretende consultar os dados dos autores do trabalho, ir para o programa ou ler o relatório (Figura 17):



Figura 17 Interface - janela inicial.

4.1. Autores

Na janela de autores, escolhe de qual dos dois pretende obter mais informação, incluindo os contactos em redes sociais (Figura 18).



Figura 18 Interface - autores.

4.2. Relatório

Clicando em “Relatório”, abre o ficheiro .pdf deste relatório.

4.3. Programa

Nesta nova janela, local de seleção das características desejadas, escolhe-se a pasta da qual pretende ler as imagens que, ao pressionar o botão “tratar imagens”, converterá para matrizes binárias (Figura 19).



Figura 19 Interface - inserir imagem e seu tratamento.

O utilizador poderá configurar a rede selecionando as funções de treino desejadas para duas camadas (Figura 20): Step, Linear, Sigmoid, Tangente hiperbólica, Sinal.

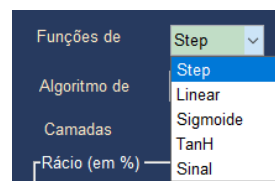


Figura 20 Interface - escolha de funções de ativação.

Escolhe o algoritmo de treino (Figura 21):

- *Perceptron training rule.*
- *Gradient Descent.*
- *Stochastic approximation to gradient descent.*

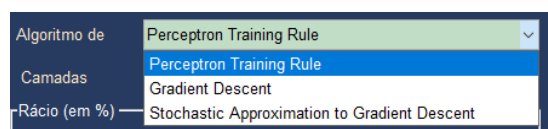


Figura 21 Interface - algoritmos de treino da rede.

E a percentagem atribuída ao treino, validação e teste (Figura 22).

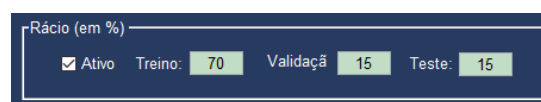


Figura 22 Interface - percentagem atribuída a treino, validação e teste.

No output sairá o total de imagens lidas e a taxa de acerto (Figura 23). Pressionando o botão “visualizar”, surge uma tabela com os dados das redes gravadas (Tabela 13), onde o campo “Tempo” corresponde ao tempo necessário para processar os dados.

Output

Espécies

Total de imagens:

Taxa de acerto:

Visualizar

Sub espécies

Total de imagens:

Taxa de acerto:

Gravar

Figura 23 Interface - output.

Tabela 13 Resultado das redes gravadas.

[illegible]

4.4. Exemplos de interface no decorrer de testes

Painel inicial de seleção de características da rede ou importação de rede previamente gravada (Figura 24).

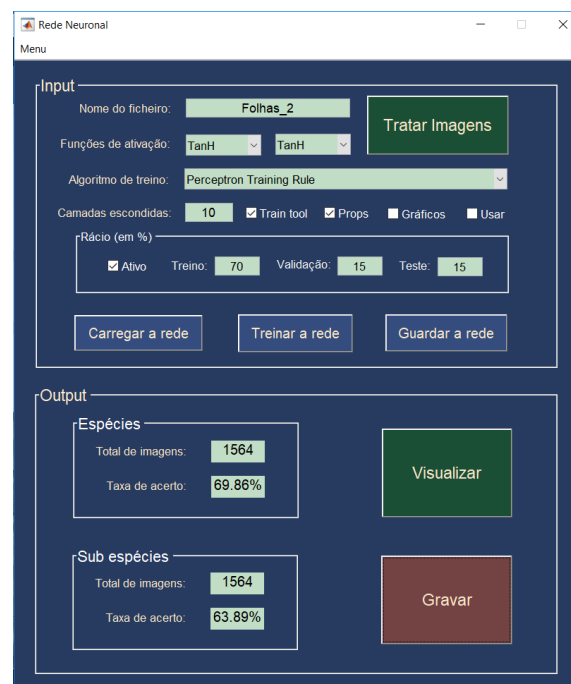


Figura 24 Resultado de treino de rede utilizando a função tangente hiperbólica.

Exemplo de um teste efetuado, neste caso, para Folhas_2 (Figura 25).

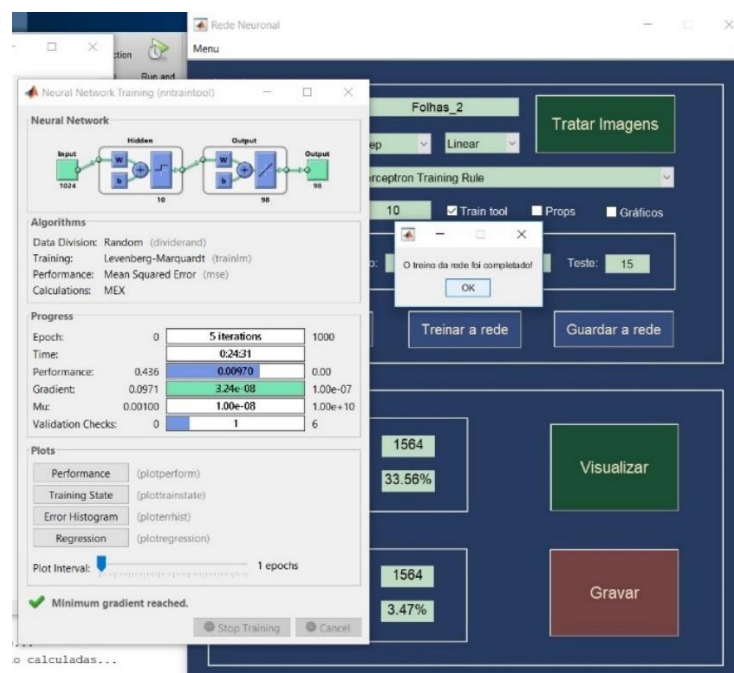


Figura 25 Teste efetuado para a pasta Folhas_2.

Pressionando o botão do *Error Histogram*, na janela automática do Matlab, acima apresentada, surge um gráfico de barras semelhante ao que abaixo se apresenta (Figura 26).

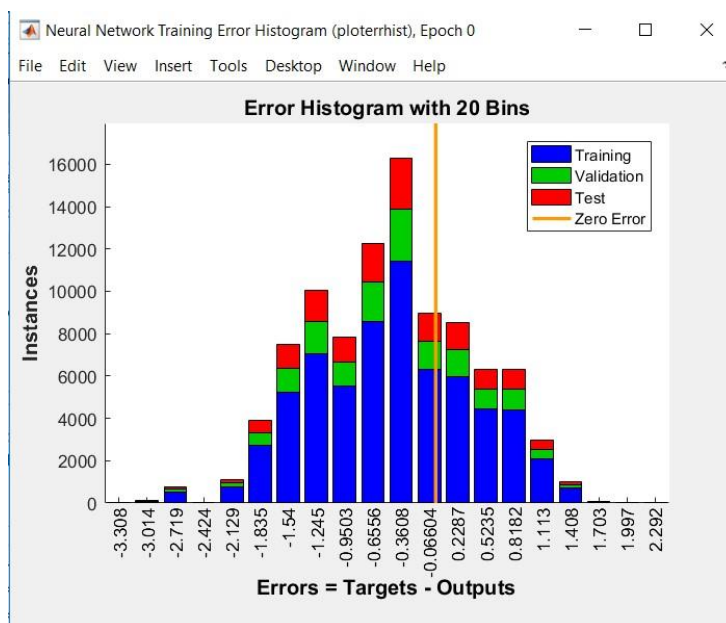


Figura 26 Histograma de erros para um dos testes efetuados.

A matriz de confusão obtida diretamente pelo Matlab não poderá ser aplicada num trabalho com tantas entradas e saídas na rede neuronal por se tornar ilegível (Figura 27), piorando para a subespécie (Figura 28).

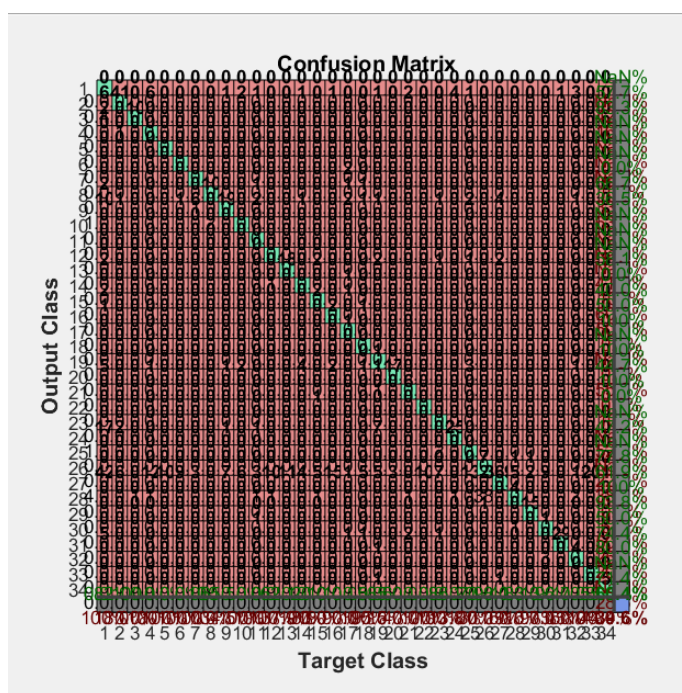


Figura 27 Matriz de confusão para espécies fornecida pelo Matlab, com taxa de acerto de 60,96%.

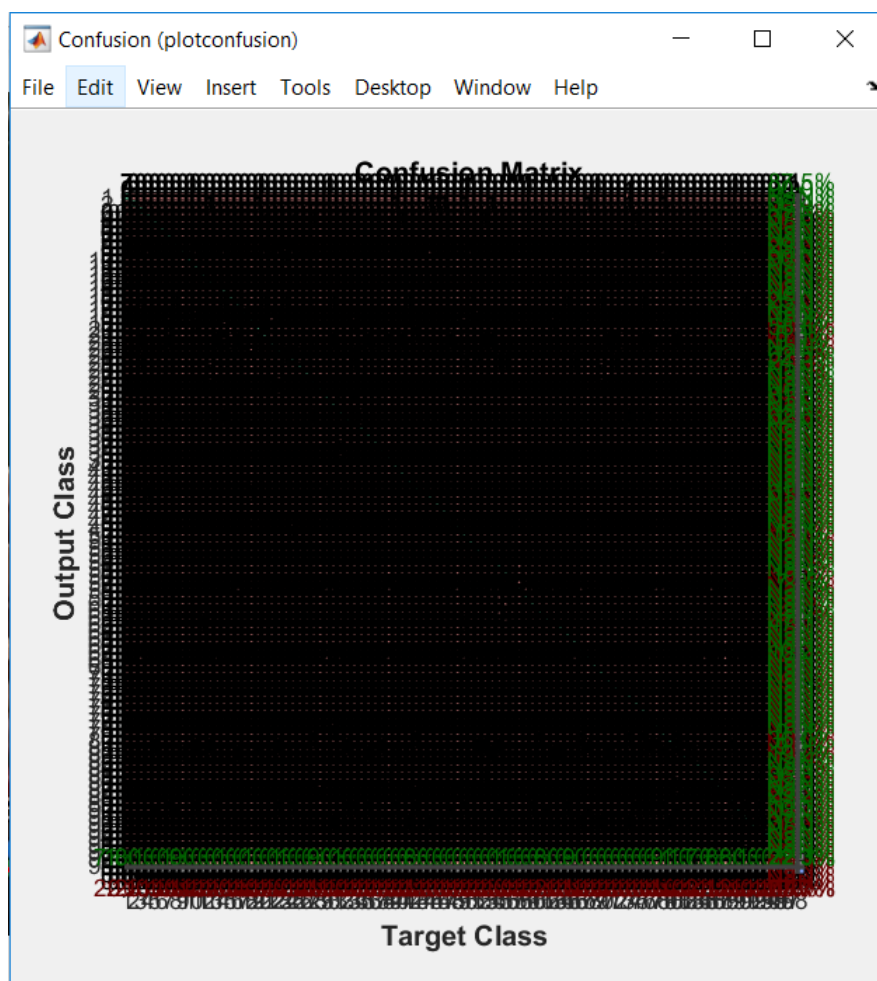


Figura 28 Matriz de confusão para espécies fornecida pelo Matlab, com taxa de acerto de 22,92%.

5. CONCLUSÃO

Às redes neuronais estão associadas 3 fases, às quais são atribuídas percentagens distintas:

- Treino.
- Validação.
- Teste.

Habitualmente o Matlab atribui-lhas aleatoriamente, mas na prática, costumam ser definidos valores à partida para cada fase: treino, com 70%, validação, com 15% e teste, também com 15%. Estas percentagens são utilizadas para verificar se a rede trabalha convenientemente e para se ter um termo de comparação entre os algoritmos utilizados.

As matrizes de confusão obtidas diretamente em Matlab são ilegíveis para redes com elevado número de inputs e outputs. Por esta razão, tem de se criar à parte, as matrizes correspondentes.

Constata-se pela utilização do algoritmo *Gradient Descent* que apresenta ótimos tempos e um elevado número de épocas. Contudo, não lhe corresponde boas taxas de acerto, pelo que não se recomenda a sua utilização.

Ao *Perceptron Training Rule* atribuem-se boas taxas de acerto, atingindo uma taxa de 54% para a espécie. O número de épocas variou entre o 0 e 47. Pelos testes realizados não é possível extrapolar se a maior número de épocas ou a maior duração corresponde uma maior taxa de acerto.

Em relação ao *Stochastic Approximation to Gradient Descent*, as suas taxas de acerto variaram dos 0 aos 53% para espécie e dos 0 aos 11% para as subespécies.

A uma boa taxa de acerto na rede da espécie não corresponde, necessariamente, uma boa taxa de acerto à subespécie. Registaram-se casos em que houve taxa de acerto na espécie de 60% e 0 na subespécie. Da mesma maneira, a elevada duração ou elevado número de épocas também são inconclusivas em relação a esta taxa.

Além dos testes supramencionados, também se testaram os resultados através da **variação do número de neurónios**, concluindo-se que, para o trabalho em causa, o ideal será utilizar 8 ou 10 neurónios. Neste trabalho, optou-se por se escolher rede com 10 neurónios, pelos bons resultados registados e por ser o modo *default* do Matlab. Aparentemente, a um maior número de neurónios, corresponde maior duração de testes.

6. BIBLIOGRAFIA

- Cortez, V. T. (janeiro de 2015). Aplicação de redes neurais artificiais à deteção e isolamento de falhas em processos industriais. *Dissertação de Mestrado*. Lisboa: Instituto Superior de Engenharia de Lisboa.
- How to build your own Neural Network* . (s.d.). Obtido de towardsdatascience: <https://towardsdatascience.com/how-to-build-your-own-neural-network-from-scratch-in-python-68998a08e4f6>
- Marques, V. (s.d.). Redes Neurais. *Aula 1 - Redes Neurais*. Instituto Superior de Engenharia de Coimbra.
- Matlab. (s.d.). *regionprops*. Obtido de MathWorks: <https://www.mathworks.com/help/images/ref/regionprops.html>
- Menezes, P. (s.d.). *O Que é uma Rede Neuronal?* Obtido de Instituto de Sistemas e Robótica - Universidade de Coimbra: <http://home.isr.uc.pt/~paulo/PROJ/NN95/node24.html>
- Moura, W. (4 de setembro de 2016). *Resampling: separando os dados entre treino, validação e teste*. Obtido de hacking analytics: <https://hackinganalytics.com/2016/09/04/resampling-separando-os-dados-entre-treino-validacao-e-teste/>
- Redes neuronais*. (s.d.).
- Santos, J. M. (s.d.). *Redes neuronais - conceitos*. Obtido de Instituto Superior de Engenharia do Porto: [http://www.isep.ipp.pt/files/Redes%20Neuronais%20-%20Conceitos%20-%20JMS%20\(vers%C3%A3o_final\).pdf](http://www.isep.ipp.pt/files/Redes%20Neuronais%20-%20Conceitos%20-%20JMS%20(vers%C3%A3o_final).pdf)
- Sharma, S. (22 de setembro de 2017). *Epoch vs Batch Size vs Iterations*. Obtido de Towards Data Science: <https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9>

ANEXOS

ANEXO A

ID's correspondentes às espécies

id_especie	especie
1	Acer
2	Alnus
3	Arundinaria
4	Betula
5	Callicarpa
6	Castanea
7	Celtis
8	Cercis
9	Cornus
10	Cotinus
11	Crataegus
12	Cytisus
13	Eucalyptus
14	Fagus
15	Ginkgo
16	Ilex
17	Liquidambar
18	Liriodendron
19	Lithocarpus
20	Magnolia
21	Morus
22	Olea
23	Phildelphus
24	Populus
25	Prunus
26	Pterocarya
27	Quercus
28	Rhododendron
29	Salix
30	Sorbus
31	Tilia
32	Ulmus
33	Viburnum
34	Zelkova

ANEXO B

ID's correspondentes às subespécies

id_subespecie	subespecie
1	Capillipes
2	Circinatum
3	Mono
4	Opalus
5	Palmatum
6	Pictum
7	Platanoids
8	Rubrum
9	Rufinerve
10	Saccharinum
11	Cordata
12	Maximowiczii
13	Rubra
14	Sieboldiana
15	Viridis
16	Simonii
17	Austrosinensis
18	Pendula
19	Bodinieri
20	Sativa
21	Koraiensis
22	Siliquastrum
23	Chinensis
24	Controversa
25	Macrophylla
26	Coggygria
27	Monogyna
28	Battandieri
29	Glaucescens
30	Neglecta
31	Urnigera
32	Sylvatica

id_subespecie	subespecie
33	Biloba
34	Aquifolium
35	Cornuta
36	Styraciflua
37	Tulipifera
38	Cleistocarpus
39	Edulis
40	Heptapeta
41	Salicifolia
42	Nigra
43	Europaea
44	Adenopoda
45	Grandidentata
46	Nigra
47	Avium
48	Shmittii
49	Stenoptera
50	Afares
51	Agrifolia
52	Alnifolia
53	Brantii
54	Canariensis
55	Castaneifolia
56	Cerris
57	Chrysolepis
58	Coccifera
59	Coccinea
60	Crassifolia
61	Crassipes
62	Dolicholepis
63	Ellipsoidalis
64	Greggii

id_subespecie	subespecie
65	Hartwissiana
66	Hispanica
67	Ilex
68	Imbricaria
69	Infectoria
70	Kewensis
71	Nigra
72	Palustris
73	Phellos
74	Phillyraeoides
75	Pontica
76	Pubescens
77	Pyrenaica
78	Rhysophylla
79	Rubra
80	Semecarpifolia
81	Shumardii
82	Suber
83	Texana
84	Trojana
85	Turneri
86	Variabilis
87	Vulcanica
88	Russellianum
89	Fragilis
90	Intergra
91	Aria
92	Oliveri
93	Platyphyllos
94	Tomentosa
95	Bergmanniana
96	Rhytidophylloides
97	Tinus
98	Serrata

