Обзор сверточных нейронных сетей на графах

Илья Игашов

16 декабря 2019 г.

В последние годы значительно возрос интерес научного сообщества к вопросу создания алгоритмов машинного обучения на нерегулярных структурах данных, таких как графы и выпуклые многообразия. Учитывая то, насколько успешным оказалось использование сверточных нейронных сетей в области компьютерного зрения, в последние годы особо остро встал вопрос применения операции свертки к нерегулярным структурам типа графов. В силу отсутствия четкой структуры на множестве узлов графа, вопрос построения свертки в данном случае становится нетривиальным, так как в классической теории обработки сигналов свертка определеятся через оператор трансляции, смысл которого утрачивается, если мы говорим о нерегулярных структурах типа графов.

В данном обзоре мы рассмотрим два метода определения операции свертки на графах: спектральный и пространственный. Спектральный метод основан на применении теории Фурье к графам [7] – в рамках этого подхода удалось получить математическое выражение для операции свертки на графах, а также использовать полученную операцию для обучения сверточных нейронных сетей. Второй метод, пространственный, основан на более житейском и логическом (и менее математическом) подходе к формулированию понятия свертки в терминах графов, он является более интерпретируемым и универсальным.

Как мы увидим ниже, в случае с графами сверточным нейросетям необходимо, чтобы узел в графе представлялся как вектор признаков. Одним из возможных методов синтеза признаков является получение эмбеддингов. В данном обзоре мы познакомимся с простым способом отображать узлы графа в d-мерное вещественное пространство [5].

1. Обозначения и постановка задачи

Будем рассматривать неориентированный взвешенный связный граф $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, где $\mathcal{V}$ – множество вершин, $|\mathcal{V}| = N$, $\mathcal{E}$ – множество ребер, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Для удобства пронумеруем все вержины в графе от 1 до N и будем считать, что $\mathcal{V} = \{1, \dots, N\}$. Обозначим через $\mathbf{W}$ матрицу смежности графа: $\mathbf{W}_{ij} = w(i,j)I_{\mathcal{E}}(i,j)$, где $w : \mathcal{E} \to \mathbb{R}$ – функция веса ребер графа $\mathcal{G}$, а $I_{\mathcal{E}}$ – индикаторная функция множества $\mathcal{E}$. Степенью i-й вершины графа $\mathcal{G}$ называется величина $d_i = \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij}$, где $\mathcal{N}_i = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$ – множество соседей вершины i.

Будем считать, что каждая вершина графа обладает вектором параметров $\mathbf{x} \in \mathbb{R}^m$, и, таким образом, граф может быть представлен парой двух матриц: $(\mathbf{X}, \mathbf{W})$, где $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)^\top$.

Следует отметить, что существует множество различных задач, связанных с графами. В частности, в задачах машинного обучения на графах целевая функция может быть определена как на множесте графов, так и на множестве вершин графов. Для простоты в этом обзоре мы будем говорить о втором варианте.

2. Спектральный подход

Спектральная теория графов

Рассмотрим некоторую функцию $f : \mathcal{V} \to \mathbb{R}$, определенную на множестве $\mathcal{V}$ вершин графа $\mathcal{G}$. Функцию f можно представить как вектор $\mathbf{f} \in \mathbb{R}^N$ (сигнал функции f), i-я компоненкта которого равна значению функции f на i-й вершине графа $\mathcal{G}$ (в наших терминах сигнал является признаком вершины графа $\mathcal{G}$). Определим Лапласиан графа $\mathcal{G}$ как матрицу $\mathbf{L} = \mathbf{D} - \mathbf{W}$, где $\mathbf{D} = \operatorname{diag}\{d_1, \dots, d_N\}$ – диагональная матрица степеней вершин графа $\mathcal{G}$. Лапласиан является оператором разности на множестве функций $f : \mathcal{V} \to \mathbb{R}$, поскольку, как легко заметить, для $\mathbf{f} \in \mathbb{R}^N$ справедливо равенство:

$$(\mathbf{L}f)(i) = \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij}[f(i) - f(j)]. \tag{1}$$

Поскольку Лапласиан графа является симметричной вещественной матрицей, для нее существует полный набор ортонормированных собственных векторов $\{\mathbf{u}_l\}_{l=1}^N$ и соответствующий ему набор неотрицательных

вещественных собственных значений (или частот, по аналогии с частотами сигнала в разложении Фурье) $\{\lambda_l\}_{l=1}^N$. Положим $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_N)$.

Обратимся к классической теории обработки сигналов и к теории Фурье в пространстве вещественных функций. Если поставить в соответствие бесконечномерному базису $\{e^{2\pi i\xi t}\}_{\xi\in\mathbb{R}}$ одномерного Лапласиана Δ_t наш набор собственных векторов $\{\mathbf{u}_l\}_{l=1}^N$, а множеству частот $\xi \in \mathbb{R}$ – наш набор собственных значений $\{\lambda_l\}_{l=1}^N$, то мы сможем определить преобразование Фурье на пространстве функций $f : \mathcal{V} \to \mathbb{R}$:

$$\hat{f}(\lambda_l) = \langle \mathbf{f}, \mathbf{u}_l \rangle = \sum_{i=1}^{N} f(i) u_l(i). \tag{2}$$

Также можно получить формулу для обратного преобразования Фурье:

$$f(i) = \sum_{l=0}^{N} \hat{f}(\lambda_l) u_l(i). \tag{3}$$

Операция свертки

Особенностью работы с графами является то, что множество вершин графа не обладает четкой и однозначной структурой. В частности, для функций на графах невозможно определить оператор трансляции, потому что попросту непонятно, что значит "$i - j$" для двух вершин $i, j \in \mathcal{V}$. Этот факт не позволяет использовать оригинальное определение операции свертки:

$$(f * g)(t) = \int_{\mathbb{R}} f(\tau) g(t - \tau) d\tau. \tag{4}$$

Но тут к нам на помощь приходит понятие фильтра из классической теории обработки сигналов. Фильтром называется функция $\hat{h}$, роль которой – усиливать или ослаблять вклад каких-либо частот ξ в выходной сигнал:

$$\hat{f}_{\mathrm{out}}(\xi) = \hat{f}_{\mathrm{in}}(\xi)\hat{h}(\xi). \tag{5}$$

С помощью обратного преобразования Фурье выходного сигнала можно получить операцию свертки:

$$f_{\mathrm{out}}(t) = \int_{\mathbb{R}} \hat{f}_{\mathrm{in}}(\xi)\hat{h}(\xi) e^{2\pi i\xi t} d\xi = \int_{\mathbb{R}} f_{\mathrm{in}}(\tau) h(t - \tau) d\tau = (f_{\mathrm{in}} * h)(t). \tag{6}$$

Таким образом, можно записать определение операции свертки для функций на графе $\mathcal{G}$:

$$(f * g)(i) = \sum_{l=1}^{N} \hat{f}(\lambda_l)\hat{g}(\lambda_l) u_l(i). \tag{7}$$

Пользуясь формулой (2), запишем формулу свертки в матричном виде:

$$(f * g)(i) = \sum_{l=1}^{N} \langle \mathbf{f}, \mathbf{u}_l \rangle \langle \mathbf{g}, \mathbf{u}_l \rangle u_l(i) = \mathbf{U}(\mathbf{U}^\top \mathbf{f} \odot \mathbf{U}^\top \mathbf{g}) = \mathbf{U}\mathbf{G}\mathbf{U}^\top \mathbf{f}, \tag{8}$$

где $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_N)$ – матрица базисных векторов, $\odot$ – поэлементное произведение, а $\mathbf{G} = \mathrm{diag}(\mathbf{U}^\top \mathbf{g})$.

Спектральные сверточные сети на графах

Результат свертки полностью зависит от значений матрицы $\mathbf{G}$, и она может выступать, например, непараметрическим фильтром, т.е. матрицей, где все параметры оптимизируются. Так мы получаем формулу для сверточного слоя из работы Spectral Convolutional Neural Network [1]:

$$\mathbf{f}'_j = \sigma\left(\sum_{i=1}^{d_{\mathrm{in}}} \mathbf{U}\mathbf{G}_{ij}(\theta)\mathbf{U}^\top \mathbf{f}_i\right), \tag{9}$$

где на вход сверточному слою подается $\mathbf{F} = (\mathbf{f}_1, \dots, \mathbf{f}_{d_{\mathrm{in}}})$ – матрица входного сигнала, $\mathbf{G}(\theta) \in \mathbb{R}^{d_{\mathrm{in}} \times d_{\mathrm{out}}}$ – обучаемая матрица параметров, $\mathbf{F}' = (\mathbf{f}_1, \dots, \mathbf{f}_{d_{\mathrm{out}}})$ – матрица выходного сигнала.

Одним из недостатков такого подхода является то, что непараметрический фильтр не обладает свойством локализации: поскольку элементы диагонали матрицы $\mathbf{G}$ из формулы (8) соответствуют коэффициентам разложения сигнала в ряд Фурье, а значит, и определенным частотам λ_i, то можно ввести явную

зависимость от различных собственных значений в конструкцию оптимизируемых параметров. Например, можно оптимизировать коэффициенты в полиноме r-ой степени от матрицы частот Λ:

$$\mathbf{G}(\theta) = \sum_{j=0}^{r-1} \theta_j \Lambda^j, \tag{10}$$

или можно построить параметр с помощью полиномов Чебышева [2]

$$T_j(x) = 2xT_{j-1}(x) - T_{j-2}(x), \tag{11}$$
$$T_0(x) = 1, \tag{12}$$
$$T_1(x) = x, \tag{13}$$

и отнормированной матрицы частот $\hat{\Lambda} = 2\lambda_N^{-1}\Lambda - \mathbf{I}$:

$$\mathbf{G}(\theta) = \sum_{j=0}^{r-1} \theta_j \mathbf{U} T_j(\hat{\Lambda}) \mathbf{U}^\top. \tag{14}$$

Спектральная свертка обладает красивой математической базой, однако в данном методе существует один большой практический недостаток. Поскольку Лапласиан графа напрямую связан с топологией графа, спектральные сверточные сети не могут быть применены к различным графам, так как у каждого графа будет свой Лапласиан. Это обстоятельство резко сокращает круг задач, в которых спектральная свертка может найти себе применение.

3. Пространственный подход

В основе пространственного подхода лежит идея о том, что информация о вершине графа содержится не только в признаках самой вершины, но и в признаках соседей этой вершины. По аналогии с тем, как в классических 2D-свертках небольшой фильтр "сканирует" пиксели изображения, в случае с графами было предложено "сканировать" каждый узел вместе с его соседями.

Примером такого механизма является сверточный слой из работы Neural Networks For Graphs [6]:

$$\mathbf{H}' = f\left(\mathbf{X}\mathbf{\Theta} + \mathbf{W}\mathbf{H}\mathbf{\Xi}\right), \tag{15}$$

где $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_N)$ – матрица входных признаков вершин, $\mathbf{h}_i \in \mathbb{R}^{d_{\text{in}}}$, $\mathbf{H}' = (\mathbf{h}'_1, \ldots, \mathbf{h}'_N)$ – матрица выходных признаков вершин, $\mathbf{h}'_i \in \mathbb{R}^{d_{\text{out}}}$, $\mathbf{X}$ – матрица изначальных признаков вершин, $\mathbf{\Theta}$ и $\mathbf{\Xi}$ – матрицы оптимизируемых параметров, f – функция активации.

С развитием механизма пространственной свертки появилось понятие "Message-Passing Network" [4]. Поскольку для каждой вершины один слой захватывает ее соседей, благодаря нескольким сверточным слоям информация перемещается между несмежными вершинами графа. В общем виде формула для Message-Passing-слоя выглядит следующим образом:

$$\mathbf{h}'_i = U\left(\mathbf{h}_i, \sum_{j \in \mathcal{N}_i} M\left(\mathbf{h}_i, \mathbf{h}_j, \mathbf{W}_{ij}\right)\right), \tag{16}$$

где U и M – функции с оптимизируемыми параметрами.

Существует множество вариантов пространственных сверточных слоев, некоторые из них уже реализованы в фреймворке PyTorch Geometric [3]. В целом пространственный подход пользуется большой популярностью засчет своей простоты и универсальности.

4. Эмбеддинги

Наконец, рассмотрим механизм построения эмбеддингов узлов графа $\mathcal{G}$. Определим функции энкодера и декодера.

$$\text{ENC} : \mathcal{V} \to \mathbb{R}^d, \tag{17}$$
$$\text{DEC} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+. \tag{18}$$

Энкодер будет переводить узел графа в d-мерный вещественный вектор. Попарный декодер, получая на вход эмбеддинги двух узлов, восстанавливает число – меру близости этих узлов в графе $\mathcal{G}$:

$$\text{DEC}\left(\text{ENC}(i), \text{ENC}(j)\right) = \text{DEC}\left(\mathbf{z}_i, \mathbf{z}_j\right) \approx s_{\mathcal{G}}(i, j), \tag{19}$$

где $s_{\mathcal{G}}(i,j)$ может быть, например, величиной кратчайшего пути между вершинами i и j в графе $\mathcal{G}$, или вероятностью того, что в процессе случайного блуждания фиксированной длины со стартом в вершине i вершина j будет посещена. Функция потерь в процессе обучения выглядит следующим образом:

$$\mathcal{L} = \sum_{(i,j)\in\mathcal{E}} l\left(\mathrm{DEC}\left(\mathbf{z}_i, \mathbf{z}_j\right), s_{\mathcal{G}}(i,j)\right). \tag{20}$$

Энкодер может быть любой моделью, на вход ему обычно подается матрица с one-hot-представлением узлов. От выбора декодера часто зависит выбор функции потери. Например, в качестве декодера можно взять расстояние между векторами, а в качестве функции потери – произведение:

$$\mathrm{DEC}(\mathbf{z}_i, \mathbf{z}_j) = \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \tag{21}$$

$$\mathcal{L} = \sum_{(i,j)\in\mathcal{E}} \mathrm{DEC}(\mathbf{z}_i, \mathbf{z}_j) s_{\mathcal{G}}(i,j). \tag{22}$$

Другой вариант – это скалярное произведение и сумма квадратов разности:

$$\mathrm{DEC}(\mathbf{z}_i, \mathbf{z}_j) = \mathbf{z}_i^\top \mathbf{z}_j \tag{23}$$

$$\mathcal{L} = \sum_{(i,j)\in\mathcal{E}} \left(\mathrm{DEC}(\mathbf{z}_i, \mathbf{z}_j) - s_{\mathcal{G}}(i,j)\right)^2. \tag{24}$$

Как и в случае со спектральными свертками, главным недостатком данного класса методов является то, что для одной модели топология графов, которые она использует, должна быть одной и той же.

Список литературы

[1] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[2] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[3] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[4] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[5] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[6] Alessio Micheli. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.

[7] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.