Student ID : 20194293
Name : Go, Kyeong Ryeol

[AI 602] Graph Attention Networks

1. Paper Summary

Graph neural networks was introduced as a generalized version of the recurrent neural network to handle a more general class of graphs such as cyclic, directed and undirected graphs. Basically, it can be applied to data which has non-grid-like structure and consists of the iterative process ahead equilibrium followed by the neural network to compute the output for each node. Here, the author pointed out several limitations on the previous approaches and tackle those issues using the self-attention mechanism to neighbors as a means of computing the hidden representation of each node. The main strength of the proposed idea can be summarized into 3 folds; i) computational efficiency, ii) assigning different importance to different nodes within different size neighborhoods, iii) graph structure agnosticity.

Input and output of the graph attentional layer are both the set of node features with potentially different cardinality, $\mathrm{h} = \{\vec{\mathrm{h}}_1, \ldots, \vec{\mathrm{h}}_\mathrm{N}\}$ and $\mathrm{h}' = \{\vec{\mathrm{h}}'_1, \ldots, \vec{\mathrm{h}}'_\mathrm{N}\}$, respectively. A shared linear combination W and a shared attentional mechanism a are applied to compute the attention coefficients $\mathrm{e}_{\mathrm{ij}} = a(W\vec{h}_i, W\vec{h}_j)$ as the importance of node j's features to node i. Here, to exploit the structural information, the masked attention is applied and to make the coefficients easily comparable across the nodes, the normalization through the soft-max function is further applied

$$\alpha_{\mathrm{ij}} = softmax(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})}$$

Then, a linear combination of the corresponding features is computed potentially following the non-linearity $\sigma(\cdot)$. Furthermore, following Transformer, the usage of the multi-head attention module is further considered, which is nothing but leveraging K independent attention mechanisms and concatenating or averaging the computed features, resulting in the following hidden representation of the node i

$$\vec{h}'_i = \|_{k=1}^{K} \sigma\left(\sum_{j \in N_i} \alpha_{ij}^k W^k \vec{h}_j\right) \; or \; \vec{h}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in N_i} \alpha_{ij}^k W_k \vec{h}_j\right)$$

2. Discussion

I. What would be the pros and cons of concatenation and averaging the features over each other? In terms of the amount of the information, concatenation is better, but when it comes to deal with the noisy data, averaging would be better.

II. Which nodes can be further considered in addition to its neighborhoods? Neighborhood may not be sufficient in some cases when there are some redundant nodes that block estimating the true relationship among others.