VAE 实践报告

519030910383 张哲昊

1. 摘要

本次实践项目实现了 VAE 的 enocder 和 decoder，完成了在 MNIST 上的训练并生成了质量较高的图片。同时探究了隐层向量 z 的维度对于生成图像的关系。同时实现了最优化验证集上的重构误差。同时通过查阅相关文献，理解了 VAE 的数学原理。

2. VAE 的数学原理[1]

2.1. 概率模型

2.1 概率模型

VAE 的本质是一个含隐变量的生成模型，而其中的隐变量相对图像而言维度较低。其结构如图所示 生成模型的联合概率密度函数可以分解为:

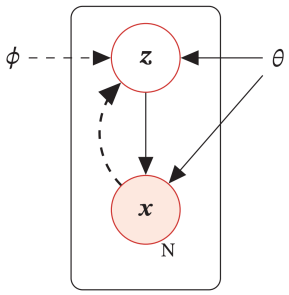

图 1: 一个含隐变量的生成模型，实线表示生成模型，虚线表示变分近似.

$$p(x,z;\theta)=p(x|z;\theta)p(z;\theta)$$

其中 $p(z;\theta)$ 变量 z 先验分布的概率密度函数，$p(x|z;\theta)$ 为已知 z 时观测变量 x 的条件概率密度函数，θ 表示两个密度函数的参数。
对于一个数据 x 其对数边际似然 $\log p(x;\theta)$ 能够被分解为:

$$\log p(x;\theta)=ELBO(q,x;\theta,\phi)+KL(q(z;\phi),p(z|x;\phi))$$

其中 $q(z;\phi)$ 额外引入的变分密度函数，其参数为 ϕ,$ELBO(q,x;\theta,\phi)$ 为证据下界:

$$ELBO(q,x;\theta,\phi)=\mathbb{E}_{z\sim q(z;\phi)}[\log\frac{p(x,z;\phi)}{q(z;\phi)}]$$

后验概率密度函数 $p(z|x;\phi)$ 计算，通常需要通过变分推断来近似估计。
VAE 利用神经网络来分别建模两个复杂的条件概率密度函数。
用神经网络来估计变分分布 $q(z;\phi)$ 为 encoder。
用神经网络来估计概率分布 $p(z|x;\phi)$, 为 decoder。
图2展示了 VAE 的整体结构。

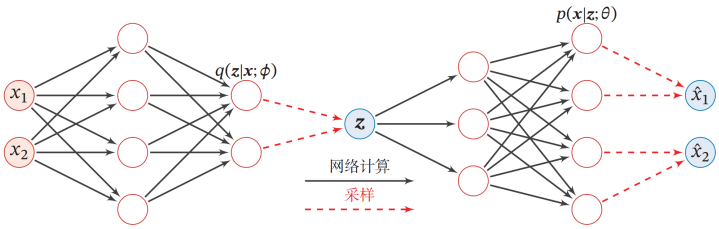

图 2: *VAE* 的结构图，其中实线为网络计算，虚线为采样.

2.2. Enocder 的训练目标

2.2 Enocder 的训练目标

Encoder 的目标是使得 $q(z|x;\phi)$ 能接近真实的后验 $p(z|x,\theta)$ 需要找到一组网络参数 ϕ^* * 来最小化两个分布的 KL 散度。然而 $p(z|x,\theta)$ 无法计算。利用变分分布 $q(z|x;\phi)$ 与真实后验 $p(z|x,\theta)$ 散度等于对数边际似然 $\log p(x;\theta)$ 与其下界 $ELBO(q,x;\theta,\phi)$ 的差，推断网络的目标函数可以转换为如下形式。

$$\phi^*=argmax_\phi ELBO(q,x;\theta,\phi)$$

[1]参考文献：邱锡鹏，神经网络与深度学习，机械工业出版社，https://nndl.github.io/, 2020.

2.3. Decoder 的训练目标

2.3 Decoder 的训练目标

生成模型的联合分布 $p(x,z;\phi)$ 分解为两部分：隐变量 z 的先验分布 $p(z;\theta)$，条件概率分布 $p(x|z;\theta)$。假设隐变量 z 的先验分布为各向同性的标准高斯分布 $\mathcal{N}(z|0,I)$ 变量 z 每一维之间都是独立的。条件概率分布 $p(x|z;\theta)$ 通过生成网络来建模。生成网络 $f_D(z;\theta)$ 的目标是找到一组网络参数 θ^* 来最大化证据下界 $ELBO(q,x;\theta,\phi)$。

2.4. 模型训练

2.4 模型训练

VAE 的总目标函数为：

$$
\begin{aligned}
&max_{\theta,\phi}ELBO(q,x;\theta,\phi)\\
&=max_{\theta,\phi}\mathbb{E}_{z\sim q(z;\phi)}[\log\frac{p(x|z;\theta)p(z;\theta)}{q(z;\phi)}]\\
&=\mathbb{E}_{z\sim q(z;\phi)}[\log p(x|z;\theta)]-KL(q(z|x;\phi),p(z;\theta))
\end{aligned}
$$

第一项的期望在实际训练过程中可以中可以通过采样来近似。为了使得梯度能够计算，需要使用再参数化的方式将 z 和 ϕ 之间随机性的采样关系转变为确定性函数关系。具体方式为引入一个分布为 $p(\epsilon)$ 的随机变量 ϵ，这样 z 和参数 ϕ 的关系从采样关系变为确定性关系，使得 $z\sim q(z|x;\phi)$ 性独立于参数 ϕ 从而可以求 z 关于 ϕ 的导数。
第二项 KL 散度可以直接通过定义计算。综上，假设 $p(x|z;\theta)$ 服从高斯分布 $\mathcal{N}(x|\mu_D,\lambda I)$，其中 $\mu_D=f_D(z;\theta)$ 是 decoder 的输出，则目标函数可以简化为：

$$
L(\phi,\theta|x)=-\frac{1}{2}||x-\mu_G||-\lambda KL(\mathcal{N}(\mu_I,\sigma_I),\mathcal{N}(0,I))
$$

3. 模型结构

3 模型结构

整体模型结构按照图2进行实现。首先是 encoder 部分，通过全连接神经网络把输入映射到另一个维度，同时加上激活函数 (RELU)，接着利用两个不同的全连接网络将其隐藏层映射到两个相同维度的向量，分别用于建模 z 的期望和方差。

encoder 部分代码结构如下：

```
class encoder(nn.Module):
    def __init__(self, dim_image, dim_hid, latent_dim):
        super(encoder, self).__init__()
        self.mean_net  = nn.Linear(dim_hid, latent_dim)
        self.varianece_net = nn.Linear (dim_hid, latent_dim)
        self.fc_layer = nn.Sequential(nn.Linear(dim_image, dim_hid),nn.ReLU(),nn.Linear(dim_hid, dim_hid),nn.ReLU())
    def forward(self, x):
        hid = self.fc_layer(x)
        var = self.varianece_net(hid)
        mean = self.mean_net(hid)
        return mean, var
```

将 ecoder 中得到的期望和方差通过再参数化的方式进行采样，得到隐层向量 z。然后将 z 通过全连接层和激活函数输出生成的图像。encoder 部分代码结构如下：

```
decoder = nn.Sequential(nn.Linear(args.dim_z, args.dim_hid),nn.ReLU(),nn.Linear(args.dim_hid, args.dim_image*args.dim_image),nn.Sigmoid())
```

4. 模型训练

4 模型训练

模型的训练过程首先将图片输入到由 encoder 和 decoder 组成的 VAE 模型中，然后将得到的生成图片以及 z 的期望方差。通过 2.4 中的损失函数（使用了 nn.functional.binary_cross_entropy）以及 Adam 优化器对模型参数进行优化。模型训练代码如下：

```
for batch_idx, (data, _) in enumerate(train_loader):
            data = data.view(args.batch_size, args.dim_image*args.dim_image)
            data = data.to(DEVICE)
            optimizer.zero_grad()
            x_hat, mean, var = model(data)
            loss = loss_function(data, x_hat, mean, var)
            overall_loss += loss.item()
            loss.backward()
            optimizer.step()
```

4.1. z 的维度为 1 的图像生成效果

4.1 z 的维度为 1 的图像生成效果

将 z 的维度设定为 1。由于 z 的分布假设为期望为 0 的高斯分布，故 z 的采样范围设置为(-1.5,1.5), 采样间隔为 0.03，得到的实验效果图如图3所示。可以发现随着 z 的值的不断增加。整个生成图像的形状由扁平到圆润，同时观察发现图像的变化是相对比较连续的。形状相似的数字的 z 的分布比较接近。通过这些发现可以看出当 z 的维度为 1 时，该维度所体现的信息主要为数字的扁平特性。

4.2. z 的维度为 2 的图像生成效果

4.2 z 的维度为 2 的图像生成效果

将 z 的维度设定为 2，在 2 个维度上在（-5，5）的范围上以 0.2 分别等间距采样生成的图像如图4所示。通过实验生成的图像发现，大部分数字集中于原点（0.0）附近。同时发现横轴方向与数字的倾斜程度相关性较强。同时横轴方向与对比度相关性较大。

4.3. 最优化重构目标

4.3 最优化重构目标

在损失函数中只保存重构误差进行实验（同时设置 z 的维度为 1）。在实验过程中发现，z 的分布更加分散，故选择范围在（-16，5）以 0.1 为间隔采样得到的结果进行展示。如图5所示。与之前的生成图像进行对比可以发现，其对比度相对较低，更加模糊。

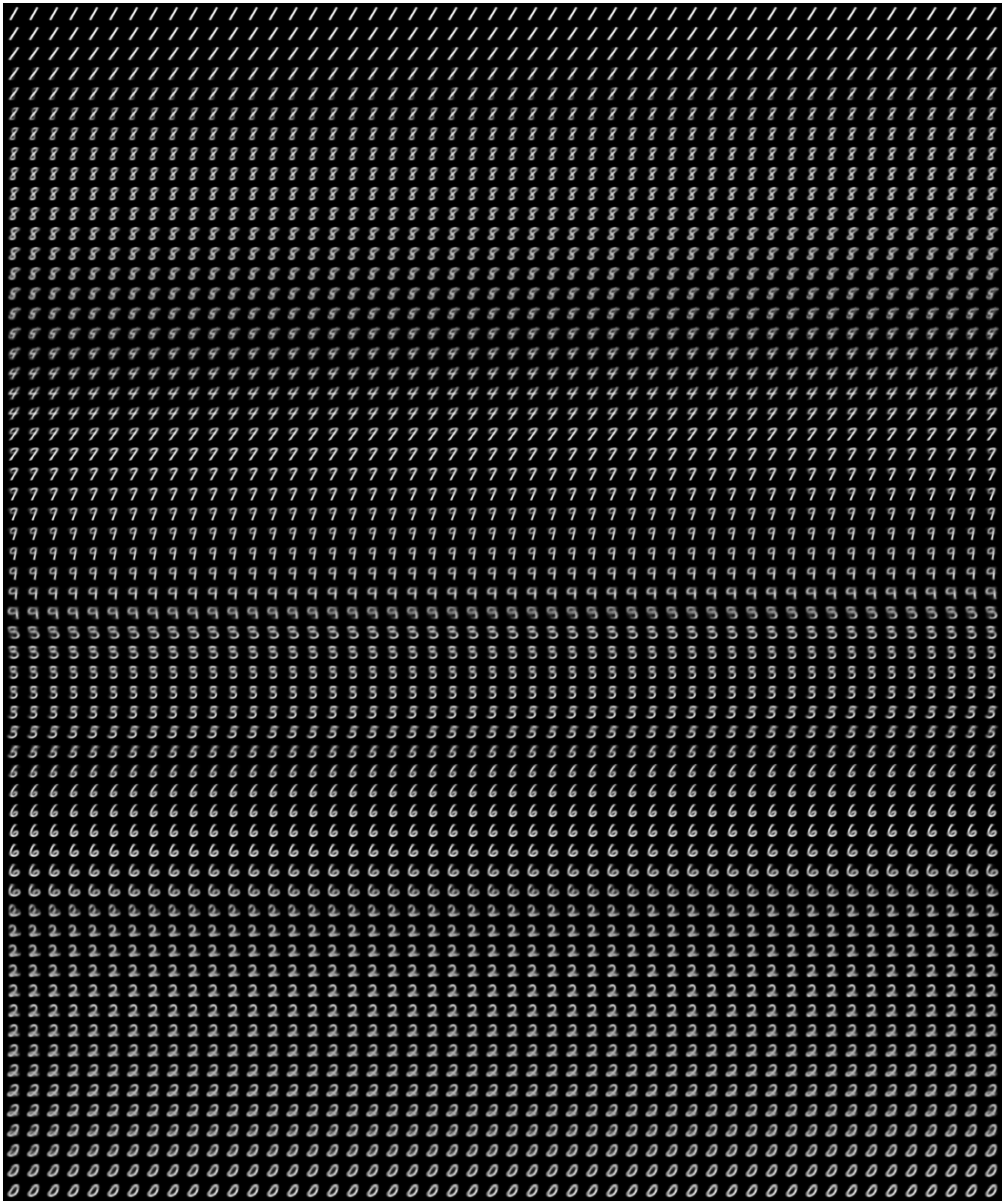

图 3: z 的维度为 1 的图像生成效果

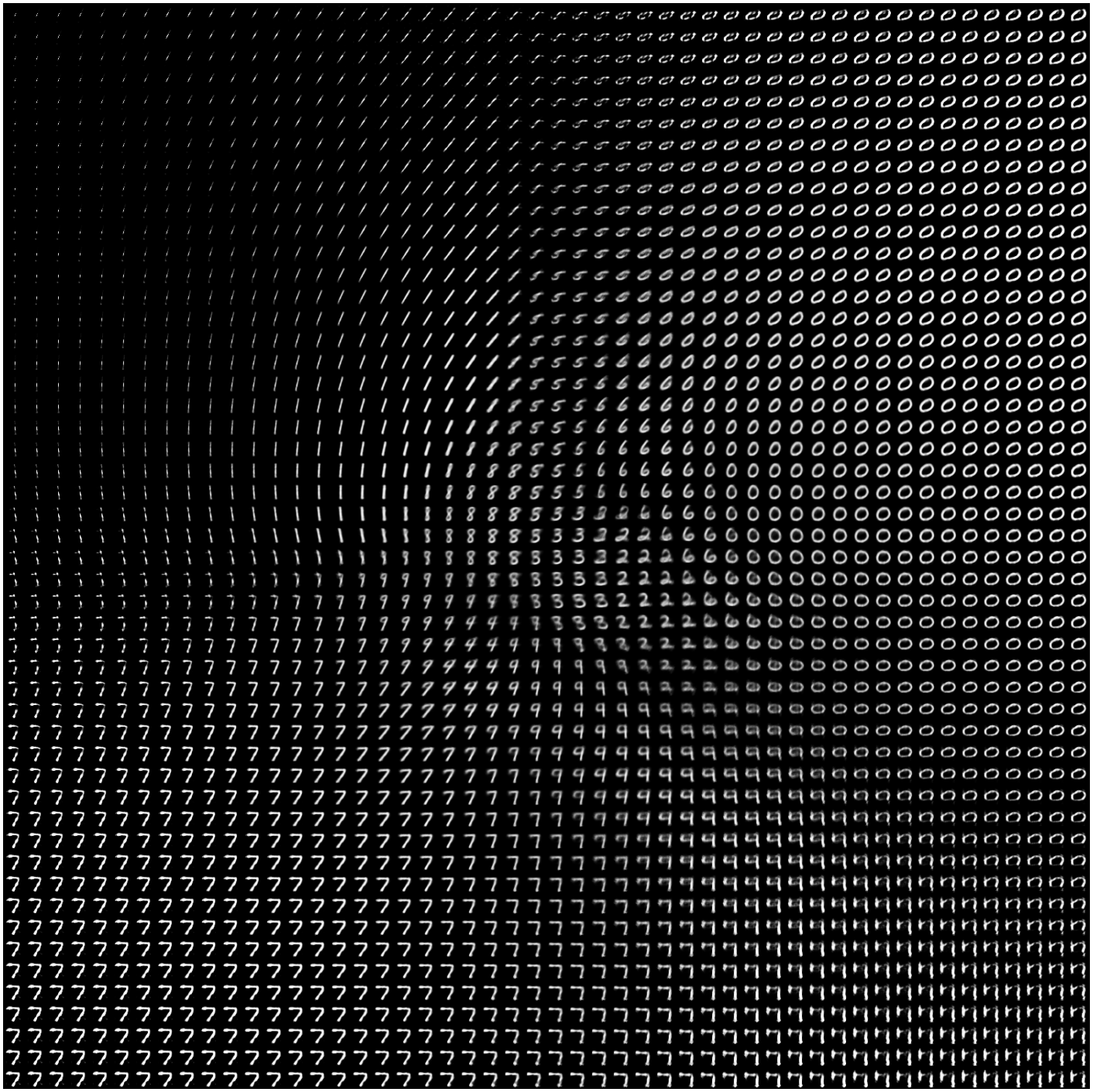

图 4: z 的维度为 2 的图像生成效果

图 5: 维度为 *1* 且只有重构损失训练之后的生成效果