Instituto Superior Técnico

Aprendizagem Automática

Laboratory 3 - Neural Networks

Authors:
Diogo Moura - nº 86976
Diogo Alves- nº 86980

Lab Shift:
Tuesday 17h-18h30m

29th of October of 2019

1.4 Comments

1) Explain in your own words how Early Stopping works and what is the goal.

During training, weights in the neural networks are updated so the model better fits the training data. For a while, improvements on the training set correlate positively with improvements on the validation set. However, there comes a point where you begin to overfit on the training data and further "improvements"will result in lower generalization performance. This is known as overfitting. After that point, the validation performance gets worse (loss increases and accuracy decreases). Early stopping is introduced to terminate the training before overfitting happens.

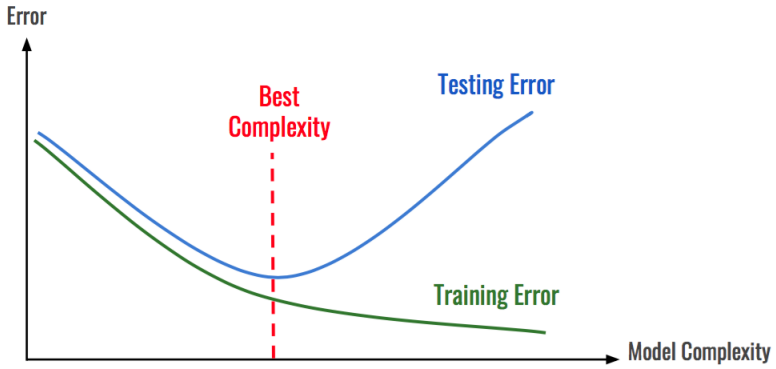

Figura 1: Overfitting

```
keras.callbacks.callbacks.EarlyStopping(patience=15
,restore_best_weights=True)
```

patience: number of epochs that produced the monitored quantity with no improvement after which training will be stopped.
restore_best_weights: whether to restore model weights from the epoch with the best value of the monitored quantity. If False, the model weights obtained at the last step of training are used.

In our case we set patience to 15 and restore_best_weights to True. This way, after 15 epochs, if there is no improvement in the accuracy we restore the weights of the network, to the best fit for accuracy of those 15 weights, so that we assure that "overfitting"did not occur.

2) Comment on the differences in loss evolution, execution time and test set performance between the MLP model with and without Early Stopping.

Without Early Stopping, the loss in the training set is always decreasing, but in the validation set it starts decreasing, then hits a minimum and starts increasing. With Early Stopping this doesn't happen, since the execution is terminated before the loss starts increasing, So, the loss in the situation without Early Stopping is lower in the training set and higher in the validation set, compared to the situation with Early Stopping. This is due to the occurrence of overfitting in the training set, in the situation without Early Stopping.

The execution time is much lower in the situation with Early Stopping, since it only iterates through about 20 epochs instead of the full 400.

The values obtained for accuracy in the test set are: 0.924 with Early Stop and 0.93 without Early Stop. As we can see, although these values are very close, the performance of an independent set is higher in the model in which Early Stopping was not used, due to the fact that we are trying to minimize the cross-entropy and not the loss and also due to the fact that the test dataset is almost two times smaller than the validation dataset, wich makes the result accuracy quite unpredictable on the test dataset.

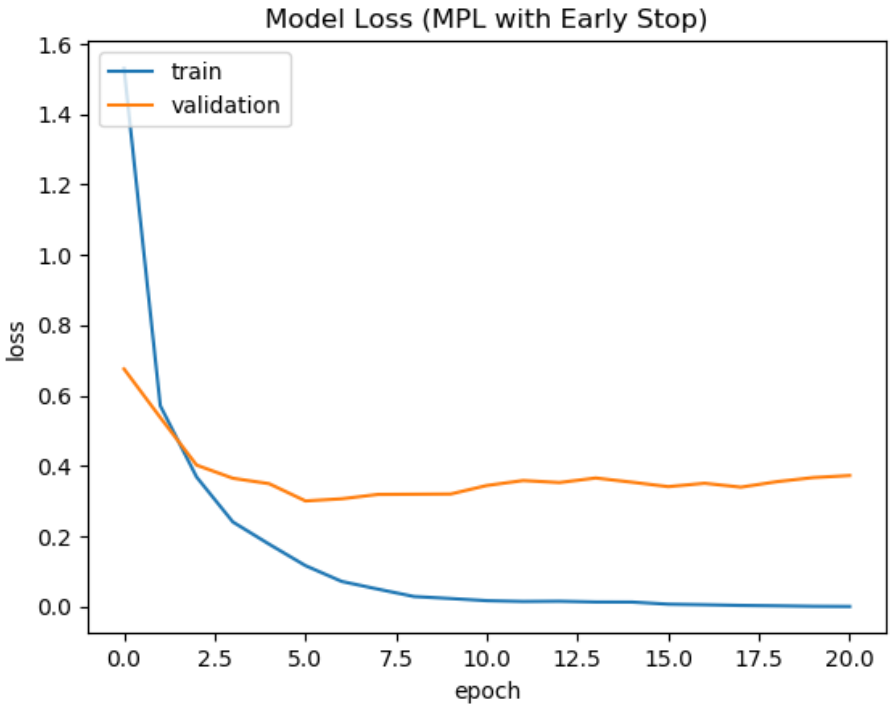

Figura 2: Model Loss (MPL with Early Stop)

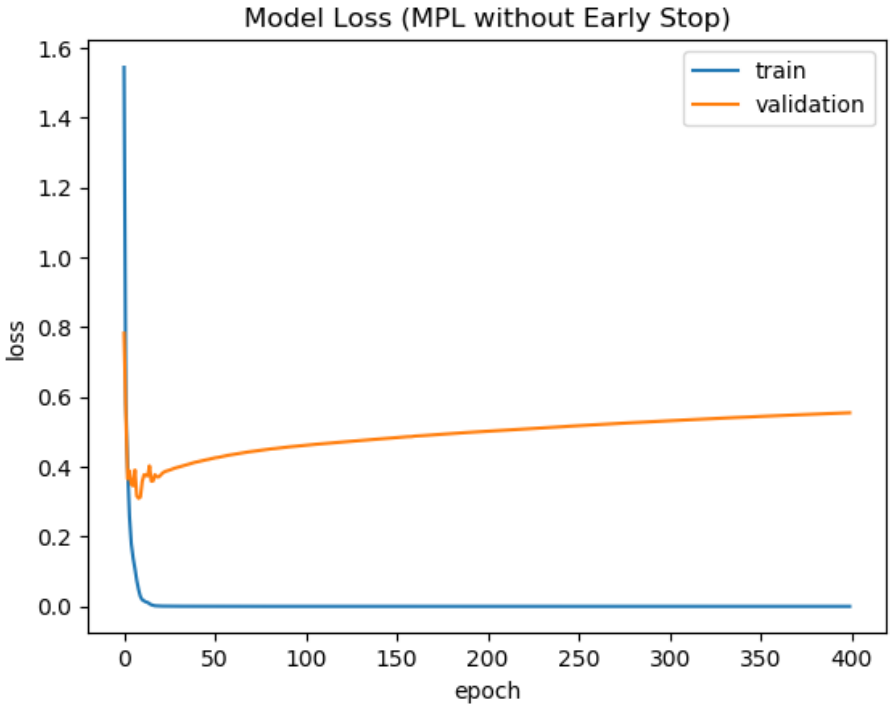

Figura 3: Model Loss (MPL without Early Stop)

3) Comment on the differences between MLP and CNN with Early Stopping, in what regards performance and number of parameters.

Layer (type)	Output Shape	Param #
flatten_2 (Flatten)	(None, 784)	0
dense_5 (Dense)	(None, 64)	50240
dense_6 (Dense)	(None, 128)	8320
dense_7 (Dense)	(None, 10)	1290

Total params: 59,850
Trainable params: 59,850
Non-trainable params: 0

Figura 4: MLP summary

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 16)	160
max_pooling2d (MaxPooling2D)	(None, 13, 13, 16)	0
conv2d_1 (Conv2D)	(None, 11, 11, 32)	4640
max_pooling2d_1 (MaxPooling2	(None, 5, 5, 32)	0
flatten_1 (Flatten)	(None, 800)	0
dense_3 (Dense)	(None, 64)	51264
dense_4 (Dense)	(None, 10)	650

Total params: 56,714
Trainable params: 56,714
Non-trainable params: 0

Figura 5: CNN summary

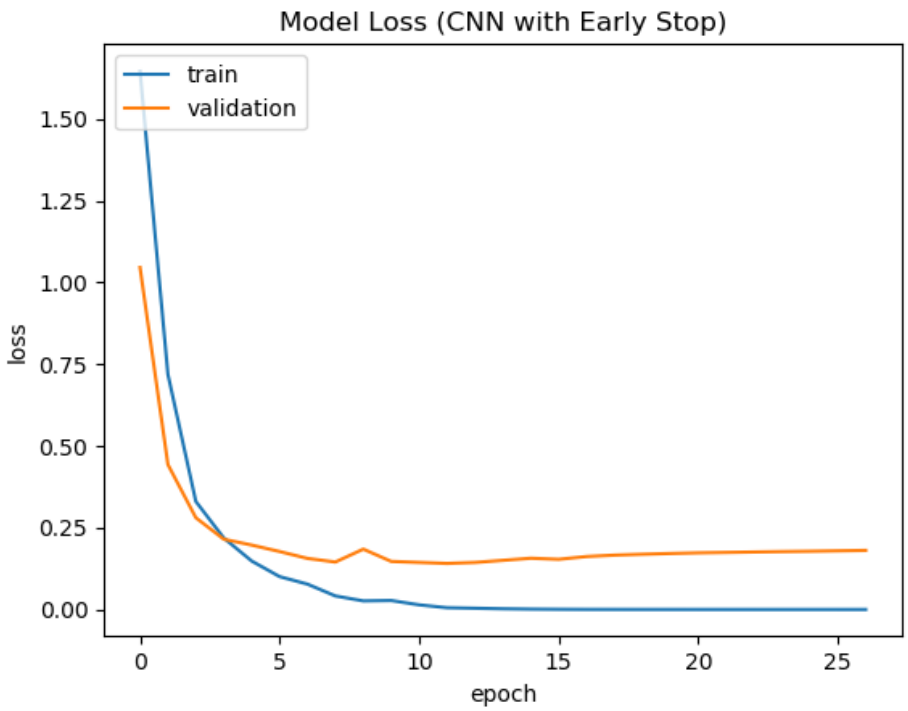

Figura 6: Model Loss (CNN with Early Stop)

As we can see from figures 2 and 6, loss is smaller in the validation dataset, using the CNN.

```
Accuracy (MPL with Early Stop): 0.924
Accuracy (CNN with Early Stop): 0.968
```

Accuracy in the test dataset is also better (bigger) using a CNN, despite using less parameters (56714) compared to the MLP (59850).

This shows that CNN significantly improves the performance, compared to the MLP when it comes to image classification, as expected.
Despite this fact, in terms of computational resources, CNN is more demanding, since it requires the use of convolution, and has more layers. This is evident, since the execution is much bigger with the use of the CNN.