ПОЛУКОНТРОЛИРУЕМАЯ КЛАССИФИКАЦИЯ С ГРАФОВЫЕ СВЕРТОЧНЫЕ СЕТИ

Томас Н. Кипф
Амстердамский университет
T.N.Kipf@uva.nl

Макс Веллинг
Амстердамский университет
Канадский институт перспективных исследований (CIFAR)
M.Welling@uva.nl

arXiv:1609.02907v4 [cs. LG] 22 фев 2017

АБСТРАКТНЫЙ

Мы представляем масштабируемый подход к полуконтролируемому обучению на графово-структурированных данных, основанный на эффективном варианте сверточных нейронных сетей, работающих непосредственно с графами. Мы обосновываем выбор нашей сверточной архитектуры с помощью локализованной аппроксимации спектральных графов первого порядка. Наша модель линейно масштабируется по количеству ребер графа и изучает представления скрытых слоев, которые кодируют как локальную структуру графа, так и особенности узлов. В ряде экспериментов на сетях цитирования и на массиве данных графа знаний мы продемонстрировали, что наш подход значительно превосходит родственные методы.

A ЗНАКОМСТВО

Мы рассматриваем проблему классификации узлов (например, документов) в графе (например, в сети цитирования), где метки доступны только для небольшого подмножества узлов. Эту задачу можно сформулировать как полуконтролируемое обучение на основе графов, где информация о метках сглаживается по графу с помощью некоторой формы явной регуляризации на основе графа (Zhu et al., 2003; Zhou et al., 2004; Belkin et al., 2006; Weston et al., 2012), например, используя член регуляризации графа Лапласа в функции потерь:

$$\mathrm{L} = \mathrm{L0} + \lambda Lreg\,, \qquad \text{с} \qquad Лрег = \sum_{И,Дж} \mathrm{XAijkf}(Кси) - f(Хдж)\mathrm{k2} = f(X)^{>\Delta\mathrm{f}}(X). \tag{1}$$

Здесь L0 обозначает контролируемые потери по сравнению с размеченной частью графика, $f(\cdot)$ может быть дифференцируемой функцией, подобной нейронной сети, λ — весовой коэффициент, а X — матрица векторов признаков узла Xi. $\Delta = D - A$ обозначает ненормированный граф Лапласа неориентированного графа $\mathrm{G} = (V,E)$ с N узлами $vi \in \mathrm{V}$, ребрами $(vi,vj) \in \mathrm{E}$, матрицей смежности $A \in \mathrm{RN{\times}N}$ (двоичной или взвешенной) и матрицей степеней $Dii = \sum_{\mathrm{Pj}} Aij$. Формулировка уравнения 1 основана на предположении, что соединенные узлы в графе, вероятно, будут иметь одну и ту же метку. Это предположение, однако, может ограничить возможности моделирования, поскольку ребра графа не обязательно должны кодировать сходство узлов, но могут содержать дополнительную информацию.

В этой работе мы кодируем структуру графа напрямую с помощью нейросетевой модели $f(X,A)$ и обучаем на контролируемой цели L0 для всех узлов с метками, тем самым избегая явной регуляризации на основе графа в функции потерь. Кондиционирование $f(\cdot)$ на матрице смежности графа позволит модели распределять градиентную информацию от

контролируемой потери L0 и даст возможность обучаться представлениям узлов как с метками, так и без них.

Наш вклад двоякий. Во-первых, мы вводим простое и хорошо работающее правило послойного распространения для моделей нейронных сетей, которые работают непосредственно с графами, и показываем, как его можно мотивировать из приближения первого порядка спектральных извилин графов (Hammond et al., 2011). Во-вторых, мы демонстрируем, как эта форма нейросетевой модели на основе графов может быть использована для быстрой и масштабируемой полуконтролируемой классификации узлов в графе. Эксперименты на ряде наборов данных показывают, что наша модель выгодно отличается как по точности классификации, так и по эффективности (измеряемой по настенным часам) по сравнению с современными методами полуконтролируемого обучения.

Б БЫСТРЫЕ ПРИБЛИЖЕННЫЕ СВЕРТКИ НА ГРАФИКАХ

В этом разделе мы приводим теоретическое обоснование для конкретной модели нейронной сети на основе графов $f(X,A)$, которую мы будем использовать в оставшейся части этой статьи. Мы рассмотрим многослойную графовую сверточную сеть (GCN) со следующим послойным правилом распространения:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right). \tag{2}$$

Здесь $\tilde{A} = A + IN$ — это матрица смежности неориентированного графа G с добавленными самосвязями. IN — тождественная матрица, $\tilde{D}ii = \sum_j \tilde{A}ij$, а $W^{(l)}$ — специфичная для слоя обучаемая весовая матрица. $\sigma(\cdot)$ обозначает функцию активации, такую как $\mathrm{ReLU}(\cdot) = \max(0,\cdot)$. $H^{(l)} \in$ RN×D — матрица активаций в *l-м* слое; $H^{(0)} = X$. Далее мы показываем, что форма этого правила распространения может быть мотивирована[1] с помощью приближения первого порядка локализованных спектральных фильтров на графах (Hammond et al., 2011; Defferrard et al., 2016).

Б.А СВЕРТКИ СПЕКТРАЛЬНЫХ ГРАФОВ

Мы рассматриваем спектральные свертки на графиках, определяемых как умножение сигнала x ? RN (скаляр для каждого узла) с фильтром $g\theta$ = diag(θ), параметризованным как $\theta \in$ RN в области Фурье, т.е.:

$$g\theta \ ? \ x = Ug\theta U^{\top}x, \tag{3}$$

где U — матрица собственных векторов нормализованного графа Лапласа $L = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = U\Lambda U^{\top}$, где диагональная матрица его собственных значений Λ и $U^{\top}x$ является графовым преобразованием Фурье для x. Мы можем понимать $g\theta$ как функцию собственных значений L, т.е. $g\theta(\Lambda)$. Вычисление уравнения 3 требует больших вычислительных ресурсов, так как умножение на матрицу собственного вектора U равно $O(N^2)$. Кроме того, вычисление собственного разложения L в первую очередь может быть непомерно дорогим для больших графов. Чтобы обойти эту проблему, в работе Hammond et al. (2011) было высказано предположение, что $g\theta(\Lambda)$ может быть хорошо аппроксимировано усеченным разложением в терминах полиномов Чебышева $Tk(x)$ до *K-го* порядка:

[1] Мы предлагаем альтернативную интерпретацию этого правила распространения, основанную на алгоритме Вейсфейлера-Лемана (Weisfeiler & Lehmann, 1968) в Приложении А.

$$g_{\theta'}(\Lambda) \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda}) \quad , \tag{4}$$

с перемасштабированным $\tilde{\Lambda} = \frac{2}{\lambda_{\max}}\Lambda - I_N$. $\lambda_{\max}$ значением обозначает наибольшее собственное значение L. $\theta 0 \in$ РК теперь является вектором коэффициентов Чебышева. Многочлены Чебышева рекурсивно определены как $Tk(x) = 2x\mathrm{Tk}{-}1(x) - Tk{-}2(x)$, где $T0(x) = 1$ и $T1(x) = x$. Читатель может обратиться к Hammond et al. (2011) для углубленного обсуждения этого приближения.

Возвращаясь к нашему определению свертки сигнала x с фильтром $g\theta 0$, мы получаем:

$$g_{\theta'} \star x \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{L}) x \,, \tag{5}$$

с $\tilde{L} = \frac{2}{\lambda_{\max}} L - I_N$; это можно легко проверить, заметив, что $(U\Lambda U>)^k = U\Lambda kU>$. Обратите внимание, что это выражение теперь K-локализовано, так как в лапласиане оно является многочленом *K-го* порядка, т.е. зависит только от узлов, которые находятся максимум *на K* шагах от центрального узла (окрестности *K-го* порядка). Сложность вычисления уравнения 5 равна $\mathrm{O}(|\mathrm{E}|)$, т.е. линейный по количеству ребер. Defferrard et al. (2016) используют эту K-локализованную свертку для определения сверточной нейронной сети на графах.

Б.Б ПОСЛОЙНАЯ ЛИНЕЙНАЯ МОДЕЛЬ

Таким образом, модель нейронной сети, основанная на свертках графов, может быть построена путем наложения нескольких сверточных слоев в виде уравнения 5, за каждым слоем которых следует точечная нелинейность. Теперь представьте, что мы ограничили послойную операцию свертки *до* $K = 1$ (см. уравнение 5), т.е. функции, которая является линейной w.r.t. L и, следовательно, линейная функция на графе спектра Лапласа.

Таким образом, мы все еще можем восстановить богатый класс сверточных фильтрующих функций, сложив несколько таких слоев, но мы не ограничены явной параметризацией, задаваемой, например, полиномами Чебышева. Мы интуитивно ожидаем, что такая модель может облегчить проблему переобучения на локальных структурах окрестностей для графов с очень широкими распределениями степеней узлов, таких как социальные сети, сети цитирования, графы знаний и многие другие реальные наборы данных графов. Кроме того, для фиксированного вычислительного бюджета эта послойная линейная формулировка позволяет нам строить более глубокие модели, что, как известно, улучшает возможности моделирования в ряде областей (He et al., 2016).

В этой линейной формулировке GCN мы дополнительно аппроксимируем $\lambda max \approx 2$, поскольку мы можем ожидать, что параметры нейронной сети будут адаптироваться к этому изменению масштаба во время обучения. В этих приближениях уравнение 5 упрощается до:

$$g_{\theta'} \star x \approx \theta'_0 x + \theta'_1 \left(L - I_N\right) x = \theta'_0 x - \theta'_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x \,, \tag{6}$$

с двумя свободными параметрами θ'_0 и θ'_1. Параметры фильтра могут быть общими для всего графика. Последовательное применение фильтров этой формы затем эффективно свертывает *окрестности узла k-го порядка, где* k *— количество последовательных операций фильтрации или сверточных слоев в модели нейронной сети.*

На практике может быть полезно дополнительно ограничить количество параметров, чтобы решить проблему переобучения и свести к минимуму количество операций (таких как умножение матриц) на слой. Это оставляет нас со следующим выражением:

$$g_\theta \star x \approx \theta \left(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right) x \,, \tag{7}$$

с одним параметром $\theta = \theta_0' = -\theta_1'$. Обратите внимание, что $I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ теперь имеет собственные значения в диапазоне [0,2]. Таким образом, повторное применение этого оператора может привести к численным неустойчивостям и взрывающимся/исчезающим градиентам при использовании в модели глубокой нейронной сети. Чтобы решить эту проблему, мы вводим следующий *трюк с перенормировкой* $I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \rightarrow \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$: *с* $\tilde{A} = A + IN$ и $\tilde{D}ii = \sum_j \tilde{A}ij$.

Мы можем обобщить это определение на сигнал $X \in \mathbb{R}^{N\times C}$ с входными каналами C (т.е. вектором признаков C-размерности для каждого узла) и *фильтрами* F или картами признаков следующим образом:

$$Z = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta, \tag{8}$$

где $\Theta \in \mathbb{R}^{C\times F}$ — матрица параметров фильтра, а $Z \in \mathbb{R}^{N\times F}$ — матрица сверточного сигнала. Эта операция фильтрации имеет сложность $O(|E|FC)$, так как $A\tilde{X}$ может быть эффективно реализован как произведение разреженной матрицы с плотной матрицей.

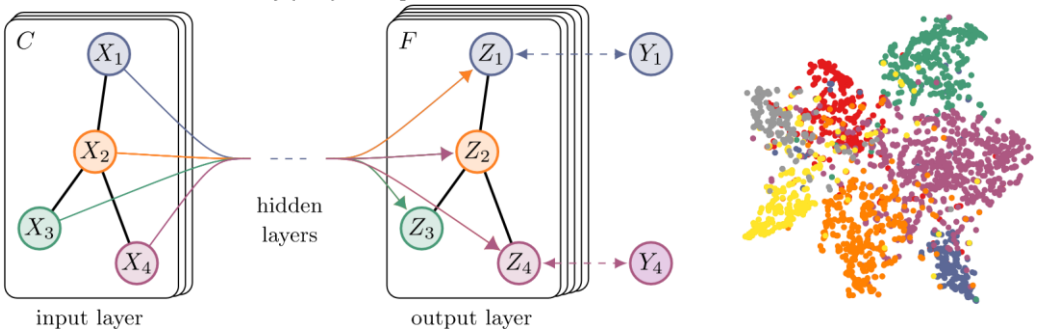
B КЛАССИФИКАЦИЯ УЗЛОВ С ПОЛУОБУЧЕНИЕМ

Введя простую, но гибкую модель $f(X,A)$ для эффективного распространения информации на графах, мы можем вернуться к проблеме классификации узлов с частичным обучением. Как было показано во введении, мы можем ослабить некоторые предположения, обычно делаемые при полуконтролируемом обучении на основе графов, обуславливая нашу модель $f(X,A)$ как данными X, так и матрицей смежности A базовой структуры графа. Мы ожидаем, что этот параметр будет особенно эффективен в сценариях, где матрица смежности содержит информацию, отсутствующую в данных X, например, ссылки на цитаты между документами в сети цитирования или отношения в графе знаний. Общая модель, представляющая собой многоуровневую ОЦС для полуконтролируемого обучения, схематически изображена на рисунке 1.

B.A ПРИМЕР

Далее мы рассмотрим двухуровневую GCN для полуконтролируемой классификации узлов на графе с симметричной матрицей смежности A (двоичной или взвешенной). Сначала мы рассчитываем $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ на этапе предварительной обработки. Наша прямая модель принимает простую форму:

$$Z = f(X,A) = \text{softmax}\Big(\hat{A}\,\text{ReLU}\Big(\hat{A}XW^{(0)}\Big)W^{(1)}\Big). \tag{9}$$



(a) Графовая сверточная сеть (b) Активации скрытого слоя

Рисунок 1: *Слева*: Схематическое изображение многослойной сверточной сети графа (GCN) для полуконтролируемого обучения с входными каналами C и картами признаков F в

выходном слое. Структура графика (ребра показаны черными линиями) является общей для слоев, метки обозначаются буквой Yi. *Справа*: t-SNE (Maaten & Hinton, 2008) визуализация активации скрытого слоя двухслойного GCN, обученного на наборе данных Cora (Sen et al., 2008) с использованием 5% меток. Цвета обозначают класс документа.

Здесь $W^{(0)} \in$ RC×H – это матрица входных и скрытых весов для скрытого слоя с *картами признаков* H.$W^{(1)} \in$ RH×F — скрытая для вывода весовая матрица. Функция активации softmax, определенная как $\mathrm{softmax}(xi) = \frac{1}{Z} \exp(xi)$ с $Z = \sum_{i} \exp(xi)$, применяется построчно. Для полуконтролируемой многоклассовой классификации мы затем оцениваем ошибку перекрестной энтропии по всем помеченным примерам:

$$L = -\sum_{l \in YL} \sum_{f=1}^{F} Ylf \ln Zlf, \tag{10}$$

где YL — набор индексов узлов, имеющих метки.

Веса нейронной сети $W^{(0)}$ и $W^{(1)}$ обучаются с помощью градиентного спуска. В этой работе мы выполняем пакетный градиентный спуск с использованием полного набора данных для каждой обучающей итерации, что является жизнеспособным вариантом, если наборы данных помещаются в памяти. При использовании разреженного представления для A требования к памяти равны $O(|E|)$, т.е. линейный по количеству ребер. Стохастичность в тренировочный процесс внедряется через отсев (Srivastava et al., 2014). Мы оставляем эффективные по памяти расширения с мини-пакетным стохастическим градиентным спуском для будущей работы.

В.Б РЕАЛИЗАЦИЯ

На практике мы используем TensorFlow (Abadi et al., 2015) для эффективной реализации уравнения 9 на основе GPU [2] с использованием умножения матриц с разреженной плотностью. Вычислительная сложность вычисления уравнения 9 равна тогда $O(|E|CHF)$, т.е. линейный по количеству ребер графа.

Г СВЯЗАННАЯ РАБОТА

Наша модель черпает вдохновение как из области полуконтролируемого обучения на основе графов, так и из недавних работ по нейронным сетям, работающим на графах. Далее мы дадим краткий обзор соответствующей работы в обеих областях.

Г.А ПОЛУКОНТРОЛИРУЕМОЕ ОБУЧЕНИЕ НА ОСНОВЕ ГРАФОВ

В последние годы было предложено большое количество подходов к полуконтролируемому обучению с использованием графовых представлений, большинство из которых делятся на две большие категории: методы, использующие ту или иную форму явной регуляризации графов Лапласа, и подходы, основанные на встраивании графов. Яркими примерами регуляризации графов Лапласа являются распространение меток (Zhu et al., 2003), многообразная регуляризация (Belkin et al., 2006) и глубокое полуконтролируемое встраивание (Weston et al., 2012).

В последнее время внимание сместилось на модели, которые обучаются встраиванию графов с помощью методов, вдохновленных моделью пропуска грамм (Mikolov et al., 2013). DeepWalk (Perozzi et al., 2014) изучает вложения путем прогнозирования локального

[2] Код для воспроизведения наших экспериментов доступен по адресу https://github.com/tkipf/gcn.

соседства узлов, отобранных из случайных блужданий на графе. LINE (Tang et al., 2015) и node2vec (Grover & Leskovec, 2016) расширяют возможности DeepWalk с помощью более сложных схем случайного блуждания или поиска в ширину. Однако для всех этих методов требуется многоступенчатый конвейер, включающий генерацию случайного блуждания и полуконтролируемое обучение, где каждый шаг должен быть оптимизирован отдельно. Планетоид (Yang et al., 2016) смягчает эту проблему, вводя информацию о метках в процесс обучения встраиванию.

Г.Б НЕЙРОННЫЕ СЕТИ НА ГРАФАХ

Нейронные сети, работающие на графах, ранее были представлены в Gori et al. (2005); Scarselli et al. (2009) в качестве формы рекуррентной нейронной сети. Их структура требует многократного применения карт сокращения в качестве функций распространения до тех пор, пока представления узлов не достигнут стабильной фиксированной точки. Это ограничение было позже смягчено в работе Li et al. (2016) путем внедрения современных практик рекуррентного обучения нейронных сетей в исходную структуру графовых нейронных сетей. Duvenaud et al. (2015) ввели правило сверточного распространения для графов и методы классификации на уровне графов. Их подход требует изучения весовых матриц для конкретных степеней узлов, которые не масштабируются до больших графов с широкими распределениями степеней узлов. Вместо этого наша модель использует одну весовую матрицу для каждого слоя и имеет дело с различными степенями узлов посредством соответствующей нормализации матрицы смежности (см. раздел 3.1).

Связанный с этим подход к классификации узлов с помощью нейронной сети на основе графов был недавно представлен в работе Atwood & Towsley (2016). Они сообщают о сложности $\mathrm{O}(N2)$, ограничивая диапазон возможных применений. В другой, но связанной модели Niepert et al. (2016) локально преобразуют графы в последовательности, которые подаются в обычную одномерную сверточную нейронную сеть, которая требует определения порядка узлов на этапе предварительной обработки.

Наш метод основан на сверточных нейронных сетях спектрального графа, введенных в работе Bruna et al. (2014) и позже расширенных Defferrard et al. (2016) с быстрыми локализованными свертками. В отличие от этих работ, здесь рассматривается задача классификации трансдуктивных узлов в сетях значительно большего масштаба. Мы показываем, что в этих условиях в исходные структуры Bruna et al. (2014) и Defferrard et al. (2016) может быть внесен ряд упрощений (см. раздел 2.2), которые улучшают масштабируемость и производительность классификации в крупномасштабных сетях.

Д ЭКСПЕРИМЕНТЫ

Мы протестировали нашу модель в ряде экспериментов: полуконтролируемая классификация документов в сетях цитирования, полуконтролируемая классификация сущностей в двудольном графе, извлеченном из графа знаний, оценка различных моделей распространения графов и анализ случайных графов во время выполнения.

Д.А НАБОРОВ ДАННЫХ

Мы внимательно следим за экспериментальной установкой в Yang et al. (2016). Статистика набора данных сведена в таблицу 1. В наборах данных сети цитирования — Citeseer, Cora и Pubmed (Sen et al., 2008) — узлы — это документы, а ребра — ссылки на цитаты. Скорость меток обозначает количество помеченных узлов, используемых для обучения, деленное на общее количество узлов в каждом наборе данных. NELL (Carlson et al., 2010; Yang et al., 2016) представляет собой двудольный набор данных графов, извлеченный из графа знаний с 55 864 узлами отношений и 9 891 узлом сущностей.

Таблица 1: Статистика набора данных, представленная в Yang et al. (2016).

Набор данных	Тип	Узлов	Края	Классы	Функции	Норма маркировки
Citeseer	Сеть цитирования	3,327	4,732	6	3,703	0.036
Кора	Сеть цитирования	2,708	5,429	7	1,433	0.052
Пабмед	Сеть цитирования	19,717	44,338	3	500	0.003
НЕЛЛ	Граф знаний	65,755	266,144	210	5,414	0.001

Мы рассматриваем три набора данных сети цитирования: Citeseer, Cora и Pubmed (Sen et al., 2008). Наборы данных содержат разреженные векторы признаков для каждого документа и список ссылок на цитаты между документами. Мы рассматриваем ссылки цитирования как (ненаправленные) ребра и строим двоичную, симметричную матрицу смежности A. У каждого документа есть метка класса. Для обучения мы используем только 20 меток на класс, но все векторы признаков.

NELL NELL представляет собой набор данных, извлеченный из графа знаний, представленного в (Carlson et al., 2010). Граф знаний — это набор сущностей, связанных направленными, помеченными ребрами (отношениями). Мы следуем схеме предварительной обработки, описанной в Yang et al. (2016). Мы присваиваем отдельные узлы отношения $r1$ и $r2$ для каждой пары сущностей ($e1,r,e2$) как ($e1,r1$) и ($e2,r2$). Узлы сущностей описываются разреженными векторами признаков. Мы расширяем количество признаков в NELL, назначая уникальное одногорячее представление для каждого узла отношения, что фактически приводит к 61 278 dim вектору разреженных признаков на узел. В задаче с частичным обучением рассматривается крайний случай, когда на класс в обучающем наборе используется только один помеченный пример. Мы строим двоичную, симметричную матрицу смежности из этого графа, устанавливая записи $Aij = 1$, если между узлами i и j присутствует одно или несколько ребер .

Случайные графы Мы моделируем наборы данных случайных графов различных размеров для экспериментов, в которых измеряем время обучения в каждую эпоху. Для набора данных с N узлами мы создаем случайный граф, равномерно распределяя $2\mathrm{N}$ ребер случайным образом. Мы принимаем матрицу идентичности IN в качестве матрицы входных признаков X, тем самым неявно применяя безликий подход, при котором модель информируется только об идентичности каждого узла, заданной уникальным вектором one-hot. Добавляем фиктивные метки $Yi = 1$ для каждого узла.

Д.Б ЭКСПЕРИМЕНТАЛЬНАЯ УСТАНОВКА

Если не указано иное, мы обучаем двухуровневую GCN, как описано в разделе 3.1, и оцениваем точность прогнозирования на тестовом наборе из 1000 размеченных примеров. В Приложении В мы проводим дополнительные эксперименты с использованием более глубоких моделей с 10 слоями. Мы выбираем те же разделения наборов данных, что и в Yang et al. (2016), с дополнительным валидационным набором из 500 размеченных примеров для оптимизации гиперпараметров (частота отсева для всех слоев, коэффициент регуляризации

L2 для первого слоя GCN и количество скрытых единиц). Мы не используем метки проверочных наборов для обучения.

Для наборов данных сети цитирования мы оптимизируем гиперпараметры только на Cora и используем тот же набор параметров для Citeseer и Pubmed. Мы обучаем все модели максимум за 200 эпох (обучающих итераций) с использованием Adam (Kingma & Ba, 2015) с коэффициентом обучения, равным 0.01 и ранняя остановка с размером окна 10, т.е. останавливаем обучение, если потери валидации не уменьшаются в течение 10 последовательных эпох. Мы инициализируем веса с помощью инициализации, описанной в Glorot & Bengio (2010) и, соответственно, нормализуем векторы входных признаков. В наборах данных случайных графов мы используем скрытый размер слоя в 32 единицы и опускаем регуляризацию (т.е. ни выпадение, ни регуляризацию L2).

Д.В БАЗОВЫХ ЛИНИЙ

Мы сравниваем с теми же базовыми методами, что и в работе Yang et al. (2016), т.е. с распространением меток (LP) (Zhu et al., 2003), полуконтролируемым встраиванием (SemiEmb) (Weston et al., 2012), многообразной регуляризацией (ManiReg) (Belkin et al., 2006) и встраиванием графов на основе пропуска (DeepWalk) (Perozzi et al., 2014). Мы опускаем TSVM (Joachims, 1999), так как он не масштабируется до большого количества классов в одном из наших наборов данных.

Далее мы сравниваем с алгоритмом итерационной классификации (ICA), предложенным в работе Lu & Getoor (2003) в сочетании с двумя классификаторами логистической регрессии, один только для локальных узловых признаков, а другой для реляционной классификации с использованием локальных признаков и оператора агрегирования, как описано в Sen et al. (2008). Сначала мы обучаем локальный классификатор с использованием всех помеченных узлов обучающего набора и используем его для начальной загрузки меток классов неразмеченных узлов для обучения реляционного классификатора. Мы запускаем итерационную классификацию (реляционный классификатор) со случайным упорядочиванием узлов за 10 итераций на всех неразмеченных узлах (с использованием локального классификатора). Параметр регуляризации L2 и оператор агрегирования (*count* vs. *prop*, см. Sen et al. (2008)) выбираются на основе производительности валидационного набора для каждого набора данных отдельно.

Наконец, мы сравниваем с Planetoid (Yang et al., 2016), где мы всегда выбираем их наиболее эффективный вариант модели (трансдуктивный против индуктивного) в качестве базового уровня.

Е РЕЗУЛЬТАТЫ

Е.А КЛАССИФИКАЦИЯ УЗЛОВ С ПОЛУОБУЧЕНИЕМ

Результаты обобщены в таблице 2. Сообщаемые числа обозначают точность классификации в процентах. Для ICA мы сообщаем о средней точности в 100 прогонов со случайным упорядочиванием узлов. Результаты для всех других базовых методов взяты из статьи Planetoid (Yang et al., 2016). Planetoid* обозначает лучшую модель для соответствующего набора данных из вариантов, представленных в их статье.

Таблица 2: Сводка результатов с точки зрения точности классификации (в процентах).

Метод	Citeseer	Кора	Пабмед	НЕЛЛ
МаниРег [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
Пластинка [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1

Планетоид* [29]	64.7 (26c)	75.7 (13c)	77.2 (25c)	61.9 (185c)
GCN (этот документ)	**70.3** (7 с)	**81.5** (4с)	**79.0** (38с)	**66.0** (48с)
GCN (рэнд. шпагат)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

Далее мы указываем время обучения настенных часов в секундах до сходимости (в скобках) для нашего метода (включая оценку ошибки валидации) и для Планетоида. Для последнего мы использовали реализацию, предоставленную авторами [3] и обученную на том же оборудовании (с графическим процессором), что и наша модель GCN. Мы обучили и протестировали нашу модель на тех же разбиениях наборов данных, что и в Yang et al. (2016), и сообщили о средней точности в 100 прогонов со случайной инициализацией весов. Для Citeseer, Cora и Pubmed мы использовали следующие наборы гиперпараметров: 0,5 (коэффициент отсева), $5 \cdot 10{-}4$ (L2 регуляризация) и 16 (количество скрытых единиц), а для NELL: 0,1 (коэффициент отсева), $1{\cdot}10{-}5$ (L2 регуляризация) и 64 (количество скрытых единиц).

Кроме того, мы сообщаем о производительности нашей модели на 10 случайно выбранных разбивках набора данных того же размера, что и в Yang et al. (2016), обозначенных GCN (rand. splits). Здесь мы сообщаем среднюю и стандартную ошибку точности предсказания на тестовом наборе, разбитом в процентах.

Е.Б ОЦЕНКА МОДЕЛИ РАСПРОСТРАНЕНИЯ РАДИОВОЛН

Мы сравниваем различные варианты предложенной нами модели распространения по слоям на наборах данных сети цитирования. Мы следуем экспериментальной схеме, описанной в предыдущем разделе. Результаты обобщены в таблице 3. Модель распространения нашей исходной модели GCN обозначена трюком *с перенормировкой* (выделен жирным шрифтом). Во всех остальных случаях модель распространения обоих слоев нейронной сети заменяется моделью, указанной в разделе *Модель распространения*. Сообщаемые числа обозначают среднюю точность классификации для 100 повторных прогонов со случайной инициализацией матрицы весов. В случае множественных переменных Θi на слой мы применяем L2-регуляризацию ко всем весовым матрицам первого слоя.

Таблица 3: Сравнение моделей распространения.

Описание	Модель распространения	Citeseer	Кора	Пабмед
Фильтр Чебышева (уравнение 5)				
$K = 3$	$\sum_{k=0}^{K} T_k(\tilde{L})X\Theta_k$	69.8	79.5	74.4
$K = 2$		69.6	81.2	73.8
Single parameter (Eq. 7)	$(I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})X\Theta$	69.3	79.2	77.4
Renormalization trick				
Multi-layer perceptron	Θ	46.5	55.1	71.4
(Eq. 8)	$\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta$	**70.3**	**81.5**	**79.0**
1st	$X\Theta_0 + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}X\Theta_1$	68.3	80.0	77.5

заказать модель (уравнение 6)

[3] https://github.com/kimiyoung/planetoid

первый-только срок заказа $D{-}12AD{-}12X\Theta$ 68.7 80.5 77.8 X

Е.В ВРЕМЯ ОБУЧЕНИЯ НА ЭПОХУ

Здесь мы сообщаем результаты для среднего времени обучения за эпоху (прямой проход, расчет кросс-энтропии, обратный проход) за 100 эпох на смоделированных случайных графиках, измеренных в секундах настенного времени. В разделе 5.1 приведено подробное описание набора случайных графов, используемого в этих экспериментах. Мы сравниваем результаты на GPU и на реализации только на CPU[4] в TensorFlow (Abadi et al., 2015). На рисунке 2 обобщены результаты.

Ж ОБСУЖДЕНИЕ

Ж.А ПОЛУКОНТРОЛИРУЕМАЯ МОДЕЛЬ

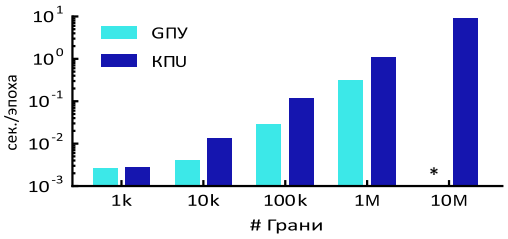


Рисунок 2: Время настенных часов на эпоху для случайных графиков. (*) указывает на ошибку нехватки памяти.

В экспериментах, показанных здесь, наш метод полуконтролируемой классификации узлов значительно превосходит современные родственные методы. Методы, основанные на графовой регуляризации Лапласа (Zhu et al., 2003; Belkin et al., 2006; Weston et al., 2012), скорее всего, ограничены из-за их предположения, что ребра кодируют простое сходство узлов. Методы, основанные на скип-граммах, с другой стороны, ограничены тем фактом, что они основаны на многоступенчатом конвейере, который трудно оптимизировать. Предложенная нами модель может преодолеть оба ограничения, при этом выгодно отличаясь с точки зрения эффективности (измеряемой по настенному времени) от родственных методов. Распространение информации об объектах из соседних узлов в каждом слое повышает производительность классификации по сравнению с такими методами, как ICA (Lu & Getoor, 2003), где агрегируется только информация о метках.

Кроме того, мы продемонстрировали, что предложенная ренормированная модель распространения (уравнение 8) обеспечивает как повышенную эффективность (меньшее количество параметров и операций, таких как умножение или сложение), так и лучшую прогностическую производительность на ряде наборов данных по сравнению с наивной моделью 1-го порядка (уравнение 6) или графовыми сверточными моделями более высокого порядка, использующими полиномы Чебышева (уравнение 5).

Ж.Б ОГРАНИЧЕНИЯ И БУДУЩАЯ РАБОТА

В этой статье мы опишем несколько ограничений нашей текущей модели и наметим, как они могут быть преодолены в будущей работе.

Требования к памяти В текущей конфигурации с полным пакетным градиентным спуском потребность в памяти линейно увеличивается по размеру набора данных. Мы показали, что для больших графиков, которые не помещаются в памяти графического процессора, обучение на процессоре все еще может быть жизнеспособным вариантом. Мини-пакетный стохастический градиентный спуск может решить эту проблему. Процедура генерации мини-пакетов, однако, должна учитывать количество слоев в модели GCN, так как *окрестность Kth-порядка для GCN с K* слоями должна быть сохранена в памяти для точной процедуры. Для очень больших и плотно связных наборов графовых данных могут потребоваться дополнительные аппроксимации.

[4] Используемое аппаратное обеспечение: 16-ядерный процессор Intel R Xeon R E5-2640 v3 @ 2.60GHz, GeForce R GTX TITAN X

Направленные ребра и функции ребер В настоящее время наша платформа естественным образом не поддерживает функции ребер и ограничена неориентированными графами (взвешенными или невзвешенными). Однако результаты работы с NELL показывают, что можно обрабатывать как направленные ребра, так и ребра, представляя исходный ориентированный граф в виде неориентированного двудольного графа с дополнительными узлами, представляющими ребра в исходном графе (подробнее см. раздел 5.1).

С помощью аппроксимаций, представленных в разделе 2, мы неявно предполагаем локальность (зависимость от *окрестности Kth-порядка для GCN с* K слоями) и одинаковую важность самосвязей и ребер к соседним узлам. Однако для некоторых наборов данных может быть полезно ввести в *определение* $\tilde{A}$ *компромиссный параметр* λ:

$$\tilde{A} = A + \lambda B\,. \tag{11}$$

Этот параметр теперь играет ту же роль, что и параметр компромисса между контролируемыми и неконтролируемыми убытками в типичных условиях с полуконтролируемым наблюдением (см. уравнение 1). Здесь, однако, этому можно научиться с помощью градиентного спуска.

3 ЗАКЛЮЧЕНИЕ

Мы представили новый подход к полуконтролируемой классификации на графовых структурированных данных. В нашей модели GCN используется эффективное правило послойного распространения, основанное на аппроксимации спектральных сверток первого порядка на графиках. Эксперименты на ряде наборов сетевых данных показывают, что предложенная модель GCN способна кодировать как структуру графа, так и особенности узлов таким образом, что это полезно для полуконтролируемой классификации. В этих условиях наша модель значительно превосходит несколько недавно предложенных методов, оставаясь при этом вычислительно эффективной.

ПОДТВЕРЖДЕНИЯ

Мы хотели бы поблагодарить Кристоса Луизоса, Тако Коэна, Джоан Бруна, Жилин Янга, Дэйва Хермана, Прамода Синха и Абдул-Сабура Шейха за полезные дискуссии. Это исследование было профинансировано компанией SAP.

ССЫЛКИ

Мартин Абади и др. TensorFlow: Крупномасштабное машинное обучение в гетерогенных системах, 2015.

Джеймс Этвуд и Дон Таусли. Диффузионно-сверточные нейронные сети. В *журнале «Достижения в области нейронных систем обработки информации» (NIPS),* 2016.

Михаил Белкин, Парта Нийоги и Викас Синдхвани. Многообразная регуляризация: Геометрическая структура для обучения на размеченных и непомеченных примерах. *Журнал исследований машинного обучения (JMLR),* 7(ноябрь):2399–2434, 2006.

Ульрик Брандес, Даниэль Деллинг, Марко Гертлер, Роберт Горке, Мартин Хофер, Зоран Николоски и Доротея Вагнер. О модульной кластеризации. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008.

Джоан Бруна, Войцех Заремба, Артур Слам и Ян Лекун. Спектральные сети и локально связанные сети на графах. В *Международной конференции по представлениям об обучении (ICLR),* 2014.

Эндрю Карлсон, Джастин Беттеридж, Брайан Кисил, Берр Сеттлз, Эстевам Р. Грушка-младший и Том М. Митчелл. На пути к архитектуре для бесконечного изучения языков. В *AAAI,* том 5, стр. 3, 2010.

Михаэль Дефферрар, Ксавье Брессон и Пьер Вандергейнст. Сверточные нейронные сети на графах с быстрой локализованной спектральной фильтрацией. В *журнале «Достижения в области нейронных систем обработки информации» (NIPS),* 2016.

Брендан Л. Дуглас. Метод Вейсфейлера-Лемана и проверка изоморфизма графов. *Препринт arXiv arXiv:1101.5211*, 2011.

Дэвид К. Дювено, Дугал Маклорен, Хорхе Ипаррагирре, Рафаэль Бомбарелл, Тимоти Хирцель, Алан Аспуру-Гузик и Райан. Адамс. Сверточные сети на графах для обучения молекулярных отпечатков. В *книге «Достижения в области нейронных систем обработки информации» (NIPS),* стр. 2224–2232, 2015.

Ксавье Глорот и Йошуа Бенжио. Понимание сложности обучения нейронных сетей с глубокой прямой связью. В *AISTATS,* том 9, стр. 249–256, 2010.

Марко Гори, Габриэле Монфардини и Франко Скарселли. Новая модель обучения в графовых областях. В *материалах Международной объединенной конференции IEEE по нейронным сетям 2005 года.*, том 2, стр. 729–734. IEEE, 2005.

Адитья Гровер и Юре Лесковец. node2vec: Масштабируемое обучение функций для сетей. *В материалах 22-й Международной конференции ACM SIGKDD по открытию знаний и интеллектуальному анализу данных*. ACM, 2016.

Дэвид К. Хаммонд, Пьер Вандергейнст и Реми Грибонваль. Вейвлеты на графах с помощью спектральной теории графов. *Прикладной и вычислительный гармонический анализ*, 30(2):129–150, 2011.

Каймин Хэ, Сянъюй Чжан, Шаоцин Жэнь и Цзянь Сунь. Глубокое остаточное обучение для распознавания изображений. На *конференции IEEE по компьютерному зрению и распознаванию образов (CVPR),* 2016 г.

Торстен Иоахимс. Трансдуктивный вывод для классификации текста с использованием метода опорных векторов. *В Международной конференции по машинному обучению (ICML),* том 99, стр. 200–209, 1999.

Дидерик. Кингма и Джимми Лей Ба. Адам: Метод стохастической оптимизации. В *Международной конференции по учебным представлениям (ICLR),* 2015.

Юйцзя Ли, Дэниел Тарлоу, Марк Брокшмидт и Ричард Земель. Нейронные сети с последовательностью графов. На *Международной конференции по учебным представлениям (ICLR),* 2016 г.

Цин Лу и Лиз Гетур. Классификация по ссылкам. В *Международной конференции по машинному обучению (ICML),* том 3, стр. 496–503, 2003.

Лоренс ван дер Маатен и Джеффри Хинтон. Визуализация данных с помощью t-sne. *Журнал исследований машинного обучения (JMLR),* 9(ноябрь): 2579–2605, 2008.

Томас Миколов, Илья Суцкевер, Кай Чен, Грег С. Коррадо и Джефф Дин. Распределенные представления слов и словосочетаний и их композиционность. В *книге «Достижения в области нейронных систем обработки информации» (NIPS),* стр. 3111–3119, 2013.

Матиас Ниперт, Мохамед Ахмед и Константин Куцков. Обучение сверточных нейронных сетей для графов. На *Международной конференции по машинному обучению (ICML),* 2016 г.

Брайан Перонцци, Рами Аль-Рфу и Стивен Скина. Deepwalk: Онлайн-обучение социальным представлениям. В *материалах 20-й международной конференции ACM SIGKDD по обнаружению знаний и интеллектуальному анализу данных*, стр. 701–710. ACM, 2014.

Франко Скарселли, Марко Гори, А Чунг Цой, Маркус Хагенбухнер и Габриэле Монфардини. Графовая нейросетевая модель. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

Притхвирадж Сен, Галилео Намата, Мустафа Билгич, Лиз Гетур, Брайан Галлигер и Тина Элиасси-Рад. Коллективная классификация в сетевых данных. *Журнал AI*, 29(3):93, 2008.

Нитиш Шривастава, Джеффри Э. Хинтон, Алекс Крижевский, Илья Суцкевер и Руслан Салахутдинов. Отсев: простой способ предотвратить переобучение нейронных сетей. *Журнал исследований в области машинного обучения (JMLR),* 15(1):1929–1958, 2014.

Цзянь Тан, Мэн Цюй, Минчжэ Ван, Мин Чжан, Цзюнь Янь и Цяочжу Мэй. Линия: Встраивание крупномасштабных информационных сетей. В *материалах 24-й Международной конференции по Всемирной паутине*, стр. 1067–1077. ACM, 2015.

Борис Вайсфейлер и А. А. Леман. Приведение графа к канонической форме и алгебра, возникающая при этой редукции. *Научно-техническая информатика*, 2(9):12–16, 1968.

Джейсон Уэстон, Фредерик Ратл, Хоссейн Мобахи и Ронан Коллоберт». Глубокое обучение с помощью полуконтролируемого встраивания. В *Нейронные сети: хитрости торговли*, стр. 639–655. Springer, 2012.

Жилин Янг, Уильям Коэн и Руслан Салахутдинов. Возвращаясь к полуконтролируемому обучению с помощью встраивания графов. На *Международной конференции по машинному обучению (ICML),* 2016 г.

Уэйн В. Закари. Модель информационного потока для конфликтов и деления в малых группах. *Журнал антропологических исследований*, стр. 452–473, 1977.

Дэнъюн Чжоу, Оливье Буске, Томас Навин Лал, Джейсон Уэстон и Бернхард Шолкопф¨. Обучение на местном и глобальном уровнях. *В журнале Advances in neural information processing systems (NIPS),* том 16, стр. 321–328, 2004.

Сяоцзинь Чжу, Зубин Гахрамани и Джон Лафферти. Полуконтролируемое обучение с использованием гауссовских полей и гармонических функций. В *Международной конференции по машинному обучению (ICML),* том 3, стр. 912–919, 2003.

A СВЯЗЬ С АЛГОРИТМОМ ВЕЙСФЕЙЛЕРА-ЛЕМАНА

Модель нейронной сети для структурированных графами данных в идеале должна быть способна обучаться представлениям узлов в графе, принимая во внимание как структуру графа, так и описание характеристик узлов. Хорошо изученная структура для уникального присвоения меток узлов по заданному графу и (опционально) дискретным начальным меткам узлов обеспечивается алгоритмом Вейсфейлера-Лемана (WL-1) с 1 димом (Weisfeiler & Lehmann, 1968):

Алгоритм 1: Алгоритм WL-1 (Weisfeiler & Lehmann, 1968)

Входные данные: Начальная раскраска нод $(h_1^{(0)}, h_2^{(0)}, ..., h_N^{(0)})$

Вывод: Окончательная раскраска нода $(h_1^{(T)}, h_2^{(T)}, ..., h_N^{(T)})$

t ← 0;

повторя

  для $v_я \in V$ де

    $h_я^{(t+1)} \leftarrow$ меша $P_{j \in H_я} h_j^{(t)}$;

  $t \leftarrow t+1$;

до *достигнута стабильная окраска нод*

Здесь $h_i^{(t)}$ обозначает раскраску (присвоение метки) узла *vi* (на итерации *t*), а Ni — набор индексов соседних узлов (независимо от того, включает ли граф самосвязи для каждого узла или нет). Хеш(·) — хеш-функция. Для углубленного математического обсуждения алгоритма WL-1 см., например, Douglas (2011).

Мы можем заменить хеш-функцию в алгоритме 1 на нейронную сеть, послойную дифференцируемую функцию с обучаемыми параметрами следующим образом:

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}_i} \frac{1}{c_{ij}} h_j^{(l)} W^{(l)}\right), \tag{12}$$

где *cij* — правильно выбранная константа нормализации для ребра (*vi,vj*). Далее, теперь мы можем $h_i^{(l)}$ принять за вектор *активаций* узла *i* в *слое l-й* нейронной сети. $W^{(l)}$ — весовая матрица для конкретного слоя, а *σ*(·) обозначает дифференцируемую, нелинейную функцию активации.

Выбрав *cij* = $\sqrt{\text{didj}}$, где *di* = |Ni| обозначает степень узла *vi*, мы восстанавливаем правило распространения нашей модели сверточной сети графа (GCN) в векторной форме (см. уравнение 2).[5]

Это, грубо говоря, позволяет нам интерпретировать нашу модель GCN как дифференцируемое и параметризованное обобщение алгоритма Вейсфейлера-Лемана на графах.

A.1 ВЛОЖЕНИЯ УЗЛОВ СО СЛУЧАЙНЫМИ ВЕСАМИ

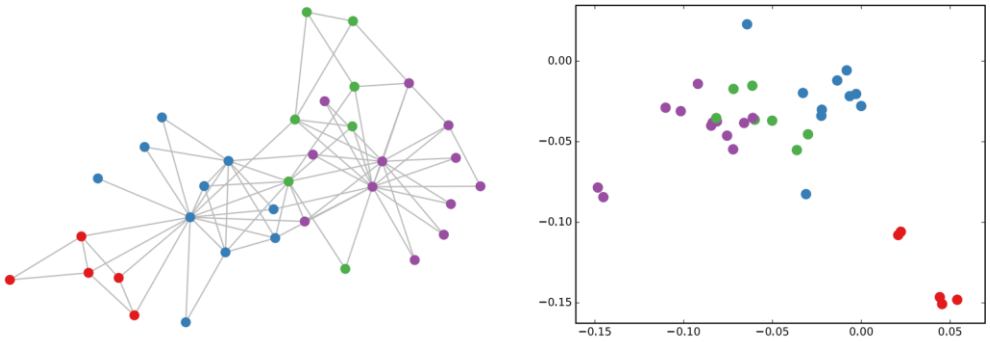
Из аналогии с алгоритмом Вейсфейлера-Лемана можно понять, что даже необученная модель GCN со случайными весами может служить мощным экстрактором признаков для узлов в графе. В качестве примера рассмотрим следующую 3-слойную модель GCN:

$$Z = \tanh\Big(\hat{A}\,\tanh\Big(\hat{A}\,\tanh\Big(\hat{A}XW^{(0)}\Big)\,W^{(1)}\Big)\,W^{(2)}\Big), \tag{13}$$

с весовыми матрицами $W^{(l)}$, инициализированными случайным образом с использованием инициализации, описанной в Glorot & Bengio (2010). $\hat{A}$, X и Z определены в соответствии с разделом 3.1.

Мы применяем эту модель к сети карат-клубов Закари (Zachary, 1977). Этот граф содержит 34 узла, соединенных 154 (ненаправленными и невзвешенными) ребрами. Каждый узел помечен одним из четырех классов, полученных с помощью кластеризации на основе модульности (Brandes et al., 2008). Иллюстрация приведена на рисунке 3a.

[5] Обратите внимание, что здесь мы неявно предполагаем, что самосвязи уже добавлены к каждому узлу в графе (для обозначения без беспорядка).



(a) Karate club network (b) Random weight embedding

Рисунок 3: *Слева*: сеть клубов карате Закари (Zachary, 1977), цветами обозначены сообщества, полученные с помощью кластеризации на основе модульности (Brandes et al., 2008). *Справа*: Вложения, полученные из необученной 3-слойной модели GCN (уравнение 13) со случайными весами, примененными к сети клуба карате. Лучше всего смотреть на экране компьютера.

Мы используем безликий подход, устанавливая $X = IN$, где IN — *это матрица тождества* N *на* N.N — количество узлов в графе. Обратите внимание, что узлы упорядочиваются случайным образом (т.е. порядок не содержит никакой информации). Кроме того, мы выбираем размерность скрытого слоя[6] 4 и двумерный вывод (чтобы вывод можно было сразу визуализировать на графике с 2 размерами).

На рисунке 3b показан репрезентативный пример встраивания узлов (выходы Z), полученные из необученной модели GCN, примененной к сети клуба карате. Эти результаты сопоставимы с внедрениями, полученными с помощью DeepWalk (Perozzi et al., 2014), в котором используется более дорогая процедура обучения без учителя.

A.2ПОЛУКОНТРОЛИРУЕМЫЕ ВСТРАИВАНИЯ УЗЛОВ

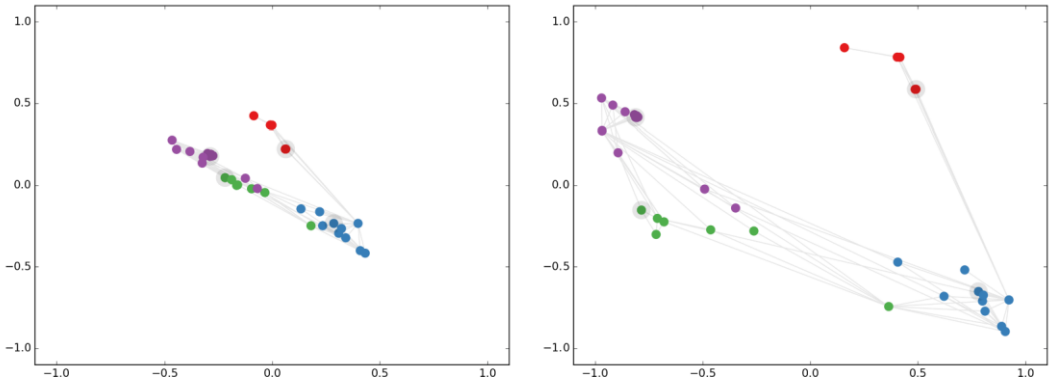
На этом простом примере ОУИ, примененного к сети карат-клубов, интересно понаблюдать, как вложения реагируют во время тренировки над задачей на полуконтролируемую классификацию. Такая визуализация (см. рис. 4) дает представление о том, как модель GCN может использовать структуру графа (и функции, извлеченные из структуры графа на последующих уровнях) для изучения вложений, полезных для задачи классификации.

Мы рассмотрим следующую схему полуконтролируемого обучения: мы добавляем слой softmax поверх нашей модели (уравнение 13) и обучаемся, используя только один размеченный пример для каждого класса (т.е. общее количество 4 помеченных узлов). Мы обучаемся в течение 300 итераций обучения с использованием Adam (Kingma & Ba, 2015) с коэффициентом обучения, равным 0.01 о потерях перекрестной энтропии.

На рисунке 4 показана эволюция встраивания узлов в течение ряда обучающих итераций. Модель успешно разделяет сообщества на основе минимального контроля и только структуры графа. Видео полного процесса обучения можно найти на нашем сайте[7].

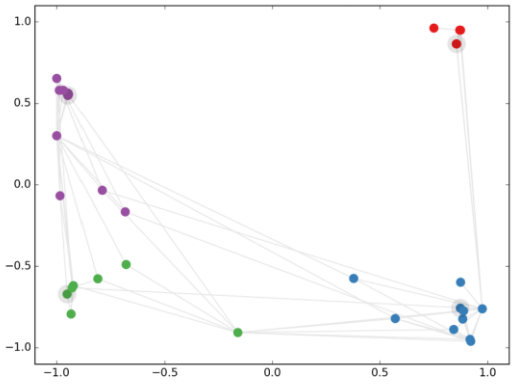
[6] Первоначально мы экспериментировали с размерностью скрытого слоя 2 (т.е. такой же, как и у выходного слоя), но заметили, что размерность 4 приводит к менее частому насыщению $\tanh(\cdot)$ единиц и, следовательно, визуально более приятные результаты.

[7] http://tkipf.github.io/graph-convolutional-networks/

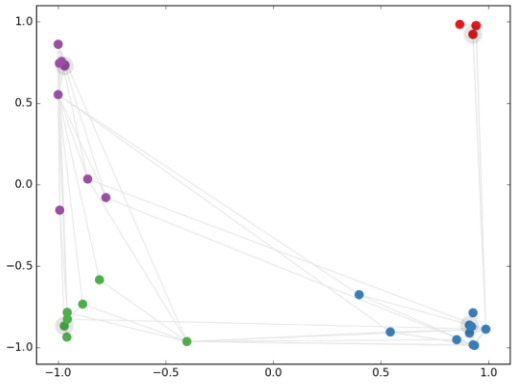


(a) Итерация 25

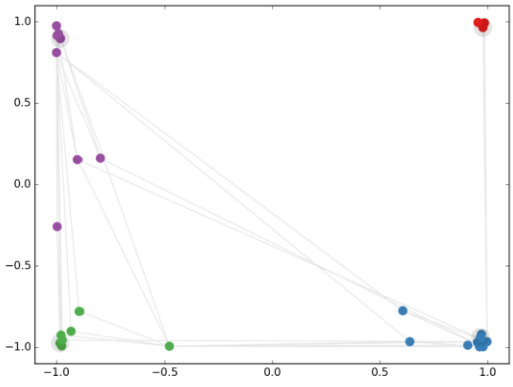
(b) Итерация 50



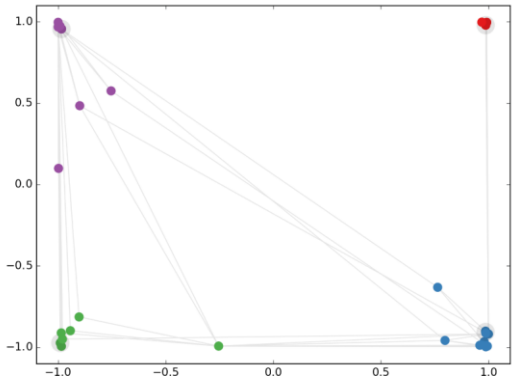
(c) Итерация 75



(d) Итерация 100



(e) Версия 200



(f) Итерация 300

Рисунок 4: Эволюция встраиваемых узлов сети клуба карате, полученных из модели GCN после ряда полуконтролируемых итераций обучения. Цвета обозначают класс. Узлы, метки которых были предоставлены во время обучения (по одной на класс), выделены серым контуром. Серые связи между узлами обозначают ребра графа. Лучше всего смотреть на экране компьютера.

ВЭКСПЕРИМЕНТЫ ПО ГЛУБИНЕ МОДЕЛИ

В этих экспериментах мы исследуем влияние глубины модели (количества слоев) на эффективность классификации. Мы сообщаем о результатах 5-кратного эксперимента по перекрестной проверке наборов данных Cora, Citeseer и Pubmed (Sen et al., 2008) с использованием всех меток. В дополнение к стандартной модели GCN (уравнение 2) мы сообщаем о результатах по варианту модели, в котором мы используем остаточные связи (He

et al., 2016) между скрытыми слоями, чтобы облегчить обучение более глубоких моделей, позволяя модели переносить информацию из входных данных предыдущего слоя:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) + H^{(l)} \quad (14)$$

На каждом сплите перекрестной валидации мы обучаемся в течение 400 эпох (без ранней остановки) с помощью оптимизатора Адама (Kingma & Ba, 2015) с коэффициентом обучения, равным 0.01. Остальные гиперпараметры выбираются следующим образом: 0,5 (коэффициент отсева, первый и последний слой), $5 \cdot 10{-}4$ (регуляризация L2, первый слой), 16 (количество единиц для каждого скрытого слоя) и 0,01 (коэффициент обучения). Результаты обобщены на рисунке 5.

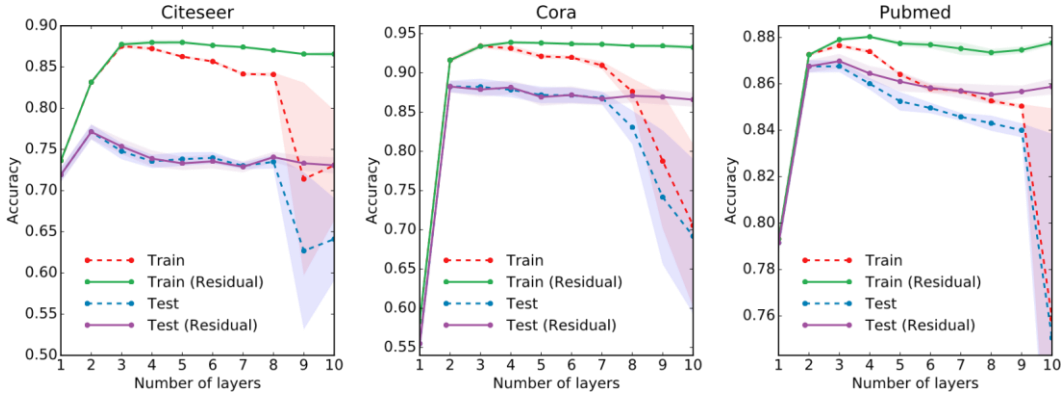


Рисунок 5: Влияние глубины модели (количества слоев) на эффективность классификации. Маркеры обозначают среднюю точность классификации (обучение или тестирование) для 5-кратной перекрестной валидации. Затененные области обозначают стандартную ошибку. Мы показываем результаты как для стандартной модели GCN (пунктирные линии), так и для модели с добавленными остаточными связями (He et al., 2016) между скрытыми слоями (сплошные линии).

Для рассматриваемых здесь наборов данных наилучшие результаты достигаются при использовании 2- или 3-слойной модели. Мы видим, что для моделей с глубиной более 7 слоев обучение без использования остаточных связей может стать затруднительным, так как эффективный размер контекста для каждого узла увеличивается на величину его *окрестности K-го* порядка (для модели с K слоями) с каждым дополнительным слоем. Кроме того, переобучение может стать проблемой, так как количество параметров увеличивается с глубиной модели.