

000
001
002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053

Deep Networks on Graph-Structured Data

Anonymous Author(s)

Affiliation
Address
email

Abstract

1 Experiments

In order to measure the performance of spectral networks on real-world data and to explore the effect of the graph estimation procedure, we performed experiments on three datasets from computer vision, text categorization and computational biology. All experiments were done using the Torch machine learning environment with a custom CUDA backend, which is publicly available at URL.

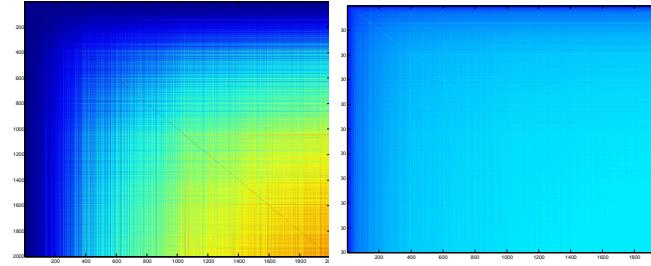
In this section we adopt the following notation to describe network architectures: GCK denotes a graph convolution layer with k feature maps, Pk denotes a graph pooling layer with stride k and pool size $2k$, and FCk denotes a fully connected layer with k hidden units.

1.1 Reuters

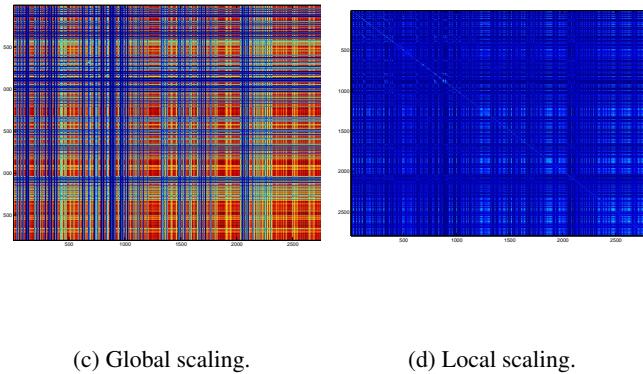
We used the Reuters dataset described in [?], which consists of training and test sets each containing 201,369 documents from 50 mutually exclusive classes. Each document is represented as a log-normalized bag of words for 2000 common non-stop words. As a baseline we used the fully-connected network of [?] with two hidden layers consisting of 2000 and 1000 hidden units regularized with dropout.

We based the spectral network architecture on that of a classical convolutional network, namely by interleaving graph convolution and graph pooling layers and ending with a fully connected layer. Performing pooling at the beginning of the network was especially important to reduce the dimensionality in the graph domain and alleviate the expensive Graph Fourier Transform operation. We chose hyperparameters by performing initial experiments on a validation set consisting of one-tenth of the training data. Specifically, we set the number of subsampled weights to $k = 60$, learning rate to 0.01 and used max pooling rather than average pooling. We also found that using AdaGrad [?] made training faster. All architectures were then trained using the same hyperparameters. Since the experiments were computationally expensive, we did not train all models until full convergence. This enabled us to explore more model architectures and obtain a clearer understanding of the effects of graph construction.

054
055
056
057
058
059
060
061
062
063
064
065
066
067



068
069
070
071
072
073
074
075
076
077
078
079
080
081
082



083
084

Figure 1: Similarity graphs for the Reuters (top) and Merck DPP4 (bottom) datasets.

085
086
087

Table 1: Results for Reuters dataset

Graph	Architecture	Parameters	Acc. (200)	Acc. (500)
-	FC-2000-1000-50	$6 \cdot 10^6$	70.2	70.2
Supervised	GC4-P4-FC1000	$2 \cdot 10^6$	69.41	70.0
Supervised	GC8-P8-FC1000	$2 \cdot 10^6$	69.15	-
Supervised	GC16-P4-GC16-P4-FC1000	$2 \cdot 10^6$	69.04	-
Supervised	GC64-P8-GC64-P8-FC1000	$2 \cdot 10^6$	running cims	-
RBF kernel	GC4-P4-FC1000	$2 \cdot 10^6$	running	-
RBF kernel	GC8-P8-FC1000	$2 \cdot 10^6$	running	-
RBF kernel	GC16-P4-GC16-P4-FC1000	$2 \cdot 10^6$	running cims	-
RBF kernel	GC64-P8-GC64-P8-FC1000	$2 \cdot 10^6$	running cims	-
RBF kernel (local)	GC4-P4-FC1000	$2 \cdot 10^6$	68.56	-
RBF kernel (local)	GC8-P8-FC1000	$2 \cdot 10^6$	runnnng cims	-

100
101
102

1.2 Merck Molecular Activity Challenge

103
104
105
106
107

The Merck Molecular Activity Challenge is a computational biology benchmark where the task is to predict activity levels for various molecules based on the distances in bonds between different atoms. For our experiments we used the DPP4 dataset which has 8193 samples and 2796 features. We chose this dataset because it was one of the more challenging and was of relatively low dimensionality

108 which made the spectral networks tractable. As a baseline architecture, we used the state-of-the-art
109 network of [?] which has 4 hidden layers and is regularized using dropout and weight decay.
110

111 As before, we used one-tenth of the training data to tune hyperparameters of the network. For this
112 task we found that $k = 40$ subsampled weights worked best, and that average pooling performed
113 better than max pooling. Since the task is to predict a continuous variable, all networks were trained
114 by minimizing the Root Mean-Squared Error loss.

115 Table 2: Results for Merck DPP4 dataset
116

Graph	Architecture	Parameters	R^2
-	FC-4000-2000-1000-1000-50	$22.1 \cdot 10^6$	0.2728
Supervised	GC64-P8-GC64-P8-1000-1000-1	$3.8 \cdot 10^6$	0.2629
RBF Kernel	GC64-P8-GC64-P8-1000-1000-1	$3.8 \cdot 10^6$	0.1992
RBF Kernel (local)	GC64-P8-GC64-P8-1000-1000-1	$3.8 \cdot 10^6$	running cims

124 1.3 ImageNet 125

126 In the experiments above our graph construction relied on an approximate estimation from the data.
127 To measure the influence of the graph construction compared to the filter learning in the frequency
128 domain, we performed the same experiments on the ImageNet dataset for which the graph is already
129 known, namely it is the 2-D grid. The spectral network was thus a convolutional network whose
130 weights were defined in the frequency domain. Training was performed exactly as in Figure 1,
131 except that the linear transformation was a Fast Fourier Transform.
132

133 Our network consisted of 4 convolution/ReLU/max pooling layers with 48, 128, 256 and 256 feature
134 maps, followed by 3 fully-connected layers with 4096 hidden units each regularized with dropout.
135 We trained two versions of the network: one classical convolutional network and one as a spectral
136 network where the weights were defined in the frequency domain only and were interpolated using
137 a spline kernel. Both networks were trained for 40 epochs over the ImageNet dataset where input
138 images were scaled down to 128×128 to accelerate training.
139

Graph	Architecture	Parameters	Test Accuracy (Top 5)	Test Accuracy (Top 1)
2-D Grid	Convolutional Network	$3.8 \cdot 10^6$	71.854	46.24
2-D Grid	Spectral Network	$3.8 \cdot 10^6$	71.998	46.71

140 We see that both models yield nearly identical performance. Interestingly, the spectral network learns
141 faster than during the first part of training, although both networks converge around the same time.
142 This requires further investigation.
143

144 References 145

151
152
153
154
155
156
157
158
159
160
161

```
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215
```

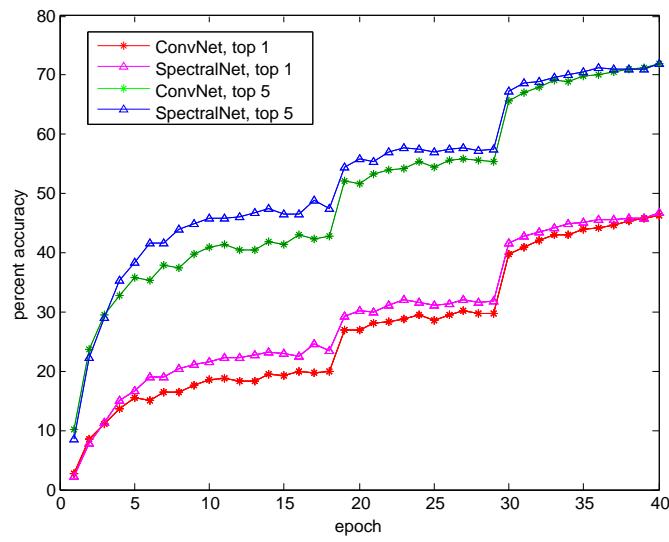


Figure 2: ConvNet vs. SpectralNet on ImageNet.