Pattern Recognition Exercise 2

April 2024

Tobias Verheijen, Mattéo Bonvin, Zhihui Wang

1 SVM

For the SVM, we used Sklearn's svm.SVC. This was easy to implement, but without adjusting certain parameters our dataset was too large for the model to run within a reasonable amount of time (e.g. over 6 hours of computation on 16 CPU cores without a result). Once we found the *maxiters* parameter it was much more manageable to run. With a *maxiters* value of 500, we reduced run-time to around 35 minutes per core. The results of the Cross Validation are as follows:

The accuracy on the test set was:

2 MLP

For the MLP, we used PyTorch. We tried first with other librairies but the process would take too much time. With *torch*, the process takes about 30 minutes to find a result. The Epoch result is the following :

For an accuracy of 96.05%

Here is the graph generated :

CV idx	C Value	Score
1	0.1	0.834
1	1	0.822
1	10	0.822
1	100	0.822
2	0.1	0.837
2	1	0.815
2	10	0.815
2	100	0.815
3	0.1	0.829
3	1	0.816
3	10	0.816
3	100	0.816

Table 1: Linear SVM Cross-Validation

CV idx	C Value	Score
1	0.1	0.891
1	1	0.960
1	10	0.967
1	100	0.967
2	0.1	0.888
2	1	0.956
2	10	0.966
2	100	0.965
3	0.1	0.887
3	1	0.959
3	10	0.965
3	100	0.965

Table 2: RBF SVM Cross-Validation

Kernel	C Value	Accuracy
Linear	0.1	0.7981
RBF	1	0.9691

Table 3: SVM Test Set accuracy

Epoch	Train Loss	Validation Loss
1/10	0.5429	0.1659
2/10	0.1832	0.1409
3/10	0.1572	0.1336
4/10	0.1515	0.1420
5/10	0.1424	0.1200
6/10	0.1372	0.1302
7/10	0.1334	0.1420
8/10	0.1287	0.1192
9/10	0.1246	0.1001
10/10	0.1259	0.1032

Table 4: Evolution of Train and Validation Loss

3 CNN

For the CNN, we also used PyTorch library. The process takes 15 minutes to find the result. For an accuracy of 85.71%. Here is the graph generated :

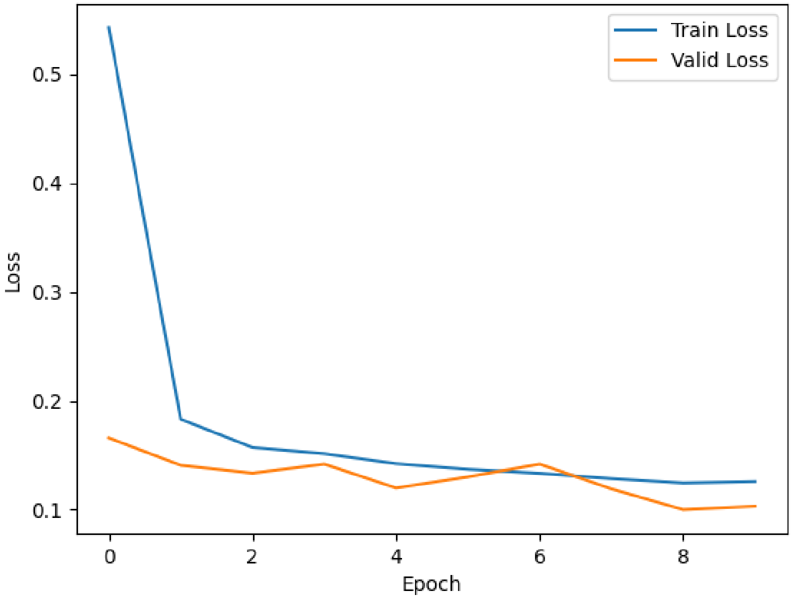

Figure 1: Train and Valid Loss of the MLP

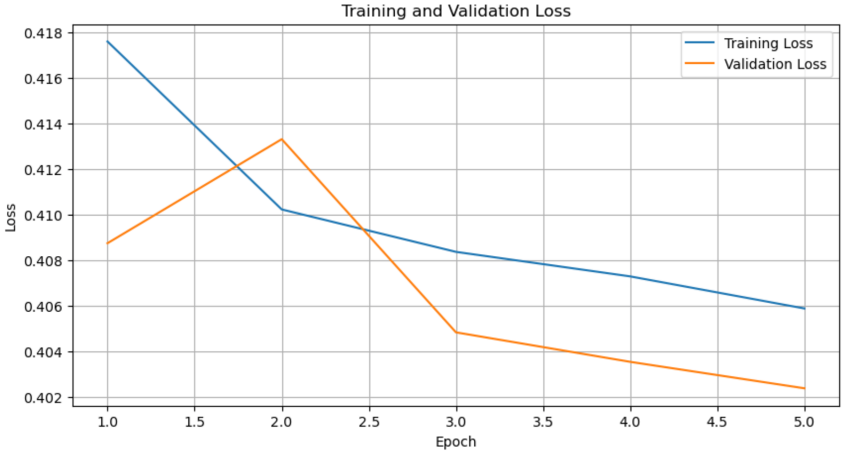

Figure 2: Train and Valid Loss of the CNN