

# Predicting the Onset of Alzheimer's Disease using Graph Neural Networks and Diffusion Tensor Imaging

By Siddarth Iiju, Cherry Creek High School

## Introduction

### Research Problem

**Alzheimer's disease** is an irreversible, progressive brain disorder that slowly destroys memory and thinking skills and is the most common cause of dementia among older adults. Experts suggest that as many as 5.5 million Americans age 65 and older have the disease. [1] Due to the complex pathology of the disease and the late onset of clinical symptoms, it is difficult for researchers to adequately detect and treat Alzheimer's effectively. Researchers are currently working on methods of diagnosing Alzheimer's preemptively and predicting whether individuals will go on to develop the pathology and cognitive decline that is characteristic of the disease.

Currently, neuroimaging is the standard for diagnosing AD (Alzheimer's disease). Structural and functional MRI scans can aid in determining loss of brain tissue due to neurodegeneration and reduced cognitive function. However, recent work has suggested that **diffusion tensor imaging (DTI)** can better predict future cognitive decline in patients. [2]

One of the more popular methods of processing DTI images is through deep learning, although work in the field has only started recently. Separately, a recent development in the field of deep learning led to the creation of **graph neural networks (GNNs)**. The underlying motivation behind graph neural networks lies in determining an effective way to perform deep learning on non-Euclidean data such as brain connectomes, social networks, and chemical states. [3]

Thus, my research question for this year was:

**Is it possible to accurately predict the future onset of Alzheimer's disease using Diffusion Tensor Imaging and Graph Neural Networks?**

### Methodology

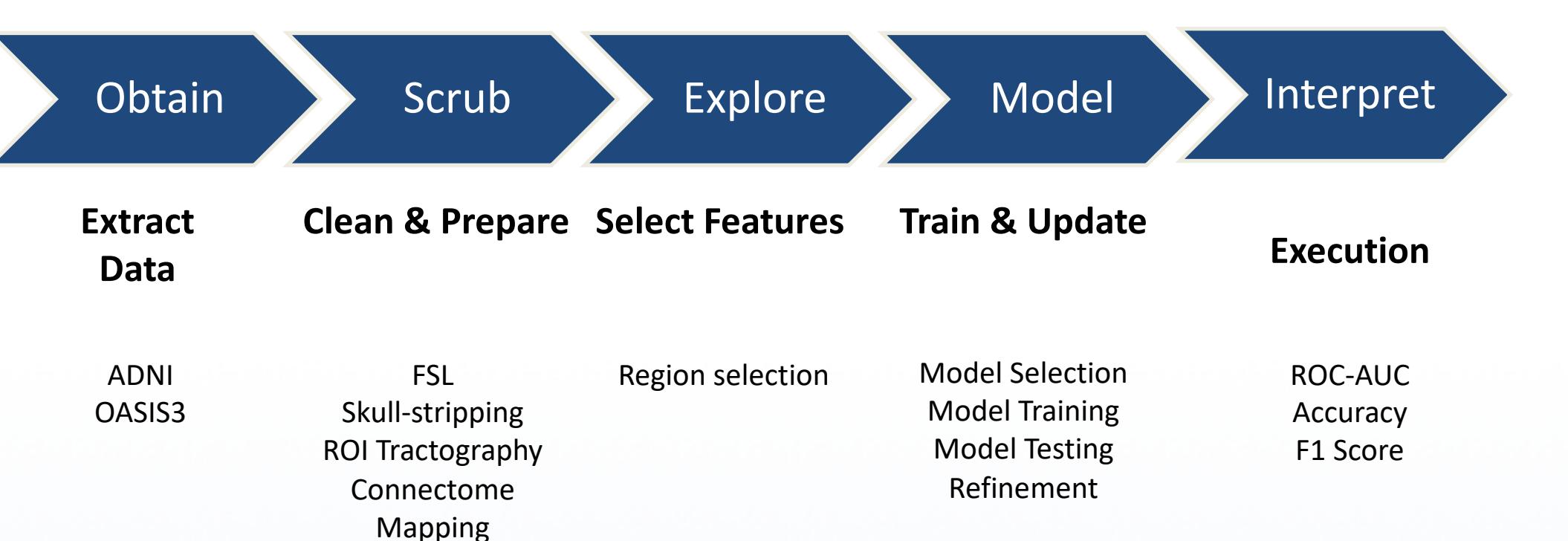
Previous approaches to the problem tended to err on the biological aspects of it, i.e., finding various biomarkers or pathways in the brain that demonstrated a connection to common symptoms of Alzheimer's. However, most of these studies produced inconclusive results due to the pure complexity of the problem.

Due to the easily representable nature of the problem, researchers turned towards computer science in their solution methodologies. Many previous approaches within deep learning utilized the classic Convolutional Neural Network architecture, which showed some improvement over older approaches. [4]

The concept of Graph Neural Networks was very recently realized and has yet to be applied to the problem of detecting Alzheimer's through image analysis. I set out to discover if the GNN would be capable of approaching similar, or even improving on, the results of the CNN.

### Procedure

The figure below shows the overall steps taken for this project from the initial stages to the final results:



### Model Variation and Experimentation

Throughout model creation, various **filters** were experimented with within the graph convolution layer. The filter represents the propagation function  $f$  and can take several forms. Within the context of the actual implementation of the graph neural network, the filter is expressed as an operation on  $\mathbf{A}$ , the adjacency matrix of the graph, as seen below.

$$\mathbf{X}^{(l+1)} = \mathcal{A}\mathbf{X}^{(l)}\mathbf{W}^{(l)}$$

Classic GNN Layer Formula (Medium)

Various other filters were experimented with, including Edge-Conditioned and Chebyshev filters. The final filter used was the Spectral Filter, as described by the equation below.

$$\mathbf{X}^{(l+1)} = \mathbf{V}(\mathbf{V}^T \mathbf{X}^{(l)} \odot \mathbf{V}^T \mathbf{W}_{\text{spectral}}^{(l)})$$

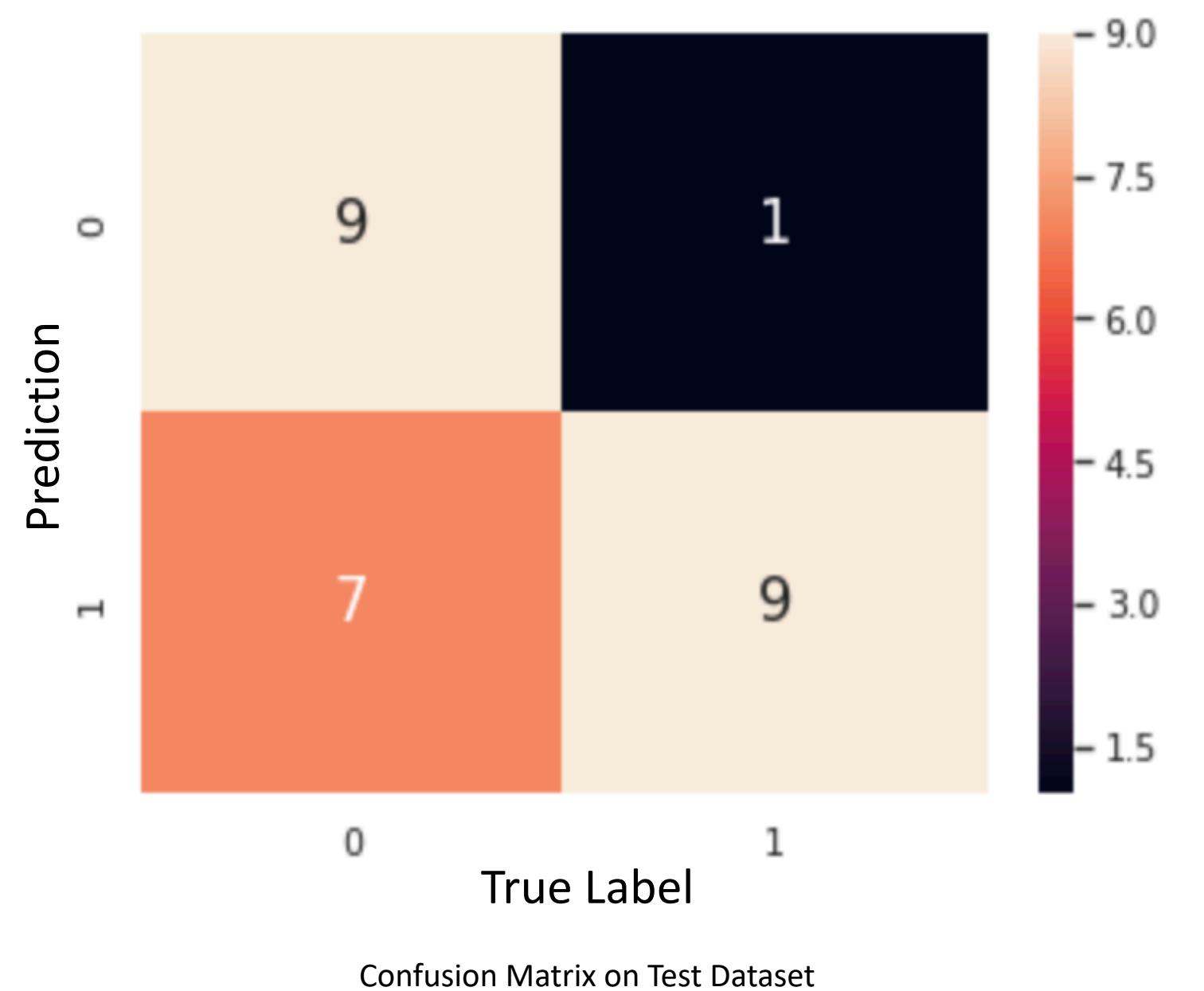
Spectral Graph Convolution Layer (Medium)

The spectral graph convolution filter utilizes the eigenvectors of the **graph Laplacian L**. The eigenvectors of a matrix  $\mathbf{A}$  are the vectors  $\mathbf{v}$  that solve the equation  $\mathbf{v}(\mathbf{A} - \lambda\mathbf{I})$ , where  $\lambda$  represents the corresponding eigenvalue of  $\mathbf{v}$  and  $\mathbf{I}$  represents the identity matrix. The Laplacian can then be defined as  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ , where  $\mathbf{D}$  is the diagonal matrix of vertex degrees in the graph represented by  $\mathbf{A}$ .

A non-zero eigenvector of  $\mathbf{L}$  is called a **Fiedler vector**. The Fiedler vector can then be used to partition the graph into separate clusters based off the sign of the coefficients of the vector. We can choose how many partitions we take of the graph by changing the number of eigenvectors used. In this project, I used 8 eigenvectors because each vertex in my graph was guaranteed to have maximum 8 neighbors. Displayed below are the adjacency matrix  $\mathbf{A}$  and corresponding Laplacian  $\mathbf{L}$ .



Additionally, the distribution of results in the confusion matrix show clinical promise. The model appeared to err on the side of predicting the onset of Alzheimer's even if there wasn't, implying that there would be little to no false negatives when implemented. The confusion matrix for the test dataset is displayed below.



### Discussion

The model underwent many trials and revisions before arriving at the final version. To start, the model was trained on the OASIS3 dataset (which consisted of 28 DTI images) and was implemented using a vanilla graph neural network with an aggregation filter. After multiple iterations using the vanilla GNN with minimal success, and upon experimenting with various other filters, the neural network was modified in order to use a spectral filter. The results significantly improved under this filter, which was compounded by switching from the OASIS3 dataset to the ADNI dataset, which consisted of 55 DTI images.

## Background

The main data format for this project was the Diffusion Tensor Imaging modality of MRI. DTI functions by measuring the diffusion of water molecules in the brain. The anisotropic diffusion measured in tissue varies with the direction of the tissue, due to cellular membranes and the packing of axons in neurons. Thus, anisotropic diffusion can indicate the underlying tissue orientation within the brain.

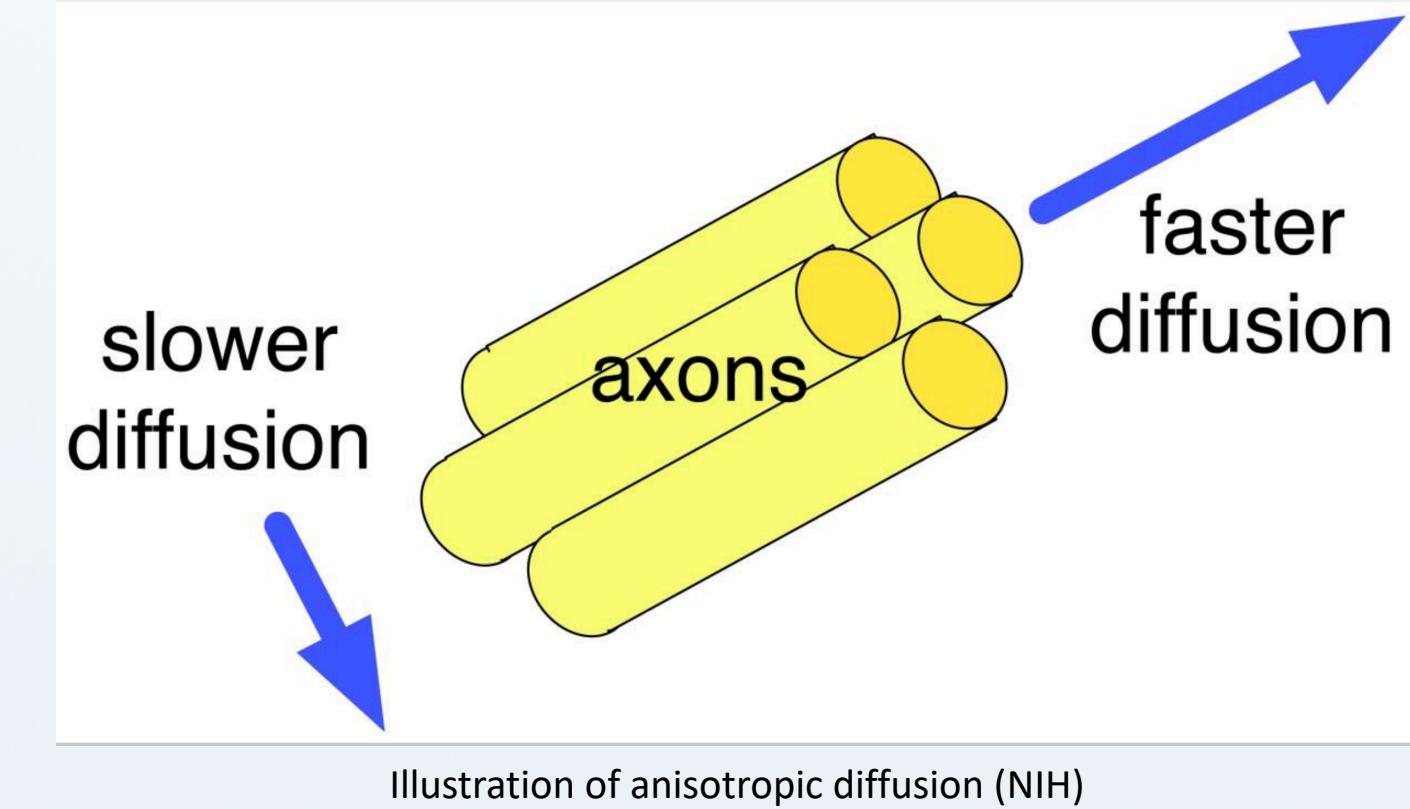
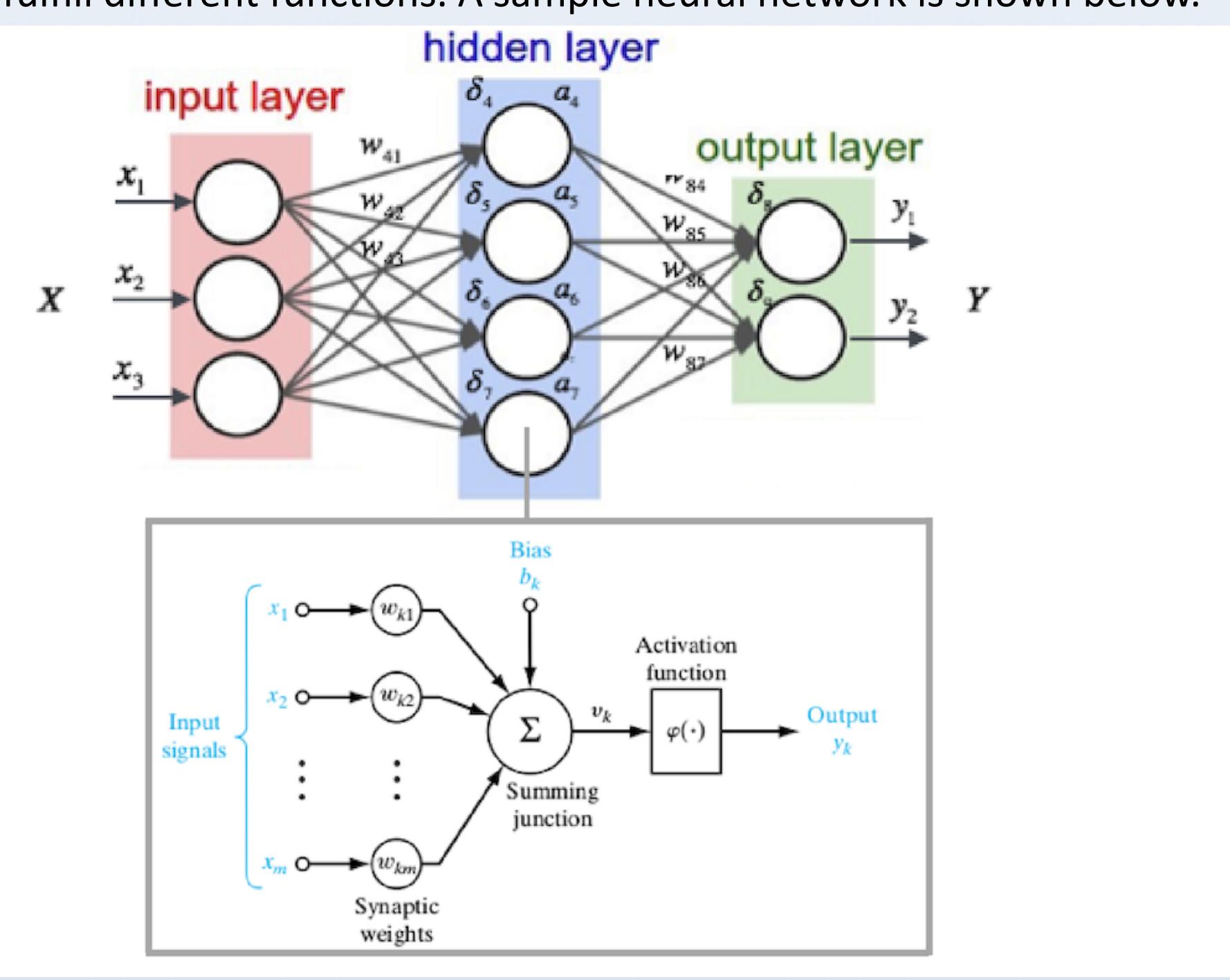


Illustration of anisotropic diffusion (NIH)

## Neural Networks and Deep Learning

The basic structure of neural networks is modeled off the function of the human brain itself. Similar to how neurons in the human brain are interconnected in vast networks, neurons in neural networks are connected in layers to fulfill different functions. A sample neural network is shown below.



Neural network graphic (adapted from Medium)

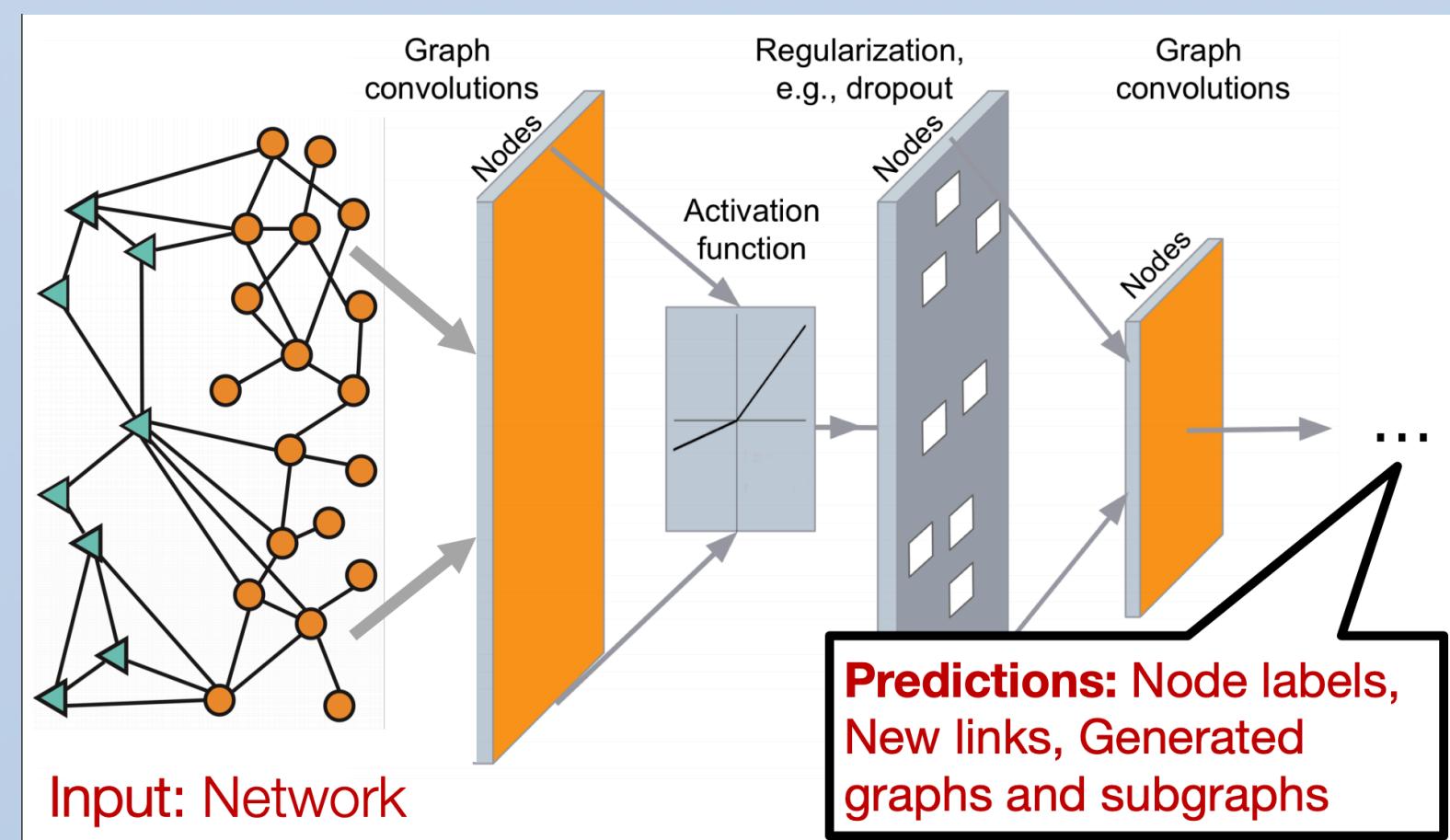
The goal of a neural network is to minimize the cost function, which is the average of the errors for individual training samples taken from the entire training dataset. This individual error is known as a loss function. To minimize the loss function of the network, we use the processes of forward and backward propagation. Forward propagation consists of feeding the input data through the network. Backward propagation consists of computing the gradients of change necessary to minimize the loss function. To do so, it utilizes the chain rule and partial derivatives. This process is summarized below.

$$\begin{aligned} \mathbf{dW}^{[l]} &= \frac{\partial L}{\partial \mathbf{W}^{[l]}} = \frac{1}{m} \mathbf{dZ}^{[l]} \mathbf{A}^{[l-1]T} \\ \mathbf{db}^{[l]} &= \frac{\partial L}{\partial \mathbf{b}^{[l]}} = \frac{1}{m} \sum_{i=1}^m \mathbf{dZ}^{[l](i)} \\ \mathbf{dA}^{[l-1]} &= \frac{\partial L}{\partial \mathbf{A}^{[l-1]}} = \mathbf{W}^{[l]T} \mathbf{dZ}^{[l]} \\ \mathbf{dZ}^{[l]} &= \mathbf{dA}^{[l]} * g'(\mathbf{Z}^{[l]}) \end{aligned}$$

Backpropagation graphic (Piotr Skalski)

## Graph Neural Networks

A graph can be defined as  $G = (V, E)$ , where  $V$  and  $E$  are the set of vertices and the set of edges connecting elements of  $V$ , respectively. A graph neural network takes as its input a feature matrix  $X$  of dimension  $N \times F$  where  $N$  is the number of nodes and  $F$  is the number of features per node and an  $N \times N$  matrix representation of the graph structure such as the adjacency matrix  $A$ . A hidden layer in a GNN can be written as  $H^l = f(H^{l-1}, A)$  where  $H^0 = X$  and  $f$  is a propagation. Variants in GNNs tend to only differ in the choice of propagation rule  $f$ . Due to the assumption of the disorder of nodes in the graph, the traditional operation of the convolution cannot be performed on a GNN in the same way it can be performed on a convolutional neural network. Thus separate filters must be used in the propagation function  $f$  in order to aggregate features from surrounding nodes in steps of the training process.



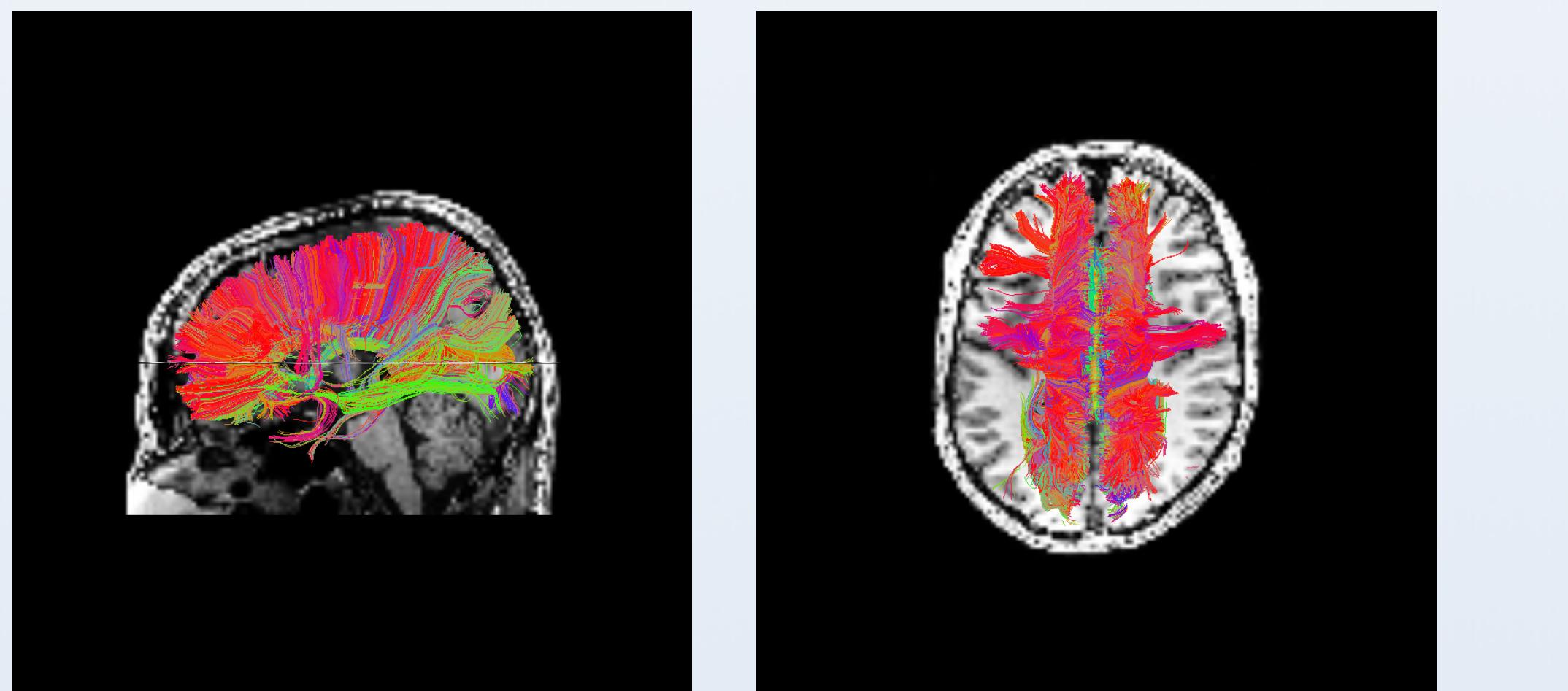
GNN Visualization (Cite Source Here)

## Data Processing

The datasets of choice for this project were the OASIS3 and ADNI dataset. Both datasets consisted of DTI volumes taken with 41 gradient directions using a Philips 3.0T MRI Scanner. All images were de-identified and obtained with permission.

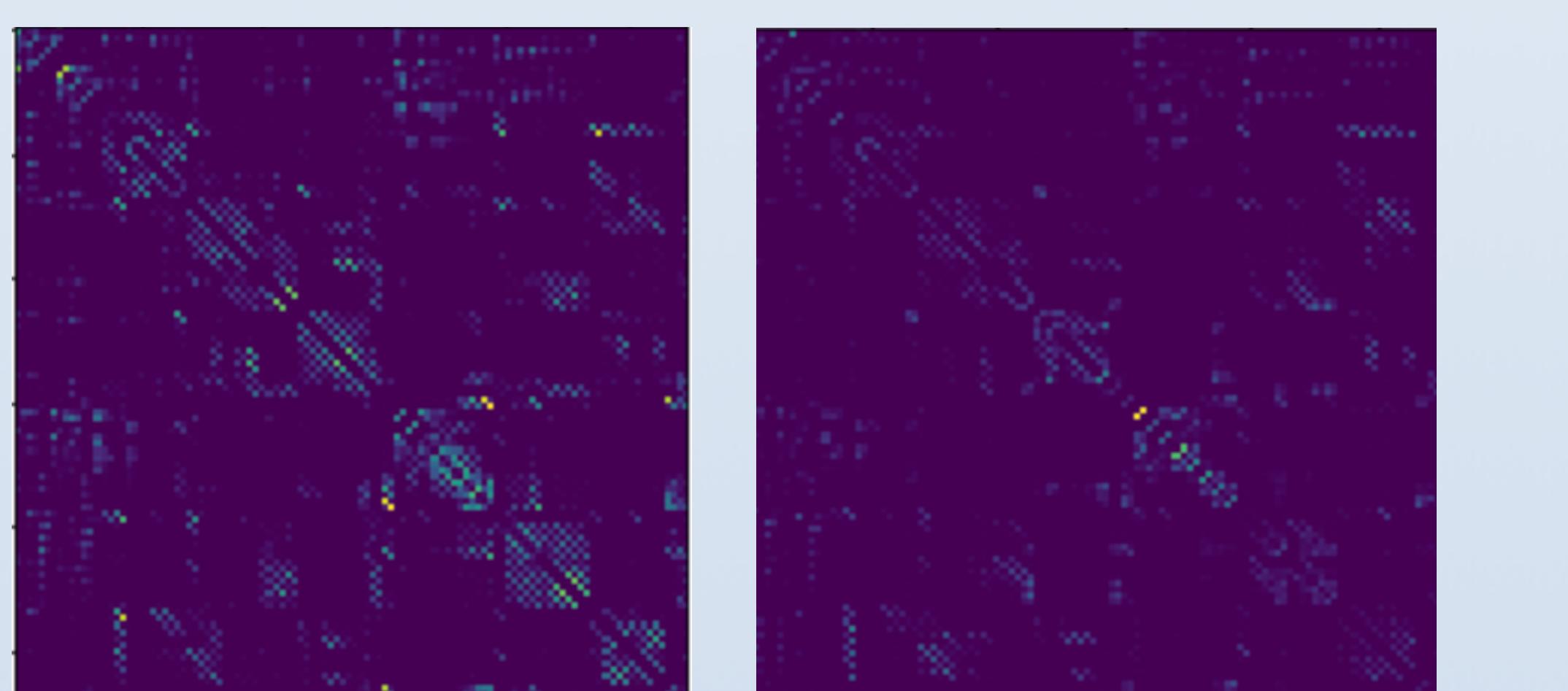
The first step in processing the data was converting the DTI images from the datasets into usable graphical data for the project. The most feasible method of doing so was to convert the DTI images from 3D volumes into 2D connectivity matrices, which would describe the connections between different regions of the brain.

First, I registered each DTI image to their T1 weighted MRI counterpart. I then used FSL to select multiple regions of the brain (noted to be related to Alzheimer's in past studies), and extracted various Regions of Interest (ROIs) from the data. Finally, I performed a tractography analysis to acquire the diffusion data on each scan.



Visualization of the sagittal and axial tractography analyses

The next step of the process of converting the data into a usable format consisted of converting the DTI images into **connectomes**. Connectomes are maps of the connections between regions in the brain that allow brain activity and function to be expressed in a graphical, 2-D format. I utilized various online python libraries such as dipy to import the DTI files into a Jupyter notebook and process them into the connectomes.



Visualization of connectivity matrices

## Model Structure

The network was relatively sparse, to prevent overfitting, and consisted of 3 Graph Convolutional Layers of 32 nodes each, along with 2 Dense Layers and a Dropout Layer. L2 Regularization was used in order to further ensure no overfitting would take place.

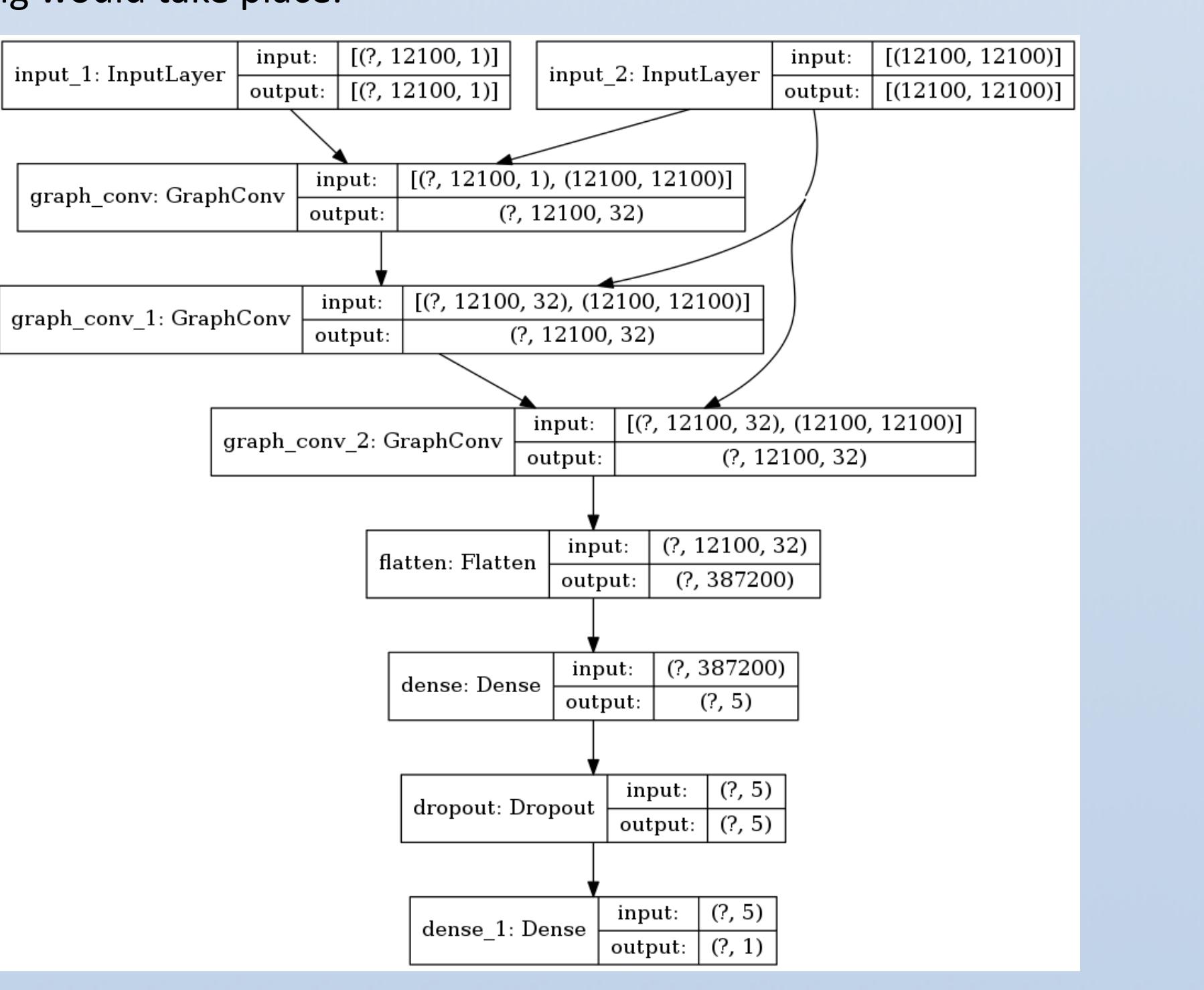
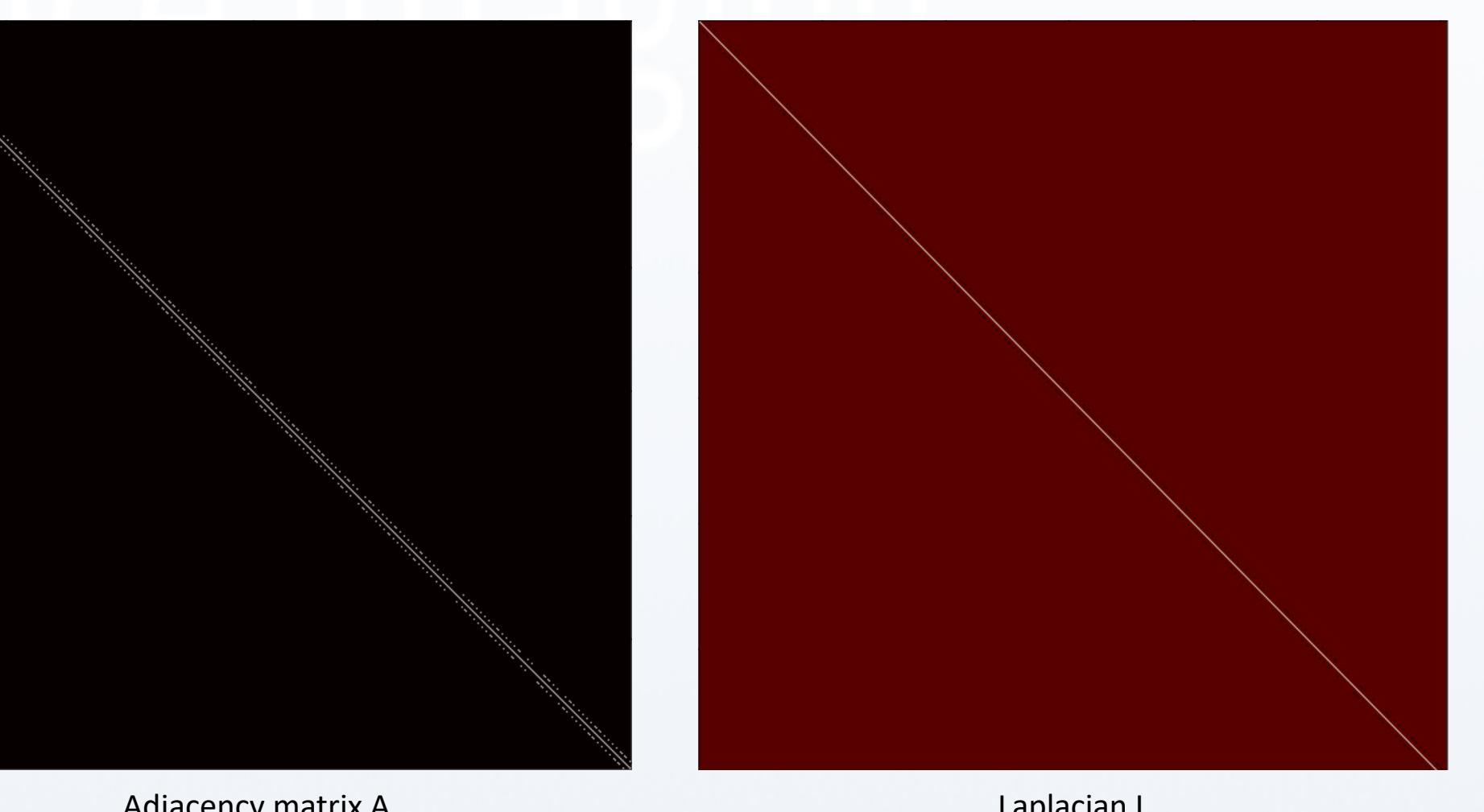


Diagram of model structure

Due to the high intensity computations and amount of data involved in this project, efficient processing units were necessary for training. Various training platforms and model structures were experimented with, including Google Colaboratory. Final iterations of the model were trained on Floydhub (another ML training platform), made use of Floydhub's GPUs and also implemented Tensorboard to track training. The final model was trained on a Tesla V100 GPU for 250 epochs over a period of 5 hours by utilizing weight-saving measures to re-use storage space. The model was trained on an Adam optimizer using a learning rate of 1E-3.



Adjacency matrix A

Laplacian L

I also experimented with various different loss functions in order to improve model classification accuracy. Originally, the model was trained using a Hinge loss function, which is written below.

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Hinge Loss Function (Medium)

Due to the small size of the dataset, the hinge loss was unable to adequately account for extreme predictions made by the model. The loss function used in the final model was the **binary cross-entropy loss** function, which penalizes extreme errors much more heavily than smaller errors.

$$L_{\text{cross-entropy}}(\hat{\mathbf{y}}, \mathbf{y}) = - \sum_i y_i \log(\hat{y}_i)$$

BCE Loss Function (Medium)

## Project Criteria

There were two criteria used to determine the success of this project:

- Creation of a working GNN model that could be trained on DTI connectome data.
- Validation of the model by measuring ROC-AUC (True Positive vs. False Positive rates), F1 score, and accuracy.

Below are the metrics used to evaluate the performance of the model.

Metric	Formula	Expected Range
F1 Score	$2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$	0.0 – 1.0 Perfect = 1.0
ROC-AUC	Total area under ROC curve	0.0 – 1.0 Perfect = 1.0

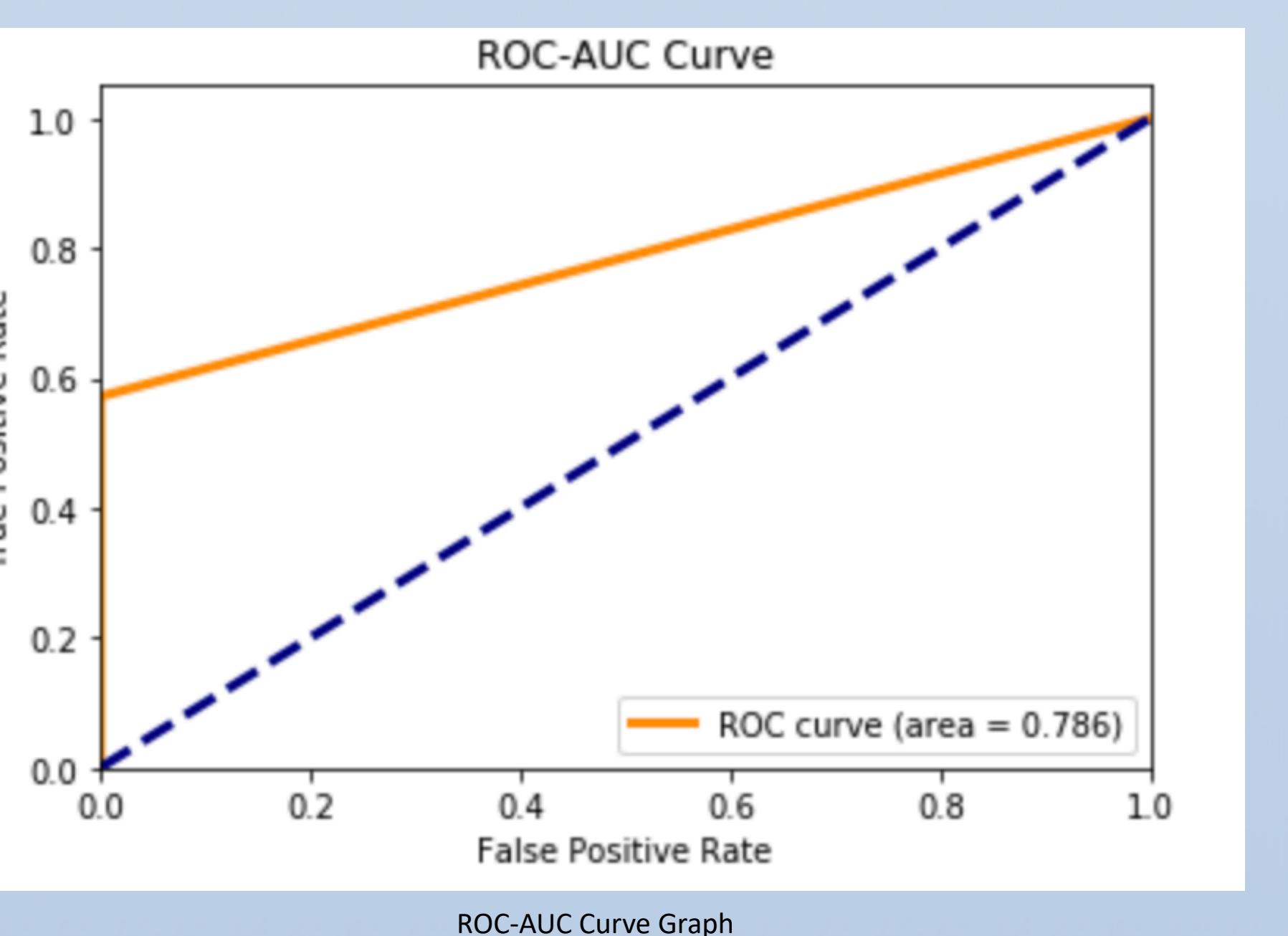
## Results

To objectively compare the performance of a GNN-based model with other previous attempts, I compared my model with a previous CNN-based model baseline. The CNN model was adapted to perform analysis on brain connectome images, without utilizing the graphical nature of the connectome. The results of the model compared to the CNN baseline are shown in the table below.

Model	Metrics		
	Accuracy	F1 Score	ROC-AUC
My model	.735	.727	0.786
CNN Baseline	.682	.714	0.695

The GNN-based model outperformed the baseline model in all categories, with significant improvements on the ROC-AUC metric. These results objectively show the success of the GNN-based approach in comparison with previous baselines.

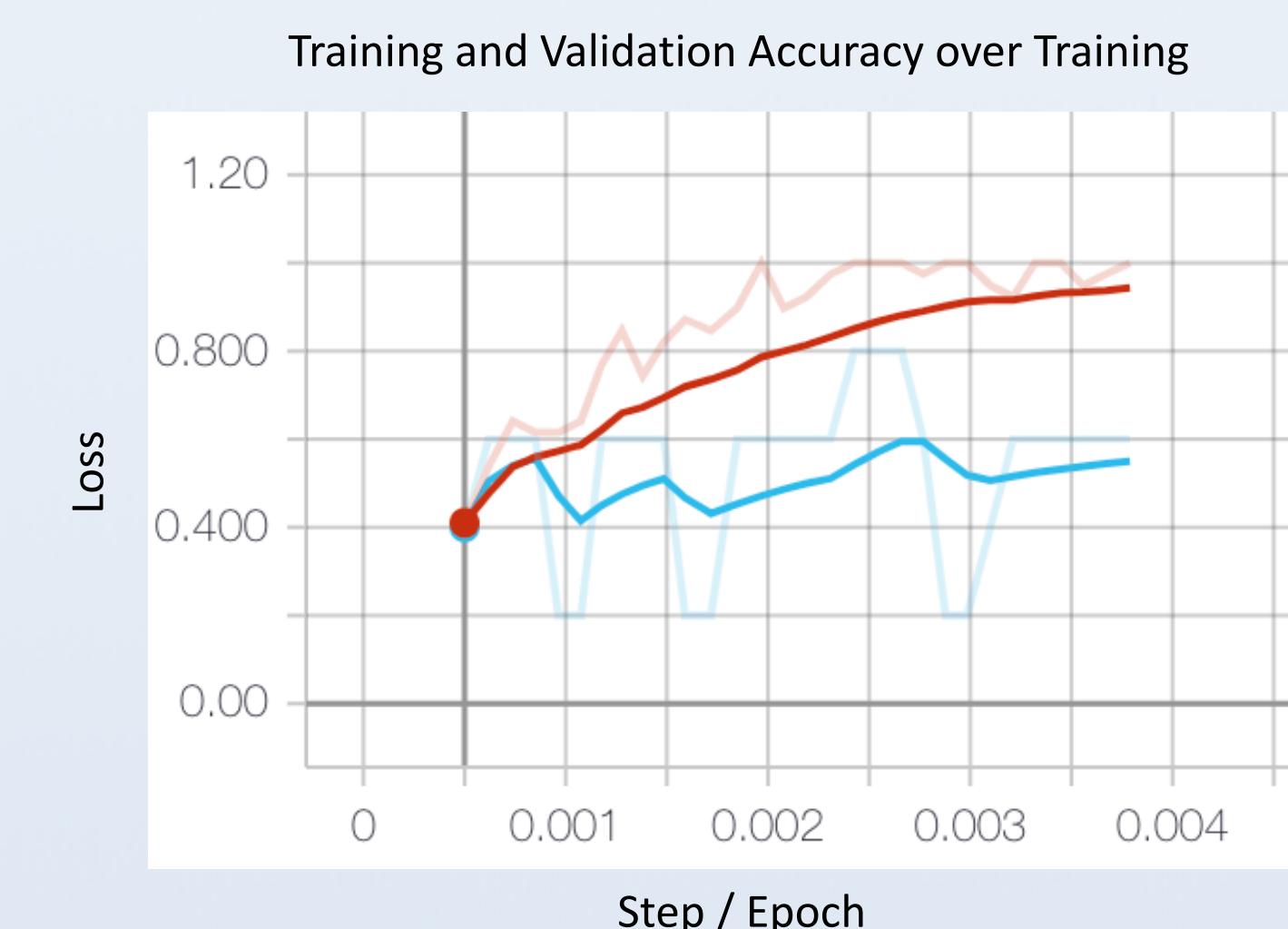
Displayed below is the ROC-AUC curve of the model's performance on the test dataset.



We can observe the success of the model by analyzing the graphs of the loss functions throughout training. In the graph for the loss function (displayed below), we notice an initial stage of wild variation, which eventually smoothens out and stabilizes as is desired. The validation loss, while increasing in the segment shown below, eventually smoothens out as well. The blue line represents validation, and the brown training.



We also witness the same variation in the graph of the accuracy of the model throughout training. The accuracy of the model slowly increases throughout training, and this improvement is similarly reflected in the validation accuracy.



## Conclusion

Overall the project was a success. The project criteria were met, with a successful construction of a GNN based model to predict the onset of Alzheimer's disease. The GNN was measured for accuracy, F1 score, and ROC-AUC, and outperformed the baseline neural network on all statistics. The scientific contributions of this project include exploration of the applicability of deep learning and specifically GNNs to medical imaging problems in lieu of more expensive and computationally intensive methods.

## Project Constraints

There were a few limitations to the project that contributed to the end results. First, due to the amount of computational power involved, a high performance GPU was required to perform the training of the model. Even with this GPU, the model was trained for only 250 epochs. The results could be improved immensely if more time and power was used to train it (in professional and academical ML research, multiple clusters of GPUs are used over periods of months to fully train models). In addition, due to the complexity and rarity of DTI data, there was a significant shortage of data as compared to many other common machine learning problems. With more DTI data specifically acquired from Alzheimer's patients, the accuracy of the model could be significantly improved.

## Future Pathways and Implications

While the project in general was a success, there are multiple future pathways and additions that could be explored. Possible extensions include:

- Experimenting with different ROIs when processing DTI data
- Experimenting with various other filters in the Graph Convolution
- Using more GPUs and time to fully train the model to completion

The implications of this project are far reaching in terms of the clinical applications to Alzheimer's. Typically, the symptoms of Alzheimer's emerge roughly 5-10 years after the tissue in the brain begins degenerating. Using this algorithm, we could feasibly begin the treatment process for Alzheimer's even earlier and slow the progression of the disease. This study could also be useful in determining regions of the brain that are important or impaired when a patient develops Alzheimer's.

## Works Cited

- National Institute of Health. "What Is Alzheimer's Disease?" National Institute on Aging, U.S. Department of Health and Human Services, 16 May 2017, www.nia.nih.gov/health/what-alzheimers-disease.
- Alexander, Andrew, et al. Diffusion Tensor Imaging of the Brain. Diffusion Tensor Imaging of the Brain, www.ncbi.nlm.nih.gov/pmc/articles/PMC2041910/.
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P.S. (2019). A Comprehensive Survey on Graph Neural Networks. *ArXiv*, abs/1901.00596.
- Kawahara, J., Brown, C.J., Miller, S.P., Booth, B.G., Chau, V., Grunau, R.E., Zwicker, J.G., & Hamarneh, G. (2017). BrainNetCNN: Convolutional neural networks for brain networks; towards predicting neurodevelopment. *NeuroImage*, 146, 1038-1049.

All images and figures made by the author with exceptions: 1. OASIS3 and ADNI images (citations in paper) 2. Other citations underneath respective images.