一、变分自编码器生成 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）

推荐使用以下函数初始化参数，可以避免一部分模式坍塌问题。

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

1、模型架构：

① 编码器（全连接层）：

输入图片维度：784 (28 × 28)

隐藏层维度（ReLU）：256

输出层维度（Tanh）：512

② 生成均值（全连接层）：

输入层维度：512

输出层维度：2

③ 生成标准差（全连接层）：

输入层维度：512

输出层维度：2

④ 使用均值和标准差生成隐变量z

⑤ 解码器（全连接层）：

输入维度：2

隐藏层维度（ReLU）：512

输出层维度（Sigmoid）：784

训练完网络，需要提交重构损失和KL散度的随迭代次数的变化图，以及10 张生成的手写数字图片。

二、Encoder-Decoder架构用于看图说话(Image Captioning)任务（结合代码描述实现步骤以及 提交下面要求提交的结果）

1. 看图说话的目的是将给定的输入图像转换为自然语言描述。编码器-解码器框架被广泛用于这项任务。一般来说，图像编码器是一个卷积神经网络(CNN)。解码器是一个长短期记忆(LSTM)网络。编码器起特征提取的作用，将图像转化成特征向量，解码器本质上是个语言模型，通过对特征向量解码，解码出特征向量最可能对应的自然语言描述。

2. 数据集：ILSVRC2012：

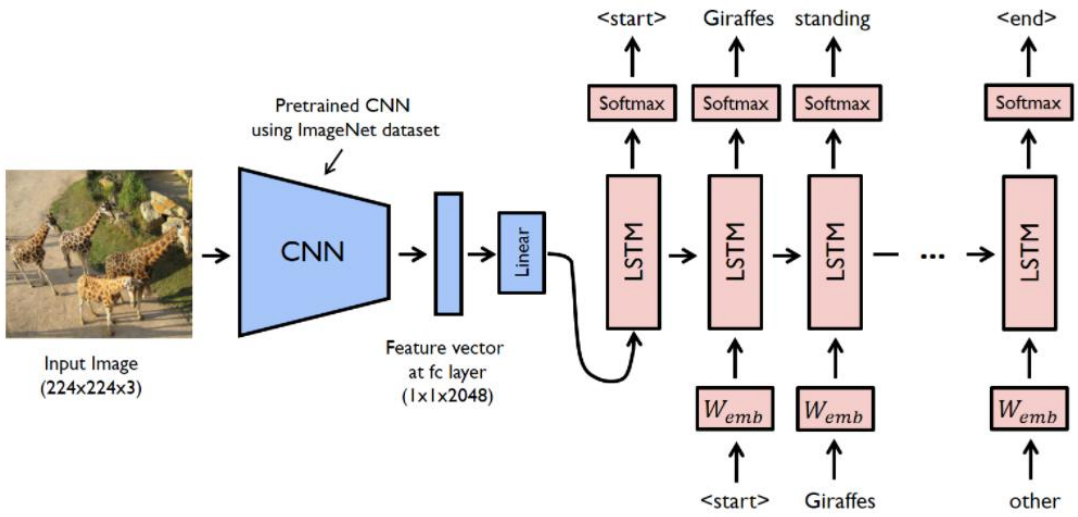
3. 数据预处理说明：

后续会提供一份预处理代码供参考，主要为数据集下载、建立词表、图片resize等基础代码。数据预处理方式不做硬性要求。

4. 模型说明：

Encoder：众多多层CNN模型都可。参考：Resnet-152，实现可以用tensorflow或pytorch的高级接口，无需自己实现。Resnet后接一个全连接层，参考输出维度为256，因此Encoder的输出是一个长度为256的特征向量。

Decoder：采用LSTM即可，实现可以用tensorflow或pytorch的高级接口，无需自己实现。需要注意的是，由于解码器是个语言模型，需要使用到word2vec等词向量作为输入，后续会提供参考资料。LSTM的参数参考：隐藏维度为512，LSTM层数为1。LSTM的隐藏状态通过softmax计算出当前时间步可能的word。结构图如下：



损失函数采用交叉熵损失，优化器可选择Adam。

5. 提交说明：

提交训练集和测试集的损失随迭代次数的变化图，训练结束后挑选2个测试集中的例子进行预测与展示。

6. 加分项：数据集分析与更好的预处理方式，不同的实验设置如不同的优化器选择等等。考虑到实验可能比较难，后续会提供一份基于pytorch的实现版本供参考。用tensorflow实现会酌情加分。