COMP 6440 Topics in Data Mining — Project

Due date: 11:59pm Saturday May 1

In this project, we will implement two models: Graph Attention Networks (GAT) and Long Short-Term Memory (LSTM). For GAT, we will apply GAT to the CORA citation dataset (Node classification and Clustering). For LSTM, we will apply it to the NY Taxi dataset (Prediction).

Graph Attention Networks

The CORA citation dataset consists of 2708 publications classified into one of the seven classes. The citation network consists of 5429 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1433 unique words.

We will use StellarGraph (a Python library for machine learning on graphs and networks; *all the hyperlinks and datasets are provided in a section below*) to build the GAT model. We also need some other packages/libraries: NetworkX (a Python package for the creation, manipulation, and study of complex networks), pandas (a powerful tool for data analysis and manipulation), Keras (An Neural Network library), sklearn (we use it for splitting data) and matplotlib (for producing figures about training process and results). In the second part (LSTM), we will also need some of these tools.

Before we implement the GAT model, we need to do some preprocessing work on this dataset. This link (https://stellargraph.readthedocs.io/en/v1.2.1/demos/basics/loading-pandas.html) contains a step-by-step tutorial (The 'Real data: Homogeneous graph from CSV files' part) on it.

(Task 1) First, we will use the GAT model to classify a paper's subject. You can read this (https://stellargraph.readthedocs.io/en/v1.2.1/demos/node-classification/gat-node-classification.html) and follow the instruction to build your own GAT model (We will use 200 nodes for training, 700 for validation, and the rest for testing).

(Task 2) When we apply the GAT model to the dataset, we can get each node's (paper's) embedding. By using these embeddings, we can cluster the nodes. You need to write a program to exploit these nodes' embeddings for node clustering (by the K-means method) and present the result with figures (you can use the matplotlib library to do it). You need to implement K-means by yourself. "K" is a hyperparameter that you can explore.

Long Short-Term Memory

In the second part of this project, we will build a LSTM model and apply it to New York Taxi dataset (provided below). Our goal is a prediction task: For a particular taxi (with a unique id), when will it arrive at a specific target zone with **longitude ϵ [-73.986191, -73.980927] and latitude ϵ [40.745251, 40.753326]**?

For each record of this dataset, it has the following attributes:

Distance to target: indicating how long (in minutes) this taxi will arrive at our target zone above.

Id: the unique id of taxi

Time period of the day: indicating the time block (we partition a day into 7 time blocks)

Trip_time_in_secs: the total time of one trip

Trip_distance: the total distance of one trip

Longitude & Latitude: the corresponding location data

Is_pickup: 1 means the taxi picks up passengers at the location and 0 means drop-off.

Your LSTM model should use all the attributes above (except the first) to predict "Distance to target". The records are sorted first by taxi ID, and then by timestamp (which is removed from the dataset as it is not needed for the LSTM model). We will use the same packages/libraries except StellarGraph and NetworkX.

Before the implementation, we recommend that you read this (https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/). This article gives a step-by-step tutorial for building LSTM and predicting the pollution value of the Air Pollution dataset.

(Task 3) Once you are familiar with LSTM, you need to implement your own LSTM model for our goal: to predict how soon a taxi will arrive at the target zone, and to present the training process and result with figures (Use 50 taxis as training data and the rest for testing).

Links to Datasets and Packages/libraries:

You can download the dataset via this link (the NY Taxi dataset is named ny_taxi_proj.csv, and the CORA and Air pollution datasets are from the tutorials):
https://drive.google.com/drive/folders/1PIh2CzhSgi_eW3JfLzWpkmrWeXRu7ykU?usp=sharing

Here are the links to the aforementioned tutorials or packages/libraries:

StellarGraph: https://stellargraph.readthedocs.io/en/v1.2.1/README.html

NetworkX: https://networkx.org/

Pandas: https://pandas.pydata.org/

Keras: https://keras.io/getting_started/

Sklearn: https://scikit-learn.org/stable/

Matplotlib: https://matplotlib.org/

How to load CORA dataset and preprocessing:
https://stellargraph.readthedocs.io/en/v1.2.1/demos/basics/loading-pandas.html

Tutorial of LSTM:

https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/

Environment Requirement

You need a Linux-base Operating System (Ubuntu, etc.) with packages (mentioned above) and their pre-required packages, as well as Python 3 (https://www.python.org/).

Or you can try the CS virtual system. Using Bitvise (https://www.bitvise.com/ssh-client-download) or other tools to login. The host is mercury.cs.uml.edu (port 22) and the authentication needs your CS account (username) and password. If you don't have a CS account, you may send an email with the request form (http://www.cs.uml.edu/account.html) to help@cs.uml.edu.

What to submit

Code: The files for the three tasks with the GAT model and the LSTM model (with a readme file).

Report: A summary report about your work with the output of each task. A discussion about your models (hyperparameter tuning, etc.).

Your grade will be based on these 3 parts: 40% for GAT (20% for GAT model implementation and 20% for clustering), 40% for LSTM, and 20% for the report.