Homework #10 – Take-home Final
Due Friday, May 12, 2022, 11:59pm

Problem 10.1

Use the CovidNYT database on the SciServer to perform the following statistical analysis from a Jupyter notebook:

(1) Using the Census data in the database, write and execute SQL queries to select the 200 Counties in the US with the highest and lowest population densities. Extract the total population in both categories. You can use the CasJobs of the SciServer to explore the databases interactively, and to debug your queries.

(2) The StatsC table contains the cumulative counts of both the Covid19 infections and deaths for each county in the US, for each day. Using the queries from the previous step, extract the maximum of the cumulative infections and deaths for the top and bottom 200 counties (in population density). This should give a total of 4 numbers (corresponding to Apr 18, 2021).

(3) Using the population counts, calculate the values normalized to a population of 100,000 and determine if there is a statistically significant difference between the highest and lowest population densities. Try to estimate the difference between the values in terms of standard deviations, both for the infection counts and for the deaths.

Hint: a notebook called HW10-sql-helper.ipynb has been placed into the shared directory. Run this in the regular class container.

Problem 10.2

The task is to classify a set of traffic signs from Germany. There are two binary files with the images in the shared data folder. The file signs1.bin contains RGB images of 12630 traffic signs, while signs2.bin has the gray scale version. The color images are 32x32x3 uint8, while the gray scale images are 32x32 uint8. There are altogether 43 distinct classes among the signs. There is a notebook called trafficsigns.ipynb in the shared folder that shows how to access the images. Primarily use the gray scale images in signs2.bin for this project. Play with the RGB if you have extra time.

Follow Chapter 17 on the Geron book to build an autoencoder to reproduce the images. Once done, use the lower layers of the autoencoder as a classifier. First run the example on the Fashion MNIST data, and then modify the code to accommodate the traffic signs.

In order to do this, you will need to split the sample into a training set and a validation set. Vary the size of the latent layer of the autoencoder to see how the categories found will depend on this parameter. Explain the results of the experiments.

Hint: Use the Job submission in the V100 queue for maximum speed, as running the Deep Learning task on regular CPUs will be too slow. Ada has created a detailed document describing the shared use of the GPUs on the SciServer. Also, you may not be able to access the shared data path from the GPU container, but you will be able to see your own home directory. If this is the case, copy over the signs2.bin file and the class file to your home.

Since you will be working in batch mode, you may want to save your model at the end of the batch run, and then you can reload it in interactive (non-GPU) mode. See

https://www.tensorflow.org/guide/keras/save_and_serialize