PEKING UNIVERSITY

Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering

Michaël Defferrard, Xavier Bresson, Pierre Vandergheynst
EPFL, Lausanne, Switzerland

Zhang Chi, Yan Tingyun, Chen Guangyao

2019/10/17

Outline

- Introduction
- Learning Fast Localized Spectral Filters
- Graph Coarsening and Pooling
- Numerical Experiments
- Consultions

Introduction

PEKING UNIVERSITY

- CNNs are good at learning local stationary structures and compose them to form multi-scale hierarchical patterns(only defined for regular grids)

- Extend CNN to graphs

  How to define:

  - localized graph filters
  - Pooling operations

  On non-Euclidean domains

图的傅里叶变换

北京大学 PEKING UNIVERSITY

图的定义： $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$

- V是顶点集，E是边集，W是顶点边权的邻接矩阵。
- X是输入的信号，可视作一维向量。

拉普拉斯矩阵及性质：

普通形式 $L = D - A$　　对称归一化 $L^{sys} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2}$

- 半正定实对称矩阵：特征向量相互正交，可构成正交矩阵U作为GFT的基。
- 特征值非负，最小特征值是0，特征值可作为图的频率，越小的特征值对应越低频的信息。
- 拉普拉斯矩阵利用傅里叶的基U通过$L=U\Lambda U^T$对角化。

图的傅里叶变换形式

$$F(\lambda_l) = \hat{f}(\lambda_l) = \sum_{i=1}^{N} f(i) u_l(i)$$

$$f(i) = \sum_{l=1}^{N} \hat{f}(\lambda_l) u_l(i)$$

$$\begin{pmatrix} \hat{f}(\lambda_1) \\ \hat{f}(\lambda_2) \\ \vdots \\ \hat{f}(\lambda_N) \end{pmatrix} = \begin{pmatrix} u_1(1) & u_1(2) & \dots & u_1(N) \\ u_2(1) & u_2(2) & \dots & u_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ u_N(1) & u_N(2) & \dots & u_N(N) \end{pmatrix} \begin{pmatrix} f(1) \\ f(2) \\ \vdots \\ f(N) \end{pmatrix}$$

$$\begin{pmatrix} f(1) \\ f(2) \\ \vdots \\ f(N) \end{pmatrix} = \begin{pmatrix} u_1(1) & u_2(1) & \dots & u_N(1) \\ u_1(2) & u_1(2) & \dots & u_N(2) \\ \vdots & \vdots & \ddots & \vdots \\ u_1(N) & u_2(N) & \dots & u_N(N) \end{pmatrix} \begin{pmatrix} \hat{f}(\lambda_1) \\ \hat{f}(\lambda_2) \\ \vdots \\ \hat{f}(\lambda_N) \end{pmatrix}$$

f在Graph上傅里叶变换的矩阵形式为：

$$\hat{f} = U^T f$$

f 在Graph上傅里叶逆变换的矩阵形式为：

$$f = U\hat{f}$$

图信号的谱滤波

图的卷积定义：

$$(f * g)_G = U((U^T g) \odot (U^T f))$$

将$U^T g$整体看作可学习的卷积核，这里可写作$\mathbf{g}_\theta$：

$$(f * g)_G = U g_\theta U^T f$$

图谱卷积网络的关键就在于$\mathbf{g}_\theta$的选择

谱CNN

$$X_{:,j}^{k+1} = \sigma\left(\sum_{i=1}^{f_{k-1}} U\Theta_{i,j}^{k} U^{T} X_{:,i}^{k}\right) \qquad (j = 1, 2, \cdots, f_k)$$

- $X^k \in \mathbb{R}^{N \times f_{k-1}}$是输入图信号,对应图上就是点的输入特征
- N是节点数量
- f_{k-1}是输入通道的数量
- f_k是输出通道的数量
- $\Theta_{i,j}^{k}$是一个可学习参数的对角矩阵,就跟三层神经网络中的weight一样是任意的参数，通过初始化赋值然后利用误差反向传播进行调整
- $\sigma(\cdot)$是激活函数

缺点

☹ Filters are basis-dependent ⇒ does not generalize across graphs

☹ Only undirected graphs (symmetric Laplacian matrix required for orthogonal eigendecomposition)

☹ $\mathcal{O}(n)$ parameters per layer

☹ $\mathcal{O}(n^2)$ computation of forward / inverse Fourier transforms $\mathbf{\Phi}^{\top}$, $\mathbf{\Phi}$

☹ No guarantee of spatial localization of filters

J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” in Proceedings of International Conference on Learning.

基于多项式逼近的谱CNN

北京大学 PEKING UNIVERSITY

为解决谱CNN中滤波器无法定位到局部信息以及学习复杂度为O(n)的缺陷，可以用多项式来做拟合：

$$g_\theta * x = U g_\theta U^T x \qquad g_\theta = g_\theta(\Lambda) \qquad g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k$$

$d_{\mathcal{G}}(i,j) > K$ implies $(L^K)_{i,j} = 0$, where $d_{\mathcal{G}}$ is the shortest path distance.

保证了K-localized，以及将参数复杂度降到了O(k)

J. Bruna等人提出用B样条曲线来对上式做逼近

$$g_\theta(\Lambda) = B\theta$$

$B \in \mathbb{R}^{n \times K}$ is the cubic B-spline basis

parameter $\theta \in \mathbb{R}^K$ is a vector of control points

由于U是对称归一化的拉普拉斯矩阵，计算复杂度仍然是O(n^2)。

切比雪夫多项式逼近

为了解决这个问题，Hammond et al.(2011)通过Chebyshev多项式Tk(x)的Kth-阶截断展开来拟合gθ(Λ)

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda})$$

- $\tilde{\Lambda} = 2\Lambda/\lambda_{max} - I_N$(为缩放后的特征向量矩阵,缩放后范围是[−1,1]，单位矩阵的特征值是n重1)，缩放的目的是为了满足Chebyshev多项式$T_k(x)$ 的K^{th} 阶截断展开的条件：自变量范围需要在[−1,1]之间
- λ_{max}是L 的最大特征值，也叫**谱半径。**
- $\theta \in \mathbb{R}^K$ 是切比雪夫系数的向量
- Chebyshev多项式递归定义为$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$，其中$T_0(x) = 1$，$T_1(x) = x$ 。

Chebyshev谱CNN

北京大学 PEKING UNIVERSITY

如何解决计算复杂度的问题?

$$
\begin{aligned}
g_\theta * x &= U g_\theta U^T x \\
&= U g_\theta(\Lambda) U^T x \\
&= U\left(\sum_{k=0}^{K} \theta_k T_K(\tilde{\Lambda})\right) U^T x \\
&= \left(\sum_{k=0}^{K} \theta_k T_K(U \tilde{\Lambda} U^T)\right) x \\
&= \sum_{k=0}^{K} \theta_k T_K(\tilde{L}) x \qquad (5)
\end{aligned}
$$

$\Rightarrow$

利用数学归纳法证明

$$U T_k(\tilde{\Lambda}) U^T = T_k(U \tilde{\Lambda} U^T)$$

Chebyshev谱CNN

$$UT_k(\tilde{\Lambda})U^T = T_k(U\tilde{\Lambda}U^T)$$

Chebyshev多项式递归定义为$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$，其中$T_0(x) = 1$，$T_1(x) = x$。

当n=1时显然成立

$$UT_0(\tilde{\Lambda})U^T = UU^T = 1 = T_0(U\tilde{\Lambda}U^T)$$
$$UT_1(\tilde{\Lambda})U^T = U\tilde{\Lambda}U^T = T_1(U\tilde{\Lambda}U^T)$$

假设n=k时成立

$$UT_{k-2}(\tilde{\Lambda})U^T = T_{k-2}(U\tilde{\Lambda}U^T)$$
$$UT_{k-1}(\tilde{\Lambda})U^T = T_{k-1}(U\tilde{\Lambda}U^T)$$

证明n=k+1时成立

$$\begin{aligned}
UT_k(\tilde{\Lambda})U^T &= 2U\tilde{\Lambda}T_{k-1}(\tilde{\Lambda})U^T - UT_{k-1}(\tilde{\Lambda})U^T \\
&= 2(U\tilde{\Lambda}U^T)\left[UT_{k-1}(\tilde{\Lambda})U^T\right] - UT_{k-1}(\tilde{\Lambda})U^T \\
&= 2(U\tilde{\Lambda}U^T)T_{k-1}(U\tilde{\Lambda}U^T) - T_{k-1}(U\tilde{\Lambda}U^T) \\
&= T_k(U\tilde{\Lambda}U^T)
\end{aligned}$$

Chebyshev谱CNN

如何解决计算复杂度的问题?

$$
\begin{aligned}
g_\theta * x &= U g_\theta U^T x \\
&= U g_\theta(\Lambda) U^T x \\
&= U(\sum_{k=0}^{K} \theta_k T_K(\tilde{\Lambda}))U^T x \\
&= (\sum_{k=0}^{K} \theta_k T_K(U\tilde{\Lambda}U^T))x \\
&= \sum_{k=0}^{K} \theta_k T_K(\tilde{L})x \qquad (5)
\end{aligned}
$$

利用数学归纳法证明

$$UT_k(\tilde{\Lambda})U^T = T_k(U\tilde{\Lambda}U^T)$$

整个运算的复杂度是O(K|E|)，即与边数E呈线性关系。当graph是稀疏图的时候，计算加速尤为明显，这个时候复杂度远低于$O(n^2)$。

Chebyshev谱CNN

图卷积第s个样本的第j个输出的特征图为：

$$y_{s,j} = \sum_{i=1}^{F_{in}} g_{\theta_{i,j}}(L) x_{s,i} \in \mathbb{R}^n,$$

$x_{s,j}$表示输入的特征图，θ是切比雪夫系数。

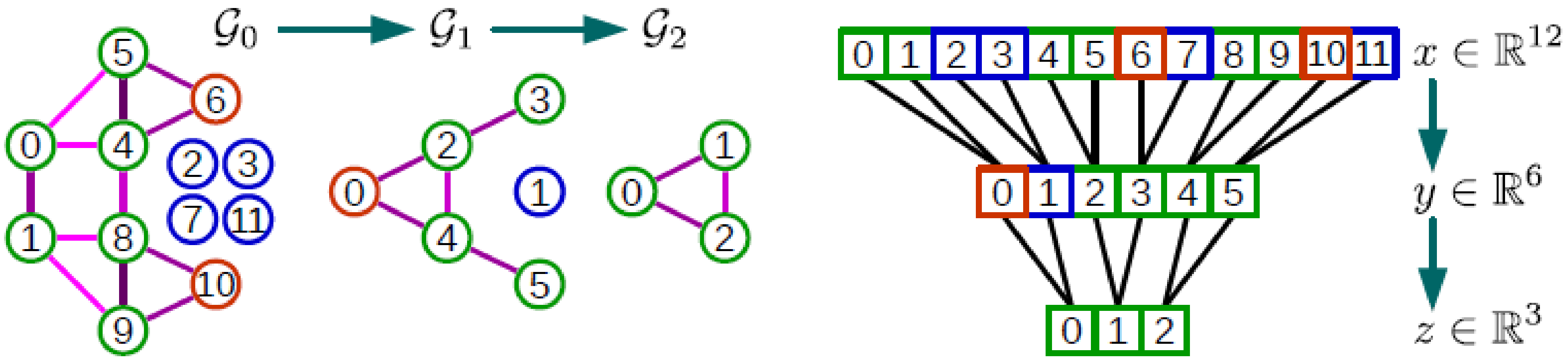
反向传播公式为：E为损失函数

$$\frac{\partial E}{\partial \theta_{i,j}} = \sum_{s=1}^{S} [\bar{x}_{s,i,0}, \ldots, \bar{x}_{s,i,K-1}]^T \frac{\partial E}{\partial y_{s,j}}$$

Graph Coarsening and Pooling

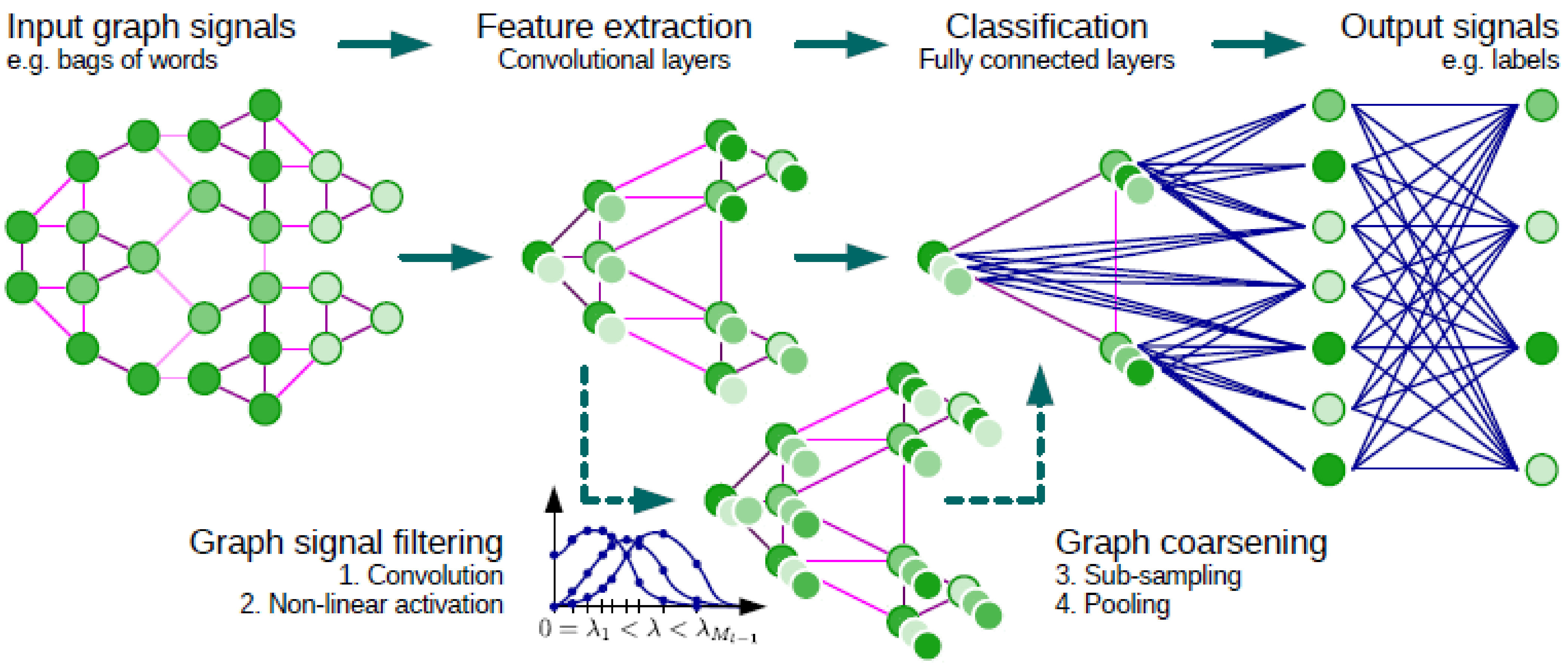
北京大学 PEKING UNIVERSITY

- 池化过程采用Graclus贪心粗化方法
- 蓝色的为随机初始化的假节点，保证黄色的单独节点成对匹配，池化过程成整除。
- 最后产生的池化结果z可以理解为



$$[\max(x_0, x_1), \max(x_4, x_5, x_6), \max(x_8, x_9, x_{10})]$$

Architecture of a CNN on graphs



Revisiting Classical CNNs on MNIST

- 8-NN graph, 976=784+192 fake nodes, $|\varepsilon|$ = 3198 edges
- The weights of a k-NN similarity graph $$W_{ij} = \exp\left(-\frac{\|z_i - z_j\|_2^2}{\sigma^2}\right)$$
- K = 25

Model	Architecture	Accuracy
Classical CNN	C32-P4-C64-P4-FC512	99.33
Proposed graph CNN	GC32-P4-GC64-P4-FC512	99.14

Table 1: Classification accuracies of the proposed graph CNN and a classical CNN on MNIST.

Text Categorization on 20NEWS

- Each document x is represented using the bag-of-words model
- 16-NN graph, z_i is the word2vec embedding $$W_{ij} = \exp\left(-\frac{\|z_i - z_j\|_2^2}{\sigma^2}\right)$$
- $n = 10{,}000, |\varepsilon| = 132{,}834$

Model	Accuracy
Linear SVM	65.90
Multinomial Naive Bayes	68.51
Softmax	66.28
FC2500	64.64
FC2500-FC500	65.76
GC32	68.26

Table 2: Accuracies of the proposed graph CNN and other methods on 20NEWS.

Comparison between Spectral Filters and Computational Efficiency

- Non-Param, Spline, Chebyshev

$$g_\theta(\Lambda) = \mathrm{diag}(\theta), \quad g_\theta(\Lambda) = B\theta, \quad g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda}),$$

		Accuracy		
Dataset	Architecture	Non-Param (2)	Spline (7) [4]	Chebyshev (4)
MNIST	GC10	95.75	97.26	97.48
MNIST	GC32-P4-GC64-P4-FC512	96.28	97.15	99.14

Table 3: Classification accuracies for different types of spectral filters ($K = 25$).

Comparison between Spectral Filters and Computational Efficiency

- Non-Param, Spline, Chebyshev

$$g_\theta(\Lambda) = \mathrm{diag}(\theta), \quad g_\theta(\Lambda) = B\theta, \quad g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda}),$$

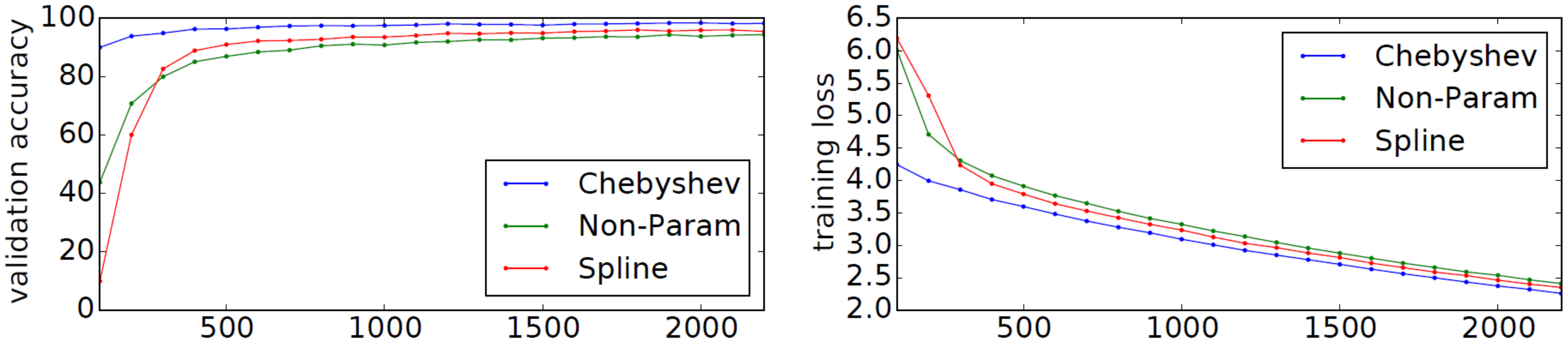


Figure 4: Plots of validation accuracy and training loss for the first 2000 iterations on MNIST.

Comparison between Spectral Filters and Computational Efficiency

- Non-Param, Spline, Chebyshev

$$g_\theta(\Lambda) = \mathrm{diag}(\theta), \quad g_\theta(\Lambda) = B\theta, \quad g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda}),$$

$\boldsymbol{O(n^2)}$ $\boldsymbol{O(n)}$

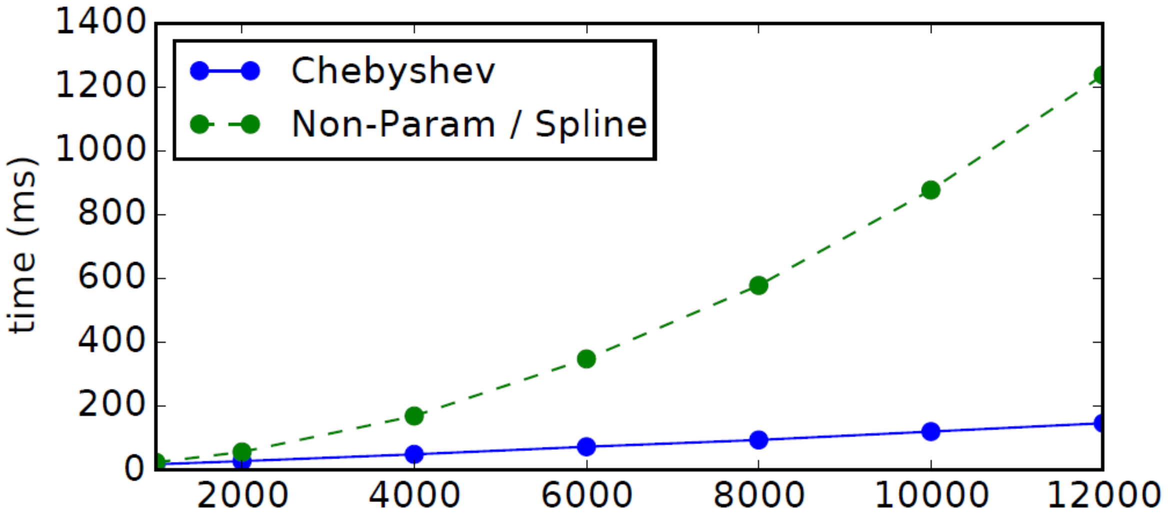


Figure 3: Time to process a mini-batch of $S = 100$ 20NEWS documents w.r.t. the number of words n.

Comparison between Spectral Filters and Computational Efficiency

- Non-Param, Spline, Chebyshev

$$g_\theta(\Lambda) = \mathrm{diag}(\theta), \quad g_\theta(\Lambda) = B\theta, \quad g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda}),$$

Model	Architecture	Time (ms) CPU	GPU	Speedup
Classical CNN	C32-P4-C64-P4-FC512	210	31	6.77x
Proposed graph CNN	GC32-P4-GC64-P4-FC512	1600	200	8.00x

Table 4: Time to process a mini-batch of $S = 100$ MNIST images.

Influence of Graph Quality

Architecture	8-NN on 2D Euclidean grid	random
GC32	97.40	96.88
GC32-P4-GC64-P4-FC512	99.14	95.39

Table 5: Classification accuracies with different graph constructions on MNIST.

	word2vec			
bag-of-words	pre-learned	learned	approximate	random
67.50	66.98	68.26	67.86	67.75

Table 6: Classification accuracies of GC32 with different graph constructions on 20NEWS.

Conclusions

- Benefits
  - Chebyshev谱CNN是K-localized，具有局部连接性。
  - 参数复杂度为$O(K)$
  - 整个运算的复杂度是$O(K|\varepsilon|)$，当graph是稀疏图的时候，计算加速尤为明显，这个时候复杂度远低于$O(n^2)$。

Q & A