ADL TD3 Report

Zhenning Li, Qiyun WU, Brieg L'Hostis

1 Diagram of the best architecture we used

The following is final the Graph Attention Network (GAT) model we used:

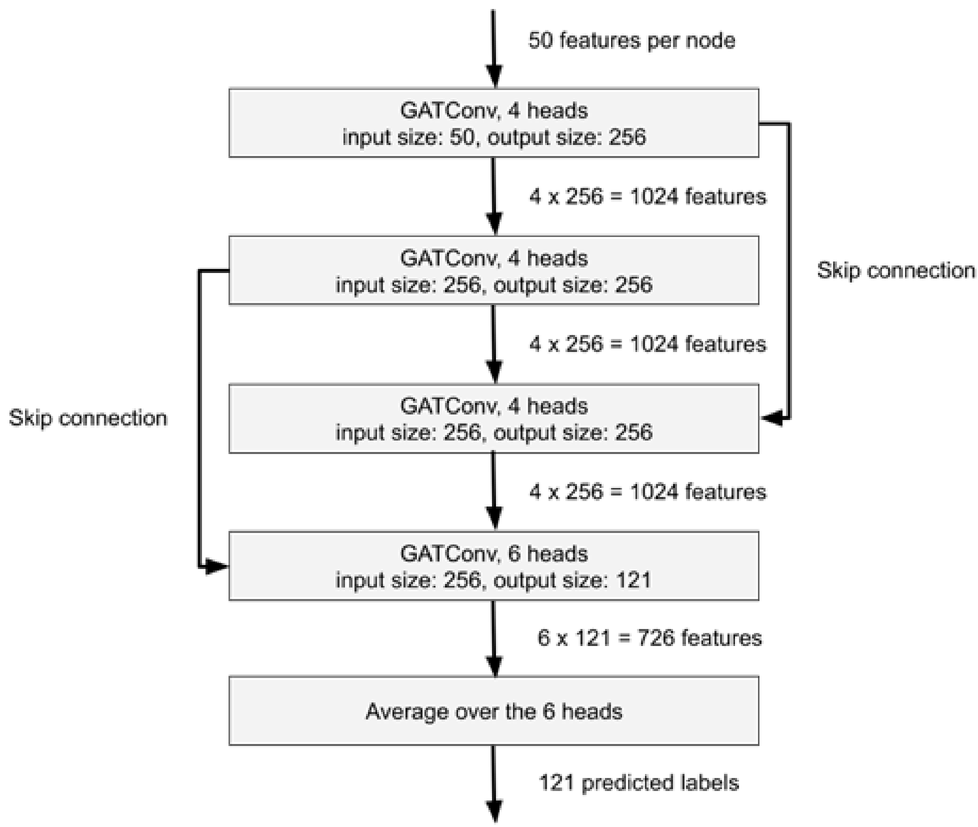

Figure 1: Architecture of the best model (GAN with multi-head output and skip connections, 3 layers)

As input of the model is a vector on 50 features per node and as output is a vector of 121 labels. We used a GAT model to predict the labels for each node. This model is composed of four GATConv layers including the input layer and three hidden layers. All of them have a size of 256 and the first three one have four heads while the last one has six heads. The output layer of the model is an average over the six heads of the last input layer. Finally, we added skip connections around the second and third layers.

2 The architectures we tried

In this project, we tried the following architectures:

- First the baseline model with two GraphConv layers of size 256:

  BasicGraphModel: n_layers=2, input_size=n_features, hidden_size=256, output_size=n_classes, nonlinearity=F.elu

- Then we replaced the GraphConv layers by GATConv layers with 4 heads:

  AttentionGraphModel: layer_sizes=[256, 256], input_size=n_features, output_size=n_classes, nonlinearity=F.elu, num_heads=4

- Then we changed the output layer to have 6 heads and computed the average over the heads as described previously:

  GAT Model: input_size=n_features, output_size=n_classes, nonlinearity=F.elu, n_layers=2, num_heads=4, skip_connections=**False**

- Then we added the skip connections around the middle layer:

  GAT Model: input_size=n_features, output_size=n_classes, nonlinearity=F.elu, n_layers=2, num_heads=4, skip_connections=**True**

- And finally we added one more hidden layer:

  GAT Model: input_size=n_features, output_size=n_classes, nonlinearity=F.elu, n_layers=3, num_heads=4, skip_connections=**True**

3 Results and interpretation

The following table is our result using different models.

	Final train loss	Final train F1-Score	Test F1-Score
Original model	0.4497	0.5105	0.5081
GAN, 2 layers	0.2847	0.7432	0.7367
GAN with multi-head output, 2 layers	0.1634	0.8633	0.8450
GAN with multi-head output and skip connections, 2 layers	0.1407	0.8595	0.8559
GAN with multi-head output and skip connections, 3 layers	0.0597	0.9375	0.9298

Figure 2: F1-Scores for different models trained on 250 epochs

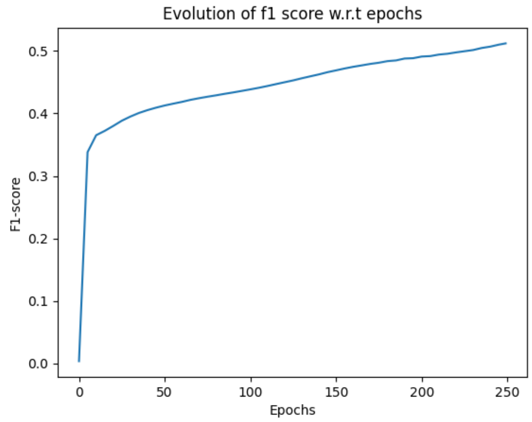

Figure 3: F1-Scores evolution using BasicGraphModel

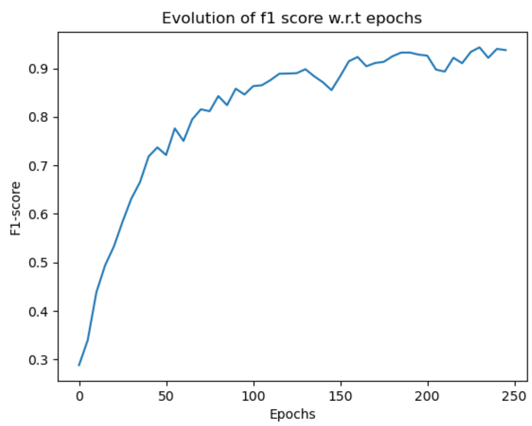

Figure 4: F1-Scores evolution using the best GAT model

Original model result: n_layers=2, hidden_size=256 -> train loss: 0.4497; train f1 score: 0.5105; test f1 score: 0.5081

Best model result: with skip connection, n_layers=3, hidden_size=256, 250 epochs -> train loss: 0.0597; train f1 score: 0.9375; test f1 score: 0.9298

Interpretation: From the table and graph above we can see, the original model performs the worst and the GAN with multi-head output and skip connections with 3 layers performs the best, the best model has a f1 score almost to 1 while the original model only has 0.5.

4 Reason that Attention Network performs better than GraphConv

Attention Network, for example the GAT, its layer expands the basic aggregation function of the GCN layer, assigning different importance to each edge through the attention coefficients. It defines an anisotropy operation in the recursive neighborhood diffusion, and it exploits the anisotropy paradigm, and the learning capacity is improved by the attention mechanism.

Compared with GraphConv, the GAT's attention layer carries more useful information than a regular convolutional layer thanks to the multiple heads, thus it utilized more information then GraphConv and performs better.