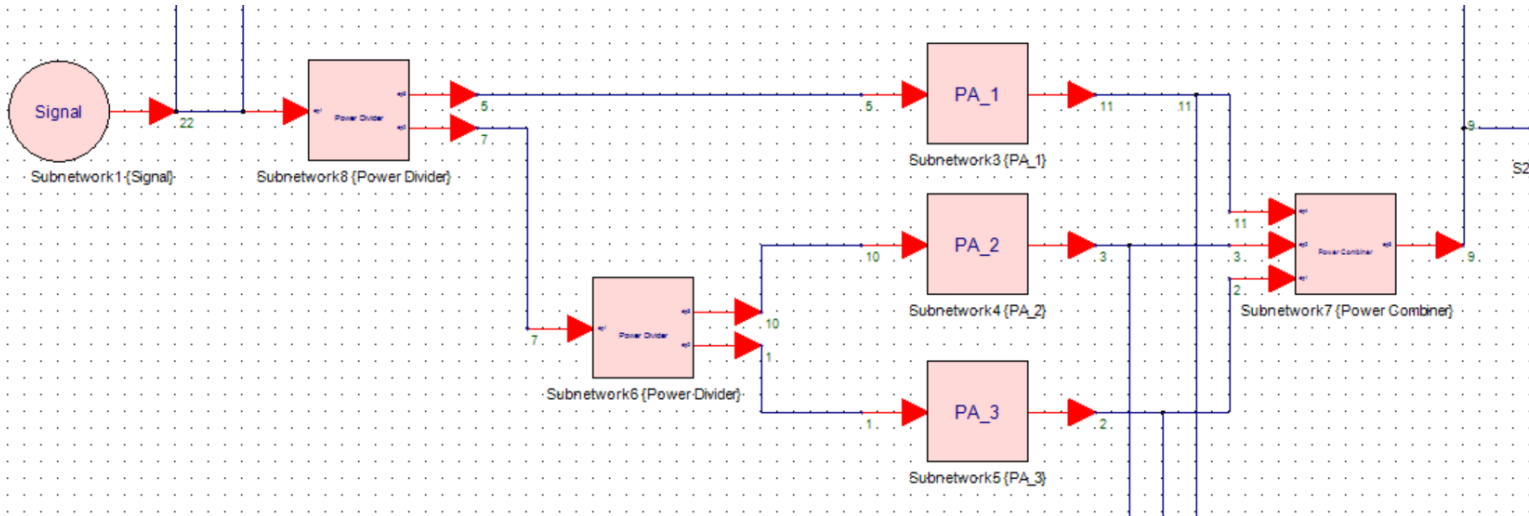


# 整体优化



% 运行前请先将子文件夹添加到路径  
clc;

## -----Model A Train-----

### 读取数据

```
clear;
% 读取文件并处理数据
[Data_matrix,dim]=ReadFile_Train();
% 根据需要选取原始数据
Sys_input = Data_matrix(:,1) + 1i * Data_matrix(:,2);
Sys_output = Data_matrix(:,end-1) + 1i * Data_matrix(:,end);
% 参数需要保持一致
M = 4;
path_temp = 'Method_A\Whole\FNN_temp.mat';
path_opt = 'Method_A\Whole\FNN_opt.mat';
Train_flag = 1;
times = 10;
nmse_min = 0;
```

### 【训练】整体训练


```
for ii=1:times
    % 模型训练
    if (Train_flag)
        [model,tr] = Train_sim(M,Sys_input,Sys_output,[36,24],0);
        fprintf('本次训练迭代了 %f 轮\n', tr.num_epochs);
        save(path_temp, 'model');
    end
    % 模型精度
    [model_input,model_output] = pre_train(M,Sys_input,Sys_output,dim);
```

```
data = load(path_temp);
DPD_output_m = sim(data.model,model_input);
DPD_output = DPD_output_m(1,:)+1j*DPD_output_m(2,:);
nmse = NMSE_dB(model_output(1,:)+1j*model_output(2,:), DPD_output);
fprintf('模型的拟合精度 NMSE 为 %f dB\n', nmse);
```

本次训练迭代了 65.000000 轮  
模型的拟合精度 NMSE 为 -53.809115 dB  
本次训练迭代了 25.000000 轮  
模型的拟合精度 NMSE 为 -53.993540 dB  
本次训练迭代了 117.000000 轮  
模型的拟合精度 NMSE 为 -54.045184 dB  
本次训练迭代了 35.000000 轮  
模型的拟合精度 NMSE 为 -53.573718 dB  
本次训练迭代了 172.000000 轮  
模型的拟合精度 NMSE 为 -53.814949 dB  
本次训练迭代了 17.000000 轮  
模型的拟合精度 NMSE 为 -53.459928 dB  
本次训练迭代了 16.000000 轮  
模型的拟合精度 NMSE 为 -53.965848 dB  
本次训练迭代了 76.000000 轮  
模型的拟合精度 NMSE 为 -54.295016 dB  
本次训练迭代了 63.000000 轮  
模型的拟合精度 NMSE 为 -53.512342 dB

网络图

## 训练结果

训练结束: 满足验证条件 

## 训练进度

单位	初始值	当前值	目标值	
轮	0	22	1000	▲
历时	-	00:00:04	-	
性能	5.75	1.29e-06	0	
梯度	23.8	0.000656	1e-07	
Mu	0.001	1e-08	1e+10	
验证检查	0	4	4	▼

## 训练算法

数据划分: 索引 divideind

训练: 莱文贝格-马夸特 trainlm

性能: 均方误差 mse

计算: MEX

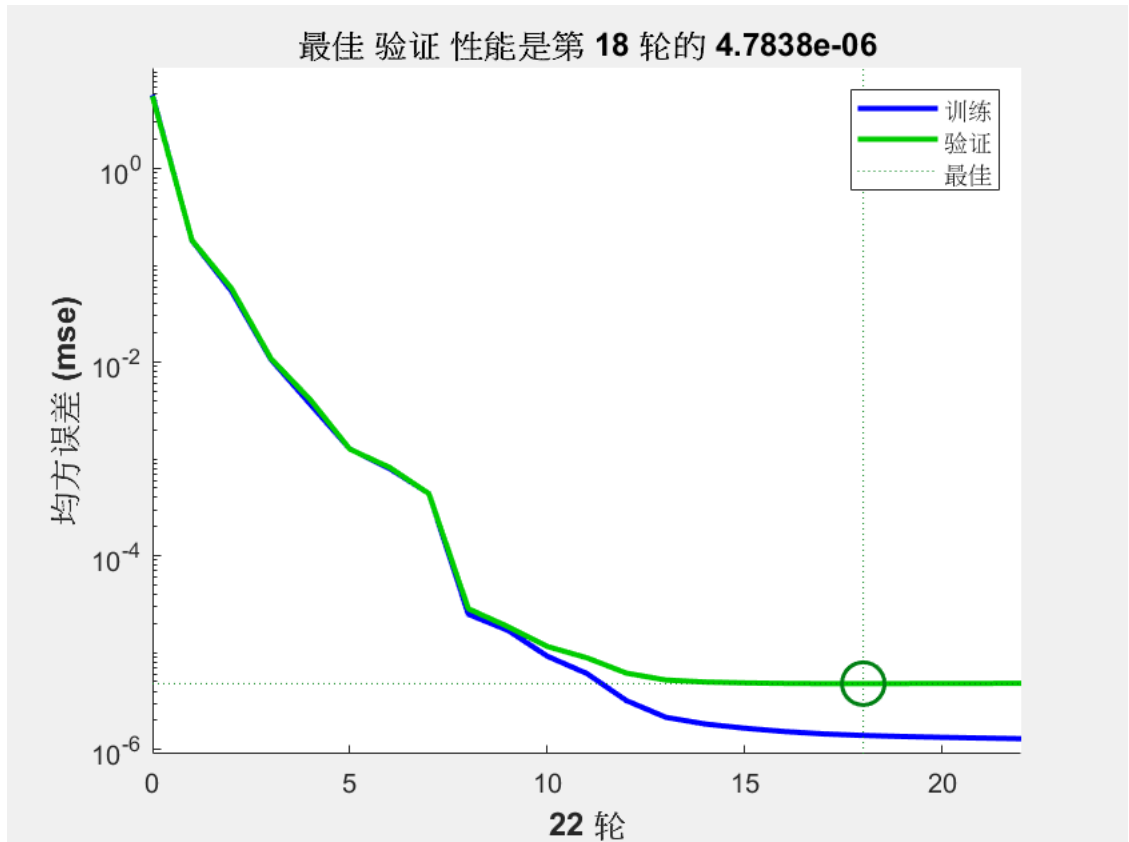
## 训练图

性能

训练状态

误差直方图

回归



本次训练迭代了 22.000000 轮  
模型的拟合精度 NMSE 为 -53.676484 dB

## Model Test

```
% 模型仿真
Sys_input_DPD=DPD_sim(Sys_input,M,path_temp);
[A1_input,A23] = Power_Divider(Sys_input_DPD);
[A2_input,A3_input] = Power_Divider(A23);
A1_output = PA_1(A1_input);
A2_output = PA_2(A2_input);
A3_output = PA_3(A3_input);
Sys_output_sim = Power_Combiner(A1_output,A2_output,A3_output);
% 性能评估
% Plot_AM(Sys_input,Sys_output_sim,'AM Behavior of Model A System in Train
Set','with DPD');
nmse_sim= NMSE_dB(Sys_input,Sys_output_sim);
fprintf('使用 Model A, 系统的 NMSE(Train)仿真结果为 %f dB\n', nmse_sim);
if(nmse_sim<nmse_min)
    nmse_min=nmse_sim;
    save(path_opt, 'model');
end
end
```

使用 Model A, 系统的 NMSE(Train)仿真结果为 -48.974995 dB  
使用 Model A, 系统的 NMSE(Train)仿真结果为 -49.359189 dB  
使用 Model A, 系统的 NMSE(Train)仿真结果为 -48.718944 dB

使用 Model A, 系统的 NMSE(Train)仿真结果为 -49.517500 dB  
使用 Model A, 系统的 NMSE(Train)仿真结果为 -49.044850 dB  
使用 Model A, 系统的 NMSE(Train)仿真结果为 -48.483085 dB  
使用 Model A, 系统的 NMSE(Train)仿真结果为 -49.259174 dB  
使用 Model A, 系统的 NMSE(Train)仿真结果为 -49.036481 dB  
使用 Model A, 系统的 NMSE(Train)仿真结果为 -49.591771 dB  
使用 Model A, 系统的 NMSE(Train)仿真结果为 -48.799567 dB

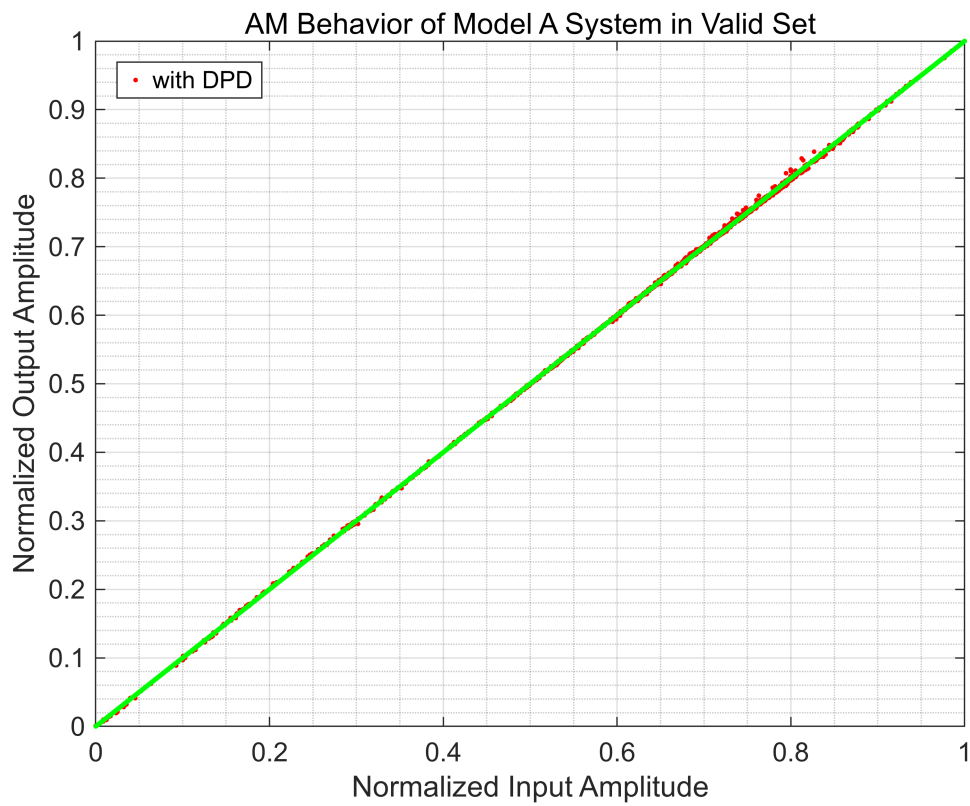
## -----Model A Valid-----

### 读取数据

```
clear;  
% 读取文件并处理数据  
[Data_matrix,dim]=ReadFile_Valid();  
% 根据需要选取原始数据  
Sys_input = Data_matrix(:,1) + 1i * Data_matrix(:,2);  
% 参数需要保持一致  
M = 4;  
path = 'Method_A\Whole\FNN_opt.mat';
```

### Model Test

```
% 模型仿真  
Sys_input_DPD=DPD_sim(Sys_input,M,path);  
[A1_input,A23] = Power_Divider(Sys_input_DPD);  
[A2_input,A3_input] = Power_Divider(A23);  
A1_output = PA_1(A1_input);  
A2_output = PA_2(A2_input);  
A3_output = PA_3(A3_input);  
Sys_output_sim = Power_Combiner(A1_output,A2_output,A3_output);  
% 由于 PA 和 DPD 的延时, 所以少算 (4+M) 个值  
Sys_input = Sys_input(5+M:end);  
Sys_output_sim = Sys_output_sim(5+M:end);  
% 性能评估  
Plot_AM(Sys_input,Sys_output_sim,'AM Behavior of Model A System in Valid Set','with  
DPD');
```



```
nmse_sim= NMSE_dB(Sys_input,Sys_output_sim);  
fprintf('使用 Model A, 系统的 NMSE(Valid)仿真结果为 %f dB\n', nmse_sim);
```

使用 Model A, 系统的 NMSE(Valid)仿真结果为 -50.067245 dB