Ruben Cardoso
ruben.cardoso@dauphine.eu

Lab Session 7
ALTEGRAD 2025

For 12/9/25

QUESTION 1

1. For a given node i and attention head k, the projected node features are

$$\mathbf{z}_i'^{(k)} = \mathbf{W}^{(k)}\mathbf{z}_i^{(t)} \in \mathbb{R}^{F'_{\text{out}}}.$$

The unnormalized attention coefficients for any neighbor $j \in \mathcal{N}(i)$ are defined as

$$e_{ij}^{(k)} = \text{LeakyReLU}\Big((\mathbf{a}^{(k)})^\top \left[\mathbf{z}_i'^{(k)} \parallel \mathbf{z}_j'^{(k)}\right]\Big).$$

After normalization, the attention weights are given by

$$\alpha_{ij}^{(k)} = \frac{\exp\left(e_{ij}^{(k)}\right)}{\sum_{l\in\mathcal{N}(i)} \exp\left(e_{il}^{(k)}\right)}.$$

The output feature vector produced by head k is then

$$\mathbf{z}_i^{(t+1,k)} = \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(k)}\, \mathbf{z}_j'^{(k)}\right),$$

where $\sigma(\cdot)$ denotes a non-linear activation function, such as sigmoid or tanh.

2. The final node representation is obtained by concatenating the outputs of all K attention heads:

$$\mathbf{z}_i^{(t+1)} = [\mathbf{z}_i^{(t+1,1)} || \dots || \mathbf{z}_i^{(t+1,K)}]$$

Since each head outputs a vector in $\mathbb{R}^{F'_{\text{out}}}$, the final representation has dimension

$$\mathbf{z}_i^{(t+1)} \in \mathbb{R}^{KF'_{\text{out}}}.$$

3. For a single head k, the learnable parameters consist of:

$$\mathbf{W}^{(k)} \in \mathcal{M}_{F'_{\text{out}},F_{\text{in}}}(\mathbb{R}), \qquad \mathbf{a}^{(k)} \in \mathbb{R}^{2F'_{\text{out}}}.$$

This corresponds to $F_{\text{in}}F'_{\text{out}} + 2F'_{\text{out}}$ parameters for each head. Therefore, for K independent attention heads, the total number of learnable parameters is

$$K\big(F_{\text{in}}F'_{\text{out}} + 2F'_{\text{out}}\big) = KF'_{\text{out}}(F_{\text{in}} + 2).$$

QUESTION 2

We assume that all node features are identical:

$$\mathbf{x}_i = \mathbf{c} \in \mathbb{R}^d \quad \text{for all } v_i \in V.$$

1. For a given attention head k, the projected features satisfy

$$\mathbf{z}_i'^{(k)} = \mathbf{W}^{(k)}\mathbf{c} =: \mathbf{u}^{(k)} \quad \text{for all } i.$$

Hence, for any edge (i, j),

$$e_{ij}^{(k)} = \text{LeakyReLU}\Big((\mathbf{a}^{(k)})^\top [\mathbf{u}^{(k)} \,\|\, \mathbf{u}^{(k)}]\Big) = \text{LeakyReLU}\Big(2\,(\mathbf{a}^{(k)})^\top \mathbf{u}^{(k)}\Big) =: \beta^{(k)},$$

which is a constant for the given head k, and does not depend on (i, j). After softmax normalization over the neighbors of i,

$$\alpha_{ij}^{(k)} = \frac{\exp(\beta^{(k)})}{\sum_{\ell \in \mathcal{N}(i)} \exp(\beta^{(k)})} = \frac{1}{|\mathcal{N}(i)|}.$$

That does not depend anymore on k or j.

2. Since all neighbors receive the same attention weight, the GAT layer no longer acts as an adaptive attention mechanism. It reduces to a uniform aggregation rule,

$$\mathbf{z}_i^{(t+1)} = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{z}_j',$$

which is equivalent to a mean-aggregation Graph Convolutional / GraphSAGE-type propagation rule.

3. Repeated neighborhood averaging captures structural properties (e.g. node degrees and community membership). Since the Karate network exhibits a strong community structure aligned with the labels, this structural signal can be sufficient to achieve performance better than random guessing.

TASK 4 :

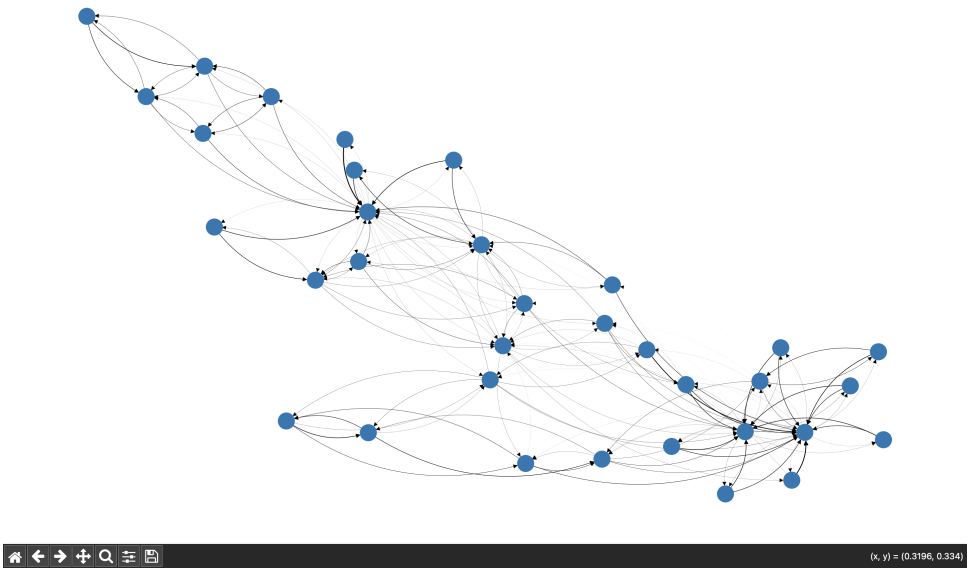

Figure 1: Karate Network Visualization of Attention Weights

QUESTION 3

Let c denote a conditioning variable that encodes prior information about the graph (such as the number of communities, block assignments, or other global properties).

The encoder is then defined as a conditional variational posterior

$$q_\phi(z \mid X, A, c),$$

where the condition c is provided as an additional input.

Similarly, the decoder is conditioned on c and is written :

$$p_\theta(A \mid z, c),$$

so that the reconstructed adjacency matrix $\hat{A}$ depends on both the latent representation z and the condition c.

QUESTION 4:

The reparameterization trick is required to allow gradient-based optimization. Directly sampling $z \sim \mathcal{N}(\mu, \sigma^2 I)$ is not differentiable with respect to μ and σ, which prevents backpropagation.

By rewriting the sampling as

$$z = \mu + \sigma \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I),$$

the stochasticity is isolated in ε, while z becomes a deterministic function of μ and σ. This makes the model fully differentiable and allows gradients to flow through the encoder.

TASK 8:

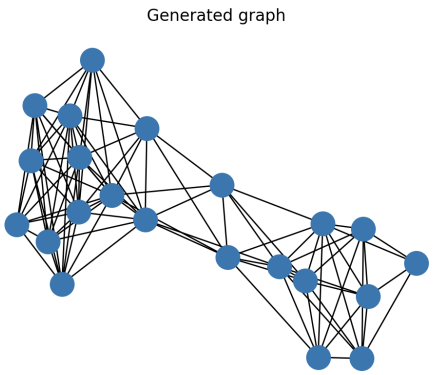

Figure 2: *
Generated graph

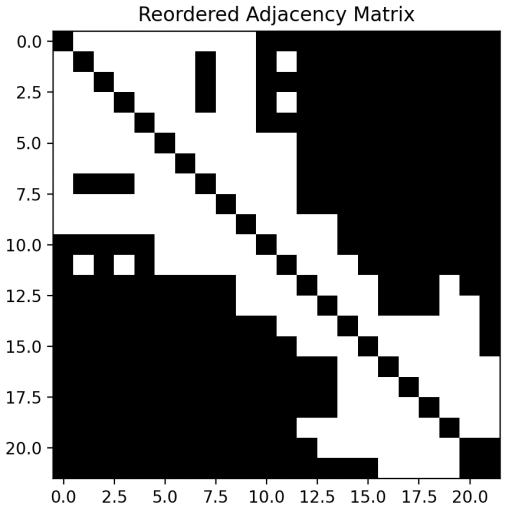

Figure 3: *
Reordered adjacency matrix

Figure 4: Stochastic Block Model dataset: generated graph and reordered adjacency matrix

The generated graph is sampled from the Stochastic Block Model (SBM) dataset, which is characterized by a clear community structure. The model successfully reconstructs this structure: nodes are densely connected within each community, while connections between communities are sparse. The reordered adjacency matrix exhibits a clear block-diagonal pattern, confirming that the VGAE captures the underlying SBM community structure.