Graph Spectral Regularization for Neural Network Interpretability

Alexander Tong[*†] **David van Dijk**[*†] **Jay S. Stanley III**[†] **Matthew Amodio**[†]

Kristina Yim[†] **Rebecca Muhle**[†] **James Noonan**[†] **Guy Wolf**[‡] **Smita Krishnaswamy**[†§]

Abstract

While neural networks have been used to classify or embed data into lower dimensional spaces, they are often regarded as black boxes with uninterpretable features. Here we propose a novel class of *Graph Spectral Regularizations* for making hidden layers interpretable — and even informative of phenotypic state space. This regularization uses a graph Laplacian and encourages activations to be smooth either on a predetermined graph or on a feature-space graph learned from the data itself via co-activations of a hidden layer of the neural network. We show numerous uses for this including cluster indication and visualization in biological and image data sets.

1 Introduction

Many machine learning methods reduce data complexity by lowering the dimensionality of data (PCA, tSNE, autoencoders) Maaten & Hinton (2008); Hinton, Geoffrey E & Salakhutdinov, Ruslan R (2006). However, most dimensionality reduction methods are largely information-preserving, i.e., the new dimensions are generally irredundant and informative of the remainder of the feature space, but this often renders features even less interpretable. To address this issue, we propose a class of regularizations called *Graph Spectral Regularization* where the internal layers of a neural network are constrained to take the structure of a graph, i.e. with graph neighbors activating on similar inputs. We show that graph-structuring a hidden layer causes useful, interpretable features to emerge.

The main contributions of this manuscript are as follows: (1) Demonstration of interpretability, and visualizability of resultant feature spaces after application of regularization with standard graphs such as a rings and grids. (2) A novel method for learning and reinforcing the natural graph structure for complex feature spaces. (3) Demonstration of graph learning and abstraction on single-cell RNA-sequencing data.

2 Related Work

Graph Penalties Graph based penalties have been used in the graph signal processing literature (see, e.g., Belkin et al., 2004; Zhou & Schölkopf, 2004; Shuman et al., 2013), but are rarely used in a network learning setting. In the biological data setting, Min et al. used a graph penalty in sparse logistic regression on gene expression data Min et al. (2018).

Graph Neural Networks Graph Neural Networks (GNN) are a related body of work introduced by Gori et al. (2005), and expanded on by Scarselli et al. (2009), but focus on a different set of problems (For an overview see Wu et al. (2019)). We focus on learning a small graph representation of general data while graph neural networks only apply to data that comes as signals on a predefined feature graph. While a similarity graph between the

[*]equal contribution
[†]Yale University
[‡]University of Montreal
[§]corresponding author smita.krishnaswamy@yale.edu

features can be learned prior to applying a GNN, this graph may be large in high dimensional datasets, and we believe may be better learned in a compressed feature setting.

3 Enforcing Graph Structure

To enforce smoothing we use the Laplacian smoothing loss on activation vector on some activation vector z and fixed Laplacian L we formulate the graph spectral regularization function G as

$$G(z, \mathbf{L}) = z^T \mathbf{L} z$$

We add it to the reconstruction or classification loss with a weighting term α. For an autoencoder with input vector x and matching output y this then becomes,

$$Loss(x, y, z, L) = ||x - y||_2^2 + \alpha G(z, \mathbf{L}) \tag{1}$$

and thus optimizes for reconstruction and smoothness along the graph defined by L. This optimization procedure applies to any multi layer model and valid graph Laplacian. We apply this algorithm to grid, and hierarchical graph structures on both autoencoder and classification architectures.

3.1 Learning and Reinforcing an Abstracted Feature-space Graph

Instead of enforcing smoothness over a fixed graph, we can learn a feature graph from the data. Note, that most graph and kernel-based methods are applied over the space of observations but not over the space of features. One of the reasons is because it is even more difficult to define a distance between features than it is between observations. To circumvent this problem, we propose to learn a feature graph in the latent space of a neural network using feature co-activations as a measure of similarity.

We proceed by creating a graph using feature activation similarity, then applying this graph using Laplacian smoothing for a number of iterations. This converges to a graph of a latent feature space at the level of granularity of the number of dimensions in the corresponding layer.

Our algorithm for learning the graph consists of two phases. First, a pretraining phase where the model is learned with no graph regularization. Second, we alternate between constructing the graph from the similarities of the embedding layer features and further training the network for reconstruction and smoothness on the graph. There are many ways to create a graph from the feature $\times$ datapoint activation matrix. We use an adaptive gaussian kernel,

$$K(z_i, z_j) = \frac{1}{2} exp\left(\frac{||z_i - z_j||_2^2}{\sigma_i^2}\right) + \frac{1}{2} exp\left(\frac{||z_i - z_j||_2^2}{\sigma_j^2}\right)$$

where σ_i is the adaptive bandwidth for node i which we set as the distance to the k^{th} nearest neighbor of feature. An adaptive bandwidth gaussian kernel is necessary as the scale of the activations is not fixed in a standard neural network.

Since we are smoothing on the graph then constructing a new graph from the smoothed signal the learned graph converges to a steady state where the mean squared error acts as a repulsive force to stop the graph collapsing any further. We present the results of graph learning a biological dataset and show that the learned structure adds interepretability to the activations.

4 Results

We present three examples of graph spectral regularization, two cases where we have a graph structure in mind, and a third where we learn the underlying graph structure of a high-dimensional biological dataset, and show that this structure has biological relevance.

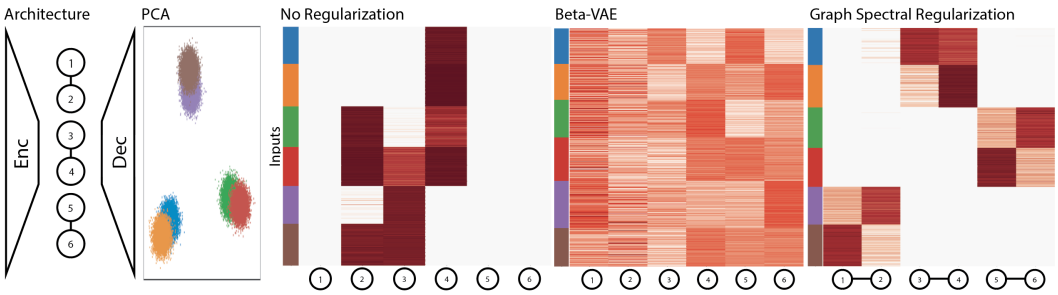

Figure 1: Graph architecture, PCA plot, activation heatmaps of a standard autoencoder, β-VAE Higgins et al. (2017) and a graph regularized autoencoder. In the model with graph spectral we are able to clearly decipher the hierarchical structure of the data, whereas with the standard autoencoder or the β-VAE the structure of the data is not clear.

Enforcing Hierarchical Structure We demonstrate graph spectral regularization on data that is generated with a hierarchical cluster structure. Our data contains three large-scale structures, each comprising two Gaussian subclusters generated in 15 dimensions (See Figure 1). We use this dataset as it has both global and local structure. We demonstrate that our graph spectral regularized model is able to pick up on both the global and local structure of this dataset. We use a graph-structure layer with six nodes with three connected node pairs and employ the graph spectral regularization. After training, we find that each node pair acts as a "supernode" that detects each large scale cluster. Within each supernode, each of the two nodes encodes one of each of the two Gaussian substructures. Thus, this specific graph topology is able to extract the hierarchical topology of the data.

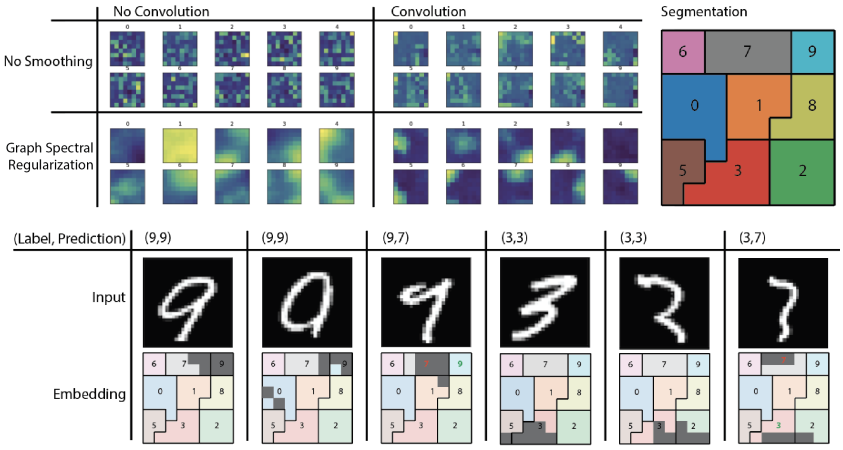

Figure 2: Shows average activation by digit over a 64 (8x8) 2D grid using graph spectral regularization and convolutions following the regularization layer. Next, we segment the embedding space by class to localize portions of the embedding associated with each class. Notice that the digit 4 here serves as the null case and does not show up in the segmentation. Finally, we show the top 10% activation on the embedding of some sample images. For two digits (9 and 3) we show a normal input, a correctly classified but transitional input, and a misclassified input. By inspection of the embedding space we can see the highlighted regions of the embedding space correlate with the semantic description of the digit type.

Enforcing Grid Structure on Mnist On MNIST we show the capability of graph spectral regularizations to create pseudo-images from data. We apply regularization to a classifier of mnist digits. Without graph-structured regularization, activations appear unstructured to the human eye and as a result are hard to interpret (See Figure 2). However, using graph spectral regularization over a grid graph we can make this representation more

visually distinguishable. Since we can now take this embedding as an image, it is possible to use a standard convolutional architecture in subsequent layers in order to further filter the encodings. When we add 3 layers of 3x3 2D convolutions with 2x2 max pooling we see that representations for each digit are compressed into specific areas of the image. This leads to the formation of receptive fields over the network pertaining to similar datapoints. Using these receptive fields, we can now extract the features responsible for digit classification. For example, features that contribute to the activation of the top right of our lattice we can associate with those features that contribute to being in the class of nines.

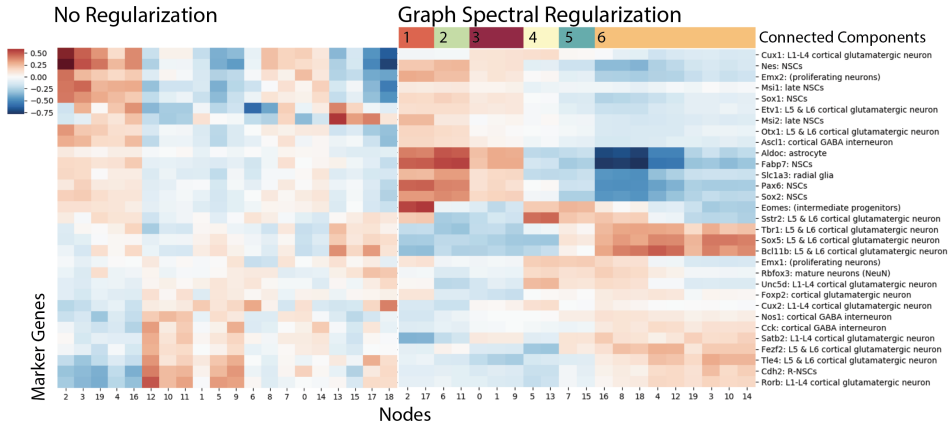

Figure 3: Shows correlation between a set of marker genes for specific cell types and embedding layer activations. First with the standard autoencoder, then our autoencoder with graph spectral regularization. The left heatmap is biclustered, the right heatmap is grouped by connected components in the learned graph. We can see progression especially in the largest connected component where features on the right of the component correspond to less developed neurons.

Extracting Cluster and Trajectory Structure on Single-cell RNA-Sequencing Data In Figure 3 we learn a graph on a single-cell RNA-sequencing dataset of over 4000 cells and over 8000 genes. The data contains a set of cells in the process of developing from neural stem cells to full neurons in the mouse brain. While there are many gene modules that contribute to the neuronal development, there are some states that have been studied. We use a list of cell type marker genes to validate our method. We use 1000 PCA components of the data in an autoencoder with a 20 dimensional embedding space. We learn the graph using an adaptive bandwidth gaussian kernel with the bandwidth for each feature set to the euclidean distance to the nearest neighboring feature.

Our graph learns six components that represent meta features over the gene space. We can identify each with a specific type of cell or related types of cells. For example, the light green component represents the very early stage neural stem cells as it is highly correlated with increased Aldoc, Pax6 and Sox2 gene expression. Most interesting to examine is cluster six, the largest component, which represents development into mature neurons. Within this component we can see a progression from intermediate progenitors on the left (showing Eomes expression) to more mature neurons with higher expression of Tbr1 and Sox5. With a standard autoencoder we cannot see progression structure like that in component six. While some of the more global structure is captured, we fail to see the data progression from intermediate progenitors to mature neurons.

5 Conclusion

We have introduced a novel method for regularizing features of the internal layers of a neural network to take the shape of a graph. We show that useful features emerge on datasets that are in the form of a ring, a grid, or cell type indicators on single-cell RNA-sequencing data.

Furthermore, when the intended graph is not known apriori, we have presented a method for learning the graph structure. This regularization framework has broad applicability for future work seeking to reveal important structure in real-world biological datasets as we have demonstrated here.

Acknowledgements

We thank the reviewers for their constructive feedback which helped to improve this work. We are grateful for support from the Chan Zuckerberg Initiative (CZI) award 182702.

References

Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In *International Conference on Computational Learning Theory*, pp. 624–638. Springer, 2004.

M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pp. 729–734, Montreal, Que., Canada, 2005. IEEE. ISBN 978-0-7803-9048-5. doi: 10.1109/IJCNN.2005.1555942.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. *International Conference on Machine Learning*, 2017.

Hinton, Geoffrey E and Salakhutdinov, Ruslan R. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, July 2006. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1127647.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Wenwen Min, Juan Liu, and Shihua Zhang. Network-Regularized Sparse Logistic Regression Models for Clinical Risk Prediction and Biomarker Discovery. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(3):944–953, May 2018. ISSN 1545-5963. doi: 10.1109/TCBB.2016.2640303.

F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80, January 2009. ISSN 1045-9227, 1941-0093. doi: 10.1109/TNN.2008.2005605.

David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *arXiv:1901.00596 [cs, stat]*, January 2019.

Dengyong Zhou and Bernhard Schölkopf. A regularization framework for learning from graph data. In *ICML workshop on statistical relational learning and Its connections to other fields*, volume 15, pp. 67–8, 2004.