Deep Networks on Graph-Structured Data

Anonymous Author(s)
Affiliation
Address
email

Abstract

1 Experiments

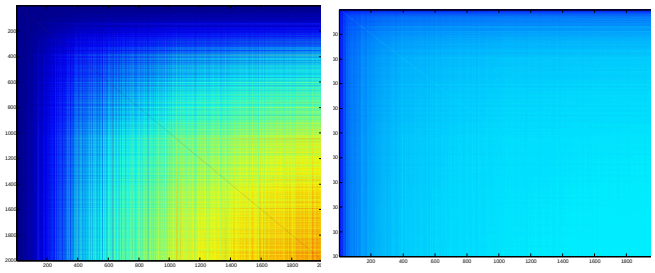
In order to measure the performance of spectral networks on real-world data and to explore the effect of the graph estimation procedure, we performed experiments on three datasets from computer vision, text categorization and computational biology. All experiments were done using the Torch machine learning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

In this section we adopt the following notation to descibe 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.

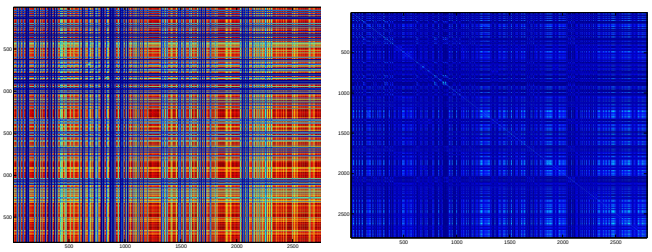
1.1 Reuters

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.



(a) Global scaling.. (b) Local scaling.



(c) Global scaling. (d) Local scaling.

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

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	$2 \cdot 10^6$	runnning cims	-

1.2 Merck Molecular Activity Challenge

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

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

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

Table 2: Results for Merck DPP4 dataset

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

1.3 ImageNet

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

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

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

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

References

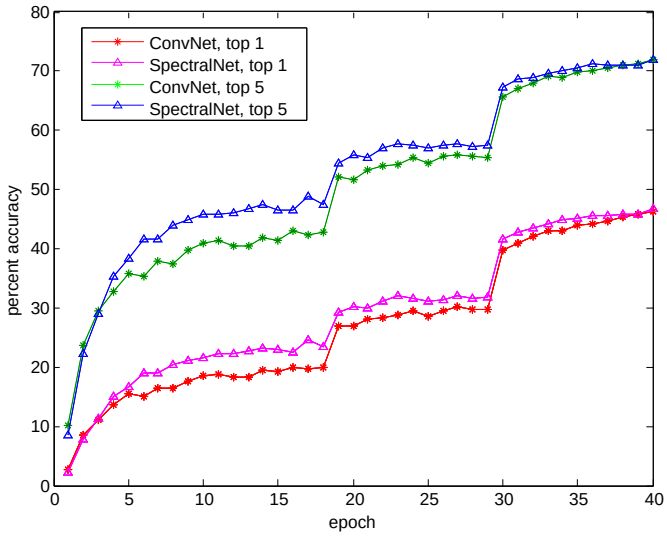


Figure 2: ConvNet vs. SpectralNet on ImageNet.