Detecting Brain Tumors in Low Quality Magnetic Resonance Images

Venkat Yerramsetti
venkaty@stanford.edu

https://youtu.be/QT5JTdtXMmE

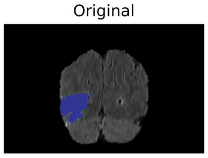
Introduction

Quick and accurate detection of brain tumors using MRIs is very helpful for treatment planning and evaluation. Many CNN models like U-Net, V-Net, etc. were shown to produce very good segmentation results for medical images. However, in reality, MRI quality could be degraded due to device calibration errors, movement of subject, etc. In this project, I explored whether the proven, existing models perform just as well on the degraded images by training the models on an artificially created low quality dataset. Unfortunately, results are inconclusive due to bad data.

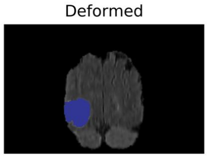
Data and Preprocessing

- Used BraTS 2018 dataset.
- It contains 285 sets of MRIs with tumor labels.
- Train/dev/test set split is 200/55/30.
- Each image is a 240 x 240 x 155 volume.
- The depth is padded to 160 to to work correctly with the convolutional and up-convolutional layers in the 3D models.
- Input is normalized to zero mean and standard deviation of one.
- A low quality dataset is generated by applying small random deformations to the original data. Train/dev/test sets contain equal proportions of original and generated images.

Original



Deformed



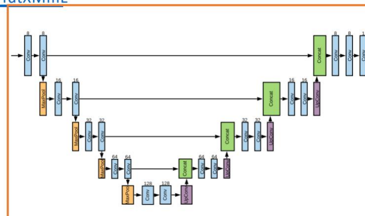
Example slice from the original image (left) and the corresponding deformed image (right).

Models

3 models were implemented.

- 2D Unet
- 3D Unet
- 3D Vnet

All models are based on the original architectures with few changes to adapt them to the data at hand and to satisfy resource constraints.



Structure of 2D and 3D U-Net model; Number of kernels is shown on top of each block.

Loss Functions

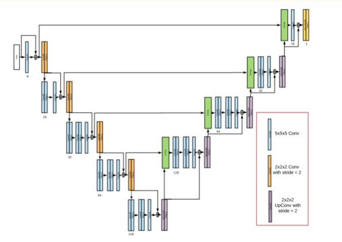
3 loss functions were used

- Pixelwise cross entropy loss for base line model
- Dice loss

$$dice_{loss} = -\frac{2 * \sum(\hat{y} \ * y)}{\sum\hat{y} + \sum y}$$

- A new compound loss incorporating feature loss based on pre-trained VGG19 network.

$$L = vgg_{loss} + dice_{loss}$$



Structure of 3D V-Net model. Number of kernels is shown at the bottom of each block.

Training and Results

- Initial training spanned 50 epochs with mini-batch size of 1 for 3D models and 155 for 2D model. Learning rates range from $1e^{-3}$ to $1e^{-5}$ for the three models.
- Models suffered from covariate shift due to a bug in input data pipeline.
- They also suffered from high bias and variance. L2 regularization resolved the variance problem, but bias problem persisted even after training for 150 epochs.

Model	Train (N = 200)	Dev (N = 55)	Test (N = 30)
Unet2D	0.7499	0.6438	0.6860
Unet3D	0.7683	0.6322	0.6882
Vnet3D	0.8515	0.6225	0.6492
Unet2D (w/ vgg)	0.6885	0.6345	0.6785

Results after training for 150 epochs. Best dice scores on high quality MRIs are in 85–90% range

Analysis

- A data pipeline bug produced a different train set for each epoch, hurting the model's ability to learn.
- Ground truth labels for some of the generated low quality images were corrupted, which resulted in high bias.
- Test set happened to have more images with correct labels, which might explain the better test results.

Conclusion

Having a non-faulty data pipeline and accurate data and ground truth labels is very crucial for evaluating models. Due to the faulty data, the results are inconclusive as to whether existing models can perform well on low quality MRIs. Next steps are to fix and validate the generated data and to retrain the models with that data. Finally, the models should be evaluated on a real dataset of low quality images.

Acknowledgement

- I thank Dr. Olivier Keunen and Dr. Wintermark's lab for providing access to computing resources.

References

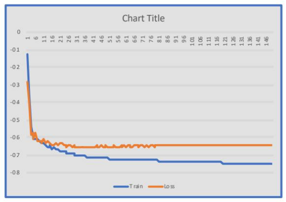
- Menze BH et. al. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)
- Ozgun Cicek et. al. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation
- Fausto Milletari et. al. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation

Data

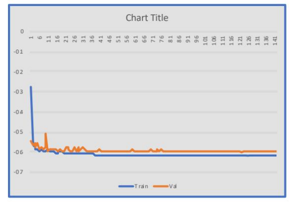
- BraTS 2018 dataset contains
- 285 sets of MRIs with tumor labels
- Train/dev/test set split is 200/55/30
- Each image is a 240 x 240 x 155 volume

Data Preprocessing

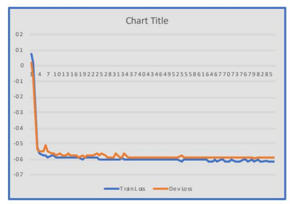
- Generated by using random deformations to the original data. Train/dev/test sets contain equal proportions of original and generated images.
- To work correctly with the convolutional and up-convolutional layers in the 3D models, the depth dimension of the input images is padded to 160 slices. This is done on the fly as images are fed into the model.
- Input is normalized to zero mean and standar deviation one.



At a learning rate of 1e-4, model started overfitting very early at ~25th epoch.



After applying L2 regularization, the overfitting problem was resolved, but the model was underfitting and plateaued at 60%



Loss Functions

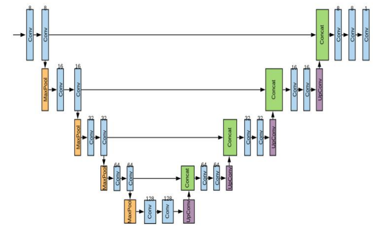
3 loss functions were used

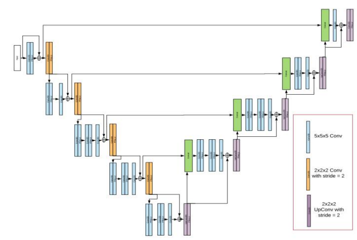
- Pixelwise cross entropy loss for base line model
- Dice loss

$$dice_{loss} = -\frac{2 * \sum(\hat{y} \ * y)}{\sum\hat{y} + \sum y}$$

- A new compound loss incorporating feature loss based on pre-trained VGG19 network.

$$L = vgg_{loss} + dice_{loss}$$





- Unet
- 2D and 3D versions use same architecture.
- Dice loss function
  $\frac{2*\sum(\hat{y} * y)}{\sum\hat{y}+\sum y}$
- A new loss function
- $L = vgg_{loss} + dice_{loss}$
- Interestingly, has a regularization effect.

- Each image is a 240 x 240 x 155 volume.
- For 3D models, the whole image is fed into the model as one example. **One channel only**
- For 2D models, a single image is split into 155 2D images before feeding into the model.

- Each image has 4 modalities, but I used only one – FLAIR because I was
- Preliminary training was done with all modality images. Later stages us