Report

Jing Zhang
LITIS
INSA Rouen

March 2019

1 U-Net

1.1 Structure of U-Net

Layers : see table 2
Total parameters: 31,031,685
Data: 30 grey cell images for training; 30 grey cell images for test.

1.2 Hyper-parameters

Learning rate
Steps_per_epoch
Epoch
Batchsize
Number of layers
etc.

1.3 Loss functions

1.3.1 Binary Cross Entropy

$$binary\_cross\_entropy = -[y \cdot log(p) + (1 - y) \cdot log(1 - p)]$$

y is label value,p is probability of right sample.

1.3.2 Dice coefficient

$$dice\_coef = 2|A \cap B|/(|A| + |B|)$$

1.3.3 Jaccard index

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

1.3.4 Focal loss

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t)$$

1.3.5 Hausdorff distance

$$HD(A, B) = max(h(A, B), h(B, A))$$

1.4 Experiment analysis

1.4.1 If over fitting

In test part of U-Net, I used test data and training data as input to validate if this neural network is over fitting, the predicted results are good, so we can conclude the U-Net is not over fitting. But this method may not accurate.

1.4.2 Epoch

Table 1: Performance of different loss functions under the same learning rate

Loss function	Loss changing	Metric(Accurcy)
Binary Cross Entropy	0.2977	0.8635
Dice Coefficient	0.1413	0.1413
Jaccard Index	0.1103	0.7812
Focal Loss	5215.9615	0.8511

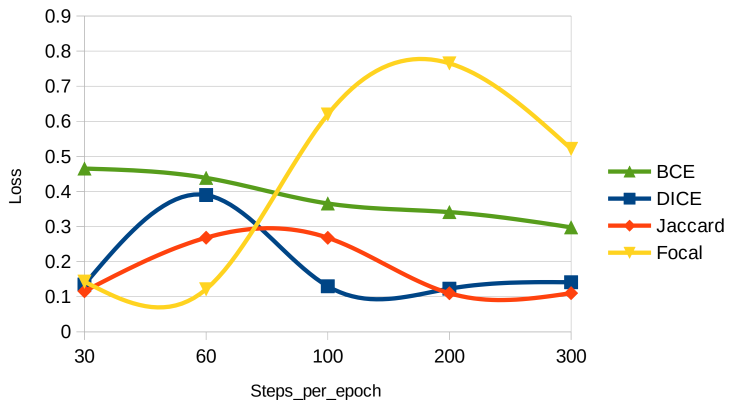


Figure 1: Loss changing under different step_per_epoch

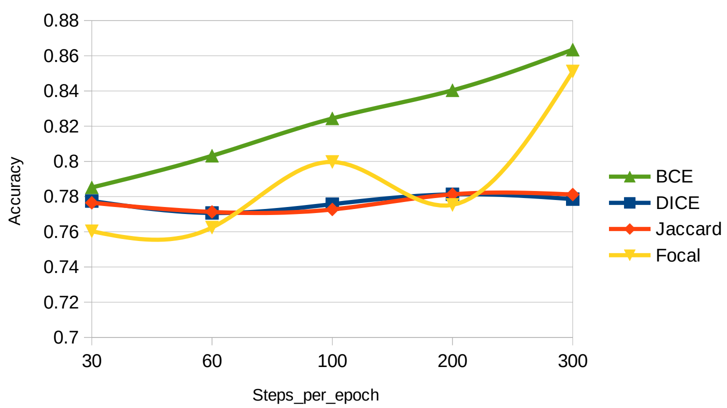


Figure 2: Accuracy under different step_per_epoch

1.4.3 Learning rate

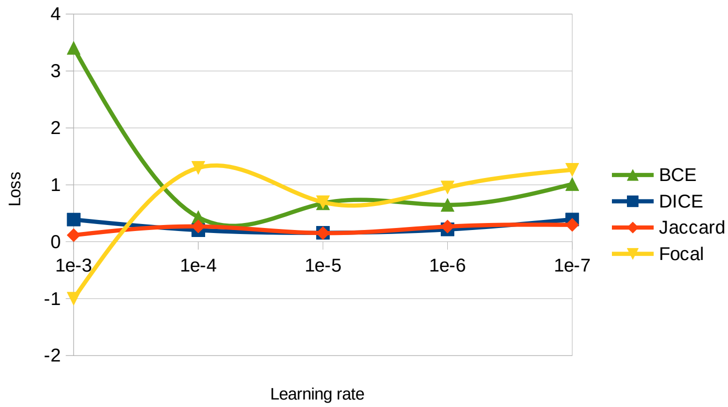


Figure 3: Loss changing under different learning rate

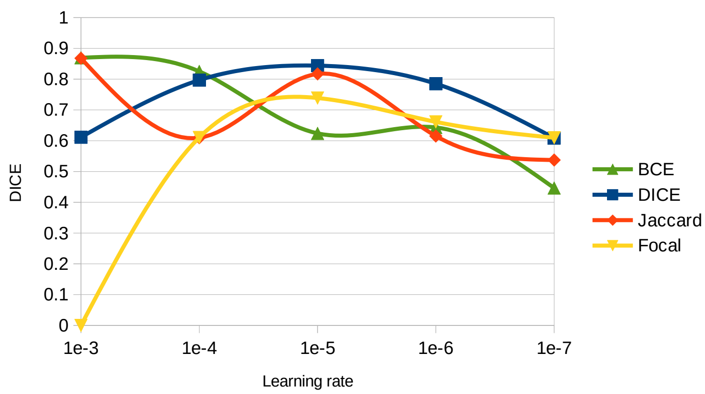
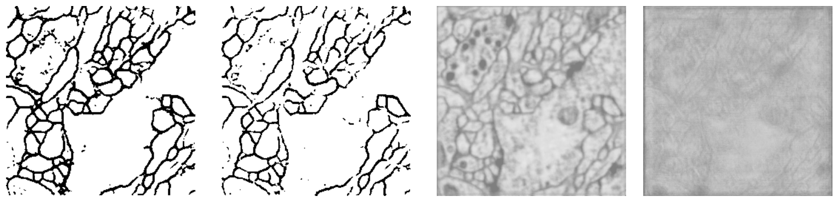


Figure 4: Metric of DICE under different learning rate

1.4.4 Strange phenomenon

Generally, with lower loss and higher accuracy, the predicted results are better. As figure 5 shown. However, sometimes the predicted results are bad. As figure 6 shown.Possible reasons are the learning rate is too fast or too slow, and the epoch value is too big or too small, which leads to the neural network not able to learn very well. As well as the selection of loss function is important, too.



(a) loss:0.2977 acc:0.8635 (b) loss:5215.96 - acc: 0.8511 (c) loss: 0.3163 - dicecoef: 0.6837 (d) loss: 0.2146 - acc: 0.7775

Figure 5: pics

(a) loss:0.1103 acc: 0.7812 (b) loss: 0.4652 - acc: 0.7852 (c) loss: 1.0126 - dice: 0.4464 (d) loss: 0.3163 - dice:0.6837

Figure 6: pics

1.5 Some good links

1.Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support
2.The Annotated Encoder Decoder.(Propose attention as loss function)
3.Boundary loss for highly unbalanced segmentation.(Propose surface-loss as loss function)
4.DeepCut(Interactive segmentation)
5.DeepGrabCut(Interactive segmentation)
6.DeepIGeoS(Interactive segmentation)

Table 2: summary of U-Net

Layer (type)	Output Shape	Param	Connected to
input1 (InputLayer)	(None, 256, 256, 1)	0	
conv2d1 (Conv2D)	(None, 256, 256, 64)	640	input1
conv2d2 (Conv2D)	(None, 256, 256, 64)	36928	conv2d1
maxpooling2d1	(None, 128, 128, 64)	0	conv2d2[0][0]
conv2d3 (Conv2D)	(None, 128, 128, 128)	73856	maxpooling2d1
conv2d4 (Conv2D)	(None, 128, 128, 128)	147584	conv2d3
maxpooling2d2	(None, 64, 64, 128)	0	conv2d4
conv2d5 (Conv2D)	(None, 64, 64, 256)	295168	maxpooling2d2
conv2d6 (Conv2D)	(None, 64, 64, 256)	590080	conv2d5
maxpooling2d3	(None, 32, 32, 256)	0	conv2d6
conv2d7 (Conv2D)	(None, 32, 32, 512)	1180160	maxpooling2d3
conv2d8 (Conv2D)	(None, 32, 32, 512)	2359808	conv2d7
dropout1	(None, 32, 32, 512)	0	conv2d8
maxpooling2d4	(None, 16, 16, 512)	0	dropout1
conv2d9 (Conv2D)	(None, 16, 16, 1024)	4719616	maxpooling2d4
conv2d10 (Conv2D)	(None, 16, 16, 1024)	9438208	conv2d9
dropout2 (Dropout)	(None, 16, 16, 1024)	0	conv2d10
upsampling2d1	(None, 32, 32, 1024)	0	dropout2
conv2d11 (Conv2D)	(None, 32, 32, 512)	2097664	upsampling2d1
concatenate1	(None, 32, 32, 1024)	0	dropout1 conv2d11
conv2d12 (Conv2D)	(None, 32, 32, 512)	4719104	concatenate1
conv2d13 (Conv2D)	(None, 32, 32, 512)	2359808	conv2d12
upsampling2d2	(None, 64, 64, 512)	0	conv2d13
conv2d14 (Conv2D)	(None, 64, 64, 256)	524544	upsampling2d2
concatenate2	(None, 64, 64, 512)	0	conv2d6 conv2d14
conv2d15 (Conv2D)	(None, 64, 64, 256)	1179904	concatenate2
conv2d16 (Conv2D)	(None, 64, 64, 256)	590080	conv2d15
upsampling2d3	(None, 128, 128, 256)	0	conv2d16
conv2d17 (Conv2D)	(None, 128, 128, 128)	131200	upsampling2d3
concatenate3	(None, 128, 128, 256)	0	conv2d4 conv2d17
conv2d18 (Conv2D)	(None, 128, 128, 128)	295040	concatenate3
conv2d19 (Conv2D)	(None, 128, 128, 128)	147584	conv2d18
upsampling2d4	(None, 256, 256, 128)	0	conv2d19
conv2d20 (Conv2D)	(None, 256, 256, 64)	32832	upsampling2d4
concatenate4	(None, 256, 256, 128)	0	conv2d2 conv2d20
conv2d21 (Conv2D)	(None, 256, 256, 64)	73792	concatenate4
conv2d22 (Conv2D)	(None, 256, 256, 64)	36928	conv2d21
conv2d23 (Conv2D)	(None, 256, 256, 2)	1154	conv2d22
conv2d24 (Conv2D)	(None, 256, 256, 1)	3	conv2d23

Table 3: Performance of different loss functions under the same epoch

Loss function	Loss changing	Metric(DICE)
Binary Cross Entropy	3.4045	0.8693
Dice Coefficient	0.2031	0.7969
Jaccard Index	0.2683	0.6097
Focal Loss	12987.4751	0.6105