Spatial Data Science & Engineering

Assignment 6

Maximum points Possible – 10

Environment Setup:

Install spatiotemporal deep learning library ***GeoTorch*** (https://github.com/DataSystemsLab/GeoTorch).

Installation instructions: https://kanchanchy.github.io/geotorch/installation.html. Make sure that dependencies are installed before you install GeoTorch.

GeoTorch Documentation: https://kanchanchy.github.io/geotorch/index.html

Tasks:

Your primary task is to create a spatiotemporal grid tensor from raw taxi trip dataset using GeoTorch preprocessing module and load the tensor as a trainable dataset in PyTorch using GeoTorch deep learning module.

Datasets: You are given two datasets – 1) Taxi zone shape file and taxi trips csv file. These two datasets will be used to create the spatiotemporal grid tensor.

Methods: You are given a Python file Assignment6.py containing 3 empty methods.

First method: *create_st_tensor* has the following parameters:

- path_to_shape_file: a String denoting the path to the taxi zone shape file
- path_to_csv_file: a String denoting the path to the csv file containing the taxi trips information
- output_path: a string path to write the final output.

The purpose of this method is to load the shape file and csv file using GeoTorch preprocessing module, create a spatiotemporal grid tensor, and write the tensor to the output path. Convert the space covered by the shape file into a 32x32 (height x width) grid. Trim the taxi trip records based on Trip_Pickup_DateTime attribute. Minimum possible value of Trip_Pickup_DateTime is '2009-01-02 00:00:00' and maximum possible value of Trip_Pickup_DateTime is '2009-01-30 23:59:59'. Divide the total temporal range into various timesteps. The length of each timestep will be 30 minutes. The feature that you need to calculate to form the tensor is the number of trip pickups happened at various timesteps in each cell of the 32x32 grid. The shape of the tensor will be TxHxWxC, where T is the number of timesteps, H is the grid height (32), W is the grid width (32), and C is the number of attributes (1). Write the tensor to the given output path. You will be using the methods available in GeoTorch preprocessing module to create the tensor.

An example:

1. https://kanchanchy.github.io/geotorch/coding_examples.html (Navigate to Preprocess Spatiotemporal Data in the left menu.)
2. https://github.com/DataSystemsLab/GeoTorch/blob/main/examples/st_preprocess.py
3. Documentation of required methods: https://kanchanchy.github.io/geotorch/api_preprocessing.html

Second Method: *load_tensor_periodical* has the following parameters: path_to_tensor, len_closeness, len_period, len_trend, T_closeness, T_period, T_trend, batch_size, batch_index.

First 7 parameters are the parameters of the Processed dataset available under GeoTorch deep learning module (geotorch.datasets.grid package). Don't change the default value of the parameters – is_training_data and test_ratio.

Processed dataset link: https://kanchanchy.github.io/geotorch/api_deep_learning.html (Navigate to Processed class under geotorch.datasets.grid package in the left menu).

Your task is to 1) load the given tensor into the Processed dataset, 2) initialize a PyTorch DataLoader with the Processed dataset and given batch size. 3) Finally return the sample batch from the DataLoader at the given batch_index. The returned sample should consist of the following: x_closeness, x_period, x_trend, y_data.

Third Method: *load_tensor_sequential* has the following parameters: path_to_tensor, len_history, len_predict, batch_size, batch_index.

This method is exactly similar to the second method *load_tensor_periodical*. Instead of len_closeness, len_period, len_trend, T_closeness, T_period, and T_trend parameters, you are given two different parameters: len_history and len_predict, which indicate the parameters of the *merge_closeness_period_trend* method under the same processed dataset class. Don't change the default value of the parameters – is_training_data and test_ratio.

Instead of returning x_closeness, x_period, x_trend, and y_data, you will be returning x_data and y_data.

Submission Instructions:

- Submit only Assignment6.py file.