1 Capstone Project: Deep Learning for Vehicle Routing Problem

1.1 Description

In this project, you will learn to solve a classic combinatorial optimization problem (VRP or TSP) using deep learning algorithms. In general, you are expected to propose different architectures to automatically learn heuristics over different sizes of graphs in order to produce a solution. Provide the rationale for your choices. Example approaches include but not limited to:

- **Graph Embedding with Reinforcement Learning: (structure2vec)**: Use a graph neural network embedding model to output which of the nodes would be inserted into a partial solution using reinforcement learning.
- **Graph Convolutional Network Technique:** Train a graph neural network on a 2D Euclidean VRP in a supervised manner to output an adjacency matrix. Then you can perform a (tree) search algorithm of your choice to convert on this adjacency matrix it into a feasible final solution.
- **Pointer Network(sequence-to-sequence):** Train an encoder and a decoder to autoregressively output a permutation of the given input sequence. You can also implement an attention-based decoder trained with reinforcement learning.

1.2 Dataset

You should consider synthetic graph data of size 20, 50, and 100. If you wish to train your network in a supervised fashion, you can refer to Concorde for an optimal tour solution to a given graph. (*http://www.math.uwaterloo.ca/tsp/concorde.html*)

1.3 Submission Requirements

- Jupyter notebooks (ipynb) or python source code for training, inference, and evaluation of your model.
  - You may choose Tensorflow or PyTorch for implementation
  - implementation must be Python-based
- Instructions to set up and run the source code (Docker file, shell script, pip commands, etc.)
- PowerPoint slides and the corresponding PDF files illustrating the graph embedding method
- Any datasets you used or links to download the datasets
- Report with the results supporting your comparative analysis: include the following information and metrics at a minimum:

- Description of the datasets
- Description of the specific problem of your choice. You can choose from Travel Salesman Problem and all its variants. e.g. VRP, capacitated VRP and multi-capacitated VRP.
- Description of the specific algorithm
- Architectures: visual graphs
- Architectures Hyper-parameters: table
- Time/Epoch: bar graphs
- Hyper-parameter Tuning: Choices, rationale, observed impact on the model
- Comparison to other heuristics and algorithms such as:
  * Nearest Insertion
  * Random Insertion
  * Farthest Insertion
  * Google OR-tools
  * Clarke-Wright Savings Heuristic
  * Sweep Heuristic

1.4 Useful Resources

- Reinforcement learning for solving the vehicle routing problem *(http://papers.nips.cc/paper/8190-reinforcement-learning-for-solving-the-vehicle-routing-problem.pdf)*
- Learning the Multiple TSP with Permutation Invariant Pooling Networks *(https://arxiv.org/pdf/1803.09621.pdf)*
- Attention Solves Your TSP, Approximately *(https://pdfs.semanticscholar.org/246e/c92581f9756b505bbc5499ca91e716d1ef3d.pdf)*
- An Efficient Graph Convolutional Network Technique for the traveling salesman problem (https://arxiv.org/pdf/1906.01227.pdf)
- Applying deep learning and reinforcement learning to traveling salesman problem (https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=8659266)