ALZHEIMER'S DISEASE PREDICTION USING GRAPH CONVOLUTION NETWORKS

Project Reference No.: *45S_BE_2459*

College ***:*** *University Visvesvaraya College of Engineering, Bengaluru*
Branch ***:*** *Department of Computer Science Engineering*
Guide(s) ***:*** *Prof. Prathibhavani P M*
Student(S) ***:*** *Mr. Stuart S*
Mr. Naveen Narayan Bhat
Mr. Harish Y
Mr. Prashanth

Keywords:

Graphs, Graph convolutional networks, Semi-supervised classification, Alzheimer's disease

Introduction:

To develop a software built using Graphical Convolution Networks(GCN) and CNN to perform multi-class classification of the MRI images that will detect the presence and severity of Alzheimer's disease in the early stages of the entire diagnosis procedure. Alzheimer's disease (AD) is a degenerative disorder of the brain that leads to memory loss. It is the most common form of dementia, caused by the build-up of beta amyloid plaques in the brain. The plaques and tangles are some of the main features of the disease.

As the number of Plaques and tangles increases, the healthy neurons begin to function less effectively and gradually lose their ability to communicate and finally die which results in overall shrinkage of the brain tissues. The death of neurons particularly in the hippocampus restricts the ability to form new memories. The hippocampus is the first region in the brain which gets affected. It is the region in the brain that is responsible in forming memories and serves as a relay structure between the brain and the body.

Objectives:

To develop a software to detect Alzheimer from MRI scans and differentiate between different levels of severity of Alzheimer's. To provide an assistive software the doctors can use to minimize the time spent on early stages of diagnosis. To provide a relatively quick way to rule out.

Methodology:

The advent of Convolutional Neural Networks as powerful models that exploit both image features and spatial context by means of neighbourhood information to yield hierarchies of features, has led to their application in numerous different problems related to 2D and 3D images, like image segmentation [2] and classification [3], long before their recent reemergence [4]. There is a direct analogy between an image segmentation task, where each pixel is to be assigned a label e.g. tissue of interest or background, and a subject classification task within a population e.g. for disease prediction. In the latter case, a subject along with its corresponding

feature vector is equivalent to an image pixel with its intensity values for the different channels, while a graph constructed based on pairwise population similarities is equivalent to the pixel grid, for which the notion of proximity is more straightforward, since both describe the neighbourhood structure for convolutions. Nevertheless, the traditional widespread formulation of CNNs for regular domains cannot be directly extended to irregular ones. The description of the local neighbourhood structure and node ordering is not straightforward for irregular graph structures and these need to be properly defined to allow convolution and pooling operations. The first method dealing with neural networks on graphs[5]. In this pioneering work, the authors devised a mapping function from graph space to an m-dimensional Euclidean space, and proposed a supervised learning method to learn the parameters of their graph neural network (GNN) model. However, no convolution was considered in this model. The first work to introduce convolutional neural networks on graphs was described in [6]. used concepts from the emerging field of signal processing on graphs, giving rise to the spectral graph convolutional networks. Spectral GCN is a generalisation of CNNs to irregular domains, which uses computational harmonic analysis to process signals observed on irregular graph structures. The concepts borrowed from this field allow the extension of CNNs to irregular graphs in a principled 2 way, by treating convolutions in the graph spatial domain as multiplications in the graph spectral domain. This kind of convolutions have several applications in computer vision, computer graphics and social network problems, among others, and have successfully been adopted to perform classification of documents in large citation datasets.

Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the Principal Components. It is one of the popular tools that is used for exploratory data analysis and predictive modeling. It is a technique to draw strong patterns from the given dataset by reducing the variances. Steps Involved in the PCA

Step 1: Standardize the dataset.

Step 2: Calculate the covariance matrix for the features in the dataset.

Step 3: Calculate the eigenvalues and eigenvectors for the covariance matrix.

Step 4: Sort eigenvalues and their corresponding eigenvectors.

Step 5: Pick k eigenvalues and form a matrix of eigenvectors.

Step 6: Transform the original matrix.

Results and Conclusions:

The classifier yields an average training accuracy of 94.57 % and a testing accuracy of 87.23 %, averaged across the three classes, over ten random trials.This project will propose a software to detect Alzheimer's from MRI scans and differentiate between different levels of severity of Alzheimer's to assist doctors in early diagnosis.They are, ‘NonDemented’, 'MildDemented', ‘VeryMildDemented’, and ‘ModerateDemented’. The project Metrics are Epoch=50, training samples=672,validation samples=288 for alzheimer.The report influence of the feature selection scheme on the ABIDE database for the four considered methods, i.e. RFE, PCA, MLP and Autoencoder (AE). The project train and evaluate a GCN using RFE, MLP and

Autoencoder for C = {250, 500, 1000, 2000} number of features. For PCA, results for C = {250, 500} which correspond to approximatively 85% and 95% of explained variance. Last but not least, it reports that the result for C = 6105, which corresponds to using the whole feature vector as input. AE is trained for 150 epochs with a learning rate of 5e−4, while the scikit-learn implementation for GCN with default parameters is used for MLP and PCA. Project reports classification accuracy as well as Area Under Curve (AUC) for the 10 different cross validation folds. The first observation is the very poor performance of PCA as a feature selection strategy.

As mentioned this approach, although very popular and efficient for the dimensionality reduction task, is limited to finding a linear projection of the features to the lowerdimensional space. Therefore, it is not as expressive as a non-linear method, like the Autoencoder. Additionally, since the ABIDE dataset is highly heterogeneous, the variance captured in the training set is likely not as representative of the test set.MLP has the best performance with respect to other methods for 250 features but drops in performance when more features are added. This can be explained by the high tendency of the MLP classifier to overfit in our particular set up (notably due to our limited number of input samples). Features are learned from the training set, therefore optimised for classification of this specific subset of the data, which in turn, leads to overfitting when training the GCN model. Autoencoders do not have this tendency, and our AE model has the best performance for 500 and 1000 features. RFE obtains the overall best performance for C = 2000 features, and its performance increases as the number of features.