EDAN20 - Assignment 4
A simple language classifier

Hugo Mattsson
hu5174ma-s

October 2024

1 Objectives and dataset

1.1 Objectives

The objectives of the assignment were to use both sklearn and PyTorch to to create neural networks that classify languages.

1.2 Dataset

The dataset used was Tatoeba, downloaded on 2014-10-05.

2 Method and program structure

2.1 Compact Language Detector v3 - CLD3

CDL3 extracts character n-grams from a text and hashes down to an id within a small range. It also computes the relative frequency for each of them within the text. These n-gram ids and frequencies are then in some way averaged down to an embedding and concatenated together to produce the embedding layer.

2.2 My program

My program finds all uni-, bi- and trigrams and their relative frequencies from a sentence. The n-grams themselves are then hashed and, using modulo, reduced to fit into a smaller interval. This modulo reduction creates overlap between different n-gram sizes, so the id ranges for bi- and trigrams are shifted upwards. All the n-gram ids and frequencies are then concatenated to create the final representation of a given sentence.

The difference between my architecture and CDL3 is that I don't perform any sort of avaraging on the n-gram frequencies, they stay as a raw relative sentence frequency.

2.3 scikit-learn vs PyTorch

Both scikit-learn and PyTorch are great libraries/frameworks when doing some sort of machine-learning. The main difference between them is that sklearn seems to be a more high-level library making it easier to use. PyTorch instead is more low-level and therefore expects more of the user, but instead granting finer control and better understanding.

3 Results

We can see that both the sklearn and PyTorch models performed similarly.

3.1 The feature matrix - X

8	0	8	1	0	0
0	0	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	0
3	1	2	1	0	0
4	1	6	1	0	0
4	0	1	1	0	0
5	2	2	0	1	0
2	0	2	1	0	0

Table 1: The feature matrix - X

3.2 sklearn

	Precision	**Recall**	**F1-Score**	**Support**
cmn	1.00	1.00	1.00	9866
dan	0.99	0.97	0.98	9963
eng	1.00	1.00	1.00	10059
fra	1.00	1.00	1.00	10041
jpn	1.00	1.00	1.00	10005
swe	0.97	0.99	0.98	10066
accuracy			0.99	60000
macro avg	0.99	0.99	0.99	60000
weighted avg	0.99	0.99	0.99	60000
Micro F1:	0.9921			
Macro F1:	0.9921			

Table 2: sklearn accurracy

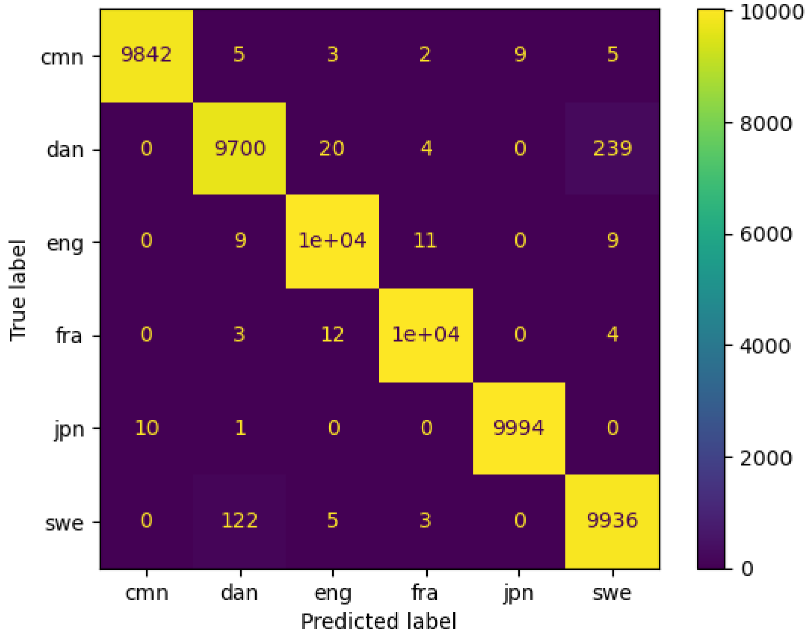
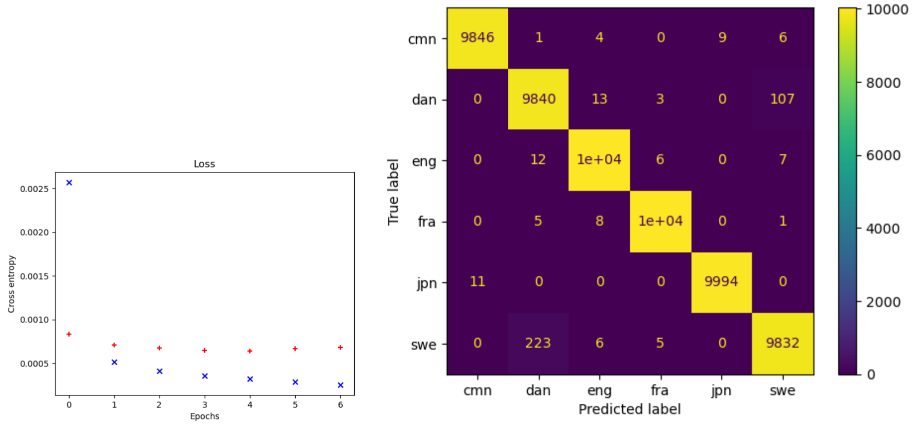


Figure 1: sklearn confusion matrix

3.3 PyTorch

	Precision	**Recall**	**F1-Score**	**Support**
cmn	1.00	1.00	1.00	9866
dan	0.98	0.99	0.98	9963
eng	1.00	1.00	1.00	10059
fra	1.00	1.00	1.00	10041
jpn	1.00	1.00	1.00	10005
swe	0.99	0.98	0.98	10066
accuracy			0.99	60000
macro avg	0.99	0.99	0.99	60000
weighted avg	0.99	0.99	0.99	60000
Micro F1:	0.9929			
Macro F1:	0.9929			

Table 3: PyTorch accuracy



(a) PyTorch Loss

(b) PyTorch Confusion Matrix

Figure 2: PyTorch Loss and Confusion Matrix

4 Conclusion

The objective of this assignment, to create a language classification model was accomplished. Both sklearn and PyTorch were used and they both performed very similarly to each other.