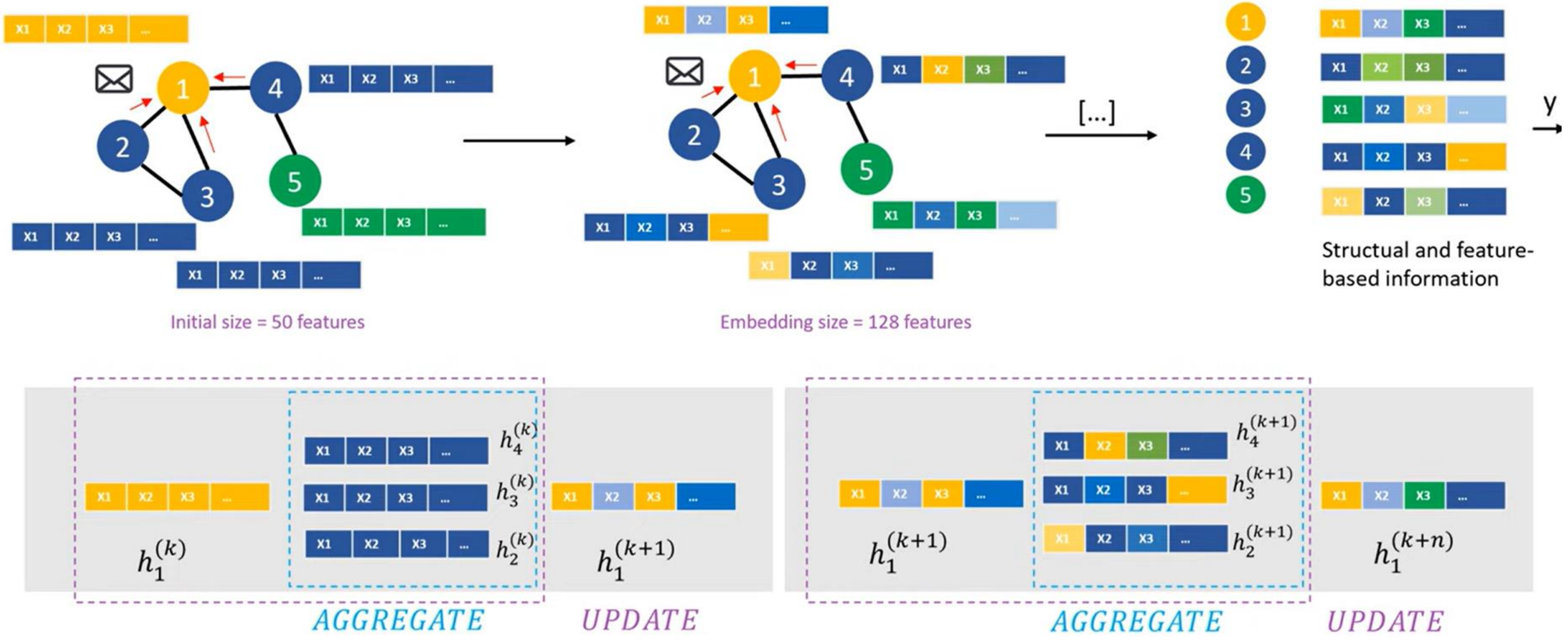
GNN

Piya Amara Palamure

May 13, 2024

Link: Understanding Graph Attention Networks - YouTube

What is happening in the Message Passing Layers?



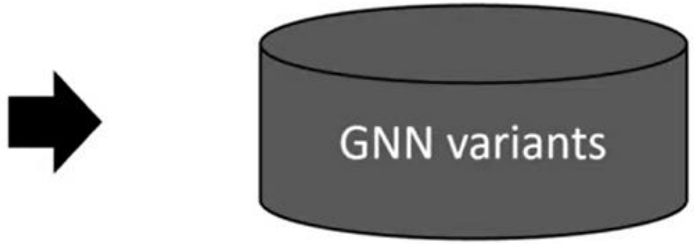
Message Passing Update and Aggregation Functions

Differentiable functions

$$h_u^{(k+1)} = UPDATE^{(k)}\left(h_u^{(k)}, AGGREGATE^{(k)}(\{h_v^{(k)}, \forall v \in \mathcal{N}(u)\})\right)$$

- Mean
- Max
- Neural Network
- Recurrent NN

- Mean
- Max
- Normalized Sum
- Neural Network



GNN variants

AGGREGATE
(permutation invariant)

X1	X2	X3	...
X1	X2	X3	...
X1	X2	X3	...

UPDATE

X1	X2	X3	...

Graph Convolutional Networks, Kipf and Welling [2016]

Self-loop

$$\mathbf{h}_v^{(k)} = \sigma\left(\mathbf{W}^{(k)} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{\mathbf{h}_v}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(v)|}}\right)$$

Sum of normalized neighbor embeddings

Multi-Layer-Perceptron as Aggregator, Zaheer et al. [2017]

Aggregated message

$$\mathbf{m}_{\mathcal{N}(u)} = \boxed{\mathrm{MLP}_\theta}\left(\sum_{v \in N(u)} \mathrm{MLP}_\phi(\mathbf{h}_v)\right)$$

trainable!

Send states through a MLP

Graph Attention Networks, Veličković et al. [2017]

$$\mathbf{m}_{\mathcal{N}(u)} = \sum_{v \in \mathcal{N}(u)} \alpha_{u,v} \mathbf{h}_v$$

Attention weights

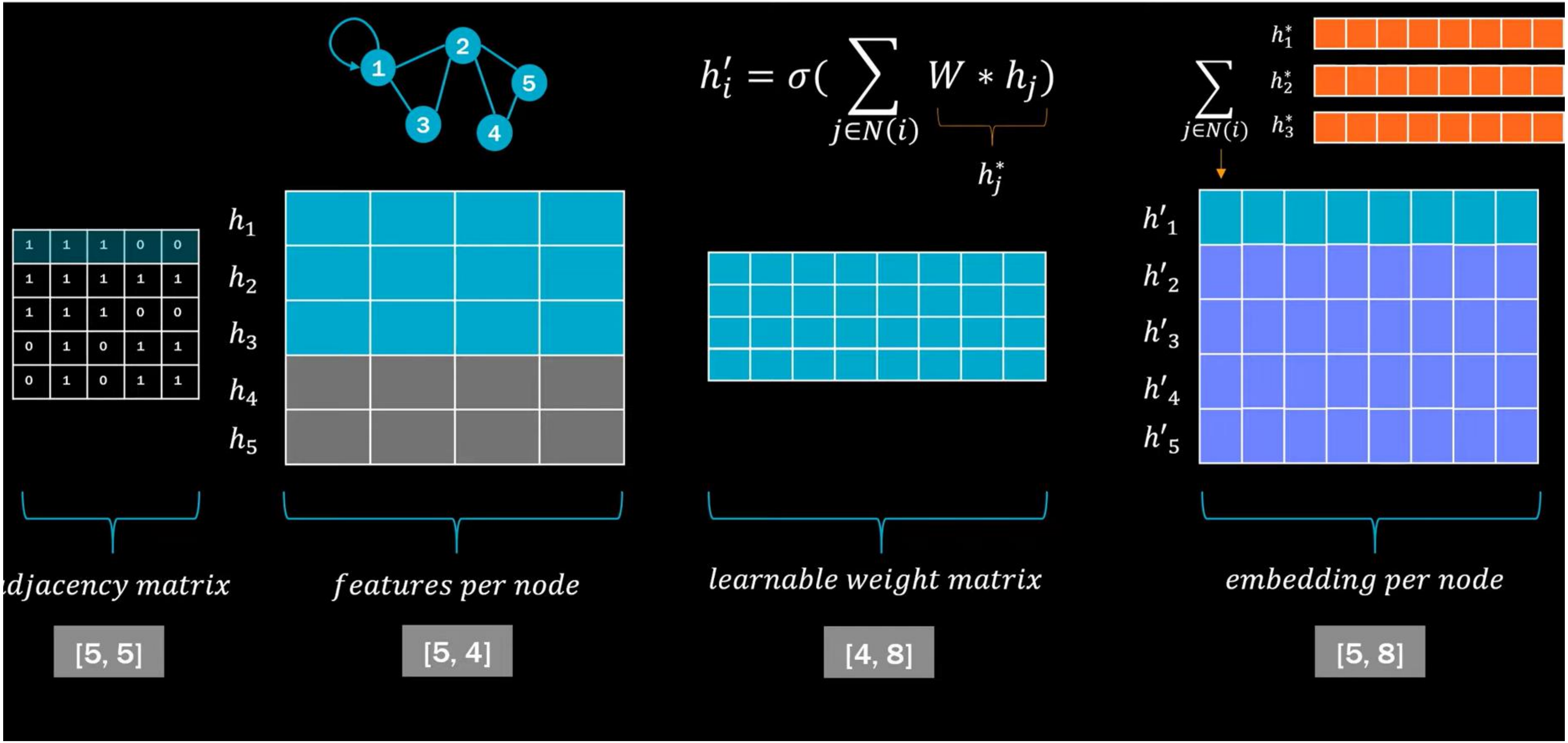
$$\alpha_{u,v} = \frac{\exp\left(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_u \oplus \mathbf{W}\mathbf{h}_v]\right)}{\sum_{v' \in \mathcal{N}(u)} \exp\left(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_u \oplus \mathbf{W}\mathbf{h}_{v'}]\right)}$$

Gated Graph Neural Networks, Li et al. [2015]

$$\mathbf{h}_u^{(k)} = \mathrm{GRU}(\mathbf{h}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)})$$

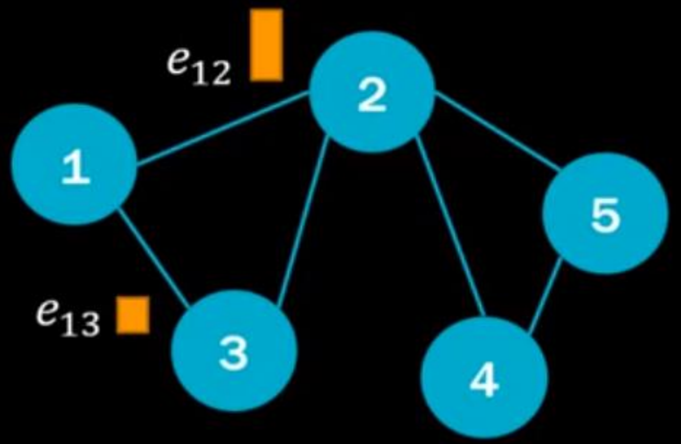
Recurrent update of the state

Graph Neural Networks



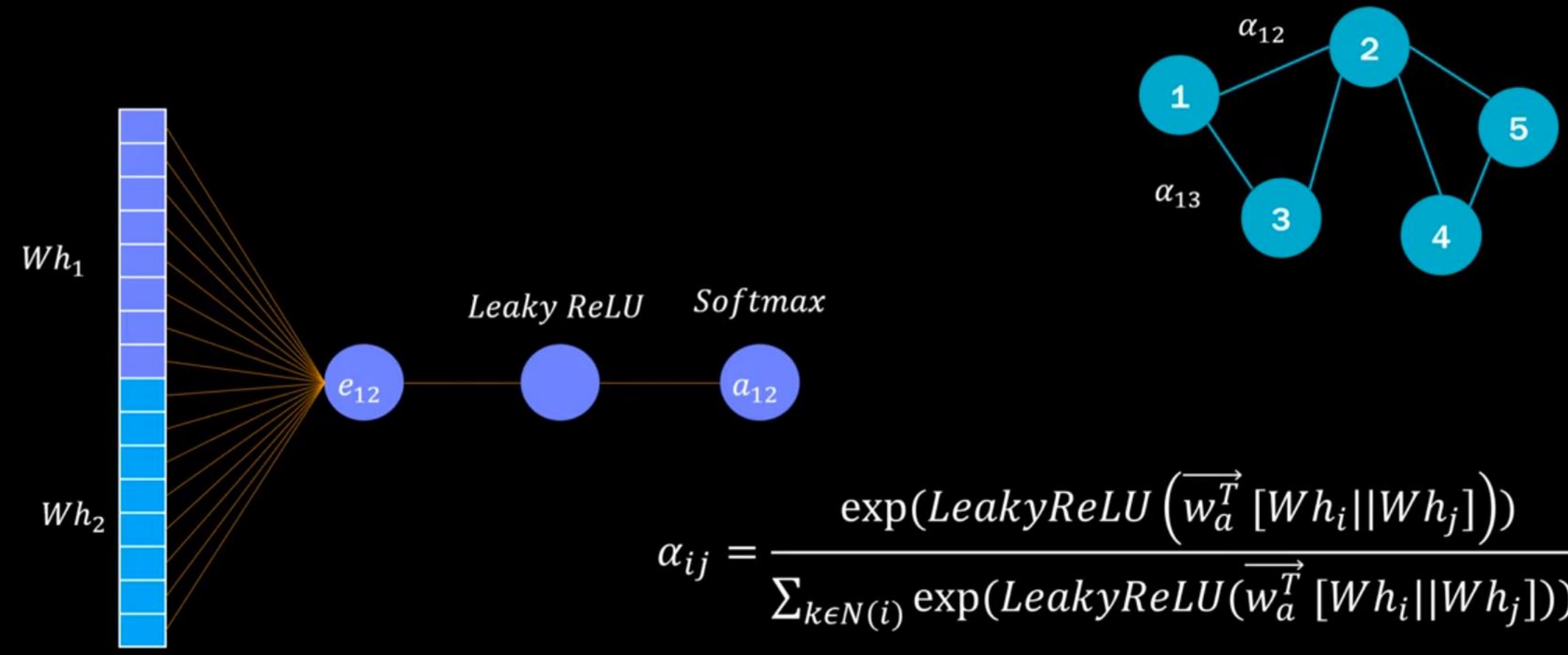
Self attention

$$\underbrace{e_{ij} = a(Wh_i, Wh_j)}_{attention\ coefficient}$$



$$\underbrace{\alpha_{ij} = softmax_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \epsilon N(i)} \exp(e_{ik})} = \frac{\exp(a(Wh_i, Wh_j))}{\sum_{k \epsilon N(i)} \exp(a(Wh_i, Wh_j))}}_{normalized\ attention\ coefficient}$$

The self attention mechanism



$$\alpha_{ij} = \frac{\exp\left(LeakyReLU\left(\overrightarrow{w_a^T}\,[Wh_i||Wh_j]\right)\right)}{\sum_{k\epsilon N(i)} \exp\left(LeakyReLU\left(\overrightarrow{w_a^T}\,[Wh_i||Wh_j]\right)\right)}$$

Overview of Graph attention Neural Network