Classification of Brain Lesions from MRI images Using a Novel Neural Network

Udbhav Bamba[1], Deepanshu Pandey[1], and Vasudevan Lakshminarayanan[2]

[1] Department of Mathematics and Computing,
Indian Institute of Technology, Dhanbad, India
{udbhav.17je002781,deepanshup003.17je003256}@am.iitism.ac.in
[2] Theoretical and Experimental Epistemology Lab,
School of Optometry and Vision Science
University of Waterloo, Waterloo, Canada
vengu@uwaterloo.ca

There is considerable interest in using convolutional neural networks for computer aided diagnostics. In this paper, we present some recent results on stroke classification and brain lesion segmentation. Stroke is the leading cause of adult disability worldwide. There are barriers in accurate stroke lesion segmentation. Currently, Manually traced lesions are the gold standard for lesion segmentation on T1-weighted MRIs, but are labor intensive and require anatomical expertise. While algorithms have been developed to automate this process, the results often lack accuracy in tracing the lesions. Here, we present our results using deep Convolutional Neural Networks on the public ATLAS (Anatomical Tracings of Lesions After Stroke) dataset from ICPSR at the University of Michigan, This dataset consists of T1-weighted MRI's with manually segmented lesions and metadata. Here, we utilize the deep U-Net architecture with 3D convolutions. The architecture consists of an encoder and a corresponding decoder along with dense residual connections from the encoder to the decoder (Figure 1). These residual connections aim to leverage both low level features (from one level lower encoder block) and high level features (from one level higher encoder block) at each encoder step or block. The architecture also makes use of an attention gate for the decoder part to suppress irrelevant parts of the incoming input while highlighting the salient features and assigning them higher weights (Figure 2). Apart from this, the model also uses a multiple-input multiple-output architecture to enhance the features generated at each encoder/decoder block, so that the features are able to recognize the lesions even at a lower level. We further analyze the use of Inception, Residual and Inception-Res blocks as individual blocks for the encoder and compare their respective performance. We also analyze the conventional single input-output and single input-multi output model on our current architecture. The dataset used had 239 examples of 3D MRI images of dimensions (197 x 233 x 189) along with their corresponding segmentation maps. The data was mean normalized and histogram equalization was performed. In addition, we augmented the data to increase the size of the dataset. The data was split into training, validation and testing examples in the ratio 7:2:1. The model was trained on a NVIDIA Tesla T4 GPU. The batch size was kept as 16 due to constraints on computation power and storage. During training, we

used the SGD optimizer with Focal Tversky loss function, using Dice score and AUC as metric. The model was trained for around 60 epochs. We were able to achieve a Dice coefficient of 0.53 with AUC of 0.95 with a single end-to-end model, unlike previous models that either focus only on one part of the brain or use other metadata, and matched their performance in every aspect. Apart from the training, the major issue with the data was noise and distortion in the images. Use of Generative Adversarial Networks or denoising auto-encoders for noisy parts can be useful, though it is debatable as to how much this will increase the performance of the model. The part of the brain detected in the image is also observed to interfere with the results. For this, using skull stripping algorithms is helpful, but they also scraped away some "parts" of the brain, which might be important in predicting the lesions. Nonetheless, extremely precise skull stripping, if possible, can be applied and can result in improvements in performance.

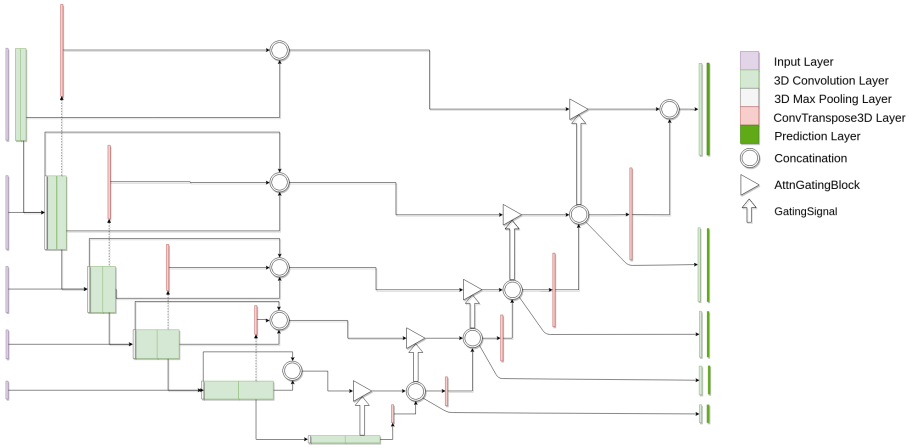

Fig. 1: Model architecture

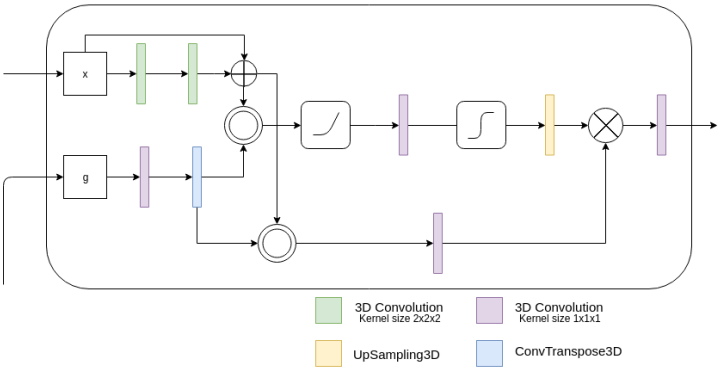

Fig. 2: Attention gate used in the decoder layer