Name: Thilakasiri P.K.V.C  
IT No: IT21244698

Batch: WE.SE.01.01

GitHub Link - https://github.com/vidura-chathuranga/DL\_LAB\_06.git

**DL – Lab 06 Answers**

increase the N from 20 to 200 and observe graph density and degree distribution and explain what you observe

when node = 20

    Graph density: 0.2105

when node = 200

    Graph density: 0.0201

When the number of nodes (N) increases without changing the number of edges (E), the density will decrease. The reason for that is when edges increase linearly but possible node pairs increase quadratically.

|  |  |  |
| --- | --- | --- |
| supervised learning | self-supervised learning | semi-supervised learning |
| Need to label all the data. Data should be human labeled. | Data does no need to be labeled.  Data will be labeled itself. | Some data are labeled (partially labeled data) but some are not.  Partially labeled. (Human labeled + Labeled data itself) |
| Require a training set of example inputs and their corresponding desired outputs when learning. | Require only a training set of input data, the desired outputs are not provided when learning. | Require a training set of input data (not all) and a set of desired outputs. |

Transductive learning – This focuses on using all available data (including test instances) to optimize for specific predictions. It works well when all instances are known during training.

Inductive learning – This aims to build a generalized model capable of predicting outcomes for new, unseen instances, making it more flexible in real world applications where new data continuously arrives.

* When num of epochs set to 50,

|  |  |  |  |
| --- | --- | --- | --- |
| Epoch | Loss | Training Accuracy | Validation Accuracy |
| 0 | 1.4324 | 25% | 35.29% |
| 10 | 1.3547 | 25% | 35.29% |
| 20 | 1.2063 | 50% | 67.65% |
| 30 | 0.9844 | 100% | 73.53% |
| 40 | 0.7593 | 100% | 64.71% |

During the first 50 epochs, training accuracy increases rapidly, reaching 100% by epoch 30. The validation accuracy also shows improvement, but not as quickly as the training accuracy. It rises from 35.29% at epoch 0 to 73.53% at epoch 30 before dropping slightly to 64.71% at epoch 40.

We can see the model learns quickly to fit the training data but may be starting to overfit the training set, as indicated by the fluctuation in validation accuracy after epoch 30.

* When num of epochs set to 500,

|  |  |  |  |
| --- | --- | --- | --- |
| Epoch | Loss | Training Accuracy | Validation Accuracy |
| 0 | 1.4324 | 25% | 35.29% |
| 10 | 1.3547 | 25% | 35.29% |
| 20 | 1.2063 | 50% | 67.65% |
| 30 | 0.9844 | 100% | 73.53% |
| 40 | 0.7593 | 100% | 64.71% |
| 50 | 0.5632 | 100% | 64.71% |
| 60 | 0.4161 | 100% | 73.53% |
| 70 | 0.3188 | 100% | 73.53% |
| 80 | 0.2548 | 100% | 76.47% |
| 90 | 0.2110 | 100% | 79.41% |
| 100 | 0.1795 | 100% | 79.41% |
| 150 | 0.0999 | 100% | 79.41% |
| 200 | 0.0667 | 100% | 79.41% |
| 250 | 0.0486 | 100% | 79.41% |
| 300 | 0.0374 | 100% | 82.35% |
| 350 | 0.0299 | 100% | 82.35% |
| 400 | 0.0246 | 100% | 82.35% |
| 450 | 0.0206 | 100% | 82.35% |
| 490 | 0.0181 | 100% | 82.35% |

Between epochs 50 and 500, the training accuracy remains at 100%, the model perfectly fits the training data. The validation accuracy gradually increases over time, reaching a peak of 82.35% by epoch 300 and stabilizing at that level through to epoch 500.

The loss continues to decrease over the epochs, indicating that the model is converging and refining its ability to predict with high accuracy.

* Without self-loops added to GCNConv() layers

|  |  |  |  |
| --- | --- | --- | --- |
| Epoch | Loss | Training Accuracy | Validation Accuracy |
| 0 | 1.4311 | 25% | 35.29% |
| 10 | 1.3475 | 25% | 35.29% |
| 20 | 1.1987 | 50% | 64.71% |
| 30 | 1.0007 | 50% | 47.06% |
| 40 | 0.8575 | 50% | 47.06% |
| 50 | 0.7766 | 75% | 50.00% |
| 60 | 0.7277 | 75% | 50.00% |
| 70 | 0.6862 | 75% | 50.00% |
| 80 | 0.6395 | 50% | 47.06% |
| 90 | 0.5895 | 50% | 50.00% |
| 100 | 0.5449 | 75% | 55.88% |
| 150 | 0.4313 | 75% | 55.88% |
| 200 | 0.3964 | 75% | 55.88% |
| 250 | 0.3806 | 75% | 55.88% |
| 300 | 0.3678 | 100% | 58.82% |
| 350 | 0.2549 | 100% | 61.76% |
| 400 | 0.1688 | 100% | 61.76% |
| 450 | 0.1335 | 100% | 61.76% |
| 490 | 0.1147 | 100% | 61.76% |

**With self-loops,**

The training accuracy reaches 100% by epoch 30 and remains there throughout the training process. The validation accuracy gradually increase, reaching a peak of 82.35% by epoch 300 and stays stable at this level until epoch 500.

The loss decreases consistently from 1.4324 at epoch 0 to 0.0181 at epoch 490, indicating a good level of convergence.

**Without Self-Loops,**

The training accuracy increases more slowly, reaching 100% only by epoch 300. The validation accuracy is significantly lower compared to the model with self-loops, peaking at 61.76% by epoch 300 and staying at that level until epoch 490.

The loss decreases less consistently, from 1.4311 at epoch 0 to 0.1147 at epoch 490. The reduction in loss is slower compared to the model with self-loops.

* Adding self-loops in the GCNConv layers improves both training speed and generalization ability. Without self-loops, the model takes longer to learn (reaching 100% training accuracy only after epoch 300) and fails to generalize as effectively, as reflected in the lower validation accuracy (peaking at 61.76% without self-loops compared to 82.35% with self-loops).
* Therefore, self-loops will enhance both the training efficiency and the overall performance of the GCN model.
* After adding 8 layers and skipping connections will increase the model performance.

|  |  |  |  |
| --- | --- | --- | --- |
| Epoch | Loss | Training Accuracy | Validation Accuracy |
| 0 | 1.4800 | 25% | 35.29% |
| 50 | 0.7063 | 75% | 52.94% |
| 100 | 0.4848 | 75% | 67.65% |
| 200 | 0.3982 | 75% | 67.65% |
| 300 | 0.3731 | 100% | 70.59% |
| 370 | 0.1666 | 100% | 85.29% |
| 400 | 0.0704 | 100% | 73.53% |
| 490 | 0.0342 | 100% | 67.65% |

Adding 8 layers with skip connections slightly improved validation accuracy (85.29%) but led to some overfitting or instability as validation accuracy decreased toward the end.

Without the additional layers and skip connections, the model achieves steady validation accuracy (82.35%) and faster convergence in terms of both loss and accuracy.

The addition of deeper layers seems to improve performance temporarily, but it may introduce some overfitting or difficulty in training the deeper model as indicated by the validation accuracy fluctuation.

4.

|  |  |
| --- | --- |
| Model Name | Explanation |
| Message Passing GNN | A general framework for GNNs where nodes aggregate information from their neighbors in a multi-step process (message passing). Each node receives 'messages' from its neighbors and updates its own embedding through aggregation. Used as the foundation for many other GNN models. |
| Graph Convolution Network | Applies convolution operation over graph data, extending classical convolution to graphs. It aggregates the features of a node and its neighbors using a fixed weight matrix, assuming the graph is undirected. Relies on spectral methods. |
| Graph Attention Network | Introduces attention mechanism to graph convolution. Nodes compute attention scores for their neighbors and weigh their neighbors' features, accordingly, allowing the network to focus on the most relevant neighbors for each node. |
| GraphSAGE | A variant of GNN that samples a fixed-size set of neighbors for each node to aggregate information. Uses different aggregation functions like mean, LSTM, or pooling, and it can generalize to unseen nodes by learning inductive representations. |