Report of CADA Aneurysm Segmentation

Zhenfeng He[1,2]

[1]College of Computer Science and Technology, Zhejiang University
[2]Real Doctor AI Research Centre, Zhejiang University

Abstract. The segmentation of aneurysms has direct diagnostic and modeling significance. Our method implements a network based on nnUNet to segment Aneurysm regions.

Keywords: Segmentation Aneurysms

1 Introduction

The segmentation of aneurysms has direct diagnostic and modeling significance. For example, the use of segmentation in diagnosis can help physicians make faster decisions or improve the success rate of surgery in subsequent operations. Aneurysm segmentation is a key component in quantitative medical image analysis. It can directly detect and assess the size of aneurysms, and indirectly support modeling and monitoring the risk of rupture. For this reason, this article proposes an effective aneurysm segmentation method for the CADA-AS competition. Specifically, our method implements a network based on nnUNet [1] to segment vertebral regions. After verification, under the given indicators, the performance of our method has reached a good level.

2 Aneurysm Segmentation

2.1 Preprocessing

nnU-Net needs information about the form of its input channel. If the modality is not CT, nnU-Net normalizes the intensity value by subtracting the average value and dividing by the standard deviation. If the modality is CT, all foreground voxels in the training set will be collected, and the intensity value will be automatically cropped according to 0.5% and 99.5% of these values. In order to conform to the typical weight initialization method, the global foreground mean and standard deviation are then used to normalize the data.

The described scheme is independently applied to each situation and each mode. Voxel spacing. nnU-Net collects all the intervals in the training data and selects the median as the target interval for each axis. Then, use third-order spline interpolation to resample all training cases. Anisotropic spacing (where the out-of-plane spacing is three times the in-plane

spacing) can cause interpolation artifacts. In this case, the nearest neighbor is used to perform out-of-plane interpolation. For the corresponding segment labels, the spline interpolation is replaced by re-sampling each label separately using linear interpolation.

2.2 Training Procedure

Configured independently of each other, designed and trained three U-net models: a 2D U-Net, a 3D U-Net and a cascade of two 3D U-Net models, the first of which generates low-resolution segments , And then subdivided and refined by the second model. The only significant changes to the original U-Net architecture are the use of padding convolution to achieve the same output and input shape, instance normalization and Leaky ReLU (instead of ReLU). Network hyperparameters. According to the shape of the preprocessed training data, specific instances of U-Net can be adjusted. Specifically, nnU-Net automatically sets the batch size, patch size, and number of pooling operations for each axis, while keeping the memory consumption within a certain budget. Therefore, a large patch size is better than a large batch size (the minimum batch size is 2) to maximize the capture of the spatial context. Completing the merging along each axis until further merging will reduce the spatial size of that axis to below 4 voxels. All U-Net architectures use 30 convolutional filters in the first layer, and this number is doubled in each pooling operation. If, in a typical training situation, the selected patch size is less than 25% of the voxels, the 3D U-Net cascade will be additionally configured and trained based on the down-sampled version of the training data. If the patch size is too small to cover the full resolution, the cascade is designed to enable nnU-Net to still acquire enough context. For network training, all UNet architectures have undergone five-fold cross-validation training. One period is defined as processing 250 batches. The sum of the cross entropy loss and the dice loss is used as the loss function. Adam is used as an optimizer for stochastic gradient descent, and its initial learning rate is 3×104, and the l2 weight decays to 3×105. Whenever the exponential moving average of training loss has not improved in the last 30 cycles, the learning rate will drop by a factor of 0.2. When the learning rate drops below 106 or 1000 epochs, training will stop. In the process of training using the batchgenerators framework, we can dynamically apply data augmentation. Specifically, we use elastic deformation, random scaling and random rotation, and gamma enhancement. If the data is anisotropic, the spatial transformation is applied as a 2D transformation on the plane.

2.3 Testing

The operation of the test set is similar to that of the previous training set and the data preprocessing of the validation set, including linear interpolation of slice thickness, normalization of HU value, etc. The only difference is that the aneurysm label is obtained through our aneurysm segmentation model. After the data is preprocessed, the aneurysm is segmented. After generating the segmented data, we perform a posteriori operation on the data. The result of segmentation may be inaccurate. In order to avoid the influence of these data on the actual prediction, we use the maximum connectivity component to optimize the generated data.

2.4 Inference

The sliding window method is used to predict cases, and the patch size between predictions overlaps by half. This increases the weight of predictions closer to the center relative to the border. Apply test time data enhancement by mirroring along all axes. nnU-Net integrates a combination of two U-Net configurations (2D, 3D and cascade), and automatically selects the best single model or integration for test set prediction based on cross-validation results. For the selected configuration, nnU-Net also uses the five models generated by cross-validation for integration.

3 Model

3.1 Loss Function

The original loss function is dice loss, which is the evaluation index of this competition. However, the result is not good in training. After observing the data, we find that the data of the validation set is very unbalanced, and there are many false positive samples. After trying to use the loss function which combines focal loss and weighted dice loss, the performance are improved.

3.2 Optimizer

We used Adam Optimizer [2] as our optimizer, which has some metrics.

Adam algorithm is an algorithm that performs a step optimization on a random objective function. The algorithm is based on adaptive low-order moment estimation. Adam algorithm is easy to implement, and has high computational efficiency and low memory requirements. The diagonal rescaling of the Adam algorithm gradient is invariant, so it is very suitable for solving problems with large-scale data or parameters. The algorithm is also suitable for solving non-stationary problems with large noise and sparse gradients. Hyperparameters can be explained intuitively and only need a few adjustments. Adam algorithm is similar to some other algorithms. The paper [2] analyzes the theoretical convergence of Adam's algorithm, provides a range of convergence rate, and proves that the convergence rate is optimal under the framework of online convex optimization. The empirical results also show that the Adam algorithm is comparable to other stochastic optimization methods in practice. Finally, the author discussed AdaMax in the paper, which is a variant of Adam based on infinity norm.

Adam is an effective stochastic optimization method. It only requires a first-order gradient and requires only a small amount of memory. This method calculates the adaptive learning rate of different parameters through the estimation of the first and second gradients; Adam's name comes from Adaptive moment estimation. The Adam method combines the advantages of two recently popular methods: AdaGrad which is effective on sparse gradients, and RMSProp, which is effective in non-stationary and online problems. Have very good performance;

4 Result

After 150 epochs of training, the dice value converged to 0.9.

5 References

1. Fabian Isensee, Paul F. Jäger, Simon A. A. Kohl, Jens Petersen, Klaus H. Maier-Hein Automated Design of Deep Learning Methods for Biomedical Image Segmentation arXiv preprint arXiv:1904.08128 (2020).
2. Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.