U-Net Based Architectures for Medical Image Segmentation

Dillon Laird

June 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.

1 Introduction

U-Net is a popular network choice for image segmentation tasks. Its simple structure makes it easy to implement and use. However, the original structure of U-Net has remained relatively unchanged since it was introduced. The goal of this paper is to examine the U-Net architecture under different loss functions and components. We experiment with these changes using a medical segmentation task akin to how U-Net is typically evaluated.

2 Related Work

There has been much work done using segmentation on medical images. One of the more widely used model architectures is U-Net by Ronnenberger et al. [5] who uses an architecture that consists of a series of downsampling maxpools and upsampling transpose convolutions with skip connections in between to create a U-like shape. Also drawing from this work is the Attention U-Net by Oktay et al. [4] who creates an network similar to U-Net but uses an attention gating mechanism in the skip connections. Segmentation is also used outside of medical images on datasets such as PASCAL VOC 2012 and Cityscapes. The DeepLab architectures have been very popular with the most recent paper from Chen et al. [2] introducing DeepLabv3+ which utilizes and encoder-decoder like structure with an atrous spatial pyramid pooling layer.

3 Data

The dataset is from the Medical Segmentation Decathlon challenge [6]. 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

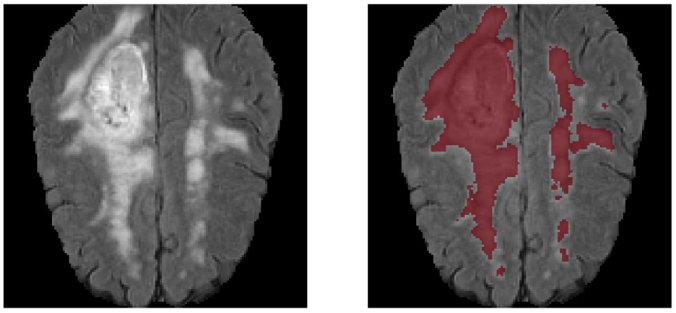

Figure 1: The left image shows a T2-FLAIR modality of the MRI scan while the right image shows the segmentation label in red.

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.

4 Model

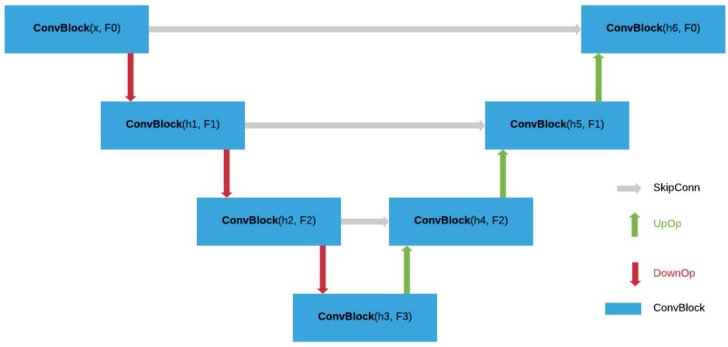

Figure 2: A generalized U-Net architecture split into 4 main modules.

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

Loss	IoU
Dice Loss	**0.9134**
Dice Loss Sq.	**0.9138**
BCE	0.9090
Focal Tversky	0.9009

Figure 3: Loss functions with their associated IoU scores.

i^{th} ConvBlock. You can see these operations and how they are combined in Figure 2. Note that the original U-Net paper shows a similar figure but they do not abstract the different components and instead list out the specific operations used for each operation.

5 Experiments

5.1 Setup

For preprocessing we first center crop all the images to 144×144. We then standardize by the image mean and standard deviation. During training we use ZCA whitening, shearing between 0 and 2 degrees, and random horizontal and vertical flipping. 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 [3] 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. All models were run on AWS p2.xlarge instances which are single NVIDIA K80's with 12 GiB of GPU memory.

5.2 Loss Functions

We examine several different loss functions. The first two loss functions we look at are the Dice loss, or 1 minus the Sorensen-Dice coefficient and the Dice loss squared which has the denominator terms squared. The Dice loss squared was chosen because it had previously shown promising results. The third loss function is the binary cross entropy loss, a popular choice for image segmentation tasks and finally we look at the focal Tversky loss which was shown by Abraham et. al. [1] to perform well on medical image segmentation tasks, however we do not see this in our experiments.

The Dice loss and Dice loss squared loss functions had similar validation performance in terms of IoU as shown in Figure 3. However, the Dice loss produced more stable results than the Dice loss squared. For all of our U-Net architecture experiments we use the Dice loss.

5.3 U-Net Architectures

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 4. Our main U-Net architecture differs from the original U-Net architecture in several minor ways. For up sampling we use a transpose convolution. We also add a batch normalization layer.

We examine two different architectures that utilize components from NASNet [8]. The first takes the normal cell from NASNet-A and uses it as the U-Net ConvBlock. The second,

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

Figure 4: The module implementation details of different architectures. Note UpOp is left out because they are all 2×2 transpose convolutions.

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

Figure 5: U-Net architectures with their associated IoU scores.

uses normal cells for the ConvBlock on the upsampling part of U-Net but uses the reduction cells for the ConvBlock on the downsampling part as well as the DownOp. Additionally we examine Attention U-Net from Oktay et. al. [4] which utilizes an attention SkipConn module and Efficient U-Net from Tan et. al. [7] which uses an MBConvBlock as it's ConvBlock module.

The NAS U-Net and Attention U-Net had the best IoU performance, both reaching approximately 0.914. Surprisingly, using the NAS reduction cells in the NAS Reduction U-Net actually hurt performance, and it's IoU was below the normal U-Net. Efficient U-Net performed significantly worse at 0.658 IoU. In addition to this we also had a difficult time fitting the model with the correct hyperparameters into memory despite having the same number of learnable parameters as all the other models. The Efficient U-Net trained 1.3 times slower than the regular U-Net and NAS U-Net trained roughly 1.9 times slower while Attention U-Net does not encur any slow down during training.

6 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.

7 Acknowledgements

We would like to thank Juan Sebastian Vega for his implementation of U-Net and Amol Tandel for his insights into abstracting U-Net.

8 Appendix

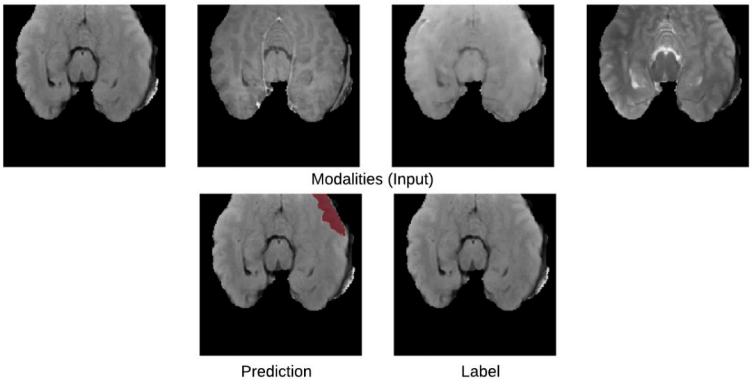

Figure 6: An example of a false positive example prediction, hence no label.

From Figure 6 you can see an example of a false positive. The 1st and 4th modalities appear to have slightly lighter areas in the upper right portions of the scan which may have lead to the model predicting a tumour there. However, the lower

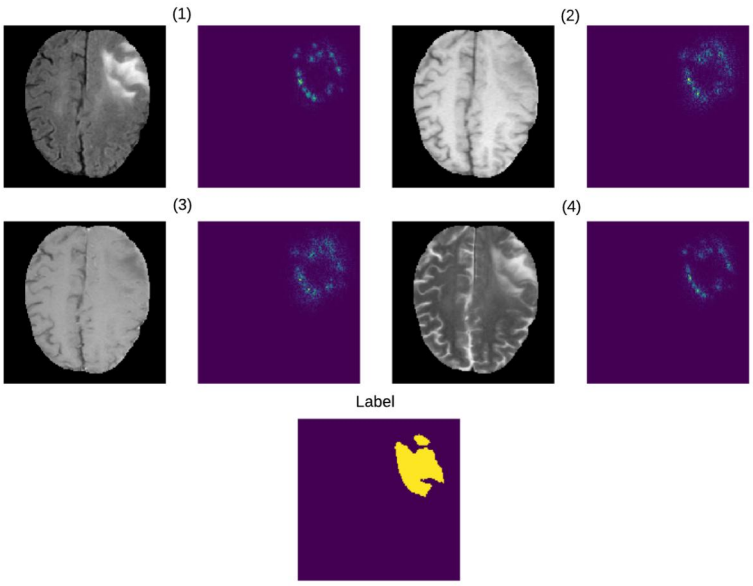

Figure 7: A saliency map for a particular example.

In Figure 7 you can see the saliency map for a particular example. The saliency map was taken by propogating the gradient back to each input modality. A few interesting observations, the gradients are more diffuse on the 2nd and 3rd modalities which contain less features of the tumour. On the 1st and 3rd, the gradients outline the tumour but don't highlight the area of the tumour itself. The gradient intensity is dotted along the edges of the tumour and not smooth.

References

[1] Nabila Abraham and Naimul Mefraz Khan. A novel focal tversky loss function with improved attention u-net for lesion segmentation. *arXiv preprint arXiv:1810.07842*, 2018.

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[4] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[6] Amber L Simpson, Michela Antonelli, Spyridon Bakas, Michel Bilello, Keyvan Farahani, Bram van Ginneken, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv preprint arXiv:1902.09063*, 2019.

[7] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[8] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.