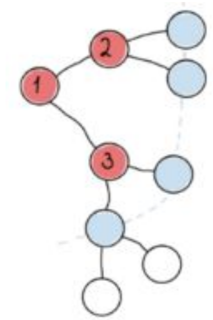
LanczosNet

Kaloshin Pavel
Kolos Maria
Melentev Nikita
Maame Gyamfua
Marina Pominova

https://github.com/mvkolos/LanczosNet

Problem

Graph classification

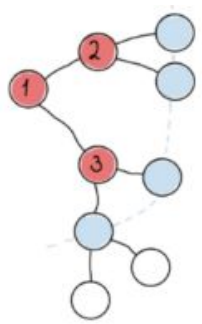


CORA	**CiteSeer**	**Pubmed**
2708 scientific publications, **7** classes.	Another scholarly dataset, **3327** publications, **6** classes	Subset of **19717** citations for biomedical literature, **3** classes

Background

Graph convolution

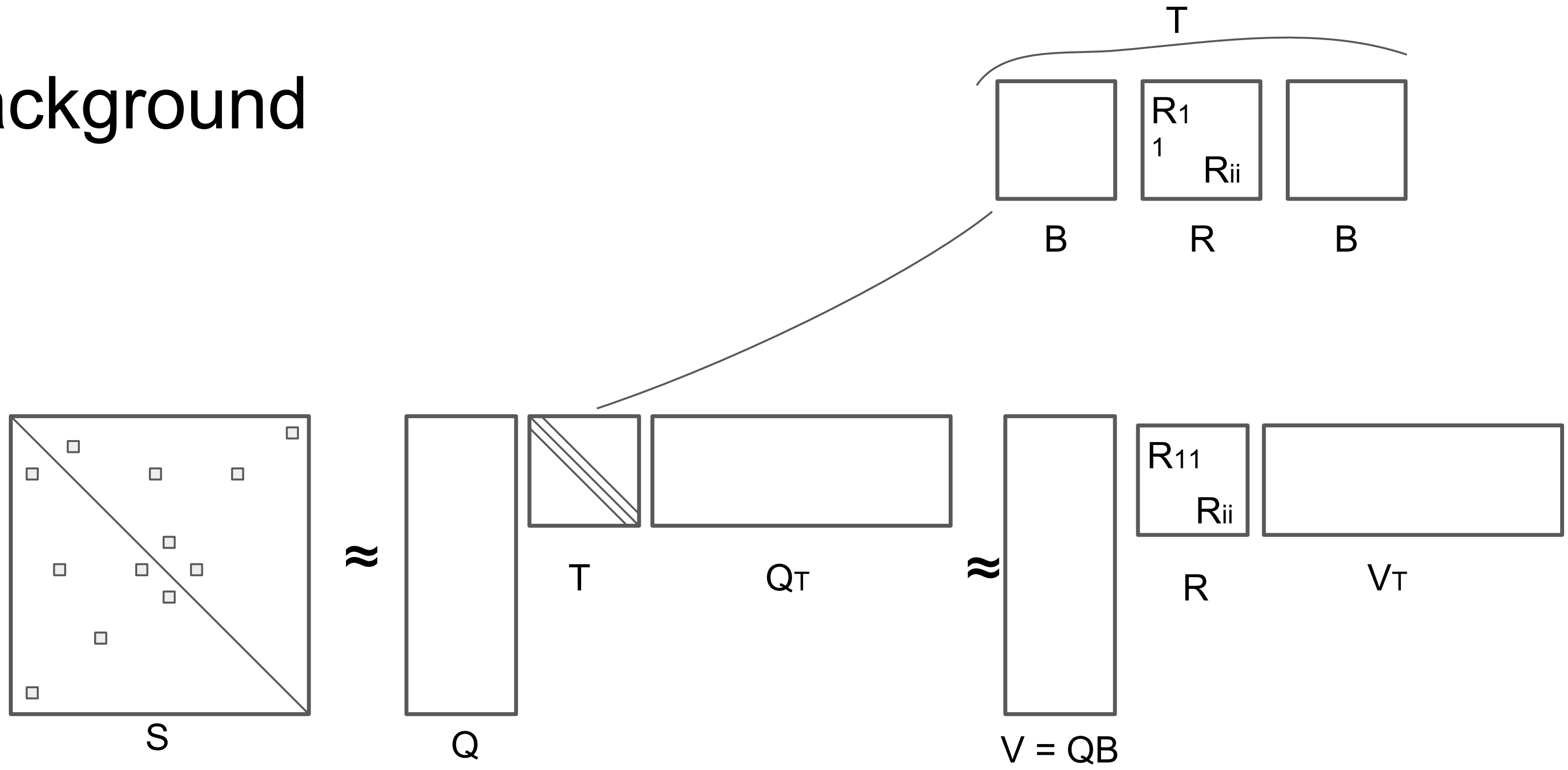
$$Y = \sigma(SXW), \; S = D^{-\frac{1}{2}}(I + A)D^{-\frac{1}{2}}$$



Graph filter

$$Y = \sigma(g(S^0, S^1, S^2, ..., S^k)XW)$$

Background



$$Y = \sigma(g(S^0, S^1, S^2, ..., S^k)XW)$$

Key Ideas

1. Learnable filters
2. Low-cost and general multi scale features
3. Exponent kernel for learnable embeddings

$$Y_j = [X_i, SX_i, \ldots, S^{K-1}X_i]\boldsymbol{w}_{i,j} \approx [X_i, VRV^\top X_i, \ldots, VR^{K-1}V^\top X_i]\boldsymbol{w}_{i,j}$$

$$Y_j = [X_i, V\hat{R}(1)V^\top X_i, \ldots, V\hat{R}(K-1)V^\top X_i]\boldsymbol{w}_{i,j}.$$

Key Ideas

1. Learnable filters
2. Low-cost multi scale features
3. Exponent kernel for learnable embeddings

low-rank

$$Y = \left[L^{\mathcal{S}_1} X, \ldots, L^{\mathcal{S}_M} X, V \hat{R}(\mathcal{I}_1) V^\top X, \ldots, V \hat{R}(\mathcal{I}_N) V^\top X \right] W$$

diagonal

Key Ideas

1. Learnable filters
2. Low-cost multi scale features
3. Exponent kernel for learnable embeddings

MLP

$$k(x_i, x_j) = \exp\left(-\frac{\|(f_\theta(x_i) - f_\theta(x_j))\|^2}{\epsilon}\right)$$

scale

The experiments

- LanczosNet. Paper re-implementation + architecture experiments
- AdaLanczosNet. Learning node embeddings
- LanczosNet. Multihead attention

LanczosNet. Paper re-implementation + architecture experiments

Long scale filters

nn.SpectralConv $$Y = \left[S^{\mathcal{S}_1}X, \ldots, S^{\mathcal{S}_M}X, Qf_1\left(T^{\mathcal{I}_1}\right)Q^\top X, \ldots, Qf_N\left(T^{\mathcal{I}_N}\right)Q^\top X\right]W,$$

Features

Affinity/Laplacian

Weight parameter

nn.GraphConv $$Y = \sigma(SXW), \; S = D^{-\frac{1}{2}}(I + A)D^{-\frac{1}{2}}$$

LanczosNet. Paper re-implementation + architecture experiments

Short scale filters

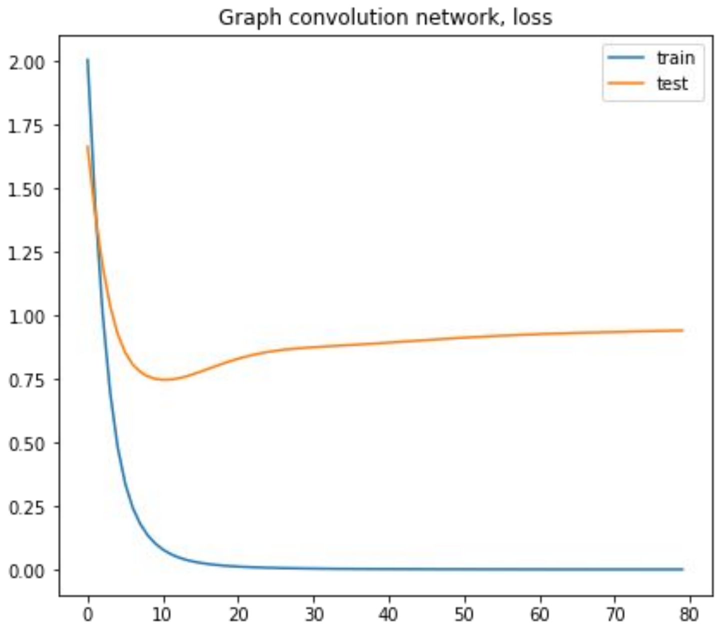
Long scale filters

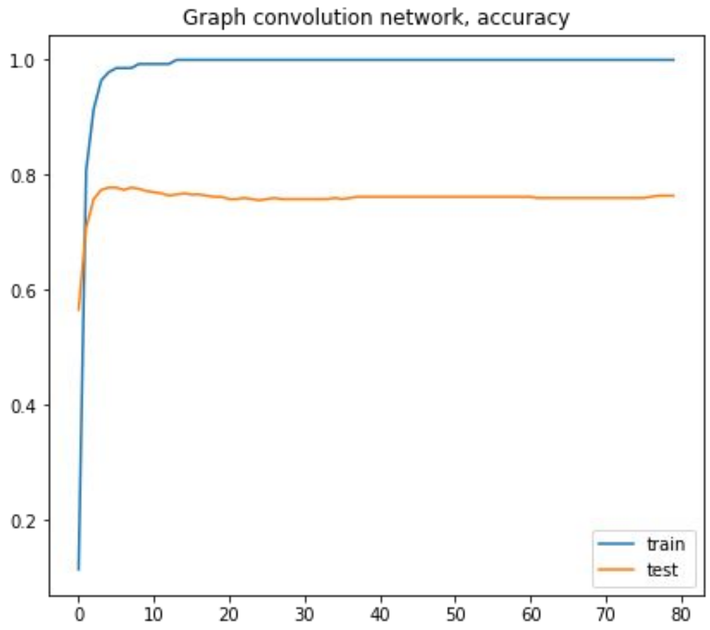
nn.SpectralConv

$$Y = \left[\underbrace{S^{S_1}X, \ldots}, \underbrace{, S^{S_M}X, Qf_1\left(T^{\mathcal{I}_1}\right)Q^\top X, \ldots, Qf_N\left(T^{\mathcal{I}_N}\right)Q^\top X}\right]W,$$

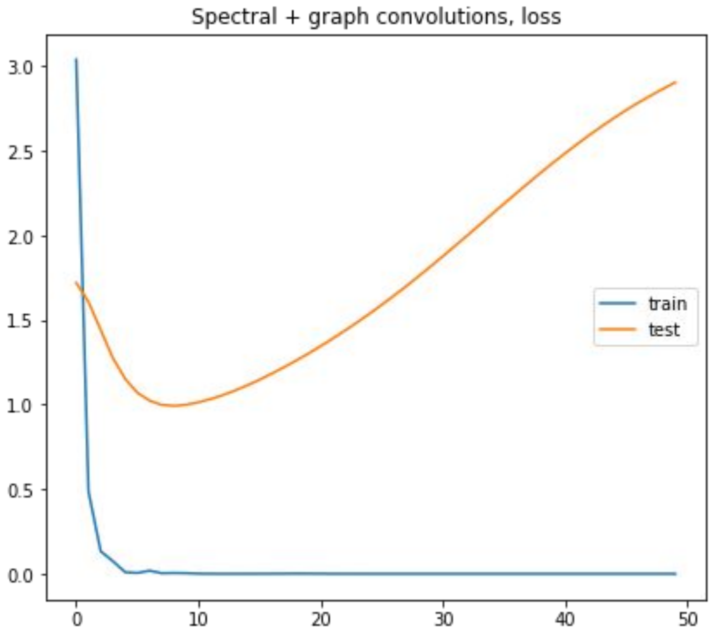
Setting	**Vanilla graph convolutions**	**Spectral convolutions**	**Spectral + MLP**	**Spectral + vanilla graph (+ Dropout + BatchNorm)**	**The paper**
Results	CORA: 78 PubMed: 76 CiteSeer: 61	CORA: 74 PubMed: 45 CiteSeer: 66	CORA: 76 PubMed: out of memory CiteSeer: 64	CORA: 78 PubMed: 76 CiteSeer: 65	CORA: 79 PubMed: 78 CiteSeer: 66
Insights	Still good	Overfit badly	Still overfit badly	Best performance	How?

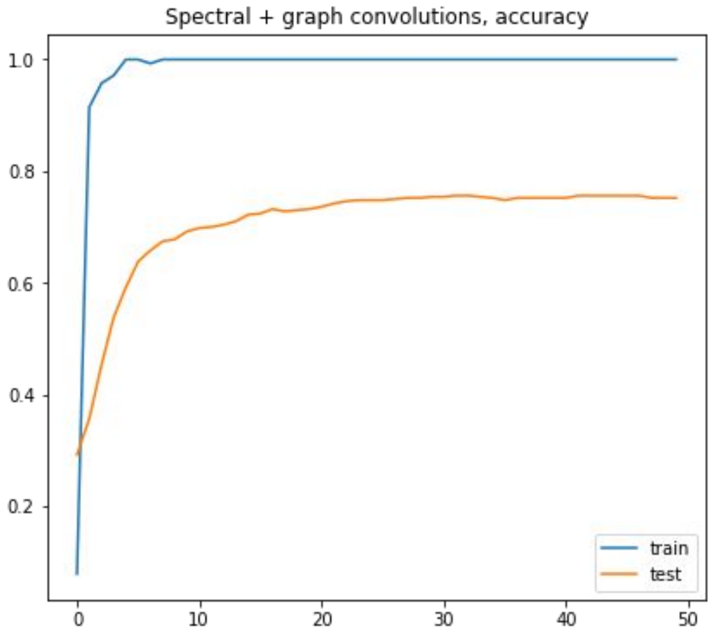
Visualization



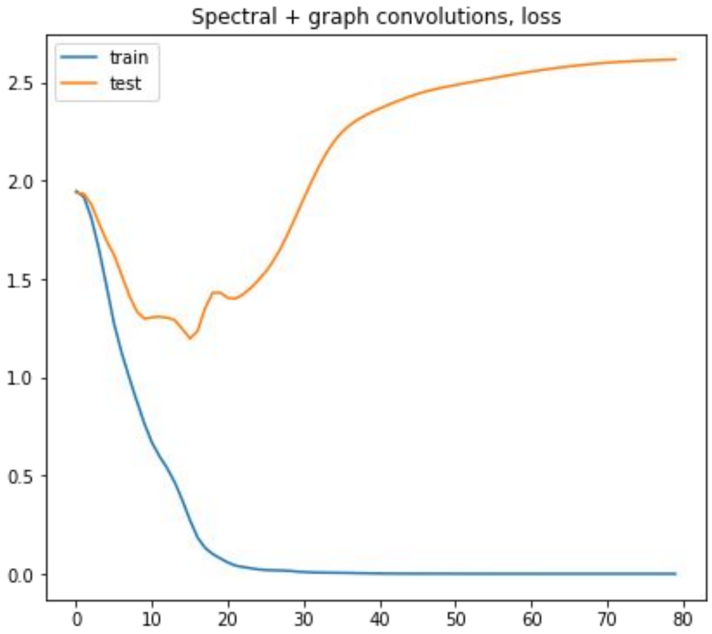


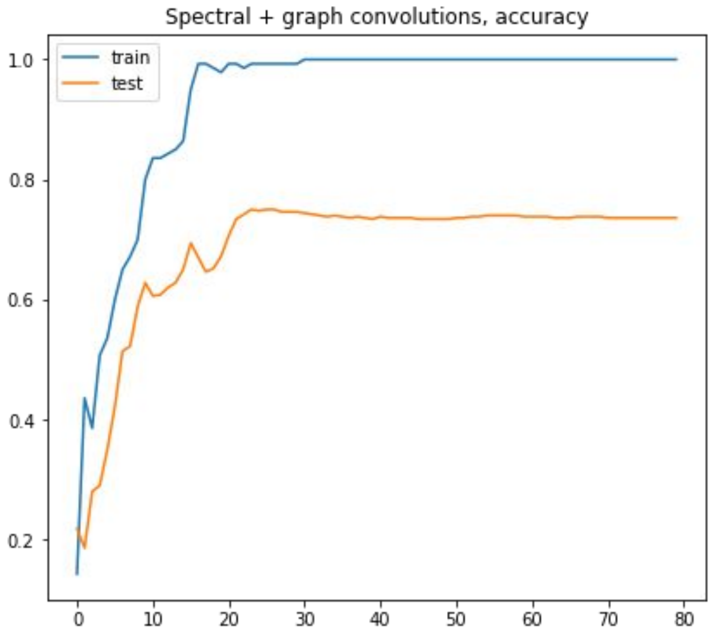
Visualization





Visualization





LanczosNet. Multihead attention

Graph attention

capturing graph structure over nodes in the neighbourhood

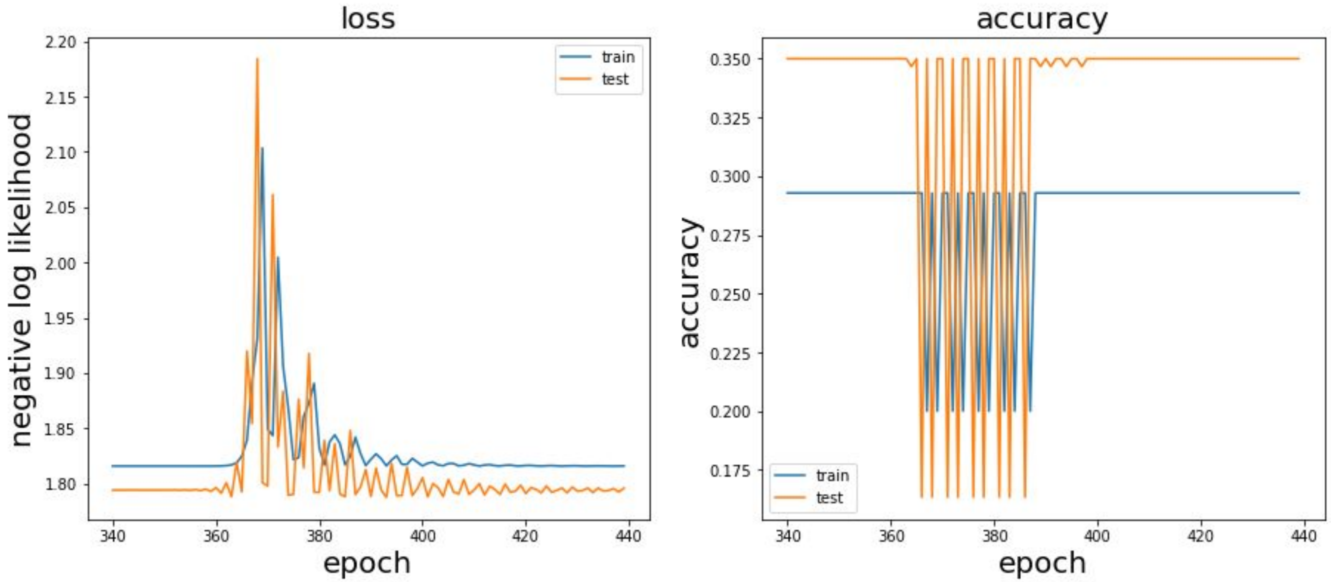
$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}$$

$$e_{ij} = a(\vec{h}_i, \vec{h}_j)$$

a - attention,
e$_{ij}$ - feature-based coefficients for nodes pairs

Setting	**SpectralConv + GraphConv**	**SpectralConv + GraphConv + Attention**	**Attention**
Results	CORA: 78 CiteSeer: 65	CORA: 82 CiteSeer: 40	CORA: 80,5 CiteSeer: 68,1
Insights	Fast	Really slow & memory intensive, almost no improvement	

AdaLanczosNet. Learning node embeddings




```
40%|████      | 4/10 [00:00<00:00, 31.91it/s]

Epoch:  0341
 loss_train:    1.8159
 acc_train:     0.2929
 loss_val:      1.7959
 acc_val:       0.3500
```

Results summary

- reproduced results from the paper for citation networks
- investigated key ideas: learnable filters, large scale features, kernel
- discovered problems with overfitting and embedding learning

https://github.com/mvkolos/LanczosNet/