The re-evaluation of Brain Tumor Segmentation using Cascaded Anisotropic Convolutional Neural Network

UNIVERSITY OF TORONTO

Speaker: Junming Zhang
Group: Braindon
Teammate:
Zhuoheng Huang, Fei Ju
Yongzhao Wu, Junming Zhang

Abstract

We reimplement a cascade of three CNNs by Wang (GitHub [1]) to detect three hierarchical tumor regions of the brain tumor (whole tumor, tumor core, enhanced tumor core) in 3D MRI scans of 4 modalities: T1-weighted, contrast enhanced T1-weighted (T1c), T2-weighted and Fluid Attenuation Inversion Recovery (FLAIR) of HGG and LGG region. Then we improved the model and achieved significant progress.

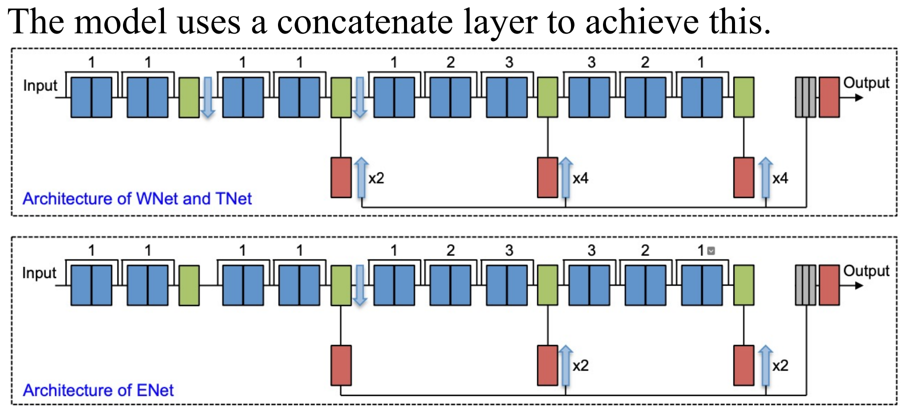
Objectives

We have two objectives for this project. First, we found a decent method to segment the three tumor regions in the 3D sample and reimplemented it. We used the model whose performance ranked the 2nd in the BraTS 2017 competition (GitHub [1]), and then we tried to further improve the model performance by preprocessing the input and postprocessing the output and different loss functions..

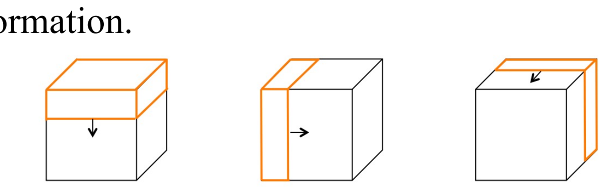
Methods & Individual contributions

Model architecture

1. We used a triple cascaded framework (Yongzhao has introduced that).
2. Anisotropic Convolutional Neural Networks (neural network kernel): shorten the dimension of receptive field in the inter slices direction (out plane) to keep the balance of model complexity and receptive field size.
3. Anisotropic and Dilated Convolution: the receptive field is split to a 3 x 3 x 1 intra-slice kernel and a 1 x 1 x 3 interslice kernel. After the downsampling layers, the intra-slice kernel is dilated to increase the receptive field.
4. Combine the input and the output of a block in some way to smoothen information propagation and speed up training convergence.
5. Multi-scale Prediction: Combine the local and low-level features represented at shallow layers and global and high-level features at deep layers. The model uses a concatenate layer to achieve this.



6. Multi-view Fusion: since the out plane receptive field is small, the model fuses the segmentation result from three orthogonal views to use 3D contextual information.



Improvements:

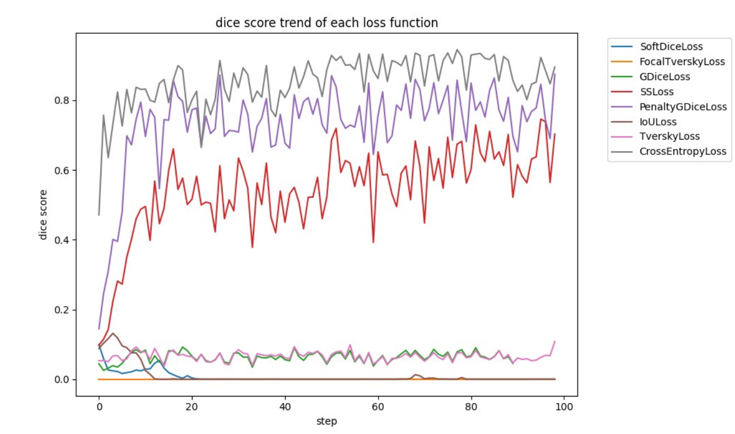
1. Applied different loss functions (Cross Entropy loss in PyTorch beat other dice loss variants we found)
2. Applied the Conditional Random Field (CRF) to postprocess the test result.
3. Applied the data augmentation methods (blurring, anisotropy, and elastic deformation) to preprocess inputs.

My contributions:

1. Reimplement the test workflow.
2. Proposed the CRF method and found the adopted library.
3. Researched and implemented the blurring and anisotropy data augmentation methods.
4. Tested different loss functions.

Loss functions

I compared the dice scores produced by different variants of dice loss) (GitHub [2]) $(1-\frac{2(y\cap p)}{|y|+|p|})$ and the cross-entropy loss $(-y\log p-(1-y)\log(1-p))$ of the PyTorch library, which are both common loss functions used in the image segmentation. We test the mean, median and standard deviation of loss during training steps and found the cross-entropy loss achieves best mean dice score and relatively low standard deviation.



Loss function	Dice Score Mean	Dice Score Median	Dice Score Standard Deviation
SoftDiceLoss	0.0060	0.0001	0.0151
FocalTverskyLoss	0.0001	0.0001	0.0000
GDiceLoss	0.0636	0.0649	0.0148
SSLoss	0.5422	0.5628	0.1261
PenaltyGDiceLoss	0.7311	0.7506	0.1230
IoULoss	0.0105	0.0001	0.0296
TverskyLoss	0.0657	0.0650	0.0132
CrossEntropyLoss	0.8542	0.8579	0.0733

Conditional Random Field (CRF)

CRF is a post processing method used in machine learning and pattern recognition, it takes the advantage of contextual information and state of neighbors to improve the prediction result. Another famous example is Hidden Markov Model (HMM), but CRF is more powerful for higher generalization ability (feature function application). Typical CRF is not good at long-range connections, but our adopted fully-connected CRF (GitHub [3]) shows a strong ability to solve this problem by establishing edge potentials on all pairs of pixels, while the edge potential is defined by a linear combination of Gaussian kernels.

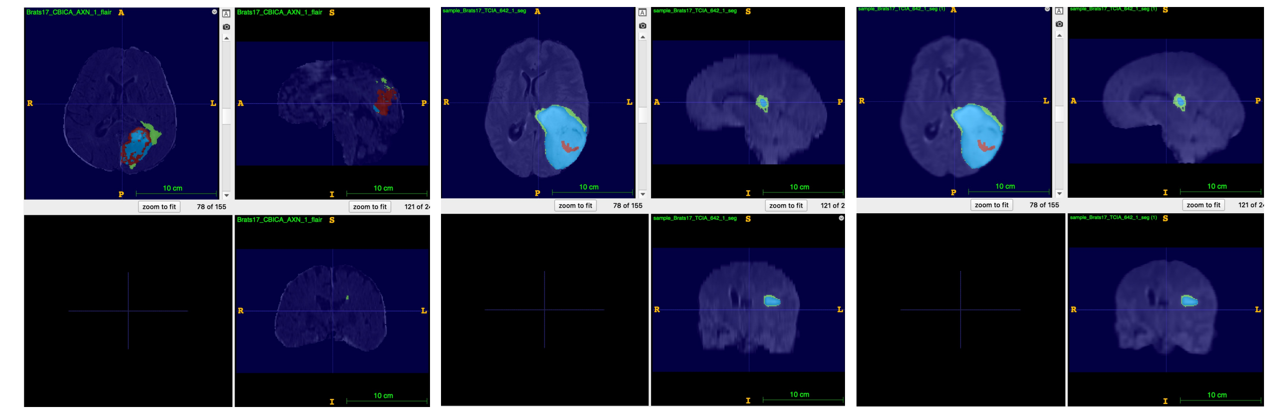
Our test results show that in general the CRF can improve the model output, but the dice score on one of the three labels might be worse than before.

Data augmentation

In the real practice, the samples might be noisy or contaminated, and not all samples will have the same properties, such as resolution. For example, a device for scanning might produce a bad result because of technological limitation. Then we can randomly augment the input to make the samples closer to the real situation, or share the similar properties with other samples, so the model trained with the augmented data have the better generalization ability. I have contributed to two augmentation methods: blurring and anisotropy.

1. Blurring: Use a Gaussian filter to smooth/blur a sample
2. Anisotropy: In practice, the voxel (3D pixel) of a sample is not always a cube, so we use this method to downsample along one axis.

Three samples are attached in the next column, the left one is not augmented, the middle one is augmented with anisotropy, and the right one is augmented with blurring.



Here are comparisons of performance (dice scores) generated by the model trained with raw data and the model trained with augmented data. The left table shows the result for the raw data, and right table shows the result of the randomly augmented data. It shows that the segmentation of enhanced tumor core is improved, but that of the whole tumor is not as precise as before (both mean and standard deviation of dice score.).

	Whole Tumor	Tumor Core	Enhanced Tumor Core
Mean	0.9173	0.8060	0.6392
Standard Deviation	0.0582	0.1815	0.3064

	Whole Tumor	Tumor Core	Enhance Tumor Core
Mean	0.8579	0.8033	0.7891
Standard Deviation	0.1237	0.1643	0.2160

Conclusion

Our experiments show that:

1. Cross Entropy Loss works better than different variants of dice loss functions in the brain tumor segmentation task for the triple cascaded network structure.
2. CRF and data augmentation can improve the segmentation results of some labels but may impair the segmentation results of other labels.

References

GitHub repository:

[1] Model: https://github.com/taigw/brats17

[2] SegLoss: https://github.com/JunMa11/SegLoss

[3] crfseg: https://github.com/migonch/crfseg#readme

[4] torchio: https://github.com/fepegar/torchio

Paper:

[1] Automatic Brain Tumor Segmentation using Cascaded Anisotropic Convolutional Neural Networks. https://arxiv.org/abs/1709.00382

[2] NiftyNet: a deep-learning platform for medical imaging." Computer Methods and Programs in Biomedicine. https://arxiv.org/pdf/1709.03485

[3] Loss odyssey in medical image segmentation. https://doi.org/10.1016/j.media.2021.102035

[4] Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. https://arxiv.org/abs/1210.5644

[5] Conditional Random Fields as Recurrent Neural Networks https://arxiv.org/pdf/1502.03240.pdf

[6] TorchIO: A Python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning. https://doi.org/10.1016/j.cmpb.2021.106236