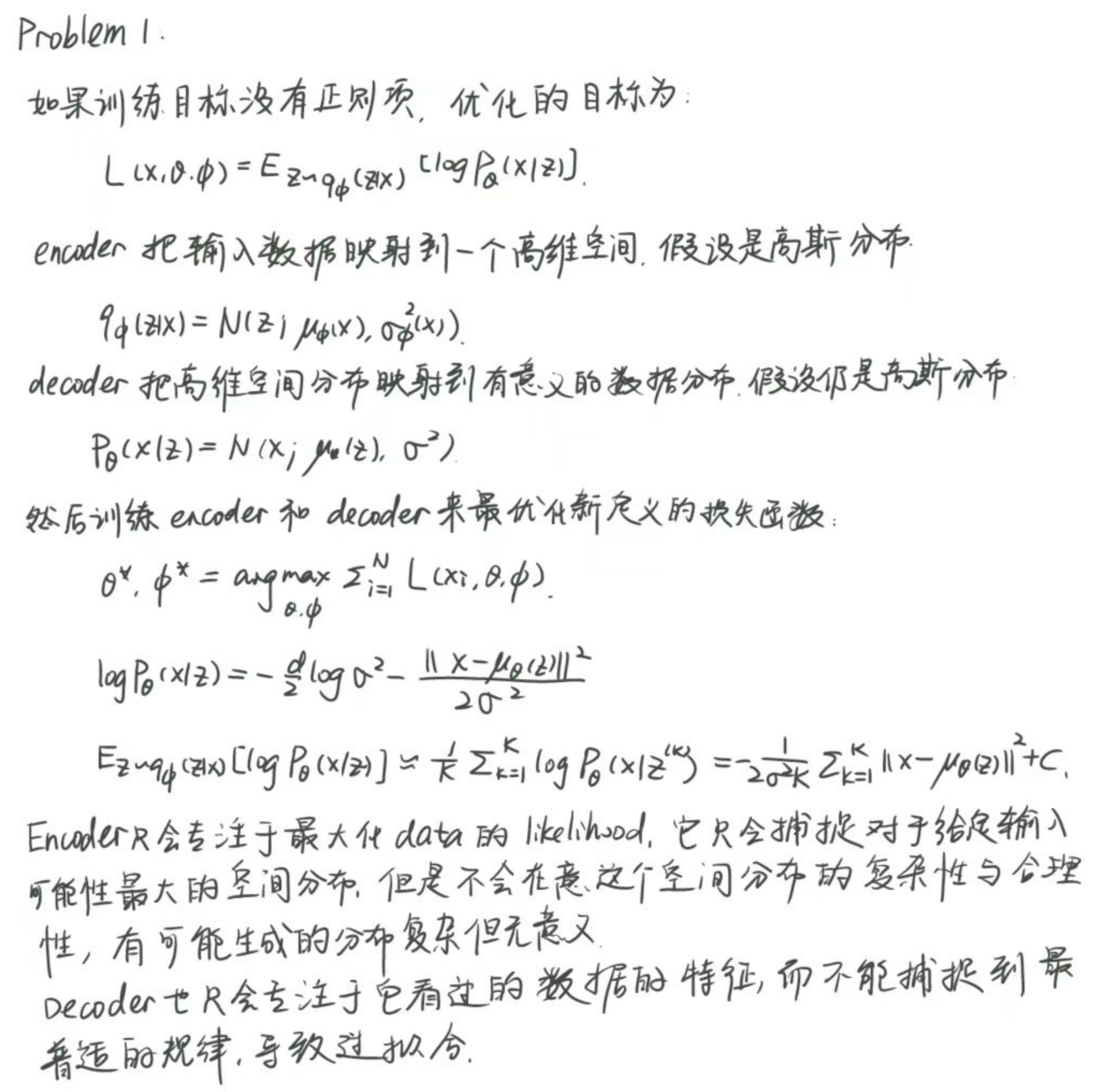
模式识别与机器学习HW13

胡清茹 2020012996

Problem1



Problem 1.

如果训练目标没有正则项，优化的目标为：

$$L(x,\theta,\phi)=E_{z\sim q_\phi(z|x)}[\log P_\theta(x|z)].$$

encoder 把输入数据映射到一个高维空间，假设是高斯分布.

$$q_\phi(z|x)=N(z;\mu_\phi(x),\sigma_\phi^2(x)).$$

decoder 把高维空间分布映射到有意义的数据分布，假设仍是高斯分布.

$$P_\theta(x|z)=N(x;\mu_\theta(z),\sigma^2).$$

然后训练 encoder 和 decoder 来最优化新定义的损失函数：

$$\theta^*,\phi^*=\arg\max_{\theta,\phi}\Sigma_{i=1}^N L(x_i,\theta,\phi).$$

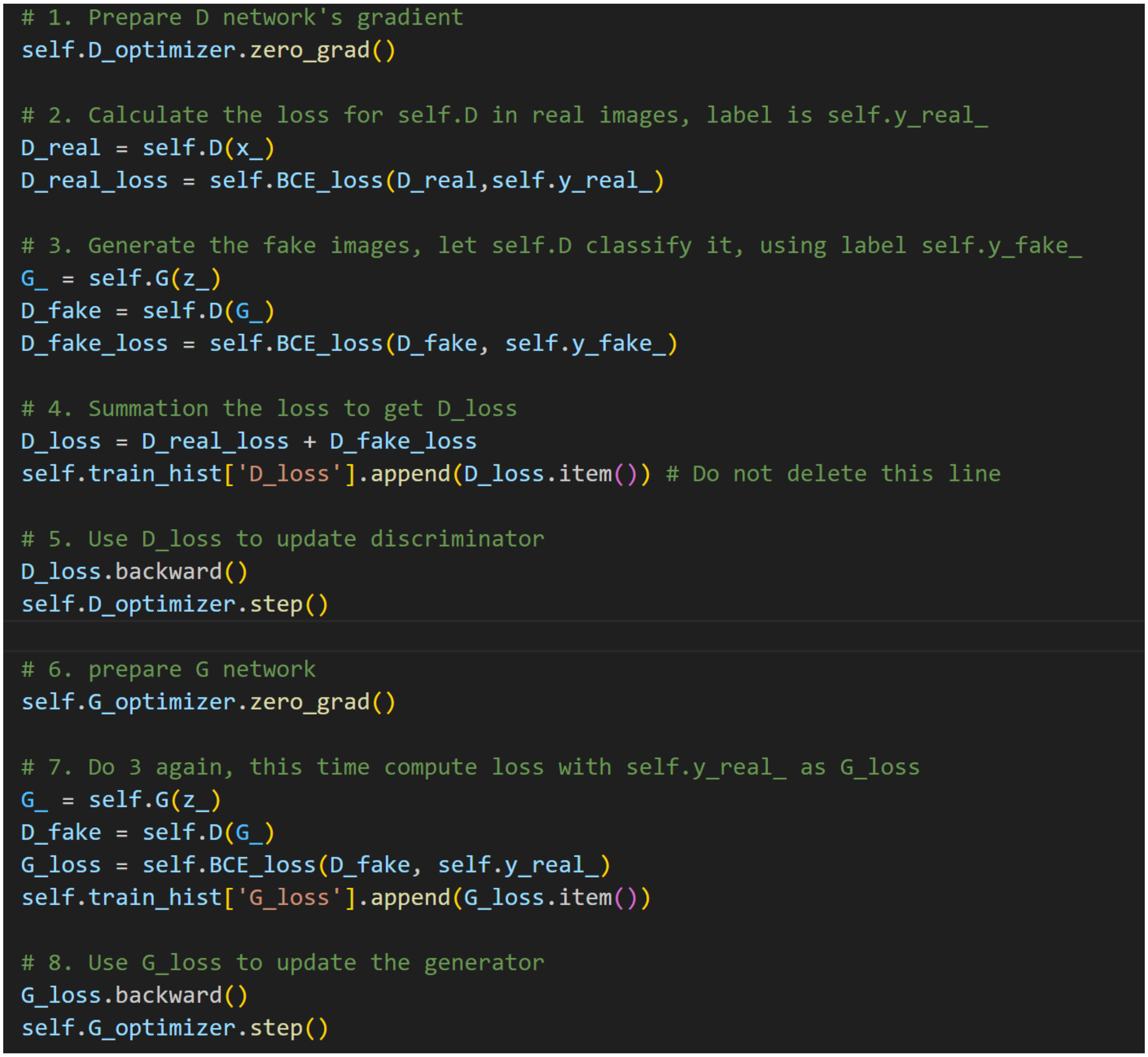
$$\log P_\theta(x|z)=-\frac{d}{2}\log\sigma^2-\frac{\|x-\mu_\theta(z)\|^2}{2\sigma^2}$$

$$E_{z\sim q_\phi(z|x)}[\log P_\theta(x|z)]\simeq\frac{1}{K}\Sigma_{k=1}^K\log P_\theta(x|z^{(k)})=-\frac{1}{2\sigma^2K}\Sigma_{k=1}^K\|x-\mu_\theta(z)\|^2+C.$$

Encoder只会专注于最大化 data 的 likelihood，它只会捕捉对于给定输入可能性最大的空间分布，但是不会在意这个空间分布的复杂性与合理性，有可能生成的分布复杂但无意义.
Decoder也只会专注于它看过的数据的特征，而不能捕捉到最普适的规律，导致过拟合.

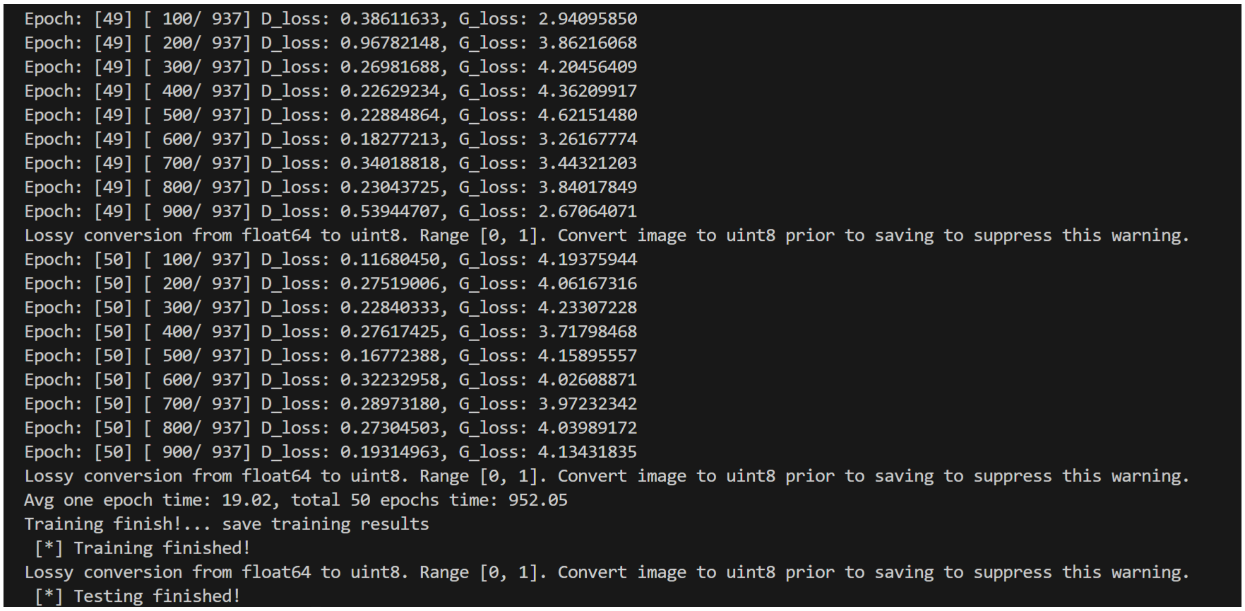
Problem2

补全的GAN代码如下：



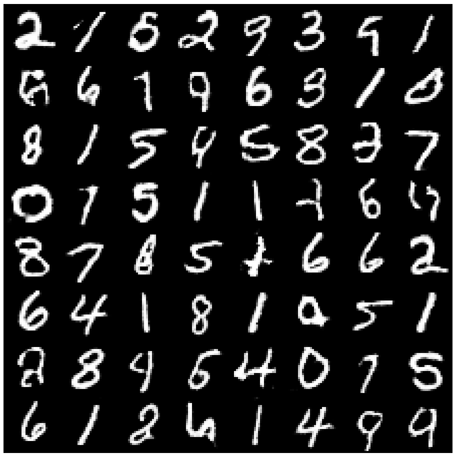
```
# 1. Prepare D network's gradient
self.D_optimizer.zero_grad()

# 2. Calculate the loss for self.D in real images, label is self.y_real_
D_real = self.D(x_)
D_real_loss = self.BCE_loss(D_real,self.y_real_)

# 3. Generate the fake images, let self.D classify it, using label self.y_fake_
G_ = self.G(z_)
D_fake = self.D(G_)
D_fake_loss = self.BCE_loss(D_fake, self.y_fake_)

# 4. Summation the loss to get D_loss
D_loss = D_real_loss + D_fake_loss
self.train_hist['D_loss'].append(D_loss.item()) # Do not delete this line

# 5. Use D_loss to update discriminator
D_loss.backward()
self.D_optimizer.step()

# 6. prepare G network
self.G_optimizer.zero_grad()

# 7. Do 3 again, this time compute loss with self.y_real_ as G_loss
G_ = self.G(z_)
D_fake = self.D(G_)
G_loss = self.BCE_loss(D_fake, self.y_real_)
self.train_hist['G_loss'].append(G_loss.item())

# 8. Use G_loss to update the generator
G_loss.backward()
self.G_optimizer.step()
```

训练过程中的Loss如下：



```
Epoch: [49] [ 100/ 937] D_loss: 0.38611633, G_loss: 2.94095850
Epoch: [49] [ 200/ 937] D_loss: 0.96782148, G_loss: 3.86216068
Epoch: [49] [ 300/ 937] D_loss: 0.26981688, G_loss: 4.20456409
Epoch: [49] [ 400/ 937] D_loss: 0.22629234, G_loss: 4.36209917
Epoch: [49] [ 500/ 937] D_loss: 0.22884864, G_loss: 4.62151480
Epoch: [49] [ 600/ 937] D_loss: 0.18277213, G_loss: 3.26167774
Epoch: [49] [ 700/ 937] D_loss: 0.34018818, G_loss: 3.44321203
Epoch: [49] [ 800/ 937] D_loss: 0.23043725, G_loss: 3.84017849
Epoch: [49] [ 900/ 937] D_loss: 0.53944707, G_loss: 2.67064071
Lossy conversion from float64 to uint8. Range [0, 1]. Convert image to uint8 prior to saving to suppress this warning.
Epoch: [50] [ 100/ 937] D_loss: 0.11680450, G_loss: 4.19375944
Epoch: [50] [ 200/ 937] D_loss: 0.27519006, G_loss: 4.06167316
Epoch: [50] [ 300/ 937] D_loss: 0.22840333, G_loss: 4.23307228
Epoch: [50] [ 400/ 937] D_loss: 0.27617425, G_loss: 3.71798468
Epoch: [50] [ 500/ 937] D_loss: 0.16772388, G_loss: 4.15895557
Epoch: [50] [ 600/ 937] D_loss: 0.32232958, G_loss: 4.02608871
Epoch: [50] [ 700/ 937] D_loss: 0.28973180, G_loss: 3.97232342
Epoch: [50] [ 800/ 937] D_loss: 0.27304503, G_loss: 4.03989172
Epoch: [50] [ 900/ 937] D_loss: 0.19314963, G_loss: 4.13431835
Lossy conversion from float64 to uint8. Range [0, 1]. Convert image to uint8 prior to saving to suppress this warning.
Avg one epoch time: 19.02, total 50 epochs time: 952.05
Training finish!... save training results
 [*] Training finished!
Lossy conversion from float64 to uint8. Range [0, 1]. Convert image to uint8 prior to saving to suppress this warning.
 [*] Testing finished!
```

训练10次和50次生成的图像依次如下所示，可以看到随着训练轮数的增加，GAN生成的图像更加清晰更加贴近实际。



致谢

感谢周亦涵同学在本次作业与我的讨论和给我的指导与帮助。