Kaggle competition

Neural Network Challenge spring 2023

Report



Gabriele Scognamiglio - 100494273

Jakub Jerzy Bazyluk - 100502918

Daniele Laporta - 100502703

Bachelor in Data Science and Engineering

Neural Networks

Prof. Daniel Barrejon Moreno

Prof. Pablo Martínez Olmos

Prof. Adrian Vega Delgado

# Brief Kaggle competition description

This [Kaggle competition](https://www.kaggle.com/competitions/neural-networks-challenge-spring-2023/overview) consists of analyzing a time series dataset and classify into 5 different categories.

The dataset is highly imbalanced and contains missing data in "bursts" (a certain number of consecutive samples that are NaN).

# Introduction

In the following sections are described the two models developed, which have an RNN, specifically a LSTM, as a strong common base.

Before explaining them, is important to mention that we did a preprocessing phase where we filled the *NaN* values, used standard scaling and prepared the dataset in order to be input to the model.

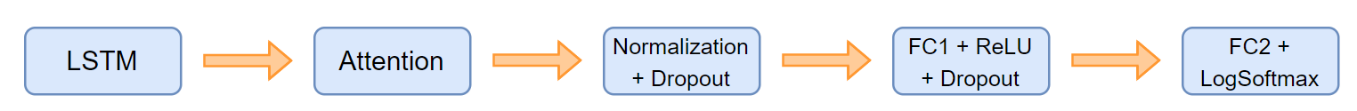
To fill the missing data, we tried to just fill with zeros, or with the median of each data column, or using a function to expand linearly a time series when it encounters consecutive zero values.

All the coding details are available at the [GitHub repository](https://github.com/jakub-bazyluk/Neural-networks-challenge-spring-2023).

# 1st model

The first model implemented is made up of the following steps:

1. LSTM layer, which takes the input file and process it.
2. Attention layer, which takes the output of the LSTM.
3. Normalization layer, where the output of the attention layer is normalized.
4. Dropout layer, to prevent overfitting and improve the generalization ability of the model.
5. Fully-Connected Layer 1 (FC1), which receives the output of the dropout.
6. ReLU Activation, which is applied to the output of the FC1.
7. Dropout, that is applied to the ReLU output.
8. Fully-Connected Layer 2 (FC2), a second FC layer receiving as input the output of the dropout.
9. LogSoftmax, an activation for the final classification.



# 2nd model

The second model implemented differs from the first one because there are two consecutive LSTMs followed by one fully connected layer and dropout.

Immagine che contiene testo, Carattere, logo, schermata

Descrizione generata automaticamente

# Conclusions

The results obtained by both the models are pretty good. We played a lot changing the possible parameters in order to see how the model would perform and on each case the performances are around 90% of correct classification.

To conclude, this challenge has been truly useful to better understand the concepts studied and apply them on a competitive scenario where we needed to always strive to improve the final results to climb positions on the board.