Exercises #13 - Convolutional Neural Network

Thales Menezes de Oliveira

Brazilian Center for Physics Research (CBPF)

April 1, 2022

1. What are the main motivations for reducing a dataset's dimensionality? What are the main drawbacks?

A: The dimensionality reduction is extremely useful for data visualization, reducing the number of dimensions to two or three. This reduction allows the plot of a high-dimensional training set on a graph, which can provide important insights about the patterns presents in the data. With the reduction of the number of dimensions, the time needed to train an algorithm in this dataset will be smaller. As the drawbacks we can point the lost of information because of the dimensionality reduction process, and the computational effort that is can be necessary. The need of an additional algorithm to deal with the data adds more complexity to the ML model.

2. What is the curse of dimensionality?

A: The curse of dimensionality refers to the arise of many problems in the high-dimensional space that doesn't exist in a low-dimensional space. For the ML framework, the data points in a highly-dimensional space vectores are generally very sparse. This behaviour evolves in two different problems. The first one is related to the overfit, due to the difficulty of patterns identfication and the second is the need of plenty of training data, which can be inacessible, for some cases.

3. Once a dataset's dimensionality has been reduced, is it possible to reverse the operation? If so, how? If not, why?

A: It is possible for the most algorithm, except for the t-SNE, to provide the reverse transformation to have the data, after the compression, with the same dimensionality as the original. However, is almost impossible to perfectly reverse the operation, since osme of the information will get lost on the dimensionality reduction operation.

4. Can be PCA used ro reduce the dimensionality of a highly nonlinear dataset?

A: The PCA was first projected to deal with linear data, however, it can reduce significantly the number of dimensions of most dataset, getting rid of useless dimensions. Therefore, for a low number of dimensions with small variance, the PCA technique will provoke the lost of too much information. In this case, it will be better to find another dimensionality reduction techniques that can deal with nonlinear datasets.

5. Suppose you perform a PCA on a 1000-dimensional dataset, setting the explained variance ratio to 95%. How many dimensions will the resulting dataset have?

A: It will be strongly dependent of the dataset. For the perfectly randomized distribution, the answer it will be 950 dimensions, since each one of the dimensions will carry the same amount of variance.

6. In what cases would you use vanilla PCA, Incremental PCA, Randomized PCA, or Kernel PCA?

A: The orginal PCA works only if the dataset fits in memory. The Incremental PCA is useful for large training sets, since it only uses part of the data each iteration. It is quite interesting for online tasks, where it necessary to apply PCA whenever a new instance arrives. The Randomized PCA: Where the number of principal componentes is much smaller than the number of features in the dataset, reducing the computational complexity and training time. The Kernel PCA is highly recommended to deal with nonlinear projections of the data, in other words, nonlinear datasets.

7. How can you evaluate the performance of a dimensionality reduction algorithm on your dataset?

A: The reconstruction error measures the distance between the output of the dimensionality reduction algorithms, which is applied in the reverse transformation, to retain the original features and the original input vector. This metric can be used as a method to evaluate the performance of the algorithm. However, for the cases which the reverse transformation is not available, the DR can be used as preprocessing step for another ML algorithm. The measure of the performance of the second algorithm, for the case where you have the DR algorithm and the case it doesn't have it, can provide a evaluation metric, since if the DR doesn't lose too much information, the performance of the second algorithm, for both cases, will remain the same.

8. Does it make any sense to chain two different dimensionality reduction algorithms?

A: It will be dependent on the dataset, but it makes sense. It is possible to use a faster algorithm that can get rid of the least important dimensions of the dataset, and apply a more complex DR algorithm in the remaining features. This chain can get similar results if only the most complex algorithm was used, taking less time.