2.28工作内容：

1. 2.25调试日志.docx 中 2.非线性层 内容。将训练数据归一化，并且将归一化后的不同snr（不同amp）的发送数据拼在一起当作训练数据，以此训练出适用于所有snr的网络。

2. 将性能评判标准由mse改为nmse。

3. 采集了足够多的不同snr（不同amp）下的光路数据，储存在文件夹light\_data\_2.28 中。

3.1工作内容：（使用数据：light\_data\_2.28）

1. 对2.28采集到的不同snr的光路数据，用LS算法计算了不同snr时的NMSE。

2. 对2.28采集到的不同snr的光路数据，将不同snr的信号拼接在一起当作训练数据，计算了不同snr时的NMSE。

3. 修改了test\_dnn，使其可以适用不同速率的收发数据。

4. 对2.28采集到的不同snr的光路数据，将单一snr的信号当作训练数据，计算了不同snr时的NMSE。

3.2工作内容：（使用数据：light\_data\_2.28）

见3.2调试日志，继续对混合snr训练的神经网络进行调整。

3.3工作内容：（使用数据：light\_data\_2.28）

1. 修改了test\_dnn3，之前的测试数据集只有xTest，现在将xTrain也当作测试集测试一遍，即xTrain既是训练数据又是测试数据。将二者的结果对比可以看出有没有过拟合。

2. 见3.3调试日志，继续对3.2的混合snr训练的神经网络进行调整。不同的是，3.3的测试数据集不仅是xTest,还有用来训练的xTrain也做一遍测试，看看有没有过拟合。

3.4 工作内容：（使用数据：light\_data\_3.4）

1. 采集了新的光路25M发送数据、150M接收数据，存放在文件夹light\_data\_3.4/data中。与之前不同的是，之前的发送数据都是8pam数据，而这次的发送数据是均匀分布的随机数据。

2. 见3.4调试日志，计算了LS方法下的NMSE和混合snr数据训练时的NMSE。

3.6 工作内容：（使用数据：light\_data\_3.4）

1. 见3.6调试日志，计算了单一snr数据训练时的NMSE。

3.8 工作内容：（使用数据：light\_data\_3.4、light\_data\_3.8）

1. 采集了新的光路10M发送数据、60M接收数据，存放在文件夹light\_data\_3.8/data中。与light\_data\_3.4不同的是，3.4的发送数据是25M，而这次的发送数据是10M，但都是均匀分布的随机数据。

rand\_bias0.3：10M发送，60M接收，均匀分布，偏置电流0.3A。

rand\_bias0.6：10M发送，60M接收，均匀分布，偏置电流0.6A。

2. 见3.8调试日志。

3.9 工作内容：（使用数据：light\_data\_3.9）

1. 采集了新的光路10M发送数据、60M接收数据，存放在文件夹light\_data\_3.9/data中。与light\_data\_3.8不同的是，3.8是按照snr划分信号，而3.9是按照发送信号幅度划分信号，但都是均匀分布的随机数据。

rand\_bias0.3：10M发送，60M接收，均匀分布，偏置电流0.3A。

rand\_bias0.6：10M发送，60M接收，均匀分布，偏置电流0.6A。

2. 见3.9调试日志。

3.10 工作内容：（使用数据：light\_data\_3.9、light\_data\_3.10）

1. 采集了新的光路10M发送数据、60M接收数据，存放在文件夹light\_data\_3.10/data中。与light\_data\_3.9不同的是，3.10还存了发送信号幅度归一化时的归一化因子。

rand\_bias0.3：10M发送，60M接收，均匀分布，偏置电流0.3A。

2. 见3.10调试日志。

3.11 工作内容：（使用数据：light\_data\_3.10）

1. 采集了新的光路10M发送数据、150M接收数据，存放在文件夹light\_data\_3.11/data中。

rand\_bias0.3：10M发送，150M接收，均匀分布，偏置电流0.3A。

2. 见3.11调试日志。

3.12 工作内容：（使用数据：light\_data\_3.10）

1. 见3.12调试日志。

3.13 工作内容：（使用数据：light\_data\_3.10、light\_data\_3.11）

1. 见3.13调试日志。

3.14 工作内容：（使用数据：light\_data\_3.10、light\_data\_3.11）

1. 见3.14调试日志。

2. 更改程序ls\_esi2.m。之前的LS都是利用一个信号来生成LS矩阵，现在改为用多个信号来生成LS矩阵。

3.15 工作内容：（使用数据：light\_data\_3.10）

1. 见3.15调试日志。

2. 新增程序test\_dnn4.m。与之前不同的是，此程序用的训练数据是幅度较大的几个数据，而不是全部幅度的数据，以此来试验用大幅度数据训练出来的网络能否适用于小幅度数据。

3.16 工作内容：（使用数据：light\_data\_3.10、light\_data\_3.16）

1. 见3.16调试日志。

2. 采集了新的光路数据，10M发送数据、60M接收数据，存放在文件夹light\_data\_3.16/data中。与之前不同的是，这次采集的数据前面幅度信号都采集的很少，大部分数据采集的是幅度最大的信号的数据。以此来实验用大幅度数据训练出来的网络能否适用于小幅度数据。

3.17 工作内容：（使用数据：light\_data\_3.17）

1. 之前采集的数据的分布如下，选用的是发送数据幅度最大时的情况：

图表, 直方图

描述已自动生成

最左边一张图是生成的均匀分布数据过了一个高通滤波器后的数据分布图，右边两张图是上采样到160M过程中过滤波器后的图像，即最右面一张图是发送数据的分布图。

可以看到发送数据中，低幅度数据占了绝大多数，高幅度数据很少。

因此，决定一开始不生成均匀分布，而是生成碗形分布。

注：舍选法（由均匀分布生成任意分布）

文本, 信件

描述已自动生成

代码：生成的分布的分布函数为f(x)=x^2，为碗形分布

X = [ ];

while size(X) < 10000

x=rand\*2-1; % [-1 1]

y=rand;

if (y < x^2)

X = [X x];

end

end

生成的分布图：

图表, 直方图

描述已自动生成

最左边的图是生成的碗形分布的分布图，中间是过了高通滤波器之后的分布图，右边是发送数据的分布图，可以看到高幅度数据多了一些。

因此，决定采用最极端的碗形分布，即bpsk分布，分布图如下：

直方图

描述已自动生成

中间为bpsk信号过了高通滤波器后的分布图，最右面为发送数据的分布图。

以此来收集10M发送、60M接收，初始分布为bpsk的光路数据。此数据用于mix amp训练方式，为的是弥补之前mix amp训练时高幅度数据数量不足的情况。存放在light\_data\_3.17/data中。

2. 收集10M发送、60M接收，初始分布为bpsk的光路数据。与light\_data\_3.17/data不同的是，此次数据采集的signal\_ori是**没有经过高通滤波器的signal\_ori\_ini，即此次采集的数据包含了低频分量**。存放在light\_data\_3.17\_2/data中。

3. 见3.17调试日志

注：除了舍选法外，还可以用 反函数法 实现由均匀分布生成任意随机分布。

3.18 工作内容：（使用数据：light\_data\_3.17、light\_data\_3.17\_2）

1. 见3.18调试日志

3.22 工作内容：（使用数据：light\_data\_3.22/data、light\_data\_3.22/data2）

1. 见3.22调试日志。

2. 采集了新的光路数据，10M发送数据、60M接收数据，存放在文件夹light\_data\_3.22/data中。与之前不同的是，这次采集的数据是在不同偏置电流bias下、幅度最大（即amp=32000）的时候的数据。

3. 采集了新的光路数据，10M发送数据、60M接收数据，存放在文件夹light\_data\_3.22/data2中。与light\_data\_3.22/data不同的是，这次采集的数据是在不同偏置电流bias下、幅度中等（即amp= 5161.6）的时候的数据。

4. 新增程序ls\_bias.m、dnn\_single\_bias.m。分别用于对不同bias的信号做LS估计；以及对不同bias的信号进行单独训练，训练数据是相同bias、不同amp的数据。

3.23 工作内容：（使用数据：light\_data\_3.22/data、light\_data\_3.22/data2）

1. 见3.23调试日志。

2. 新增程序dnn\_all\_bias.m。分别对不同bias的信号进行混合训练，训练数据是混合bias、混合amp的数据。

3. 新增程序dnn\_trainedNet.m。用已经训练好的网络去测试数据。

3.27 工作内容：（使用数据：light\_data\_3.22/data/amp0.1613、amp1）

1. 见3.27调试日志。

2. 新增程序dnn\_all\_data.m。功能为：先用一部分数据训练网络，训练好之后将网络保存，然后加载新的数据接着训练保存好的网络，这样可以做到增加训练样本数据量。