# Analyzing Multi-Layer and Convolutional Neural Networks in Schizophrenia Detection

Walters, Robert
*Department of Computer Science*
*Middle Tennessee State University*
Murfreesboro, TN 37132
rjw3x@mtmail.mtsu.edu

Peterson, Kevin
*Department of Computer Science*
*Middle Tennessee State University*
Murfreesboro, TN 37132
kjp3q@mtmail.mtsu.edu

Curl, Adam
*Department of Computer Science*
*Middle Tennessee State University*
Murfreesboro, TN 37132
amc2bg@mtmail.mtsu.edu

Fleming, Paul
*Department of Computer Science*
*Middle Tennessee State University*
Murfreesboro, TN 37132
pmf2k@mtmail.mtsu.edu

Meagher, Samuel
*Department of Computer Science*
*Middle Tennessee State University*
Murfreesboro, TN 37132
sm8z@mtmail.mtsu.edu

***Abstract*—Neural networks are finding many applications within the healthcare industry. With ever-increasing popularity, the research needed to establish the efficacy of neural network utility must also grow. Our objective is to test two neural network architectures: the multi-layer neural network, and the convolutional neural network. Training speed, network element complexity, and prediction accuracy are determined by compairing the two architectures using the COBRE fMRI dataset for patients with schizophrenia. Current findings show more success within the multi-layer neural network design. However, results do not indicate that one network architecture is inherently better than the other.**

***Index Terms*—neural network, multi-layer, convolutional, COBRE, fMRI dataset**

## I. Introduction

This project was created for Dr. Phillips' Spring 2019 Neural Networks class by our group for the purpose of learning more about Neural Network programming and architecture. A neural network is a computer system that is modeled on the human brain and nervous system. They are composed of interconnected processing elements called neurons that work together to solve a problem. Neural networks are largely trained and used to recognize patterns in data sets similar to the way a human brain would.

Our group decided on teaching two architecturally different nets to detect schizophrenia from fMRI data. The architectures we chose are multi-layer and convolutional neural networks. The learning ability of the two nets will be the primary focus of this research and success will be determined if the networks are able to detect schizophrenia with a reasonable accuracy. To do this, we have two goals we want to achieve. First, we want to obtain an accuracy above 75% in schizophrenia detection for both neural networks. This will provide evidence that neural nets could be effectivly used to detect shizophrenia. Second, we plan to compare and contrast the differences between the nets. We will observe and take note of the following characteristics: training speed, size of input data, number of neurons, number of layers, complexity of preprocessing, complexity of implementation, and overall feasibility of use within an fMRI analysis environment.

## II. Background

### A. Introduction to fMRI and fMRI Data

For the reader's convenience, we will give a brief overview of fMRI as well as the properties of the data used for this project. Functional magnetic resonance imaging (fMRI) is a technique used to map the brain. More specifically, it gives the ability to see neural activity by measuring the brain's blood-flow over a period of time. fMRI involves using two powerful electromagnets and a radiofrequency burst to detect the nuclear magnetic resonance (NMR) of hydrogen protons. This allows for detection of the presence of oxygenated blood at very specific locations. It also indicates neural activity and can be depicted as active neurons draw oxygenated blood [1]. What makes fMRI useful is that certain regions of the brain only activate when performing certain tasks. For instance, different layers of the visual cortex will activate when looking at different images; likewise, different parts of the motor cortex ”light up” when the patient is performing different physical movements. To depict this activity, fMRI data uses the Blood-Oxygen-Level Dependent (BOLD) technique to color code neural activity. The data is broken up into 2D slices that, when combined, create a 3D image of the patient's brain. This 3D image is made up of voxels (volumetric pixels or 3D pixels), each of which represents millions of neurons. Additionally, those images are taken multiple times over several minutes. This translates into very large (sometimes up to thousands of voxels) and noisy (unrelated neurons) data. Processing the data into a format more useful to analyze is not only important, but it can also be a difficult task.

### *B. Tools and Previous Works*

The tools used in this project were Nilearn and keras. Keras provided the means for the construction of the nets [2]. Nilearn was used for our data cleaning and preprocessing, masking our fMRI to a brain atlas, computing the covariance matrix for our masked fMRI data, and generating images and plots of our data [3].

Several research articles provided the resources for the foundation of this project. For building our multi-layer net, the article ”Functional connections from resting-state fMRI” provided us with an outline for cleaning the data and finding regions of interest [4]. We also used their technique for computing a correlation matrix on those regions of interest.

It is also worth mentioning that using neural nets for learning fMRI data is not untreaded territory. In the paper ”Classification of Schizophrenia Versus Normal Subjects Using Deep Learning”, the group used layers of autoencoders called stacked autoencoder (SAE) to make a based two-stage architecture to detect schizophrenia. Using the COBRE dataset they were able to achieve a classification accuracy of 92%.

x

## III. Methods

### *A. Making the Multi-Layer Network*

The data we chose for this project was the COBRE dataset. This dataset contains fMRI data from 72 patients with schizophrenia and 75 without (COBRE). Ages ranged from 18 to 65. Patients were excluded if they had any serious and unrelated mental health issues in the year leading up to the imaging. The data itself measures blood oxygen levels in a 4D matrix (subject#, X, Y, Z, t). An image of the scan can be seen in fig. 1.

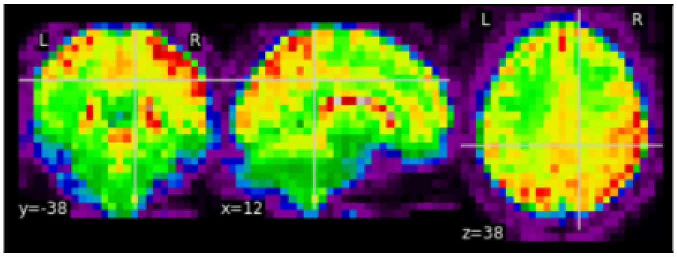

Fig. 1. Raw fMRI scans.

After obtaining our data, we moved on to preprocessing. For our multi-layer network, we approached the data with the goal to reduce it to a statistical format. We started with detrending, standardizing, and then running the data through high and low pass filters. The formatted data remained the same shape, which was still too large. To solve this, we ran it through a mask in Nilearn called MSDL atlas. The tool took the information and broke it into 39 regions (fig. 2.), each containing the averages of the fMRI blood oxygen levels and all of the voxels. Ultimately, it significantly reduced both the size and noise of the data. The data was now in the form: (subject#, region of interest, t).

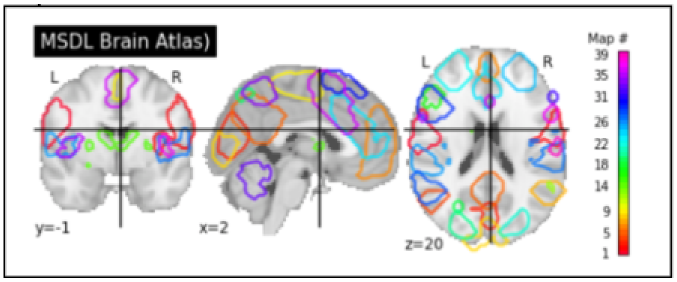

Fig. 2. Mask with brain atlas.

Next, we needed information on how the regions of interest were connected. To do this, we computed a covariance matrix of the masks using the group sparse covariance method (fig. 3.). Now we had covariance values to tell us how connected each of the 39 regions were to one another giving us the array: (subject#, region of interest, region of interest).

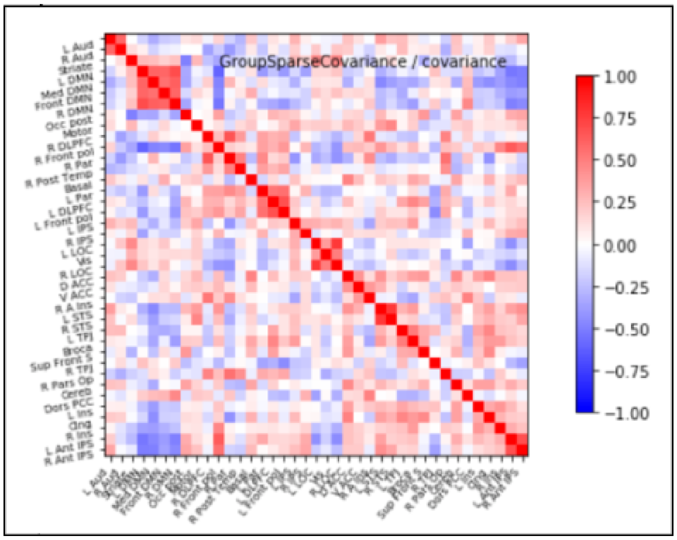

Fig. 3. Covariance matrix of masks from using group sparse covariance method.

Our final modification of the data was to flatten it into a 1D vector. Due to the symmetric characteristic of covariance matrices, only half of the values would be pulled thus making the final array: (subject#, regions of interest flattened) or in our case (145, 741). This array would be the input layer for the multi-layer neural network. It would be followed by a 10% dropout layer to get better generalization. This layer is followed by a hidden layer containing 350 parameters which use the recitifed linear unit (ReLu) activation function alongside batch normalization. The output layer uses binary cross-entropy loss and sigmoid activation. The network's output is a single binary response, telling us the network's classification of whether or not the patient had schizophrenia. Fig. 4. shows the net in its totality.

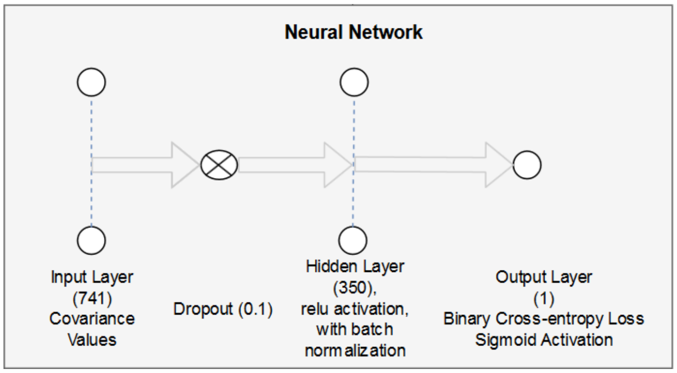

Fig. 4. Architecture of the multi-layer network.

### B. Making the Convolutional Network

Our convolutional network used a similar approach for preprocessing, up to the step where we fit to the MSDL atlas, which output data in the form (subject#, region of interest, t). From there the input data was split into a training and a testing set as inputs to the network. The network itself consisted of ten layers - most notably, two 2D convolution layers - a larger layer followed by a smaller layer, two Batch Normalization layers, and two Dropout layers. The purpose of the convolution layers was to find individual features within the region of interest with respect to time. The purpose of the batch normalization and dropout layers was to help prevent the neural network from becoming unstable. Also included were two pooling steps, a flatten step, and two simple feedforward nets for the purpose of pulling down the number of features learned into a final binary yes-or-no output. This would, in theory, allow the network to start with a large number of 2D features to detect from relations between the intensity value of the region of interest as it varies over time, which it could then combine and reduce down to a final 1D pattern which would decide if the patient did or did not have schizophrenia.

## IV. Results

To measure the quality of the multi-layer network, we used K cross validation to get the accuracy over multiple class label splits. Our results were an average of 77.87% accuracy with k=10. The convolutional net was not quite as successful. It suffered from significant overfitting preventing any meaningful accuracies. Figures 5. and 6. show the loss and accuracy from the multi-layer and convolutional models respectively.

## V. Discussion

Schizophrenia was most accurately predicted using a feed-forward multi-layer neural network with an individual covariance matrix input for each subject. This method outperformed a convolutional 2D neural network with masked fMRI signal input. These results may be explained by the fact that covariance matrix inputs take into account the connectivity of different brain regions, while the masked fMRI signal input does not consider connectivity of the brain.

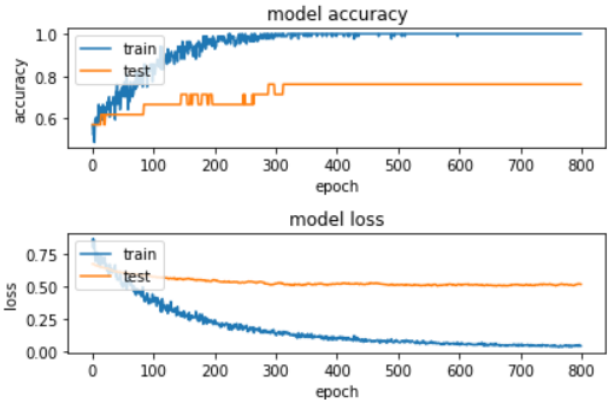

Fig. 5. Multi-layer model accuracy and loss.

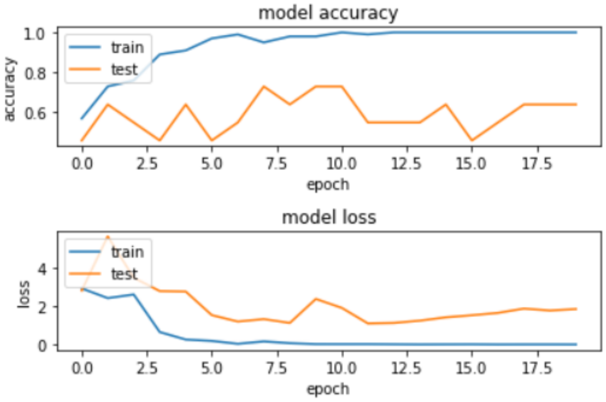

Fig. 6. Convolutional model accuracy and loss.

Functional connectivity is often cited in the literature as a helpful tool for fMRI decoding and classification problems [4]. It has also been found that brain functional connectivity strength is often reduced in patients with schizophrenia [5]. Schizophrenic patients have a more diverse mode of connectivity to the rest of the brain that is less coordinated than non-schizophrenic patients. For this reason, the covariance functional connectivity matrices calculated on the COBRE dataset allowed the neural network used in this project to pick up on these differences in schizophrenic and non-schizophrenic connectivity. Our results confirm these findings in the literature in that connectivity matrices provided the best input for neural network classification.

It is feasible that the convolutional 2D method would have been able to produce a similar accuracy to the covariance method if given enough training time. In the literature, convolutional neural networks have been used to automatically recognize functional connectivity [6]. This method was limited by the computation time and space that was available in this project. While this method of classification did not perform better than chance, we cannot confidently say that this method cannot perform well. One direction that could potentially be explored in the future is the use of convolutional networks on functional connectivity matrices. While this project yeilded good results with a regular multi-layer network, perhaps convolutional networks would be able to more accurately pick up on patterns in the functional connectivity of the brain.

## References

[1] X. Yang, "Machine learning techniques for high-dimensional neuroimaging data analysis," 2016.

[2] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[3] A. Abraham and e. a. Pedregosa, "Machine learning for neuroimaging with scikit-learn," *Frontiers in Neuroinformatics*, vol. 8, p. 14, 2014.

[4] B. C. e. a. Smith SM, Vidaurre D, "Functional connectomics from resting-state fmri," *Trends Cogn Sci*, vol. 12, pp. 666–682, 2013.

[5] K. R. e. a. Lynall ME, Bassett DS, "Functional connectivity and brain networks in schizophrenia," *J Neurosci*, vol. 30, pp. 9477–9487, 2010.

[6] S. Z. e. a. Yu Zhao, Qinglin Dong, "Automatic recognition of fmri-derived functional networks using 3-d convolutional neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 65, pp. 1975–1984, 2017.