

[< Back to Deep Learning Nanodegree](#)

Generate Faces

[审阅](#)[代码审阅](#)[HISTORY](#)

Requires Changes

还需满足 2 个要求 变化

我注意到你没有按照之前的review给的意见进行修改，是不是上传错了版本？我用中文给你重申了一边需要修改的要点。请对照修改哦。还有一些之前reviewer没给你的一些优化方向也在下面，你可以试试。

项目中是否有需要的文件、相关函数是否通过了单元测试

该项目提交应包含项目中名为“dln_d_face_generation.ipynb”的.ipynb文件。

项目中所有的单元测试都已通过。

建立神经网络

`model_inputs` 函数被正确实现。

`discriminator` 函数被正确实现。

很好，你正确的实现了卷积神经网络，batch normalization用的也很棒，这样的话卷积神经网络的效率会有所提高。

建议：

- 如果要进一步的提升模型的效率，你可以使用0.8概率的dropout和Xavier初始化你的权重。
- Xavier可以通过下面的方法实现，在tf.layers.conv2d中，将tf.contrib.layers.xavier_initializer()传入作为kernel_initializerparameter的值。
- Dropout可以通过在每一层的Leaky Relu后面加上加上dropout层（tf.nn.dropout）就好啦

`generator` 函数被正确实现。

`model_loss` 函数被正确实现。

`model_opt` 函数被正确实现。

神经网络训练

训练函数被正确实现。

- 它应使用 `model_inputs`、`model_loss` 及 `model_opt` 创建了模型。
- 同时它使用 `show_generator_output` 函数展示了 `generator` 的输出。

你的train函数实现的不错，不过注意到了，我们需要将 `real images` 进行标准化，通过下面的代码可以完成。`batch_images = batch_images * 2`

这些超参数被设置到了合理的数值。

项目代码生成的面部图像比较逼真，至少看上去像是一些人脸。

其实已经很好了，主要有几个地方需要注意一下，请看上面的意见。

[↓ 下载项目](#)



重新提交项目的最佳做法

Ben 与你分享修改和重新提交的 5 个有益的小贴士。

[🎥 观看视频 \(3:01\)](#)

[返回 PATH](#)

[学员 FAQ](#)