Advanced Data Science with IBM | Coursera
Capstone Project :
Architectural Decision Guidelines

Daniel PONT
august 2020

Table of contents

1. Introduction and usecase

The present document is the architectural decision guidelines for the advanced data science capstone project (IBM | Coursera).

We'll use the "Human Activity Recognition Using Smartphones Data Set"
[https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones]

This dataset was built thanks to experiments carried out with a group of 30 volunteers within an age bracket of 19-48 years.

Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist.

The aim is to predict the activity performed from the signals measured by the smartphone sensors (accelerometer and gyroscope)

2. Data quality assessment

It's important to note here that the dataset doesn't contain raw data: the signals measured the smartphone accelerometer and gyrosccope have been heavily pre-processed. According to the dataset description provided by the authors :

"The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain. "

For our project we'll check the two following key points :

- There's **no missing values** ;
- There's **no invalid values**. Each variable has been centered and scaled so all values are in the range [-1,1] which is good for building a neural network.

We'll also check the presence of outliers.

3. Feature engineering

Regarding the 561 input variables we'll apply a **PCA** (Principal Component Analysis) on them and keep only the first components. The main goal here is to **remove the noise** in the data. An added benefit will be a dimensionality reduction which makes the training easier and quicker.

We'll also perform a one-hot encoding on the outcome variable in order to replace the vector containing the class to predict (1 = WALKING, 2 = WALKING_UPSTAIRS, 3 = WALKING_DOWNSTAIRS, 4 = SITTING, 5 = STANDING, 6 = LAYING) by a binary matrix. This requirement is necessary to use the outcome as an output of a neural network.

4. Algorithm choice

To establish a prediction baseline we'll use one of the most commonly used and efficient traditional machine learning algorithm : linear SVM.

Then we'll use a deep learning algorithm and build a neural network. Several architectures of networks more or less complex can be used. For instance :

- CNN (Convolutional Neural Network) which is the standard architecture for image recognition problems;
- Long Short-Term Memory (LSTM) is an artificial recurrent neural network (RNN) architecture often used to work with temporal series;
- …

Initially we considered to choose a LTSM architecture since our dataset features were obtained by calculating variables from the time and frequency domain.

But after some thoughts, since the activities we want to classify are quite basic, rather elementary (they don't involve a complex sequence of temporal patterns) we chose a simpler option : a **MLP (Multi Layers Perceptron) neural network**. In order to prevent over-fitting, we added dropout on each dense layer of the network.

5. Framework choice

We chose the **keras** framework because it's quite easy to set up and we didn't need any complex low-level functionality provided by tensorflow or pythorch.

6. Model performance indicator

We explored the dataset and visualized the distribution of the outcome variable. We noted that the distribution of the outcome was quite balanced both in training and testing dataset : each class was observed almost as frequently as any other one. So we chose the accuracy metrics to assess our model performance. Accuracy is simple to calculate and easy to interpret.