

Semi-supervised Classification with Graph Convolutional Networks

Выполнил: Наумов Антон, БПМИ162

Постановка задачи

- Данные в графе (e.g. граф цитирований)
- Требуется каждой вершине сопоставить класс
- Малое подмножество вершин размечено

(semi-supervised graph-based multi-labeled classification)

Постановка задачи (чуть формальнее)

- $G = (V, E)$ – граф с N вершин (ненаправленный)
- $v_i \in V$ - вершины, $(v_i, v_j) \in E$ – рёбра
- $X \in \mathbb{R}^{N \times D}$ - матрица объект-признак для вершин
- $A \in \mathbb{R}^{N \times N}$ - матрица смежности графа (бинарная/взвешенная)
- $D_{ii} = \sum_j A_{ij}$ - степень вершины i
- $\Delta = D - A$ - Laplacian matrix of G

Классификация (как делалось до)

- Обучаем нейросеть $f(X)$ (дифференцируемая функция)
- $\mathcal{L} = \mathcal{L}_0 + \lambda \mathcal{L}_{reg} - loss$
- \mathcal{L}_0 - loss на размеченных вершинах
- $\lambda \mathcal{L}_{reg}$ - компонента в loss, отвечающая за неразмеченные вершины
- $\mathcal{L}_{reg} = \sum_{i,j} A_{ij} \|f(X_i) - f(X_j)\|^2 = f(X)^T \Delta f(X)$

(Минусы)

Основывается на предположении, что вершины, соединённые ребром, принадлежат одному классу

Предположение очень сильное и ограничивает модель

Как исправить? (идейно)

- Будем обучать нейросеть $f(X, A)$ на \mathcal{L}_0
- За счёт занесения A на вход нейросети можно использовать структуру графа при распространении градиентов через нейросеть

1. Propagation rule

Graph Convolutional Network (GCN) – layer-wise rule:

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

- $\tilde{A} = A + I_N$ - матрица смежности с петлями на всех вершинах
- $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ - степень вершины i
- $\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$
- $\sigma(\cdot)$ – нелинейность (ReLU, softmax)
- $H^{(l)} \in \mathbb{R}^{N \times D^{(l)}}$ - активации на слое l ; ($H^{(0)} = X$)
- $W^{(l)} \in \mathbb{R}^{D^{(l)} \times D^{(l+1)}}$ - веса на слое l

(мотивацию попробуем разобрать в конце, если будет время)

2. Semi-supervised node classification (example)

Рассмотрим 2-слойную GCN:

- *Preprocessing:* $\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$

- *Forward:*

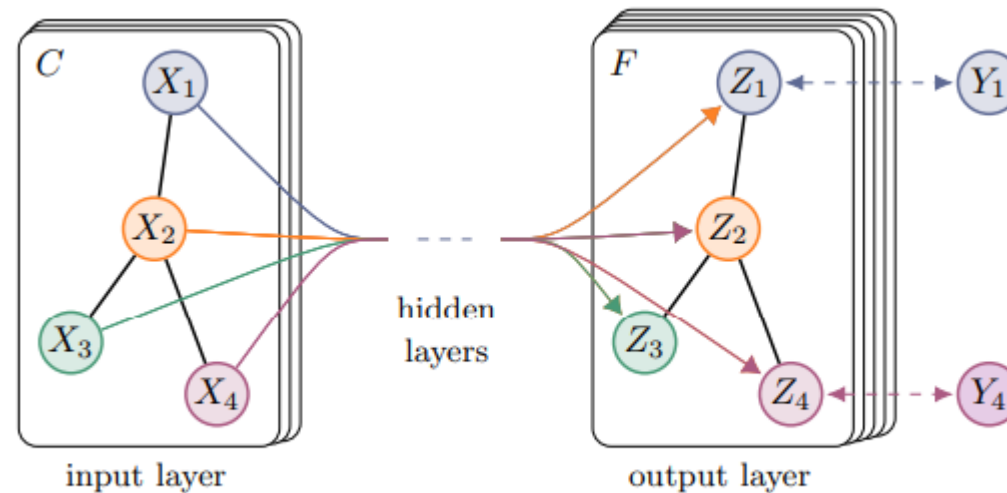
$$Z = f(X, A) = \text{softmax}(\hat{A} \text{ReLU}(\hat{A} X W^{(0)}) W^{(1)})$$

, где $X \in \mathbb{R}^{N \times C}$, $W^{(0)} \in \mathbb{R}^{C \times H}$, $W^{(1)} \in \mathbb{R}^{H \times F}$, F - количество классов

- *Cross-entropy:*

$$\mathcal{L} = - \sum_{l \in \mathcal{S}_{Labeled}} \sum_{f=1}^F Y_{lf} \ln Z_{lf}$$

- Обучаем градиентным спуском (батчи, dropout)



(a) Graph Convolutional Network

Эксперименты (данные)

Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001

- *Citation networks:*
Разряженные матрицы BoW как фичи документов
Цитирование – бинарная ненаправленная связь между документами
У каждого документа есть класс, для обучения только 20 примеров из каждого класса
- *NELL (Never-ending language learning):*
Граф сущностей, соединённый направленными именованными связями
Сущности e_i
Связи $(e_1, r, e_2) \rightarrow (e_1, r_1), (e_2, r_2)$
- *Random graphs ($|V| = N, |E| = 2N$)*

Результаты

Table 2: Summary of results in terms of classification accuracy (in percent).

Method	Citeseer	Cora	Pubmed	NELL
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [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
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	70.3 (7s)	81.5 (4s)	79.0 (38s)	66.0 (48s)
GCN (rand. splits)	67.9 \pm 0.5	80.1 \pm 0.5	78.9 \pm 0.7	58.4 \pm 1.7

- *2-слойная GCN*
- *Тестировали на 1000 размеченных примерах*
- *Максимум 200 эпох*
- *Гиперпараметры настроили на Cora и использовали во всех остальных*
- *Adam, lr=0.01, early stopping (window=10)*
- *Init весов по статье + Row-normalization признаков*
- *Dropout на всех слоях, L2-рег на первом слое, разбиение из статьи*

Ограничения

- Через приближение в *propagation rule* (не разбирали) неявно предполагается зависимость от окрестности k -го порядка для GCN с k слоями
- Ограничены ненаправленными графами, но, как видно из NELL, можно использовать трюк для перехода к ненаправленным
- На *random graphs* установили, что GPU может не тянуть по памяти без стохастических мини-батчей для больших графов

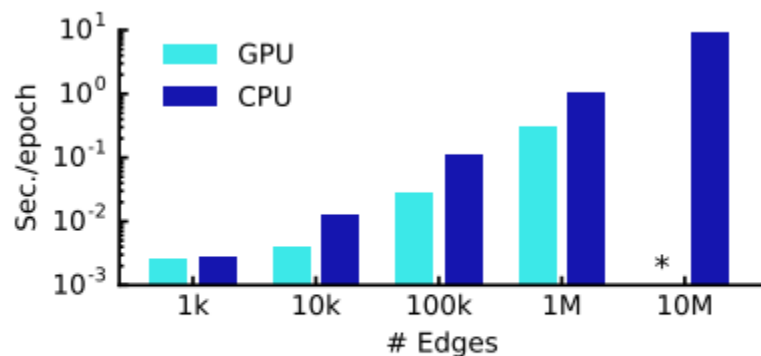
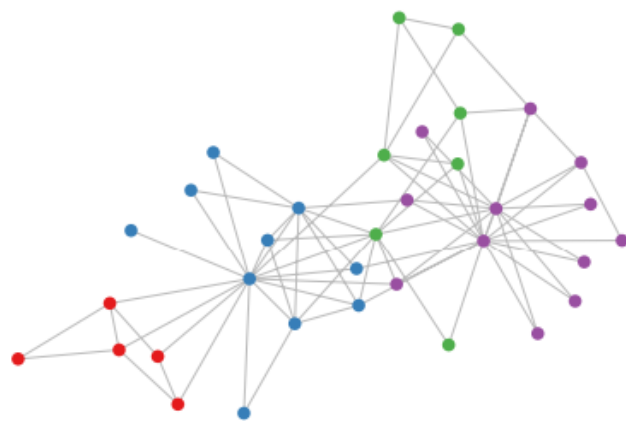
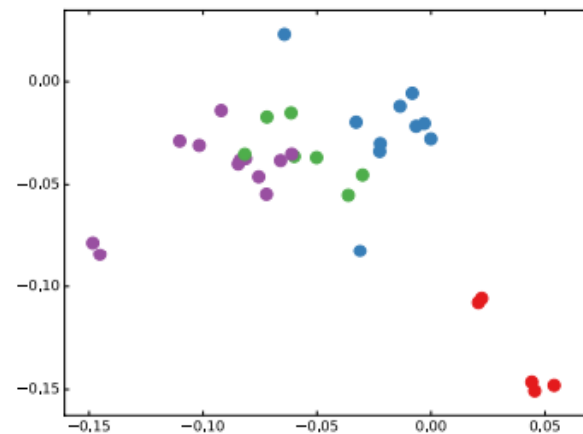


Figure 2: Wall-clock time per epoch for random graphs. (*) indicates out-of-memory error.

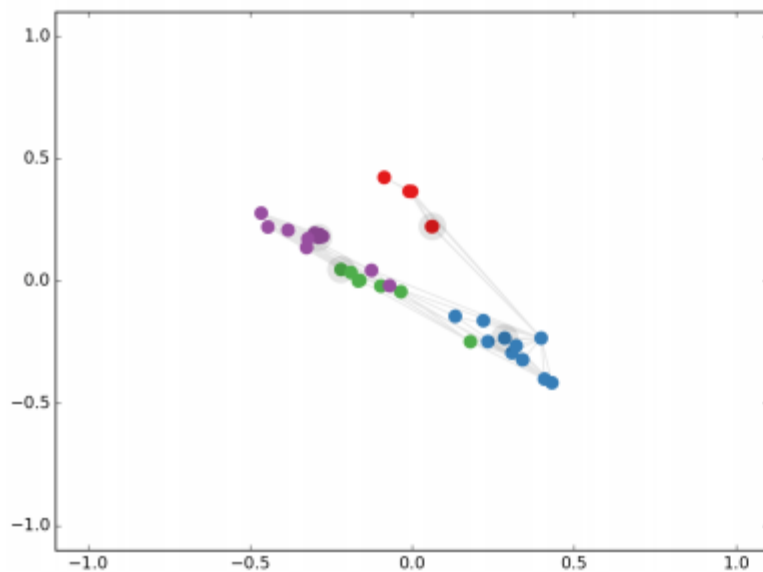
Бонус (эмбеддинги)



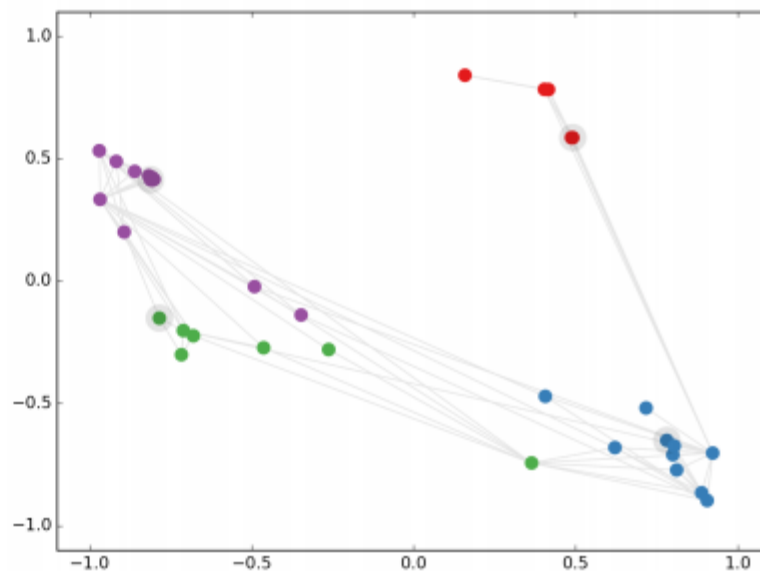
(a) Karate club network



(b) Random weight embedding

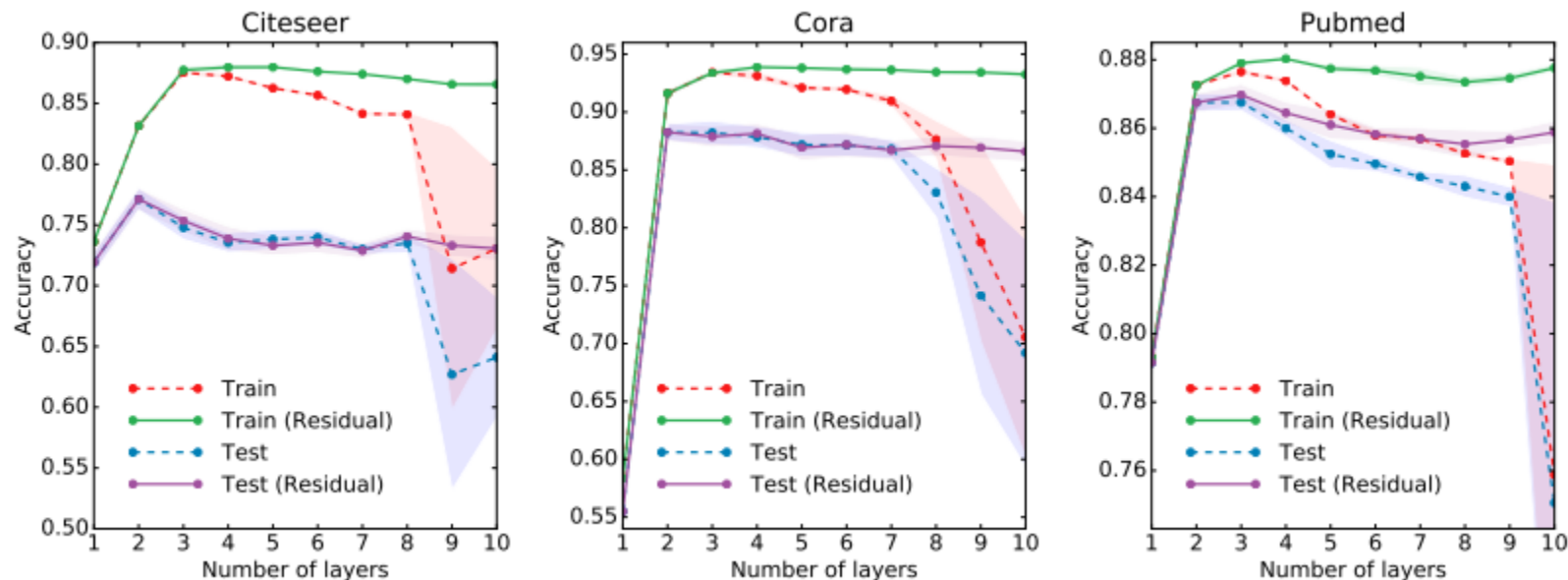


(a) Iteration 25



(b) Iteration 50

Бонус (глубина сети)



Residual connections in GCN:

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

Ссылка на статью: <https://arxiv.org/pdf/1609.02907.pdf>