Artificial Neural Networks and Deep Learning - Academic year 2022-2023
Professor Matteo Matteucci

Authors:
Antonio Sgaramella (10665339)
Dario Vernola (10629354)
Francesco Vittorini (10864252)

HOMEWORK 2 REPORT

1- Introduction

The goal of the second Homework was to design a Neural Network capable of classifying time series. Scope of this document is to show the experiments and the process that lead to the final solution, analyzing the choices that have been made during the development phase.
Firstly, we will start by describing the provided dataset and the subsequent preprocessing phase.
Then we will delve into the various steps we followed to find the best suited model for this specific classification problem.
Finally, the obtained performance results will be displayed.

2- Dataset analysis and preprocessing

The provided dataset was composed of 2429 time series, presenting each 36 points and 6 features which belong to 12 different classes.
Analyzing the data distribution among classes we obtained the following results:

- “Wish”: 34,
- “Another": 123,
- "Comfortably": 270,
- "Money": 381,
- "Breathe": 62,
- "Time": 153,
- "Brain": 313,
- "Echoes": 68,
- "Wearing": 120,
- "Sorrow": 777,
- "Hey": 77,
- "Shine”: 51.

This dataset is clearly unbalanced, we tried to assign different weights to the different classes based on the number of samples of that particular class but that led to no improvement.

The dataset has been split into two different subsets: training set and validation set.
The split was performed using the train_test_split function, choosing as split rate 0.2 (80% training and 20% validation) in a stratified manner, that is by maintaining the classes distribution among the two sets.

Multiple preprocessing functions were tried such as MinMaxScaler and MaxAbsScaler but with insufficient accuracy on the validation set, while the StandardScaler function was the only one providing satisfying results hence it was the one used for the final model construction. This type of preprocessing changed the data removing the mean and scaling to unit variance.
Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set.
The same preprocessing technique has been applied to the test set before prediction.
Sometimes it could be interesting to understand what happens when a subset of the features is removed from the dataset, since the absence of such features could help the model in better discriminating the different classes in the dataset. Starting from these considerations we tried to iteratively remove one feature at a time and observe the performance of the model on the validation set. The only interesting result was obtained removing the third feature, which led to a validation accuracy comparable to the one obtained using all the features. Due to the fact that these attempts did not lead to significant improvements in the validation accuracy of the model, this technique was ultimately removed from the final one.

3- Model selection

We started with a simple LSTM network as a feature extractor on top of a fully connected classifier network trying out different architectures but obtaining unsatisfactory results.
Slight improvements were obtained using a Bidirectional LSTM network but the most interesting results were reached using a 1D Convolutional neural network.
As for the other experiments we tried different configurations of the network: we first tried using a small amount of filters for each convolutional layer and then we tried to gradually make the model more complex in order to be able to extract more particular features by increasing the number of filters.
We also tried increasing the number of convolutional layers but such an approach did not lead to any improvement in the performance. This result let us focus more on the two layers version of the network with which we ended up obtaining the most interesting results. Another attempt to increase the performance of the model was carried out by the introduction of batch normalization layers between convolution and relu layers but to no avail.
We also tried combining convolutional layers with bidirectional LSTM layers but it was not ultimately successful.
Further improvements were achieved switching the MaxPooling layer present between the convolutional layers to an AveragePooling one and introducing a dropout layer with neuron switch off probability of 30%.
At the end of the model selection phase the best 1D convolutional network we selected was composed of the following layers:

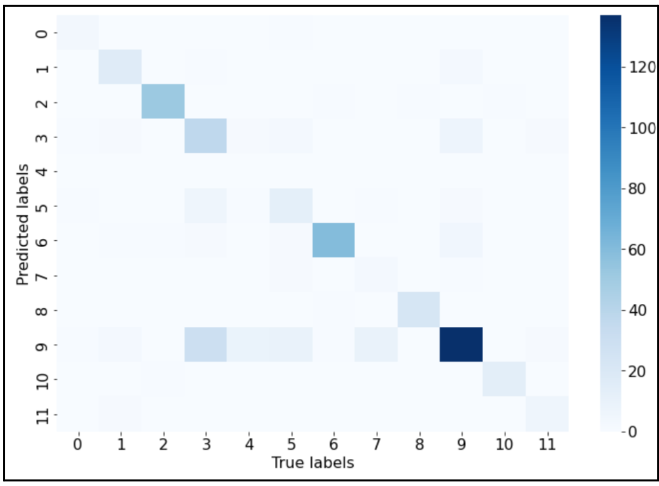
- 1DConv layer, with 512 filters, filter size equal to 3, padding same and ReLU activation,
- AveragePooling1D,
- 1DConv layer, with 512 filters, filter size equal to 3, padding same and ReLU activation,
- GlobalAveragePooling1D,
- Dropout,
- Dense layer with 256 neurons and ReLU activation,
- Output layer with 12 neurons and softmax activation.

4- Training

The training phase was carried out by selecting a batch size equal to 128 and starting with a limited number of epochs, always using EarlyStopping to avoid overfitting. After some attempts we decided to increase the number of epochs and the patience of the EarlyStopping in order to give the model enough time to learn the important features and increase its performance on the validation set. We also tried to reduce the learning rate when the performance of the model on the validation set did not improve, but no interesting results were obtained with this method.
In the attempt to improve the score on the test set we tried to perform the train-validation split without stratification and, to maximize it, we trained our network on the 95% of the dataset provided, which brought us to a score of 73.88% on the hidden test set and 74.75% on the final phase hidden test set. A final attempt was carried out by trying to perform the training on the overall dataset, but no improvement was obtained on the test set because it was difficult to understand when the training should have been stopped without the reference of its performance on a validation set.

5- Model evaluation

The models were evaluated using the confusion matrix and the classification metrics (precision, recall, F1 score and accuracy).




```
Accuracy: 0.7531
Precision: 0.7036
Recall: 0.6386
F1: 0.6601
```

The above figure represents the confusion matrix obtained on the final model previously detailed. These results refer to the model trained over a training set composed of 80% of the entire dataset. The metrics are evaluated over the performance of the model with respect to our validation set (the remaining 20% of the given dataset).