

【深度学习】【U-net】医学图像(血管)分割实验记录

原创

StevenGerrad 于 2020-04-08 00:02:54 发布 1810 收藏 5

分类专栏: [实验记录](#) 文章标签: [人工智能](#) [计算机视觉](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_40690815/article/details/105377094

版权



[实验记录](#) 专栏收录该内容

10 篇文章 1 订阅

订阅专栏

医学图像分割实验记录

[U-net介绍](#)

[数据集](#)

[实验记录](#)

[实验1](#)

[实验2\(fail\)](#)

[实验3\(fail\)](#)

[实验4\(fail\)](#)

[实验5\(fail\)](#)

[实验6\(fail\)](#)

本项目仅用于大创实验, 使用pytorch编程, 参考价值有限

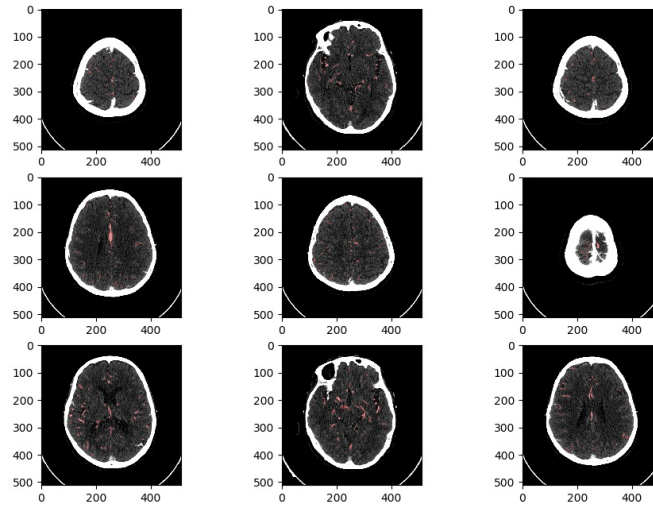
U-net介绍

这里先行挖个坑, 以后专门写一篇自己总结的资源吧。

数据集

使用hx医院提供的脑CT图像, 原图像为医学图像数据(数据范围较大)。使用 *ITK-SNAP* 软件自行标注后得到的标记数据做标签。

切片显示大致长这样:



TODO: 由于本人认为血管分割要参考每张图片上下几张空间序列上相邻切片的数据，因此想着直接把这些'相邻数据'作为每张图像的其他'channel'。且令总的channel为奇数。*eg.* 输入数据维度可能为

$(1,5,128,128)$ ，意味着 $batchsize=1$ ， $channel=5$ ， $imagesize=(128,128)$ 。

实验记录

实验1

Epoch: 15

train data: 800

test data: 200

Learning Rate: 0.05

Img Size: (128,128)

Batch Size: 1

Channel: 5

模型结构:

```

Unet(
  (conv1): DoubleConv(
    (conv): Sequential(
      (0): Conv2d(5, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (4): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
    )
  )
  (pool1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): DoubleConv(
    (conv): Sequential(
      (0): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (4): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
    )
  )
  (pool2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv3): DoubleConv(
    (conv): Sequential(
      (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (4): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
    )
  )
  (pool3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (up8): ConvTranspose2d(256, 128, kernel_size=(2, 2), stride=(2, 2))
  (conv8): DoubleConv(
    (conv): Sequential(
      (0): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (4): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
    )
  )
  (up9): ConvTranspose2d(128, 64, kernel_size=(2, 2), stride=(2, 2))
  (conv9): DoubleConv(
    (conv): Sequential(
      (0): Conv2d(128, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (4): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
    )
  )
  (conv10): Conv2d(64, 1, kernel_size=(1, 1), stride=(1, 1))
)

```

实验结果：忘记了orz

实验参数保存为 `net040801_normal.pkl`

实验2(fail)

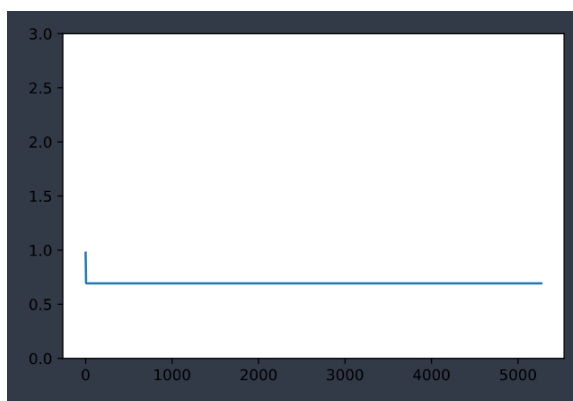
相比于实验1只改变了 `image size`。

Epoch: 15 train data: 800 test data: 200 Learning Rate: 0.05 lmg Size: (256, 256) Batch Size: 1 Channel: 5

模型结构同实验1

loss: 0.693154

误差维持不变，模型没有收敛。

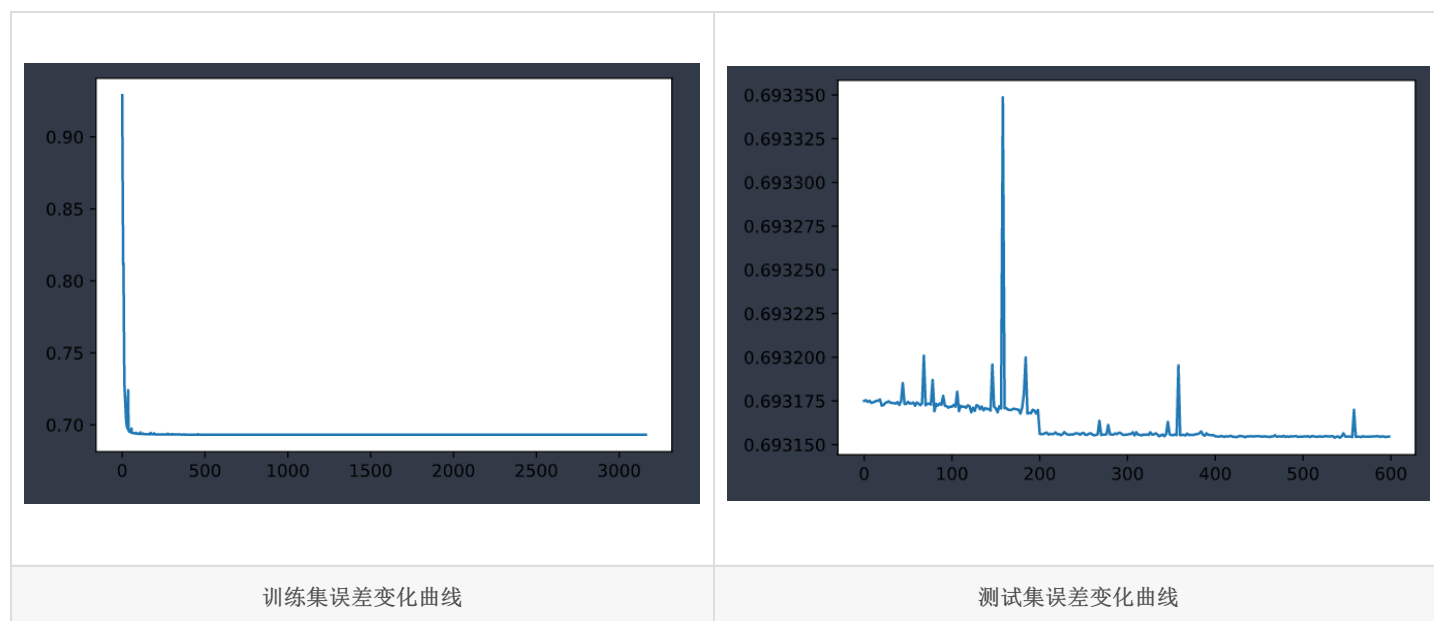


实验3(fail)

相比于实验2改变了 `learning rate`，从 `0.05` 变为 `0.005`。

Epoch: 15 train data: 800 test data: 200 Learning Rate: 0.005 lmg Size: (256, 256) Batch Size: 1 Channel: 5

模型结构同实验1

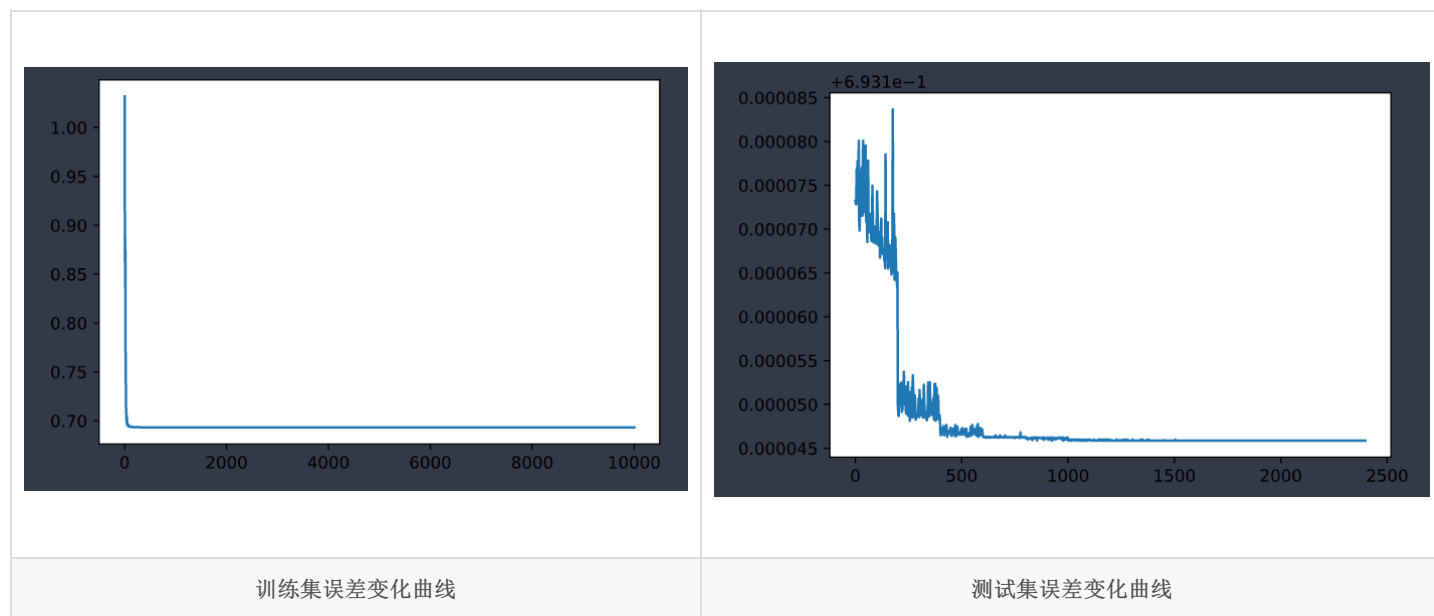


实验4(fail)

相比于实验3改变了 `image size`，从 `(256,256)` 改回 `(128,128)`。

Epoch: 15 train data: 800 test data: 200 Learning Rate: 0.005 Img Size: (128, 128) Batch Size: 1 Channel: 5

模型结构同实验1



最后误差维持在

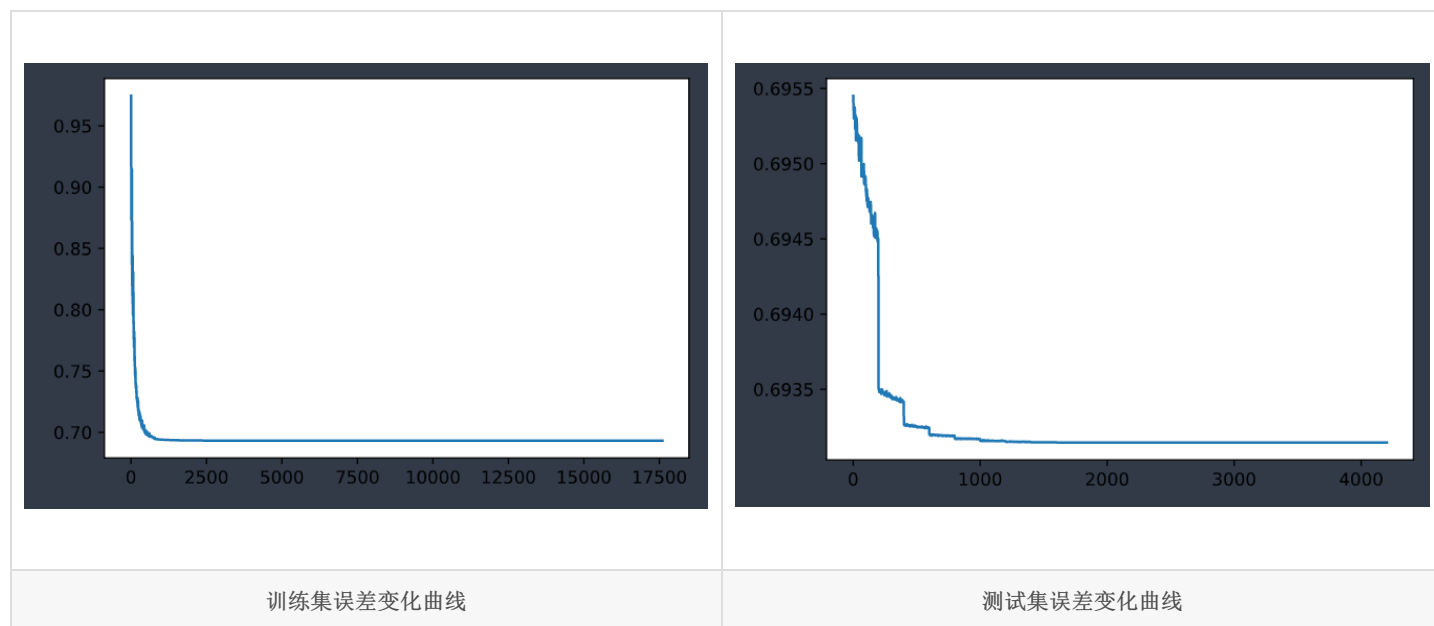
loss: 0.693146

实验5(fail)

相比于实验4改变了 `learning rate`，从 `0.005` 改为 `0.0005`。

Epoch: 20+ train data: 800 test data: 200 Learning Rate: 0.0005 Img Size: (128, 128) Batch Size: 1 Channel: 5

模型结构同实验1



最后误差维持在

loss: 0.693146

好吧这个数和实验4一样

实验6(fail)

相比于实验5改变了网络结构加了一层深度。

Epoch: 20+

train data: 800

test data: 200

Learning Rate: 0.0005

Img Size: (128, 128)

Batch Size: 1

Channel: 5

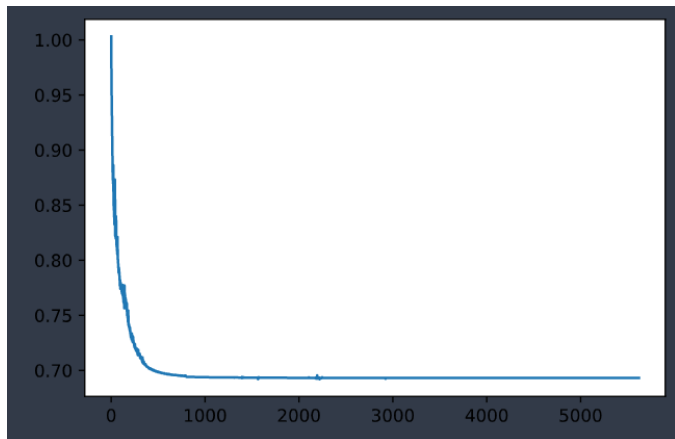
模型结构为

```
Unet(
  (conv1): DoubleConv(
    (conv): Sequential(
      (0): Conv2d(5, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (4): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
    )
  )
  (pool1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): DoubleConv(
    (conv): Sequential(
      (0): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (4): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
    )
  )
  (pool2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv3): DoubleConv(
    (conv): Sequential(
      (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (4): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
    )
  )
  (pool3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv4): DoubleConv(
    (conv): Sequential(
      (0): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (4): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
    )
  )
  (pool4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (up7): ConvTranspose2d(512, 256, kernel_size=(2, 2), stride=(2, 2))
  (conv7): DoubleConv(
    (conv): Sequential(
      (0): Conv2d(512, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
```

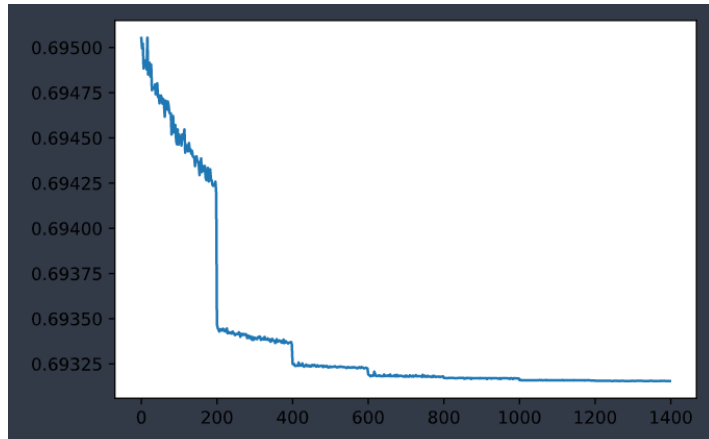
```

(1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(2): ReLU(inplace=True)
(3): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(4): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(5): ReLU(inplace=True)
)
)
(up8): ConvTranspose2d(256, 128, kernel_size=(2, 2), stride=(2, 2))
(conv8): DoubleConv(
(conv): Sequential(
(0): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(2): ReLU(inplace=True)
(3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(4): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(5): ReLU(inplace=True)
)
)
)
(up9): ConvTranspose2d(128, 64, kernel_size=(2, 2), stride=(2, 2))
(conv9): DoubleConv(
(conv): Sequential(
(0): Conv2d(128, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(2): ReLU(inplace=True)
(3): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(4): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(5): ReLU(inplace=True)
)
)
)
(conv10): Conv2d(64, 1, kernel_size=(1, 1), stride=(1, 1))
)
)

```



训练集误差变化曲线



测试集误差变化曲线

最后误差维持在

loss: 0.693156