**Anasakta Labs ML Internship**

**27/04/2020**:

* Understood basics of graph data structure – core concept and various ways of representation.
* Studied basic concepts of working of graph neural networks.
* Installed DGL and PyTorch through Jupyter notebook.
* Python programming of how to create a graph (directed and undirected) using the DGL.

**28/04/2020**:

* PyTorch Fundamentals
* Python programming of training graph neural network on Karate Club.
* Python programming of how to create empty DGL Graph and add nodes and edges subsequently.

**29/04/2020**:

* Deep dive into Training of GNN and observing effect of change of hyperparameters on clustering of Karate Club graph.

**30/04/2020**:

* Understood basics of Clustering in Machine Learning and explored concept and implementation of various popular clustering techniques such as:
  + K-Means Clustering
  + DBSCAN
  + Affinity Propagation
  + Agglomerative Hierarchical Clustering

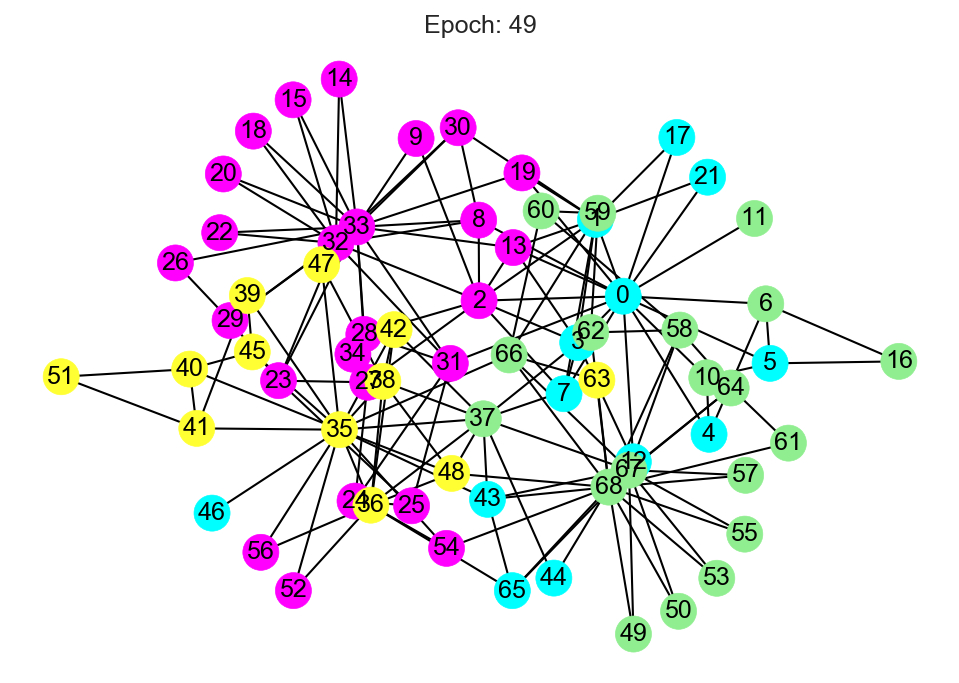
**04/05/2020**:

* Observed effect of change of hyperparameters such as number of training epochs on clustering output of 3-layer GNN.
* Comparing clustering of 2-layer GNN and 3-layer GNN.

|  |  |  |  |
| --- | --- | --- | --- |
| Image | Conv Layers | Inference | No. of epochs |
| karate\_club\_3.png | 2 | Optimum clustering; densely connected graph. | 150 |
| karate\_club\_4.png | 2 | Node 27 shifted from blue to pink cluster.  Node 25 shifted from pink to blue cluster.  Graph starts getting distributed.  Degradation of clustering result. | 200 |
| 3\_layers.png | 3 | Distributed graph.  Bad clustering. | 50 |
| 3\_layers\_2.png | 3 | Densely connected graph.  Node 31 in blue cluster.  Improvement in clustering result. | 100 |
| 3\_layers\_3.png | 3 | Node 31 in pink cluster.  Degradation of clustering result. | 150 |

**05/05/2020**:

* Generalisation of DGL graph clustering to function for required number of sets (taken as input from user).
* Run for 2 sets i.e. 4 clusters (100 epochs) and observed output.
* Downloaded files from download\_designs.py.



**06/05/2020**:

* Worked on improving clustering output with 69 nodes via more epochs and trying to use 3 layers.

|  |  |  |  |
| --- | --- | --- | --- |
| Image | Conv Layers | Inference | No. of epochs |
| 69\_nodes\_cluster.png | 2 | Bad clustering; scope of improvement. | 50 |
| 69\_nodes\_2\_layers\_1.png | 2 | Improvement in clustering.  46,56,52,54 -> shift to yellow  65,44,34,43 -> shift to green  11,6,16,10,13,19 ->shift to blue  Lot of misclassified pink nodes shifted to correct cluster. | 100 |
| 69\_nodes\_2\_layers\_2.png | 2 | Same as above.  Convergence of clustering algorithm. | 150 |
| 69\_nodes\_3\_layers.png | 3 | Little improvement as compared to 69\_nodes\_cluster.png. | 100 |

* Studied basic concept of Application Specific Integrated Circuits (ASIC) Design EDA Flow.

**07/05/2020**:

* Studied VLSI Partitioning from SlideShare link [<https://www.slideshare.net/subashjohn1/system-partitioning-and-its-considerations>].
* Understood the meaning of various notations in the VLSI netlist files and its relation to representation as DGL Graph.

**08/05/2020**:

* Python programming of parsing netlist file to convert it into Deep Graph Library (DGL) graph.
* Debugging procedure.

**11/05/2020**:

* Worked on training GCN based on DGL graph created from netlist files.
* Training for 20 clusters in the graph and 100 epochs.
* Created dictionary which maps node label (0-19) to a list which contains the nodes for each respective cluster.

**12/05/2020**:

* Training GCN for 150 epochs to reduce the loss and noting training time.
* Python programming to create dictionary which maps node label to number of nodes in each cluster as part of analysing quality of clustering.
* Tool used: Google Colab.

**13/05/2020**:

* Reporting function programming for Cluster Quality Metric #2: Number of Nodes per Cluster Distribution.
* Using DGL APIs (dgl. DGLGraph.in\_edges(),dgl. DGLGraph.out\_edges) to extract incoming and outgoing edges’ source and destination nodes tensors.
* Training GCN for 200 epochs on TPU and noting runtime.
* Creating dictionary standard\_nodes\_dict which maps the node number in a particular cluster (taken as input from user) to a standard count procedure for easy visualisation of networkx graph.

**14/05/2020**:

* Cluster Visualisation for any generic cluster label taken as input from user using Google Colab (TPU Runtime).
* Cluster Quality Metric #3: Visualisation not very good with networkx graph.
* Shifted to GEPHI for interactive visualisation.
* Download and install GEPHI v0.9.2 and Java as dependency for GEPHI. Subsequent troubleshooting.

**15/05/2020**:

* Some more visualisation as part of Cluster Quality Metric #2 for more clusters using GEPHI.
* Python programming of Cluster Quality Metric #1: Total number of edges between nodes which belong to different clusters divided by total edges.
* Created dictionary mapping pair of cluster labels to the number of edges between their nodes as part of Metric #1.

**18/05/2020**:

* Modified Cluster Quality Metric #1: separation into 2 dictionaries, one containing the absolute number of edges between all combinations of 2 clusters and the other representing the edges as a percentage of the total number of edges of graph G.
* Converted graph\_from\_netlist\_3.ipynb into a Python Module file to be imported into another .ipynb file to run directly and get metrics.

**19/05/2020**:

* Observed effect of change of hyperparameters on adaptec1.nets file: for 3 convolutional layers, 250 epochs, learning rate of 0.01.
* Taking input hyperparameters (epochs, learning rate) from user and then produced metrics and noting training time for GNN.
* Running entire program for bigblue1.nets and produced the metrics for the same.
* Results compilation can be viewed at this link: <https://docs.google.com/document/d/1JXmPivCh0flitcOSXL0E3ck1B5f0JSX8k_s0TQOJB0I/edit?usp=sharing>

**20/05/2020**:

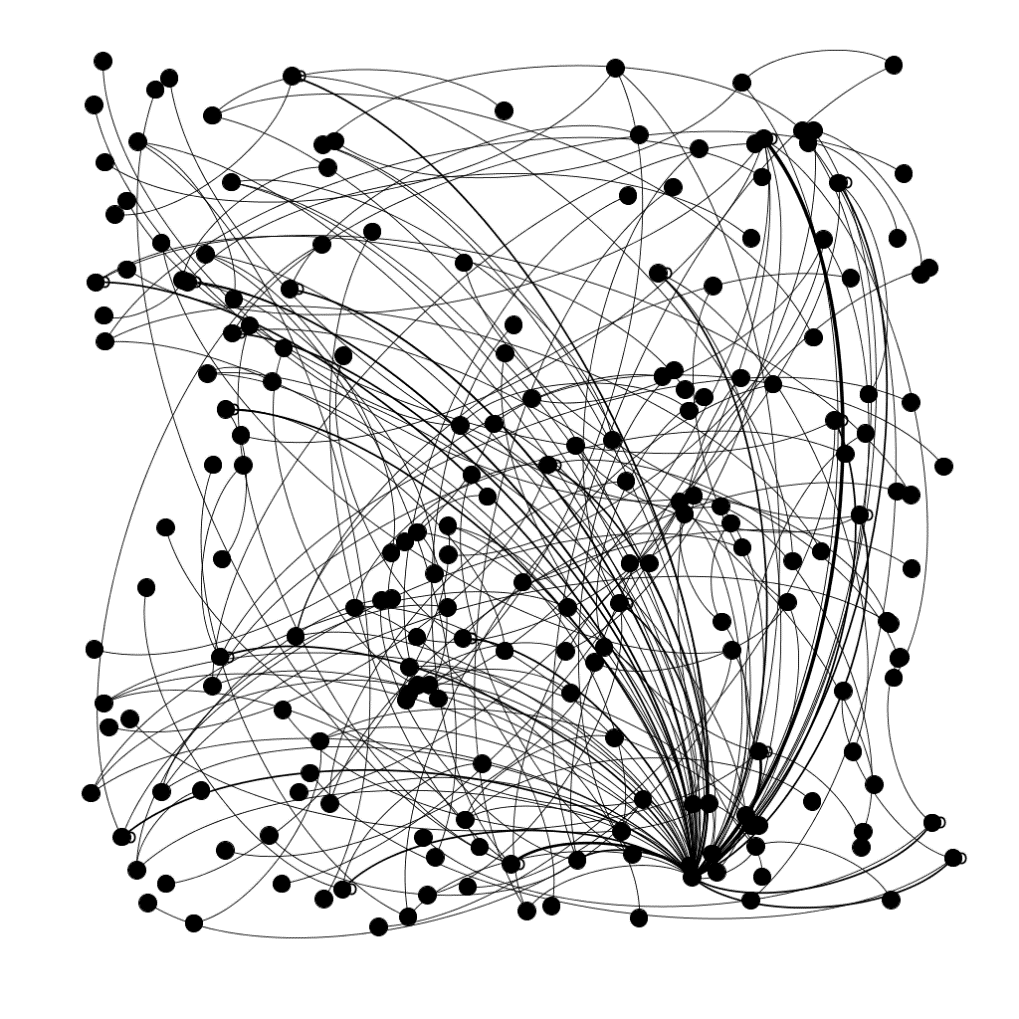
* Running clustering script on bigblue1.nets file for 3 convolutional layers and 260 epochs and produced required metrics.
* Python programming of Metric #1 by comparing the clusters of the source and destination nodes at each edge in the graph and storing in a dictionary.

**21/05/2020**:

* Fixing the new code for the Cluster Quality Metric #1; previously the edges\_dict created was mapping a pair of 2 nodes as key. Updated edges\_dict maps 2 cluster labels as key.
* Ran for 200 nodes, 170 entries in dictionary (edges\_dict); run time: 29 mins.
* Sorted edges\_dict in descending order of number of interconnected edges between 2 cluster and plotted bar graph for the following:
  + Adaptec1.nets
  + Bigblue1.nets

**22/05/2020**:

* Defined a new metric in combination with Cluster Quality Metric #1 called **Inter-Cluster Score** calculated as follows:
  + IC Score = No. of edges between 2 clusters (say c1 and c2) / (Number of nodes in c1 + Number of nodes in c2).
* Bar graph x-labels changed to represent the respective top 10 cluster pairs.
* Running above 2 metrics for the following files:
  + Adaptec1.nets
  + Bigblue1.nets
* Visualization of graphs using GEPHI for the cluster pair with highest number of edges in adaptec1.nets.



**25/05/2020**:

* Running entire script for the following files:
  + Adaptec2.nets
  + Adaptec3.nets
  + Adaptec4.nets
* Data Visualisation: 2 more bar charts
  + One for the number of nodes per cluster distribution
  + Another for Inter-Cluster Score
* Converting x-labels of graph which were previously in string representation of float to int for elegant visualisation.

**26/05/2020**:

* Converting entire script into an object-oriented (class-object) model in DigitalCluster.py and debugging procedure.
* Creation of master driver script **run.py** which imports DigitalCluster.py as module, creates an object of class DigitalClustering and runs the member functions as follows:
  + Run (): Training neural network and calculating the metrics.
  + Visualise (): Plotting the 3 bar graphs.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**END**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**VARUN SIRPAL**

**DELHI TECHNOLOGICAL UNIVERSITY**

**B.TECH. (ELECTRICAL ENGINEERING)**

**April 2020 – May 2020**