量化投资作业 4

Author’s Name：黄倪远
ID：3200101028

1. 作业要求

- 理解课堂展示的“demo_4_gan”项目，并将原始GAN改造为WGAN-GP。

2. 实验过程

1. 在网络组建中删除`nn.Sigmoid()`

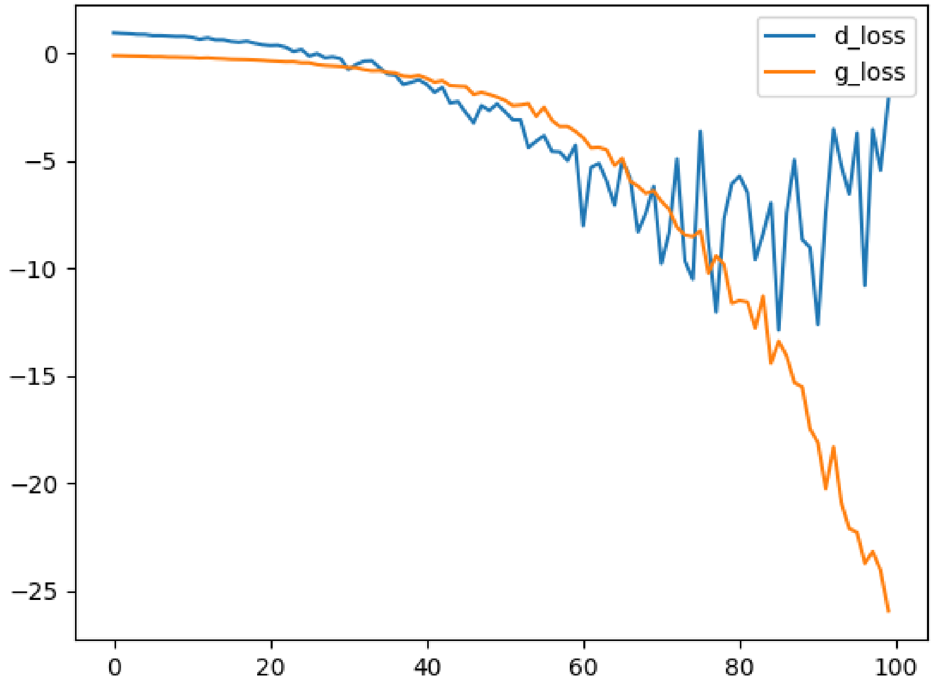
2. 修改新的loss函数

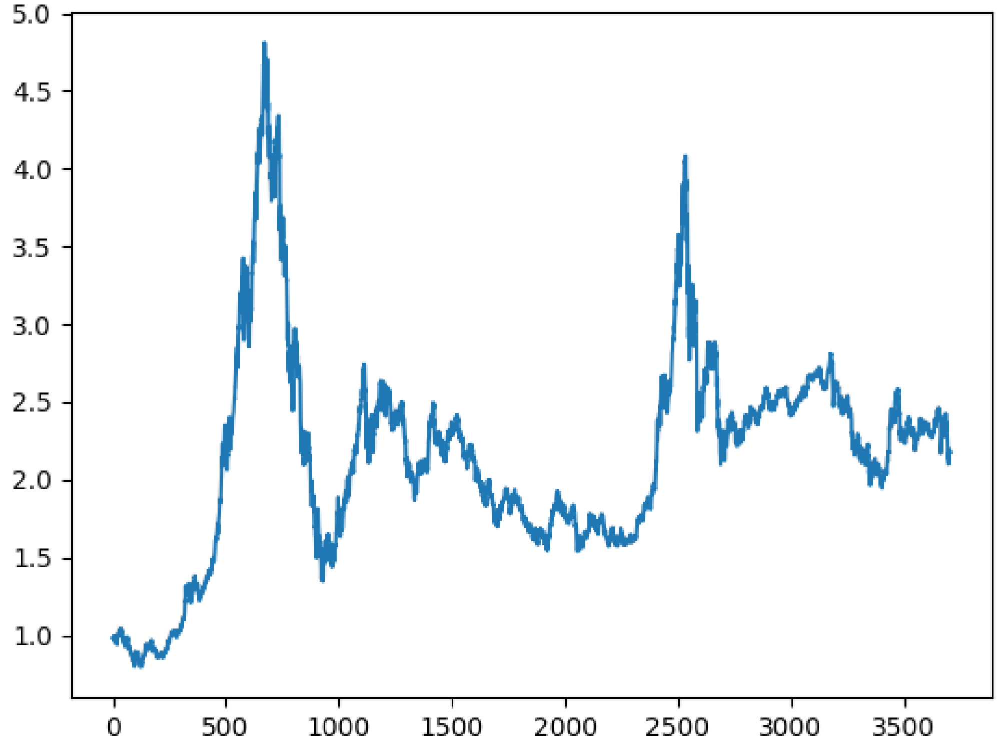
```
# a. 训练判别器
# a4. 判别器进行预测
        real_pred = D(real_data)
        fake_pred = D(fake_data)
 ...
# b. 训练生成器
# b3. 计算损失
        # 不做noise labeling
        real_label = torch.ones(param.batch_size,
1).to(param.device)
        g_loss = -fake_pred.mean()
```

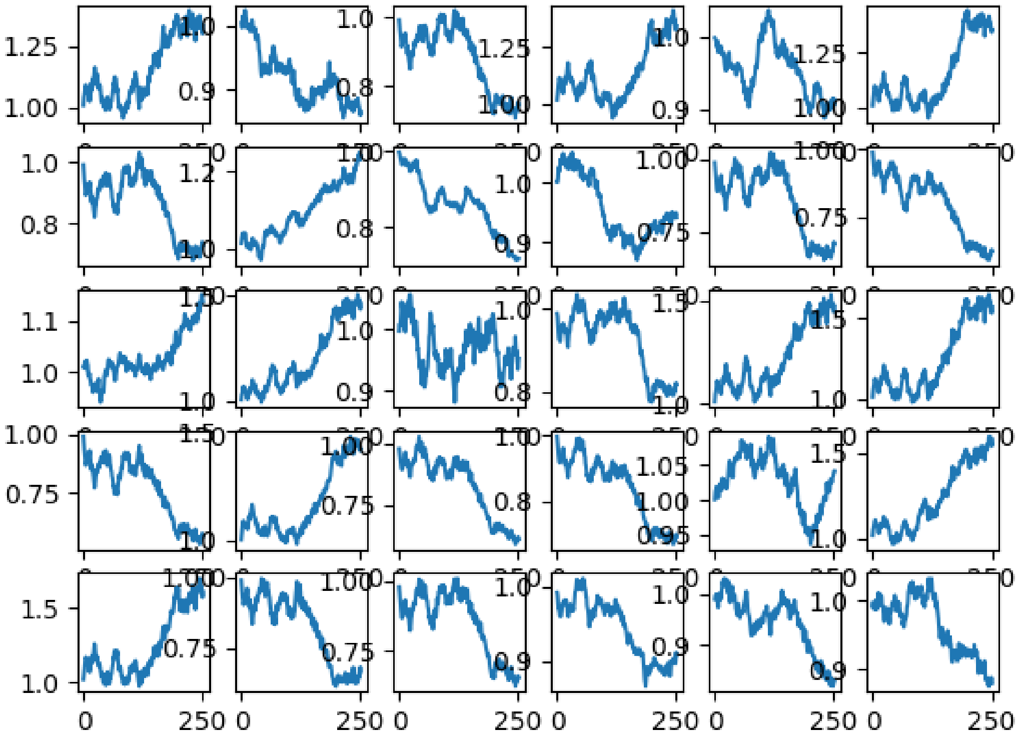
3. 设置梯度惩罚项

```
    def compute_gradient_penalty(D, real_data, fake_data,
lambda_term, batch_size, cuda_index):
        tensor = torch.FloatTensor(batch_size, 1, 1,
1).uniform_(0, 1)
        tensor = tensor.expand(batch_size, real_data.size(1),
real_data.size(2), real_data.size(3))
        if cuda:
            tensor = tensor.cuda(cuda_index)
        else:
            tensor = tensor
        interpolated = tensor * real_data + ((1 - tensor) *
fake_data)
        if cuda:
            interpolated = interpolated.cuda(cuda_index)
        else:
            interpolated = interpolated
        interpolated = Variable(interpolated, requires_grad=True)
        d_interpolated = D(interpolated)
        grads = autograd.grad(outputs=d_interpolated,
inputs=interpolated,

 grad_outputs=torch.ones(d_interpolated.size()).cuda(cuda_index)
if cuda else torch.ones(
                                  d_interpolated.size()),
create_graph=True, retain_graph=True)[0]
        grad_penalty = ((grads.norm(2, dim=1) - 1) ** 2).mean() *
lambda_term
        return grad_penalty

    d_optimizer = torch.optim.Adam(D.parameters(), lr=1e-5, betas=
(0.1, 0.999))
    g_optimizer = torch.optim.Adam(G.parameters(), lr=2e-4, betas=
(0.5, 0.999))
```

3. 实验结果




5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0
500
1000
1500
2000
2500
3000
3500


1.25
1.00
1.0
0.9
1.0
0.8
1.25
1.00
1.0
0.9
1.25
1.00
1.0
0.8
1.2
1.0
1.0
0.8
1.00
0.75
1.00
0.75
1.1
1.0
1.0
0.9
1.0
0.8
1.5
1.0
1.5
1.0
1.00
0.75
1.5
1.0
1.00
0.75
1.0
0.8
1.05
1.00
0.95
1.5
1.0
1.5
1.0
1.00
0.75
1.00
0.75
1.0
0.9
1.0
0.9
1.0
0.9
1.0
0.9
0
250
0
250
0
250
0
250
0
250
0
250