

Instituto Superior Técnico
Departamento de Engenharia Electrotécnica e de Computadores

Machine Learning

3rd Lab Assignment

Report

Shift: Fri. 17:00h Group Number: 7
Number: 83916 Name: João Manuel Prata Morais
Number: 97236 Name: Francisco Rabaça Moller Freiria

Introdução

Este relatório foi feito no âmbito da unidade curricular Aprendizagem automática e tinha como principal objetivo aplicar o conhecimento teóricos quanto a redes neuronais.

Uma rede neuronal é vista como uma série de algoritmos que por meio de um conjunto de dados, são projetados para reconhecer padrões. A forma de agrupar os dados, ou seja, padronizá-los, simulam a maneira do cérebro humano analisar e processar a informação. Os padrões numéricos reconhecidos por uma rede neuronal são agrupados em vetores, mas devem ser traduzidos para dados do mundo real.

1.1.Data

Começamos por carregar e analisar os ficheiros. Estes continham 3000 imagens de treino e 500 imagem para teste.

Fizemos a conversão da matrix de labels para codificação one-hot. Neste formato representativo da matrix de labels, esta terá 10 colunas e o componente que corresponde à classe do padrão vai ter o valor 1 e todos os outros vão ter o valor 0.

Após a separação do conjunto de dados de treino e dos dados de teste, criamos um outro conjunto de dados, o dataset de validação. Este dataset foi criado com a finalidade de verificar a eficácia da rede quanto à sua capacidade de generalização durante o treino da rede. Este conjunto de dados corresponde a 30% dos dados de treino.

1.2.MLP

Uma MLP é arquitetura de redes neurais que está organizado por layers e com múltiplas Unidades que são compostas por funções de ativação contínuas e diferenciáveis. Cada unidade é conectada entre si pode esta pertencer á mesma layer ou a uma layer seguinte.

Após uma análise dos dados, em que se procedeu a separação e criação dos dataset's de treino, teste e validação, passou-se à criação de uma rede neuronal MLP.

Criou-se uma primeira camada, flatten layer, que converte imagens 2D para 1D.

Adicionamos hidden layers, uma com 64 neurónios e uma com 128 neurónios. Cada neurónio capta sinais de entrada probabilísticos. São ótimos mecanismos de identificação de edges e algumas formas no reconhecimento de imagens.

Criamos uma última camada, softmax layer. A função softmax é uma generalização da função logística, que esmaga os valores num determinado intervalo permitindo assim criar um resultado de classificação.

Os resultados podem ser observados na figura 1.

Layer (type)	Output Shape	Param #
flatten_4 (Flatten)	(None, 784)	0
dense_9 (Dense)	(None, 64)	50240
dense_10 (Dense)	(None, 128)	8320
dense_11 (Dense)	(None, 10)	1290
Total params: 59,850		
Trainable params: 59,850		
Non-trainable params: 0		

Figura 1 : Neural network MLP with Early Stopping

De seguida utilizou-se um meio de regularizar o “learner”. Aplicamos o Early Stopping como forma de regularização para evitar ajustes excessivos ao treinar um “learner”, iterativamente. Atualiza o “learner”, de modo a ajustá-lo aos dados de treino a cada iteração. Isso melhora o desempenho do “learner” em dados fora do conjunto de treino. Esta regularização pode levar a um maior erro de generalização. As regras do Early Stopping fornecem orientação sobre quantas iterações podem ser executadas. Assim ajustamos o Early Stopping até a perda de validação não estar e diminuir.

Ajustamos a rede neuronal aos dataset's de treino e de teste em que definimos como critério de paragem de treino do modelo 400 iterações, mas o uma vez que definimos o callback como Early Stopping o modelo não passava das 20 épocas porque o método atingia um estado obsoleto por não sei mais eficiente. Os resultados podem ser observados no gráfico da figura 2.

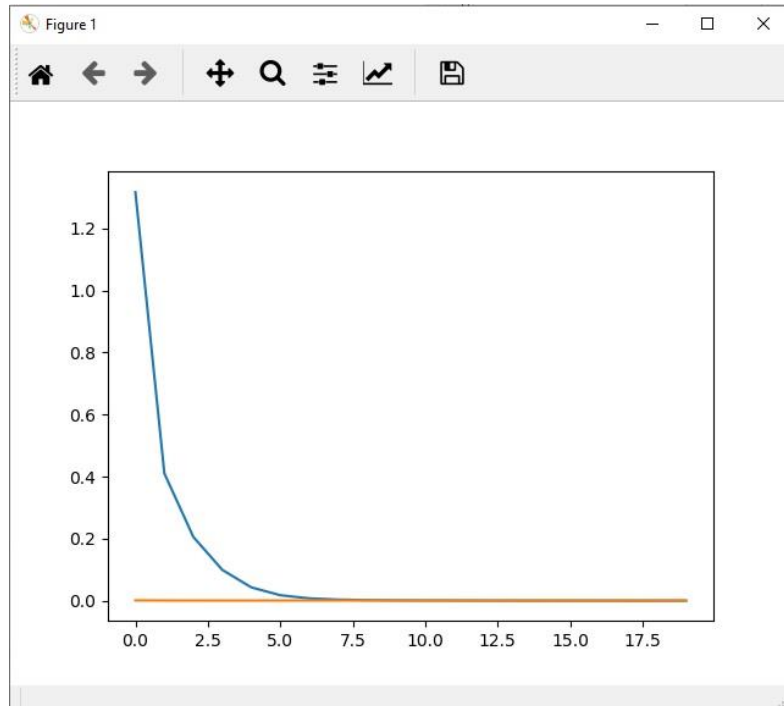


Figura 2 :: Loss and Validation Loss per epoch

Ajustamos ainda o modelo aos dataset's não aplicando o Early Stopping como callback.

Layer (type)	Output Shape	Param #
=====		
flatten_5 (Flatten)	(None, 784)	0
dense_12 (Dense)	(None, 64)	50240
dense_13 (Dense)	(None, 128)	8320
dense_14 (Dense)	(None, 10)	1290
=====		
Total params: 59,850		
Trainable params: 59,850		
Non-trainable params: 0		

Figura 3 : Neural network MLP without Early Stopping

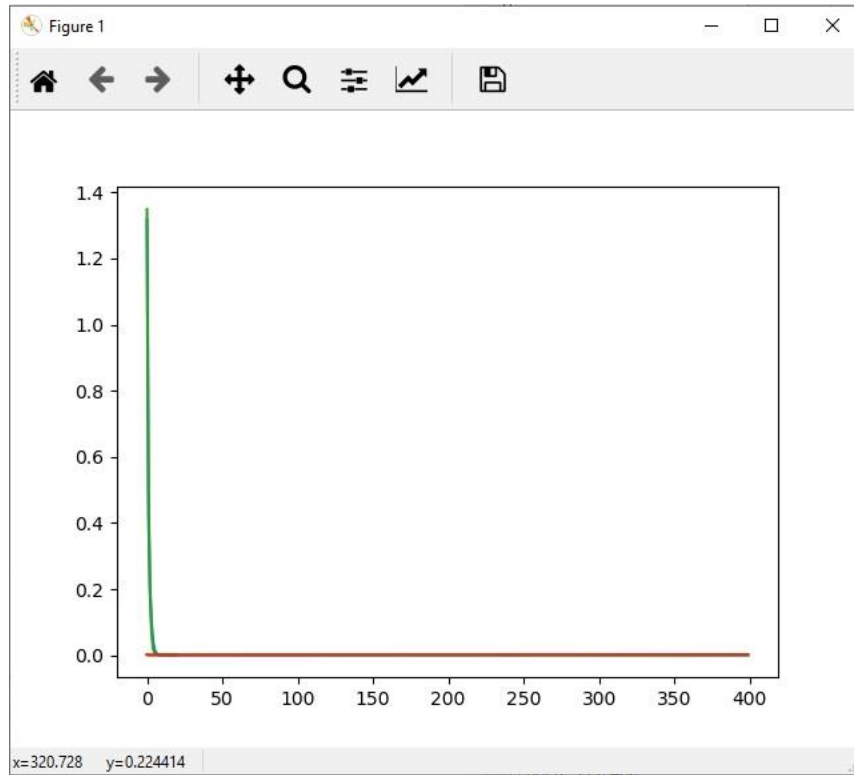


Figure 4 : Loss and validation loss per Epoch

1.3.CNN

Neste ultimo ponto, o objetivo era criar uma Convolutional neural network (CNN) que é uma classe de rede neuronal artificial do tipo feed-forward . Esta rede neuronal tende a procurar um nível mínimo de pre-processamento quando comparada a outros algoritmos de classificação de imagens. Isto significa que esta rede "aprende" os filtros que num outro algoritmo tradicional precisariam ser implementados de forma manual. Essa independência de um conhecimento anterior e da interferência humana no desenvolvimento das suas funcionalidades básicas é considerada a maior vantagem da sua utilização.

Numa primeira fase foi criada a CNN com duas Convolutional Layers intercaladas com duas MaxPooling2D layers. Foram ainda adicionadas uma flatten layer e uma dense layer. Por fim é adicionada uma camada softmax que, como referido anteriormente, limita o output da função no intervalo entre 0 e 1 que permite que estes sejam tratados como uma probabilidade. Seguidamente foi feito o sumario da network sendo o resultado apresentado na Fig.5 .

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 26, 26, 16)	160
max_pooling2d_3 (MaxPooling2D)	(None, 13, 13, 16)	0
conv2d_4 (Conv2D)	(None, 11, 11, 32)	4640
max_pooling2d_4 (MaxPooling2D)	(None, 5, 5, 32)	0
flatten_6 (Flatten)	(None, 800)	0
dense_15 (Dense)	(None, 64)	51264
dense_16 (Dense)	(None, 10)	650
Total params: 56,714		
Trainable params: 56,714		
Non-trainable params: 0		

Figura 5 : Neural network CNN with Early Stopping

A network foi seguidamente ajustada aos datasets de treino e de teste . Foi usada a função Early Stopping que tinha sido usada para o caso do MLP. Os resultados obtidos quanto ao 'loss' e ao 'validation loss' estão representados no gráfico da Fig.6.

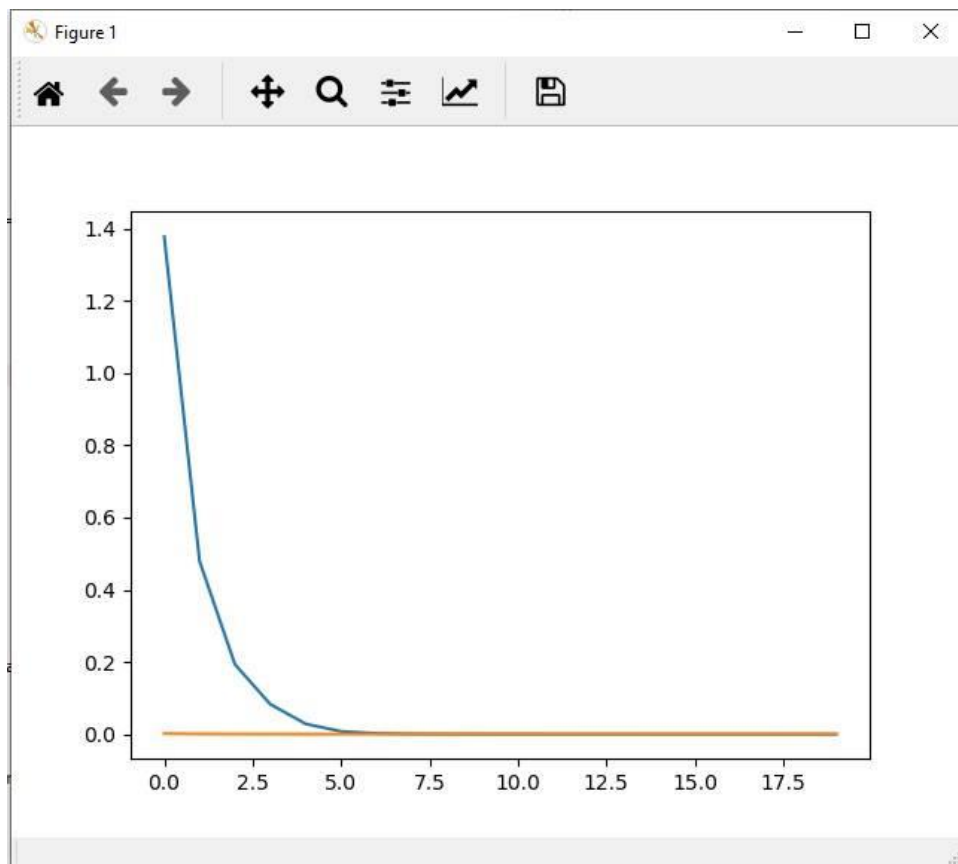


Figura 6 - Loss and validation loss per Epoch

Conclusões

Com os resultados obtidos é possível concluir que a “evolution loss” é maior e decresce menos no caso do uso do Early Stopping. É também possível observar que o tempo de execução é muito maior no caso em que o Early Stopping não é usado, precisando de passar por todos os 400 passos enquanto que no caso com Early Stopping apenas passa por 18/19 passos. Podemos então concluir que o uso do Early Stopping é uma mais valia para o tratamento deste tipo de dados. Comparando agora os resultados obtidos no caso do MLP com Early Stopping e do CNN, é possível observar que o número de passos necessários é bastante semelhante nos dois casos . Quanto ao tempo de execução de cada passo, no caso do CNN, cada passo demora muito mais tempo a ser feito do que no caso do MLP e isto deve-se ao facto de haver mais layers no caso do CNN. Em termos de ‘validation loss’ , pode-se observar que esta é muito menor no caso do CNN. Por fim , pode concluir-se que no caso do CNN esta apresenta melhor performance do que o MLP, ao contrario do que foi referido no relatório do laboratório anterior.