Semi-Supervised Classification with Graph Convolutional Networks

Thomas N. Kipf[1] & **Max Welling**[1,2], [1]University of Amsterdam, [2]CIFAR

CIFAR
CANADIAN INSTITUTE FOR ADVANCED RESEARCH

End-to-end learning on graphs

Previous state-of-the-art for **classification on graphs**:

- **Graph kernels/embeddings** [1,2]: not trainable end to end
- **Regularization-based methods** [3-5]:

$$\mathcal{L} = \mathcal{L}_0 + \lambda \mathcal{L}_{\text{reg}}\,, \text{ with } \mathcal{L}_{\text{reg}} = \sum_{i,j} A_{ij} \|f(X_i) - f(X_j)\|^2$$

A: adjacency matrix of graph
X: matrix of node feature vectors

Our approach:

Learn graph-based model $f(X, A)$ end to end.

Fast graph convolutions

Graph convolution with signal $x \in \mathbb{R}^N$ and filter g_θ:

$$g_\theta \star x = U g_\theta(\Lambda) U^\top x$$

U: eigenvectors of graph Laplacian $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$

Comp. complexity: $\mathcal{O}(N^2)$

First-order approximation:

$$g_\theta \star x \approx \theta_0 x + \theta_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x$$

Single parameter and **renormalization***:

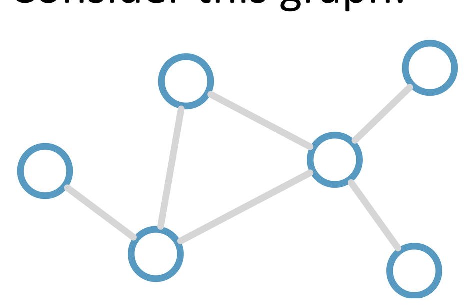
$$\boxed{g_\theta \star x \approx \theta \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} x}$$

New complexity: $\mathcal{O}(|\mathcal{E}|)$

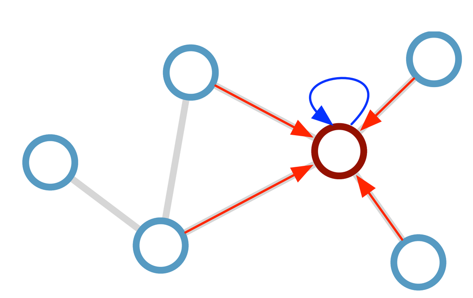
$^*\tilde{A} = A + I_N\,,\ \tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$

Message passing interpretation

Consider this graph:



Update node in red:

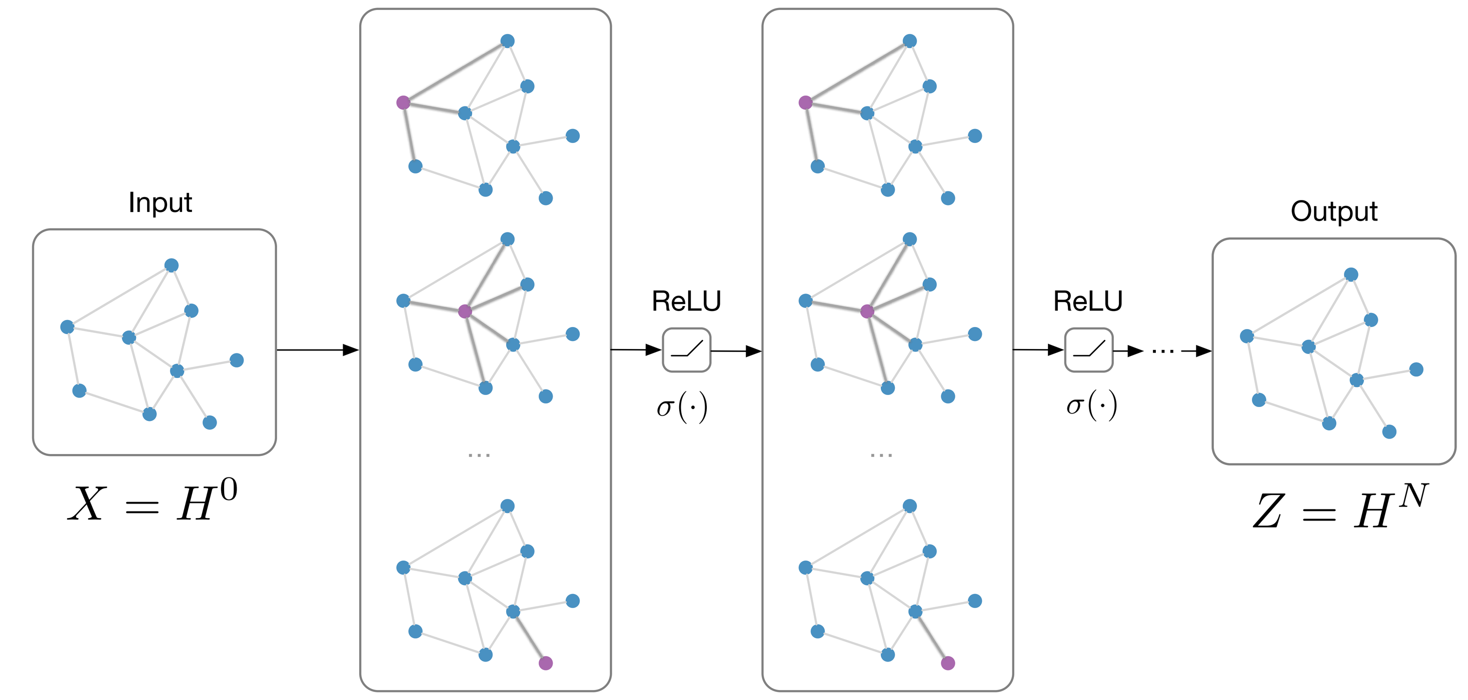


Update rule:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{h}_i^{(l)} \mathbf{W}_0^{(l)} + \sum_{j \in \mathcal{N}_i} \frac{1}{c_{ij}} \mathbf{h}_j^{(l)} \mathbf{W}_1^{(l)}\right)$$

$\mathcal{N}_i$: neighbor indices
c_{ij}: norm. constant (per edge)

Graph convolutional networks



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

Two-layer model: $Z = f(X, A) = \text{softmax}\Big(\hat{A}\ \text{ReLU}\Big(\hat{A} X W^0\Big)\, W^1\Big)$

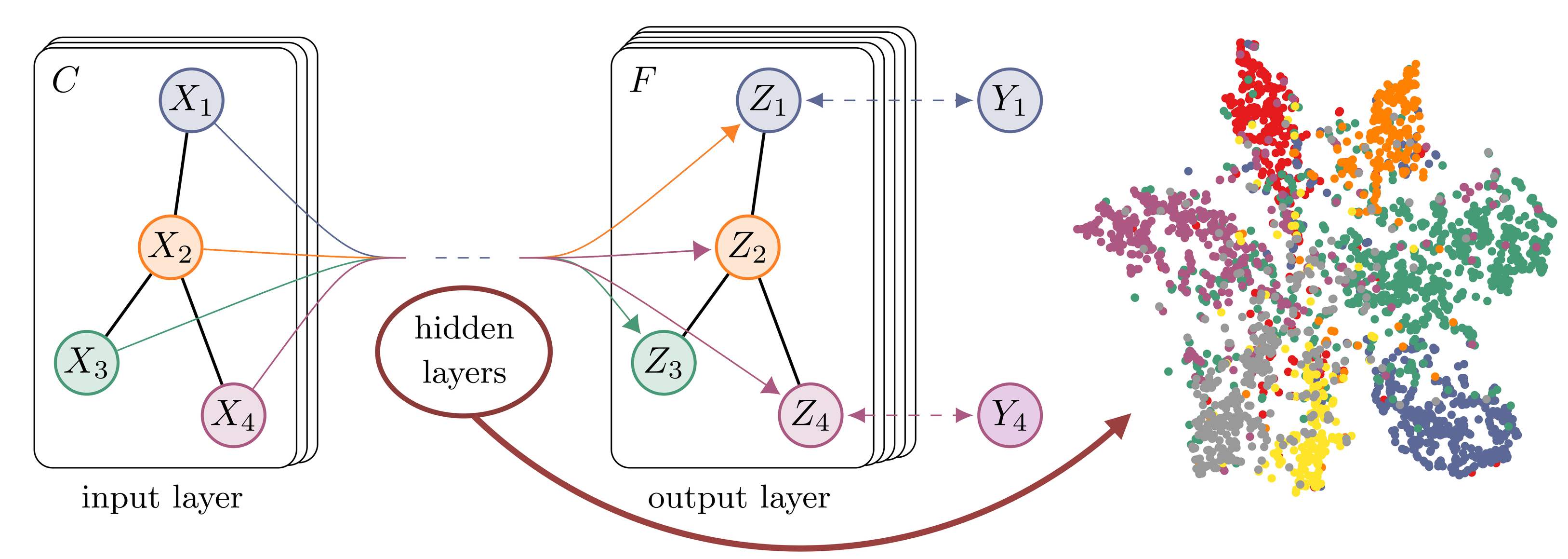
Experiments

Goal: Classify *all nodes* given only a few labels (test set: 1000 nodes)

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

Graph dataset statistics

Graph embeddings



t-SNE embedding of hidden layer activations for Cora dataset (5% labels)

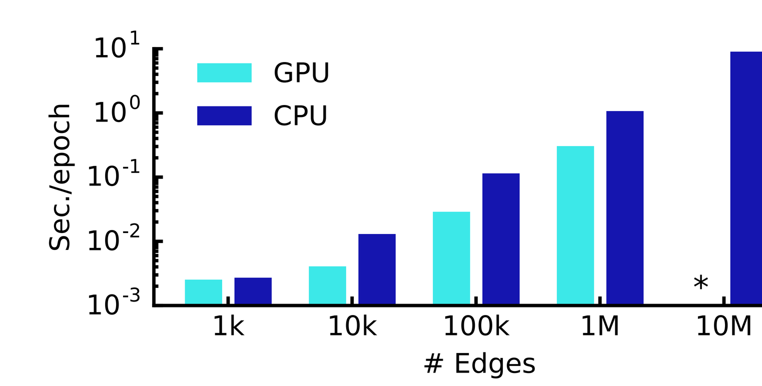
Node classification results

Method	Citeseer	Cora	Pubmed	NELL
ManiReg [4]	60.1	59.5	70.7	21.8
SemiEmb [5]	59.6	59.0	71.1	26.7
LP [3]	45.3	68.0	63.0	26.5
DeepWalk [1]	43.2	67.2	65.3	58.1
Planetoid* [2]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (ours)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

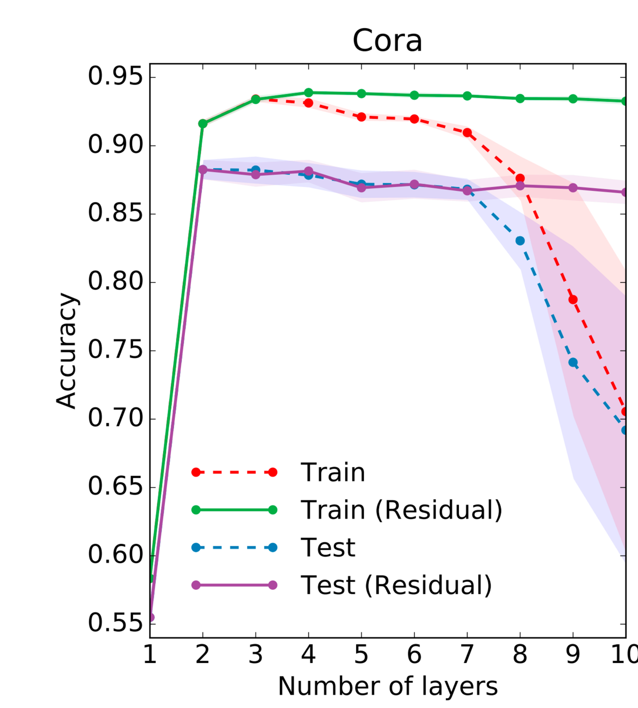
Model comparison (classification accuracy)

Description		Propagation model	Citeseer	Cora	Pubmed
Chebyshev filter [6]	$K = 3$	$\sum_{k=0}^{K} T_k(\tilde{L}) X \Theta_k$	69.8	79.5	74.4
	$K = 2$		69.6	81.2	73.8
1st-order model		$X\Theta_0 + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} X \Theta_1$	68.3	80.0	77.5
Single parameter		$(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) X \Theta$	69.3	79.2	77.4
Renormalization trick		$\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta$	**70.3**	**81.5**	**79.0**
1st-order term only		$D^{-\frac{1}{2}} A D^{-\frac{1}{2}} X \Theta$	68.7	80.5	77.8
Multi-layer perceptron		$X\Theta$	46.5	55.1	71.4

Comparison of graph propagation models



Wall-clock time per training epoch for random graphs



Performance vs. model depth

Conclusions

- End-to-end learning for node classification on graphs
- Scalable up to ~10M nodes/edges
- **Future work**: model variants (e.g. attention mechanism), edge features, scalability, unsupervised learning

Code: github.com/tkipf/gcn
Paper: arxiv.org/abs/1609.02907
Blog: tkipf.github.io/graph-convolutional-networks

Project funded by SAP

References:
[1] Perozzi et al., Deepwalk: Online learning of social representations, SIGKDD (2014)
[2] Yang et al., Revisiting semi-supervised learning with graph embeddings, ICML (2016)
[3] Zhou et al., Learning with local and global consistency, NIPS (2004)
[4] Belkin et al., Manifold regularization, JMLR (2006)
[5] Weston et al., Deep learning via semi-supervised embedding, Neural Networks: Tricks of the Trade (2012)
[6] Defferrard et al., Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, NIPS (2016)

Acknowledgements: Christos Louizos, Taco Cohen, Rianne van den Berg, Karen Ullrich, Joan Bruna, Zhilin Yang, Dave Herman, Pramod Sinha, and Abdul-Saboor Sheikh

Web: tkipf.github.io — staff.fnwi.uva.nl/m.welling
Mail: {t.n.kipf, m.welling}@uva.nl