

FASE I

Inteligência Computacional 23/24 Licenciatura em Engenharia Informática

Dinis Meireles de Sousa Falcão / <u>a2020130403@isec.pt</u>

Kevin Fernando Pereira Rodrigues / <u>a2013010749@isec.pt</u>

ÍNDICE

- ☐ ANÁLISE E RECOLHA DE DADOS
- □ PROJETO
- □ DEFINIÇÃO DE UM MODELO MLP
- □ RESULTADOS DO TREINO
- □ RESULTADOS DO TESTE
- □ CONCLUSÃO

ANÁLISE E RECOLHA DE DADOS

- □ 24335 imagens (17034 imagens de treino, 7301 imagens de teste);
- ☐ 4055 imagens de cada classe;
- ☐ 6 classes: Buildings (1), Forests (2), Mountains (3), Glacier (4), Street (5) e Sea (6);
- \square Imagens .jpg com tamanho 150 x 150;
- □ https://www.kaggle.com/datasets/nitishabharathi/scene-classification

PROJETO

- ☐ Projeto desenvolvido em MATLAB R2023b;
- ☐ Contém os seguintes ficheiros:
 - □ images
 - □ resized_images
 - □ resizedteste_images
 - □ treinoRedeMLP.m
 - □ testeRedeMLP.m
 - □ train.csv
 - □ test.csv

DEFINIÇÃO DE UM MODELO MLP

```
label = label + 1;

nimages = length(label);

images = zeros([[28, 28], nimages]);

for i = 1:nimages
    image_name = DataTable.image_name{i};
    image_path = fullfile('images', image_name);
    img = imread(image_path);
    if size(img, 3) == 3
        img = rgb2gray(img);
    end
    img = imresize(img, [28, 28]);
    images(:, :, i) = img;
    %imwrite(img, fullfile('resized_images', DataTable.image_name{i}));
end
```

Fig.1 - Redimensionamento de imagens

```
num_classes = 6;
num_samples = length(label);
one_hot_labels = zeros(num_classes, num_samples);

for i = 1:num_samples
    class = label(i);
    one_hot_labels(class, i) = 1;
end
```

Fig.2 - Conversão para One-Hot-Encoding

```
%plotconfusion(one_hot_labels, classes);
classes_vector = classes;
[~, one_hot_labels_vector] = max(one_hot_labels);
C = confusionmat(one_hot_labels_vector, classes_vector);
confusionchart(C);
```

Fig.3 - Matriz de Confusão

DEFINIÇÃO DE UM MODELO MLP

```
net = patternnet(10);
net.layers{end}.size = 6;
net = train(net, images_reshaped, one_hot_labels);
```

Fig.4 – Definição da Rede e Treino

```
y = net(images_reshaped);
classes_pred = vec2ind(y);
```

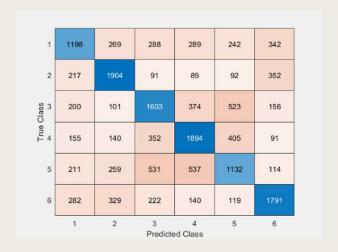
Fig.6 - Teste de Rede

```
images_reshaped = reshape(images, [], size(images, 3));
images_transposed = images_reshaped';
%label_transposed = label';
```

Fig.5 - Matriz 3D para 2D

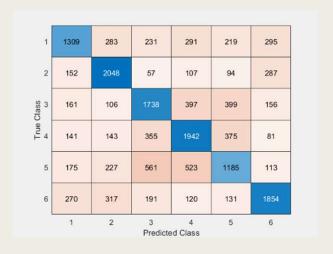
100 neurónios (54 épocas)

Accuracy: 55.90%



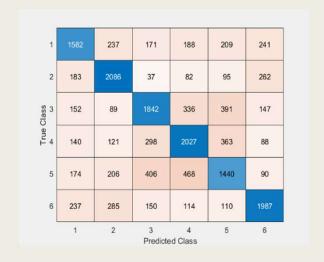
200 neurónios (78 épocas)

Accuracy: 59.15%



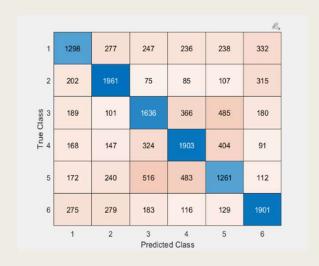
300 neurónios (103 épocas)

Accuracy: 64.37%



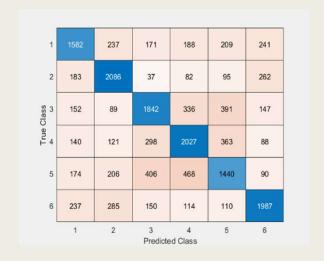
400 neurónios (102 épocas)

Accuracy: 63.37%



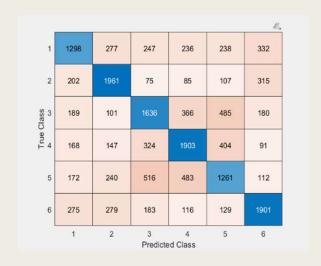
500 neurónios (103 épocas)

Accuracy: 64.78%



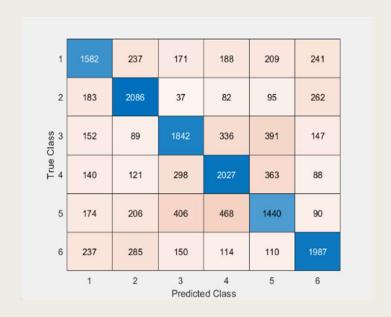
750 neurónios (120 épocas)

Accuracy: 63.37%



1000 neurónios (133 épocas)

Accuracy: 65.49%



RESULTADOS DO TESTE

- ☐ Patternnet com 300 neurónios;
- ☐ Com 1000 neurónios é superior, mas não é uma diferença significativa;
- ☐ Accuracy de teste para as 55 primeiras imagens: 40.00%
- ☐ 22 imagens corretamente classificadas.

CONCLUSÃO

- ☐ Resultados inferiores ao previsto;
- ☐ Accuracy da função de Treino baixa, para a quantidade de imagens;
- ☐ Taxa de sucesso do Teste abaixo de 50%;
- ☐ Várias dificuldades e desafios;
- ☐ Ainda assim, pretendemos que daqui para a frente consigamos encontrar as melhores soluções para aumentas estas taxas.

#