Leveraging Temporal Information to Better Understand Alzheimer's Disease Diagnosis

Stanford CS231N Course Project

Yanpei Tian
Stanford University
yanpeit@stanford.edu

Yanhao Jiang
Stanford University
jiangyh@stanford.edu

Zixuan Liu
Stanford University
zucks626@stanford.edu

Abstract

In this project, we explored different methods and techniques for Alzheimer's Disease identification via 3D brain magnetic resonance images. Each patient sample in our dataset includes a series of 3D images across multiple dates. We first built a simple baseline model using linear Softmax classifier. Then, we implemented different convolution neural network models including a 3D ResNet model and a 4-layer Conv-BatchNorm-ReLu-Pool model. Finally, considering the time evolving nature of Alzheimer's Disease, we built a CNN-RNN hybrid model in order to utilize the temporal information embedded in patient's time-series image data across multiple images. The 4-layer Conv-BatchNorm-ReLu-Pool model constructed previously is used as the feature extraction component for our hybrid model. Our best CNN-RNN hybrid model can achieve accuracy about 87% with 5-fold cross validation, with more stable training compared to pure CNN based models.

1. Introduction

Magnetic resonance imaging (MRIs) on human brain have been one of the most critical methods 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 in patients' brain compared to the normal ones. While modern ML technique gains success on analyzing MRI images to help with disease diagnosis, one remaining challenge is to apply Longitudinal Diagnosis Analysis, as many fancy methods are proposed under cross-sectional design. New difficulty lies on how to extract the intra-subject information along longitudinal 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 used a hybrid model involving CNN[11] for individual image spatial feature extraction and RNN for longitudinal analysis across a series of images. Such an architecture allows us to better utilize the temporal information presented in our dataset.

To evaluate our method, an AD/non-AD classification task will be run to see if higher accuracy is reached.

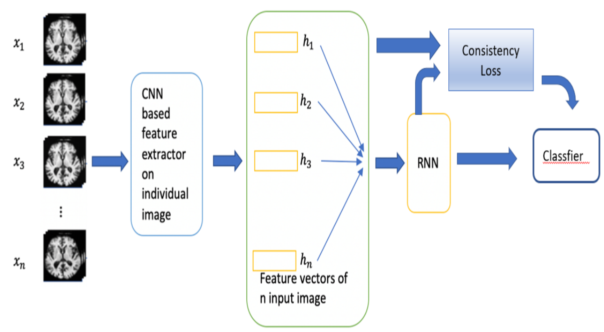

Figure 1. High-level model diagram. A time series data is first fed into a CNN, which extracts spatial information related to AD. The feature vectors are then fed into a RNN structure to generate the final prediction. We also plan to add a regularization term in the training procedure to ensure the consistency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

As mentioned above, we propose a hybrid model (CNN-RNN) that can extract temporal information over a time series data to diagnose Alzheimer's Disease. The input to our final model is a sequence of brain MRI image, each represented by a (64, 64, 64) array. Images in the same sequence are MRIs taken on the same subject at different time. For each MRI image, the model will produce a pair of logits corresponds to being class0 (normal) and class1 (AD), which will then be fed into the cross-entropy loss function.

2. Related Work

A very common way to apply longitudinal analysis on 3D brain image MRI data is to perform supervised learning tasks, e.g, trying to do classification on different subject with single image or time series data. [13, 16, 6, 3, 7]. 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[9] end-to-end trainable neural network suitable for large-scale visual understanding tasks, such as activity recognition. [4] 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. [15] There are also methods such as Mixed Effect Models tries to extract hidden correlation in the intra-subject time-series data by regarding the representation trajectories as fine-tunable parameters. [17, 14].

3. Methods

The input X that being fed into our models are 64x64x64 matrices representing 3D images. The output would be 2 logits representing probabilities of being in the 2 classes: AD and non-AD. A binary cross entropy equation shown below is being used as our loss function.

$$CE = -p_0 log(\hat{p_0}) - (1 - p_0) log(1 - \hat{p_0})$$

3.1. Baseline

Single-layer Neuron Network: As a baseline, we build a basic logistic regression model. The flattened structure of the fully connected layer cannot really build a representation of the spatial information in the 3D image. Together with a relatively small dataset, it is expected that the baseline will suffer from overfitting issue.

3.2. CNN based models

Instead of MLP, CNN[5] is widely used in image recognition tasks as the well-known feature extractor for both 2D and 3D image. Yet it is crucial to specify its structure in order to gain higher performance. In this part, we designed different structures of CNN, and run experiments on each of them to decide our final CNN structure. Here we proposed two families of model we tested.

4 Layer Conv Model: There are several widely-used sequences of CNN structures, while most of them are different composition of Conv layer, MaxPool, Dropout[12], Activation, and BatchNorm[10] layer. Different composition of these different types of fancy layer may lead to significantly different performance. As our first attempt to enable the model to leverage the spatial information embedded in the MRI images, we build a classical CNN model shown below:

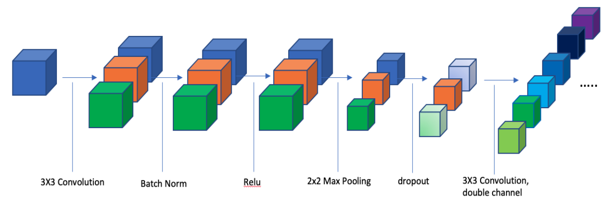

Figure 2. 4-layers Conv-BN-Relu-Pool-Dropout model.

This model has a basic structure of 3D convolution layer with kernel size 3 and padding 1, followed by a Relu activation and a 3D max pooling. To stabilize the model, we apply BatchNorm layer(after 3D Conv layer) and to have less over-fitting problem, we apply Dropout layer(after MaxPool layer).

This CNN feature extractor will process each MRI image from a (64, 64, 64) array into a vector with smaller length of 640/512/128. To reduce the much too flexible representation capacity of the CNN, the output channel of each Conv layer is set as: (16,32,64,16). This will give us a (16,4,4,4) feature map after a forward pass through the CNN. We then flatten this feature map to a 1024D vector, and a linear layer will shrink the vector to our final extracted feature. On top of the CNN-based feature extractor, we apply a single-layer classifier to gain the final prediction score. The input/output of the classifier is 512 and 2, and the number of hidden unit of the hidden layer is 64. As demonstrated in our experiments, this model can achieve better results during testing compared to the flattened baseline.

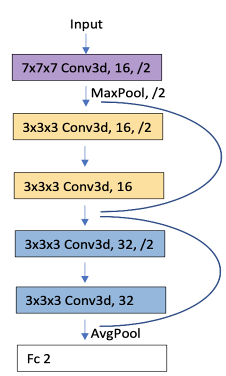

Figure 3. Model Architecture for ResNet-6

ResNet-6: First proposed by He et al.[8], Residual Network (ResNet) is famous for its superior ability to build very deep representations on image learning. The introduction of skip connections also makes the training easier. Replacing 2D convolutions by 3D convolutions, we build a mini-ResNet that can operate on 3D MRI images.

In our case, the addition of extra convolution layers and the skip connections don't necessarily improve the performance of our model: the increased model capacity and our rather small training dataset cause overfitting. The same MLP structure in the 4-layer Conv Network is used on top of the Resnet-6 to eliminate the confounder between different models, facilitating comparison.

3.3. CNN-RNN hybrid models

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 component of 3D-CNN feature extractor, which is the 4-layer Conv Network we proposed before. Instead of using a final output layer to directly convert vectors to 2 class logits, we keep the final layer vector with size of 128 or 512 and use it as the input to our RNN component.

We have tested two different structure of RNN: LSTM and GRU with 16/32/64 for hidden dimensions. A linear layer will map the extracted 128/512D feature to the same length of hidden units, and then the compressed feature would be fed into RNN model. Both LSTM and GRU[2] are varieties of RNN built to prevent vanishing gradient problems by adding a uninterrupted gradient flow. Specifically, the structure of LSTM is implemented as the equations shown below:

$$\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}$$

While GRU is constructed based on the equations below:

$$\begin{aligned}
r_t &= \sigma(W_{xr} x_t + W_{hr} h_{t-1} + b_r) \\
z_t &= \sigma(W_{xz} x_t + W_{hz} h_{t-1} + b_z) \\
\tilde{h}_t &= tanh(W_{xh} x_t + W_{hh}(r_t \circ h_{t-1}) + b_h) \\
h_t &= z_t \circ h_{t-1} + (1 - z_t)\tilde{h}_t
\end{aligned}$$

After the RNN component has processed CNN feature inputs across all time points, the hidden units in all RNN layers would be collected to be fed into a linear softmax classifier for final predictions at each time step. During training, loss would be calculated by using a mean of all timestamps, while in evaluation, only the final output of the RNN corresponds to the last valid MRI image would be used as the prediction.

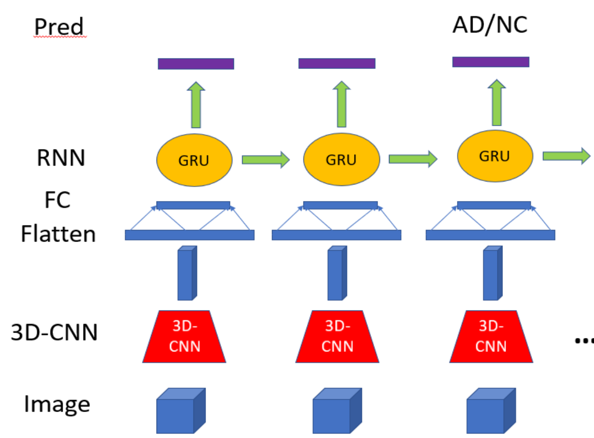

Figure 4. CNN-RNN model.

4. Dataset and Features

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 1-6 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 the raw data has been processed, the format of each MR image will be 64*64*64. Since the number of total images is less than 3,000, we consider to use 5-fold cross validation to better evaluate our model. To avoid memorizing images from same patients, the split has to be done at subject level instead of image level for all cases. This is to prevent the model from learning similarities of the same subject's brain MRI images taken at different time.

4.1. Data pre-processing and augmentation

We propose two slightly different data pre-processing and augmentation procedures with CNN/Baseline setting and longitudinal setting. For CNN/baseline setting, the procedure is:

- Data normalization (zero mean and unit variance) to facilitate training;
- Random shuffled data and then equally divided them into 5 folds on subject-level;
- Data augmentation by randomly choosing image sample in that folds, and then introducing small random ro-

tation and translation to them. Balanced the data with 2 different label(AD/NC) to both 400 images in each fold.

Note that the random shuffle procedure makes sure the ratio of NC/AD subject in each fold are roughly the same. While the original ratio of samples with different labels are imbalanced, we utilize the data augmentation procedure to balance the whole dataset. Finally we will get a total of 4,000 images for all 5 folds, thus the rate of augmented data is roughly 33%. For CNN+RNN setting, the procedure is:

- Data normalization (zero mean and unit variance) to facilitate training;
- Random shuffled data and then split the data into 5 folds on subject-level;
- Data augmentation by randomly choosing sequence data of each subject, and introducing small random rotation and translation on all the image samples in that sequence with same random parameters. Balance the data to get both 120 AD/NC subjects in each fold.

Slightly different from CNN/baseline setting, here we we apply same random parameters to the whole sequence of data because we are doing subject-level classification. Also, we balance the dataset by having equal number of subjects in each fold, this might not lead to a completely balance ratio on image-level, yet it's more reasonable in this longitudinal case.

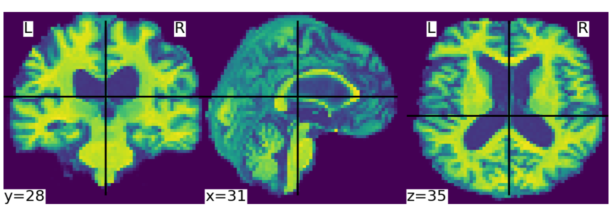

Figure 5. Exemplary MRI image in the training dataset, 3D image visualized by NiBabel.

Implementation details: For the same person, we need to make sure the age attribute for each MRI is sequenced from the smallest to largest. Only when the age fields are properly aligned can the RNN model extract any useful information as time progress. Moreover, as any subject has at most 6 images, we make the input array of size (6, 64, 64, 64) for all the subjects by zero padding any vacancy. The useful part of each subject's input is denoted by a mask. Only the masked portion of the output is considered during loss function calculation.

To be specific, a forward pass through the network involves the following steps:

1. Reshape the input as 6 individual 3D images and propagate them through the CNN based feature extractor. The output will be 6 feature vectors of length H;
2. Pack the 6 feature vectors together in the order of increasing age associated with each input image.
3. Feed the packed feature vectors through RNN and the final logistic regression layer, the output will be 6 pairs of classification logits;
4. Calculate the loss for one subject, only counting the classification logits covered by the mask.

5. Experiments/Results/Discussion

5.1. Experiment setup

- 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. To avoid this confounder, instead of randomly splitting 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 over time/other factors such as diseases or aging 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 plus softmax classifier on top of a CNN feature extractor that has already been pre-trained on the classification task. 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.

5.2. Softmax Baseline

We trained the baseline model for 50 epochs with a batch size of 16. The L2 regularization strength is 1e-4 and the learning rate starts at 5e-4, decreasing exponentially. As shown in table 1, a training accuracy of 0.99-1 and a testing accuracy around 0.63 show this model suffers from overfitting. Moreover, the linear operation that maps a (64, 64, 64) array to 2 logits leads to a rather big model size, totaling 524k parameters.

Acc	F1	F2	F3	F4	F5	Avg
Train	1	0.991	1	1	0.997	0.997
Test	0.62	0.65	0.642	0.635	0.661	0.6416

Table 1. Training results for linear Softmax baseline model.(5-fold CV)

5.3. 4-layers Conv-Norm-Relu-Pool

Result:

The learning rate used for all following 4 tests is 5e-4. All of the following experiments were trained for 50 epochs in batches of size 16. 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 data. Therefore, higher dropout rates are used to achieve a better regularization effect. The best average training accuracy and validation accuracy achieved for this model is 0.98 and 0.86.

test	dropout	avg train acc	avg val acc
1	0.3	0.91	0.72
2	0.5	0.89	0.76
3	0.6	0.95	0.82
4	**0.4**	**0.98**	**0.86**

Table 2. Training results for 4-layers Conv-Norm-Relu-Pool model.(Acc showed is average accuracy for 5-fold CV) Training plots for experiment 1 through 3 is attached in the appendix.

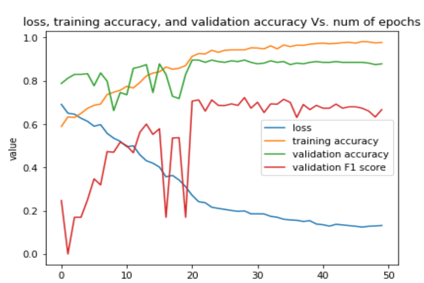

Figure 6. Training curves(average) for experiment 4, with testing accuracy converges at 0.86.

Discussion:

Compared to the flattened baseline, the CNN based model is able to build a better representation regarding the spatial information in the MRI images, this is reflected by a big increase from 0.63 to 0.86 in testing accuracy. However, the model prediction's F1 score still plateaus around 0.7. We hope to further address this issue in our hybrid model.

5.4. ResNet-6

Result:

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.(Acc is average accuracy for 5-fold CV)

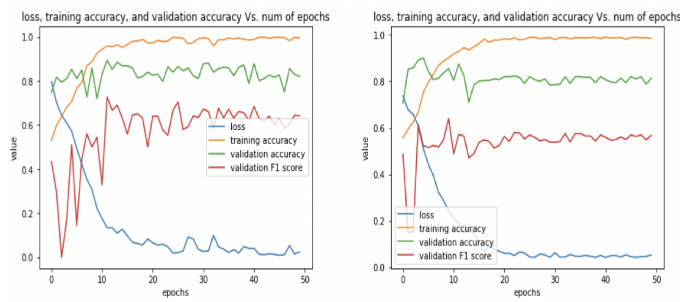

Figure 7. Training curves for ResNet-5 model(average). Left: Training without data augmentation; Right: Training with data augmentation. The stabilizing effect of data augmentation can be observed from the smoother curves for training with data augmentation.

Discussion:

As indicated in the training curve, 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. A comparison between the ResNet-6 model and the previous 4-layers Conv model shows us the residual network doesn't work in our favour this time with highest testing accuracy achieved being 0.86 and 0.83 respectively. A closer look at the training curves (Figure 6 and Figure 7) can give us some hint: the increased model capacity in ResNet-6 leads to overfitting. As the training accuracy converges to 1 faster for ResNet-6 than for the 4-layers Conv model while testing accuracy lags behind.

5.5. CNN-RNN model

Results:

The learning rate used for all following 3 test is 5e-4. We use Adam optimizer to train the model. We tested the influence of different dimension of hidden-unit in RNN and we also tested how weight decay can 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 training dataset. Therefore, smaller hidden unit size can gain a better generalization ability. Also, we found bigger weight decay helps bridging the train and val accuracy. The best average validation accuracy achieved for this model is 0.87. The configuration for the best performing model, as denoted in table 4, has 16 hidden units in the RNN and a wight decay of 1e-3.

test	hidden units	weight decay	train acc	val acc
1	64	1e-4	0.981	0.717
2	32	1e-4	0.969	0.789
3	16	1e-4	0.952	0.823
4	16	2e-4	0.961	0.856
5	**16**	**1e-3**	**0.934**	**0.871**

Table 4. Training results for CNN-RNN model.(Acc is average Accuracy for 5-fold CV)

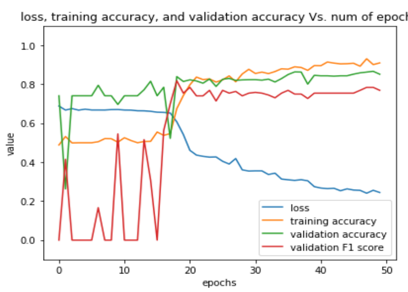

Figure 8. Training curves for test 5, with testing accuracy converges to 0.87.

Discussion:

Compared to pure CNN models that operate on individual MRI images, the CNN+RNN hybrid model helps to mitigate the overfitting issue: it has the lowest training accuracy of 0.934 and highest testing accuracy of 0.871. Moreover, the hybrid model also has the highest F1 score, approaching 0.8. This shows it has the lowest class imbalance issue caused by not perfectly balanced training and testing dataset.

5.6. Positive prediction result visualization

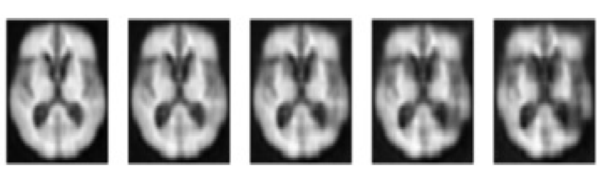

Figure 9. A MRI sequence correctly marked as AD by the hybrid model.

In Figure 9, we can see a clear deterioration trend of the brain, which is a significant feature of AD. The experiment results listed above shows the RNN component in the hybrid model is able to build a temporal connection along time and performs better in the subject level classification test, where only the last label is being used to classify a subject as normal or AD.

6. Conclusion and Future Work

The initial naive linear implementation has the lowest performance, this is expected since it did not take any spatial information of 3D images into account. The two pure CNN models we built later achieved a very decent accuracy performance. While the ResNet-6 model we built did not improve the performance compared to the original 4-layer convolution model as what we expected. This is possibly because that the skip connections mostly boosts the performance by alleviating vanishing gradient problems caused by a deeper network; however, in our case where we have relatively small dataset, the increased model capacity only imposed higher risks of over-fitting. Finally, our hybrid CNN-RNN model achieved great performance in terms of accuracy (0.87), as well as F-1 scores (0.78), indicating a precise and balanced prediction result. This proves that the temporal information indeed helps the prediction of Alzheimer's Disease. Aside from the various neural network structures, we have also explored many different data augmentation techniques to increase training dataset size and balance training classes. The results obtained did not show significant improvement in terms of prediction accuracy; however, based on the training and validation curves, it does stabilize the model during training.

Since the size of our training examples is small (400 in subject level and 3000 in image level), one of the biggest problem of this project that we faced is over-fitting. To avoid this issue, we have adopted various methods, such as data augmentation and dropout layers. In addition, smaller/less expressive models are being used to alleviate overfitting. For example, we limit the number of layers in pure CNN model to 4 and the size of hidden units of RNNs to 16.

In the future, we would like to explore more on the temporal longitudinal pooling and consistency regularization techniques introduced in the paper "Recurrent Neural Networks with Longitudinal Pooling and Consistency Regularization". Another avenue for future research is to systemically address the overfitting issue we have in this project. This is especially the case for medical applications. As the data and compute hungry nature of modern ML methods conflicts with the relative high price for collecting medical data. Another possible step forward would be studying on interpretable machine learning methods on diagnosis analysis, such as factor disentanglement and confounder-aware models. Last but not least, we should also consider 'safety' issues when we apply automatic algorithms like ML to a health care situation. Define good machine behaviour and ensure better patient outcome remain challenging research topics in the high stake scenario of health care.

7. Appendices

Training curves for 4-layers Conv model experiment 1 through 3.

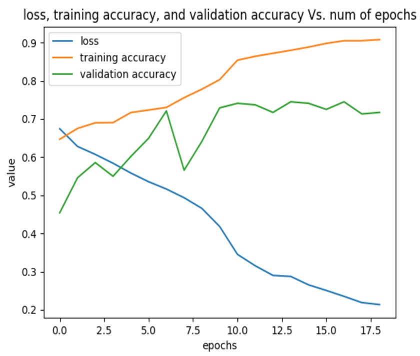

Figure 10. Experiment 1, Dropout = 0.3

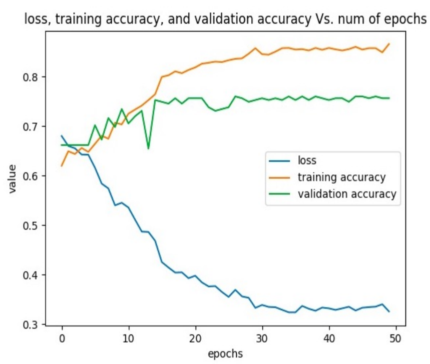

Figure 11. Experiment 1, Dropout = 0.5

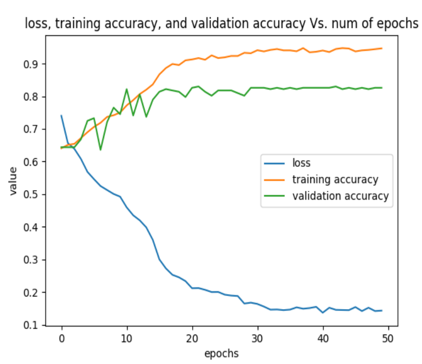

Figure 12. Experiment 1, Dropout = 0.6

8. Contributions and Acknowledgements

All of the group members contributes to Proposal/Milestone/Final Report writing.

Coding & Experiments:

Yanpei Tian: Original training loop, Baseline Softmax Classifier, ResNet-6 model, and LSTM version of the hybrid model.

Yanhao Jiang: 4-layer Convolution model, LSTM version of the hybrid model.

Zixuan Liu: Proposed idea, Getting dataset, Data processing and augmentation, 4-layer Convolution model, GRU version of the hybrid model.

Acknowledgements: Thanks to Qingyu Zhao from CNSLab, Stanford University, who helped to provide the very first pre-processing proedure on ADNI data.

References

[1] Alzheimer's disease neuroimaging initiative.

[2] C. K. H. e. a. Chung J, Gulcehre C. Empirical evaluation of gated recurrent neural networks on sequence modeling[j], 2014.

[3] R. Cui, M. Liu, A. D. N. Initiative, et al. Rnn-based longitudinal analysis for diagnosis of alzheimer's disease. *Computerized Medical Imaging and Graphics*, 73:1–10, 2019.

[4] 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.

[5] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[6] R. Gao, Y. Huo, S. Bao, Y. Tang, S. L. Antic, E. S. Epstein, A. B. Balar, S. Deppen, A. B. Paulson, K. L. Sandler, et al. Distanced lstm: Time-distanced gates in long short-term memory models for lung cancer detection. In *International Workshop on Machine Learning in Medical Imaging*, pages 310–318. Springer, 2019.

[7] M. M. Ghazi, M. Nielsen, A. Pai, M. J. Cardoso, M. Modat, S. Ourselin, L. Sørensen, A. D. N. Initiative, et al. Training recurrent neural networks robust to incomplete data: Application to alzheimer's disease progression modeling. *Medical image analysis*, 53:39–46, 2019.

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

[9] S. J. Hochreiter S. Neocognitron: Long short-term memory[j]. *Neural computation*, 9(8):1735–1780, 1997.

[10] S. C. Ioffe S. Batch normalization: Accelerating deep network training by reducing internal covariate shift[j]., 2015.

[11] B. Y. LeCun Y. Convolutional networks for images, speech, and time series[j]. *The handbook of brain theory and neural networks*, page 3361(10), 1995.

[12] B. Y. LeCun Y. Dropout: a simple way to prevent neural networks from overfitting[j]. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[13] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

[14] M. Louis, R. Couronné, I. Koval, B. Charlier, and S. Durrleman. Riemannian geometry learning for disease progression modelling. In *Information Processing in Medical Imaging*, pages 542–553, 05 2019.

[15] 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.

[16] R. Santeramo, S. Withey, and G. Montana. Longitudinal detection of radiological abnormalities with time-modulated lstm. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 326–333. Springer, 2018.

[17] Y. Xiong, H. Kim, R. Mehta, S. Johnson, and V. Singh. On training deep 3d cnn models with dependent samples in neuroimaging. In *Information Processing in Medical Imaging*, pages 99–111, 05 2019.