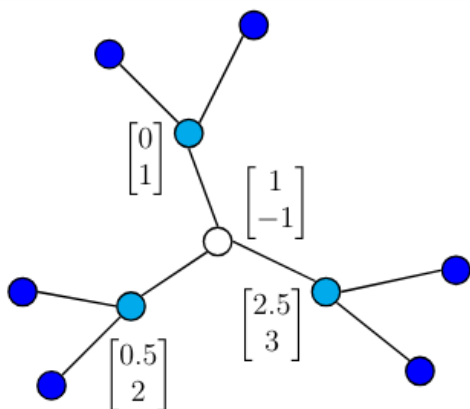


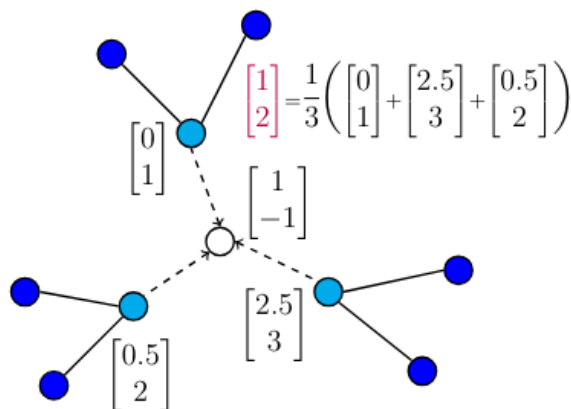
## Обзор GNN

GNN (*Graph neural network*) - архитектура, которая фокусируется на использовании информации о соседях вершины в графовой структуре. Широкий обзор подходов в GNN представлен в [Jie Zhou. et.al., 2020](#).

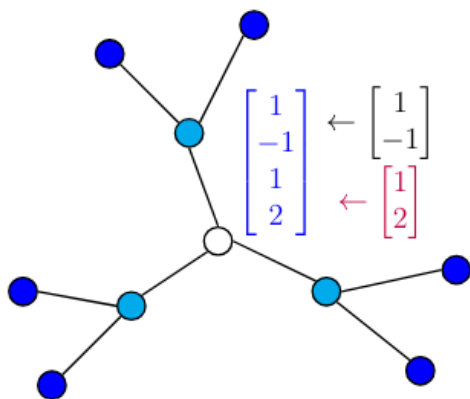
Основной модуль в GNN - это Graph Convolution (GC module). Пусть дан граф  $(V, E, G)$ , где  $V$  - множество вершин,  $E$  - мн-во ребер,  $G$  - матрица признаков вершин. У каждой вершины есть  $n$ -мерный вектор признаков. На одной итерации для каждой вершины признаки её соседей умножаются на матрицу обучаемых параметров  $W$ , а затем усредняются с признаками этой вершины.



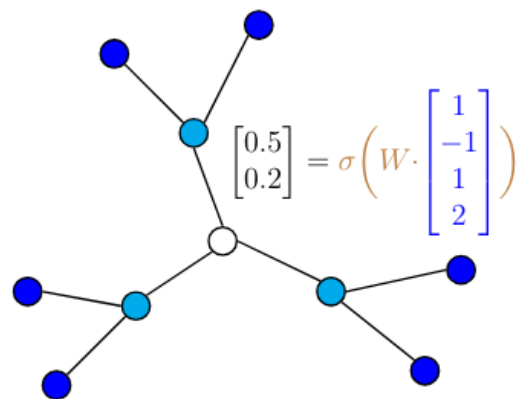
(1) Исходные векторы признаков



(2) Агрегируем признаки соседей



(3) Комбинируем векторы признаков



(4) Вычисляем новые векторы признаков

## Spatio-temporal GNN

STGNN (*Spatio-temporal GNN*) обобщает архитектуру GNN на случай временных данных. Пусть дан тот же граф, но теперь признаки вершин - не  $n$ -мерные векторы, а  $t \times n$ -мерные, i.e. признак каждой вершины - это матрица  $X \in R^{t \times n}$ , где столбец  $j$  - это признаки, измеренные в момент времени  $t_j$ .

Теперь после GC module применяется свертка по времени (например, LSTM), т.е. после усреднения информации о соседях работает LSTM независимо для каждой вершины. Итоговый вектор состояний передается на следующий слой графовой свертки.

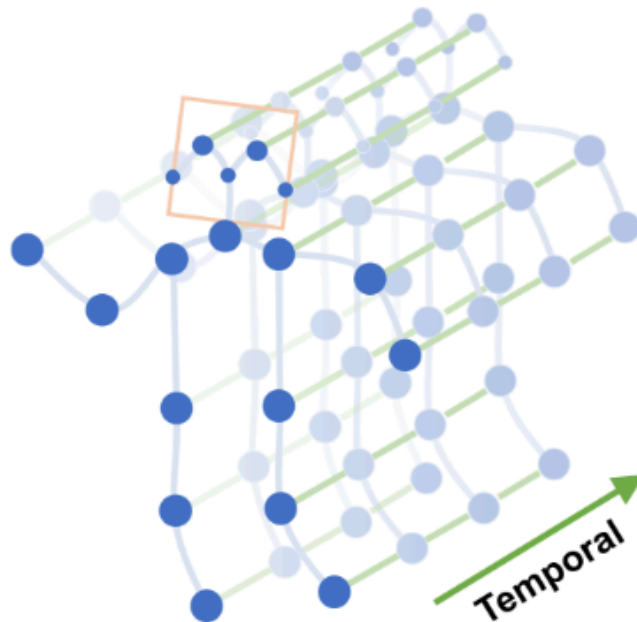


Figure 1: The spatial temporal graph of a skeleton sequence used in this work where the proposed ST-GCN operate on. Blue dots denote the body joints. The intra-body edges between body joints are defined based on the natural connections in human bodies. The inter-frame edges connect the same joints between consecutive frames. Joint coordinates are used as inputs to the ST-GCN.

## MTGNN

MTGNN (*Multivariate Time Series Graph Neural Network*) продолжает идею STGNN, но идёт дальше: авторы предлагают обучать саму графовую структуру вместо использования эвристик для её инициализации.

Пусть дано  $n$  временных рядов размерности  $T$ . Каждый временной ряд можно представить вершиной в некотором графе. Так, у каждой вершины есть история её состояний в моменте времени  $1 \dots T$ . Теперь изменим LSTM модули в STGNN на сверточные слои, и получим архитектуру MTGNN.

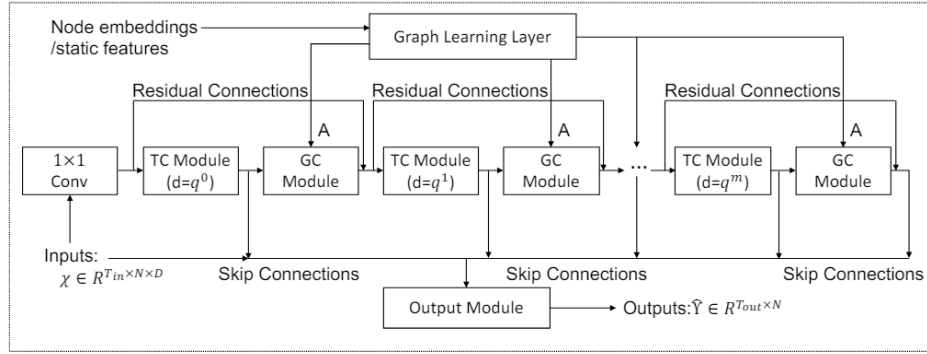


Figure 2: The framework of MTGNN. A  $1 \times 1$  standard convolution first projects the inputs into a latent space. Afterward, temporal convolution modules and graph convolution modules are interleaved with each other to capture temporal and spatial dependencies respectively. The hyper-parameter, dilation factor  $d$ , which controls the receptive field size of a temporal convolution module, is increased at an exponential rate of  $q$ . The graph learning layer learns the hidden graph adjacency matrix, which is used by graph convolution modules. Residual connections and skip connections are added to the model to avoid the problem of gradient vanishing. The output module projects hidden features to the desired dimension to get the final results.

## Graph learning layer

Авторы обучают структуру ориентированного графа с несимметричной матрицей смежности  $A$  без петель. Такая структура удобна для модуля графовой свертки, куда одновременно подаются на вход  $A$  и  $A^T$ . “In multivariate time series forecasting, we expect that the change of a node’s condition causes the change of another node’s condition such as traffic flow. Therefore the learned relation is supposed to be uni-directional.” (При прогнозировании многомерных временных рядов мы ожидаем, что изменение состояния вершины вызывает изменение состояния другой вершины, например, транспортного потока. Поэтому предполагается, что полученная взаимосвязь является однонаправленной.)

$$M_1 = \tanh(\alpha E_1 \Theta_1)$$

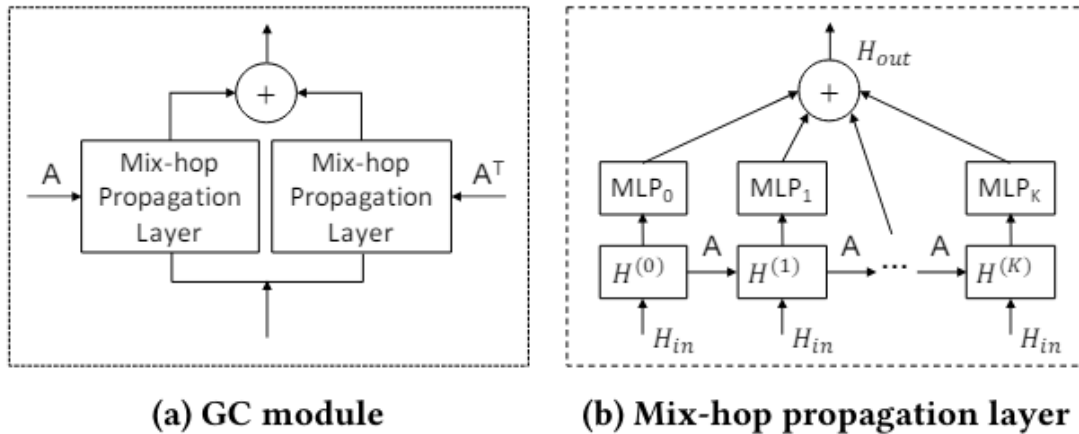
$$M_2 = \tanh(\alpha E_2 \Theta_2)$$

$$A = \text{ReLU}(\tanh(\alpha(M_1 M_2^T - M_2 M_1^T)))$$

Далее, в каждой строке выбирается top-k наибольших значений, а остальные обнуляются, чтобы сделать матрицу более разреженной.  $E_1$ ,  $E_2$  - обучаемые эмбединги вершин,  $\Theta_1$ ,  $\Theta_2$  - обучаемые параметры,  $\alpha$  - гиперпараметр, аналогичный температуре. Чем выше  $\alpha$ , тем ближе положительные значения в  $M_1$ ,  $M_2$  ближе к 1, и тем меньше значений обнуляется с помощью  $\text{ReLU}(\cdot)$  в последнем уравнении.

Две матрицы  $E_i$ ,  $\Theta_i$  нужны для того, чтобы можно было встроить внешние данные о структуре графа, если таковые имеются. Достаточно положить  $E_1 = E_2 = Z$

## Graph Convolution module (GC module)



**Figure 4: Graph convolution and mix-hop propagation layer.**

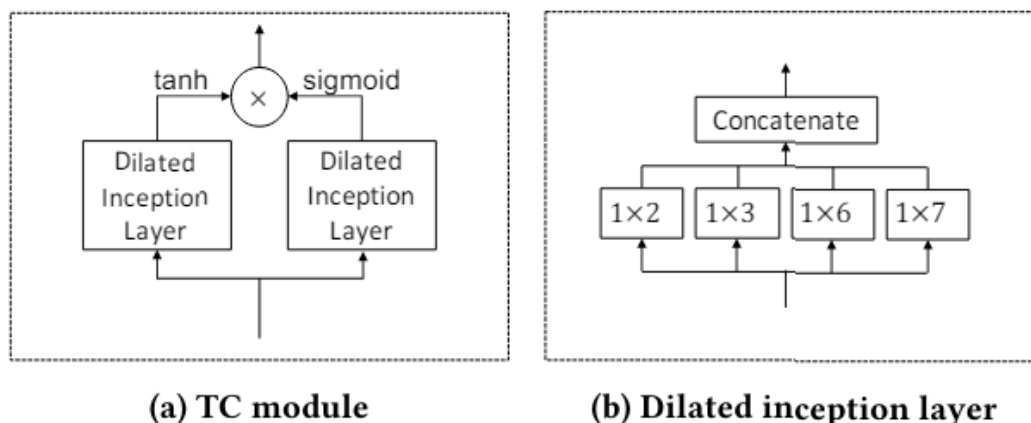
Пусть  $D^* = 1 + \sum_j A_{ij}$ ,  $A^* = D^{*-1}(A + I)$ ,  $H^{(0)} = H_{in}$ . Тогда на каждом шаге свертку по

соседям можно написать как  $H^{(k)} = \beta H_{in} + (1 - \beta)A^*H^{(k-1)}$ .  $H_{out} = \sum_{i=0}^K H^{(i)}W^{(i)}$

По существу,  $A^*$  - нормализованная по соседям матрица смежности, т.е. теперь для каждой вершины каждое ребро до её соседа отнормированно на сумму ребер до всех соседей этой вершины.

Два слоя помогают агрегировать и входящие, и выходящие связи из каждой вершины. *"The graph convolution module consists of two mixhop propagation layers to process inflow and outflow information passed through each node separately"* (Слой графовой свертки состоит из двух mixhop слоёв для обработки информации о притоке и оттоке, проходящей через каждый узел отдельно).

## Time Series Convolution module (TC module)



**Figure 5: The temporal convolution and dilated inception layer.**

На вход в TC module подается последовательность состояний каждой вершины и независимо применяется одномерная свертка одновременно для 4 фильтров. Выходы этих фильтров конкатенируются в один вектор, который подается дальше в сеть. Наличие 4 фильтров с размерностью ядер 2,3,6 и 7 обусловлено желанием проходить сверткой по часто встречающимся во временных рядах сезонностям: 7, 12, 24, 28 и 60.

Используется dilated-свертка (*dilated inception layer*), чтобы расширить рецептивное поле каждого нейрона. В идеальной ситуации последние слои свертки должны смотреть на весь изначальный временной ряд.

Два параллельных слоя нужны для реализации “фильтрации” и аналога ворот внимания в LSTM. “One dilated inception layer is followed by a tangent hyperbolic activation function and works as a filter. The other layer is followed by a sigmoid activation function and functions as a gate to control the amount of information that the filter can pass to the next module.” (За одним *dilated inception layer* следует гиперболическая функция активации, которая работает как фильтр. За другим слоем следует сигмоидная функция активации, которая выполняет функцию управления объемом информации, которую фильтр может передавать следующему модулю.)

## Источники

1. Sijie Y., Yuanjun X., Dahua L. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition // 2018
2. Zahraa A. Spatio-Temporal Graph Neural Networks: A Survey // 2023
3. Jie Z., et al. Graph neural networks: A review of methods and applications // 2020
4. Zonghan W., et al. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks // 2020