GCN

SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS (ICLR2017)

paper: [https://arxiv.org/pdf/1609.02907.pdf]

code: [https://github.com/tkipf/gcn]

Important Concepts:

1. spectral graph convolutions:

   $g_\theta \star x = U g_\theta U^\top x$

   where:

   - U: the matrix of eigenvalues of the normalized graph Laplacian $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^\top$
   - Λ: diagonal matrix of its eigenvalues
   - $g_\theta = diag(\theta) = g_\theta(\Lambda)$: parameterized by $\theta \in \mathbb{R}^N$ in the Fourier domain

Key Idea:

1. graph edges need not necessarily encode node similarity, but could contain additional information
2. $g_\theta(\Lambda)$ can be well-approximated by a truncated expansion in terms of Chebyshev polynomials $T_k(x)$ up to K^{th} order:

   $g_{\theta'}(\Lambda) \approx \sum_{k=0}^{K} \theta'_k T_k(\widetilde{\Lambda})$

   1. $\widetilde{\Lambda} = \frac{2}{\lambda_{max}} \Lambda - I_N$, λ_{max} denotes the largest eigenvalue of L
   2. $\theta' \in \mathbb{R}^K$: vector of Chebyshev coefficients
   3. $Chebyshev \quad polynomial : T_0(x) = 1, \quad T_1(x) = x, \quad T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$
   4. SO: $g_{\theta'} \star x \approx U \sum_{k=0}^{K} \theta'_k T_k(\widetilde{\Lambda}) U^\top x = \sum_{k=0}^{K} \theta'_k T_k(\widetilde{L}) x$
      1. $\widetilde{L} = \frac{2}{\lambda_{max}} L - I_N$
      2. $K^{th} - order$ polynomial in the Laplacian: it depends only on nodes that are at maximum K steps away from the cantral node
      3. with $K = 1$ and $\lambda_{max} \approx 2$:

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

         with $\theta = \theta'_0 = -\theta'_1$ and renormalization trick $I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \rightarrow \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}$

         from C input channelsand F filters:

$Z = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}X\Theta$: complexity:$\mathcal{O}(|\varepsilon|FC)$

1. $\Theta \in \mathbb{R}^{C\times F}$: matrix of filter parameters
2. $Z \in \mathbb{R}^{N\times F}$: convolved signal matrix

3. layer-wise propagation rule:

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

   $\widetilde{A} = A + I_N$: adjacency matrix with added self connection

   $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$:

   $W^{(l)}$: layer-specific trainable weight matrix

   $\sigma(\cdot)$: activation function

   $H^{(l)} \in \mathbb{R}^{N\times D}$: the matirx of activations in the l^{th} layer, $H^{(0)} = X$

4. semi-supervised node classification

   $\hat{A} = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$

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

Problem definition:

1. Graph-based semi-supervised learning: classifying nodes in a graph, where labels are only available for a small subset of nodes
2. definition of a node ordering in preprocessing step

Precessed Work:

1. explicit graph-based regularization
2. graph-Laplacian regularization: assume edges encode similarity of nodes
3. skip-gram: difficult to optimize

Metrix gain:

1. first-order approximation of spectral graph convolutions --> fast approximate convolutions
2. datasets: Citeseer, Cora, Pubmed, NELL
3. faster and higher accuracy

Challenge:

1. mini-batch stochastic gradient extension
2. not naturally support edge feature and directed graph (but possible to handle)
3. definition of locality and equal importance of self-connections and edges to neighboring nodes