

一、生成对抗网络生成 MNIST 手写数字（结合代码描述实现步骤以及提交下面要求提交的结果）

输入图片维度：784 (28×28)

输入噪声维度：100

使用以下函数初始化参数

```
def glorot_init(shape):  
    return tf.random_normal(shape=shape, stddev=1. / tf.sqrt(shape[0] / 2.))
```

1、模型架构（生成器和判别器都使用全连接网络）：

生成器：

输入维度：100

隐藏层维度（ReLU）：256

输出层维度（Sigmoid）：784

判别器：

输入维度：784

隐藏层维度（ReLU）：256

输出层维度（Sigmoid）：1

训练完网络，从噪声中生成 10 张手写数字图片（提交最终的生成器损失和判别器损失以及生成的图片）

2、模型架构（生成器使用全连接网络，判别器使用卷积神经网络）

生成器：

输入维度：100

隐藏层维度（ReLU）：256

输出层维度（Sigmoid）：784

判别器：

输入维度： 28×28

卷积层：卷积核大小为 5×5 ，卷积核个数为 16（padding 使用 ‘SAME’，stride 为 1），激活函数为 ReLU。

池化层：使用最大池化，核大小为 2×2 ，stride 为 2。

全连接层：隐含单元数为 256，激活函数为 ReLU。

输出层维度：1，激活函数为 Sigmoid。

训练完网络，从噪声中生成 10 张手写数字图片（提交最终的生成器损失和判别器损失以及生成的图片）