Deep Learning

Building a Deep Learning model for Churn Prediction

Academic year 2018/2019

Instructions

The objective of this exercise is to build a deep learning model for churn prediction. Churn prediction aims to detect customers intended to leave a service provider. This is a problem of fundamental importance, considering the ever-growing competition in different markets. Companies have become aware that they should put much effort not only trying to convince customers to sign contracts, but also to retain existing clients. This is due to the fact that in the current scenario where people are given a huge choice of offers and different service providers to decide upon, winning new customers is a costly and hard process. Therefore, putting more effort in keeping churn low has become essential for service-oriented companies.

The dataset you will use in this exercise is very popular in the field of Machine Learning for Churn prediction. It contains data related to the clients of a (fictitious) bank that operates in three different countries: France, Spain, and Germany. The bank has experienced an unusually high churn rate and decided to build a deep learning model for a better understanding of the problem and to obtain information about the clients that in the near future are more prone to leave the bank.

For each observation the following information are available:

- RowNumber: progressive number indicating the position of the observation in the file;
- CustomerID: unique client identifier;
- Surname: the surname of the client. This is a textual variable;
- CreditScore: numerical variable used by the bank to evaluate a client. The higher the credit score is, the better is the client for the bank;
- Geography: location of the client. This variable can assume only three values: "France", "Germany", and "Spain";
- Gender: "Male" or "Female";
- Age: the age of the client;

- Tenure: number of years the client is with the bank;
- Balance: the amount of money on the account of the client;
- NumOfProducts: number of products (e.g., insurance, loan, etc.) that the client subscribed with the bank;
- HasCrCard: 0 means no credit card, 1 means that the client has a credit card;
- IsActiveMember: 0 if the client is not an active member, 1 in the opposite case. An active member is a client that performed a banking operation in the last 60 days;
- EstimatedSalary: the estimated salary of the client;
- Exited: this is the target variable you want to predict. 0 means that the client is with the bank, while 1 means that the client left the bank.

You will use Keras and the concept we have seen so far in the course for developing a *binary classification* model for churn prediction.

In particular, the following steps must be performed:

- Analysis of the data and preprocessing. After importing the dataset, the first step consists of data preprocessing. In the context of this exercise, you must remove variables that are not useful for the churn prediction (like RowNumber, CustomerID, and Surname). Subsequently, you must deal with textual variable (Geography and Gender): remember that the Deep Learning models discussed require a numerical input. Finally, you must split your data into training and test observations (for the moment do not consider validation data, we will cover this part in the next exercise) and perform a feature scaling to avoid potential bias caused by the different order of magnitude of the variables. Remember that the feature scaling must use parameters extracted ONLY from the training set. You can never use test data during the training phase!
- Construction of the model. In this second step, you will build the Deep Learning architecture, by "stacking" on the top of each other different Dense layers. In particular, you can start by considering an architecture with three layers, the first two hidden layers will have 6 neurons and the last (output) layer will consist of one single neuron. This is the initial architecture, but feel free to perform more experiments by taking into account a different number of neurons and/or layers. Remember that increasing the number of layers and neurons will slow down the training process. The activation function, the loss function, and the metric are the ones we have seen in the first classes of the course for a binary classification problem. With respect to the optimizer, select one among "adam" and "rmsprop". With these parameters you can call the method "compile" and this concludes this step.

- Once the architecture is built, the training process can start: train your model for 100 epochs. Specify the parameters required by the fit method (training observations, vector with the target values, number of epochs) and train your model.
- Analyze the performance on the test data and compare it (in terms of accuracy) with respect to the training performance.