Aprendizagem Automática
Laboratório 3: **Redes Neuronais**

N.º:90007 Nome: Alice Rosa
N.º:90026 Nome: Aprígio Malveiro

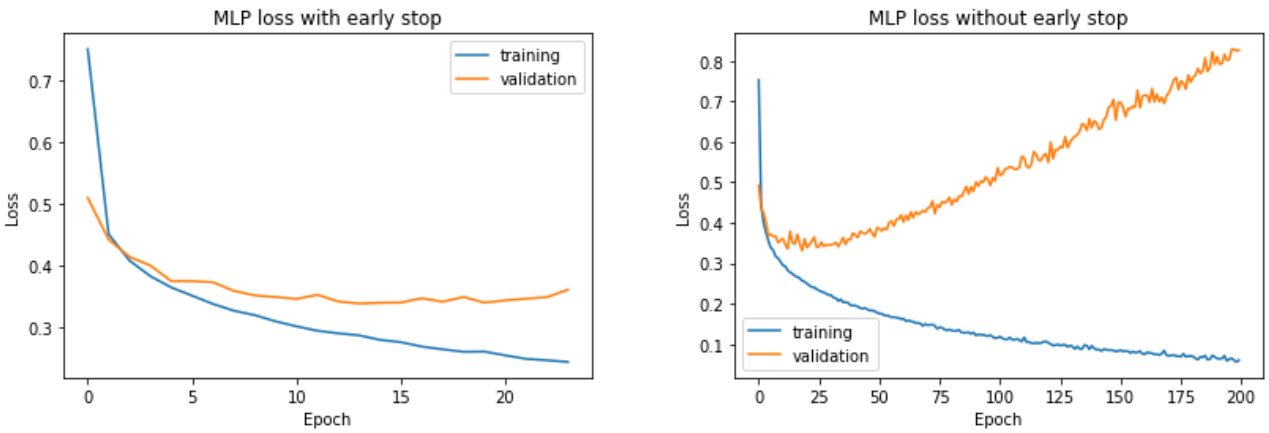
Turno: 3ª feira - 14h00

1.4.1 Existência de *overfitting* nos modelos treinados

A escolha do número de *epochs*, onde um *epoch* consiste num ciclo completo de treino com o conjunto de treino fornecido, é importante porque se este for muito elevado o modelo adapta-se demasiado aos dados com que está a ser treinado e não vai ter uma boa prestação para outros.

A escolha deste número pode ser feita da seguinte forma, começa-se por definir um conjunto de dados de validação, diferentes dos dados de treino, e aplica-se ao modelo a ser treinado. De seguida, utiliza-se o valor de perda obtido para estes dados como um monitor que indica o momento a partir do qual o modelo já não está a melhorar, isto é, em que a perda começa a aumentar em vez de diminuir, e a partir do qual se verifica a existência de *overfitting*.

O método descrito é implementado no Python com a função *EarlyStopping* (ES). Nas figuras 1-(a), 1-(b) e 2 apresenta-se a perda em função do número de *epochs* para o modelo MLP com e sem *Early Stopping* e para o modelo CNN com *Early Stopping*, respectivamente.



(a) Com *Early Stopping* (b) Sem *Early Stopping*

Figura 1: Perda do modelo MLP

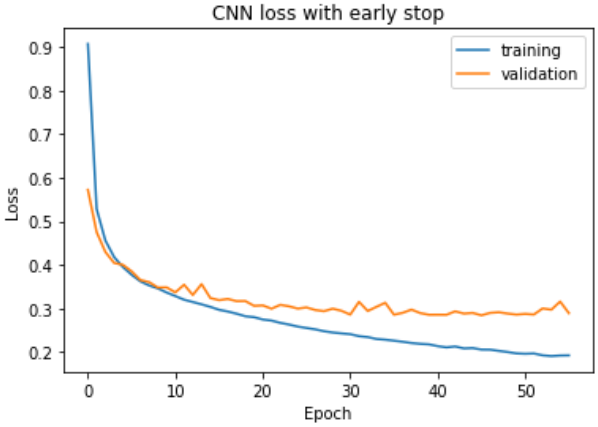


Figura 2: Perda do modelo CNN com *Early Stopping*

A partir das figuras 1-(a) e 1-(b), pode-se concluir que o número de *epochs* necessários para obter o melhor *fit* é aproximadamente 34, depois disso ocorre *overfitting* do modelo em relação aos dados de treino. Na figura 2, verifica-se que este número de *epochs* é aproximadamente 52, e ainda que o modelo CNN adapta-se melhor aos dados de validação relativamente ao MLP.

Outra forma de evitar o *overfitting* dos modelos é simplificá-los através da diminuição do número de parâmetros, por exemplo.

1.4.2 Diferenças entre MLP e CNN em termos de performance e número de parâmetros

A partir do sumário das redes criadas retira-se o número total de parâmetros utilizados pelos modelos MLP e CNN que são, respectivamente, 27.882 e 15.642. O modelo MLP é composto por *layers fully connected*, onde o número de parâmetros corresponde ao número máximo possível de ligações dentro da rede. Enquanto que o modelo CNN apresenta *layers sparsely connected* onde cada nódulo está ligado apenas a alguns nódulos da camada seguinte, em vez de todos. Isto justifica o facto do número total de parâmetros para o modelo MLP ser bastante superior ao do CNN.

A performance dos modelos é avaliada a partir do seu *accuracy score* e *confusion matrix*, para os dados de teste. Na tabela 1 apresenta-se os *accuracy scores* (precisão) para todos os modelos testados. A partir destes valores e das figuras 1 e 2 verifica-se que a performance do modelo CNN é melhor relativamente ao MLP, tanto no conjunto de teste como o de validação. Tal deve-se, em parte ao facto de o MLP ter de transformar as imagens de 2D para 1D antes de as processarem, perdendo a noção espacial, enquanto que o CNN permite processar os dados directamente em 2D.

No entanto, o CNN precisa de mais tempo para treinar o modelo, apresenta um maior número de *epochs* como se viu anteriormente, que o MLP.

Tabela 1

Modelo	MLP com ES	MLP sem ES	CNN
Precisão (%)	87.13	86.26	89.4

Concluindo, para reconhecimento de imagens o modelo CNN produz melhores resultados que o MLP e com menos memória alocada visto que tem menos parâmetros, contudo demora mais tempo na fase de treino do modelo. Todavia, é de notar pelas figuras 1-(a) e 2-(a), que se treinássemos o modelo CNN com o mesmo número de *epochs* que se utilizou para o MLP, continuava-se a obter melhores resultados para o CNN.

1.4.3 Resultados da Matriz Confusão para o modelo CNN

Tabela 2: Matriz Confusão obtida para o modelo CNN

		Previsto									
	Classes	0	1	2	3	4	5	6	7	8	9
Real	0	828	0	30	23	3	0	107	0	9	0
	1	2	972	2	18	1	0	4	0	1	0
	2	11	0	857	12	46	0	72	0	2	0
	3	11	5	10	927	18	0	27	0	2	0
	4	0	1	73	49	800	0	74	0	3	0
	5	0	0	0	1	0	974	0	16	3	6
	6	118	3	76	27	69	0	690	0	17	0
	7	0	0	0	0	0	20	0	960	0	20
	8	0	0	5	5	2	1	4	4	979	0
	9	1	0	0	0	0	6	0	40	0	953

Na tabela 2 as classes 0 a 9 correspondem às seguintes categorias: "T-shirt/top", "Trouser", "Pullover", "Dress", "Coat","Sandal", "Shirt", "Sneaker", "Bag", "Ankle boot".

A partir da matriz de confusão para o modelo CNN representada na tabela 2, verifica-se que os valores superiores encontram-se na diagonal principal da matriz, como seria de esperar uma vez que estas posições correspondem ao número de imagens classificadas corretamente. Os valores das restantes posições fora da diagonal principal correspondem ao número de imagens classificadas incorretamente. Analisando estas posições, consegue-se identificar as classes que o modelo tem mais dificuldade em identificar corretamente.

Por exemplo, para a classe 6 que corresponde à categoria "Shirt", o modelo classifica 118 imagens como "T-shirt/top", e ainda mais algumas como "Pullover", "Dress", "Coat". Tal era de esperar uma vez que todas estas peças de roupas têm bastantes características em comum. O mesmo acontece entre as categorias "Sandal", "Sneaker" e "Ankle boot".

1.4.4 Comentários das *Activation Images* obtidas

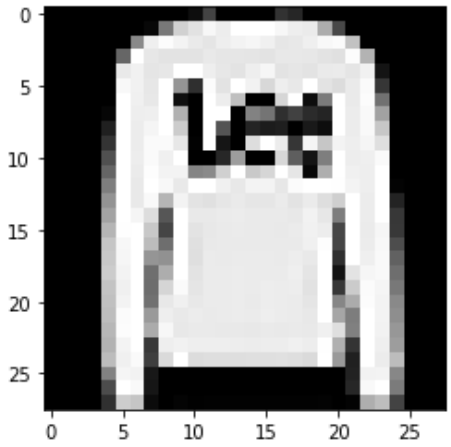
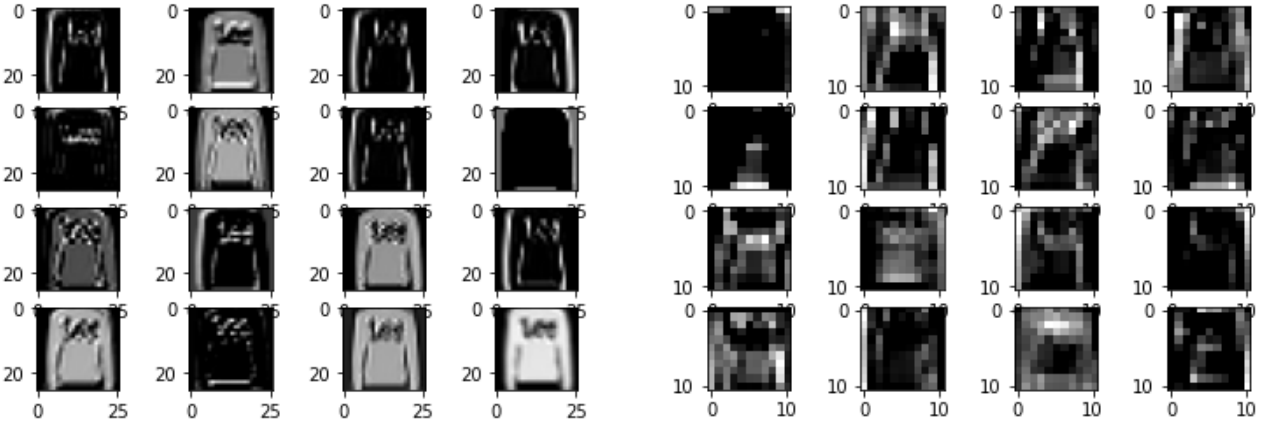


Figura 3: Imagem original



(a) Primeira *Convolutional Layer* (b) Segunda *Convolutional Layer*

Figura 4: *Activation Images* das *Convolutional Layer*

A primeira *convolution layer* recebe a imagem original, figura 3, e transforma num conjunto de 16 imagens, figura 4-(a). Cada pixel de uma *activation image* resulta na convolução de uma sub-matriz (3x3) dos pixeis da imagem original com um filtro. Os 16 filtros diferentes produzem as 16 *activation images* diferentes com dimensões inferiores em relação à imagem inicial, onde em cada uma delas são realçadas características importantes para a classificação das imagens.

Na segunda *convolution layer* o processamento é semelhante mas as imagens obtidas na figura 4-(a), passaram por uma camada de *polling* onde em cada subconjunto de dimensão (2x2) se devolve apenas o pixel com valor máximo, diminuindo a dimensão das imagens. As *activation images* da segunda camada estão representadas na figura 4-(b).

Na figura 4-(a) as imagens ainda são reconhecíveis mas na 4-(b) as imagens tornam-se mais abstractas, apenas se reconhecem padrões. Estas últimas imagens podem justificar o facto do modelo se enganar a classificar peças de roupas com características muito próximas, poderíamos obter melhores resultados se as imagens tivessem mais qualidade (maior número de pixeis).