



SeetaTech
中 科 视 拓

InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets 复现指南

主讲人：冯少雄

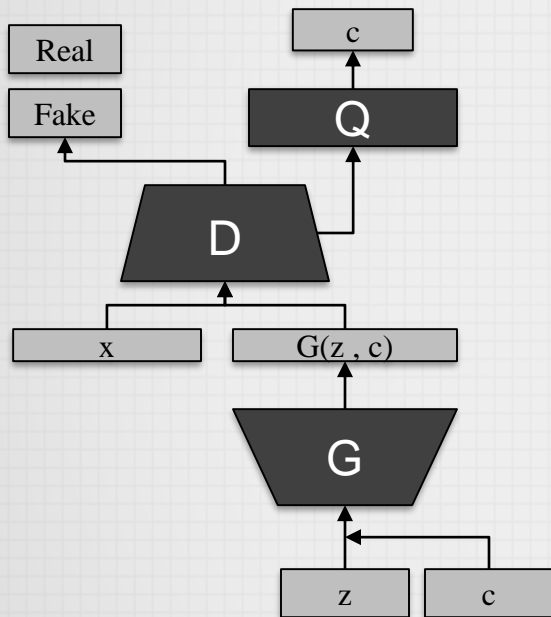
目录

- 一、GANs复现框架
- 二、基于InfoGAN网络结构与损失函数构建代码框架
- 三、代码实现
- 四、实验结果分析
- 五、模型训练注意事项
- 六、课后作业

一、GANs复现框架

```
├─ data
│   └─ mnist
│       ├── t10k-images-idx3-ubyte.gz
│       ├── t10k-labels-idx1-ubyte.gz
│       ├── train-images-idx3-ubyte.gz
│       └── train-labels-idx1-ubyte.gz
│   └─ fashion-mnist
│       ├── t10k-images-idx3-ubyte.gz
│       ├── t10k-labels-idx1-ubyte.gz
│       ├── train-images-idx3-ubyte.gz
│       └── train-labels-idx1-ubyte.gz
├─ main.py # 程序入口：参数读取启动；模型构建、训练、测试、可视化的启动
├─ infoGAN.py # 静态计算图构建：模型构建、训练、测试、可视化
├─ ops.py # 卷积、反卷积、激活函数.....
├─ utils.py # 数据加载、预处理、存储、可视化.....
├─ logs # 训练过程中指定的“参数、损失值、图片.....”的存储文件
└─ checkpoint # 模型文件
```

二、基于InfoGAN网络结构与损失函数构建代码框架



Original GAN objective function:

$$\begin{aligned} \min_G \max_D V(D, G) \\ = E_{x \sim P_{data}} [\log D(x)] + E_{z \sim noise} [\log(1 - D(G(z)))] \end{aligned}$$

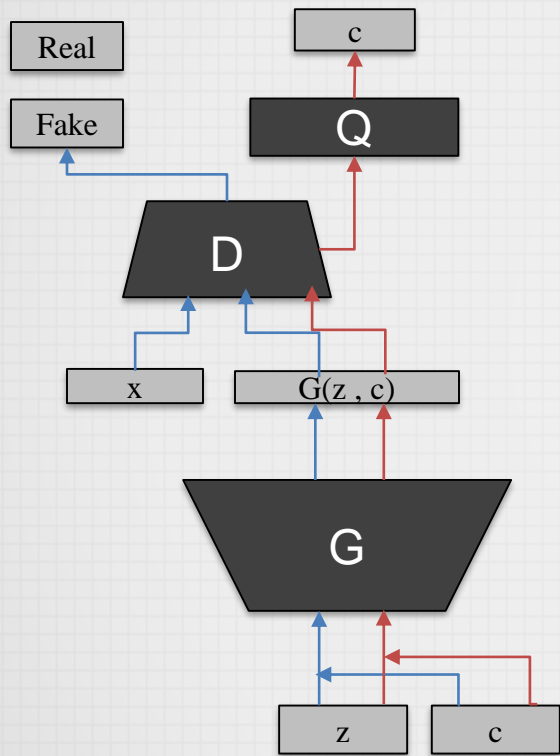
A variational lower bound of the mutual information:

$$\begin{aligned} L_I(G, Q) &= E_{c \sim P(c), x \sim G(z, c)} [\log Q(c|x)] + H(c) \\ &= E_{x \sim G(z, c)} [E_{c' \sim P(c|x)} [\log Q(c'|x)]] + H(c) \\ &\leq I(c; G(z, c)) \end{aligned}$$

Final objective function:

$$\min_{G, Q} \max_D V_{InfoGAN}(D, G, Q) = V(D, G) - \lambda L_I(G, Q)$$

二、基于InfoGAN网络结构与损失函数构建代码框架



```
class infoGAN
├─ 构建模型静态计算图
│   ├── def __init__
│   ├── def generator
│   ├── def discriminator
│   ├── def classifier
│   └─ def build_model
│       ├── Graph Input
│       ├── Loss Function
│       ├── Training
│       ├── Testing
│       └─ summary
├─ 构建模型优化、测试方案
│   ├── def train
│   └─ def visualize_results
└─ 构建模型加载和储存方案
    ├── def model_dir
    ├── def save
    └─ def load
```

三、代码细节

- 1. 环境配置
 - Ubuntu14.04
 - CUDA8.0、cuDNN6.0
 - Python3
 - Tensorflow1.4 、Numpy1.14
 - Pillow5.1、Scipy1.0、matplotlib2.2、python3-tk(使用apt下载)
- 2. 数据集准备：MNIST、fashion-MNIST
- 3. 实现目标：探索隐变量(z 、 c)表示的语义信息

四、实验结果分析

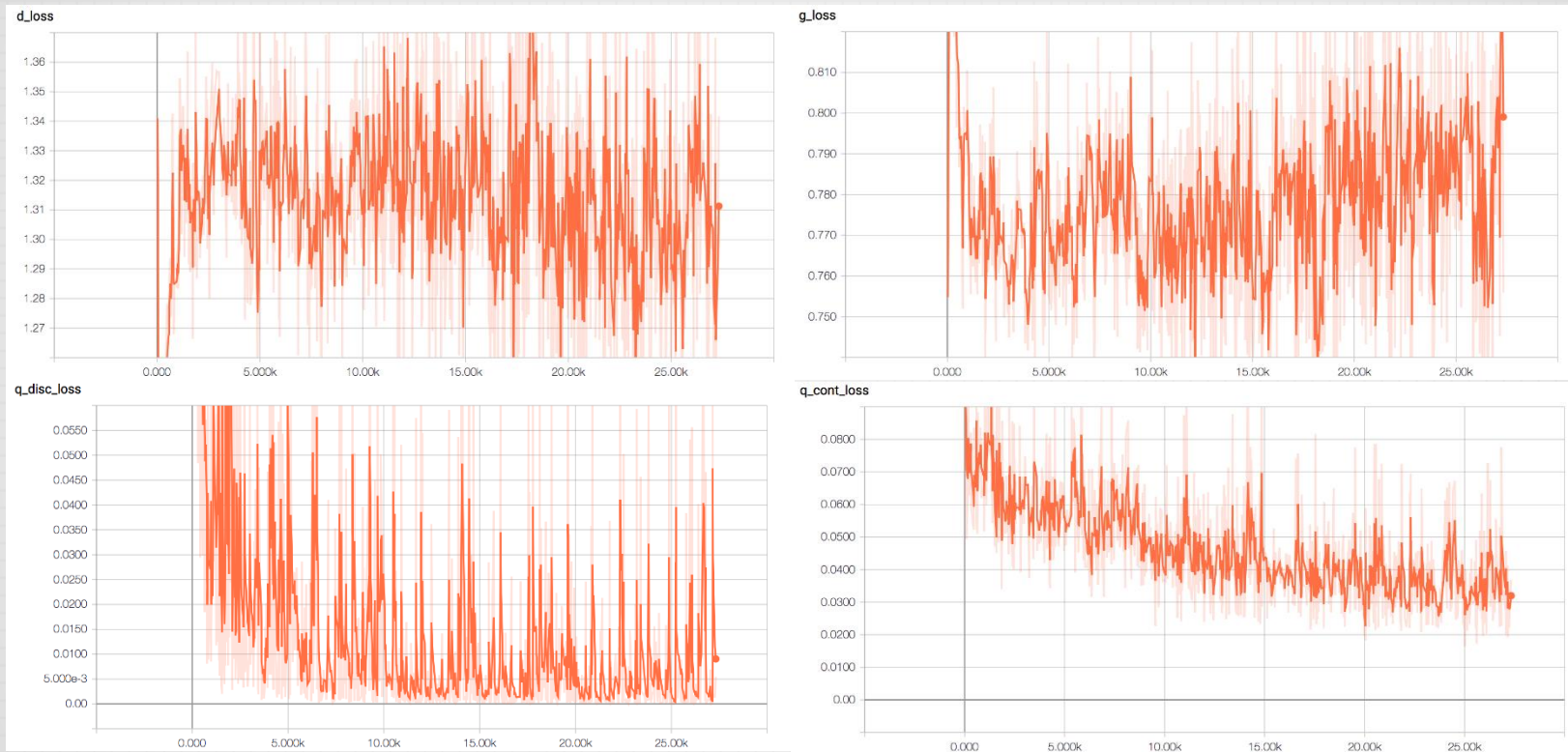
定量

D、G、Q的损失函数
值曲线变化

定性

基于人来判断生成图
片的真实性、隐变量
的可解释性

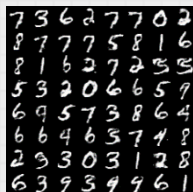
损失函数值的变化曲线



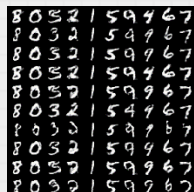
生成图片展示-MNIST

random noise
random discrete code
fixed continuous code

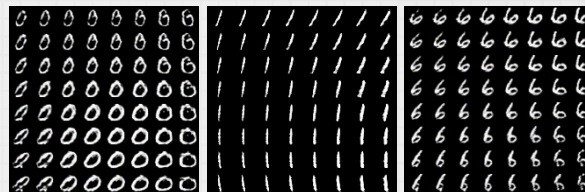
epoch8



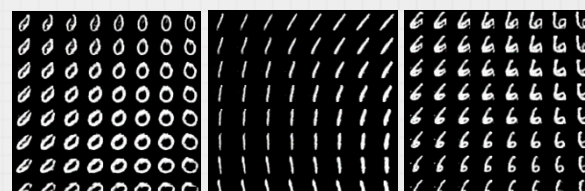
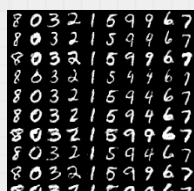
random noise
specified discrete code
fixed continuous code



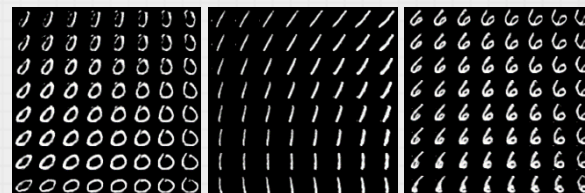
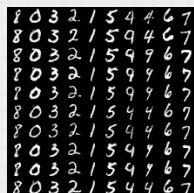
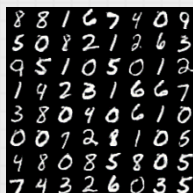
fixed noise
specified discrete code
gradual change continuous code



epoch16



epoch24



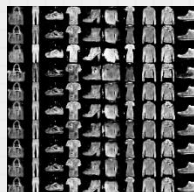
生成图片展示-fashion MNIST

random noise
random discrete code
fixed continuous code

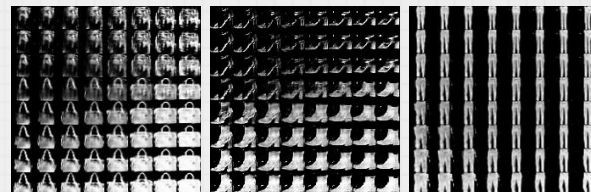
epoch13



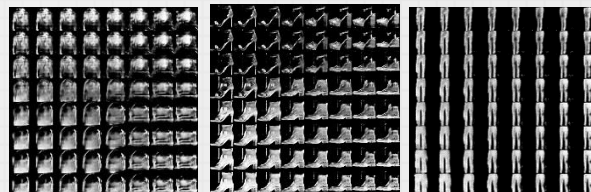
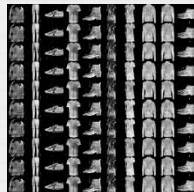
random noise
specified discrete code
fixed continuous code



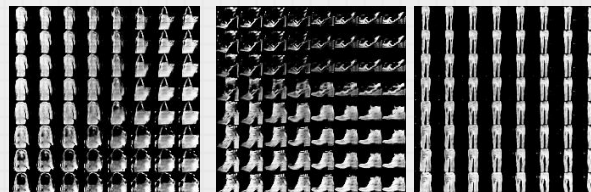
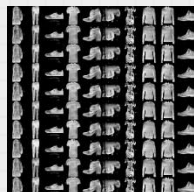
fixed noise
specified discrete code
gradual change continuous code



epoch26



epoch39



五、模型训练注意事项

Batch normalization的使用

1. 在InfoGAN的实现中D的第一层、G的最后一层分别通过卷积、反卷积后不进行batch normalization, 因为它会造成样本震荡和模型的不稳定。
2. Batch normalization计算公式: $y = \frac{\gamma(x-\mu)}{\sigma} + \beta$, y 是输出, μ 是均值, σ 是方差, γ 和 β 是缩放、偏移系数。 μ 和 σ 在训练的时候, 使用的是一个batch数据的统计值, 但是测试的时候, 采用的是训练时计算出的滑动平均值。
3. 在训练的时候需要注意: training设置为True, 且要添加以下代码, 进行 μ 和 σ 的滑动平均值的更新操作:

```
update_ops = tf.get_collection(tf.GraphKeys.UPDATE_OPS)
with tf.control_dependencies(update_ops):
    train_op = optimizer.minimize(loss)
```

4. 在测试的时候需要注意: training设置为False

五、模型训练注意事项

Summary的使用

1. 可根据损失函数值的变化曲线是否震荡来调节学习率、动量因子
2. 可以利用直方图分析模型中各个模块参数的分布状况

六、课后作业

1. 公式推导：通过变分推断求解互信息 $I(c; G(z, c))$ 的下界
2. 代码编写：目前 c 由一个10维的one-hot向量和两个基于高斯分布的连续变量组成，尝试给 c 再增加一个基于高斯分布的连续变量，探索该变量控制的语义信息

Thank you.