Input data:

$\mathbf{X} \in \mathbb{R}^{\mathbf{N*T*F}}$,$\mathrm{N}$: node number, T: sequence length, F: feature number.

$y \in \mathbb{R}$

Problem state:

$$y_t = f_\theta(\mathbf{X_{t-T+1}}, \mathbf{X_{t-T+2}}, \mathbf{...}, \mathbf{X_t}, \mathbf{y_{t-T}}, \mathbf{...}, \mathbf{y_{t-1}})$$

$$\theta^* = \underset{\theta}{\text{minimize}} \sum_{i=1}^{i=N_{test}} (y_i - f_\theta(\cdot))$$

Time lagged Module

计算每个模型输入序列的父节点相对于子节点的时滞

Input data：

- $\mathbf{X_t} \in \mathbb{R}^{\mathbf{N*T}}$, N: node number, T:sequence length.
- $\mathbf{A}$: adjacency matrix, $a_{ij} \neq 0$ means that node i is point to node j, a_{ii} is 0.

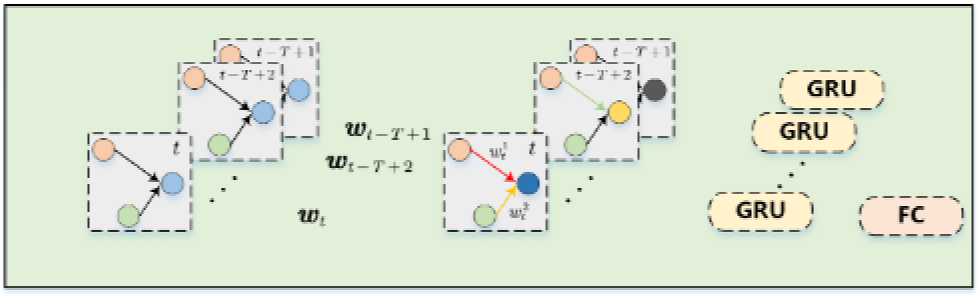
Output data:

- $\mathbf{X_{delay}} \in \mathbb{R}^{\mathbf{N*N*T}}$, ith row jth col element item is the shifted sequence according to the time lagged with its children node, recorded in d_{ji}.
- $\mathbf{D} \in \mathbb{R}^{\mathbf{N*N}}$
- K: max time delay.

$$d_{ji} = max(f_{cr}(\boldsymbol{x}_i \leftarrow k, \boldsymbol{x}_j), \forall k \in (1, 2, ..., \mathrm{K}))$$

f_{cr}: 互相关计算公式，**上式未展开**

AttenGGRU(attention based Graph Gated recurrent units)



不考虑时滞时的formula:

Input data:

- $\mathbf{X_t} \in \mathbb{R}^{\mathbf{N*T}}$
- $\mathbf{X_{delay}}$
- $\mathbf{A}$

Output data:

- y_t

$y_t = f_{aggru}(\mathbf{X_t}, \mathbf{X_{delay}}, \mathbf{A})$, f_{aggru}包括两部分：dynamic local adjacency matrix calculation & data aggregation.

$$\mathbf{H_{i,t}} = \mathbf{aggregate}(\mathbf{H_{i,t-1}}, \mathbf{H_{j,t-1}}), \forall \mathbf{j} \in \mathcal{N}_\mathbf{i}$$

For a single node:
Aggregator这边用的自注意力加权：q: Node i, $kand\ v$: node j, $\forall j \in \mathcal{N}_i$
$q_i = \boldsymbol{x}^i W_Q,\ k_i = \boldsymbol{x}^i W_K,\ v_i = \boldsymbol{x}^i W_V$

$$e_{ij} = softmax(\frac{q_i k_i^T}{\sqrt{d_k}})v_i$$

For matrix form(for computation efficiency):
$\mathbf{Q_t} = \mathbf{X_t W_Q}, \mathbf{K_t} = \mathbf{X_t W_K}, \mathbf{V_t} = \mathbf{X_t W_V}$

$$E = mask(softmax(\frac{QK^T}{\sqrt{d_k}}V))$$

$$\mathbf{H_{i,t}} = \theta_\mathbf{1}\mathbf{H_{i,t-1}} + \theta_\mathbf{2}\sum_{\mathbf{j=1}}^{\mathbf{j}\in\mathcal{N}_i} \mathbf{e_{ij}}\mathbf{H_{j,t-1}}$$

used attention based graph convolution as matrix operation of GRU, can get the formulation:
traditional GRU:
Denotation: j:jth unit.

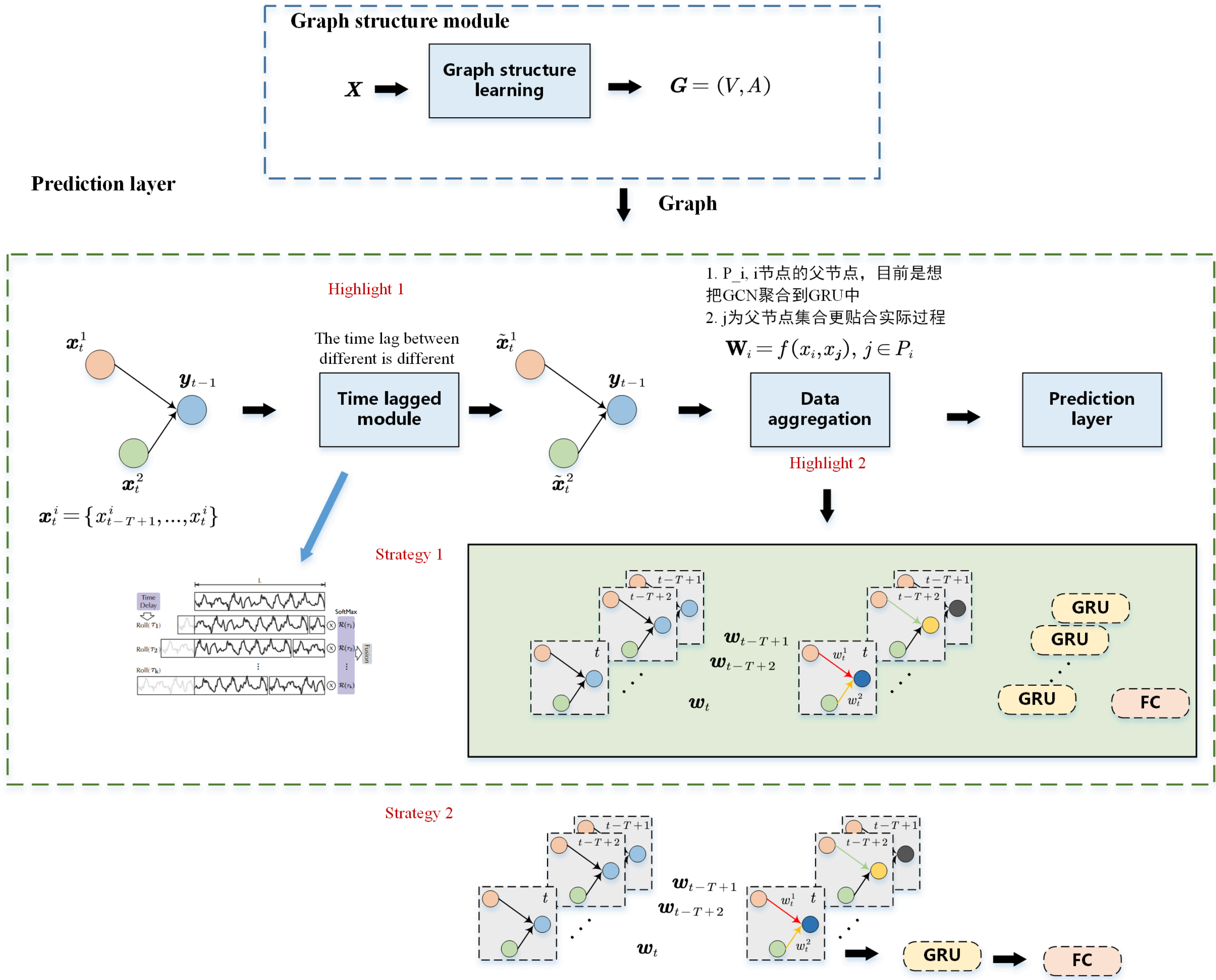
$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j\tilde{h}_t^j, \quad (liear\ interpolation)$$

$$z_t^j = \sigma(W_z\boldsymbol{x}_t + U_z\boldsymbol{(}h_{t-1})) \quad (update\ gate)$$

$$\tilde{h}_t^j = \tanh(W\boldsymbol{x}_t + U(\boldsymbol{r}_t \odot \boldsymbol{h}_t))^j \quad (candidate\ activation)$$

$$r_t^j = \sigma(W_r\boldsymbol{(}x)_t + U_r\boldsymbol{h}_{t-1})^j, \quad (reset\ gate)$$

For AGGRU: replaced matrix W_z, W, W_R with attention based graph convolution.


Graph structure module
X
Graph structure learning
G = (V, A)
Prediction layer
Graph
Highlight 1
1. P_i, i节点的父节点，目前是想
把GCN聚合到GRU中
2. j为父节点集合更贴合实际过程
Wi = f(xi, xj), j ∈ Pi
The time lag between
different is different
Time lagged module
Data aggregation
Prediction layer
Highlight 2
Strategy 1
GRU
GRU
GRU
FC
Strategy 2
GRU
FC