Clothing classification using CNN

Foglia Milan

///////
For all the labs, I created the following repo if you need to check the code:
https://github.com/Foglia-m/IA
////////

Ex1:

using np.concatenate for the first 5 elements of testX.

Ex2:

No filters	8	16	32	64	128
accuracy	88.35	89.61	89.1	89.36	88.69

Accuracy reaches a maximum with 16 filters, over this value the model might be too complex to fit the data well. Although I could not monitor it well, the convergence time logically increases with the number of filters (3ms/step for 8 filters to 27ms/step for 128 filters).

Ex3: Filters are set at 32 here

No neurons	16	64	128	256	512
accuracy	88.95	90.25	90.12	91.05	90.76

Accuracy reaches a maximum for 256 neurons, it's a classical result for classification tasks, 512 is too much and leads to overfitting of the data.

Ex4: neurons set to 16

No epochs	1	2	5	10	20
accuracy	85.34	87.98	88.95	89.60	90.59

System accuracy is increasing with the number of epochs, we could monitor the test and validation behavior to see how worth it is to increase the number of epochs (mainly to see the difference between the slopes to monitor overfitting) Using early stopping could also resolve automatically the issue.

Ex5: epochs set to 5

Learning rate	0.1	0.01	0.001	0.0001	0.00001
accuracy	84.81	88.95	88.09	82.96	71.32

the optimal learning rate is about 0.01

Ex6:

Dropout	0.1	0.2	0.3	0.4	0.5
accuracy	88.95	89.20	88.68	88.98	88.65

The best dropout value seems to be 0.2 meaning that the model fits the data well but doesn't overfit it as a higher value of dropout is not necessary.

Ex7:
The optimal values are the following:
No filters = 64
No epochs = 20
No neurons = 256
Learning Rate = 0.01
Which gives the following: Accuracy = 91.53%

Ex8: See code for the functions,
We get 2 as output which corresponds to a pullover if we take a look at the labels list.

PART 2: K Folds

These are the accuracy value I get:
Accuracy = 90.98%
Accuracy = 90.91%
Accuracy = 89.72%
Accuracy = 90.65%
Accuracy = 90.76%

I get the following error:
pyplot.plot(histories[i].history['loss'], color='green', label='train')
AttributeError: 'float' object has no attribute 'history'

```
Epoch 5/5
1500/1500 [==============================] - 36s 24ms/step - loss: 0.1802 - accuracy: 0.9334 - val_loss: 0.2436 - val_accuracy: 0.9137
313/313 [==============================] - 2s 5ms/step - loss: 0.2672 - accuracy: 0.9076
Accuracy = 90.76%
Traceback (most recent call last):
  File "C:/Users/milan/PycharmProjects/AIProject/FashionRecognition.py", line 211, in <module>
    main()
  File "C:/Users/milan/PycharmProjects/AIProject/FashionRecognition.py", line 199, in main
    summarizeLearningCurvesPerformances(histories, scores)
  File "C:/Users/milan/PycharmProjects/AIProject/FashionRecognition.py", line 29, in summarizeLearningCurvesPerformances
    pyplot.plot(histories[i].history['loss'], color='green', label='train')
AttributeError: 'float' object has no attribute 'history'

Process finished with exit code 1
```

Ex8,9:
Using padding, i get the following accuracy:

Accuracy = 91.33% which is the best result we can get here, it's logical because it includes the border pixels that can eventually contain some information, especially with low res images such as the ones in the dataset.
(I was already using 64 filters so 8 and 9 were the same)