

light_data_3.17/result/3.18:

(1) /rand_bias0.3: 采样率 10M, 接收速率 60M, bpsk 分布, 偏置电流 0.3A。

1. /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 bpsk 分布的随机信号, 采样率为 10M, 接收速率 60M, 偏置电流 0.3A。与之前不同的是, 由于此次用的是 bpsk 信号, 所以用来训练的数据中高幅度数据会多一些。

1.1 /Threenonlinear5: P = 5 , F = 0.1 , r=8, add_zero = 24

```
Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 400 ,
DropPeriod = 5 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)
H order = 48
Hidden Units = 60
Add zero num = 24
```

1.2 /Threenonlinear6: P = 6 , F = 0.1 , r=8, add_zero = 24

```
Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 400 ,
DropPeriod = 6 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)
H order = 48
Hidden Units = 60
Add zero num = 24
```

1.3 /Threenonlinear7: P = 7 , F = 0.1 , r=8, add_zero = 24

```
Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 400 ,
DropPeriod = 7 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)
H order = 48
Hidden Units = 60
Add zero num = 24
```

1.4 /Threenonlinear8: P = 8 , F = 0.1 , r=8, [add_zero = 24](#)

```
Threenonlinear ,  
ini learningRate = 1.000000e-02 ,  
min batch size = 400 ,  
DropPeriod = 8 ,  
DropFactor = 0.100000 ,  
amp begin = 2 , amp end = 26 , amp step = 1  
data_num = 100  
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)  
H order = 48  
Hidden Units = 60  
Add zero num = 24
```

1.5 /Threenonlinear9: P = 9 , F = 0.1 , r=8, [add_zero = 24](#)

```
Threenonlinear ,  
ini learningRate = 1.000000e-02 ,  
min batch size = 400 ,  
DropPeriod = 9 ,  
DropFactor = 0.100000 ,  
amp begin = 2 , amp end = 26 , amp step = 1  
data_num = 100  
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)  
H order = 48  
Hidden Units = 60  
Add zero num = 24
```

1.6 /Threenonlinear10: P = 10 , F = 0.1 , r=8, [add_zero = 24](#)

```
Threenonlinear ,  
ini learningRate = 1.000000e-02 ,  
min batch size = 400 ,  
DropPeriod = 10 ,  
DropFactor = 0.100000 ,  
amp begin = 2 , amp end = 26 , amp step = 1  
data_num = 100  
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)  
H order = 48  
Hidden Units = 60  
Add zero num = 24
```

light_data_3.17_2/result/3.18:

3.22 注: 3.18 跑的时候代码忘记改了, 用成高通的数据了。3.22 重新跑了几条线, 结果对的只有 LS 的 norm_LS4 、 single_amp 的 Threenonlinear4 和 mix_amp 的 Threenonlinear8、Threenonlinear9。

(1) /rand_bias0.3: 采样率 10M, 接收速率 60M, bpsk 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。用很多信号来生成 LS 矩阵。

1.1 norm_LS1: r = 5, add_zero = 15

```
LS
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
train num = 90
origin rate = 1.000000e+07 , receive rate = 6.000000e+07
H order = 30
Add zero num = 15
```

1.2 norm_LS2: r = 5, add_zero = 24

```
LS
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
train num = 90
origin rate = 1.000000e+07 , receive rate = 6.000000e+07
H order = 30
Add zero num = 24
```

1.3 norm_LS3: r = 8, add_zero = 15

```
LS
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
train num = 90
origin rate = 1.000000e+07 , receive rate = 6.000000e+07
H order = 48
Add zero num = 15
```

1.4 norm_LS4: r = 8, add_zero = 24

```

LS
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
train num = 90
origin rate = 1.000000e+07 , receive rate = 6.000000e+07
H order = 48
Add zero num = 24

```

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 bpsk

分布的随机信号, 采样率为 10M, 接收速率 60M, 偏置电流 0.3A。

2.1 /Threenonlinear1: r = 5, add_zero = 15

```

Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 200 ,
DropPeriod = 12 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(size(xTrain{1},2)/miniBatchSize
origin rate = 1.000000e+07 , receive rate = 6.000000e+07
H order = 30
Hidden Units = 60
Add zero num = 15

```

2.2 /Threenonlinear2: r = 5, add_zero = 24

```

Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 200 ,
DropPeriod = 12 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(size(xTrain{1},2)/miniBatchSize
origin rate = 1.000000e+07 , receive rate = 6.000000e+07
H order = 30
Hidden Units = 60
Add zero num = 24

```

2.3 /Threenonlinear3: r = 8, add_zero = 15

```

Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 200 ,
DropPeriod = 12 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(size(xTrain{1},2)/miniBatchSize
origin rate = 1.000000e+07 , receive rate = 6.000000e+07
H order = 48
Hidden Units = 60
Add zero num = 15

```

2.4 /Threenonlinear4: r = 8, add_zero = 24

```

Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 200 ,
DropPeriod = 12 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(size(xTrain{1},2)/miniBatchSize
origin rate = 1.000000e+07 , receive rate = 6.000000e+07
H order = 48
Hidden Units = 60
Add zero num = 24

```

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 **bpsk** 分布的随机信号,采样率为 10M,接收速率 60M,偏置电流 0.3A。与之前不同的是,由于此次用的是 **bpsk** 信号,所以用来训练的数据中高幅度数据会多一些。

3.1 /Threenonlinear1: P = 5 , F = 0.1 , r=8, [add_zero = 15](#)

```

Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 400 ,
DropPeriod = 5 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)
H order = 48
Hidden Units = 60
Add zero num = 15

```

3.2 /Threenonlinear2: P = 6 , F = 0.1 , r=8, [add_zero = 15](#)

```

Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 400 ,
DropPeriod = 6 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)
H order = 48
Hidden Units = 60
Add zero num = 15

```

3.3 /Threenonlinear3: P = 7 , F = 0.1 , r=8, [add_zero = 15](#)

```

Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 400 ,
DropPeriod = 7 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)
H order = 48
Hidden Units = 60
Add zero num = 15

```

3.4 /Threenonlinear4: P = 8 , F = 0.1 , r=8, [add_zero = 15](#)

```

Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 400 ,
DropPeriod = 8 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)
H order = 48
Hidden Units = 60
Add zero num = 15

```

3.5 /Threenonlinear5: P = 5 , F = 0.1 , r=8, [add_zero = 24](#)

```

Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 400 ,
DropPeriod = 5 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)
H order = 48 ,related num = 8
Hidden Units = 60
Add zero num = 24

```

3.6 /Threenonlinear6: P = 6 , F = 0.1 , r=8, [add_zero = 24](#)

```

Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 400 ,
DropPeriod = 6 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)
H order = 48 ,related num = 8
Hidden Units = 60
Add zero num = 24

```

3.7 /Threenonlinear7: P = 7 , F = 0.1 , r=8, [add_zero = 24](#)

```

Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 400 ,
DropPeriod = 7 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)
H order = 48 ,related num = 8
Hidden Units = 60
Add zero num = 24

```

3.8 /Threenonlinear8: P = 8 , F = 0.1 , r=8, [add_zero = 24](#)

```

Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 400 ,
DropPeriod = 8 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)
H order = 48 ,related num = 8
Hidden Units = 60
Add zero num = 24

```

3.9 /Threenonlinear9: P = 9 , F = 0.1 , r=8, [add_zero = 24](#)

```

Threenonlinear ,
ini learningRate = 1.000000e-02 ,
min batch size = 400 ,
DropPeriod = 9 ,
DropFactor = 0.100000 ,
amp begin = 2 , amp end = 26 , amp step = 1
data_num = 100
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)
H order = 48 ,related num = 8
Hidden Units = 60
Add zero num = 24

```

3.10 /Threenonlinear10: P = 10 , F = 0.1 , r=8, [add_zero = 24](#)

```
Threenonlinear ,  
ini learningRate = 1.000000e-02 ,  
min batch size = 400 ,  
DropPeriod = 10 ,  
DropFactor = 0.100000 ,  
amp begin = 2 , amp end = 26 , amp step = 1  
data_num = 100  
validationFrequency is floor(numel(xTrain)/miniBatchSize/4)  
H order = 48 ,related num = 8  
Hidden Units = 60  
Add zero num = 24
```