Attention Explainer

1. Introduction

An explanation tool that uses attention coefficients generated by an attention-based Graph Neural Network (GNN), such as GATConv, GATv2Conv, or TransformerConv, for edges explanation. The attention scores across layers and heads are aggregated according to the **reduce** argument.

2. Simple explanation

We perform self-attention on the nodes in a neighborhood using a shared attention mechanism. This mechanism computes attention coefficients that indicatie the importance of node j's features to node i.

3. Formulation

We'll begin by discussing a single graph attentional layer, which is consistently used in all GAT architectures in the paper's experiments. Attentional approach of the paper is inspired by *Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. International Conference on Learning Representations (ICLR), 2015.*, but the framework in the paper is flexible and not tied to any specific attention mechanism.

In this layer, we take a set of node features represented as $h = \{h_1, h_2, ..., h_N\}$, where each h_i belongs to a feature space $\mathbb{R}^F$. Here, N is the number of nodes, and F is the number of features per node. The layer then generates a new set of node features, $h' = \{h'_1, h'_2, ..., h'_N\}$, where each h_i is now in a different feature space $\mathbb{R}^{F'}$.

To enhance the expressive power and transform input features into higher-level ones, we need at least one learnable linear transformation. Initially, a shared linear transformation is applied to each node using a weight matrix $W \in \mathbb{R}^{F' x F}$. Subsequently, self-attention is performed among the nodes using a shared attention mechanism. This mechanism computes attention coefficients, $e_{ij} = a(Wh_i, Wh_j)$, indicating the importance of node j's features to node i.

The model allows every node to attend to every other node, but we introduce the graph structure by using masked attention. This means we only compute attention coefficients for nodes $j \in N_i$, the neighborhood of node i in the graph (typically the first-order neighbors).

To compare coefficients across different nodes, we normalize them using the softmax function. The attention mechanism in our experiments is a single-layer feedforward neural network parametrized by a weight vector $a \in \mathbb{R}^{2F'}$ and applying the LeakyReLU nonlinearity (with a negative input slope of $\alpha = 0.2$).

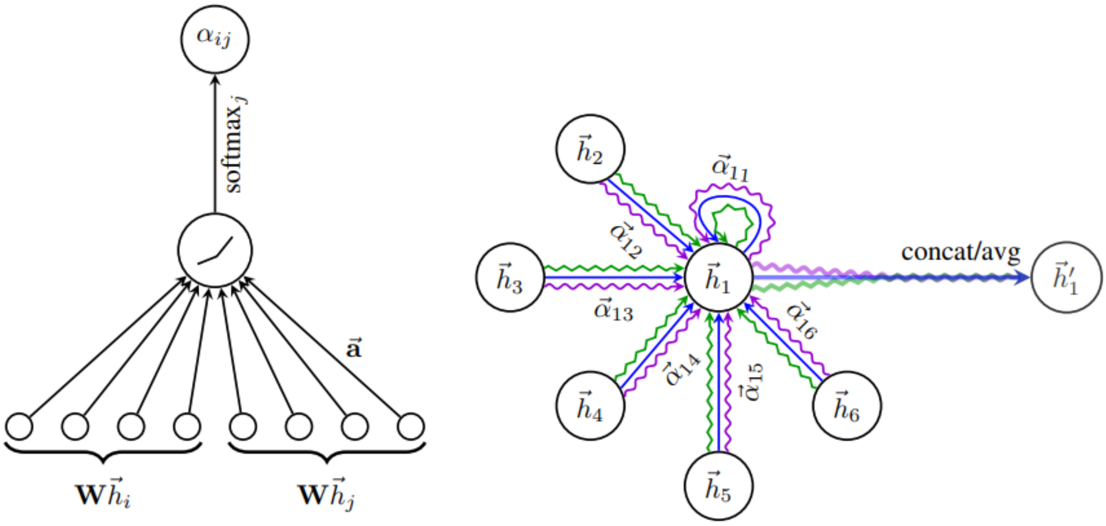

Fig 1. Multi-head attention in GAT network - image from paper [1710.10903] Graph Attention Networks

4. Example

- **CORA Dataset (Multi-Class Classification):**

The model is a 2-layer Graph Convolutional Network (GCN) trained on the CORA dataset for node classification.

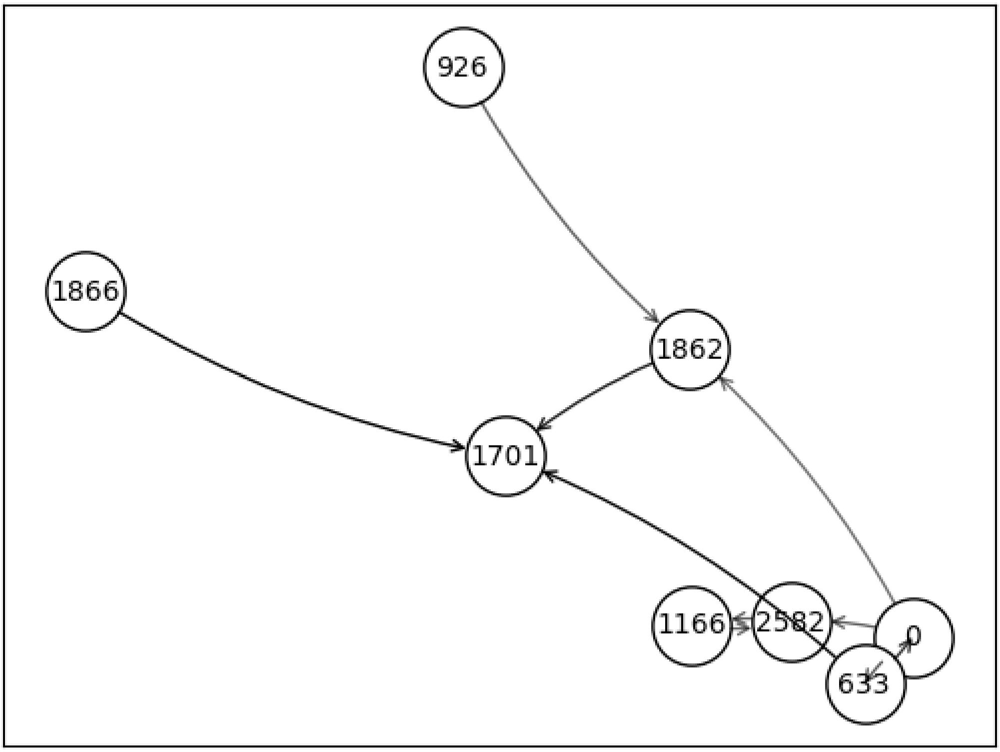

Fig 2. GAT model - The output of the explainer in case of the prediction for node 0.

- ## Minesweeper Dataset (Binary Classification):

The model is a GCN designed for binary classification on the Minesweeper dataset, where nodes represent cells in a Minesweeper grid.

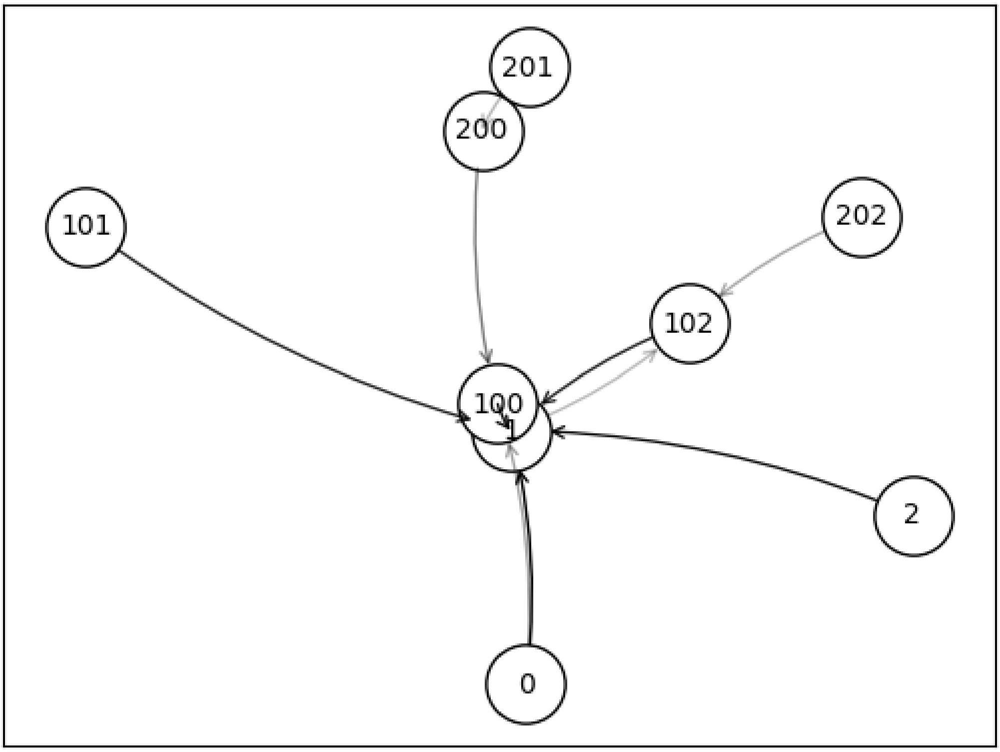

Fig 3. GAT model - The output of the explainer in case of the prediction for node 0.

5. References

- [1710.10903] Graph Attention Networks
- torch_geometric.explain.algorithm.AttentionExplainer — pytorch_geometric documentation