light_data_3.9/result/3.10：

（1）/rand_bias0.3：采样率 10M，均匀分布，偏置电流 0.3A。

1　/single_amp: 单一幅度数据作为训练数据，且数据归一化。发送信号是均匀分布的随机信号，采样率为 10M，偏置电流 0.3A。

1.1 /single_amp/Threenonlinear1：

三层非线性层，隐藏层点数改为 60

```
Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 200 ,
DropPeriod = 12 ,
DropFactor = 0.100000 ,
amp begin = 1 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(size(xTrain{1},2)/miniBatchSize
Hidden Units = 60
```

1.2 /single_amp/Threenonlinear2：

三层非线性层，隐藏层点数改为 60，更改归一化因子

```
Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 200 ,
DropPeriod = 12 ,
DropFactor = 0.100000 ,
amp begin = 1 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(size(xTrain{1},2)/miniBatchSize
Hidden Units = 60
```

原来：

```
load_path = "data_save/light_data_2.28/result/3.1/25M/8pam/mix_amp/Twononlinear";
```

现在：

```
load_path = "data_save/light_data_3.9/data/10M/rand_bias0.3/";
norm_mat = load(load_path+"/save_norm.mat");
```

使用的归一化因子不同。

light_data_3.10/result/3.10：

（1） /rand_bias0.3：采样率 10M，均匀分布，偏置电流 0.3A。

1. /norm_LS: 用 LS 算法，求出各个幅度信号的 NMSE。LS 估计时用的信号是归一化之后的信号，但归一化的方式变了。

   原来：

```
xTrain_tmp = cellfun(@(cell1)(cell1*100*1.1^amp),xTrain_tmp,'UniformOutput',false);
xTest_tmp = cellfun(@(cell1)(cell1*100*1.1^amp),xTest_tmp,'UniformOutput',false);
```

   现在：

```
xTrain_tmp = cellfun(@(cell1)(cell1*32000*(0.0015+(amp-1)*0.03994)),xTrain_tmp,'UniformOutput',false);
xTest_tmp = cellfun(@(cell1)(cell1*32000*(0.0015+(amp-1)*0.03994)),xTest_tmp,'UniformOutput',false);
```

2. /single_amp: 单一幅度数据作为训练数据，且数据归一化。发送信号是均匀分布的随机信号，采样率为 10M，偏置电流 0.3A。归一化的方式变了，同时训练数据还多乘了一个归一化因子，该因子是光路发送信号调整幅度时用来幅度归一化的。

   原来：

```
xTrain_tmp = cellfun(@(cell1)(cell1*100*1.1^amp),xTrain_tmp,'UniformOutput',false);
xTest_tmp = cellfun(@(cell1)(cell1*100*1.1^amp),xTest_tmp,'UniformOutput',false);
```

   现在：

```
xTrain_tmp = cellfun(@(cell1)(cell1*32000*(0.0015+(amp-1)*0.03994)),xTrain_tmp,'UniformOutput',false);
xTest_tmp = cellfun(@(cell1)(cell1*32000*(0.0015+(amp-1)*0.03994)),xTest_tmp,'UniformOutput',false);
```

   多乘了一个归一化因子：

```
upsample_norm(name_order) = gather(eval(strcat('upsample_norm_mat.',upsample_norm_names{name_order})));

x{10*(name_order-1)+i} = [zeros(1,15),data_ori(1000*(i-1)+1:1000*i)]/upsample_norm(name_order);
```

   2.1 /single_amp/Threenonlinear1

```
Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 200 ,
DropPeriod = 12 ,
DropFactor = 0.100000 ,
amp begin = 1 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(size(xTrain{1},2)/miniBatchSize
Hidden Units = 40
```

3. /mix_amp: 混合幅度数据作为训练数据，且数据归一化。发送信号是均匀分布的随机信号，采样率为 10M，偏置电流 0.3A。归一化的方式变了，同时训练数据还多乘了一个归一化因子，该因子是光路发送信号调整幅度时用来幅度归一化的。

原来：

```
xTrain_tmp = cellfun(@(cell1)(cell1*100*1.1^amp),xTrain_tmp,'UniformOutput',false);
xTest_tmp = cellfun(@(cell1)(cell1*100*1.1^amp),xTest_tmp,'UniformOutput',false);
```

现在：

```
xTrain_tmp = cellfun(@(cell1)(cell1*32000*(0.0015+(amp-1)*0.03994)),xTrain_tmp,'UniformOutput',false);
xTest_tmp = cellfun(@(cell1)(cell1*32000*(0.0015+(amp-1)*0.03994)),xTest_tmp,'UniformOutput',false);
```

多乘了一个归一化因子：

```
upsample_norm(name_order) = gather(eval(strcat('upsample_norm_mat.',upsample_norm_names{name_order})));
x{10*(name_order-1)+i} = [zeros(1,15),data_ori(1000*(i-1)+1:1000*i)]/upsample_norm(name_order);
```

3.1 /mix_amp/Twononlinear1

```
Twononlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 400 ,
DropPeriod = 5 ,
DropFactor = 0.100000 ,
amp begin = 1 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)
Hidden Units = 25
```