



Ano 2022/2023

Licenciatura em Engenharia Informática

Conhecimento e Raciocínio

Relatório Trabalho Prático

Francisco Carvalho – 2019129635

João Fernandes – 2018011501

1. Introdução.....	3
2. Decisões tomadas	3
2.1. Pré processamento das imagens.....	3
2.2. Targets	3
2.3. Testes	4
2.4. Dados guardados.....	4
3. Treino e Estudo Estatístico.....	4
3.1. Pasta Start	4
3.2. Pasta Train	4
3.2.1 Treino de modelo unico: numeros + operadores.....	5
3.2.2 Treino de modelo distinto: numeros e operadores	5
3.3. Dataset	7
4. Interface Gráfica	8
5. Conclusão	8
6. Bibliografia	9

1. Introdução

Este trabalho foi realizado no âmbito da Unidade Curricular de Conhecimento e Raciocínio, tendo como objetivo o estudo de redes neuronais para identificação de números (0 a 9) e operadores (+, /, x, -), de modo a fazer uma app parecida com uma calculadora, isto, com recurso a uma rede neural do tipo feed forward.

2. Decisões tomadas

2.1. Pré processamento das imagens

De modo a otimizar a nossa aplicação, redimensionámos as imagens para uma resolução de 30x30 (previamente 150x150). Depois, transformámos as imagens para binário e guardámo-las em matrizes binárias.

Desenvolvemos funções de processamento para processar as imagens das pastas start e train. Esse processamento começa por redimensionar a imagem para um tamanho de 30x30 pixels, que posteriormente são binarizadas para obtermos cada uma das imagens em código binário. Com isto, iremos obter uma matriz com os códigos binários de cada pasta.

2.2. Targets

Para os targets decidimos fazer uma matriz com vetores com os números e operadores (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, /, x, -).

Na pasta "Start" a matriz dos targets ficou com dimensão 1x70, e na pasta "Train1" ficou com 1x700.

2.3. Testes

Decidimos utilizar os seguintes parâmetros para os testes:

- Epochs: 20, e para modelos mais rápidos, utilizamos 1000.
- Número de treinos por configuração: 10.
- As demais configurações foram alteradas de acordo com o Excel.

2.4. Dados guardados

Para cada configuração, guardamos o melhor modelo dos 10 treinos na pasta “networks”.

3. Treino e Estudo Estatístico

3.1. Pasta Start

Esta pasta foi usada para treinar a rede criada usando 5 imagens. Mesmo com poucas imagens para treinar o modelo, conseguimos uma accuracy global de 81%.

Runs	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	Total
Accuracy	80	82,9	81,4	82,9	67,1	84,3	84,3	81,4	88,6	77,1	81

3.2. Pasta Train

Foi aqui que fizemos a maior parte dos testes, com configurações diferentes.

Na configuração por defeito obtivemos 77% de accuracy global e 45% de precisão de teste.

Configuração por defeito	1	10	tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	77	45
--------------------------	---	----	-----------------	---------	--------------------------------	----	----

3.2.1 Treino de modelo unico: numeros + operadores

A melhor configuração neste caso foi a configuração 6 da tabela onde foram alteradas as funções de ativação, com as seguintes configurações:

- Número de camadas escondidas: 1
- Número de neurónios: 10
- Funções de ativação: tansig,softmax
- Função de treino: trainlm
- Divisão dos exemplos : 0.7, 0.15, 0.15

Com isto, obtemos uma precisão global de 90% e uma precisão de teste de 70%.

Conf6	1	10	tansig, softmax	trainlm	dividerand = {0.7, 0.15, 0.15}	90	70
-------	---	----	-----------------	---------	--------------------------------	----	----

3.2.2 Treino de modelo distinto: numeros e operadores

Neste caso, escolhemos a tabela que tinha a configuração de melhor desempenho (tabela das funções de ativação) e realizámos os testes para números e operadores separadamente, obtendo as seguintes tabelas:

Treino dos números							
Conf1	1	10	logsig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	75	57
Conf2	1	10	tansig, logsig	trainlm	dividerand = {0.7, 0.15, 0.15}	47	40
Conf3	1	10	logsig,softmax	trainlm	dividerand = {0.7, 0.15, 0.15}	91	74
Conf4	1	10	purelin, softmax	trainlm	dividerand = {0.7, 0.15, 0.15}	61	45
Conf5	1	10	softmax, softmax	trainlm	dividerand = {0.7, 0.15, 0.15}	45	39
Conf6	1	10	tansig, softmax	trainlm	dividerand = {0.7, 0.15, 0.15}	90	72

Treino dos operadores							
Conf1	1	10	logsig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	93	81
Conf2	1	10	tansig, logsig	trainlm	dividerand = {0.7, 0.15, 0.15}	90	79
Conf3	1	10	logsig, softmax	trainlm	dividerand = {0.7, 0.15, 0.15}	94	81
Conf4	1	10	purelin, softmax	trainlm	dividerand = {0.7, 0.15, 0.15}	72	70
Conf5	1	10	softmax, softmax	trainlm	dividerand = {0.7, 0.15, 0.15}	82	69
Conf6	1	10	tansig, softmax	trainlm	dividerand = {0.7, 0.15, 0.15}	94	81

De seguida, fizemos a média da precisão obtida nos testes com estes modelos, resultando nos seguintes dados:

- Configuração 1: Precisão total global de 84% e precisão total de teste 69%
- Configuração 2: Precisão total global de 68% e precisão total de teste 60%
- Configuração 3: Precisão total global de 93% e precisão total de teste 78%
- Configuração 4: Precisão total global de 67% e precisão total de teste 57%
- Configuração 5: Precisão total global de 63% e precisão total de teste 54%
- Configuração 6: Precisão total global de 92% e precisão total de teste 77%

Assim, comparando com os valores do treino de modelo unico, pudémos concluir que a rede neuronal com a configuração 3 usando o modelo de treino distinto era a mais precisa.

Média obtida com modelos separados.

Precisão total Global	Precisão total Teste
84	69
68	60
93	78
67	57
63	54
92	77

Média obtida com modelos combinados.

Precisão total Global	Precisão total Teste
77	45
23	19
89	68
50	38
40	34
90	70

3.3. Dataset

Criámos um dataset para cada dígito e operador composto por 3 imagens cada. De seguida, à semelhança das imagens da pasta “Start” e “Train1”, transcrevêmo-las para matrizes binárias. Por fim, de modo a classificar o nosso dataset, usámos a melhor rede neuronal obtida, com as seguintes configurações:

- Número de camadas escondidas: 1
- Número de neurónios: 10
- Funções de ativação: logsig, softmax
- Função de treino: trainlm
- Divisão dos exemplos : 0.7, 0.15, 0.15

Com isto, obtemos uma percentagem de sucesso de 90% para os números e 40% para os operadores.

4. Interface Gráfica

Esta aplicação permite carregar individualmente dados dos datasets, detetando quais são os caracteres e atuando assim como uma calculadora simples, tal como mostra a seguinte figura



5. Conclusão

Percebemos que a configuração dos parâmetros das redes desempenha um papel crucial no seu desempenho, e essa configuração varia dependendo do problema específico. Além disso, o pré-processamento adequado das imagens é de extrema importância, pois permite que a rede se concentre nas características relevantes para o problema em questão, como a forma dos caracteres.

Em resumo, consideramos este trabalho sobre redes neurais um sucesso, tanto em termos de compreensão e aplicação dos conceitos como no desenvolvimento de redes neurais com desempenho satisfatório.

6. Bibliografia

mathworks.com/help/matlab/

Moodle da UC de Conhecimento e Raciocínio