# GRAPH LEARNING METHODS FOR CLIMATE MODELS WITH GRAPH SPARSIFICATION

Justin Clarke

Supervisor: Associate Professor Yanan Fan

School of Mathematics and Statistics
UNSW Sydney

November 2023

Submitted in partial fulfillment of the requirements of the degree of
Bachelor of Science with Honours

# Plagiarism statement

I declare that this thesis is my own work, except where acknowledged, and has not been submitted for academic credit elsewhere.

I acknowledge that the assessor of this thesis may, for the purpose of assessing it:

- Reproduce it and provide a copy to another member of the University; and/or,
- Communicate a copy of it to a plagiarism checking service (which may then retain a copy of it on its database for the purpose of future plagiarism checking).

I certify that I have read and understood the University Rules in respect of Student Academic Misconduct, and am aware of any potential plagiarism penalties which may apply.

By signing this declaration I am agreeing to the statements and conditions above.

Signed: Justin Clarke Date: 17/11/2023

# Acknowledgements

# Abstract

Graph sparification techniques for graph neural networks have traditionally been used to accelerate training and inference on real-world graphs which have billions of paramaters. There are also many different climate models which use complex mathematical models to model the interactions between energy and matter over the world. Many of these models share components and the structure of these relationships is not easily found due to the complexity of these climate models. The space of all possible graphs grows super-exponentially with the number of nodes and as such any correlation or causality is difficult to find. In this paper, I attempt to quantify these relationships with graph sparsification techniques. (Talk more about climate?)

# Contents

# Chapter 1

# Introduction

## 1.1 Climate Models

Climate models are based on well-known scientific processes and attempt to simulate the movement of fluids and energy throughout a system. The simplest of these have existed since the 1950's with the very first computers modelling simple variables on small two-dimensional climates [5]. Modern Global Climate Models (GCMs) have been scaled up into globally sized three dimensional sizes and most model the complex interactions between various physical, chemical, biological and geological processes. These models are constantly being updated with new data as many different groups and institutions implement higher spatial and temporal resolutions. This has in part been driven by developments in computational techniques and the vast amount of data available worldwide to train these models. [25] A common approach to studying the complex system of Earth's climate has been through sophisticated mathematical modelling with a range of temporal and spatial data [16] and at the centre of most of these approaches are the Navier-Stokes equations which are used describe the movements of liquids and gases in our oceans and atmosphere. [26]

The primary focus of these models has been forecasting time series data, as climate science aims to address a fundamental question: What impact have contemporary human greenhouse gas emissions had on the Earth and its future? Understanding this is important as it can inform us on the potential harms to society and the environment and guide the population and policy makers who are implementing change. The Coupled Model Intercomparison Project is a collaborative project between many meteoriological institutions that aims to improve our understanding of climate change. The sixth iteration of these models or CMIP6 are the premier models for this task but while it was expected to contain around 100 models made by 49 separate groups, delays have caused only 40 have been published so far. However, the results from these 40 models so far indicates a far greater sensitivity to increases in greenhouse gases when compared to the previous generation of CMIP5 models. [9] The basis behind the Intergovernmental Panel on Climate Change and its 2021 IPCC sixth assessment from the Coupled Model Intercomparison Project (CMIP6) were the Shared Socioeconomic Pathways (SSP). [17] These SSP's represent a broad set of possible changes in population, economic and technological growth, and urbanisation that would influence future changes to the climte. [30] These are directly related to Representative Concentration Pathways (RCP) introduced by the previous CMIP5 which are categorisations based on the estimated future concentrations of greenhouse gases in the atmosphere. [8].

| SSP | Description |
|---|---|
| SSP1 | Sustainability: The world shifts gradually, but pervasively, toward a more sustainable path, emphasizing more inclusive development that respects perceived environmental boundaries. |
| SSP2 | Middle of the road: The world follows a path in which social, economic, and technological trends do not shift markedly from historical patterns. |
| SSP3 | Regional rivalry: A resurgent nationalism, concerns about competitiveness and security, and regional conflicts push countries to increasingly focus on domestic or, at most, regional issues. |
| SSP4 | Inequality: Highly unequal investments in human capital, combined with increasing disparities in economic opportunity and political power, lead to increasing inequalities and stratification both across and within countries. |
| SSP5 | Fossil-fueled development: This world places increasing faith in competitive markets, innovation and participatory societies to produce rapid technological progress and development of human capital as the path to sustainable development. Global markets are increasingly integrated. |

Table 1.1: Shared Socioeconomic Pathways Descriptions

| RCP | Temperature Increase (2081–2100) | Sea Level Rise (2081–2100) |
|---|---|---|
| 2.6 | 1.0° C | 0.4m |
| 4.5 | 1.8° C | 0.47m |
| 6.0 | 2.2° C | 0.48m |
| 8.5 | 3.7° C | 0.63m |

Table 1.2: Representative Concentration Pathways

Scenarios are named based on the conjunction of their SSP level and RCP values. For example, on the lower end, SSP126 assumes an increasingly sustainable world where consumption is oriented towards minimising material resource and energy usage while SSP585 assumes a worst case scenario where fossil fuel usage and an energy-intensive lifestyle intensifies. The main output of CMIP6 models are the "scenario runs" which predict various outcomes in temperature, precipitation, air pressure and solar radiation given a certain SSP over time from 1850–2100.

## 1.2 Ensemble Models

Ensemble modelling is a process where multiple models are used in combination for a task and are more performant when the base models are diverse and independent [15]. CMIP6 is known as an 'ensemble of opportunity' [14], where the makeup the ensemble is determined by the ability of each base model to contribute. [1] This is the main benefit of ensemble methods as models can be weighted based on how accurate they are certain predictions. However, as research has become far more interconnected in the modern era, many aspects such as expertise, code and literature are often shared between groups. As such, many of the models that contribute to CMIP6 are highly likely to be dependent of each other. [1]. The degree of dependence between these models is difficult to ascertain as this would require a qualitative investigation into the personel, code and references between each component of CMIP6. Various novel approaches such as stochastic Markov chains [16] have been used to provide a more optimal ensemble mean which may account for this dependence. However, one would expect there to be some ground truth graph structure that links all models together through some dependence.

## 1.3 Deep Learning

There are many different kinds of machine learning but the advent of deep learning has led to countless advancements in many practical applications. The name deep learning refers to the ability of deep learning models to extract high-level, abstract features from raw data by using many layers of simple, computer understandable representations. Computers perform well on complex logical tasks such as arithmetic but often stuggle with more simplistic tasks that are harder to quantify such as visual recognition and language. The ability of deep learning models to quantify these simple tasks have allowed artifical intelligence to apporach near human level understanding. [7] The first neural networks developed in the late 1950's sought to simulate how human brain learned and operated. [31] The next development in neural networks also came from neuroscientific principles [11] with Convolutional Neural Networks (CNN) which could train models to be equivariant to translations in data and process data with grid-like structure. In recent years, Graph Neural Networks (GNN) have become the premier method of processing data with non-cartesian structure as standard convolutions on a graph structure much harder to define. The main feature of GNNs is the message passing framework, where information from features on each node is passed to neighbouring nodes then aggregated and embedded. This is then propagated through a neural network structure to perform a range of tasks on the entire graph, individual nodes or edges. (INSERT A DIAGRAM OF CONVOLUTION VS GRAPH CONVOLUTION) Much of this data exists in the world in applications such as chemical analysis [37], social networks analysis [29], link prediction [39] and unstructured data processing [24].

## 1.4 Sparsification of Deep Learning

An estimated 80 to 90 percent of the worlds' 79 zetabytes of data is unstructured and graphs make up a significant proportion of this data [12]. Traditional neural networks are not able to extract meaninful insights from this unstructured data without significant cleaning. The modern age has also produced many advancements in computing such as Massively Parallel Processing (MPP) [23] and big data which has led to an exponential growth in the size and complexity of these deep learning models. The well-known Generative-Pretrained Transformer 3 (GPT-3) model by OpenAI commonly used for ChatGPT had variants with 175 billion parameters which required 800 gigabytes to store. [27] Although Deep Neural Networks (DNN) tend to generalise well even when overparameterised [2], this level of overparameterisation makes inference and prediction highly costly when the same performance could be achieved on a far more simple model. To address this, the concept of the Lottery Ticket Hypothesis (LTH) [6] was introduced which explored the possibility of simplifying redundant models by trainable sparse subnetworks whilst still training to full accuracy. Chen et.al. [3] extended the LTH to Graph

Neural Networks by co-optimising graph and neural networks weights and zeroing out edges with the lowest magnitude. This reduced computational costs by over 85% depending on the size of the graph whilst maintaining a strong baseline accuracy.

## 1.5 Motivation

The goal of this thesis is to investigate whether these graph sparisifcation techniques can be used to determine some dependence structure within a graph of models which are all attempting to model the same scenario in the climate. Graph structure and dependence learning is already possible with unsupervised methods such as Variational Graph Autoencoders (VGAE) [38] but to our best knowledge, graph sparsification has not been used before as a method for graph structure learning or infering dependence. Existing methods for determining multiple correlation such as partial correlation and multiple correlation coefficient $R^2$ are linear methods. Graph sparsification is primarily used for simplifying graphs which have been overparameterised and grown too large but by removing these edges, the edges that remain should theoretically hold some relation to dependence in the graph.

## 1.6 Outline

The thesis is structured as follows. In Chapter 2, we will review the studies relating to neural networks, the extensions towards graph neural networks and the lottery ticket hypothesis (LTH). Current methods along with the benefits and shortcomings will be discussed and the terminology and definitions for will also be outlined in this section.

In Chapter 3, we will perform an exploratory data analysis (EDA) and provide a high-level overview of the dataset. This will provide a more in-depth understanding of the scenarios and CMIP6 models. There will also be a more comprehensive analysis of an individual climate modeL? (pick an example?) the climate modelling process and ensemble weighting method? We then introduce our problem formulation with by formalising the regression problem we will be using to perform the sparsification algorithm.

In Chapter 4, we visualise the results of the sparsification algorithm and we compare it to various correlation and partial correlation matrices. We may also look at Mutual Information Criterion (MIC) and compare with Variational Graph Autoencoder?

# Chapter 2

# Methods and Related Techniques

## 2.1 Deep Learning

The advent of deep learning provided algorithms that could automatically extract higher-level features from raw data. [4] The multi-layer perceptron [32] is formulated using linked layers of nodes which transforms a set of inputs into an output. The single layer version of this model can be represented as

$$f(x) = \sigma(\Theta^T X) \text{ where } \Theta = \begin{bmatrix} b \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \text{ and } X = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \tag{2.1.1}$$

Where $\sigma(.)$ is some activation function such as ReLu, hyperbolic tangent or logistic function, $X$ is the data, $\Theta$ is the learned parameters and $b$ represents a bias term. These parameters are set to some initial values and are iteratively updated in a back-propagation training process,

$$\theta^{t+1} = \theta^t - \eta \frac{\partial E(X, \theta^t)}{\partial \theta} \tag{2.1.2}$$

Where $E(.)$ is some loss function and $\eta$ is the learning rate.

The next advancement in the deep learning space came with the Convolutional Neural Network (CNN) which was a regularised MLP that could handle data with data with structure and multiple dimensions far better than the traditional MLP due to its use of weight sharing, sampling and local receptive fields. [7] Suppose we have an image or some other kind of data in two-dimensional matrix form. Let $\mathbf{X} \in \mathbb{R}^{H \times W}$ be the input image and $\mathbf{W} \in \mathbb{R}^{h \times w}$ be the kernel or filter. By performing a convolution, we are effectively 'sliding' our weight matrix kernel over our input and the resulting feature map $\mathbf{Z} = \mathbf{X} * \mathbf{W}$,

$$Z_{i,j} = \sum_{u=0}^{h-1} \sum_{v=0}^{w-1} x_{i+u,j+v} w_{u,v} \tag{2.1.3}$$

The novelty of the convolutional layer compared to a linear layer is that the kernel is shared across all locations of the input and therefore if a pattern in the input moves, the corresponding output will also follow this movement. This provides shift equivariance which is something that early MLP's failed to achieve. [22]

## 2.2 Graph Neural Networks

When it comes to data in a graph-like structure, standard CNN's cannot be applied due to the non-euclidean nature of a graph. In an image or a matrix, our kernel is generally a $n \times n$ matrix which can be applied to the entirety of the data. In graphs this is not always possible due to the fact that any number of nodes can be connected by any number of edges. [33] The Graph Neural Network (GNN) was developed for this purpose and they can be broadly categorised into gating and attention based methods [35] and spectral or spatial methods within Graph Convolutional Network (GCN) research. [13]

We define a graph as $\mathcal{G} = (\mathcal{V}, \mathbf{A})$, where $\mathcal{V}$ represents a set of verticies which contains a list of nodes $\{v_1, \dots, v_n\}$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$ the adjacentcy matrix which contains information on the graph topology. If an edge exists between two node $v_i$ and $v_j$, then $\mathbf{A}_{ij} = 1$ else, $\mathbf{A}_{ij} = 0$. We also define the degree matrix as $\mathbf{D} = \sum_j A_{ij}$ where each entry on the diagonal is equal to the row sum of the adjacency matrix $\mathbf{A}$. Each node has a p-dimensional feature vector $x_i \in \mathbb{R}^p$ which describes some information about the node in the graph. By combining all $n$ feature vectors from all nodes, we have a feature matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. The graph also has a response $Y \in \mathbb{R}^p$ which is for a graph-level task but a node level task would simply have $Y \in \mathbb{R}^{n \times p}$. The two-layer GNN from [13] can be expressed as

$$f(\mathbf{A}, \mathbf{X}) = \sigma_2(\hat{\mathbf{A}}_2 \sigma_1(\hat{\mathbf{A}}_1 \mathbf{X} \Theta^{(0)}) \Theta^{(1)}) \tag{2.2.1}$$

where $\sigma_1(.)$ and $\sigma_2(.)$ are an activation function such as ReLU, and $\hat{A} = \tilde{D}^{-1/2}(A + I)\tilde{D}^{-1/2}$ is the symmetrically normalised adjacency matrix. The propagation rule is then as follows:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} \Theta^{(l)}) \tag{2.2.2}$$

Here, $\tilde{A} = A + I_N$ is the adjacency matrix of the undirected graph $\mathcal{G}$ with self-connections from the identity matrix $I_N$. $\tilde{D_{ii}} = \sum_j \tilde{A}_{ij}$ is the row sum of the adjacency matrix and $W^{(l)}$ is a layer trainable weight matrix. $\sigma(.)$ is an activation function such as $\mathrm{ReLU}(.) = \max(0, .)$. $H^{(l)} \in \mathbb{R}^{N \times D}$ is the matrix of activations in the lth layer with $H^0 = \mathbf{X}$.

## 2.3 Sparsification of Graph Neural Networks

Hornik [10] showed that in a single layer MLP and provided enough hidden neuron units, the neural network could model any smooth truth function as for each added neuron, the decision space can be segmented to added linear regions to conform to any response. Many experiments have also found that deep networks with many layers perform better than shallow ones [21] [28] as the usage of many layers allows deeper layers to leverage features produced by earlier layers. The noveltly of neural networks is that they do not tend to be affected greatly by overparameterisation [2], and as they generally improve with more neurons and layers these models have grown exponentially in size with some models using billions of parameters and most of these models having more parameters than training observations which has made both inference and prediction incredibly costly. (Reference some math from the number of linear decision regions and some graphs?)

The most basic approach to inducing model sparsity is through an $l_1$ penalty in the loss function. [22] Unlike the $l_2$ regularisation which penalises large magnitude weights, the $l_1$ regularisation minimises and zeros weights which encourages sparsity. On a linear regression, this is done with a MAP estimation with a Laplace prior and is equivalent to optimising

$$\mathcal{L}(\Theta) = ||\mathbf{X}\Theta - \mathbf{Y}||_2^2 + \lambda||\Theta||_1 \tag{2.3.1}$$

where $\lambda$ is some tunable sparsity parameter. This is easily extended to neural networks by applying this penalty to the weights in the layers of the network. [19] Despite the this benefit, modern GPUs are optimised for dense matrix multiplication and as such there aren't many computational benefits from regularisation if certain weights across the network are zero. Methods that encourage *group* sparsity are able to prune whole nodes and layers out of our model result in *block sparse* weight matricies which provide much more substanial computational savings. [34] [36] [20] [18]

The Lottery Ticket Hypothesis (LTH) [6] explored the possibility of simplifying redundant models by trainable sparse subnetworks whilst still training to full accuracy. Chen et. al. [3] extended the LTH to Graph Neural Networks by co-simplifying both the adjacentcy matrix of the graph and the weights in the network of the model. For a semi-supervised classification task, the objective function is:

$$\mathcal{L}(\mathcal{G}, \Theta) = -\frac{1}{|\mathcal{V}_{\text{label}}|} \sum_{v_i \in \mathcal{V}_{\text{label}}} y_i \log(z_i), \tag{2.3.2}$$

where $\mathcal{L}$ is the cross-entropy error of all samples and $y_i$ is the label vector of node $v_i$. The Unified GNN Sparsification (UGS) framework then introduced two masks $m_g$ and $m_\theta$ with the same shape as the adjacency matrix $\mathbf{A}$ and the weights matrix $\Theta$, which gives the following objective function:

$$\mathcal{L}_{\text{UGS}} = \mathcal{L}(\{m_g \odot A, \mathbf{X}\}, m_\theta \odot \Theta) + \gamma_1||m_g||_1 + \gamma_2||m_\theta||_1, \tag{2.3.3}$$

where $\odot$ is the element-wise product, $\gamma_1$ and $\gamma_2$ are hyperparameters to control the shrinkage of $m_g$ and $m_\theta$. After training, the lowest magnitude elements in $m_g$ and $m_\theta$ are set to zero with respect to some set values of $p_g$ and $p_\theta$. These sparse masks are then applied which prune $\mathbf{A}$ and $\Theta$. The algorithm is then as follows

## 2.4 Variational Autoencoder

This section should be done with more time if the original sparsification section is completed. Show how this is an alternative in graph discovery.

**Algorithm 1** Unifed GNN Sparsification [3]

```
Input: Graph $\mathcal{G} = \{A, \mathbf{X}\}$, GNN $f(\mathcal{G}, \Theta_0)$, weight initialisation $\Theta_0$, masks $m_g^0 = A$ and $m_\theta^0 = 1 \in \mathbb{R}^{||\Theta_0||_0}$, step size $\eta$, $\lambda_g$ and $\lambda_\theta$.
Output: Sparse masks $m_g$ and $m_\theta$
for iteration $i = 0, 1, 2, \ldots, N - 1$ do
    Forward $f(\cdot, m_\theta^i \odot \Theta_i)$ with $\mathcal{G} = \{m_g^i \odot A, \mathbf{X}\}$        ▷ Computes 2.3.3
    Backpropagate to update $\Theta_{i+1} \leftarrow \Theta_i - \eta \nabla_{\Theta_i} \mathcal{L}_{UGS}$
    Update $m_g^{i+1} \leftarrow m_g^i - \eta \nabla_{m_g^i} \mathcal{L}_{UGS}$
    Update $m_\theta^{i+1} \leftarrow m_\theta^i - \eta \nabla_{m_\theta^i} \mathcal{L}_{UGS}$
end for
Set $p_g = 5\%$ of the lowest magnitude values in $m_g^N$ to 0 and others to 1, then
obtain $m_g$.
Set $p_g = 20\%$ of the lowest magnitude values in $m_\theta^N$ to 0 and others to 1, then
obtain $m_\theta$.
```

# Chapter 3

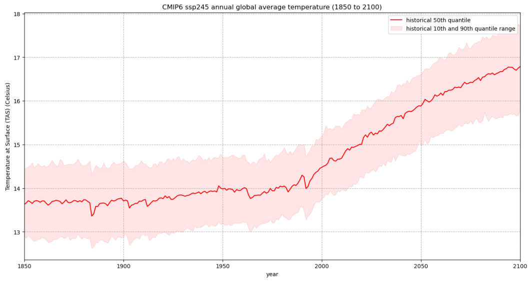
# Framework

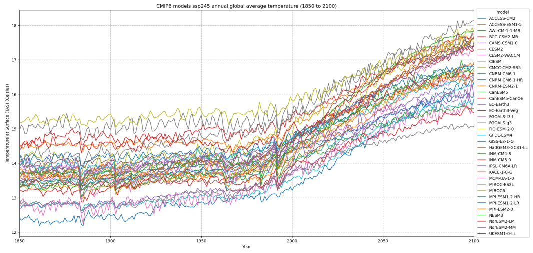
## 3.1 Climate Models

What climate models are used for etc. Use Yanan's climate papers.

(Can we talk about how one model works?)

## 3.2 Exploratory Data Analysis





Looking at a plot of all the models, there is a clear correlation between all the models and the correlation heatmap affirms this as the correlation between each ranges from 0.96–1.

## 3.3 Dataset

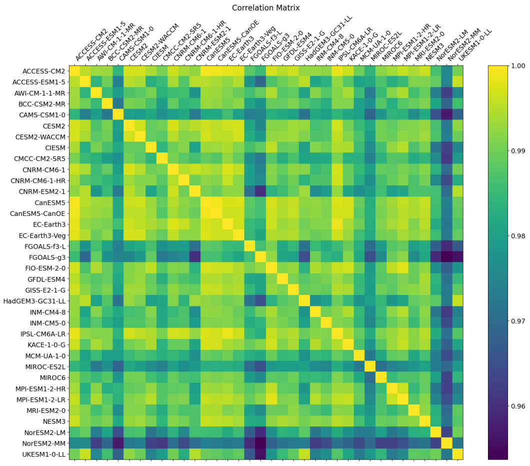
The datasets used are from the CMIP6 scenario runs made available on the KNMI Climate Explorer website. The KNMI is part of the World Meteoriological Organization (WMO) and obtained from `https://climexp.knmi.nl/`. The scenario runs include monthly predictions for temperature, min temperature, max temperature precipitation, radiation and pressure all over the globe in a 192×144 grid between 1850–2100 for 40 different models. Due to the time and computational limitations, this was filtered to just temperature during the 1960–1980 period in just Australia or latitudes -44° to -12° and longtitudes 288° to 336°.

Talk a bit more about how these models in the dataset are all related by certain parts.

## 3.4 Problem Formulation

*Maybe add this to the background section and put more of the regression, diagrams of the process and shrinkage prior stuff here that it more specific to this thesis*

The final regression problem can be formulated as As mentioned earlier, other variables such as precipitation, pressure and radiation are available but for simplicity, just temperature is currently being used.



$$f : L \times X \to Y \tag{3.4.1}$$

where $f$ denotes the learning function, $L$ the graph, $X$ denotes the time series input and $Y$ the regression target.

Need to describe the math behind graph sparsification. More about shrinkage see Xiongwens.

## 3.5 Computational features

computation of neural network models. See georges paper

## 3.6 Implementation

Need to finish code to finish this section.

# Chapter 4
# Results

## 4.1 Model verification

If the VGAE section is completed, we can compared the sparsified graph with the VGAE produced graph to determine how good graph sparsification is when used for graph discovery and thereby correlation in a graph structure.

## 4.2 Model results

Is there some way we can test the models results depending on how sparse we make the graph etc. Research required to find some quantitative measure for this.

Some figures of the NN structure would also be helpful for this. Need to use nx or some other graph representation tool in python for this.

# Chapter 5

## Discussion

# Chapter 6

## Conclusion

# Chapter 7

# Appendix

# References

[1] G. Abramowitz and C. H. Bishop. Climate model dependence and the ensemble dependence transformation of cmip projections. *Journal of Climate*, 28(6):2332 – 2348, 2015.

[2] Yuan Cao and Quanquan Gu. Generalization error bounds of gradient descent for learning over-parameterized deep relu networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3349–3356, Apr. 2020.

[3] Tianlong Chen, Yongduo Sui, Xuxi Chen, Aston Zhang, and Zhangyang Wang. A unified lottery ticket hypothesis for graph neural networks, 2021.

[4] Li Deng and Dong Yu. *Deep Learning: Methods and Applications.* Now Foundations and Trends, 2014.

[5] Paul N Edwards. History of climate modeling. *Wiley Interdisciplinary Reviews: Climate Change*, 2(1):128–139, 2011.

[6] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis, 2020.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016.

[8] Thomas Harrisson. Explainer: How ‘shared socioeconomic pathways’ explore future climate change, Apr 2018.

[9] Thomas Harrisson. Cmip6: The next generation of climate models explained, Oct 2021.

[10] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[11] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of Physiology*, 160(1):106–154, 1962.

[12] W.H. Inmon and A. Nesavich. *Tapping into Unstructured Data: Integrating Unstructured Data and Textual Analytics into Business Intelligence.* Pearson Education, 2007.

[13] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[14] Reto Knutti, Gab Abramowitz, M. Collins, Veronika Eyring, Peter Gleckler, Bruce Hewitson, and Linda Mearns. Good practice guidance paper on assessing and combining multi model climate projections. pages 1–15, 01 2010.

[15] Vijay Kotu and Bala Deshpande. Chapter 2 - data science process. In Vijay Kotu and Bala Deshpande, editors, *Data Science (Second Edition)*, pages 19–37. Morgan Kaufmann, second edition edition, 2019.

[16] Max Kulinich. *A Markov chain method for weighting climate model ensembles and uncertainty estimation on spatially explicit data.* PhD thesis, UNSW, 2022.

[17] June-Yi Lee, Jochem Marotzke, Govindasamy Bala, Long Cao, Susanna Corti, John P Dunne, Francois Engelbrecht, Erich Fischer, John C Fyfe, Christopher Jones, et al. Future global climate: scenario-based projections and near-term information. In *Climate change 2021: The physical science basis. Contribution of working group I to the sixth assessment report of the intergovernmental panel on climate change*, pages 553–672. Cambridge University Press, 2021.

[18] Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. *Advances in neural information processing systems*, 30, 2017.

[19] Rongrong Ma, Jianyu Miao, Lingfeng Niu, and Peng Zhang. Transformed l1 regularization for learning sparse deep neural networks. *Neural Networks*, 119:286–298, 2019.

[20] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, pages 2498–2507. PMLR, 2017.

[21] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[22] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction.* MIT Press, 2022.

[23] Tomas Nordström and Bertil Svensson. Using and designing massively parallel computers for artificial neural networks. *Journal of parallel and distributed computing*, 14(3):260–285, 1992.

[24] Sara Nouri Golmaei. *Improving the Performance of Clinical Prediction Tasks by Using Structured and Unstructured Data Combined with a Patient Network.* PhD thesis, 2021.

[25] Jonathan Overpeck, Gerald Meehl, Sandrine Bony, and David Easterling. Climate data challenges in the 21st century. *Science (New York, N.Y.)*, 331:700–2, 02 2011.

[26] Tim Palmer and Phoebe Williams. Stochastic physics and climate models. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 366:2421–7, 04 2008.

[27] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[28] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2847–2854. PMLR, 06–11 Aug 2017.

[29] Bhavtosh Rath, Aadesh Salecha, and Jaideep Srivastava. Detecting fake news spreaders in social networks using inductive representation learning. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 182–189. IEEE, 2020.

[30] Keywan Riahi, Detlef P. van Vuuren, Elmar Kriegler, Jae Edmonds, Brian C. O'Neill, Shinichiro Fujimori, Nico Bauer, Katherine Calvin, Rob Dellink, Oliver Fricko, Wolfgang Lutz, Alexander Popp, Jesus Crespo Cuaresma, Samir

KC, Marian Leimbach, Leiwen Jiang, Tom Kram, Shilpa Rao, Johannes Emmerling, Kristie Ebi, Tomoko Hasegawa, Petr Havlik, Florian Humpenöder, Lara Aleluia Da Silva, Steve Smith, Elke Stehfest, Valentina Bosetti, Jiyong Eom, David Gernaat, Toshihiko Masui, Joeri Rogelj, Jessica Strefler, Laurent Drouet, Volker Krey, Gunnar Luderer, Mathijs Harmsen, Kiyoshi Takahashi, Lavinia Baumstark, Jonathan C. Doelman, Mikiko Kainuma, Zbigniew Klimont, Giacomo Marangoni, Hermann Lotze-Campen, Michael Obersteiner, Andrzej Tabeau, and Massimo Tavoni. The shared socioeconomic pathways and their energy, land use, and greenhouse gas emissions implications: An overview. *Global Environmental Change*, 42:153–168, 2017.

[31] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[32] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[33] Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B. Wiltschko. A gentle introduction to graph neural networks. *Distill*, 2021. https://distill.pub/2021/gnn-intro.

[34] Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

[35] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[36] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. *Advances in neural information processing systems*, 29, 2016.

[37] Katherine Xu and Janice Lan. Chemistry insights for large pretrained GNNs. In *NeurIPS 2022 AI for Science: Progress and Promises*, 2022.

[38] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7154–7163. PMLR, 09–15 Jun 2019.

[39] Muhan Zhang, Pan Li, Yinglong Xia, Kai Wang, and Long Jin. Revisiting graph neural networks for link prediction. 2020.