

DOI 10.37882/2223-2966.2023.08.09

# ИСПОЛЬЗОВАНИЕ ГРАФОВЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КЛАССИФИКАЦИИ ВЕРШИН ГРАФА

## USE OF GRAPH NEURAL NETWORKS FOR SOLVING THE PROBLEM OF CLASSIFICATION OF GRAPH VERTICES

R. Vasiliev

*Summary.* The tasks of classifying objects are encountered quite often in the framework of work in the industry. In most cases, they are solved on the basis of basic characteristics (by classical machine learning algorithms), without taking into account graph information. The hypothesis of the study was that methods that take into account the graph structure will give higher quality models.

In this paper, several approaches to solving the classification problem using graph information are presented and tested. They are based on real data from a large Russian telecommunications company. Such methods have not been used in the Russian telecom before.

Ultimately, in 2 out of 3 studied projects, it was possible to obtain increases in the quality of the models.

*Keywords:* neural networks, graph information, graph vertices.

Васильев Роман Александрович

Московский государственный университет  
r.a.vasiliev1998@gmail.com

*Аннотация.* Задачи классификации объектов встречаются достаточно часто в рамках работы в индустрии. В большинстве случаев они решаются на основе базовых характеристик (классическими алгоритмами машинного обучения), без учёта графовой информации. Гипотеза исследования заключалась в том, что методы, которые учитывают графовую структуру, дадут более высокое качество моделей.

В данной работе приведены и протестированы несколько подходов к решению задачи классификации с использованием графовой информации. Они основываются на реальных данных крупной российской телекоммуникационной компании. Подобные методы в российском телекоме прежде не использовались.

В конечном итоге в 2 из 3 исследуемых проектов удалось получить приросты качества моделей.

*Ключевые слова:* нейронные сети, графовая информация, вершин графа.

Нейронные сети (в частности — свёрточные нейронные сети) произвели большой бум в области обработки изображений, сигналов и текста. Но напрямую применить их к графам невозможно — графы, в отличие от описанных сущностей, не имеют чёткой точки отсчёта, порядка и, как правило, динамические. Решить эту проблему позволяют графовые нейронные сети и более продвинутые подходы — свёрточные графовые нейронные сети (GCN, Graph Convolutional Networks). Они агрегируют информацию наборов вершин, что позволяет чётко отобразить графовую структуру, и, кроме того, способны работать с характеристиками вершин графа.

Формализуем задачу.

Дан граф  $G$ , в котором:

- $V$  — набор вершин;
- $A$  — матрица смежности (предполагаем бинарную);
- $X$  — матрица характеристик вершин

Метод GCN напрямую опирается на структуру графа и характеристики «соседей». Основная идея этого подхода заключается в использовании своего рода «свёрток» для вершин графа. Однако, в отличие от изображений, в графовых нейронных сетях именно сам граф будем определять структуру нейронной сети. На (Рис. 1) представлен алгоритм обработки вершины  $i$ . Её эмбединг

будет строиться на основе многоступенчатой агрегации и преобразования признаков окружающих вершин. Главным вопросом является то, как строить эти агрегаты, т.е. как распространять информацию сквозь граф. Рассмотрим идею, предложенную William L. Hamilton, Rex Ying, Jure Leskovec в [1].

Самый базовый подход — усреднять входящие в оператор вектора и применять к результату нейронную сеть. Тогда, согласно описанному выше:

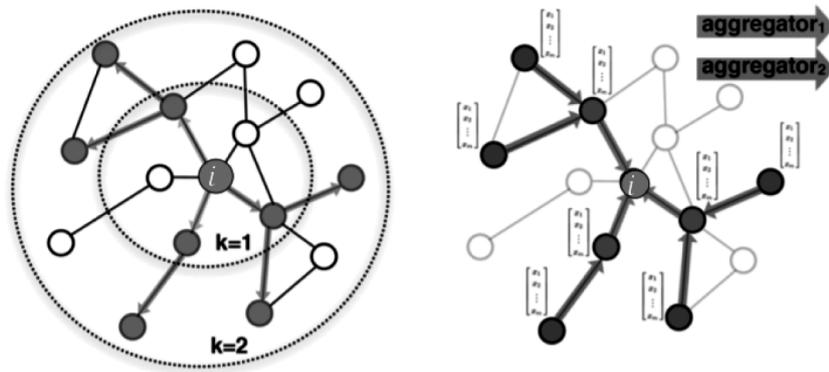
$$\begin{cases} h_v^0 = x_v \\ h_v^k = \sigma(W_k \sum_{u \in N(v)} \frac{h_u^{k-1}}{|N(v)|} + B_k h_v^{k-1}), \forall k \in 1, \dots, K, \\ z_v = h_v^K \end{cases}$$

где  $h_v^i$  — представление эмбединга на  $i$ -м слое;  
 $\sigma(x)$  — некоторое нелинейное преобразование;  
 $x_v$  — вектор признаков вершины  $v$ ;  
 $z_v$  — итоговый эмбединг вершины  $v$ ;  
 $W_k, B_k$  — параметры сети.

Чтобы подобрать параметры сети, нужно определить функцию потерь и запустить стохастический градиентный спуск.

Далее есть 2 пути:

- Unsupervised Learning — обучение без учителя. В этом случае можно заняться решением задачи



Формирование вычислительного графа для вершины  $i$

Распространение и преобразование информации

Рис. 1. Соседи вершины  $i$  определяют алгоритм расчёта эмбединга для неё

построения эмбедингов вершин. В таком случае, в качестве метрики можно взять ту же, что и в методе node2vec;

- Supervised Learning — обучение с учителем. Примером такой задачи может служить определение класса вершин. В этом случае известно к какому классу принадлежат те или иные вершины. Тогда, на основе графовой информации (методами GCN) можно предсказать, к какому из них принадлежат остальные. Например, с помощью этого метода на основе данных телекоммуникационной компании можно определить является ли данный человек фанатом той или иной команды или нет.

Для решения задачи бинарной классификации адаптируем Cross-entropy loss под нашу задачу. Тогда функция потерь будет выглядеть следующим образом:

$$L = \sum_{v \in V} y_v \log(\sigma(z_v^T \theta)) + (1 - y_v) \log(1 - \sigma(z_v^T \theta)),$$

- где  $z_v$  — эмбединг вершины  $v$ ;
- $\theta$  — вектор весов для классификации;
- $y_v \in \{0, 1\}$  — реальный класс вершины.

Несомненным преимуществом данного подхода является то, что имеется возможность обучаться (подбирать параметры  $W_k$  и  $B_k$  на части графа). Затем, основываясь на них, мы можем рассчитать предсказание для новых вершин, не пересчитывая всю модель.

GCN — самая базовая архитектура, которая часто применяется на практике. Тем не менее, распространены и применяются в том числе следующие подходы:

- Graph Neural Networks (GNN) — классические графовые нейронные сети, без свёрток;
- GraphSAGE — развитие идей GCN, с более широким спектром возможных функций агрегации
- Graph Attention Networks (GAT) — графовые нейронные сети с механизмом внимания, которые позволяют автоматически (с помощью отдельной сети) настраивать веса для агрегации.

В рамках исследования было решено провести тестирование моделей на трёх проектах:

- Определение абонентов-мошенников. Одной из важных задач компании является выделение абонентов-спамеров, которые проявляют подозрительную активность и выполняют неправомерные действия;
- Предсказание оттока пользователей. Задача заключается в прогнозировании склонности клиента сменить оператора и во многом завязана на графовой структуре (если контакты рассматриваемого клиента меняют оператора, то и он с высокой вероятностью это сделает);
- Модель предсказания пола абонента. В современных реалиях маркетинг является неотъемлемой частью работы телекоммуникационной компании. Часто возникает задача определения демографических характеристик пользователей (пол, возраст и т.д.). Эти параметры указываются при оформлении номера, но фактически часто симкартами пользуются не те люди, на которых они оформлены, поэтому задача весьма актуальна.

Все описанные выше задачи сводятся к построению моделей бинарной классификации на уровне абонентов (вершин графа).

В рамках компании уже были разработаны классические модели машинного обучения (на основе XGBClassifier). Тем не менее, они не учитывают графовую структуру, поэтому ожидается получить приросты при использовании графовых подходов. При сравнении моделей рассматривается прирост метрики в процентном соотношении от первоначального (т.е. если метрика была 0.8, а стала 0.88, то прирост составил 10 %). В качестве целевой метрики использовался Precision on top.

Все модели обучались с тщательным подбором гиперпараметров (с помощью кросс-валидации). В качестве функционала потерь использовался Cross entropy

loss, в качестве оптимизатора — Adam optimizer (тестировался в том числе SGD optimizer, на Adam показал лучшее качество). За основу программных реализаций были взяты модули GCNConv, GATConv, SAGEConv, GeneralConv из открытой библиотеки torch geometric.

В конечном итоге удалось получить следующие приросты:

Модель	GNN	GCN	GraphSAGE	GAT
Отток	0.11%	0.86%	0.84%	1.15%
Мошенники	0.46%	0.95%	1.13%	1.67%
Пол	<0%	<0%	0.17%	0.13%

Таким образом, в задачах Оттока и Поиске Мошенников использование графовых нейронных сетей действительно позволило получить более высокое качество моделей. Стоит отметить, что подходы GraphSAGE и Graph Attention Network показали себя более гибкими в применении и дали значимые приросты метрик.

#### ЛИТЕРАТУРА

1. William L. Hamilton, Rex Ying, Jure Leskovec: Representation Learning on Graphs: Methods and Applications: <https://arxiv.org/pdf/1709.05584.pdf>
2. Yoav Goldberg and Omer Levy: word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method <https://arxiv.org/pdf/1402.3722.pdf>
3. S. Cao, W. Lu, and Q. Xu. Deep neural networks for learning graph representations. In AAAI, 2016

© Васильев Роман Александрович (r.a.vasiliev1998@gmail.com)  
Журнал «Современная наука: актуальные проблемы теории и практики»