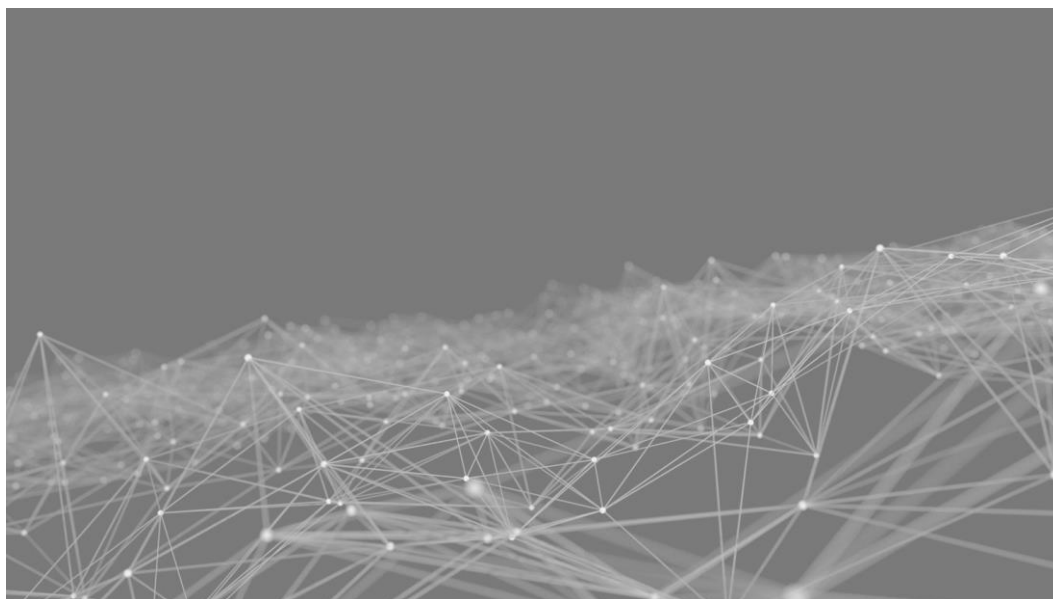




Trabalho Prático de Conhecimento e Raciocínio

Tema: Redes Neurais

Docente: Anabela Simões



Sérgio Apolinário da Costa (2020129026)

Diogo Santos (2020131432)

Licenciatura em Engenharia Informática – Curso Europeu

Coimbra, 2021/2022

Índice

Introdução.....	3
Tratamento das Imagens: Inputs e Targets.....	3
Treino e Estudo Estatístico	4
Alínea a.....	4
Alínea b.....	4
Alínea c.....	5
Alínea d.....	5
Interface Gráfica.....	6
Conclusão	6

Introdução

Este projeto surgiu no âmbito da unidade curricular de Conhecimento e Raciocínio no ano letivo de 2021/2022. Este trabalho foi feito em Matlab, e consiste na realização de um estudo estatístico de redes neuronais para a identificação de algumas figuras geométricas. No final pretende incluir uma aplicação para poder criar, treinar e simular redes neuronais, com dados introduzidos pelo utilizador e desenhos de novas figuras geométricas para apresentar novos casos a rede neuronal.

Tratamento das Imagens: Inputs e Targets

O dataset usado no projeto é composto por 4 pastas, cujo uma delas foi criada propositalmente por nos com várias figuras geométricas para a rede neuronal poder testá-las e adivinhar a figura. As restantes pastas foram fornecidas junto com o enunciado, são elas a pasta **train**, **test** e **start**, estas pastas tem dentro delas mais 6 pastas divididas por tipos diferentes de figuras geométricas.

Cada imagem tem uma proporção de 224*224 pixels, como treinar redes neuronais com estas custa tempos e velocidades elevadas, decidimos redimensionar as imagens antes de treinar por um tamanho de 25*25 pixels.

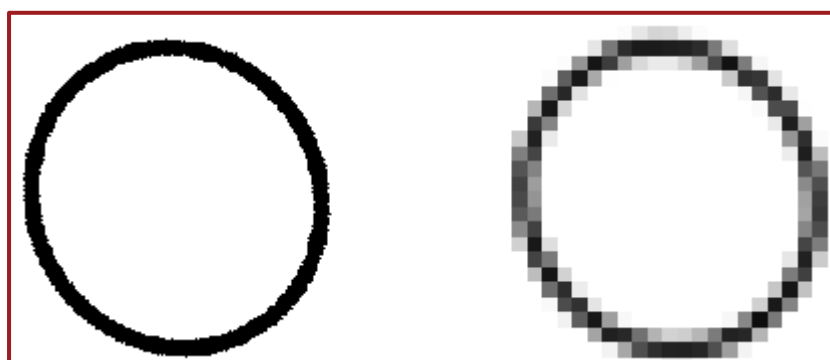


Fig. 1: Redimensionamento das imagens em Matlab - original vs redimensionada

Após o redimensionamento convertimos as imagens para binário e colocamo-las em matrizes binárias em uma única coluna, pois, no MATLAB, as redes neuronais recebem os seus inputs na forma de colunas de uma matriz.

De seguida são definidos os targets da rede neuronal. O target é uma matriz de 6 colunas (por causa dos 6 tipos diferentes de imagens) por a quantidade de imagens lidas, a matriz vai ser preenchida com um valor de 0 ou 1, 1 caso a imagem correspondente contenha o carater que a linha representa, ou 0, em caso contrário.

Abaixo esta representado o target da pasta start, em que a primeira linha representa o círculo, que é lido em primeiro lugar e as 5 primeiras células é onde estão figuras geométricas que representam um círculo, e assim é nos outros casos das outras figuras geométricas.

1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Treino e Estudo Estatístico

Alínea a

Iniciando-se o estudo da rede neuronal, após a sua criação e os devidos testes, que estão no Excel em anexo, onde foram testadas várias arquiteturas (topologias), funções de ativação e de treino, e usando as imagens da pasta **test**, chegamos as seguintes conclusões.

Adaptando a configuração base em que apenas existe uma camada escondida com 10 neurónios, para uma topologia com 2 camadas ocultas e com 10 neurónios em cada camada, notamos um impacto positivo. Já quando adaptamos a configuração base para camadas escondidas mais elevadas, geralmente os resultados são piores, ou até mesmo com duas camadas escondidas, mas com um número de neurónios em cada camada menores, conseguimos observar o impacto negativo comparado a configuração base.

Quando observamos o impacto da função de treino no desempenho da rede neuronal, podemos dizer que nos nossos testes não existe resultado melhor que a configuração base, mesmo sendo a que tem o resultado temporal mais elevado.

Por fim, no que diz respeito a função de ativação podemos considerar que quando usamos a função **purelin**, na camada de output, temos os melhores resultados. Já as outras funções não sobressairão a configuração base.

Porem, neste momento, ainda não podemos tirar conclusões precipitadas porque os testes efetuados não são necessariamente significativos, até porque todos os casos estão a ser usados para treino

Alínea b

Neste ponto do projeto, o objetivo é continuar a observar e testar o comportamento da rede neuronal com tipos de parâmetros diferentes tal como na alínea anterior, mas desta vez vamos distribuir diferentes valores para o dataset (treino, validação e teste) invés de colocar todo o peso no treino da rede neuronal e vamos usar as imagens da pasta **train**.

Seguindo a mesma orientação da primeira alínea, podemos dizer que os resultados do uso de camadas escondidas e do número de neurónios não são os mesmos, mas, a conclusão é igual, o uso de mais camadas escondidas compensa e conseguimos melhores resultados.

Experimentados outras funções de treino e observando os resultados obtidos, podemos dizer mais uma vez, que a melhor função de treino é a **trainlm**, e desta vez com resultados muitos distintos dos obtidos na alínea anterior.

Baseado nos testes realizados, chegamos a conclusão que, neste caso a função de ativação base, não é a melhor. As funções de ativação com melhor resultado para apenas uma camada escondida de neurónios são as funções **tansig** e **tansig** e no caso do uso de 3 camadas o melhor resultado a que nos chegamos foram as funções **logsig**, **tansig** e **purelin**.

Mudando a segmentação do dataset, podemos observar que o melhor resultado obtido é com a disposição de 90% dos exemplos para treino e os restantes 10% divididos de igual forma entre a validação e o teste.

Agora com a segmentação do dataset por defeito, fomos observar as funções de divisão e chegamos a conclusão de que a melhor função é a usada por defeito para a divisão, ou seja, a função **dividerand**.

Por fim decidimos pegar em todos os resultados obtidos, escolher os melhores de cada observação e juntá-los, para conseguir ter uma boa rede neuronal. Depois de testar 6 redes criadas, determinamos escolher três delas para usar na próxima alínea, sendo as redes

neurônais 3, 6 e 7 representadas na folha pointB do Excel. Todas elas partem do mesmo número de camadas, número de neurónios, função de treino e função de distribuição, sendo as variações na segmentação do dataset e na função de divisão.

Alínea c

Nesta alínea foram usadas as imagens da pasta **test**.

Importante dizer que a rede **net1** é o teste 3 feito na folha PointB do Excel, a rede **net2** é o teste 6 e a rede **net3** representa o teste 6 da mesma folha.

No primeiro teste fomos averiguar o resultado que a rede neuronal dava para este dataset, sem re-treinar a rede. Os resultados rondam os 70% de acertos, sendo a melhor rede, para este caso, a **net2**.

No segundo teste fomos treinar a rede com os exemplos da pasta test e depois testamo-la nos exemplos fornecidos nas pastas **start**, **train** e **test**, separadamente. Os melhores resultados das redes foram com as imagens da pasta **test**, como é obvio. No geral a melhor rede obtida foi a rede **net3**, que obteve os melhores resultados em todos os testes.

Como último teste, treinamos a rede com todas as imagens de todas as pastas fornecidas, e de seguida testemo-las com os exemplos fornecidos nas pastas **start**, **train** e **test**. No geral a melhor rede foi a **net1**, onde os resultados obtidos rondaram os 96%, e foram sempre melhores do que as outras redes.

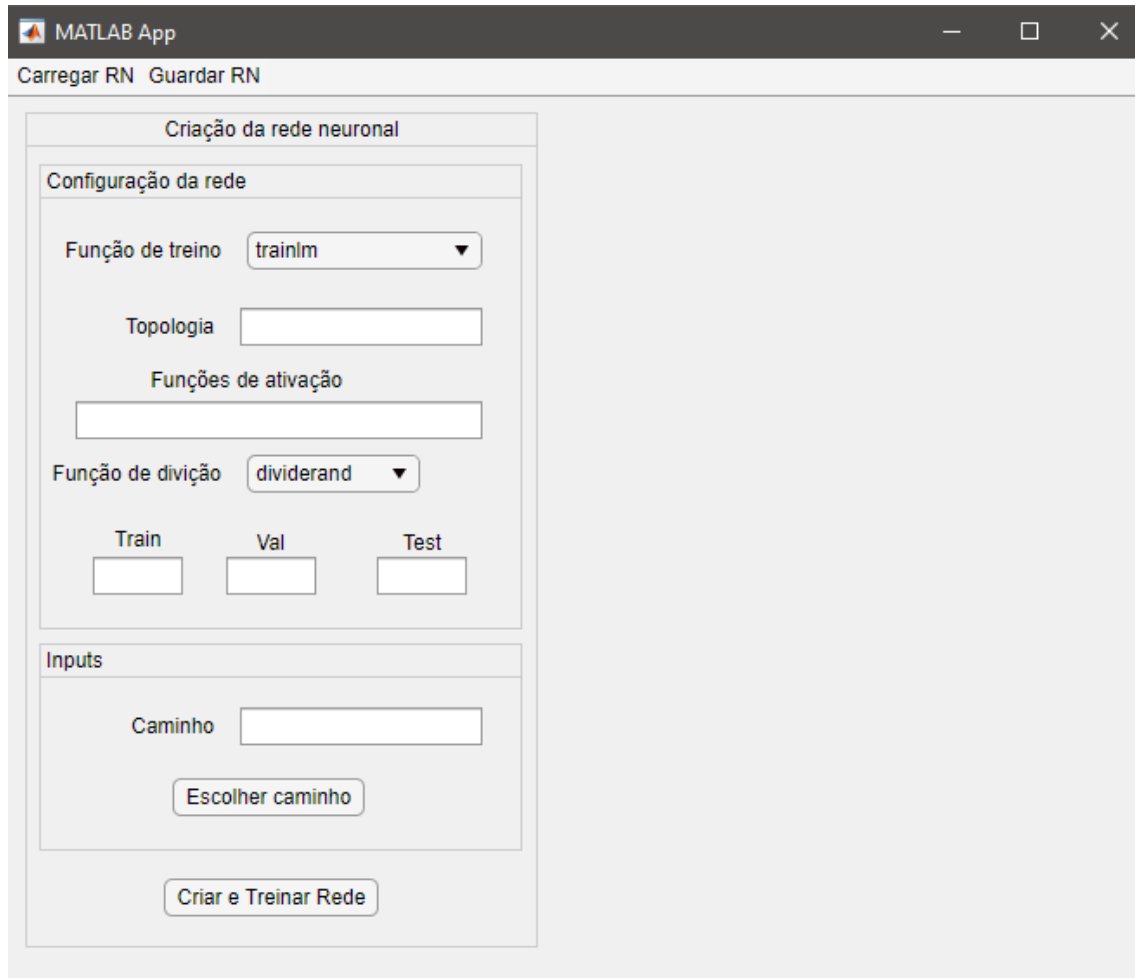
Alínea d

Como último teste, criamos 3 imagens de cada figura geométrica e fomos testar se a melhor rede neuronal da alínea anterior as conseguia adivinhar. A rede neuronal obtida foi usando a **net1**, aplicada ao último caso de teste, gerando uma nova rede neuronal, nomeada **net4**.

Nesta alínea acabamos por cometer alguns erros, como criar figuras geométricas com um traço muito fino, e quando estas eram redimensionadas, perdiam a cor e consequentemente a transformá-las em binário perdia-se a qualidade e a rede neuronal não as conseguia adivinhar. Depois de algum tempo percebemos o erro, criamos imagens e a partir daí o resultado foi incrível, conseguindo chegar 50% de acertos das figuras geométricas. Infelizmente a rede neuronal não consegui-o adivinhar nenhum trapézio nem triângulo, mas consegui-o acertar em todos os círculos e paralelogramas e as outras figuras acertou parcialmente. Outro problema identificado nas imagens é a resolução delas comparada ao espaço livre, como o caso do quadrado (**img16**), em que é muito pequeno e a rede neuronal confunde-o com um círculo.

Interface Gráfica

Infelizmente não tivemos tempo para a criação da interface gráfica por completo, mas fica um pequeno protótipo do desenho.



Conclusão

Com este projeto conseguimos aprender ainda mais sobre redes neuronais, a sua aplicação e eficiência. Consideramos que conseguimos obter resultados bons e no final obtivemos uma rede neuronal minimamente eficiente, para poder apresentar novos casos, que ela nunca viu.

Alem disso também podemos observar o processamento de imagens e a sua dimensão, no nosso caso tivemos de redimensionar para o menor tamanho possível (aconselhado), para poder ter resultados mais rápidos, e infelizmente, menos eficientes.