Pedro Thomas Pereira antunes

Identificação de autoria e classificação de textos usando redes convolucionais em grafos

Belo Horizonte

2019/2

Pedro Thomas Pereira antunes

Identificação de autoria e classificação de textos usando redes convolucionais em grafos

Relatório técnico apresentado como requisito da disciplina de Projeto Orientado em computação do DCC/UFMG

Orientador:
Dr. Renato Vimieiro - Departamento de Ciência da Computação

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Belo Horizonte

2019/2

Resumo

Nesse trabalho é explorado o conceito de convolução em grafos para aprendizado de máquinas e como é possível expandí-lo para outras finalidades. Para tal, usando um conjunto de dados contendo textos em diferentes domínios, determinar sua autoria a partir da estruturação como um sub-grafo de um vocabulário aplicando operações de convolução dos nós.

Palavras-chaves: Grafo, Convolução, Aprendizado profundo.

Sumário

LISTA DE FIGURAS

1 INTRODUÇÃO p. 5

2 CONTEXTUALIZAÇÃO E TRABALHOS RELACIONADOS p. 6

3 DESENVOLVIMENTO DO TRABALHO p. 8

3.1 Rede de publicações . . . . . p. 8

3.2 Atribuição de autoria . . . . . p. 8

4 RESULTADOS E DISCUSSÃO p. 9

4.1 Rede de publicações . . . . . p. 9

4.2 Atribuição de autoria . . . . . p. 10

5 CONCLUSÕES E TRABALHO FUTUROS p. 12

Referências p. 13

LISTA DE FIGURAS

1 Acurácia do modelo no conjunto de dados da rede de citações . . . . . p. 9

2 Acurácia para os individuais problemas em atribuição de autoria . . . . p. 10

3 F1-macro para os individuais problemas em atribuição de autoria . . . p. 10

1 INTRODUÇÃO

Um grafo é um tipo abstrato de dados que possibilita representar dados e suas relações. Apesar de poderem ser usados de inúmeras formas diferentes, algoritmos de mineração de dados em grafos ainda não são amplamente usados devido a explosão da complexidade de tempo em um número de vértices grande o suficiente para o uso em casos reais(1). Em contraponto a soluções analíticas, algoritmos de aprendizado de máquina com base em redes neuronais possibilitam modelar classificadores e regressores sem incorrer aos mesmo altos custos computacionais.

Uma nova técnica emergente dentro do área de redes neurais para lidar com esse tipo dados chama-se Graph Convolutional Networks (GCN) e possibilita que cada neurônio extraia informações de topologia e informações estruturais de dados que podem ser representados como grafos. O objetivo deste trabalho sendo explorar esse tipo de rede em outros contextos além do aprendizado de representações de nós na identificação de autoria e classificação de textos estruturados como sub-grafos de um vocabulário e avaliá-los para entender como esse tipo de rede se comporta.

2 CONTEXTUALIZAÇÃO E TRABALHOS RELACIONADOS

Esse trabalho tem primeiramente como base os modelos de convolução formulados em Semi-Supervised Classification with Graph Convolutional Networks(2) que utiliza uma matriz simétrica $\hat{A} = D^{-1/2}AD^{-1/2}$ representando a topologia do grafo, onde A é a matriz de adjacência do grafo e $D_{ii} = \sum_j A_{ij}$ e as convoluções são feitas usando a função de *forward* $f(X, A) = ReLU(\hat{A}XW + b)$, onde W é o *kernel* de convolução, b é o viés e X é a entrada que vêm da camada de convolução anterior, onde na primeira camada é apenas uma matriz identidade.

Também é necessário entender a solução proposta em Graph Convolutional Networks for Text Classification(3) para o uso da GCN em textos. Para tal, considera-se que existem nós documentos e nós tokens no grafo onde a matriz de adjacência definida como:

$$A_{ij} = \begin{cases} \text{PMI}(i,j), & i,j \text{ são tokens}, \text{PMI}(i,j) > 0 \\ \text{TF-IDF}_{ij}, & i \text{ é um documento e } j \text{ é um token} \\ 1, & i = j \\ 0, & \text{resto} \end{cases}$$

Sendo PMI é calculado:

$$\text{PMI}_{ij} = log\frac{p(i,j)}{p(i)p(j)}$$
$$p(ij) = \frac{\#\text{W}(i,j)}{\#\text{W}}$$
$$p(i) = \frac{\#\text{W}(i)}{\#\text{W}}$$

onde $\#\text{W}(i)$ é o número de janelas que contém o token i, $\#\text{W}(i,j)$ é o número de

janelas que contém ambos os tokens i e j e #W é o número de janelas.

As bases de dados usadas para testar hipóteses foram o conjunto de dados CORA(4), que é uma rede de citações de artigos científicos com 2708 artigos e 5429 citações, e *Cross-Domain Authorship Attribution 2019*(5) que contém 20 problemas com cada problema constituindo de 9 autores e 7 textos conhecidos por autor em vários universos de fanfics diferentes para treino do modelo, e um número variado de textos para teste. Os problemas são em 4 línguas (Inglês, Espanhol, Francês e Italiano) com 5 problemas para cada língua.

3 DESENVOLVIMENTO DO TRABALHO

3.1 Rede de publicações

O desenvolvimento inicial é facilitado imensamente com diversas implementações da GCN disponíveis, sendo a usada a StellarGraph(6) para o conjunto de dados CORA. Nesse problema o grafo foi representado como os publicações sendo nós e as arestas indicavam as citações entre elas. Cada nó do grafo tinha features codificadas na matriz X de entrada pelo modelo proposto por Thomas Kipf.

3.2 Atribuição de autoria

O objetivo mais interessante de atribuição de autoria, onde é necessário usar a convolução de nós para representar sub-grafos. Inicialmente foram feitas usando a biblioteca spaCy para criar representações vetoriais de cada token e ligá-los quando houver coocorrência no corpo dos textos, gerando para cada autor um grafo conexo. Para o treino do modelo, cada token era classificado pertencente a um autor se ele a usou. Dessa forma uma mesmo token teria várias classificações diferentes e essa abordagem foi rapidamente abandonada devido aos péssimos resultados iniciais e aconselhamento do orientador.

Usando a mesma abordagem que Soh Wee Tee(7) que faz uma implementação baseado no artigo da text-GCN(3) para classificar passagens da bíblia com seu capítulo, foi feito o mesmo para tentar melhorar os resultados da classificação de autoria, porém, diferentemente de Soh Wee Tee, uma implementação mais fiel e rápida para préprocessar da text-gcn implementado tanto usando a biblioteca PyTorch(8).

4 RESULTADOS E DISCUSSÃO

4.1 Rede de publicações

O aprendizado de representações na rede de citações de publicações científicas é algo já estudado e os resultados também foram próximos de outras publicações. Os modelos consistiam de uma camada com 64 *kernels* na primeira camada e 32 na segunda, com 70% de *dropout*.

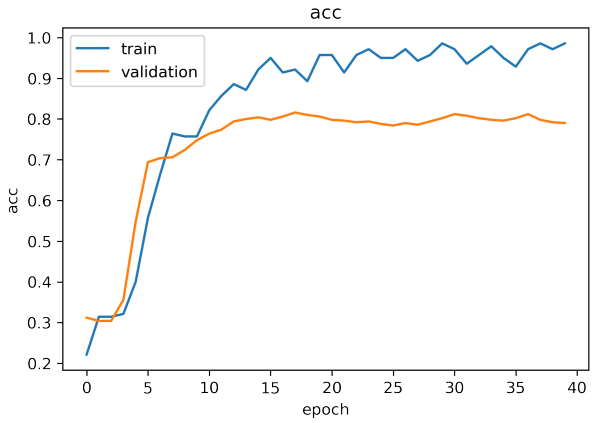

Figura 1: Acurácia do modelo no conjunto de dados da rede de citações

Método	Acurácia	Ano
DFNet-ATT(9)	**86.0%**	2019
GResNet (GAT)(10)	85.5%	2019
GCN (nosso)	81.4%	2019
MTGAE(11)	79.0%	2018

Tabela 1: Comparativo entre métodos no conjunto de dados da rede de citações.

Sendo que existem 7 categorias diferentes, uma acurácia de 81.4% é bem razoável. Além do mais, o modelo converge rapidamente com apenas 40 épocas e leva apenas alguns

segundos em uma nVidia GeForce GTX 1650.

4.2 Atribuição de autoria

Para o conjunto de dados de atribuição de autoria deve se notar a diante que nem todos os 20 problemas foram executados com sucesso usando uma máquina com 16GB de ram e uma nVidia tesla P100 (16GB), faltando memória nos problemas 8 e 19. Os testes foram rodados 10 vezes com apenas uma camada de 64 *kernels* de convolução para gerar representações e uma camada com final de convolução para cada uma das 9 classes representando os autores e 1000 épocas: table

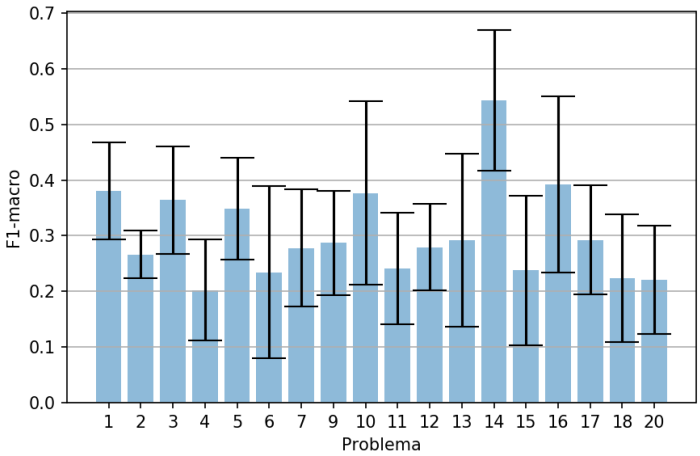

Figura 2: Acurácia para os individuais problemas em atribuição de autoria

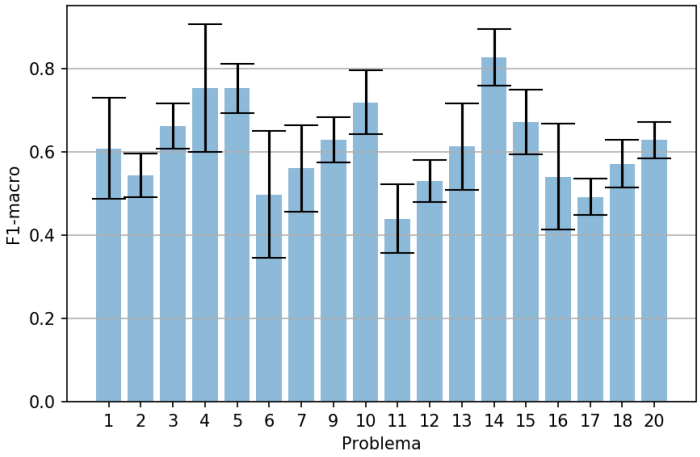

Figura 3: F1-macro para os individuais problemas em atribuição de autoria

Sendo a métrica f1-macro a escolhida pelos organizadores da competição, é possível ver que houve uma grande variância nos resultados obtidos com média de 0.303 e desvio

padrão de 0.0337. A acurácia média obtida foi de 61.3% com desvio padrão de 13.4%.

Como dito anteriormente, para alguns testes a memória foi um impedimento para rodá-los, apesar de os testes levarem de 30 a um pouco mais de um minuto para treinar os modelos, esse aspecto dificulta o emprego desta técnca já que a matriz de adjacência não pode ser dividida para o treino ser feito em batches.

Algo interessante que ocorreu com esse conjunto de dados foi que o modelo com mais de uma camada para extrair representações causava *overfitting* rapidamente e os resultados ficavam muito piores, sendo que Thomas Kipf(2) em sua implementação e o artigo do text-gcn(3) notou que em vários conjuntos de dados durante a classificação de nós em grafos obtiam os melhores resultados com duas camadas.

Implementações e algoritmos deste trabalho estão disponíveis em:
`https://github.com/pedrotpa/poc_text_gcn`

5 CONCLUSÕES E TRABALHO FUTUROS

Ficou evidente durante o desenvolvimento que ainda é necessário solucionar problemas relacionados ao gerenciamento da memória para viabilizar o treinamento usando GPUs. Em contrapartida, os modelos convergem rapidamente. Algoritmos de aprendizado profundo baseados em grafos apareceram recentemente e ainda é necessário experimentar e descobrir como aplicá-los onde a modelagem como um problema de classificação em grafos não é óbvia. Contudo, em problemas de classificação de nós os resultados são excelentes como visto não apenas neste trabalho como em outros. Para trabalhos futuros ainda é necessário avaliar os custos computacionais em cada conjunto de dados, se possível, mudar a implementação para possibilitar o treinamento dos modelos em *batches*.

Referências

1 WASHIO, T.; MOTODA, H. State of the art of graph-based data mining. *Acm Sigkdd Explorations Newsletter*, Acm, v. 5, n. 1, p. 59–68, 2003.

2 KIPF, T. N.; WELLING, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

3 YAO, L.; MAO, C.; LUO, Y. Graph convolutional networks for text classification. *CoRR*, abs/1809.05679, 2018. Disponível em: <http://arxiv.org/abs/1809.05679>.

4 MCCALLUM, A. *Cora Dataset*. Texas Data Repository Dataverse, 2017. Disponível em: <https://doi.org/10.18738/T8/HUIG48>.

5 KESTEMONT, M. et al. Overview of the Cross-domain Authorship Attribution Task at PAN 2019. In: CAPPELLATO, L. et al. (Ed.). *CLEF 2019 Labs and Workshops, Notebook Papers*. CEUR-WS.org, 2019. (CEUR Workshop Proceedings). Disponível em: <http://ceur-ws.org/Vol-2380/>.

6 DATA61, C. *StellarGraph Machine Learning Library*. [S.l.]: GitHub, 2018. `https://github.com/stellargraph/stellargraph`.

7 TEE, p. S. W. *Bible Text GCN*. [S.l.]: GitHub, 2019. `https://github.com/plkmo/Bible_Text_GCN`.

8 PASZKE, A. et al. Automatic differentiation in PyTorch. In: *NeurIPS Autodiff Workshop*. [S.l.: s.n.], 2017.

9 WIJESINGHE, A.; WANG, Q. *DFNets: Spectral CNNs for Graphs with Feedback-Looped Filters*. 2019.

10 ZHANG, J.; MENG, L. *GResNet: Graph Residual Network for Reviving Deep GNNs from Suspended Animation*. 2019.

11 TRAN, P. V. Learning to make predictions on graphs with autoencoders. In: *5th IEEE International Conference on Data Science and Advanced Analytics*. [S.l.: s.n.], 2018.