Deep Learning in Practice - Practical Session 3
Graph-NN

Siwar Mhadhbi - Eya Ghamgui - Saifeddine Barkia

February 24, 2022

1. Architecture of the model

The architecture we have chosen for the classification task is based on the attention mechanism. It is composed of 3 GAT layers each one takes as input a set of node features, $h = \{h_1, h_2, ...h_N\}$, and outputs a new set of node features, $h' = \{h'_1, h'_2, ..., h'_N\}$. Each GAT layer applies a shared linear transformation with the fully connected layer, then performs a shared multi-head attentional mechanism with 4 heads of size 350 each, afterwards concatenates the results. The last layer is used for (multi-label) classification and consists of a fully connected layer followed by a Softmax layer.

2. Hyper-parameters tuning

For this task, we performed the selection of the hyperparameters sequentially. We decided to use as default values for the hyperparameters the ones proposed in the paper [1]. The first step is to find the best architecture that gives the highest performance. We first searched for the optimal number of GAT layers to use in the model. Thus, we trained and evaluated architectures with 1, 2, 3, 4, 5 and 6 GAT layers. We found that 3 is the best value. Next, we changed the hidden size of the GAT layer. Trying 128, 256, 350 and 400 as hidden sizes, we noticed that 350 gives the best training and test scores. We also changed the number of heads by trying values from 1 to 5. The best F1 score was reached with 4 heads. Finally, we evaluated the activation functions, we found that ReLU and Leaky ReLU gave better results than the others (elu, sigmoid and tanh), with a slight advantage for Leaky ReLU.

In order to train the model on the given data set, we chose the Adam optimizer and tuned its learning rate parameter. The value 0.01 gave an unstable accuracy. In fact, high learning rates cause drastic updates that lead to divergent behavior. The value 0.001 resulted in a slow convergence of the model. Indeed, a small learning rate requires many updates before reaching the optimal solution. Thus, the best value of the learning rate and which gives higher training and testing scores is 0.005. In addition, we chose the best batch size 2 among the tested values 2, 4, 8, 16, 32 and 64 and noticed that 300 epochs are better than 100, 200, 400 and 500 epochs.

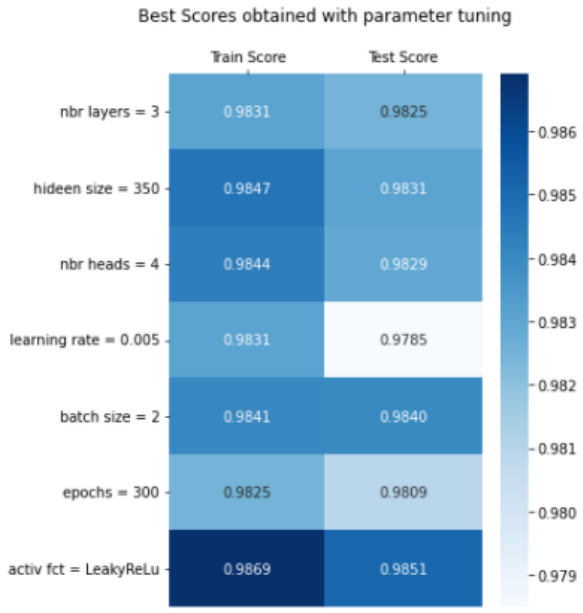


Figure 1: Best scores obtained with parameter tuning

3. Results

After training the best model chosen with hyper-parameters tuning, we obtained an F1 score of 98.69% in the training phase and 98.51% in the testing phase. Moreover, the following graph clearly shows that our model is outperforming the baseline model.

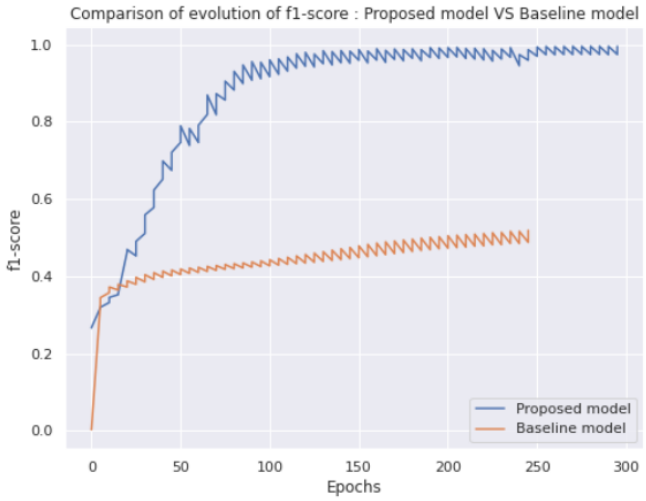


Figure 2: Comparison of evolution of F1-score between proposed model and baseline model

4. Why would Attention Network perform better than GraphConv?

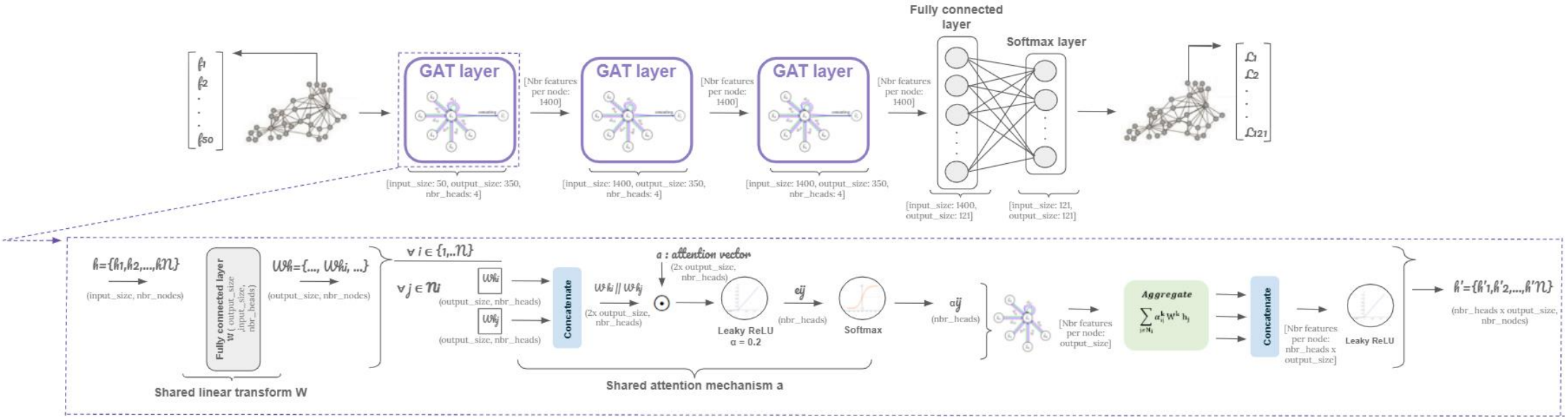
The graph attention network outperforms the convolutional graphs for various reasons.

- The first reason is related to the contribution of neighboring nodes to the central node. As opposed to convolutional graphs (both spectral and spatial method) where the weights are pre-determined, GAT calculates the weights based on the features that are generated within an end-to-end architecture so that more important nodes receive larger weights. Thus, GAT allows for assigning different importance to nodes having the same neighborhood.
- Second, the attention mechanism is applied in a shared way. Therefore, the order of access of the graph is no longer important (the graph can be also directed).
- Finally, the use of the multi-head attention in GAT increases the model's expressive capability compared to GraphConv. We also note that the operation of self-attention can be parallelized so that the output of all features is indeed parallelized across all edges. As a matter of fact, we observed that the training of the "complicated" GAT model took less time than training the simple baseline GraphConv.

References

[1] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *stat*, 1050:20, 2017.

5. Diagram of the architecture



https://drive.google.com/file/d/1E1_EotXlBi93DrL9iuJftE7liOvMXqJM/view?usp=sharing