Classification of Alzhermer Disease and Normal Congnitive Status with Recurrent Neural Networks in Resting State fMRI

Tao Sun

Ohio University

2016

Classification of Alzhermer Disease and Normal Congnitive Status with Recurrent Neural Networks in Resting State fMRI

Introduction

A human brain is a complex system composed of structural regions that are functionally specialized. Due to the conclusion that these locally segregated regions are actively interconnected even when a subject is at resting-state (Biswal, Yetkin, Haughton, & Hyde, 1995), the resting-state functional Magnetic Resonance Imaging (rs-fMRI) , which is a neuroimaging procedure that measures the changes of signals associated with blood flow, has become a prevailed tool for investigation of brain functional networks. Since functional connectivity in the brain is an significant measure that could indicates disease-induced changes in the network, it could provide assist to the diagnosis of brain diseases such as Alzheimer Disease (AD) or its early stage Mild Cognitive Impairment (MCI).

With the typical assumption that the functional networks in a brain is stationary, many rs-fMRI based diagnosis methods of MCI and AD model the network with correlation analysis such as Pearson's correlation, Independent Component Analysis (Li et al., 2012). However, recent studies (Hutchison et al., 2013) suggest that significant temporal changes exist in functional connectivity. Thus, valuable information could be lost when connectivity estimation is solely based on analysis restricted to a single value obtained from the entire scanning time.

In this paper, we present a novel framework to provide diagnosis assist for AD by combining Deep Auto-Encoder (DAE) and Recurrent Neural Networks (CNN). Initially in the framework, rs-fMRI images data is preprocessed and mean time series of Regions of Interest (ROIs) are extracted. Then high-dimensional time-series is reduced to a lower dimensionality by a DAE, which in succession is splitted into multiple identical-sized sub-series. A RNN classifier is trained on the sub-series classifying each of them as either AD or Normal healthy Control (NC). Finally, the diagnosis suggestion for a subject is made by ensemble of the outputs of the sub-series classifier. Tests show that accuracy of the framework approaches 70% on test data.

Problem Definition and Algorithm

Data set and Preprocessing

The data used for training and test of the proposed classifier are retrieved from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database. After filtering, images of 33 AD subjects and 50 NC subjects, are downloaded. With most of these subjects are scanned more than once, we have 89 AD examples and 139 NC examples in the data set.The data set are divided into training data, test data, and validation data with a ratio of 7:2:1.

The preprocess can be divided into 4 phases ((Suk, Weeb, Leea, & Shen, 2016)) :

1. preprocessing of anatomical images;
2. preprocessing of functional images;
3. anatomical standardization of functional images;
4. removal of noise signal.

The results of the preprocess are a set of mean time series

$$F^{(n)} \in \{F|F = [\mathbf{f}_1, \cdots, \mathbf{f}_t, \cdots, \mathbf{f}_T], \mathbf{f}_t \in \mathbb{R}^R\}, n = 1, \cdots, N \ ,$$

where $N = 228$ is number of the scans, $R = 120$ is the number of ROIs, and $T = 135$ is the length of a time series.

Dimensionality Reduction with DAE

Suk et al. (2016) proposed that a stacked DAE can used as an intermediate building block for deeper models in neuroimaging analysis. DAE is an unsupervised multilayer feed-forward neural networks, the goal of which is setting a latent representation of feature vectors of its input by training a nonlinear approximation function $h(\mathbf{f}_t) \approx \mathbf{f}_t$ (Figure 1). Concretely,

$$G = [\mathbf{g}_1, \cdots, \mathbf{g}_t, \cdots, \mathbf{g}_T], \mathbf{g}_t \in \mathbb{R}^R$$

is expected to converted from its original form

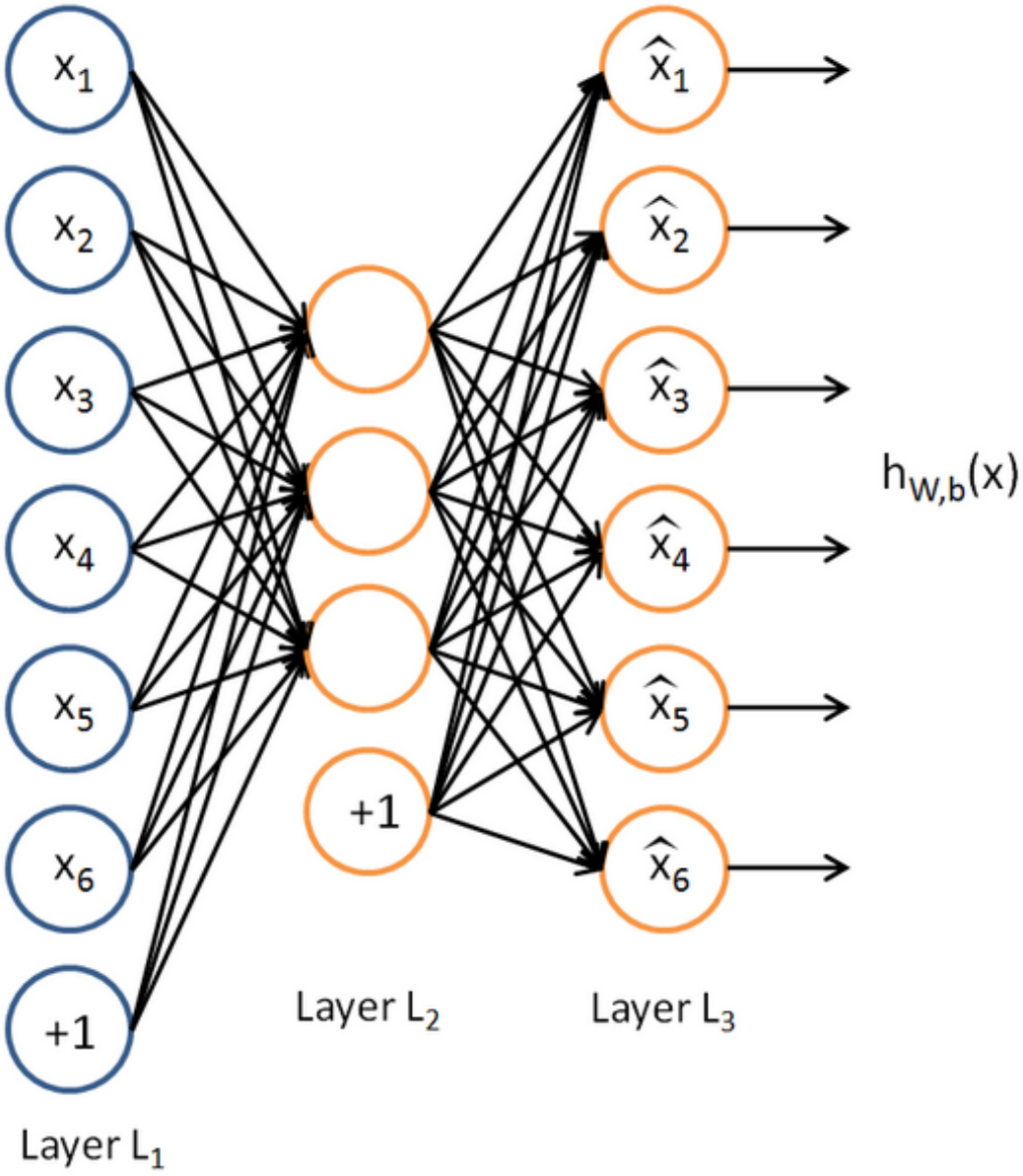

Figure 1. Autoencoder.

$$F = [\mathbf{f}_1, \cdots, \mathbf{f}_t, \cdots, \mathbf{f}_T], \mathbf{f}_t \in \mathbb{R}^R.$$

A stacked autoencoder (Figure 2) is composed of multiple layers of autoencoders, in which the outputs of each layer is fed as the inputs of successive layer ('Stacked Autoencoders', n.d.). Usually greedy layer-wise training is applied to train a stacked autoencoder. In current implementation, training for the stacked model is similar to the non-stacked one. Least-square loss function with is selected to train the stacked model, in which the difference is directly computed between input layer and final target layer. Although that setting of the hidden layer configure is heuristic, the combination in Suk

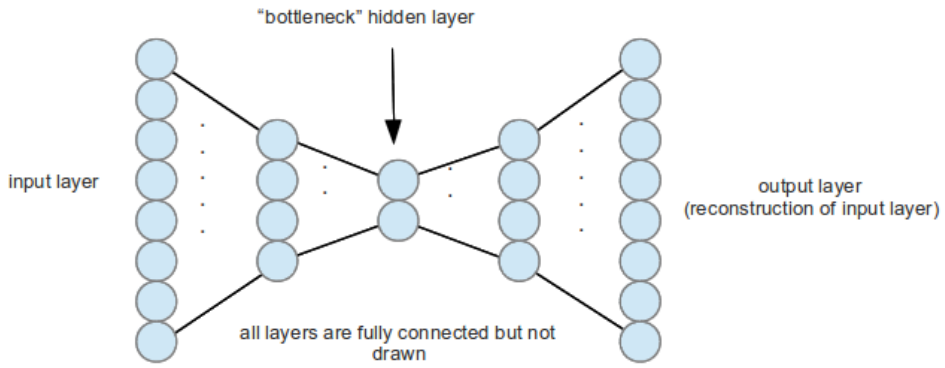

Figure 2. Stacked autoencoder.

et al. (2016)'s paper is followed.

After training, only the first part of the DAE (i.e. from input layer to bottleneck hidden layer) is used to transform each $\mathbf{f}_t \in \mathbb{R}^R$ into $\mathbf{x}_t \in \mathbb{R}^r$, where $r < R$. As a result, the encoded representation of a scan becomes

$$X = [\mathbf{x}_1, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_T] \,.$$

High-level RNN classifiers and their ensemble for subject diagnosis

Wee, Yang, Yap, and Shen (2015) proposes a framework for brain functional connectivity analysis, in which the time series of a scan are decomposed into multiple overlapping sub-series by a sliding window. Justified by their work, an encoded time-series is splitted into identical-sized sub-series

$$X = [\mathbf{x}_1, \cdots, \mathbf{x}_s, \cdots, \mathbf{x}_S], \text{ where } T = n * S, n \text{ is an integer.}$$

A RNN is a specialized class of neural network that is suitable for dynamic temporal sequences. Connections between units in a RNN form a directed cycle. For each unit, hidden nodes are created as internal memory to process sequences of inputs, which enables RNN to condition the model on all previous units in a sequence (Figure 3). Below are the formulas in the network.

$$S_k = \sigma(W^{(rec)} S_{k-1} + W^x x_k)$$

$$y = softmax(W^{(s)} S_k)$$

The detailed notations are explained below:

– $\mathbf{x}_k \in \mathbb{R}^r, k = 1, \cdots, n$: the input of a unit

– S_k: current hidden state

– $W^{(rec)}$: weights matrix used to condition the hidden state of the previous time-step

– W^x: weights matrix used to condition the input of a unit $\mathbf{x}_k$

– $\sigma()$: the non-linearty function ()

– y: predicted class for the sequence

– $W^{(s)}$: weight matrix that transform S_k to y

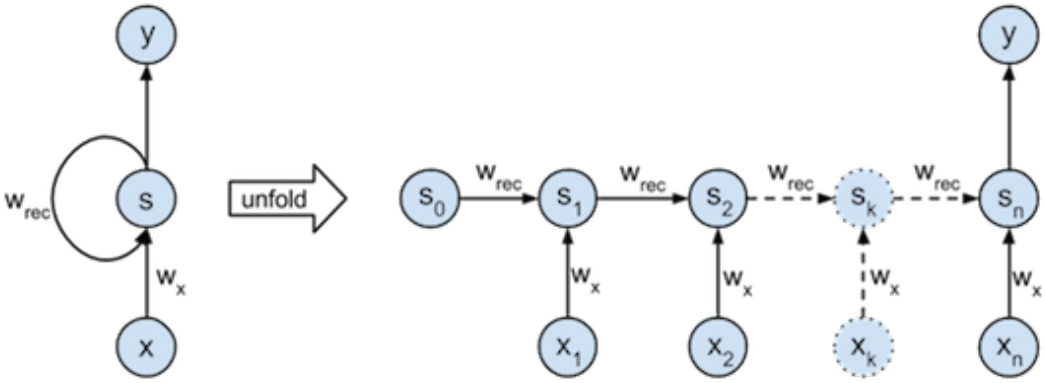

Figure 3. Recurrent neural networks.

In terms of the sub-series setting, data in each time point of a series $\mathbf{x}_t$ become the input of a unit. The final output at the end of the sequence is the predicted class (AD or NC) for the whole sub-series. For the purpose of diagnosis of a subject, ensemble learning is used to predict the class for the time-series of a scan. Each high-level classifier votes with equal weight and the class with the most votes is selected as the class of a time series. For the hight-level classifier, Adadelta optimizer is applied to replace mini-batch Gradient Descent Optimizer. Two regularization techniues, dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) and L2 regularizer, are used to reduce overfitting. In training, 20% cell input units and 50% hidden node units are dropped out when computing S_k and y respectively. In testing, the corresponding two weights, i.e. $input_weights$ and $output_weights$, are scaled down the same versions. In addition, each weight and bias in the RNN model is associated

with a L2 regularizer in the cross-entropy loss function.

Experimental Evaluation

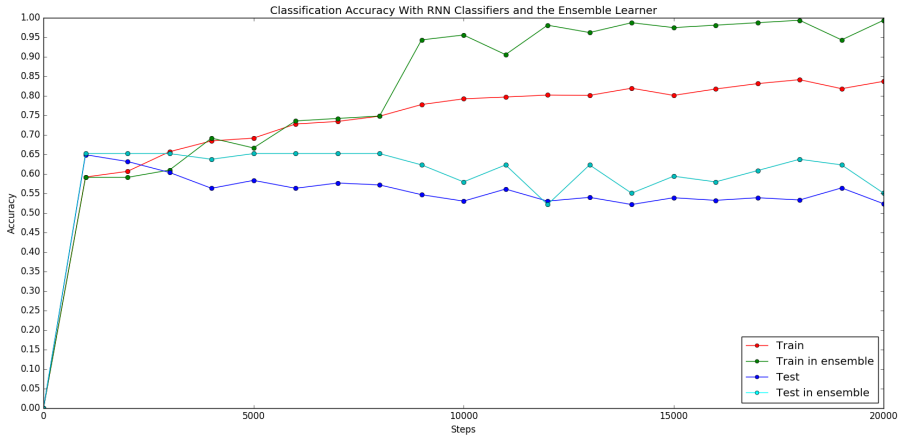

Figure 4. A typical training case.

Figure 4 is a typical accuracy plot when training the RNN model. In this experiment, the length of the time series is set as 135, which is splitted into 15 segments with each sub-series length 9. The vector size of each time point in the time series (i.e. the output dimension of the DAE) is 5. Each batch of 15 examples is fed into the Adadelta optimizer with its initial learning rate 1. We choose tanh activation function, which apparently accelerate the learning in the series of experiments. Both accuracy for sub-series and ensemble accuracy are reported in the graph. Compared to mini-batch Gradient Descent Optimizer (which is not shown here), Adadelta optimizer shows better converge property for this classification task. After 200,000 steps (about 58 epoches), the ensemble training accuracy arrives 100% although the training accuracy is just beyond 80%. In the steps before 5000, there is a obvious platform for the training accuracies, which might be brought by neuron saturation. The testing performance reaches its maximum value (around 65%) very quickly in the training. However, it begins to drop since training accuracy begin a relative rapid increase period. This implies that early stop technique need to be employed in this classification task although dropout technique already makes significant improvements in preventing

overfitting. The ensemble accuracies for the testing data are always higher than their corresponding accuracies without ensemble. Since data fed into the high-level classifiers are not dependent, these accuracies are much lower than what are expected.

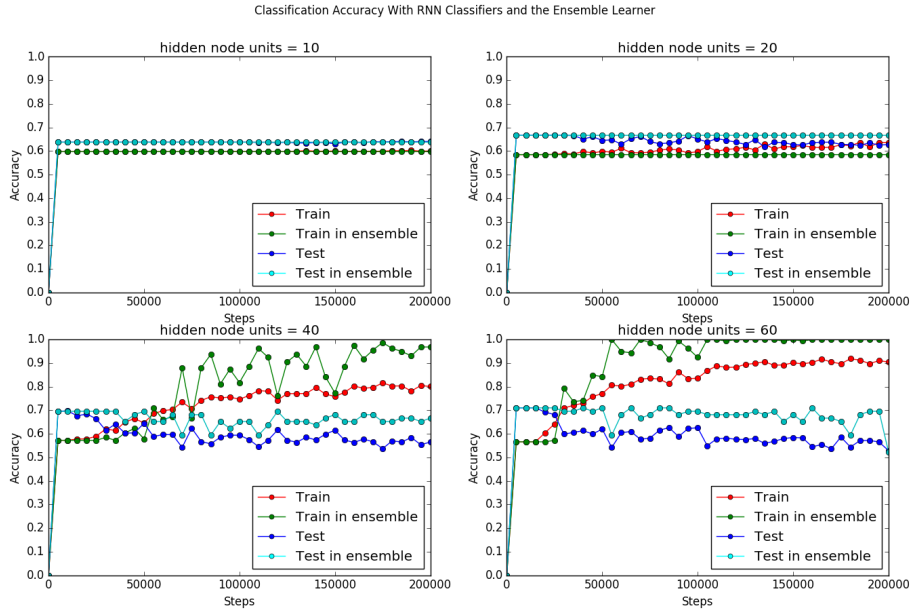

Figure 5. The influence of hidden node units number on accuracy.

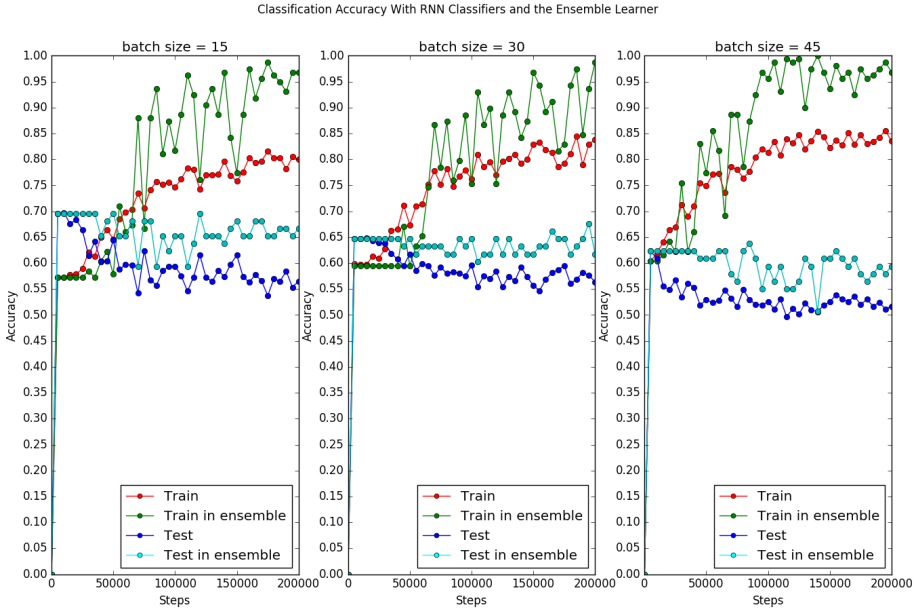

Figure 6. The influence of batch size on accuracy.

It is proved by the experiments that number of hidden note units have substantial

influence on the final testing accuracy, which can be indicated Figure 5. In this graph, model with 4 different hidden node configuration is plotted. It can be concluded that the learning speed raises with the increase size of hidden units. Even in the 10 hidden node units case, no obvious improvements can be identified between training steps. In terms of testing accuracy, the 60 units case does not have essential enhancement comparing to the 40 units.

Figure 6 shows the influence of batch size on the accuracy. All the hyper-parameter settings but the batch-size are the same, which shows that relatively low batch size setting can lead to better testing accuracies.

Related Work

To model temporal dynamics of a brain network, (Eavani, Satterthwaite, Gur, Gur, & Davatzikos, 2013) proposes a Hidden Markov Model (HMM) framework, which associates discrete hidden states with distinct connectivity patterns. (Suk et al., 2016) builds a probabilistic model combining Deep Auto-Encoder (DAE) and HMM to model functional dynamics in rs-fMRI and estimate the likelihood of a subject as belonging to Mild Congnitive Impairment (MCI) status or NC. After preprocess, the model traines a DAE by stacking multiple RBMs for the purpose of dimension reduction. With low-dimensional data, two separate HMM models are trained for the classes of NC subjects and MCI subjects respectively. During testing, the class of the model with higher likelihood is taken as a clinical decision. These two models shares a common hypothesis, i.e. it is possible to "decode connectivity dynamics into a temporal sequence of hidden network 'states 'for each subject"(Eavani et al., 2013). However, the authors does not fully justified the hypothesis. Actually, for the diagnosis purpose, these hidden states and their definitions are not relevant.

Future Work

Currently identical-sized sub-series data is used in the RNN classifiers. In the future, it could be replaced by the more generalized overlapping sliding window data .Compared to current solution, sliding window setting could generate richer data. Since

there are 2 hyper-parameters, sliding window size and stride size, that could potentially be related to the classification performance (Wee et al., 2015), we can tune it by experimenting reasonable combinations of the two parameters. After obtaining more training data, we can improving our RNN traning by tring more sophisticated model such as LSTM. In terms of ensemble learning, there are much room for us to improvement. E.g. In the application of ensemble learning for AD diagnosis, Liu, Zhang, and Shen (2014) employ a greedy approach to select high-level classifiers. (Shi, Chen, Zhang, Smith, & Liu, 2017) trains a stacked DAE by combining sparse autoencoder and denoising auto-encode, which also can be one of our potential DAE solutions.

Conclusion

In this project, a new model is proposed to classify subjects with AD and NC subjects. Specially, we devise a hybrid architecture by involving RNN into the final decision process. To overcome the constraints of limited data, the high dimensional data output by the previous step is splitted into multiple identical-sized segments. We evaluates the performance of the proposed and analyzed several influential tuning decisions. As part of our future work, we hope to substitute the overlapping sliding window for identical-sized segments to further improve the performance of our model.

References

Biswal, B., Yetkin, F. Z., Haughton, V. M., & Hyde, J. (1995). Functional connectivity in the motor cortex of resting human brain using echo-planar mri. *Magnetic resonance in medicine*, *34*, 537–541.

Eavani, H., Satterthwaite, T. D., Gur, R. E., Gur, R. C., & Davatzikos, C. (2013). Unsupervised learning of functional network dynamics in resting state fmri. In *International conference on information processing in medical imaging* (pp. 426–437). Springer Berlin Heidelberg.

Hutchison, R. M., Womelsdorf, T., Allen, E. A., Bandettini, P. A., Calhoun, V. D., Corbetta, M., ... Chang, C. (2013). Dynamic functional connectivity: promise, issues, and interpretations. *Neuroimage*, *80*, 360–378.

Li, Y., Wang, Y., Wu, G., Shi, F., Zhou, L., Lin, W., & Shen, D. (2012). Reasoning with exceptions: an event-related brain potentials study. *Neurobiology of aging*, *33*(2), 427.e15–427.e30.

Liu, M., Zhang, D., & Shen, D. (2014). Hierarchical fusion of features and classifier decisions for alzheimer's disease diagnosis. *Human Brain Mapping*, *35*(4), 1305–1319.

Shi, B., Chen, Y., Zhang, P., Smith, C. D., & Liu, J. (2017). Nonlinear feature transformation and deep fusion for alzheimer's disease staging analysis. *Pattern Recognition*, *63*, 487–498.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Stacked Autoencoders. (n.d.). Retrieved from http://ufldl.stanford.edu/wiki/index.php/Stacked_Autoencoders

Suk, H.-I., Weeb, C.-Y., Leea, S.-W., & Shen, D. (2016). State-space model with deep learning for functional dynamics estimation in resting-state fmri. *Neuroimage*, *129*, 292–307.

Wee, C.-Y., Yang, S., Yap, P.-T., & Shen, D. (2015). Temporally dynamic resting-state functional connectivity networks for early mci identification. *Brain Imaging and Behavior*, 1–15.