Graph Convolutional Neural Networks on High Intrinsic Dimensional Data

Nolwazi M. Dube
Student Number: 565597
Supervisors: Main - Dr Terence Van Zyl
Core - Dr Bubacarr Bah

A Research Proposal submitted to the Faculty of Science,
University of the Witwatersrand, Johannesburg,
requirements for the degree,
Doctor of Philosophy

Abstract

In Machine Learning, Convolutional Neural Networks have proved successful in classifying images because of their ability to make use of the compositionality, stationarity and locality in natural images, video and speech. Unlike such data that lies on a Euclidean domain with a regular grid structure, there exists data that lies on a non-Euclidean domain with an irregular structure. Some examples include social networks in computational social sciences, sensor networks in communications, functional networks in brain imaging and regulatory networks in genetics.

Graphs are generic data representation forms which are useful for describing the geometric structures in high dimensional data. Data can be visualized as a weighted graph where the nodes are data points and the similarities between them are represented by the weighted edges connecting the nodes. Signal processing on graphs merges algebraic and spectral graph theoretic concepts with computational harmonic analysis to process such signals on graphs. In this research we construct Graph Convolutional Neural Networks (GCNs) on high intrinsic dimensional data whose underlying graph structure is a weighted directed graph. We in particular construct different graph Convolutional Neural Networks based on three different graph connectivity matrices (Directed Laplacian, Random Walk Laplacian and Incidence Matrix) and two different transforms namely the Graph Fourier Transform and the Graph Spectral Wavelet Transform. We then measure their performances and analyse which one best classifies this type of data.

— Nolwazi M. Dube

Declaration

I, Nolwazi M. Dube, hereby declare the contents of this research proposal to be my own work. This proposal is submitted for the degree of Doctor of Philosophy at the University of the Witwatersrand. This work has not been submitted to any other university, or for any other degree.

Acknowledgement

I would like to thank my supervisors, Dr. Terence Van Zyl and Dr. Bubacarr Bah, for their help during the initial phases of this research. Their input on both the content of this document, as well as background in the area have been of great help and are much appreciated.

Contents

1 Introduction

1.1 Background

Dimensionality in statistics refers to how many attributes a dataset has. For example, health care data is notorious for having vast amounts of variables (e.g. blood pressure, weight, cholesterol level). Technological innovations have revolutionized the process of scientific research and knowledge discovery. The availability of massive data and challenges from frontiers of research and development have reshaped statistical thinking, data analysis and theoretical studies. The challenges of high-dimensionality arise in diverse fields of sciences and the humanities, ranging from computational biology and health studies to financial engineering and risk management.

The fields of applications where high dimensional data is encountered includes processing of sensor arrays, image processing, multivariate data analysis and data mining. In machine learning, high dimensional data refers to the high number of features included in a dataset. In all of these fields, variable selection and feature extraction are crucial for knowledge discovery. The complexity of such networks [14] and interactions means that the data now resides on irregular and complex structures that do not lend themselves to standard tools [15]. Graphs offer the ability to model such data and complex interactions among them and we say such data lives in a non-Euclidean domain.

Deep Machine Learning models have proven extremely successful on a wide variety of tasks, from computer vision and acoustic modeling to natural language processing [13]. Much of their success lies in the statistical properties such as stationarity and compositionality through local statistics of the data, which are present in natural images, video and speech. These properties are best utilized by Convolutional Neural networks [11,12].

Data examples (images, video and speech) that can be classified by Convolutional Neural Networks can also be visualized as signals using ideas from digital signal processing [17]. These signals are on a low dimensional grid (data is in a Euclidean domain) and stationarity is well defined through the natural translation operator on the grid, locality is defined via the metric of the grid, and compositionality is obtained from downsampling, or equivalently thanks to the multi-resolution property of the grid [10]. However, there are also data examples that live in a non-Euclidean domain such as text documents which can be represented as bags of words which can be

referred to as signals defined on a graph whose nodes are vocabulary terms and whose weights represent some similarity measure between terms, such as co-occurrence statistics.

Data can be visualized as a graph where the nodes are data points and the similarities between them are represented by the edges connecting the nodes. For example, users on Twitter can be modeled as nodes while their friend connections can be modeled as edges. A graph signal is a set of values residing on a set of nodes. These nodes are connected via (possibly weighted) edges [18]. Both a signal on a graph with N vertices and a classical discrete-time signal with N samples can be viewed as vectors in $\mathbf{R}^N$ Graph signal processing, a process inspired by discrete signal processing is a tool that is used to process such kind of data. The intrinsic dimension of a graph signal describes how many features are needed to represent the signal.

In computer vision and audio, which is the main focus of deep learning, the only kind of data that is represented is defined on a simple low intrinsic dimensional graph. Complex graphs arising in other domains might be of higher intrinsic dimension, and the statistical properties of data defined on such graphs might not satisfy the stationarity, locality and compositionality assumptions previously described. For such type of data of intrinsic dimension N, deep learning strategies are reduced to learning with fully-connected layers, which have $O(N^2)$ parameters, and regularization is carried out via weight decay and dropout [21,22].

In this work we build up on the work from [10], where they use spectral graph theory [7] as a leveraged tool to define frequency spectra and expansion bases for graph Fourier transforms. The bases used here in particular are weighted undirected graph Laplacian eigenvectors. We are going to simulate varaious random graphs such as text and images whose stationarity, locality and compositionarity properties are controlled. We will then test these simulations on existing techniques discussed in [4] and [10].

However, there exists data that is examined in various application areas such as transportation, social networks, or biology that lies on weighted directed graphs. We shall explore the construction of deep Graph Convolutional Neural networks based on this type of data. This means that we shall use directed Laplacian eigenvectors as a basis for the graph Fourier Transform [5,6,23]. This requires a construction of a directed graph Fourier Transform which is already discussed in [18,20]. We will then test the performance of the deep Graph Convolutional Neural network based on real world data.

There are other matrices that describe the connectedness of a graph such as the Random Walk Laplacian matrix and Incidence matrix that we shall explore and test how the graph convolutional network performs when they are implemented during spectral embedding. The Fourier transform has a disadvantage that it cannot localize a signal in time but only in frequency. The Graph Fourier transform informs us that somewhere in the graph there is a high or low frequency but it cannot localize where exactly. The Graph Wavelet transform informs us where in particular this the high or low frequency is. Therefore it gives more information about the signal. We will construct a Graph Convolutional network based on the Graph Wavelet transform and test its performance.

1.2 Problem Statement

Can Graph Convolutional Neural Networks (GCNs) be constructed for high intrinsic dimensional data that depicts a directed graph? Which type of connectivity matrix best preserves the graphs connectivity properties when spectral embedding is implemented? Will the Graph Fourier Transform or the Graph Wavelet Transform be the optimal transform to use during the spectral embedding process?

1.3 Significance And Motivation

Existing research around the architecture of a Graph Convolutional Neural network had only focused on intrinsic high dimensional data that depicts weighted undirected graphs. There, however also exists high intrinsic dimensional data that depicts a weighted directed graph and has numerous applications. For example, Google maps uses graphs for building transportation systems, where the intersection of two (or more) roads are considered to be a vertex and the road connecting two vertices is considered to be an edge, thus their navigation system is based on the algorithm to calculate the shortest path between two vertices. In Operating System, we come across the Resource Allocation Graph where each process and resource is considered to be a vertex. Edges are drawn from resource to the allocated process, or from requesting process to the requested resource. If this leads to any formation of a cycle then a deadlock will occur. The Graph Convolutional Neural network model naturally integrates the connectivity patterns and feature attributes of graph-structured data and outperforms many state-of-the-art methods significantly on some benchmarks [4],[10]. Thus the development of algorithms to handle graphs is of major interest in the field of Computer Science.

1.4 Research Aims and Objectives

The aim of this research is to provably and practically construct and compare deep Graph Convolutional Neural Networks on high intrinsic dimensional graph-structured data. In particular on data that depicts a weighted directed graph.

The following are objectives we hope to achieve for this research:

1. Simulate a dataset for weighted undirected graphs whose stationarity, locality and compositionality are controlled. Then use these simulations to test at least two existing techniques to evaluate the performance of the Graph Convolutional Neural Network.

2. Simulate a new dataset for weighted directed graphs whose stationarity, locality and compositionality are controlled. Then use these simulations to test the existing techniques which will be chosen initially and investigate their performance.

3. Based on our observations, we develop an extension on ways to construct and analyse a Graph Convolutional Neural Network that best classifies data that depicts a weighted direct graph as follows:

    - Construct and analyze deep Graph Convolutional Neural Networks based on weighted directed graphs by using the directed Laplacian eigenvectors as a basis for the Graph Fourier Transform[5,20,23].
    - Construct and analyze deep Graph Convolutional Neural networks based on weighted directed graphs by using the Random Walk Laplacian eigenvectors as a basis for the graph Fourier Transform.
    - Construct and analyze deep Graph Convolutional Neural networks based on weighted directed graphs by using the Incidence Matrix eigenvectors as a basis for the graph Fourier Transform.
    - Construct and analyze deep Graph Convolutional Neural networks by using the different connectivity matrices named above (Laplacian, Random Walk Laplacian and Incidence matrix) but using their eigenvectors as a basis for the Graph Spectral Wavelet Transform.

4. Measure the performance of each of the graph Convolutional Network on real-world data by conducting experiments on three datasets from text categorization, computational biology and computer vision.

5. Observe and compare which graph Convolutional Neural Network best analyses this type of data.

1.5 Hypotheses and Research Questions

Hypotheses

The construction of Graph Convolutional Neural Networks based on other connectivity matrices besides the undirected Laplacian during the spectral embedding process can better analyze data that lies on a weighted digraph.

Research Questions

1. How will existing Graph Convolutional neural networks perform on simulated random graphs whose stationarity, locality and compositionality are controlled?

2. Which of the following methods of constructing a Graph Convolutional Neural Network will best classify weighted directed graph-structured data:

    - Using the directed Laplacian matrix to describe the connectedness of the graph then performing spectral embedding implemented through the Graph Fourier Transform or the Graph Wavelet Transform?
    - Using the Random Walk Laplacian matrix to describe the connectedness of the graph then performing spectral embedding implemented through the Graph Fourier Transform or the Graph Wavelet Transform?
    - Using the Incidence matrix to describe the connectedness of the graph then performing spectral embedding implemented through the Graph Fourier Transform or the Graph Wavelet Transform?

1.6 Delineations, Limitations and Assumptions

The graph convolution of the Graph Convolutional Network model is simply a special form of Laplacian smoothing, which mixes the features of a vertex and its nearby neighbors. The smoothing operation makes the features of vertices in the same cluster similar, thus greatly easing the classification task,

which is the key reason why Graph Convolutional Networks work so well.

However it also brings potential concerns of over-smoothing. If a Graph Convolutional Network is deep with many convolutional layers, the output features may be oversmoothed and vertices from different clusters may become indistinguishable. The mixing happens quickly on small datasets with only a few convolutional layers. Also, adding more layers to a Graph Convolutional Network will make it much more difficult to train. Also although the Graph Convolutional Network model compares favorably with other state-of-the-art methods, its mechanisms are not clear and it still requires considerable amount of labeled data for validation and model selection.

1.7 Definitions

Graphs (sometimes referred to as networks) offer a way of expressing relationships between pairs of items, and are one of the most important abstractions in computer science. The following are some Graph Theory preliminaries that we shall use in this research:

Definition 1. *A **graph** consists of a finite nonempty set V=V(G) of p vertices and a set $E = E(G)$ of q pairs of vertices of V called edges. A graph is determined by its edge and vertex set, so if G is a graph, we can write $G = (V, E)$. Similarly, $V(G)$ and $E(G)$ denote the **vertex set** and **edge set** of a graph G, respectively.*

Definition 2. *A **weighted undirected graph** is a graph in which edges have no orientation and each edge has an assigned weight to it.*

Definition 3. *A **directed graph or digraph** is a graph in which edges have orientations and each edge has an assigned weight to it.*

Definition 4. *The graph $G_1 = (V_1, E_1)$ is a **subgraph** of $G = (V, E)$ if $V_1 \subseteq V$ and every edge of G_1 is also an edge of G. A subgraph G_1 of a graph G is a spanning subgraph if $V_1 = V$. For any set S of vertices of G, the induced subgraph, denoted by $G[S]$ (or $< S >$) is the maximal subgraph of G with vertex set S.*

Definition 5. *Let u and v be vertices of a graph G. A u-v **walk** of G is a finite, alternating sequence $u = u_0, e_1, u_1, e_2, ..., e_p$, $u_p = v$ of vertices and edges in G beginning with vertex u and ending with vertex v such that $e_i = u_{i-1}u_i$, $i = 1, 2, ..., p$. The number p is called the **length** of the walk. A walk in which all the vertices are distinct is called a **path**.*

Definition 6. *A graph is* ***connected*** *if every pair of vertices are joined by a path. A maximal connected subgraph of G is called a* ***component*** *of G.*

We now state the rest of the preliminaries used in this research:

Definition 7. *In machine learning, a* ***Convolutional Neural Network*** *(CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery.*

Definition 8. ***Spectral graph theory*** *is the study of the properties of a graph in relationship to the characteristic polynomial, eigenvalues, and eigenvectors of matrices associated with the graph, such as its Adjacency matrix or Laplacian matrix.*

Definition 9. *Given a simple graph G with n vertices, its* ***Laplacian matrix****, $L_{n\times n}$,is defined as $L = D$, where D is the degree matrix and A is the adjacency matrix of the graph. The Laplacian in its normalized form is also one of the key operators that we are going to need. It is defined by $L_{sym} = D^{\frac{-1}{2}} L D^{\frac{-1}{2}}$.*

Definition 10. *A* ***Random Walk matrix*** *of a graph is defined as $W_{rw} = D^{-1}W$, where W is the weighted adjacency matrix.*

Definition 11. *A* ***Random Walk Laplacian matrix*** *of a graph is defined as $W_{rw} = D^{-1}L$, where L is the Laplacian matrix of the graph.*

Definition 12. *An* ***Incidence matrix*** *of a graph is a $|V| \times |E|$ matrix is defined as*

$$\boldsymbol{B}(v, e_{i,j}) = \begin{cases} \sqrt{w_{i,j}}, & \text{if } v = i. \\ -\sqrt{w_{i,j}}, & \text{if } v = j. \end{cases} \tag{1}$$

where $w_{i,j}$ is the weight between vertices i and j. v denotes a vertex in the graph.

Definition 13. *The* ***Fourier Tansform****, $\hat{f}$, decomposes a function of time (a signal), f, into the frequencies that make it up. It is defined as*

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx, \tag{2}$$

where ξ is the frequency and is any real number.

Definition 14. *The **Graph Fourier Transform**, $\hat{g}(\lambda_l)$, describes the frequency of a function, g, defined on a graph and it is defined as*

$$\hat{g}(\lambda_l) = \sum_{i=1}^{N} g(i)u(i)^l, \tag{3}$$

where $u(i), i = 1, ..., N$ are eigenvectors of any connectivity graph matrix such as the Laplacian matrix and $\lambda(_l), l = 1, ..., N$ are the consequent eigenvalues[19].

Definition 15. *The **Integral Wavelet Transform** is used to decompose a signal in both time and frequency and is defined as*

$$[W_\psi f](a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} \psi^*\Big(\frac{x-b}{a}\Big) f(x)dx, \tag{4}$$

where a is the frequency input variable and b is the time input variable. ψ denotes the mother wavelet. Wavelets at different locations and spatial scales are formed by translating and scaling the mother wavelet[9].

Definition 16. *The **Spectral Graph Wavelet Transform** is generated by operator-valued functions of the Laplacian, called wavelet operators. The wavelet operator at scale s, for a wavelet generating kernel g, is defined as $T_g^s = g(sL_{n\times n})$. It is defined by*

$$W_f(s, n) = (T_g^s)f(n) = \sum_{0}^{N-1} g(s\lambda_l)\hat{f}(l)u_l(n), \tag{5}$$

where f is the signal or function and $u_i, i = 0, ..., N-1$ are orthonormal eigenvectors[9].

2 Related Work

There have been several works which have explored architectures using the so-called local receptive fields [10], mostly with applications to image recognition. In particular, they propose a scheme to learn how to group together features based upon a measure of similarity that is obtained in an unsupervised fashion. However, it does not attempt to exploit any weight-sharing strategy.

Recently, [10] proposed a generalization of convolutions to graphs via the Graph Laplacian. By identifying a linear, translation-invariant operator in

the grid (the Laplacian operator), with its counterpart in a general graph (the Graph Laplacian), one can view convolutions as the family of linear transforms commuting with the Laplacian. By combining this commutation property with a rule to find localized filters, the model requires only O(1) parameters per "feature map". However, this construction requires prior knowledge of the graph-structure, and was shown only on simple, low-dimensional graphs. More recently, [10] introduced Shapenet, another generalization of convolutions on non-Euclidean domains based on geodesic polar coordinates, which was successfully applied to shape analysis, and allows comparison across different manifolds. However, it also requires prior knowledge of the manifolds.

3 Research Methodology

3.1 Introduction

Our research methodology will consist of acquiring knowledge through literature reviews as well as collecting data and running experiments in order to test the success of our Graph Convolutional Neural Network constructions.

3.2 Research Design

In this research, we use the qualitative and quantitative analysis design. It is qualitative because we start of with providing provable constructions of the Graph Convolutional Neural Network. We theoretically explore and gain insights into the problem of analyzing high intrinsic dimensional data. This will be done through Graph theory, Graph Signal Processing, Harmonic analysis and Functional analysis.

It is quantitative since we are going to test simulated data on traditional Graph Convolutional Networks. We generate models and hypotheses, collect empirical data and analyze it. We then create new simulations of digraphs and run experiments on the same traditional Graph Convolutional Networks and finally make an extension where we construct new Graph Convolutional Networks based on the results and analysis derived from previous experiments.

3.3 Methodology

Since we are building up on the work of [10], the methodology will be similar and the content is as follows:

1. Use Linear Algebra, Graph Theory and Spectral Graph theory to define graph signal processing [1,3,7,19]. However, graph signal processing uses tools from discrete signal processing [17], hence we shall first describe that. We will show how the directed graph Laplacian eigenvalues and eigenvectors provide a similar notion of frequency as the eigenvalues and eigenvectors in [5,8,20,23] in classical Fourier analysis.

2. Simulate various random weighted undirected graphs whose stationarity, locality and compositionality properties are controlled and then test these graphs on the existing techniques described in [4,10].

3. We then simulate a new data set of random directed graphs whose stationarity, locality and compositionality properties are controlled and test these graphs on existing techniques and measure their performance. Then finally construct the Graph Convolutional Networks and use real world-data to test its performance.

4. These two experiments will be compared and hence their performance measured through classification accuracy.

5. Since the graph structure is unknown, we will estimate a similarity matrix W from the data before constructing the spectral network. We will then consider two possible graph constructions, one unsupervised by measuring joint feature statistics, and another one supervised using an initial network as a proxy for the estimation [10].

6. Test the new model's performance by conducting experiments on three datasets from text categorization, social network and transportation as follows:

    - We will base the spectral network architecture on that of a classical convolutional network, namely by interleaving graph convolution, ReLU and graph pooling layers, and ending with one or more fully connected layers.
    - We will perform pooling at the beginning of the network to reduce the dimensionality in the graph domain hence mitigating the cost of the expensive Graph Fourier Transform operation.

- We will use the Reuters dataset described in [22], 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 [22] with two hidden layers consisting of 2000 and 1000 hidden units regularized with dropout.The performance will be measured through binary classification accuracy.
- We will use the Network Repository dataset which is a network data repository network with interactive graph analytics and visualization. We are going to use the social network dataset from it and then test its performance through binary classification accuracy.
- We also use Network Repository dataset to extract the transportation network dataset. We will test and therefore measure its performance through binary classification accuracy.
- We will use other relevant publicly available datasets for example the UCI data repository if the ones mentioned above are not helpful in training the model.

7. We will then conclude on the performance of the deep Graph Convolutional Neural network based on the experiments we will conduct and the theoretical analysis that preceded it.

3.4 Research Instruments

Research instruments are the experiments we will run to test the performance of the different formulations of the graph Convolutional Neural Networks we will construct. These experiments will require High Performance Computing (HPC) platforms like CHPC and Amazon Web Services (AWS).

4 Research Plan

4.1 Introduction

This research encompasses mathematical fields such as Graph Theory, Spectral Graph Theory, Functional Analysis and Harmonic Analysis. The computational aspect comes in when constructing the deep Graph Convolutional Network. The construction requires a thorough understanding in all these

different disciplines. This whole project should take about 36 months to complete allowing for some minor setbacks that might be faced.

4.2 Deliverables

The main deliverable for this work is the construction of Graph Convolutional Networks for data that lies on a weighted digraph. However, the success of this will depend on the following deliverables:

1. The success of simulating data that resides on an undirected weighted graph and testing on already existing Graph Convolutional Networks. We hope the simulations will corroborate new results.

2. We can then proceed and make new simulations for data that resides on a weighted digraph and test this on the existing Graph Convolutional Networks and analyze how these techniques classify such data. The analysis of this process will allow us some insight on how we can manipulate the construction of Graph Convolutional Networks into to accommodate this new dataset.

3. We can finally construct the different types of Graph Convolutional Networks based on our results and conclusions based on the previous steps. This whole process will allow us to make educated assumptions and therefore be efficient in our main goal.

4.3 Time Plan

1. Point 1 in section 4.2 requires an in-depth understanding of the reconstruction of the data through spectral embedding. It also required a knowledge in how the different types of already existing techniques work. Lastly, it requires knowledge on how to simulate such kind of data. This should take about six to eight months to achieve. If this process is a success a journal can be produced.

2. Point 2 in section 4.2 has similar requirements as in point 1 in section 4.2. It should take about the same amount of time if not less.

3. Point 3 in section 4.2 is our main goal and this should take at most 24 months to achieve.

4.4 Potential Issues

Potential issues with this research is that the main goal is dependent on previous results and if one of those results is false positive it might lead us to making wrong assumptions in the construction of the Graph Convolutional Networks. Another potential issue is that Convolutional Neural Networks require a huge amount of data for them to work efficiently and this might be computationally expensive. Finally, the statistical assumptions of stationarity and compositionality are not always verified in such situations, the constraints imposed by the model risk to reduce its capacity for no reason. One possibility for addressing this issue is to insert Fully connected layers between input and spectral layers, such that the data can be transformed into the appropriate statistical model.

5 Conclusion

While deep learning models have been particularly successful when dealing with signals such as speech, images, or video, in which there is an underlying Euclidean structure, recently there has been a growing interest in trying to apply learning on non-Euclidean geometric data. Such kinds of data arise in numerous applications such as transportation, social networks and biology. Graph Convolutional Networks have inspired a new wave of analyzing such graph-structured data.

The overall objective of this project is to extend the techniques of constructing deep Graph Convolutional Neural Networks for data that lies on a weighted digraph. Its success will open new avenues of its applications to different fields. In this research we aim to provably and practically construct different Graph Convolutional Neural Networks. We hope to achieve this by using different connectivity matrices to capture the topology of the graph-structured data. We also plan on using two different transforms namely to Graph Fourier Transform and the Graph Spectral Wavelet Transform in the spectral embedding process. We will then compare their performances and conclude which one classifies data that lies on a digraph the best. These new graph Convolutional Neural Networks will be a functional extension in this field as it can potentially inspire more research in the different application fields.

References

[1] A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," *in Int. Conf. Acoust., Speech, Signal Process. (ICASSP), pp. 3864–3868, IEEE*, May 2014.

[2] C. Apté, F. Damerau, and S. Weiss, "Automated learning of decision rules for text categorization," *ACM Trans. Inf. Syst.*, vol. 12, no. 3, pp. 233–251, 1994.

[3] D. S. Bernstein, Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear Systems Theory, Princeton University Press, 2005.

[4] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and deep locally connected networks on graphs. In *Proceedings of the 2nd International Conference on Learning Representations*, 2013.

[5] M. Chen, Q. Yang, and X. Tang. Directed graph embedding. *In Proceedings of IJCAI*, pages 2707–2712, 2007.

[6] F. Chung, "Laplacians and the Cheeger inequality for directed graphs," *Annals of Combinatorics*, vol. 9, no. 1, pp. 1–19, 2005.

[7] F. Chung, Spectral Graph Theory, *American Mathematical Society*, 1997.

[8] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, 2016.

[9] D. Hammond, P. Vandergheynst, R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis, Elsevier, 2011, 30 (2)*, pp.129–150.

[10] M. Henaff, J. Bruna, and Y. LeCun. Deep Convolutional Networks on Graph-Structured Data. *arXiv:1506.05163*, 2015.

[11] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. Signal Processing Magazine, 2012.

[12] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc.*, 2012.

[13] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. Nature, 521(7553):436–444, 05 2015.

[14] M. Newman, Networks: an introduction. Oxford University Press, 2010.

[15] A. Ortega, P. Frossard, J. Kovacevi c, J. M. F. Moura, and P. Vandergheynst, Graph signal processing: Overview, challenges, and applications, *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[16] W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: A comprehensive review, Neural computation 29 (9) (2017) 2352–2449.

[17] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[18] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, "On the graph Fourier transform for directed graphs," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 796–811, Sep. 2017.

[19] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp.3–98, May 2013.

[20] R. Singh, A. Chakraborty, and B. S. Manoj, "Graph Fourier transform based on directed Laplacian," *in Proc. Int. Conf. Signal Process. Commun.*, Jun. 2016, pp. 1–5

[21] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," *in Proc. Ann. Conf. Comp. Learn. Theory, ser. Lect. Notes Comp. Sci.,B. Schölkopf and M. Warmuth, Eds.* Springer, 2003, pp. 144–158.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdi-nov. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1):1929–1958, 2014.

[23] Q. Zheng and D.B. Skillicorn. Spectral embedding of directed networks. *In the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, forthcoming 2015.