南京大学
NANJING UNIVERSITY

PASA Lab

图神经网络的并行化

报告人：王云攀

时间：2019.10.31

汇报内容

PASA Lab

一、图神经网络概述

二、1stChebNet算法[1]介绍及实现

三、下一步工作

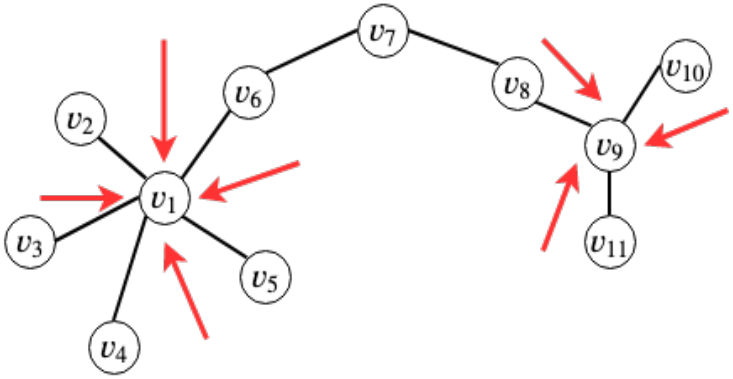
1. Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

一、图神经网络概述

一、图神经网络概述

1. 什么是图神经网络?

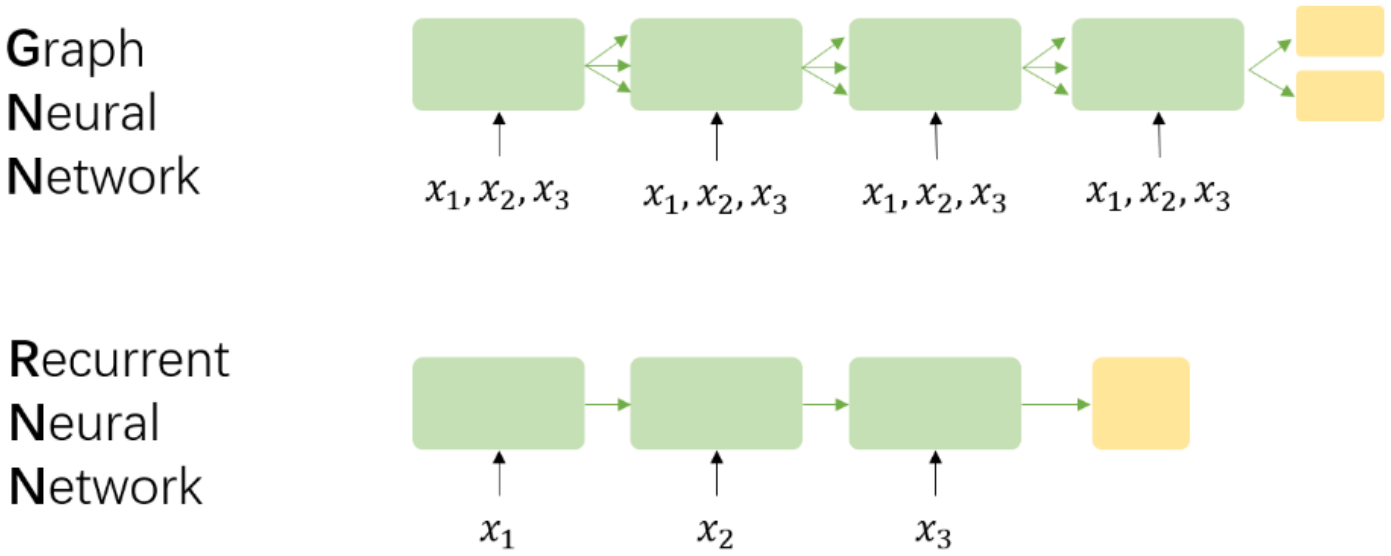
- 输入
  - 图的结构和节点的内容信息
- 输出
  - Node-level，关于节点的回归和分类任务，最后一层为softmax函数或者感知机
  - Edge-level，对边的分类或者预测任务，一个额外的函数用于连接边的表示
  - Graph-level，图的分类任务，最后一层为pooling层
- 学习类型
  - 半监督学习，Node-level classification，先学习有标记数据，再对未标记数据进行分类
  - 监督学习，Graph-level classification, 给出图数据集，预测整个图的类别标签。
  - 无监督学习：Graph-embedding, 学习图的表示
- 每层的基本操作
  - 聚集邻居节点的信息产生新的表示
  - 每个节点的表示再表示



一、图神经网络概述

1. 什么是图神经网络?

- GNN vs RNN[1]



[1]. 图片来源于https://www.cnblogs.com/SivilTaram/p/graph_neural_network_1.html

一、图神经网络概述

PASA Lab

2. 为什么使用图神经网络?

- 背景[1]
  - 深度学习在图像、视频和文本应用广泛，从数据中挖掘出复杂的模式的表达能力已得到公认。
  - 另一方面，Graph在现实世界中无处不在，表示对象以及关系，例如社交网络、电子商务网络、生物学网络和交通网络。
  - 图具有复杂的结构，包含丰富的基础价值。
- 特点[2]
  - Each graph has a variable size of unordered nodes
  - Each node in a graph has a different number of neighbors

[1]. Zhang Z, Cui P, Zhu W. Deep learning on graphs: A survey[J]. arXiv preprint arXiv:1812.04202, 2018.

[2]. Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. arXiv preprint arXiv:1901.00596, 2019.

PASA Lab

3. 图神经网络的历史

- 萌芽期（2005~2009）
  - Gori et al.(2005)[1]，基于不动点理论，考虑节点和边的特征，学习节点的表示，想法是通过结点的信息传播使整张图达到收敛，在其基础上进行预测。
    - 问题：试图寻找确定解，达到收敛使得迭代长度无法确定，采用AP算法进行反向传播，需要对函数进行特别设计，满足收敛性。同时，计算开销巨大。
  - Scarselli et al.(2008) [2], Micheli (2009) [3], 在以上的基础上进行改进，但仍然基于迭代达到稳定状态的想法来做，没有解决计算开销的问题。

[1]. Gori M, Monfardini G, Scarselli F. A new model for learning in graph domains[C]//Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. IEEE, 2005, 2: 729-734.

[2]. Scarselli F, Gori M, Tsoi A C, et al. The graph neural network model[J]. IEEE Transactions on Neural Networks, 2008, 20(1): 61-80.

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

PASA Lab

3. 图神经网络的历史

- 蓬勃发展期（2013至今）
  - Spectral CNN，ICLR 2014[1], 首次将卷积操作应用到图神经网络中，基于频域的方法进行考虑
  - Graph Sequence Neural Network, ICLR 2016[2] ，将GRU引入，将GNN固定在了T步，并且可以来预测序列任务
  - Graph Attention Network, ICLR 2017[3] 开始考虑了邻居节点的权重
  - Inductive representation learning on large graphs, NIPS 2017[4] 提出了GraphSage, 关于图神经网络的一种框架

[1]. Bruna J, Zaremba W, Szlam A, et al. Spectral networks and locally connected networks on graphs[J]. arXiv preprint arXiv:1312.6203, 2013.

[2]. Li Y, Tarlow D, Brockschmidt M, et al. Gated graph sequence neural networks[J]. arXiv preprint arXiv:1511.05493, 2015.

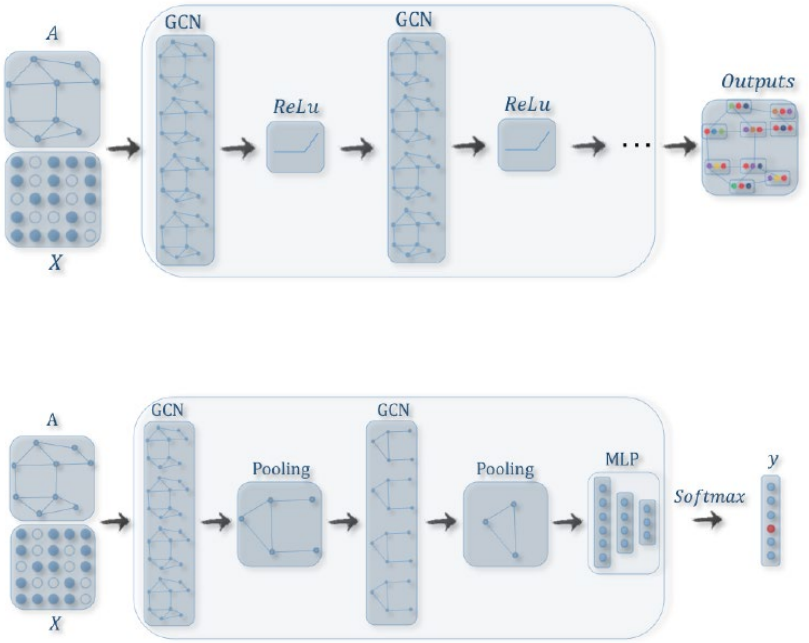
[3]. Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.

[4]. Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[C]//Advances in Neural Information Processing Systems. 2017: 1024-1034.

PASA Lab

4. 图神经网络的分类[1]

- Graph Convolution Networks：模拟传统数据集上的卷积操作，通过自身和邻居的特征来学习节点的表示
  - Spectral Based 方法： 基于频域的方法
  - Spatial Based 方法： 基于空间的方法



[1]. Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. arXiv preprint arXiv:1901.00596, 2019.

4. 图神经网络的分类[1]

- Graph Convolution Networks

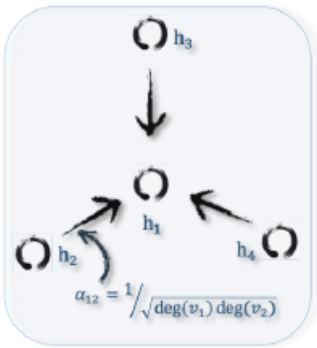
Category	Approach	Inputs (allow edge features?)	Outputs	Output Mechanisms	
				Intermediate	Final
Spectral Based	Spectral CNN (2014) [21]	✗	Graph-level	cluster+max pooling	softmax function
	ChebNet (2016) [12]	✗	Graph-level	efficient pooling	mlp layer+softmax function
	1stChebNet (2017) [14]	✗	Node-level	activation function	softmax function
	AGCN (2018) [23]	✗	Graph-level	max pooling	sum pooling
Spatial Based	GNN (2009) [18]	✓	Node-level	-	mlp layer+softmax function
			Graph-level	-	add a dummy super node
	GGNNs (2015) [19]	✗	Node-level	-	mlp layer/softmax function
			Graph-level	-	sum pooling
	SSE (2018) [20]	✗	Node-level	-	softmax function
	MPNN (2017) [13]	✓	Node-level		softmax function
			Graph-level	-	sum pooling
	GraphSage (2017) [25]	✗	Node-level	activation function	softmax function
	DCNN (2016) [47]	✓	Node-level	activation function	softmax function
			Graph-level	-	mean pooling
	PATCHY-SAN (2016) [27]	✓	Graph-level	-	mlp layer+softmax function
	LGCN (2018) [28]	✗	Node-level	skip connections	mlp layer+softmax function

[1]. Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. arXiv preprint arXiv:1901.00596, 2019.

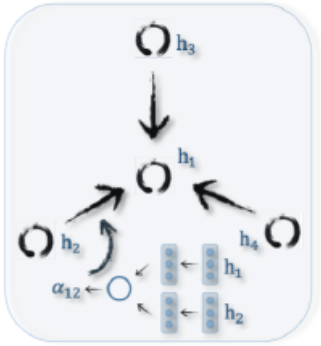
一、图神经网络概述

4. 图神经网络的分类[1]

- Graph Attention Networks：跟GCNs类似，试图融合邻居节点，随机行走(Random Walk), 候选模型来寻找函数。采用了注意力机制，会分配更大的注意力权重给更重要的节点、步行或者模型。



(a) Graph Convolution Networks [14] explicitly assign a non-parametric weight $a_{ij} = \frac{1}{\sqrt{deg(v_i)deg(v_j)}}$ to the neighbor v_j of v_i during the aggregation process.



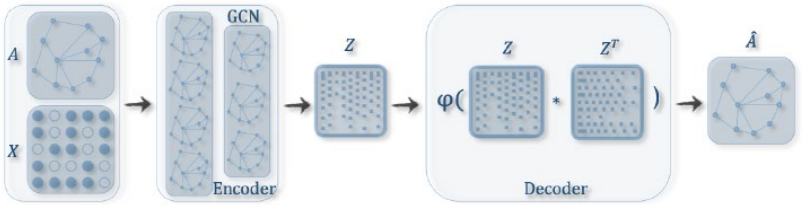
(b) Graph Attention Networks [15] implicitly capture the weight a_{ij} via an end-to-end neural network architecture, so that more important nodes receive larger weights.

[1]. Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. arXiv preprint arXiv:1901.00596, 2019.

PASA Lab

4. 图神经网络的分类[1]

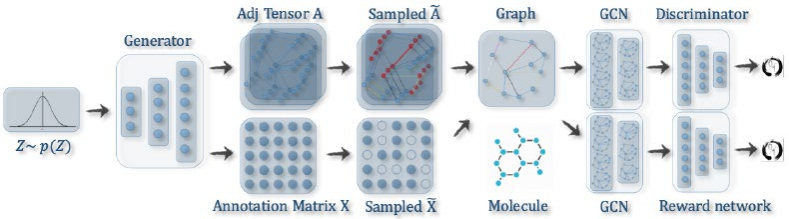
- Graph Auto-encoders: 无监督的学习框架，旨在通过编码器学习节点低维表示，并且能够通过解码器重构图形。它是一种流行的学习图形嵌入（Graph Eembedding）的方法。
  - 在化学图中，原子被视为节点，化学键被视为边缘。 任务是发现新的可合成分子具有某些化学和物理性质。



[1]. Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. arXiv preprint arXiv:1901.00596, 2019.

4. 图神经网络的分类[1]

- Graph Generative Networks: 旨在生成合理图形数据的结构。从图的经验分布上产生图本身是个很难的问题，通常的方法是通过探索将产生的因素交替形成节点和边，进行生成式对抗训练。最广阔的应用领域是化合物的合成

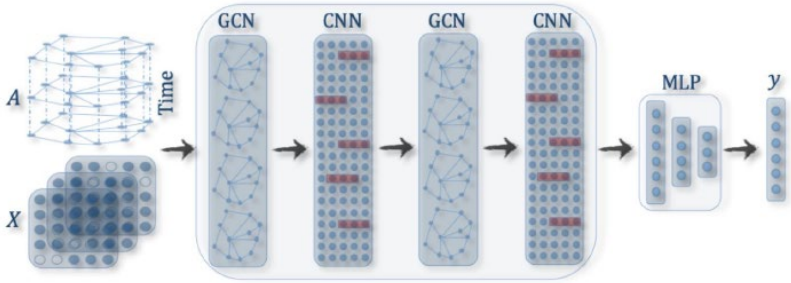


[1]. Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. arXiv preprint arXiv:1901.00596, 2019.

PASA Lab

4. 图神经网络的分类[1]

- Graph Spatial-temporal Networks:
  旨在从时空图中学习看不见的模式，在交通流量预测和人类活动预测应用越来越广泛。图是考虑空间依赖性和时间依赖性。许多当前的方法采用GCN和一些RNN或CNN来对时空依赖性建模



[1]. Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. arXiv preprint arXiv:1901.00596, 2019.

PASA Lab

4. 图神经网络的分类[1]

- 其他

Category	Approaches	Inputs			Outputs	Tasks	GCN Based
		A	D	S			
Graph Attention Networks	GAT (2017) [15]	✓	✓	✗	node labels	node classification	✓
	GAAN (2018) [29]	✓	✓	✗	node labels	node classification	✓
	GAM (2018) [60]	✓	✓	✗	graph labels	graph classification	✗
	Attention Walks (2018) [61]	✗	✗	✗	node embedding	network embedding	✗
Graph Auto-encoder	GAE (2016) [62]	✓	✗	✗	reconstructed adajacency matrix	network embedding	✓
	ARGA (2018) [64]	✓	✗	✗	reconstructed adajacency matrix	network embedding	✓
	NetRA (2018) [65]	✗	✗	✗	reconstructed sequences of random walks	network embedding	✗
	DNGR (2016) [42]	✗	✗	✗	reconstructed PPMI matrix	network embedding	✗
	SDNE (2016) [43]	✗	✓	✗	reconstructed adajacency matrix	network embedding	✗
	DNRE (2018) [66]	✓	✗	✗	reconstructed node embedding	network embedding	✗
Graph Generative Networks	MolGAN (2018) [69]	✓	✗	✗	new graphs	graph generation	✓
	DGMG (2018) [68]	✗	✗	✗	new graphs	graph generation	✓
	GraphRNN (2018) [67]	✗	✗	✗	new graphs	graph generation	✗
	NetGAN (2018) [70]	✗	✗	✗	new graphs	graph generation	✗
Graph Spatial-Temporal Networks	DCRNN (2018) [73]	✗	✗	✓	node value vectors	spatial-temporal forecasting	✓
	CNN-GCN (2017) [74]	✗	✗	✓	node value vectors	spatial-temporal forecasting	✓
	ST-GCN (2018) [75]	✗	✗	✓	graph labels	spatial-temporal classification	✓
	Structural RNN (2016) [76]	✗	✗	✓	node labels/value vectors	spatial-temporal forecasting	✗

[1]. Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. arXiv preprint arXiv:1901.00596, 2019.

PASA Lab

5. 图神经网络未来的方向[1]

- Go Deep：根据[2]，图形卷积的影响推动相邻节点的表示彼此更靠近，因此，理论上，随着无限次的卷积，所有节点的表示将收敛到单个点。这就提出了一个问题，即深入学习图形结构化数据是否仍是一个不错的策略
- Receptive Field：观察一个节点时，考虑更远的邻居；
- Scalability：将图扩展到更大规模的图上，现有办法：快采样和子图训练
- Dynamics and Heterogeneity：现有的图都是静态的同质的图，考虑图的结构变化，节点和边的来源不同的情况

[1]. Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. arXiv preprint arXiv:1901.00596, 2019.

[2]. Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in Proceedings of the AAAI Conference on Artificial Intelligence, 2018.

二、1stChebNet算法介绍及其实现

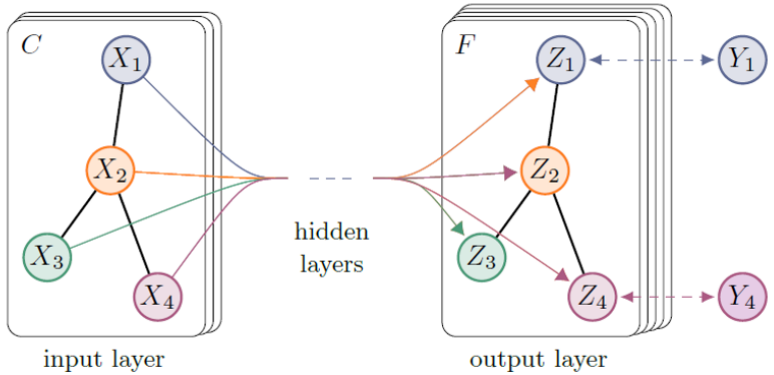
二、1stChebNet算法介绍及实现

1. 算法介绍

- 该论文属于Graph Convolution Network中Spectral Based的方法，实验结果只使用了三层网络就取得了很好的实验结果。
- 所属类型：
  - 半监督学习
  - inductive learning
  - Node-level, 学习节点的类别
- 该三层网络结构也可以很容易的扩展到多层（n+1层）网络结构，表达式如下所示：

$$\mathcal{L} = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf},$$

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



(a) Graph Convolutional Network

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

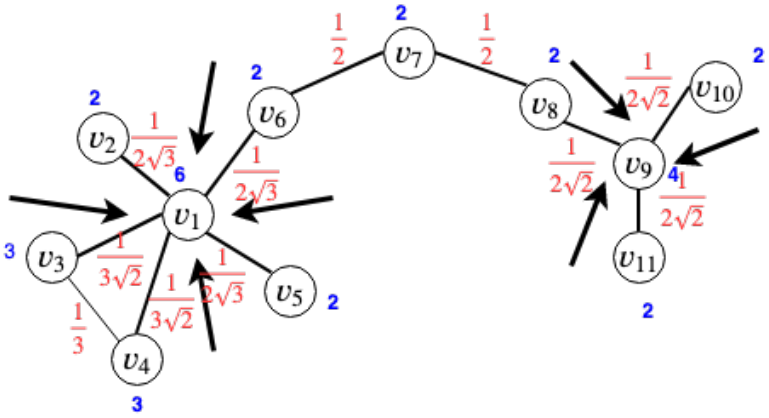
二、1stChebNet算法介绍及实现

PASA Lab

1. 算法介绍

- 关键矩阵
  1. $\tilde{A} = A + I_N$
  2. $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$
  3. $\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$

$$a_{ij} = \frac{1}{\sqrt{d_i}\sqrt{d_j}}$$

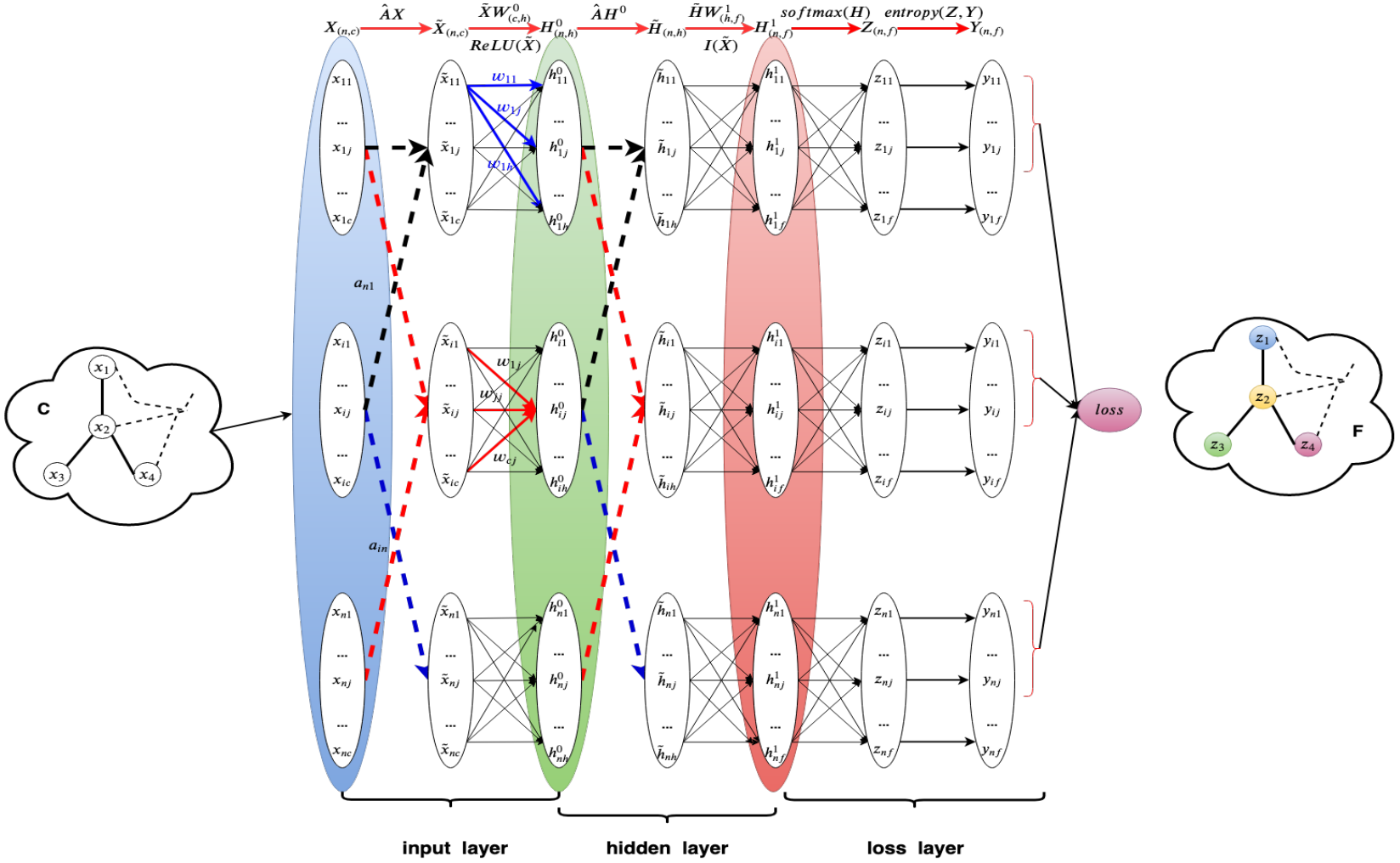


- 解释
  - A是邻接矩阵（对称，要求无向图）
  - 第1步，添加自身的权重
  - 第2步，统计每个节点的度数
  - 第3步，考虑边
- 实际意义
  - 减少度数多的节点对自己的影响

二、1stChebNet算法介绍及实现

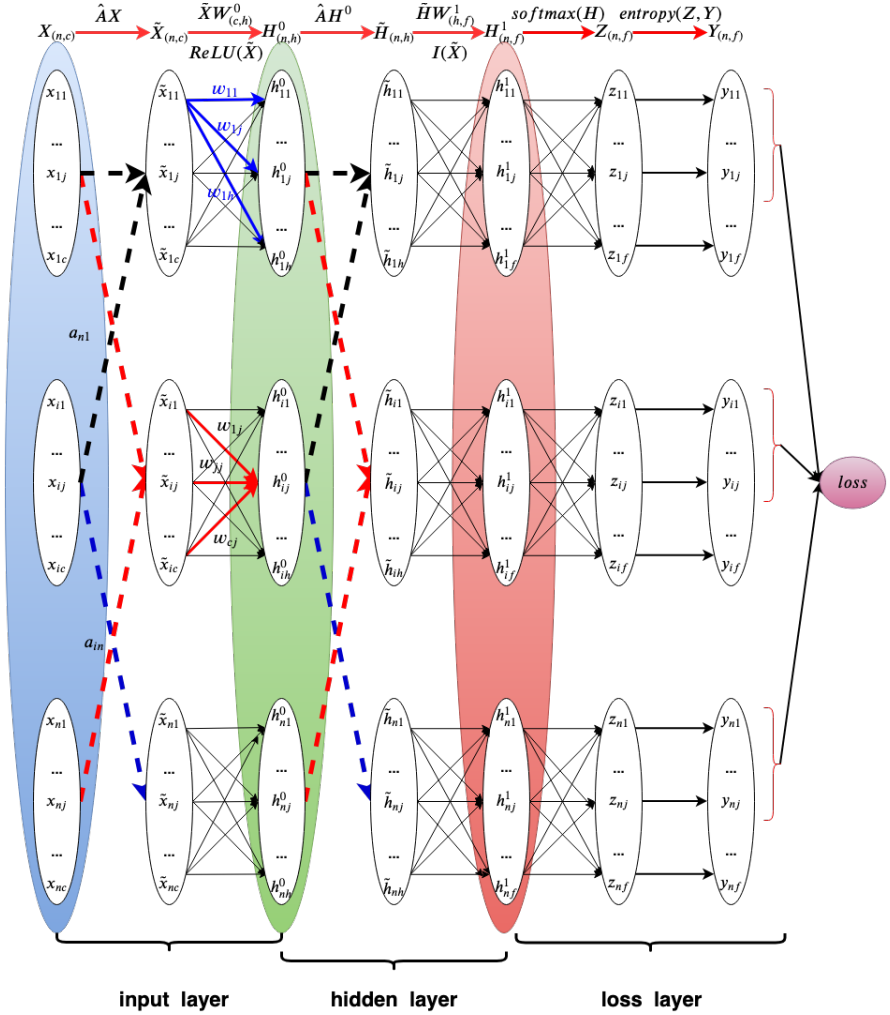
PASA Lab

2. 算法实现——前向传播(整个过程)

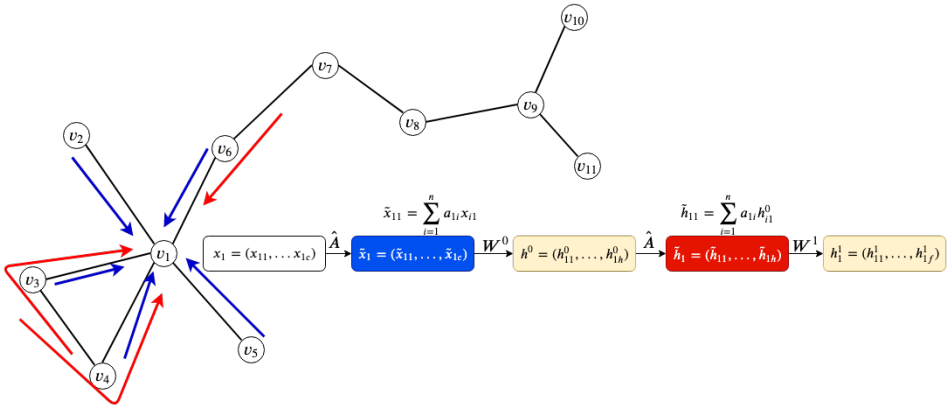


二、1stChebNet算法介绍及实现

2. 算法实现——前向传播(图的角度)

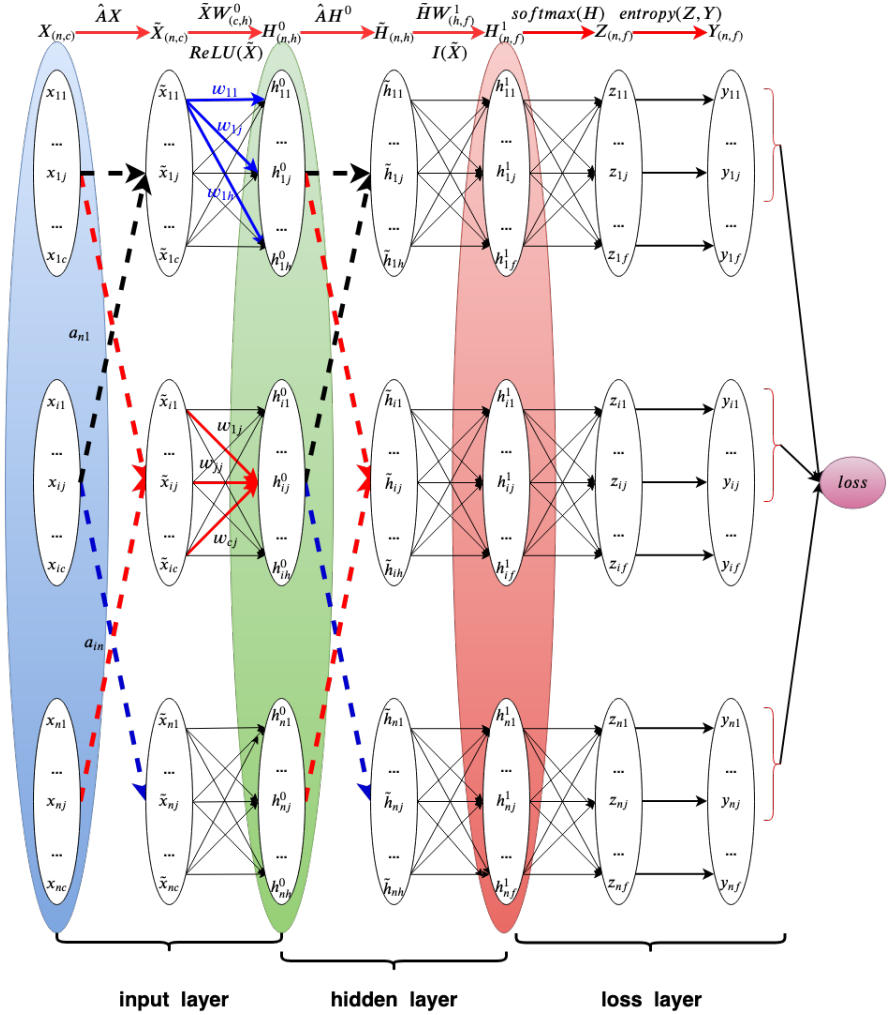


- 从实际图的角度看，实际上几层网络结构，就是当前节点收集几跳邻居节点的信息



二、1stChebNet算法介绍及实现

2. 算法实现——前向传播（矩阵角度）



input layer

$$\tilde{X}_{(n,c)} = \hat{A}_{(n,n)} \ X_{(n,c)}$$

$$H^0_{(n,h)} = ReLU(\tilde{X}_{(n,c)} \ W^0_{(c,h)})$$

hidden layer

$$\tilde{H}_{(n,h)} = \hat{A}_{(n,n)} \ H^0_{(n,h)}$$

$$H^1_{(n,f)} = \tilde{H}_{(n,h)} \ W^1_{(h,f)}$$

loss layer

$$Z_{(n,f)} = softmax(H^1_{(n,f)})$$

$$loss = entropy(Z_{(n,f)}, \ Y_{(n,f)})$$

二、1stChebNet算法介绍及实现

PASA Lab

2. 算法实现——反向传播（梯度下降，矩阵角度）

- 对于反向传播，正向传播转换为了矩阵相乘的操作，所以试图从矩阵的角度概括反向传播，经过学习总结如下:
  - 链式法则：连接各个layer
  - 标量对矩阵的求导loss layer
  - 矩阵对矩阵的求导 hidden layer和input layer
- 正确性检验
  - 导数的定义

$$loss = f(Y_{(n,m)})$$

$$Y_{(n,m)} = X_{(n,f)} W_{(f,m)}$$

$$\{\frac{dloss}{dX}\}_{(n,f)} = \{\frac{dloss}{dY}\}_{(n,m)} * W^T_{(m,f)}$$

$$\{\frac{dloss}{dW}\}_{(f,m)} = X^T_{(f,n)} \{\frac{dloss}{dY}\}_{(n,m)}$$

$$f'(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0}$$ [1]:154

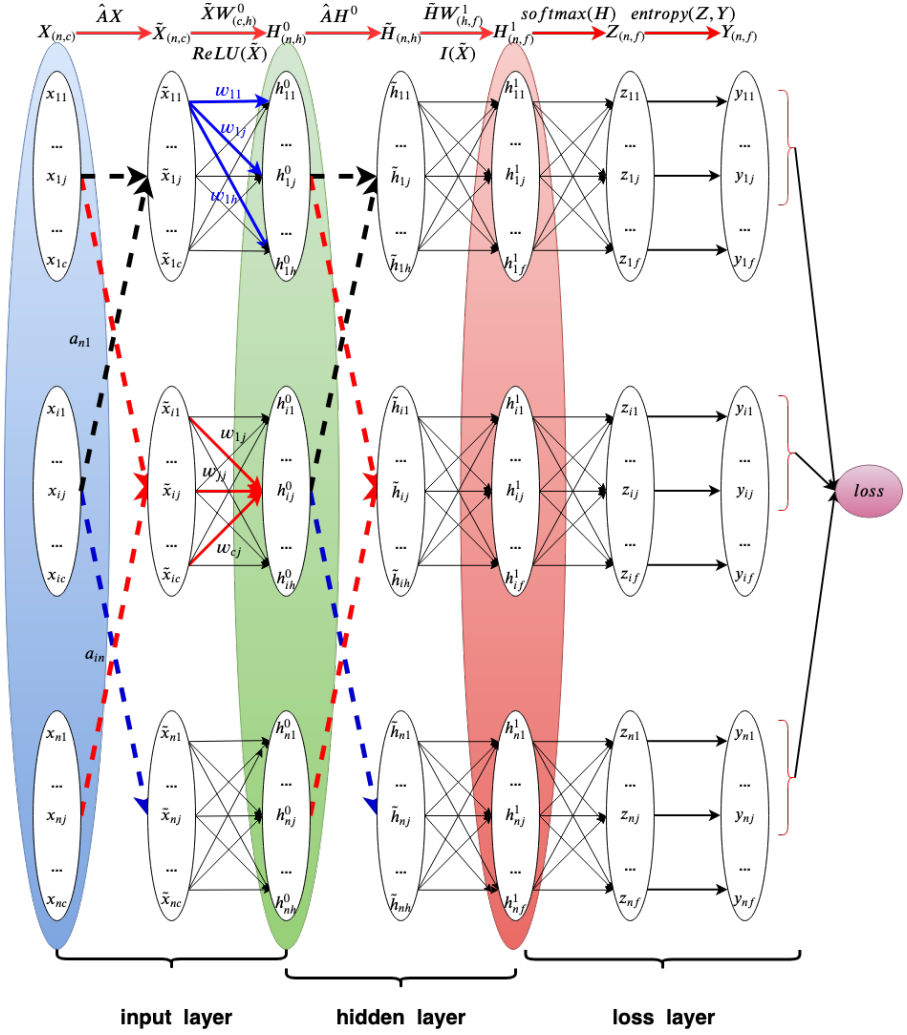
$$\{\frac{dloss}{dZ}\}_{(n,f)} = softmax(Z_{(n,f)}) - Y_{(n,f)}$$

$$\{\frac{dloss}{dW^1}\}_{(h,f)} = \tilde{H}^T_{(h,n)} * (softmax(Z_{(n,f)}) - Y_{(n,f)})$$

$$\{\frac{dloss}{dW^0}\}_{(c,h)} = \tilde{X}^T_{(c,n)} * ((softmax(Z_{(n,f)}) - Y_{(n,f)}) * (W^1)^T_{(f,h)})$$

二、1stChebNet算法介绍及实现

2. 算法实现——后向传播（矩阵角度）



Basic
矩阵求导

$$loss = f(Y_{(n,m)})$$

$$Y_{(n,m)} = X_{(n,f)}\ W_{(f,m)}$$

$$\{\frac{dloss}{dX}\}_{(n,f)} = \{\frac{dloss}{dY}\}_{(n,m)} * W^T_{(m,f)}$$

$$\{\frac{dloss}{dW}\}_{(f,m)} = X^T_{(f,n)}\{\frac{dloss}{dY}\}_{(n,m)}$$

梯度

$$\{\frac{dloss}{dZ}\}_{(n,f)} = softmax(Z_{(n,f)}) - Y_{(n,f)}$$

$$\{\frac{dloss}{dW^1}\}_{(h,f)} = \tilde{H}^T_{(h,n)} * (softmax(Z_{(n,f)}) - Y_{(n,f)})$$

$$\{\frac{dloss}{dW^0}\}_{(c,h)} = \tilde{X}^T_{(c,n)} * ((softmax(Z_{(n,f)}) - Y_{(n,f)}) * (W^1)^T_{(f,h)})$$

梯度更新

$$W^1 = W^1 - \eta\frac{dloss}{dW^1}$$

$$W^0 = W^0 - \eta\frac{dloss}{dW^0}$$

2. 算法实现——其他

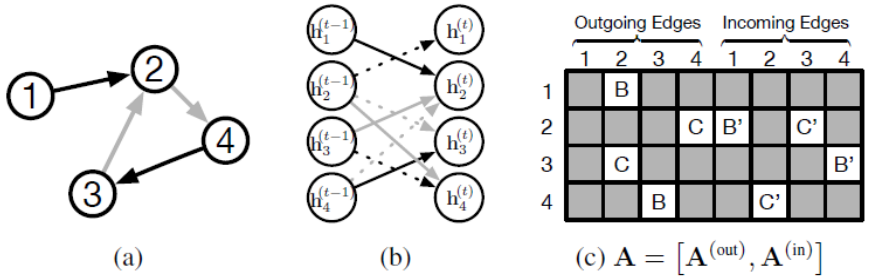
- 基于Numpy重写
  - 优化器Adam
  - 权重矩阵初始化
  - Weight Decay损失
- 结果（未考虑dropout）
  - 因为Numpy数据保存的是float64, Tensorflow是float32, 所以复现时取得了略好的结果

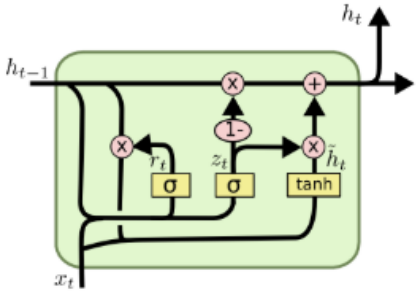
数据集	Paper	Local
Cora	0.80999	0.81000
Citeseer	0.70899	0.71100
Pubmed	0.79400	0.79399

二、1stChebNet算法介绍及实现

3. 进一步思考

- 通过对论文的复现，发现图神经网络都基于以下的模式
  - 在图上进行多步传播信息，反映在隐藏层的个数的设置上
  - 每步做两个操作
    - $T^{t-1} = AH^{t-1}$: 基于图结构的信息，用来自邻居节点的信息更新自己
    - $H^t = T^{t-1}W$: 对每个节点进行一系列复杂的操作
  - 对于GGNNs[1]，该论文属于Graph Convolution Network中Spatial Based的方法
    - A: 右上图, 来自论文[1]
    - W: 图片来源[2]





$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

[1]. Li Y, Tarlow D, Brockschmidt M, et al. Gated graph sequence neural networks[J]. arXiv preprint arXiv:1511.05493, 2015.

[2]. https://www.yunaitong.cn/understanding-lstm-networks.html

二、1stChebNet算法介绍及实现

PASA Lab

3. 进一步思考

- 通过对论文的复现和提取，发现，所谓的图神经网络都基于以下的模式
  - 在图上进行多步传播信息
  - 每步做两个操作
    - $T^{t-1} = AH^{t-1}$：基于图结构的信息，用来自邻居节点的信息更新自己
    - $H^t = T^{t-1}W$: 对每个节点进行一系列复杂的操作
  - 对于GGNNs[1]，可概括为右图：

1. 初始化

$$H^{(0)} = [X_v, P]$$
$$A_v = [A_{in}, A_{out}]$$

2. 图操作

$$T^{(t-1)} = A_v(H^{(t-1)}W^A)$$

3. 节点操作

$$H^{(t)} =$$
$$\delta(T^{(t-1)}W^{z_a}) \odot tanh(T^{(t-1)}W_{h_a}) +$$
$$\delta(T^{(t-1)}W^{z_a}) \odot tanh\{\delta[(T^{t-1)}W^{r_a} \odot H^{(t-1)})W^h]\}+$$
$$\delta(T^{(t-1)}W^{z_a}) \odot tanh\{\delta[(H^{(t-1)}W_r) \odot H^{(t-1)}] + H^{(t-1)}\}+$$
$$tanh(T^{(t-1)}W^{h_a}) \odot \delta(H^{t-1}W^{z_h})+$$
$$tanh\{\delta[(T^{(t-1)}W^{r_a} \odot H^{(t-1)})W^h)]\} \odot \delta(H^{(t-1)}W^{z_a}) +$$
$$\delta(H^{(t-1)}W^{z_h}) \odot tanh\{\delta[(H^{(t-1)}W^{r_h} \odot H^{(t-1)})W^h] - H^{(t-1)}\} + H^{(t-1)}$$

[1]. Li Y, Tarlow D, Brockschmidt M, et al. Gated graph sequence neural networks[J]. arXiv preprint arXiv:1511.05493, 2015.

三、下一步工作

三、下一步工作

PASA Lab

- 2019.08-2019.09（已完成）

  完成GNN调研，并阅读单机式并行化NeuGraph论文

- 2019.09-2019.10（已完成）

  完成Numpy复现Graph Convolution Neural Network中的Semi-GNN

- 2019.10-2019.11（进行中）

  阅读了Communication Neural Network,发现对于每个节点的邻居选择实际上不确定，放弃了该论文；现已阅读了Gated graph sequence neural network论文，正计划进行Numpy复现

- 2019.11-2020.3

  实验NeuGraph论文中的代码，提出想法

谢 谢