Leveraging Temporal Information to better understand Alzheimer's Disease Diagnosis

Yanpei Tian
Stanford University
yanpeit@stanford.edu

Yanhao Jiang
Stanford University
jiangyh@stanford.edu

Zixuan Liu
Stanford University
zucks626@stanford.edu

1. Introduction

Magnetic resonance imaging (MRIs) on human brain have been one of the most critical method for diagnosing neurological diseases like Alzheimer's Disease (AD), as an extremely important indicator of AD is that there's usually a much faster brain deterioration trend presented for patients' brain compared to the normal ones. While modern ML technique gains success on analyzing MRI images to help disease diagnosis, one remaining challenge is to study on Longitudinal MR brain image dataset. New difficulty lies on how to extract the temporal information along a time series data to find the brain's structural change for both AD patients and normal individuals reflected by MRI images. Find such a common pattern for both healthy people and AD patients can potentially leads to new insights of disease diagnosis.

We plan to use a hybrid model involving CNN for individual image spatial feature extraction and RNN for temporal analysis across a series of images. Such a architecture allows us to better utilize the temporal information presented in our dataset. Meanwhile, to achieve better temproal prediction consistency, we will explore new techniques such as new regularization loss terms to help maintain consistency among data over time. To evaluate our method, an AD/non-AD classification task will be run to see if higher accuracy is reached.

2. Problem Statement

As mentioned in the introduction, we propose a hybrid model (CNN-RNN) that can extract temporal information over a time series data to diagnose Alzheimer's Disease.

2.1. Dataset

We will use 3D MRI data in Alzheimer's Disease Neuroimaging Initiative (ADNI) [1] dataset to study the long-term progression of Alzheimer's disease. The dataset contains 811 subjects (patients/control), with around 198 AD subjects, 229 Normal Control subjects, and 384 MCI subjects. Each subject will have 1-7 MR images, with time range from 3 months to 48 months. To better acquire the trend of brain structure/function deterioration, we will include images at time 0 month(baseline), 12 months, 24 months, 36 months, 48 months (if available) in our training data.. After data processing procedure, the format of each MR image will be 64*64*64.

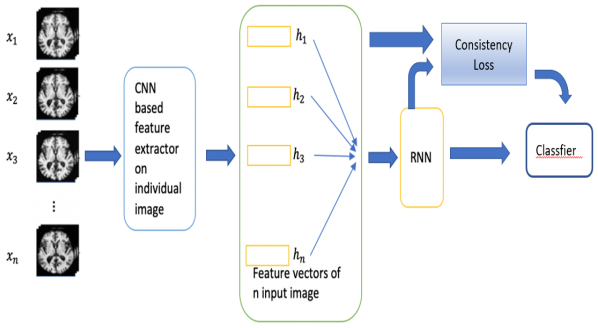

Figure 1. High-level model diagram. A time series data is first fed into a CNN, which extracts spatial information related to AD. The feature vectors are then fed into a RNN structure to generate the final prediction. We also plan to add a regularization term in the training procedure to ensure the consistency of predictions for one subject over time.

Data preprocessing and augmentation 1. Balanced training dataset with roughly equal number of Normal/AD; 2. Data normalization (zero mean and unit variance) to facilitate training; 3. Data augmentation by introducing small random rotation and translation.

2.2. Expected Results

We plan to run two groups of models:

- Plain classification model that only look at one image at a time. In this setting, we don't leverage any temporal information, yet we do consider the confounder that the correlation of images sampled from same subject(patient) is really high. In order to avoid this con-

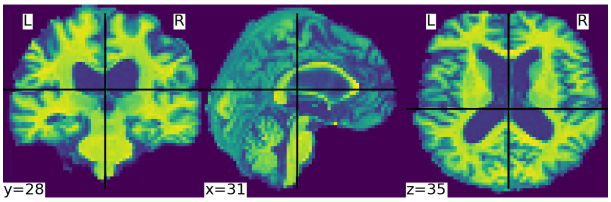

Figure 2. Examplary MRI image in the training dataset, 3D image visualized by NiBabel.

founder, instead of randomly spliting the dataset on image-level, we split the dataset on subject-level, images sampled from same subject will appears in the same split. This setting is much closer to the reality, while the variance of brain structure between different people might be much larger than the change via time/ other factors such as diseases of a single person. Thus, when diagnosing new patients, we can't assume our model has seen their brains before. Without this setting, the model will focus more on "classifying the same person", while in reality hurts the generalization ability.

- Model with RNN structure embedded to leverage temporal information. We plan to train the RNN classifier on top of a CNN model that has already been trained on the classification task where the CNN model serves as a feature extractor. We expect the performance of the hybrid model is better than the plain CNN classifier. In this setting, we consider the subject-level classification, which means we regard images from same subject as a time sequence of images, and for each subject, we predict a final label based on the final output of the RNN.

3. Related Work

There has been many related research performed in this area. A early paper "Long-term Recurrent Convolutional Networks for Visual Recognition and Description", published in 2014, proposed a class of CNN-LSTM end-to-end trainable neural network suitable for large-scale visual understanding tasks, such as activity recognition.[2] One of the more recent paper "Recurrent Neural Networks with Longitudinal Pooling and Consistency Regularization", published in early 2020, has proposed novel longitudinal pooling and consistency regularization techniques to specifically focus on AD classification task through series of temporal MRI images for each individual patients.[4]

4. Technical Approach and Preliminary Results

4.1. Softmax Baseline

We trained a simple softmax classifier to classify an image as AD/Normal as our baseline. The flattened structure of the fully connected layer cannot really build a representation of the spatial information. Together with a relative small dataset, the baseline suffers from overfitting as indicated by the huge gap between training accuracy and validation accuracy.

Model	train acc	val acc	num params	train time
Baseline	1	0.65	524k	5s/epoch

Table 1. Training results for linear Softmax baseline model.

4.2. 4-layers Conv-Norm-Relu-Pool

Method: This model has a basic structure of 3D convolution layer with kernel size 3 and padding 1, followed by 3D batch normalization, followed by a Relu activation and a 3D max pooling. This basic structure unit is being repeated for 4 times with dropout layers in between. The number of channels are small due to current GCC outage. Several channel size combinations for convolutions have been tested including: $16 \to 32 \to 64 \to 32, 8 \to 16 \to 32 \to 16$, and $10 \to 20 \to 40 \to 20$.

After the convolution layers, 2 fully connected layers are used to halve the hidden dimension at each step. Finally, the third fully connected layer is used to output binary classification result.

Results:

The learning rate used for all following 3 test is 5e-4. In test 1, the training accuracy quickly converged above 0.9, which leads validation accuracy by a large margin. This indicates a over-fitting problem due to the relatively small size of our training detests. Therefore, higher dropout rates are used to achieve a better regularization effect. The best training accuracy and validation accuracy achieved for this model is 0.98 and 0.88.

test	channels	dropout	train acc	val acc
1	8	0.3	0.91	0.72
2	8	0.5	0.89	0.76
3	10	0.4	0.95	0.82
4	16	0.4	**0.98**	**0.88**

Table 2. Training results for 4-layers Conv-Norm-Relu-Pool model.

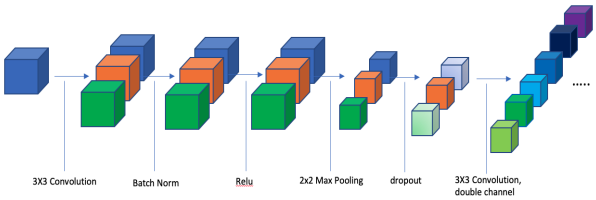

Figure 3. 4-layers Conv-Norm-Relu-Pool model.

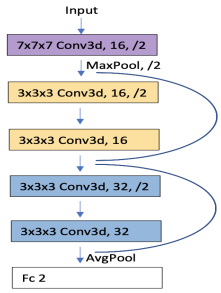

Figure 4. Model Architecture for ResNet-6

4.3. ResNet-6

Following He et al. [3], we build a minimal CNN based ResNet model. We run two models with the following configuration: lr=1e-3, Adam Optimizer, weight decay=1e-4, dropout=0.5 with and without data augmentation.

Model	train acc	val acc	num params	train time
Aug	0.98	0.82	62k	15min/epoch
No Aug	0.98	0.83	62k	1h/epoch

Table 3. Training results for ResNet-6 model.

As indicated in the training curve (see appendix), compared to training without data augmentation, data augmentation serves to make the training more stable. However, in our case, data augmentation doesn't help to increase out validation accuracy.

4.4. CNN-RNN model

Method: To leverage the benefit of temporal information hidden in the time sequence of image samples, we implement a CNN-RNN model to help extract the temporal feature. This model has a basic structure of 3D-CNN feature extractor, and a RNN structure built on top of the feature, a Linear softmax classifier is after the RNN to output the prediction. For feature extractor, we use 4-Layers Conv-Norm-Relu-Pool structure, and the output dimension(after flatten) of the structure are 512. For RNN structure, we use a single layer GRU Network with 16/32/64 dim hidden unit. Before feeding the feature into RNN, a linear layer will shrink the feature dimension to 16/32/64. After RNN, 2 fully connected layers are used to output the label.

Results: We tested the influence of different dimension of hidden-unit. The learning rate used for all following 3 test is 5e-4. We use Adam optimizer to train the model, and we also test how weight decay will influence the performance. In all test, the training accuracy quickly converged above 0.9, and still, the overfitting problem is on our way, due to small dataset size. Therefore, smaller hidden unit size can gain a better regularization effect. Also, we found bigger weight decay helps bridging the train and val accuracy. The best validation accuracy achieved for this model is 0.84.

test	hidden unit size	weight decay	train acc	val acc
1	64	1e-4	0.98	0.71
2	32	1e-4	0.96	0.78
3	16	1e-4	0.95	0.82
4	16	2e-4	0.96	0.84

Table 4. Training results for CNN-RNN model.

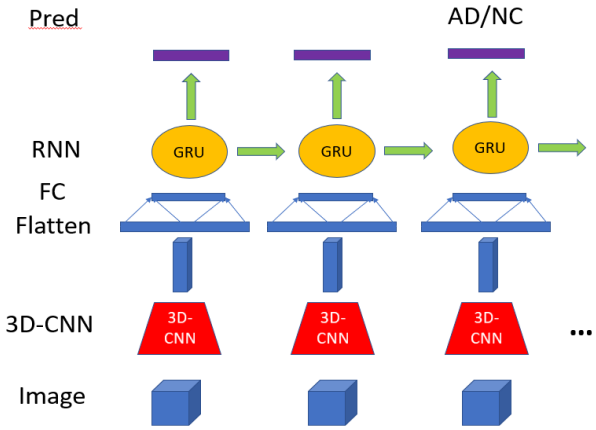

Figure 5. CNN-RNN model.

References

[1] Alzheimer's disease neuroimaging initiative.

[2] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[4] J. Ouyang, Q. Zhao, E. V. Sullivan, A. Pfefferbaum, S. F. Tapert, E. Adeli, and K. M. Pohl. Recurrent neural networks with longitudinal pooling and consistency regularization, 2020.

5. Appendix

5.1. Training Curves

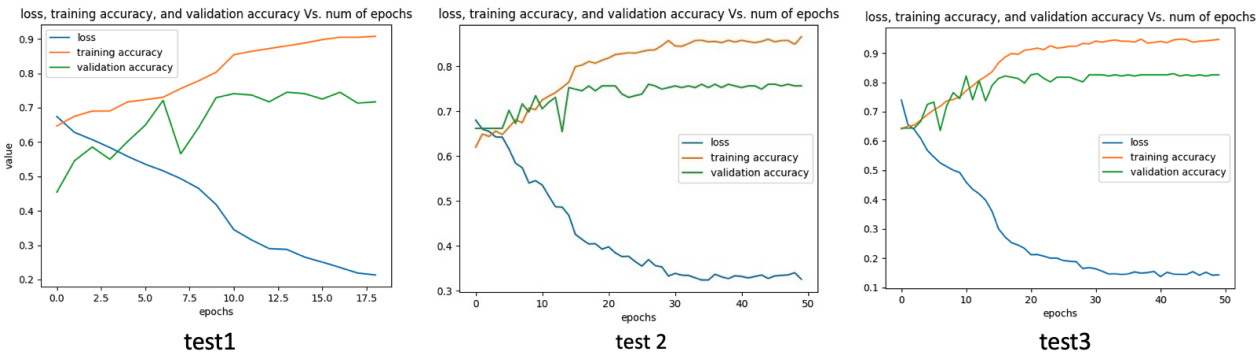

Figure 6. Training curves for 4-layers Conv-Norm-Relu-Pool model

Figure 7. Training curves for ResNet-5 model. Left: Training without data augmentation; Right: Training with data augmentation. The stabilizing effect of data augmentation can be observed from the smoother curves for training with data augmentation.