Brain Lesion Segmentation Report

Jonathan Lin, Lucas Tecot

Introduction

In this project, our goal was to segment lesions in FLAIR MRI brain scans as accurately as possible without any need for human pre-processing. Secondary to accuracy, we also desired to make the model as simplistic and fast as possible.

Methods

For generating our database, we first collected full FLAIR MRI brain scans of 12 anonymized patients, later increasing this set to 23 patients. We then extract the regions of interest into a bitmap associated with each scan. Each image is normalized by subtracting its mean and dividing by the standard deviation. We remove background pixels from the mean and standard deviation calculation through a threshold value.

After each image has been extracted and properly assigned lesion labels, we randomly sample patches from the images in order to train the model. The only constraint added here is that half the patches must be centered on a lesion pixel, which in practice creates a rough 40-60 percent distribution of lesion and non-lesion pixels in our dataset. We sample 100,000 training patches and 10,000 validation patches. The set of images that produce these patches are separate and selected from a randomly-shuffled array of the input scans. Additionally, we randomly flip a small percentage of these patches in order to artificially increase the robustness of the dataset.

In terms of our deep learning model, we largely relied upon a simplified 2D version of the model Kamnitsas et al. used in their paper. We tested two different convolutional neural network models, each with and without batch normalization. One is a four layer model with 5x5 kernels, and the other is an eight layer model with 3x3 kernels. Each one uses valid padding, which reduces the size of the patches from 25x25 to 9x9. Then these are passed through a fully connected layer for classification. The output is a 9x9 image with two channels, for probability of lesion and non-lesion pixel respectively. This output is applied to a softmax function in order to achieve the final probabilities, with a greater than 0.5 probability of lesion signifying a prediction of a lesion pixel.

Cross-entropy is used for the loss function, with ground truth predictions being a binary 0 or 1. Additionally L2 regularization and dropout are applied in all models. We use an adam optimizer for training, which seemed to consistently give faster and better results than other gradient descent algorithms. Furthermore, an image opening function is applied on the results of the network in order to remove irregular patches in the prediction.

Evaluation

For evaluating the model, we primarily use metrics of accuracy and similarity. The accuracy coefficient is defined as the number of correctly predicted pixels divided by the total number of pixels. The similarity coefficient is similar to the Dice coefficient, and is defined as the intersection divided by the union of the pixels predicted to be lesions and the pixels that are lesions in the ground truth. We apply these metrics both during training on the image patches as well as on entire images when testing the model.

Unfortunately, because it takes several hours to train each model given our computational resources, we could not cross-validate our results in time. As such, this data should not be trusted fully and only used as a rough measure.

Results

	Accuracy	**Similarity**	**Time (s)**
544CNN+ Norm	**0.977**	**0.427**	**1.223**
544CNN	**0.981**	**0.485**	**1.014**
833CNN+ Norm	**0.984**	**0.521**	**1.741**
833CNN	**0.984 / 0.982**	**0.513 / 0.376**	**1.347 / 1.251**
833CNN+ PostProc	**0.992 / 0.993**	**0.653 / 0.574**	**1.360 / 1.355**

The above table plots the average accuracy, similarity, and time in seconds (run with an i5-3570k Intel CPU) for each image that contains a lesion in the validation dataset. The first three numbers indicate the

convolutional neural network depth and its kernel size. Each value depicts a single training, aside from the last two 833CNN models which contain two numbers, each for a separate training on different validation and test sets. Additionally, graphs for the accuracy and similarity metric are included in the figures section, along with a few examples of segmentations of varying success.

As is indicated in the graph, overfitting is observed with batch normalization in the more shallow network. With both models, batch normalization causes a faster convergence, however it also causes fluctuation in later stages of training. However, the most significant improvement is achieved through the image opening post-processing, which eliminates the patchy results the CNN tends to produce.

Although not included in the table, L2 regularization and dropout are applied at constant rates for each model. L2 was observed to decrease accuracy if given too strong of a weight, however dropout successfully improved the results with only the cost of slower convergence.

Discussion

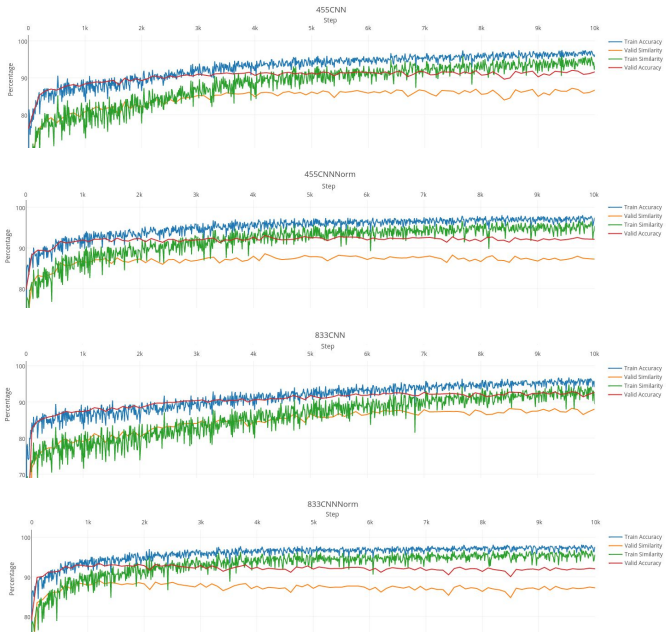
There are a few directions that likely should be explored more, and which I would pursue if given more time. Firstly, it would be wise to take leading segmentation models and retrain them on this data. Also conditional adversarial networks, which can learn a mapping from two sets of images, may be able to perform better than what we were able to produce on our own.

Secondly, a more robust form of preprocessing would be desirable. Our model has difficulty dealing with abnormally bright areas of the brain, as well as lower regions of the brain. Part of the error can be attributed to not enough quality or diversity in our training data, however a form of pre-processing that is able to center brain tissue intensity more consistently would likely help the most. Additionally, automated skull stripping of the brain scans would improve the model by allowing it to not have to distinguish between brain and non-brain tissue.

Finally, adding additional pipelines to the model should improve accuracy. Kamnitsas et al. combined both a high resolution, low area with a low resolution, high area image patch in their model. This method slightly increased the accuracy of the results. However, it may also be desirable to combine the target patch pipeline with a pipeline focused on the mirror opposite of the brain area. As humans tend to label lesions both through abnormal features and asymmetry, feeding this information into the model would likely increase accuracy, especially with bright scans and non-brain tissue. However, it would also require careful pre-processing, as failing to center the brain scan could cause the model to fail.

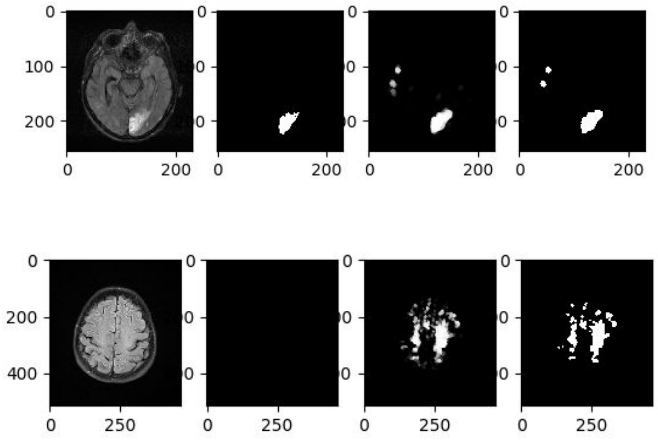
Overall, although our results are satisfying given our time limit and constraints, ultimately our segmentation could be greatly improved through these methods.

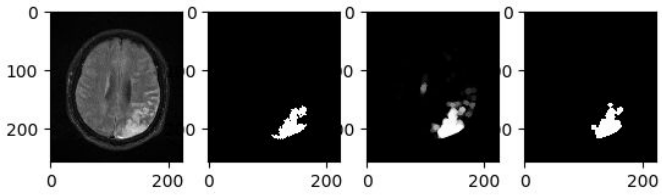
Figures



Figures 1-4

Graphs indicating the accuracy and similarity of the training and validation image patches during a training session. In these training sessions, 400 steps is equivalent to an epoch.





Figures 5-7
Segmentation examples on validation images. From left to right is the original image, the ground-truth segmentation, the probability heat-map of a lesion, and the final predicted segmentation.

References

Kamnitsas, Konstantinos, et al. "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation." *Medical Image Analysis*. 2017;36:61-78.