Report

Part 1: Time series

We have performed a regression task on the Individual household electric power consumption dataset. This dataset contains 2075259 samples which are measurements gathered in a house located in Sceaux (7km of Paris, France) between December 2006 and November 2010 (47 months). We will try to predict the future values of the energy consumption and compare the predictions to the actual values.

As for the preprocessing steps:

* We had 2 different attributes for date and time, and we decided to combine them into a single attribute.
* We added an additional column named ‘Energy consumption’, which represents the active energy consumed every minute (in watt hour) in the household by electrical equipment not measured in sub-meterings 1, 2 and 3. This is how it’s calculated: (global\_active\_power\*1000/60 - sub\_metering\_1 - sub\_metering\_2 - sub\_metering\_3)
* There were some missing values which we replaced with 0.

After viewing some graphs which described the data samples, we noticed that the data is overall balanced. We also noticed that the global active power and the intensity have a similar impact on the energy consumption and seems to be of linear nature.

Before constructing a model, we normalized the data for each attribute by computing the mean and std.

We split the data into train and test sets; out of 47 months of samples (sample for each minute), we took 2.5 years for training, and left the rest (1.5 years) for the test set.

As for the model, we added an LSTM layer with 50 neurons, and a Dense layer afterwards. We managed to get a result of val\_loss = 0.1037. The predictions seem to be close to the actual values for the Energy consumption. The score of the model was 0.9147159739131537. The model predicted the energy consumption of the house fairly well. The predicted values successfully corresponded to the energy peaks that occur during the day and week.

We thought of a few ways to improve the model such as adding a dropout layer, adding more neurons to the LSTM layer, or resampling the data from minutes to hours and days. We tried implementing all of these. The best improvement we got as from adding 25 more neurons to the LSTM layer; we managed to get a result of val\_loss = 0.1028. Adding a dropout layer to see if the model was overfitting, the change worsened the results so overfitting was not the case.

Then, we fitted a classical machine learning model to the data to get a better benchmark: Linear Regression Machine Learning algorithm. Here we received a score of 0.90. We saw that most of the values aspire to the diagonal, as we expected. From reviewing the accuracy of this model and the loss of the previous model.

Part 2: Category embeddings

The ELO merchant category recommendation competition included the following data sets: train, test and a few other tables with additional data. The train and test datasets include a few features: first\_active\_month, card\_id which are unique values and 3 numerical features that distribute in a uniform manner.

The other datasets include historical\_transactions, merchants and new\_merchant\_transactions which includes different data that concerns the transactions made by the card\_id from the train and test datasets.

We merged the tables to see the correlation of the data in the train table and the other tables by the card\_id feature. The merge produced a lot of merged data samples as for each card\_id there are a lot of entries with different values, which is hard to normalize and aggregate for the specific card\_id.

The model we constructed takes all the features and embeds them in 5 dimensions. After that the model runs the concatenated data in dense layers to reach one regressed output value.

We trained the model for 7 epochs on the train data with train test split of 0.2. The score in the validation was 0.7926107021029271. We tried to run the model with the merged data of train and historical\_transactions but the distribution of the data in the merged table was not even and was very difficult to normalize to unique values for each card\_id, so we had to remain with the train dataset alone.

We fitted the data in a classical ML linear regression algorithm to compare the results. The ML model was not able to get good predictions only with the data available in the train table.