[Question 1]:

I trained my u-net and have generated a mask_prediction.npy file. I am having trouble visualizing the file. I am trying this right now:

```
from PIL import Image

import numpy as np

import matplotlib.pyplot as plt

img_array = np.load("mask_prediction.npy")

plt.imshow(img_array)
```

The error is get is: invalid dimension for image data. Could you please help me out to view the predicted mask?

[Answer]:

I suggest you may consider to use the following codes, and I wishit would help you out.

```
imgs_mask_test = np.ndarray([num_test, 1, 512, 512], dtype=np.float32)
for i in range(num_test):
    imgs_mask_test[i] = model.predict([imgs_test[i:i + 1]], verbose=0)[0]
np.save(os.path.join(output_path + "preds/", 'masksTestPredicted.npy'), imgs_mask_test)
for i in range(num_test):
    np.save(os.path.join(output_path + "preds/", 'imgs_mask_test_%04d.npy' % (i)),
imgs_mask_test[i,0])
    cv2.imwrite(os.path.join(output_path + "ROI/preds/", 'imgs_mask_test_%04d.jpg' % (i)),
imgs_mask_test[i,0])
```

[Question 2]:

Thank you so much for your help. I just had a question regarding the mask_prediction.npy file. when I try to view the file, I only see black images with no nodule mask. Do you know what could be causing this and were you able to generate nodule mask for your experiment?

[Answer]:

I suggest that you can interate to generate the IMGs mask test%04d.npy files and convert into JPEG format, then enhance the brightness (* 1000.2) of the JPEG images using opencv, so this prediction of pulmonary nodule image will be displayed.

[Question 3]:

我对智能医疗比较感兴趣。看到您 github 项目里有很多关于这方面的项目。目前自己在做 肺癌检测这块的学习，看到您的项目代码了。想问些问题，给您添麻烦了。

我看到在 pulmonary_nodules_AI_diagnosis-master/mask_segment/masksTestPredicted.npy 这个项目下的这个文件，应该是训练好的测试输出文件是吧，但我 用 matplotlib.imshow() 显示还完全是黑色的，我查看你每个图片中的非零值，比较少，这样是不是显示不出来啊。

[Answer]:

Gihub 上的代码已经在公司服务器上用户天池数据集来训练模型和预测测试图像肺部结节坐标和疑似结节为真结节的概率。

在 pulmonary_nodules_AI_diagnosis-master/mask_segment/masksTestPredicted.npy 这个项目下的这个文件，是我本地训练好的测试输出文件，但是你在本地 checkout 出我的代码后，需要先训练你的模型，然后它会在你本地生成训练好的测试输出文件（即，覆盖我项目中的.npy 文件）。如果“显示还完全是黑色的，我查看你每个图片中的非零值，比较少”，你可以使用 opencv 中的图像亮度增强（比如：* 1000.2），然后测试图像中预测生成的肺部结节就可以显示了。

参考代码如下：

1）U-net 训练模型，生成测试数据集的预测肺部结节 mask 图像：

```
for i in range(num_test):
    imgs_mask_test[i] = model.predict([imgs_test[i:i + 1]], verbose=0)[0]
np.save(os.path.join(output_path + "preds/", 'masksTestPredicted.npy'), imgs_mask_test)
for i in range(num_test):
    np.save(os.path.join(output_path + "preds/", 'imgs_mask_test_%04d.npy' % (i)), imgs_mask_test[i,0])
    cv2.imwrite(os.path.join(output_path + "ROI/preds/", 'imgs_mask_test_%04d.jpg' % (i)), imgs_mask_test[i,0])
```

2）预测的肺部结节 mask 图像亮度调高：

```
for xi in xrange(0, w):
    for xj in xrange(0, h):
        ##set the pixel value increase to 1020%
```

```
        img[xj, xi, 0] = int(img[xj, xi, 0] * 1000.2)

        img[xj, xi, 1] = int(img[xj, xi, 1] * 1000.2)

        img[xj, xi, 2] = int(img[xj, xi, 2] * 1000.2)

# cv2.imshow('img',img)

cv2.imwrite(os.path.join(output_path + "image-coordinate-2D/bright_02/", 'imgs_mask_test_
%04d.jpg' % (index)), img)
```

[Question 4]:

1）我看到你对 nodule_mask 和 slice 都乘了 510，我不是很理解这部分，能帮我解释一下吗？

[Answer]: 因为图像是 uint8 图（值域 0~255），而 510 = 255 * 2。

2）我发现了几个问题

train_dataset_preprocessing_2DUnet.py 里面大概第 168 行左右，有代码如下：

```
nodule_mask[nodule_mask < 1.0] = 0
nodule_mask[nodule_mask > 1.0] = 1
```

这句话可能是有问题的，浮点数和整数转化的时候可能会有误差，生成的 lable 图片原本结节的位置很多 1 变成了 0，所以我把这个 nodule_mask < 1.0 改成了 nodule_mask < 0.9 了，还有这里面最后的图片都保存为了 512*512，但是之前没有做 resize，我自己加上去了。

[Answer]: 这个 1.0 是我在调试过程中修改时写上的，正确的代码如下：

```
nodule_mask = scipy.ndimage.interpolation.zoom(nodule_mask, [1.0, 1.0], mode='nearest')

nodule_mask[nodule_mask < 0.5] = 0

nodule_mask[nodule_mask > 0.5] = 1

nodule_mask = nodule_mask.astype('int8')
```

3）你文件中的代码，对输入数据用*_dataset_mask_extraction.py 和
*_dataset_segment_lung_ROI.py 做的时候是做了肺部分割的，用
*_dataset_preprocessing_2DUnet.py 做的时候是没有肺部分割的，请问你在做的时候两者的效

果差异大吗？因为我用*_dataset_preprocessing_2DUnet.py 做预处理然后训练得到的结果是全黑的，网络基本没有收敛，loss 一直在 0.0015 左右。

[Answer]:

1）天池 Train Set, Val Set 中 CT（.mhd）图像的预处理，使用项目 pulmonary-nodules-segmentation 中的代码如下：

*_dataset_preprocessing_2DUnet.py 或者， *_dataset_mask_extraction.py， *_dataset_segment_lung_ROI.py

2）对测试图像数据集的预测肺部结节 mask 图像（JPEG），使用 openCV 进行图像的亮度调高，代码如下：

adjust_preds_image_brightness.py

然后，使用 openCV 提取测试图像数据集肺部结节 mask 图像中疑似结节的坐标，代码如下：

imgs_mask_test_coordinate.py

附件中是我本地训练 U-net 深度卷积神经网络的脚本，可以参考。

[Question 5]:

1）还有些问题，对于每个像素点输出的值都太小了，所以图片显示是黑屏。因为还是训练数据的处理这部分有些问题，现在正在检查看是哪里的问题。

[Answer]: 项目 pulmonary-nodules-segmentation 中训练图像数据集预处理部分我已经提交了最新的代码。

代码如下：

*_dataset_preprocessing_2DUnet.py 或者，

*_dataset_mask_extraction.py， *_dataset_segment_lung_ROI.py

2）预测的肺部结节 mask 图像亮度调高：

[Answer]: 对测试图像数据集的预测肺部结节 mask 图像（JPEG），使用 openCV 进行图像的亮度调高，代码如下：

adjust_preds_image_brightness.py

[Question 6]:

Instead of training the network on the Segmented lungs, have you tried just extracting the patches of the nodules??

And also, i am a little confused about the LUNA 16 dataset. What is the difference between the annotations and candidates CSV files?? Doesn't both contain the nodule information??.

[Answer]:

1) Instead of training the network on the Segmented lungs, have you tried just extracting the patches of the nodules?

[Answer]: I preprocess the lung CT images, regional and interest of lung nodules in mask images, and then input to the U-net convolutional neural network training, finally using the trained model to predict the test image data.

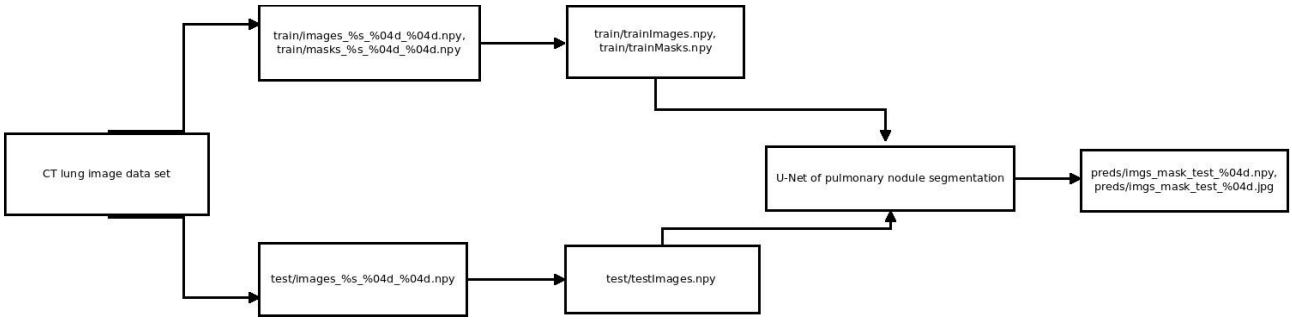


Fig. Extracting the mask of the nodules andtraining the network on the Segmented lungs

2) And also, i am a little confused about the LUNA 16 dataset. What is the difference between the annotations and candidates CSV files? Doesn't both contain the nodule information?

[Answer]: My project code just uses the annotations.csv file. Yet, the use of candidates.csv files is similar to that of annotations.csv file.

[Question 7]:

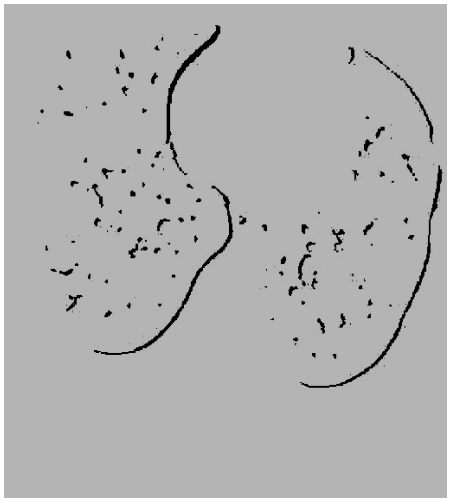
1）我运行了代码后，unet 网络一直不能很好的收敛，我分别调试了不同的 batch 大小，以及不同的学习率，虽然 loss 的收敛情况不同了，但是网络基本跑完第一个 epoch 后 loss 就不再下降，没有低于-0.2 的。你上次提供的运行脚本中，loss 最低能收敛到-0.9。

所以我现在不能确定导致网络不收敛的原因是我数据预处理的问题，还是网络参数没有调好的原因，能帮忙解答下吗？

[Answer]: U-Net 深度卷积神经网络的模型训练，这个与你服务器安装的 Tensorflow、keras 有关系（比如：你是 GPU，还是 CPU 安装？），另外这个问题与还与服务器的硬件配置和网络有关。建议你多做几次 training，然后看看训练结果是否有改进。

2）对了，不知道你能不能提供一张经过数据增亮后的图片，我想看看正常 unet 训练出来的结果是怎么样的。

[Answer]: 一张经过数据增亮后的图片，如下图所示：



[Question 8]:

Keras (2.0.2) tensorflow (1.1.0) ，模型的结构和训练参数按照您的代码里给的写的（因为 keras 版本不同，我跟新了模型函数名，但模型结构都是一样的）

1）附件中图 1：是 我在 cpu 服务器上，训练的最新结果，感觉模型 loss 很不正常啊，也没有这方面的经验，想请您帮忙看看，目前的训练是否是正常的，或者需要什么改进吗？

[Answer]: 附件中是我的 U-Net 训练脚本，请参考。

2）附件中图 2 ：是最新的 unet 预测后的图片，经过 opencv 增强后的效果图，还是看不出来什么效果。

[Answer]: 你可使用 imgs_mask_test_coordinate.py 提取 U-Net 预测生成图像中的肺结节坐标。

3）在您的*_dataset_preprocessing_2DUnet.py 程序 中：

```
slice = 510.0 * self.normalize(slice)
slice = slice.astype(np.uint8) # ---因为 int16 有点大，我们改成了 uint8 图（值域 0~255）
nodule_mask = 510.0 * nodule_mask
nodule_mask = nodule_mask.astype(np.uint8)
```

乘 510 是有什么作用吗，没有搞清楚？mask 本来都是 0 和 1 的，乘 510 的话，unet 训练的 label 就是 0 和 1 了吧？

[Answer]: 请将代码修改如下，这个“510.0”是我在代码调试过程中修改的，应该是 255.0（对于灰度图像，uint8 表示范围[0，255]，double 型表示范围[0，1]）

```
slice = 255.0 * self.normalize(slice)
slice = slice.astype(np.uint8)  # ---因为 int16 有点大，我们改成了 uint8 图（值域 0~255）
nodule_mask = 255.0 * nodule_mask
nodule_mask = nodule_mask.astype(np.uint8)
```

[Question 9]:

1）我分割好的 lable 图片，就是那些白点图，它们的值域都是 0~255 的，但是网络的输出经过 sigmoid 之后都是 0~1 之间的，我看代码中只对输入的原图做了标准化，并没有对 lable 图做标准化什么的，我不是很理解。这样的话标签和网络输出对不上了。

[Answer]: 请将代码修改如下，这个“510.0”是我在代码调试过程中修改的，应该是 255.0（为了节省存储空间，MATLAB 为图像提供了特殊的数据类型 uint8（8 位无符号整数），以此方式存储的图像称为 8 位型像；对于灰度图像，uint8 表示范围[0，255]，double 型表示范围[0，1]）

你可以使用 imgs_mask_test_coordinate.py 提取 U-Net 预测生成图像中的肺结节坐标。

```
slice = 255.0 * self.normalize(slice)
slice = slice.astype(np.uint8)  # ---因为 int16 有点大，我们改成了 uint8 图（值域 0~255）
nodule_mask = 255.0 * nodule_mask
nodule_mask = nodule_mask.astype(np.uint8)
```

2）我用 batchsize3 训练了网络大概 80 个 epoch，效果还是没有你之前发我那张图好（附件我

贴了张看着比较好的 predict），而且很奇怪的是网络的 loss 能逼近-2，我觉得-dicecof 最小化后应该是-1。我不知道是不是我哪里理解错了，还是哪里代码让我改错了。

[Answer]: 我用 Luna16 数据集训练 U-Net 模型，dice_coef 都是大于 0 的数值。（只是数值小于 1）

例如：

```
Epoch 15/20

184/184 [==============================] - 6541s - loss: -0.3221 - dice_coef: 0.3221 - val_loss: -8.3698e-06 - val_dice_coef: 8.3698e-06

Epoch 16/20

184/184 [==============================] - 6638s - loss: -6.2088e-06 - dice_coef: 6.2088e-06 - val_loss: -8.3698e-06 - val_dice_coef: 8.3698e-06

Epoch 17/20

184/184 [==============================] - 6558s - loss: -6.2665e-06 - dice_coef: 6.2665e-06 - val_loss: -8.3698e-06 - val_dice_coef: 8.3698e-06

Epoch 18/20

184/184 [==============================] - 6554s - loss: -6.1257e-06 - dice_coef: 6.1257e-06 - val_loss: -8.3698e-06 - val_dice_coef: 8.3698e-06

Epoch 19/20

184/184 [==============================] - 6545s - loss: -6.0250e-06 - dice_coef: 6.0250e-06 - val_loss: -8.3698e-06 - val_dice_coef: 8.3698e-06

Epoch 20/20

184/184 [==============================] - 6539s - loss: -6.2926e-06 - dice_coef: 6.2926e-06 - val_loss: -8.3698e-06 - val_dice_coef: 8.3698e-06
```

3）还有一个问题，我们在测试数据上，因为没有标注信息，那是不是得把所以的切片都测试一遍呢？都做一遍的话，请问下你对于相邻切片如何判断结节是否为同一个呢？

[Answer]: 我使用 U-Net 训练的模型预测测试图像的肺结节中心坐标，coordX,coordY, coordZ,diameter_mm，而这个预测的肺结节中心坐标坐标就是生成的测试图像标注信息。然后，程序使用预测坐标对测试图像进行切片。

[Question 10]:

1）关于 510 这个问题，我之前已经解决了，现在不太理解的是网络的输入范围，如下代码：

```
imgs_train = np.load(os.path.join(output_path, "train/trainImages.npy")).astype(np.float32)
imgs_mask_train = np.load(os.path.join(output_path, "train/trainMasks.npy")).astype(np.float32)
print(np.max(imgs_train))

print(np.max(imgs_mask_train))
```

这里对 imgs_train 和 imgs_mask_train 都转化成了 float32,但是转化后的矩阵值域是没有变的，还是 0~255，print 最大值后的输出如下：

255.0

255.0

如果直接把这个 imgs_mask_train 作为标签的话，值域是在 0~255 的和 unet 的输出（0~1）是对不上的。

我也尝试着做了 imgs_mask_train /= 255;但是效果如下，感觉网络并没有收敛，前两个 epoch 还正常，后面就出问题了，如下

```
776/776 [==============================] - 364s - loss: -0.0134 - dice_coef: 0.0134 - val_loss: -0.0404 - val_dice_coef: 0.0404

Epoch 2/20

776/776 [==============================] - 357s - loss: -0.0343 - dice_coef: 0.0343 - val_loss: -6.8613e-04 - val_dice_coef: 6.8613e-04

Epoch 3/20

776/776 [==============================] - 357s - loss: -7.0536e-04 - dice_coef: 7.0536e-04 - val_loss: -7.5453e-04 - val_dice_coef: 7.5453e-04

Epoch 4/20

776/776 [==============================] - 357s - loss: -7.0365e-04 - dice_coef: 7.0365e-04 - val_loss: -7.6580e-04 - val_dice_coef: 7.6580e-04

Epoch 5/20

776/776 [==============================] - 357s - loss: -7.4759e-04 - dice_coef: 7.4759e-04 - val_loss: -7.6985e-04 - val_dice_coef: 7.6985e-04
```

[Answer]: MATLAB 为图像提供了特殊的数据类型 uint8（8 位无符号整数），以此方式存储

的图像称为 8 位型像；对于灰度图像，uint8 表示范围[0，255]，double 型表示范围[0，1]）

```
slice = scipy.ndimage.interpolation.zoom(slice, [1.0, 1.0], mode='nearest')

slice = 255.0 * self.normalize(slice)

slice = slice.astype(np.uint8) # ---uint8 图（值域 0~255）

nodule_mask = 255.0 * nodule_mask

nodule_mask = nodule_mask.astype(np.uint8)
```

2）这里你提供的这个表中和上面我遇到的问题一样了，前面迭代的 loss 是正常的，但是后面就会突然变得很低，我一直不知道原因，之前因为这个原因，我一直认为我预处理出错了，所以一直在调试预处理源码，但是我发现你提供的这个输出也存在这个问题，不知道能不能帮忙解释下这个现象。而且出现这个情况后，loss 基本不会再变回之前比较正常的情况。

[Answer]: U-Net 深度训练部分的代码，我没有本地调参，只是直接进行训练，然后生成模型，这样的话，预测的 accuracy 可能不会太高，但是，整个流程都可以运行。

3）这个问题，我可能没有说清楚，因为我现在只用了 train data 和 val data，这些都是有标注结节的，直接把那些结节在结节位置切出来就 ok 了，但是 test data 中没有结节信息，我们要去预测的话，~~就得把所以的~~ ~~CT 图中的每个 slice 都放进 unet 做下预测，~~并把所有的 predict 都提出来，~~但是相邻的 slice 之间肯定有重复的结节预测出来了，~~我现在不知道该具体怎么~~去把这些重复的结节去掉~~，所以想请教下你，因为不分析下重复的情况，假阳性应该会很多。

[Answer]:

首先，天池的测试图像数据集的 annotations.csv 文件是需要程序预测生成的。

其次，将预处理后的测试图像数据集输入到 U-Net 深度模型中，输出的是疑似结节的预测坐标。你可以看到一个 CT 图像预测出来的疑似结节有很多个（其实有很多都不是疑似结节，可能是血管或其他的），所以我使用 imgs_mask_test_coordinate.py 提取肺部疑似结节坐标，并且 test_dataset_extract3d_lung.py 对测试 CT 图像进行立方体切割，从而提取 CT 图像中每个立方体的坐标，然后 test_dataset_extract_nodules_coord.py 对同一个 CT 图像将这两份坐标进行匹配，这样肺部预测结节坐标中的 x, y, z 值就获取了（一个 CT 图像只可能保留几个疑似结节）。

然后，你使用上述的测试图像的预测结节中心坐标，对测试图像进行预处理，这样就可以生

成测试 CT 图像的结节切割图像。

[Question 11]:

如果直接把这个 imgs_mask_train 作为标签的话，值域是在 0~255 的和 unet 的输出（0~1）是对不上的。

[Answer]: 这个问题在《OpenCV: Computer Vision Projects with Python》中有相关的解释。

image.dtype = np.uint8 just forcibly casts the bytes from float64 to **uint8**. Since each float64 takes 8 bytes, and **each uint8 is only 1 byte**, you're getting 8 times as many values.

To convert the values, instead of reinterpreting the bytes, you want the astype method:

image = image.astype(np.uint8)

However, that probably isn't going to be very useful, for two reasons.

First, the big one, your float values are probably all in the range 0.0-1.0. Second, astype truncates, rather than rounding.

So, converting to integers is just going to make almost all of them 0, and the rest 1, rather than smoothly ranging from 0-255.

So, what you probably want is something like:

image = (image * 255).round().astype(np.uint8)