SLAC National Accelerator Laboratory

Machine Learning for Alzheimer's Classification

Jake Treska
SLAC National Accelerator Laboratory
jtreska@slac.stanford.edu

Abstract

With the continued rise of neurodegenerative diseases such as Alzheimer's, it becomes imperative to apply our recent advancements in artificial intelligence to help combat this rising challenge. By implementing advanced machine learning concepts, medical professionals will be capable of detecting and classifying neurodegenerative diseases such as Alzhiemer's more efficiently and with increased precision.

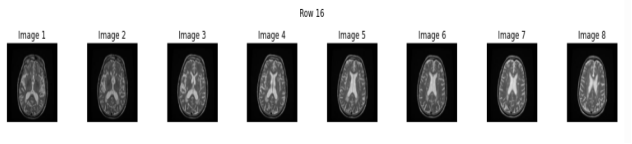
This paper serves to highlight the research present at SLAC National Accelerator Laboratory where my team and I constructed multiple accurate machine learning models capable of detecting three classifications of Alzheimer's stages: Cognitive Normal (CN), Mild Cognitive Impairment (MCI), and Alzheimer's Disease (AD). More specifically, this paper will describe all project related activities and how they lead to our result of building a deep learning model capable of achieving a diagnostic accuracy of 82% by analyzing sequences of MRI images.

Background

Arguably the most important step in building an accurate machine learning model relies in acquiring clean and accurate data. The data used for model training was acquired from the Alzheimer's Disease Neuroimaging Initiative (ADNI) website initiated by University of Southern California (USC). This website is highly reputable, and after receiving clearance to access their data, we were allowed to utilize thousands of actual patient MRI images to train our models.

After collecting our dataset needed for model training, there were still multiple steps before being able to start training the machine learning models. More specifically, in order to achieve the accuracy metrics necessary to prove the importance of machine learning in combating neurodegenerative diseases, our group needed to perform significant data preparation and image enhancements. The first problem our group faced with the data collected from ADNI was that each MRI image is roughly fifty 2D image slices, with each slice being out of order. This poses a significant problem as our plan was to analyze these 2D slices in sequence, as each slice possesses vital information across all regions of the brain. To solve this problem, multiple algorithms were written to accurately rearrange the images into their proper order. With this problem solved, there was still one significant problem to overcome before developing our model. This problem wasn't due to the ADNI website, but rather with the nature of analyzing MRI images. While each MRI image is actually a collection of numerous 2D slices, not every slice is important at classifying the existence of Alzheimer's. In fact, only a select few images are useful, thus requiring us to build a smaller machine learning model capable of analyzing each 2D slice in order to determine which images will be the most helpful in diagnosing Alzheimer's. After building and training this model, we were now able to process each MRI scan allowing us to obtain the relevant 2D slices displaying the most prominent features utilized for Alzhiemer's classification. [1]

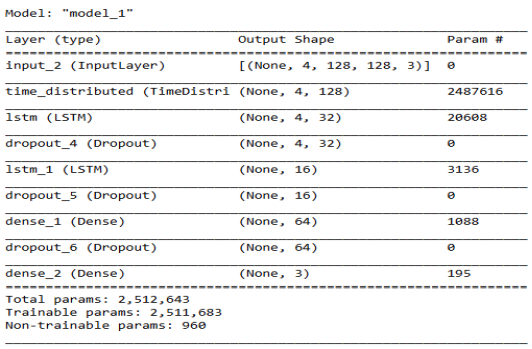
Figure 1: Dataset



The image displayed in Figure 1 illustrates a typical dataset used for model training. The machine learning model described above analyzes each MRI image and returns an array of images similar to the one present above. After receiving this array, the images in each array represent the most important 2D slices in each MRI and the middle four images will be used for training. In other words, each MRI scan is filtered down to four sequential 2D slices that best portray features necessary for accurate Alzhiemer's classification.

Implementation - Convolutional Recurrent Neural Network (CRNN)

Figure 2: CRNN Architecture

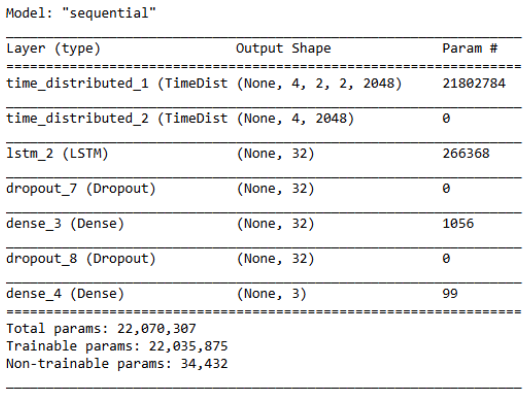


```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 4, 128, 128, 3)]  0
_________________________________________________________________
time_distributed (TimeDistri (None, 4, 128)            2487616
_________________________________________________________________
lstm (LSTM)                  (None, 4, 32)             20608
_________________________________________________________________
dropout_4 (Dropout)          (None, 4, 32)             0
_________________________________________________________________
lstm_1 (LSTM)                (None, 16)                3136
_________________________________________________________________
dropout_5 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                1088
_________________________________________________________________
dropout_6 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 195
=================================================================
Total params: 2,512,643
Trainable params: 2,511,683
Non-trainable params: 960
_________________________________________________________________
```

The first model built was a convolutional Recurrent Neural Network (CRNN), as shown in figure 2 above. This model contains approximately 2.5 million parameters and is a combination of three unique machine learning models: A convolutional Neural Network (CNN), a Recurrent Neural Network (RNN), and a Fully Connected Layer (FC). The CNN component is used to analyze each individual image, extracting key features and edges. This is done by applying hundreds of filters to each image and constantly shrinking the dimensions in order to focus on only the most important features in each image. The processing within each filter essentially creates slight variations of the original image, thus allowing the model to compare these filtered results to better learn the objects within each image. This data is then passed through the RNN component, which evaluates the temporal dimension between MRI slices. This is done by flattening each MRI slice into a 1D array, since there are four slices representing each MRI scan the CNN component will output four 1D arrays. These four arrays are passed through the RNN component where similar regions in each array are compared with one other. This allows the model to analyze changes over time as it looks at each 2D slice sequentially. After the CNN and RNN components are processed, the model is able to identify key features in each individual MRI slice, and how they change across following slices. All of this data is then passed through the Fully Connected layer (FC), or Dense Layer, that calculates a final prediction of either CN,MCI, or AD. [2]

Implementation - Inception Network (InceptionV3)

Figure 3: InceptionV3 Architecture



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
time_distributed_1 (TimeDist (None, 4, 2, 2, 2048)     21802784
_________________________________________________________________
time_distributed_2 (TimeDist (None, 4, 2048)           0
_________________________________________________________________
lstm_2 (LSTM)                (None, 32)                266368
_________________________________________________________________
dropout_7 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_3 (Dense)              (None, 32)                1056
_________________________________________________________________
dropout_8 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_4 (Dense)              (None, 3)                 99
=================================================================
Total params: 22,070,307
Trainable params: 22,035,875
Non-trainable params: 34,432
_________________________________________________________________
```

The second model that was built is an Inception network, and its architecture is highlighted in figure 3 above. Similar to the architecture described for the CRNN, this model also incorporates a CNN, RNN, and FC component for its final prediction. The main difference between the Inception model and the previous CRNN model, is how the convolutional layers are applied. In the CRNN model, the convolutional layers are applied sequentially. For example, filters are first applied to the image and then the spatial dimensions are reduced. This cycle happens numerous times in the CRNN model depending on the complexity of the image. On the other hand, in the Inception network, convolutional layers are applied in

parallel instead of in sequence. This means that instead of applying filter A and then filter B, both filters will be applied to the original image at the same time. By applying convolutional layers in parallel rather than sequentially, the model can more accurately learn distinct features throughout the whole training process. [3]

Results

Figure 4: Classification Matrix

Ensemble Accuracy: 0.82
Classification Report:

	precision	recall	f1-score	support
0	0.95	0.82	0.88	51
1	0.69	0.79	0.73	42
2	0.82	0.83	0.83	60
accuracy			0.82	153
macro avg	0.82	0.81	0.81	153
weighted avg	0.83	0.82	0.82	153

After training both models, the CRNN received an accuracy of 75%, while the InceptionV3 network received an accuracy of 81.7%. Additionally, after combining both models by averaging their predictions the accuracy increased slightly to 82%. By referencing additional metrics for the combined model shown in figure 4, for CN labeled images (class 0) the model has a precision/recall/f1-score of .95/.82/.88. For MCI (class 1) it achieved scores of .69/.79/.73, and for AD labels (class 2) it received scores of .82/.83/.83. To further explain the significance of these metrics, the precision scores for each class (labeled 0-2) represent the combined model's accuracy at predicting each individual class. Additionally, the three recall values represent the model's sensitivity towards finding all positive cases for each label. For example, looking at all the CN images, labeled as class 0, when the model thought an image was CN it was right 95% of the time, but the model was only able to find 82% of all positive cases. With regards to f1-score, it represents a balance between precision and recall, and is a very useful metric when working with unbalanced data. Finally, the support category displays how many positive labels are in each class, and since this is the test dataset there are only 153 MRI scans. Overall, while accuracy was the most important metric used towards model evaluation, by examining other metrics such as precision, recall, and f1-score, it allows us to better understand the strengths and weaknesses of our model which are necessary for finer tuning.

Conclusion

After analyzing our model's performance and capabilities, it becomes evident that implementing machine learning models in the medical world is the next step in humanity's technological development. Not only is machine learning approaching diagnostics skills similar to trained medical professionals, but the speed at which these models make their predictions is revolutionary. By coupling these advanced models with trained professionals, the medical industry will be able to more accurately and efficiently monitor, analyze, and treat most medical diseases. The future of AI in medicine is seemingly boundless as our models become capable of analyzing more and more data, leading to more efficient predictions while drastically lowering mistakes in diagnoses.

Overall, in light of research present at SLAC National Laboratory, I believe our results support the claim for continued research to improve and finalize our model. While I acknowledge 82% doesn't reach the level of trained medical professionals necessary for deployment, it was made by a small team with limited time. With continuation of the research, there are many techniques that I believe can drastically improve accuracy

allowing our model to more closely represent the skills of medical professionals. Some of these techniques include utilizing image processing services like Freesurfer, allowing us to extract more data from our MRI scans, which in turn allows us to feed more accurate data to our model which can drastically increase performance. Additionally, another goal is to analyze the complete set of 2D slices present in each MRI, instead of just filtering out the best four, allowing us to better analyze regions across the whole MRI scan.

Acknowledgements

After completion of my SULI internship, I would like to personally thank my mentor Dr. Juhao Wu. His guidance and leadership served a critical role throughout the whole internship program, ultimately allowing our group to achieve our final product. It was an honor to work with him, and I look forward to potential future collaborations. I would also like to personally thank Ms. Hillary Freeman in addition to my fellow interns and staff. I have had an amazing time working as part of the SULI program, and I am very thankful for all the connections I have made over the whole internship program. Finally, I would also like to express my gratitude to SLAC National Accelerator Laboratory and to the DOE, for allowing me the opportunity to contribute to such fascinating and significant research.

Citations

[1] “Alzheimer’s Disease Neuroimaging Initiative.” *ADNI*, adni.loni.usc.edu/. Accessed 23 Aug. 2024.

[2] Hassan, Adnan. “Deep Learning Architectures from CNN, RNN, Gan, and Transformers to Encoder-Decoder Architectures.” *MarkTechPost*, 12 Apr. 2024, www.marktechpost.com/2024/04/12/deep-learning-architectures-from-cnn-rnn-gan-and-transformers-to-encoder-decoder-architectures/.

[3] Raj, Bharath. “A Simple Guide to the Versions of the Inception Network.” *Medium*, Towards Data Science, 31 July 2020, towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202.