Automated Segmentation Mask Generation of MRI scans for Post-Stroke Lesions

Elizabeth Botbol Ponte
Department of Electrical Engineering
Stanford University
ebotbol@stanford.edu

Marcus Paul Lavengco
Department of Electrical Engineering
Stanford University
lavengco@stanford.edu

Oscar Rodriguez Guerrero
Department of Electrical Engineering
Stanford University
oguerrer@stanford.edu

Abstract

The goal of this project was to produce a segmentation algorithm that works well on ATLAS data set consisting of prepossessed MRI slices of patients with post-stroke brain lesions. Performance was measured by the dice coefficient produced by comparing the output lesion mask predicted by the selected model and a given ground-truth lesion mask. Enhancements were preformed on a given baseline model similar to AlexNet CNN, producing a Dice coefficient of 0.48, improving upon the original Dice coefficient of 0.15-0.19.

1 Introduction

According to the World Health Organization, 15 million people suffer stroke worldwide each year. Of these, 5 million die and another 5 million are permanently disabled. In the United States, stroke is the leading cause of serious, long-term disability, and the third leading cause of death, with more than 140,000 deaths each year.

Defining location and extent of irreversibly damaged brain tissue is a critical part of the decision-making process in acute stroke. Therefore, there is a great need for advanced data analysis techniques that could help to define these regions for diagnosis in a more reproducible and accurate way and eventually support clinicians in their decision-making process. However, the current standard for lesion segmentation on MRI scans is to manually trace the lesions, as a result of this demand on time and effort, this method is not suitable for larger sample sizes. As predictive algorithms improve, a long-term goal is for clinicians to use MRI to predict the likelihood of recovery, or more importantly, their likelihood of responding to different and more personalized types of therapies.

The goal of this project is to implement a segmentation algorithm that performs well on ATLAS. Performance will be measured by the Dice coefficient (Eq. 1), discussed later.

the most common metric used to evaluate segmentations of volumetric imaging data (Eq. 1), where TP, FP, FN represent the true positive, false positive, and false negative pixel counts of the lesion mask predicted by the algorithm compared to the ground-truth lesion mask. Paired with this metric we are given a fully-functional neural baseline to be able to compare to and improve from.

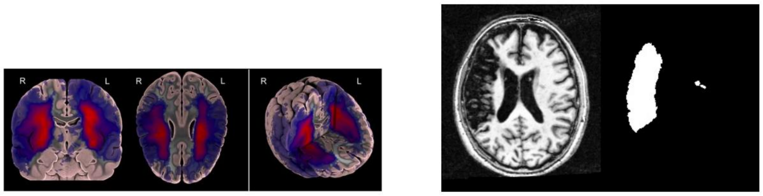
2 Dataset and Features

ATLAS (Anatomical Tracings of Lesions After Stroke) Release 1.1, is an open-source dataset consisting of 304 T1-weighted MRIs with manually segmented diverse lesions (Fig. 3) and metadata. The goal of ATLAS is to provide the research community with a standardized training and testing dataset for lesion segmentation algorithms on T1-weighted MRIs as a useful resource to improve the accuracy of current lesion segmentation methods.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

A subset of this dataset was also defaced and intensity normalized. For each MRI, brain lesions were identified and masks were manually drawn on each individual brain.

These images were collected primarily for research purposes and are not representative of the overall general stroke population (e.g., only including individuals who opt in to participate in a research study, and excluding individuals with stroke who cannot undergo MRI safely). The authors created a probabilistic spatial mapping of the lesion labels (Fig. 3) to visualize the distribution of lesion masks across the normalized ATLAS dataset. As previously stated, this provides the distribution of lesions included in this dataset, rather than the representation of a true stroke distribution.



(a) (b)

Figure 1: (a) A probabilistic lesion overlap map for lesions from the ATLAS R1.1 dataset [1]. (b) T1-weighted image scan from the ATLAS R1.1 dataset with manual mask

3 Related work

ATLAS 1.1 was released to the public only a few months ago in February 2018, thus there is no published research papers using the dataset at this time. However, there is extensive published work on image segmentation for biomedical applications, and more specifically, automated mask generation of brain lesions using MRI and CT scans. Olaf Ronneberger, et al [4] proposed U-NET, a convolutional network architecture for fast and precise of biomedical segmentation applications. This network can be trained end-to-end, and it consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. Similarly, Kayalibay Baris, et al [5] use an architecture consisting of contracting and expanding stages, where feature maps from the first stage are combined with feature maps from the second stage via long skip connections. Additionally, they propose using a loss function close to the dice similarity coefficient. Seyed Sadegh Mohseni Salehi, et al [6] use a similar implementation called the Tversky Loss. This loss function addresses data imbalance, which is a serious issue in medical image segmentation and one that can lead to predictions that are severely biased towards high precision but low recall, which tends to be undesirable in medical applications where false negatives are less tolerable than false positives. Due to the similarity of the problem addressed in both papers to our problem, we wa

Lele Chen, et al [3]. introduce a 3D Convolutional Neural Network based model to automatically segment gliomas. They use a novel approach that hierarchically segments different lesion regions based on their structure followed by a two-stage densely connected 3D CNN. Even though their work is based on volumetric data, they provide useful insight on how the size of the receptive field determines how much contextual information is taken into account as the model deepens.

4 Methods

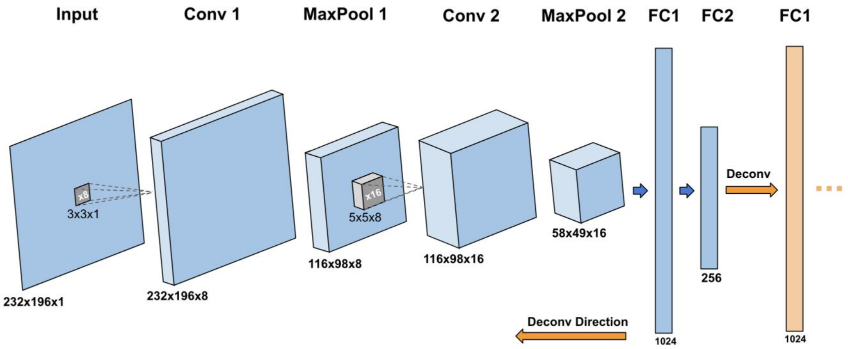


Figure 2: Baseline model developed and provided by David Eng

A baseline model designed by David Eng based of the Alex Network model and consisted of convolution, max pooling, and fully connected layers with dropout regularization, was given (Fig. 2). The model consisted of an encoding convolution path, in blue going right to left and ending at FC2, which feed into a decoding deconvolutional path, in orange and follows a mirrored right to left path starting at FC1 and ending at the after the Conv 1. The output produced a mask of same dimension of the input.

The related work gave a good starting point to understand the baseline model and insight as to how to improve components or alter its architecture. Prior to presenting the tested models, the score metric must be introduced to understand the design choices.

4.1 2D Dice Score

The 2D Dice score, also called coefficient, was used for comparing the similarity of two samples, equation 1.

$$Dice = \frac{2TP}{(2TP + FP + FN)} \tag{1}$$

Dice is computed pixel wise between the output mask and the target mask given. A true positive, TP, was produced by having the output mask pixel match the target mask pixel at the same point. False positive, FP, are produced when an output pixel produce an incorrect positive lesion prediction and false negatives, FN, are produced when the output missed a positive prodiction. The Dice values are averaged over all pi

4.2 AlexNet

The original AlexNet took in 224x224 RBG images and outputted a distribution over 1000 class labels. The AlexNet architecture, convolution (conv) layer followed by max pooling and ending with two fully connected (FC) layer, was similar to the baseline but only one directional, meaning that the output of the second FC layer, FC2 in figure 2, was used as the 1000 class label distribution. More differences from the baseline was that each FC used a ReLu output, 5 conv layers were used, number of input filters, and the the layer dimension. One unseen similarity between the baseline and AlexNext was the use of dropout regularization after to max pooling.

AlexNet, and the baseline, used conv layers for higher receptive field of deeper layers and reduction of computation cost. The max pooling was used to reduce over fitting and also decreased computation power needed. Dropout regularization after max pooling further reduces over fitting. The fully connected layers were used to interact with all the data at once and preform an operation which to everything into account.

4.3 Weighted Cross Entropy Loss Function

We use the weighted cross entropy as the loss function, which is formulated as :

$$Loss = \sum_{i=1}^{n_y} w_p * y_i log(\hat{y_i}) + (1 - y_i) log(1 - \hat{y_i}) \tag{2}$$

This formula is similar to the regular cross entropy loss function, however the positive predictions are weighted. Having $w_p >$ 1 decreases the false negative count, hence increasing the recall and $w_p < 1$ decreases the false positive count and increases the precision. Since the Dice score depends heavily on the TP, favoring a positive prediction is advantageous; alternatively one could think of it as predicting a false positive being better that missing a true positive in a field setting.

4.4 Baseline Model

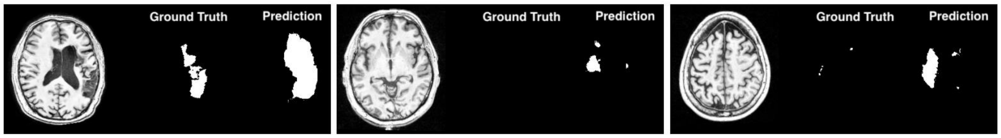
The base model had to ultimately create a 2D segmentation mask, so a deconvolution (deconv) layer decoding path had to be also implemented; this is partly seen in orange (Fig. 2). The deconv path was just the reverse path of the conv path, utilizing convolution transpose as deconv. This allow for the logit output to great a segmentation mask

5 Results and Discussion

The baseline model achieves a Dice coefficient of 0.15 to 0.19 after training for 10 epochs. Analysis of dev set predicted masks shows that the model tends to grossly overestimate.

We made a couple hypotheses as to why, the first being that the cross entropy (CE) weight was too large, the second, that the batch size was too small.

By design, the baseline loss function rewards true positives over true negatives according to a weight parameter. While this is desirable due to the relative rarity of a lesion, we hypothesized that using too high of a weight can cause the model to over predict. Over prediction may be practical for the application, but it degrades performance according to the evaluation metric.



(a) (b) (c)

Figure 3: Examples of the baseline model overestimating the size of the lesions

The original weight was set to 100. In alignment with our hypothesis, we chose to test weights of 10 and 1, with a weight of 1 effectively being unweighted cross entropy. Results show that with enough training time, lower weights do achieve higher Dice coefficients.

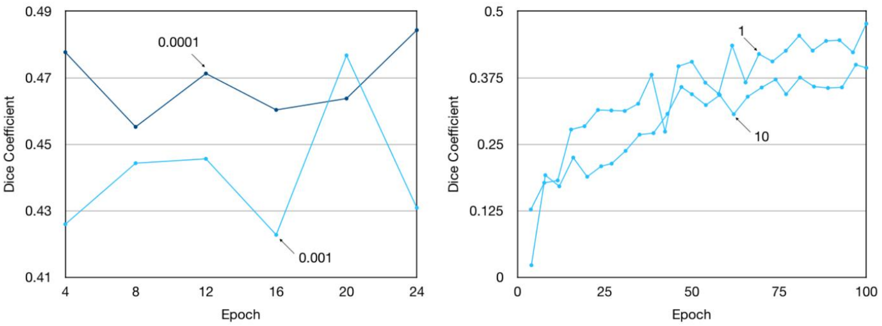


Figure 4: Dice Coefficient vs. epoch for different learning rates (left) Dice Coefficient vs. epoch for different weights (right)

Second, we hypothesized that due to the large variance of the dataset, a small batch size would cause the model to update inefficiently, causing it to overshoot, and prevent it from finding the optimum. Thus, we chose to test batch sizes of 10 and 200, comparing to the baseline of 100.

While our hypothesis was correct, shown by the model's inability to converge with a batch size of 10, a larger batch size showed significantly worse performance according to the dev Dice coefficient. We predict that this is due to the batch size's effect on the model's ability to generalize. This could be seen in the large difference between train and dev Dice coefficients, reaching as high as 0.2.

While optimizing for CE weight and batch size, we realized that after training for a while, the Dice coefficient ceased monotonic improvement, and showed large oscillations, exhibiting Dice coefficients that varied as much as 20%, relative to the maximum. Large oscillations imply that the model repeatedly overshoots the optimum. We attributed this behavior to having too large of a learning rate. However, training from scratch with a small learning rate would require longer overall training times. To overcome this, we proposed utilizing learning rate decay, allowing for fast training early, and fine tuning later. In order to test this, we simulated a decaying learning rate by further training with a smaller learning rate (on top of our earlier pre-trained weights).

This proved to be fruitful, as our oscillations decreased to less than 10%, allowing the exhibited Dice coefficient to stay closer to the maximum.

Aside from hyperparameter sweep and optimization, we also experimented with a few model changes.

The first was to implement the Tversky loss function. Because the Tversky loss attempts to imitate the Dice coefficient, implementing it would allow the model to optimize more directly for our evaluation metric. In practice the Tversky loss performed poorly due to its focus on solely true positives. The equation is shown below, where p_i is the probability of a lesion being present, and g_i is the ground truth.

$$Loss = 1 - \frac{\sum_{i=1}^{N} p_i g_i}{\sum_{i=1}^{N} p_i g_i + \sum_{i=1}^{N} p_i (1 - g_i) + \sum_{i=1}^{N} (1 - p_i) g_i} \tag{3}$$

The loss effectively updates the model only when a true positive pixel is found. With the sparsity of the lesions in our dataset, the model could not train efficiently, thus we abandoned it. The advantage of cross entropy loss is that it updates both on hits and misses. That being said, Tversky loss may prove to be useful in applications where the model is trained off of datasets densely populated with lesions.

Another model tweak was to use average pooling rather than max pooling. We believed that average pooling may allow deeper layers to retain more information from shallow layers, improving performance. Unfortunately, this caused the model to overfit more to the training set, shown by an increase in the difference between the train and dev Dice coefficients. We believe that the max pooling's second order effect as a regularizer gave it the advantage over average pooling.

The last model change that we experimented with was depth. The main goal in designing the architecture of the model was to ensure that the deepest convolution layers had receptive fields about the size of a lesion. By increasing the depth, the deeper layers would be able to "see" more of the original image, improving the model's ability to predict. We increased depth in a style similar to AlexNet, including additional "same" padding convolutions before each maxpool, convolving a total of three times before each size reduction. However, adding depth to the model caused it to overfit and overestimate, reducing performance on the dev set.

Shown are some examples of the predicted masks generated by the model. While the Dice score is far from the optimal value of 1, the model predicts with decent accuracy the overall shape and location of the lesion.

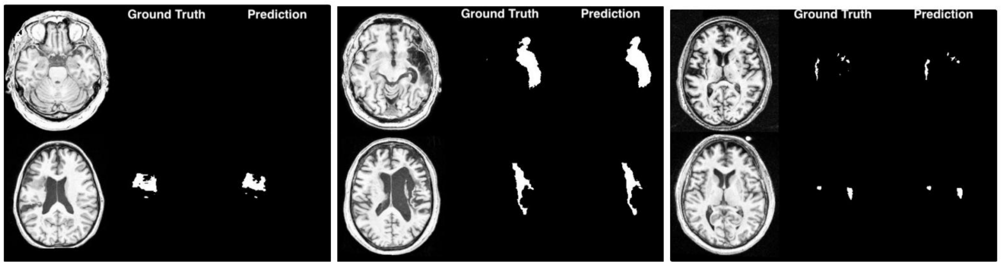


Figure 5: Examples of mask predictions made by the highest performing model

6 Conclusion and Future Work

In the end the highest performing model utilized the original architecture, a CE weight of 1 (or regular cross entropy, and a decaying learning rate (implemented manually rather than in the software).

Although many of the models performed well after long periods of training, skepticism arises when taking into account the nature of the dataset and the train/dev split method used. The ATLAS dataset is relatively small in terms of patient base, and may lack expressivity of the full range of brain lesion types. As the dataset includes only 220 patients, we cannot feasibly split the train and dev sets by patient, else we risk overfitting the model to specific lesions or lesion types. Due to this, we split the dataset on the slice level. Unfortunately, this may mean that the train and dev sets are highly similar, as lesions in different slices of the same MRI are highly correlated. Thus the train and dev sets may not be very independent, and may additionally lack expressivity.

Typically we are concerned with the model overfitting on the training set – this did occur in the current model, shown by the train and dev Dice coefficients differing by a nominal value of 0.1. However, due to the likely similarity between the train and dev sets, if the model overfits to the training data, it may see an unnatural boost in performance in the dev set.

However, if the dataset does prove to be expressive, or if the dataset were to be extended to do so, the current performance of the model may be sufficient enough to feasibly aid an MRI technician.

In order to make further advancements in performance, two paths of development are suggested.

The first is to develop a cascaded network. We are attempting to solve a complex problem in an end-to-end fashion with a limited dataset. By dividing the problem into several smaller problems, specifically binary classification, bounding box estimation, and then image segmentation, several models may be trained to specialize in accomplishing parts of the whole task. Drawbacks to this are that the entire cascaded network's performance is dependent on each individual model, and bottlenecks may incur. Additionally, the individual models may struggle in achieving high accuracy due to the limited dataset.

The second suggestion is to develop a volumetric model. This would likely see large improvements due to the high correlation of lesion shapes and sizes in adjacent slices of the same MRI. Again, the training would be severely limited by the small patient base included in the dataset.

7 Contributions and Acknowledgements

Development was largely a team operation with all members collaborating together and equally on research, theory, performance analysis, and coding.

Thanks to the teaching staff of CS230: Deep Learning, with special thanks to David Eng for development of the baseline model, and guidance!

References

[1] Sook-Lei Liew, et. al., A large, open source dataset of stroke anatomical brain images and manual lesion segmentations. Scientific Data, 2018; 5: 180011 DOI: 10.1038/sdata.2018.11 (2018).

[2] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification With Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, page 1097–1105 (2012).

[3] Li, X., Chen, H., Qi, X., Dou, Q., Fu, C. W., & Heng, P. A. . H-DenseUNet: Hybrid densely connected UNet for liver and liver tumor segmentation from CT volumes. arXiv preprint arXiv:1709.07330 (2017).

[4] Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. International Conference on Medical image computing and computer-assisted intervention 9351, 234–241 (2015).

[5] Baris Kayalibay, Grady Jensen and Patrick van der Smagt. CNN-based Segmentation of Medical Imaging Data. CoRR, abs/1701.03056 (2017).

[6] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, “Tversky loss function for image segmentation using 3D fully convolutional deep networks,” International Workshop on Machine Learning in Medical Imaging (2017).

[7] A. Krizhevsky, I. Sutskever, G.E. Hinton. ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25 NIPS’2012, (2012).