U-Net Based Architectures for Medical Image Segmentation

Dillon Laird
2019

Abstract

U-Net has become a popular network for many segmentation tasks, particularly medical segmentation. However, the overall network has remained relatively unchanged since its introduction in 2015. Here we examine the basic U-Net architecture under different loss functions and components on a medical segmentation task, similar to how the original U-Net was evaluated. We find that both a modified U-Net with a NAS cell and Attention U-Net lead to better performance.

Data

The dataset is from the Medical Segmentation Decathlon challenge. We use the data provided in the **Task01_BrainTumour** dataset which consists of 750 multi-parametric medical resonance imaging scans (MRI) scans. The multi-parametric MRI sequences include 4 different modalities, so the input images have a channel dimension of 4. You can see an example of one of the modalities in Figure 1. There are also 3 classes of segmentation, edema (swelling), non-enhancing tumour and enhancing tumour. For the purposes of our experiments, we collapse all classes into a single class. We use 484 volumes for training and leave 266 for testing. Because each volume contains many 2D images, we end up with 58 thousand images for training and 11 thousand images for our validation set.

Example Input Data

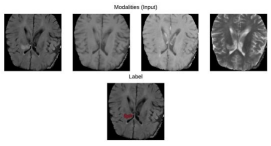

Figure 1: An example of the 4 modalities captured in the dataset along with the segmentation label.

Model

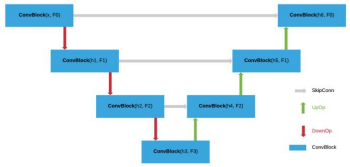

Figure 2: Caption

We can parameterize the U-Net architecture with a few key modules: ConvBlock, SkipConn, UpOp and DownOp.

$$h^D_{i+1} = \text{DownOp}(\text{ConvBlock}(h^D_i, F_i))$$
$$h^U_{i+1} = \text{UpOp}(\text{ConvBlock}(\text{SkipConn}(h^U_i, h^D_{2(L-1)-i}), F_{2(L-1)-i}))$$

Where h^D_i is the i^{th} hidden layer going down (comint out of a DownOp) and h^U_i is the i^{th} hidden layer going up (coming out of an UpOp). F_i is the number of feature maps for the i^{th} ConvBlock. You can see these operations and how they are combined in Figure **??**.

Experiments

Module	ConvBlock	SkipConn	DownOp
U-Net	$2 \times$ (Conv k3×3, ReLU	Concatenate, Crop	Max Pool k2×2
Attn U-Net	$2 \times$ (Conv k3×3, ReLU	AttnCell	Max Pool k2×2
NAS U-Net	NASNet-A Normal Cell	Concatenate, Crop	Max Pool k2×2
NAS Red. U-Net	NASNet-A Reduction Cell	Concatenate, Crop	Conv skip2×2
Efficient U-Net	MBConvBlock	Concatenate, Crop	Max Pool k2×2

Figure 3: The module implementation details of different architectures. Note UpOp is left out because they are all 2×2 transpose convolutions.

For preprocessing we first center crop all the images to 144×144. We build each model such that it has 8.5 million parameters to make the comparisons more fair. When training we pick the best model based off of its validation IoU score. All models are trained with the Adam optimizer [**?**] with a learning rate of 0.0001, a batch size of 40 (the largest batch size that could be fit into memory) and 20 epochs or until convergence.

3 **Architecture**	**IoU**
U-Net	0.9134
NAS U-Net	**0.9143**
NAS Reduction U-Net	0.9073
Attn U-Net	**0.9140**
Efficient U-Net	0.6575
3	

Figure 4: U-Net architectures with their associated IoU scores.

We explore several variations of the U-Net architecture. Each one consists of changing either the ConvBlock, SkipConn, UpOp or DownOp module as show in Figure 3. We examine two different architectures based off of NASNet utilizing the normal and reduction cells from the NASNet-A block. Additionally we examine Attention U-Net from Oktay et. al. which utilizes an attention SkipConn module and Efficient U-Net from Tan et. al. which uses an MBConvBlock as it's ConvBlock module.

Analysis

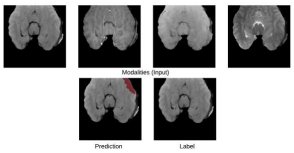

Figure 5: An example of a false positive example prediction, hence no label.

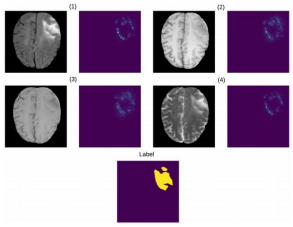

Figure 6: A saliency map for a particular example.

Conclusion

We present a way to abstract U-Net into 4 separate modules: ConvBlock, SkipConn, UpOp and DownOp. We then examine several U-Net architectures by testing different modules and find that using attention cells for the SkipConn and NASNet normal cells for the ConvBlock both lead to better performance for medical segmentation. We hope this abstraction can be used to quickly explore new forms of U-Net with improved performance.