G4.PL-6

David Emanuel Craciunescu Laura Pérez Medeiro

December 15, 2019

Introduction

The objective of the sixth and last laboratory practice is to teach students how to perform and achieve high quality data visualization. The authors of this practice have decided to split it into 2 main parts that deal with different types of visualization: (1) static visualization, and (2) interactive data visualization.

In order to achieve this, the authors have used the *Python* programming language, the *TensorFlow* library, and a wide array of common methods and tools for data analysis and visualization.

The dataset used in this exercise is commonly referred to as *MNIST* or *MNIST Database*. This is a large dataset of handwritten digits that is commonly used for training various image processing systems. The database is also widely used for training and testing in the field of machine learning. It contains around 60,000 training images and 10,000 testing images, although this practice itself will only deal with maximum 10,000 images for the sake of simplicity.

In order to better visualize the elements in the *MNIST* Dataset, the *t-SNE* algorithm (T-distributed Stochastic Neighbor Embedding Algorithm) will be used. This specific algorithm is well suited for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions. As mentioned, *t-SNE* is a commonly used dataset, and it has been used in a wide range of applications, including computer security research, music analysis, cancer research, bioinformatics, etc.

Lastly, since the final result of the code itself is the visualization, the use of interactive and shareable code formats has been discarded for the actual visualization method. This means that this practice will deliver the source code, this memory, and the visualizations themselves.

Static two-dimensional visualization

For this specific part, the information will be shown as a two-dimensional plot of the different clusters of elements of *MNIST*.[1]

In this visualization, the different words are clustered thanks to the *t-SNE* algorithm and then plotted in two-dimensional space. These only have labels and do not have colors, for simplicity of processing and in order to improve the rendering time.

Attached to this submission, the image "bidimensional.png" is the result of this visualization method and algorithm.

[1] The code of this part can be found in "bidimensional.py".

Interactive three-dimensional visualization

Three-dimensional visualization is much more complex than a simple plot. In fact, for such an amount of data and the interaction this practice is looking for, the *TensorFlow* library was used.

This is a free and open-source software lirary for dataflow and differentiable programming across a range of tasks. It was initially designed by the *Google Brain* team for internal Google use but it has since been released under the Apache License 2.0 for public use. It is commonly used for tasks such as visualization and some of its modules, such as *Projector*, are extremely useful when manipulating data. In fact, the very *Projector* module will be used in this practice.

The code related to this part can be found in "tridimensional.py". This will be attached *as is* with the result and the visualization attached as "tridimensional.mkv". Since the creation of the project and the recording of the file "tridimensional.mkg", some minor and *untrackable* errors have been introduced into the code that make it non-executable. Therefore, the provided code is *not 100% functional*. The projector module used for the provided video comes from an official online Google implementation and not the code itself.