COL761 Data Mining: HW3

Akash Suryawanshi, Pranjal Agrawal, Harsh Agrawal

November 16, 2022

Task 1

In Task 1, we used GatConv model, Adam as the Optimizer and loss function as Mean Squared Error (MSE).

GAT Architecture

In a single layer, we have input as a set of node features

$$h = \{\vec{h_1}, \vec{h_2}, \vec{h_3}, \ldots, \vec{h_N}\},\ h_i \in \mathbb{R}^F$$

where N is the number of nodes, and F is the number of features in each node. We then apply a shared linear transformation, parametrized by a weight matrix, $W \in R^{F'XF}$ is applied to every node. We then compute attention coefficients by performing self attention on the nodes, $e_{ij} = a(\mathbf{W}\vec{h_i}, \mathbf{W}\vec{h_j})$. We then normalize the coefficients so that they are easily comparable across different nodes.

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N} \exp(e_{ik})}$$

applying the LeakyReLU nonlinearity we get:

$$\alpha_{ij} = \frac{\exp(LeakyReLU(\vec{a}^T[\mathbf{W}\vec{h_i} \| \mathbf{W}\vec{h_j}]))}{\sum_{k \in N} \exp(LeakyReLU(\vec{a}^T[\mathbf{W}\vec{h_i} \| \mathbf{W}\vec{h_k}]))}$$

We then use the obtained α to compute a linear combination of the features corresponding to them to get the output for every node by applying a nonlinearity,

$$\vec{h'} = \sigma\left(\sum_{j \in N_i} \alpha_{ij} \mathbf{W}\vec{h_j}\right)$$

extending our mechanism to employ multi-head attention, we can execute K independent attention mechanisms and then concatenate their features, we get:

$$\vec{h'} = \Big\|_{k=1}^{K} \sigma\left(\sum_{j \in N_i} \alpha_{ij}^k \mathbf{W}^k \vec{h_j}\right)$$

where || represents concatenation, α_{ij}^k are normalized attention coefficients computed by the k-th attention mechanism (a^k), and $\mathbf{W}^k$ is the corresponding input linear transformation's weight matrix.
If we perform multi-head attention on the final layer of the network, instead of concatenation, we employ averaging, and delay applying the final nonlinearity :

$$\vec{h'} = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in N_i}\alpha_{ij}^{k}\mathbf{W}^{k}\vec{h_j}\right)$$

Task 2

In task 2, we used the GMAN Model [1]. The Architecture is given below has an encoderde-coder structure:

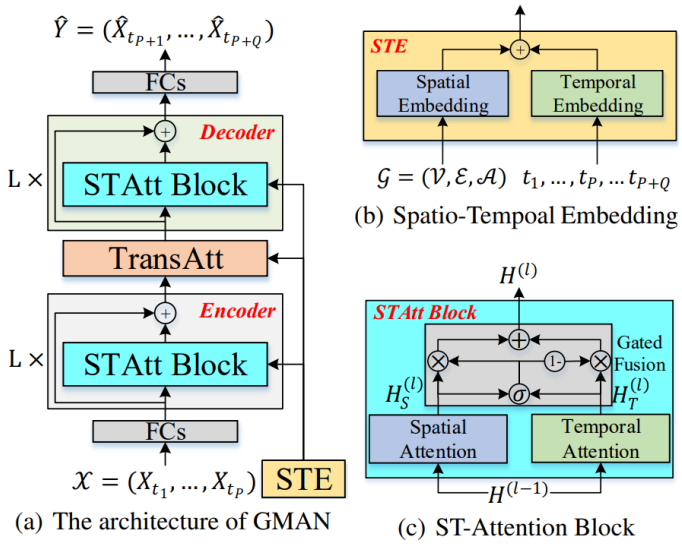

Figure 1: Architecture of GMAN

Each ST-Attention block is composed of spatial and temporal attention mechanisms with gated fusion. To incorporate the road network information into prediction models, we propose a spatial embedding to encode vertices into vectors that preserve the graph structure information. We then leverage the node2vec approach to learn the vertex representations. To co-train the pre-learned vectors with the whole model, these vectors are fed into a two-layer fully-connected neural network. Then, we obtain the spatial embedding, represented as $e_{v_i}^S \in \mathbb{R}^D$, where $v_i \in V$.
We thus further propose a temporal embedding to encode each time step into a vector. Specifically, let a day be with T time steps. We encode the day-of-week and time-of-day of each time step into R^7 and R^T T using one-hot coding, and concatenate them into a vector R^{T+7}. We then apply a two-layer fully-connected neural network to transform the time feature to a vector R^D.

To obtain the time-variant vertex representations, we fuse the aforementioned spatial embedding and temporal embedding as spatio-temporal embedding. . Specifically, for vertex v_i at time step t_j , the STE is defined as $e_{v_i,t_j} = e_{v_i}^S + e_{t_j}^T$ Therefore, the STE of N vertices in $P + Q$ time steps is represented as $E \in R^{(P+Q)} \times N \times D$.
As shown in Figure, the ST-Attention block includes a spatial attention, a temporal attention and a gated fusion.

Contributions

Name	Contribution
Harsh Agrawal	33.33%
Pranjal Aggarwal	33.33%
Akash Suryawanshi	33.33%