|  |  |
| --- | --- |
|  | School of Information Science and Engineering  射频电路建模与CAD方法（双语）  编号:B0433111 |

Final Lab: DPD Extraction and Validation

Student Number: 04022212, 04022114

Name: 钟源, 贾岩森

·Report

# Introduction

In this laboratory assignment, we are given a 3-way PA with memory effects. We aim to build a DPD model for given PA circuit using knowledge learned from the course. To achieve this goal, We have tried a lot of methods, including Feedforward Neural Network (FNN), Recurrent Neural Network(RNN) and Support Vector Machine (SVM). In the Networks we design, we have combined FCN, CNN, LSTM, Bi-LSTM, GRU and other useful networks. 【未完】. These results will be shown in the upcoming section.

# The selection of methods

## 2.1 Fully Connected Network (FCN)

首先，我们使用MATLAB中的Deep Network Designer设计常见的神经网络。

作为神经网络中最基础的网络结构——Fully Connected Network具有全连接性、无空间假设、层级堆叠等特点，并且能捕捉输入特征之间的复杂全局关系、理论上可逼近任意连续函数，非常适合DPD输入数据的处理。

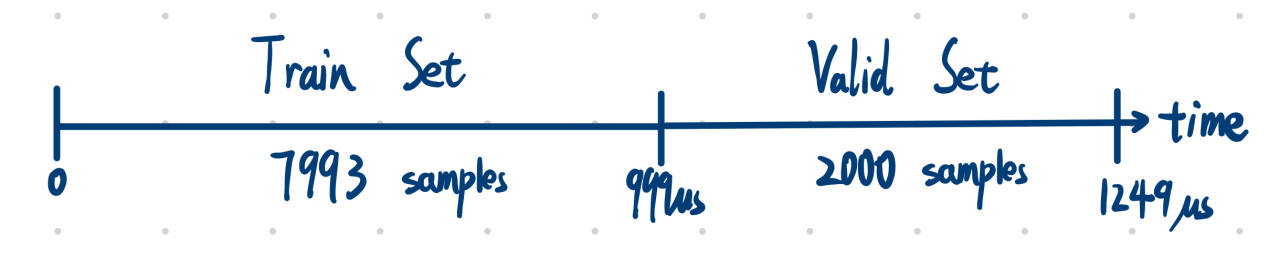
我们选择adam作为优化算法，构建了一个简单的完全由全连接层组成的神经网络，如下：



设置Memory Depth(M)=5，最大训练次数为15000轮（在11320轮左右满足验证条件停止），其拟合精度结果为-35.315dB。



为了更好地衡量网络模型的泛化能力，我们设置两个数据集来考察其是否能很好地适用于原电路，两个数据集的选取如下所示。开始模拟的0-999us等间隔采样7993个样本点，作为数据集1，取名Train Set（由于取得的数据来自该时间段）；之后的250us里等间隔采样2000个样本点，作为数据集2，取名Valid Set。要注意的是Train Set并不真正是训练模型的数据集，因为模型的训练集输入数据是PA的输出数据，输出数据是PA的输入数据，而当DPD应用于电路时，其输入是PA的原输入数据。



经过验证，全FCN网络在Train Set上的NMSE为-35.276dB，在Valid Set上的NMSE为

-35.324dB。





得到的AM-AM图像如下：

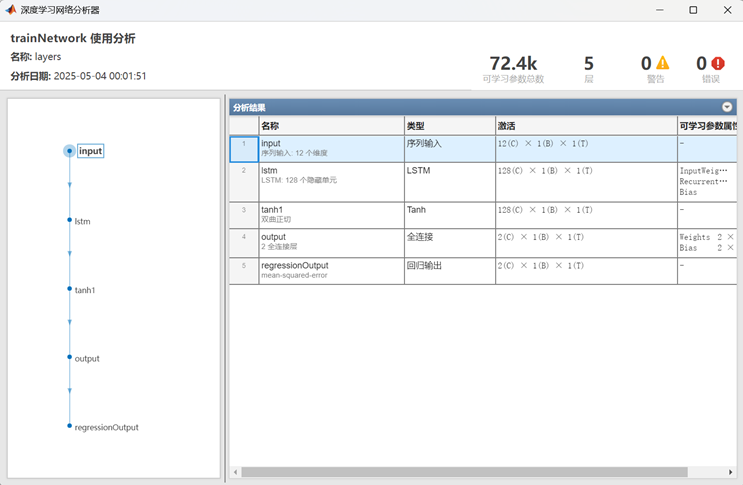
|  |  |
| --- | --- |
|  |  |

可以看到，相比于无DPD时，NMSE下降了大约10dB，即缩小为原来的1/10，可见使用DPD的巨大作用，并且可见FCN网络的潜能。

## 2.2 Long Short-Term Memory (LSTM)

LSTM适用于时间序列数据的处理和预测，适用于大数据环境。LSTM可以有效地捕获和记忆序列数据中的长期依赖关系，十分适用于解决PA非线性问题，因此而广泛应用于DPD优化领域。

依旧选择adam作为优化算法，设置Memory Depth(M)=5，最大训练次数为4000轮（在900轮左右满足验证条件停止），构建了一层LSTM网络（如下所示），其拟合NMSE结果为-42.341dB。



经过验证，LSTM网络在Train Set上的NMSE为-38.654dB，在Valid Set上的NMSE为

-38.929dB。





得到的AM-AM图像如下：

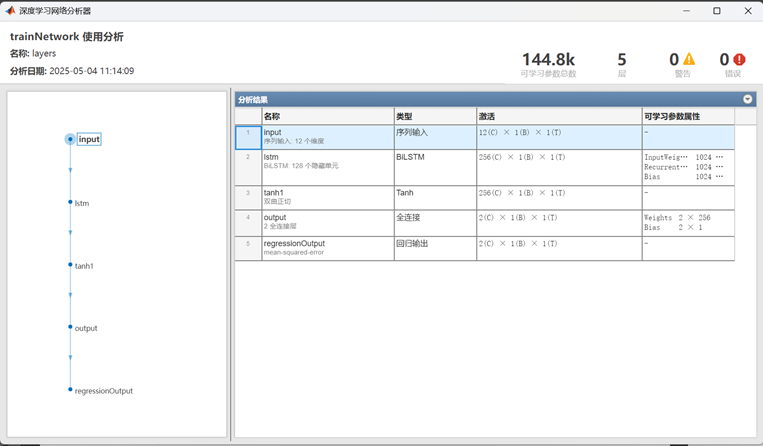
|  |  |
| --- | --- |
|  |  |

可以看到，虽然LSTM的模型拟合精度更高了，但是它真正用于电路时表现反而下降了大约4dB，这说明出现了一定程度的过拟合现象，在后面几个同样复杂的网络中可以看到，越是复杂的网络可能出现过拟合，这启示我们不能无止境地增加网络复杂度，否则模型性能反而可能下降。

## 2.3 Bi-directional Long Short-Term Memory (Bi-LSTM)

由于Bi-LSTM可以更好地捕获双向序列依赖，我们也尝试了这种网络。

依旧选择adam作为优化算法，设置Memory Depth(M)=5，最大训练次数为4000轮（在840轮左右满足验证条件停止），构建了一层Bi-LSTM网络（如下所示），其拟合NMSE结果为-44.254dB。



经过验证，Bi-LSTM网络在Train Set上的NMSE为-33.932dB，在Valid Set上的NMSE为

-33.205dB。





得到的AM-AM图像如下：

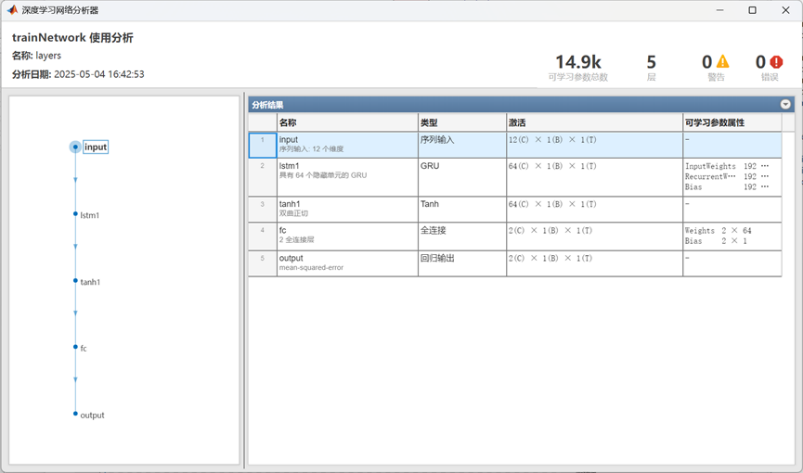
|  |  |
| --- | --- |
|  |  |

可以看到，Bi-LSTM的网络更加复杂（学习参数超过144k！），其过拟合趋势更加严重，虽然模型拟合NMSE达到了-44dB左右，但其真正应用时NMSE却不到-34dB（劣于FCN），并且AM图像中出现了离群点，这启示我们要寻找更简单的网络。

## 2.4 Gated Recurrent Unit (GRU)

GRU一般被视为LSTM的简化版本，其要训练的参数比LSTM少1/3，我们同样尝试了该网络，并且基于防止过拟合的思路，我们尝试减少隐藏层数量，发现得到了不错的效果。

依旧选择adam作为优化算法，设置Memory Depth(M)=5，最大训练次数为4000轮（在1640轮左右满足验证条件停止），构建了一层Bi-LSTM网络（如下所示），其拟合NMSE结果为-47.393dB。



经过验证，GRU网络在Train Set上的NMSE为-43.417dB，在Valid Set上的NMSE为

-43.694dB。





得到的AM-AM图像如下：

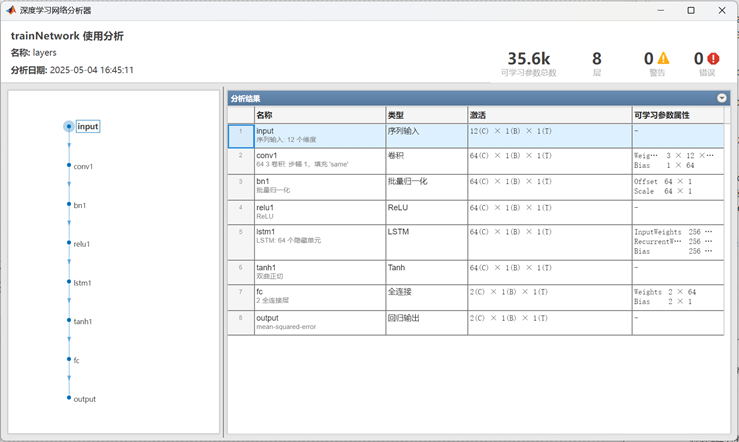
|  |  |
| --- | --- |
|  |  |

可以看到，虽然GRU也存在着过拟合现象，但已经得到改善，其应用时的NMSE可以达到-43dB以上了，性能提升非常明显。

## 2.5 CNN-LSTM

在查阅资料时，我们发现由于1D-CNN能够捕捉模型非线性失真并降低模型复杂度，故经常与LSTM一起出现。我们也试着将CNN与LSTM结合，生成混合型网络。

依旧选择adam作为优化算法，设置Memory Depth(M)=5，最大训练次数为4000轮（在1160轮左右满足验证条件停止），构建了一层CNN-LSTM网络（如下所示），其拟合NMSE结果为-38.850dB。



经过验证，CNN-LSTM网络在Train Set上的NMSE为-36.615dB，在Valid Set上的NMSE为-36.823dB。





得到的AM-AM图像如下：

|  |  |
| --- | --- |
|  |  |

可见引入CNN确实有助于降低模型复杂度，有效改善了过拟合，而且训练时间非常短。不过可惜的是该模型本身的性能并太强，可以应用到对DPD精度不太高但需要快速建模的场景。

## 2.6 Multi-layer FCN

经过一定的探索后，我们发现，一般而言，在不过拟合的前提下，增加FCN的隐藏层数和隐藏层数量能显著增加DPD的优化效果，基于此我们提出了使用adam优化算法的多层结构全FCN网络，如下：



设置Memory Depth(M)=5，初始学习率为0.005，最大训练次数为15000（在12360轮左右满足验证条件停止），其拟合NMSE为-46.003dB。经过验证，多层FCN网络在Train Set上的NMSE为-44.859dB，在Valid Set上的NMSE为-45.179dB。

得到的AM-AM图像如下：

|  |  |
| --- | --- |
|  |  |

## 2.7 Temporal Convolutional Network (TCN)

综合训练速度和模型表现，Multi-layer FCN基本上就是我们探索出的最优结果，不过小组成员在课余时间还探索了其他的几种网络，虽然模型性能并不算十分理想，但也可以作为参考，展示如下。

TCN是一种专门为处理时序数据设计的卷积神经网络，旨在替代传统RNN（如LSTM、GRU）在序列建模任务中的应用。其核心特点是通过改进卷积操作，实现对时间序列的高效建模。对比传统的RNN，TCN可并行计算（同时处理整个序列），并且通过多层级扩张卷积，有效建模远距离时序关系，而且训练更稳定。

选择adam作为优化算法，设置Memory Depth(M)=4，最大训练次数为10000轮（在6000轮左右满足验证条件停止），构建MATLAB中系统默认提供的TCN网络（如下所示），其拟合NMSE结果为-42.507dB。



经过验证，CNN-LSTM网络在Train Set上的NMSE为-38.627dB，在Valid Set上的NMSE为-37.308dB。





得到的AM-AM图像如下：

|  |  |
| --- | --- |
|  |  |

可以看到，该模型的结果较不如预期，一方面，这是因为模型设置时MATLAB工具箱中自带的，并未对任务作专门调整（也是因为不知道怎么调整）；另一方面，也是由于参数量较大，导致训练成本较高，小组成员没有充分的时间来调整超参数以达到更好的效果。另外关注到该模型也有较严重的过拟合问题。

## 2.8 Transformer

Transformer与RNN逐时间步计算不同，它通过矩阵运算一次性处理整个序列，大幅提升训练速度。其核心为自注意力机制，即直接建模序列中任意两个元素的关系，无论距离多远，天然解决长距离依赖问题，并且可以由多个相同结构的编码器/解码器块堆叠，易于扩展模型容量。

仍然选择adam作为优化算法，设置Memory Depth(M)=4，最大训练次数为8000轮（在1000轮左右满足验证条件停止），构建Transformer架构网络（如下所示），其拟合NMSE结果为-28.096dB。



经过验证，Transformer架构网络在Train Set上的NMSE为-25.923dB，在Valid Set上的NMSE为-26.051dB。





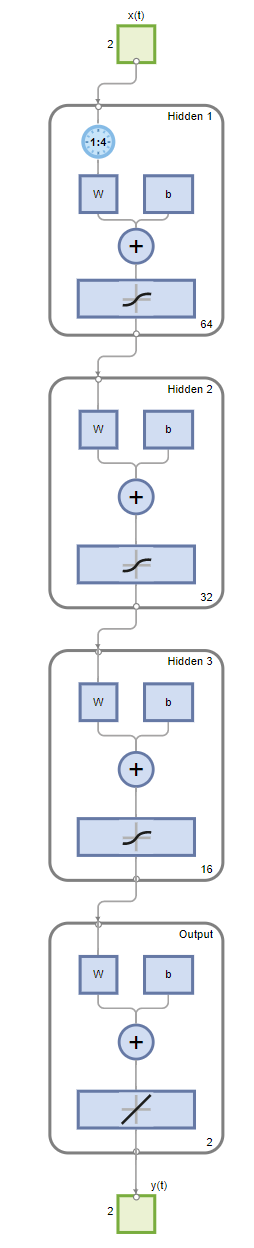
可以看到，Transformer架构在DPD表现场景上较差，是众多模型中唯一一个做到负优化的。造成训练网络失败的核心原因主要是，由于时间和知识都十分有限，小组成员的transformer架构设置不够合理，错误使用了某些网络或激活函数导致了这样的结果。其次可以看到，一般transformer常用于较复杂的问题和数据量较大的场景，故而可能不太适合DPD优化场景。

## 2.9 Time Delay Net

时间延迟神经网络是 MATLAB 中专门用于处理时间序列预测的神经网络结构，其核心思想是通过引入时间延迟捕捉序列中的时序依赖关系，适用于单变量或多变量时间序列建模。

选择LM作为优化算法，设置Memory Depth(M)=4，最大训练次数为4000轮（在600轮左右满足验证条件停止），构建时间延迟神经网络（如下所示），其拟合NMSE结果为

-44.655dB。



经过验证，时间延迟神经网络在Train Set上的NMSE为-30.378dB，在Valid Set上的NMSE为-30.719dB。





得到的AM-AM图像如下：

|  |  |
| --- | --- |
|  |  |

## 2.10 Volterra

出于好奇心，我们还尝试了之前实验中常用的Volterra模型。

核心函数如下所示：

function DPD\_coefficient=Train\_Voltrra(M,N,dim,input,output)

output=output/max(abs(output));

input=input/max(abs(input));

DPD\_input\_matrix = zeros(dim, (M+1)\*N);

for i=1:M+1

    for j= 1:N

        DPD\_input\_matrix(:,N\*(i-1)+j) = Delay(output,i-1).\*((Delay(output,i-1)).^(2\*j-2));

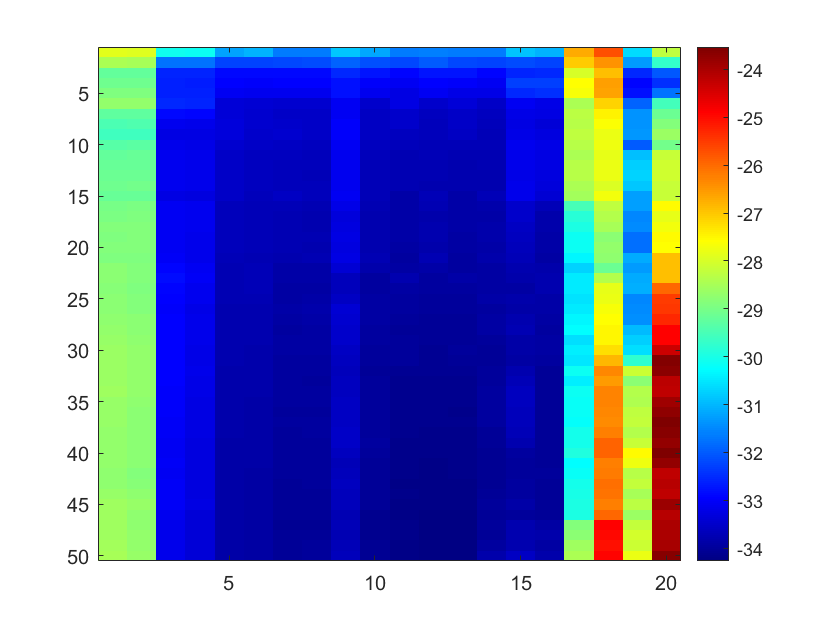
    end

end

DPD\_coefficient=DPD\_input\_matrix\input(:);

end

为了确定最优的参数M和N（阶数），我们在M[1,50]和N[1,20]上进行了参数搜索，得到的热力图如下：



得到的结论是当N=13，M越大则性能越好，当然M过大会导致过拟合。最后我们取N=13，M=100，得到的拟合NMSE为-34.707dB。经过验证，Volterra模型在Train Set上的NMSE为-32.241dB，在Valid Set上的NMSE为-32.349dB。

得到的AM-AM图像如下：

|  |  |
| --- | --- |
|  |  |

可见Volterra模型的精度有限，遇到较复杂的模型时很难有效优化。

## 2.11 Comparison and Summary

我们对比了不同方法提高系统线性化程度的性能，得到了下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 拟合NMSE(dB) | Train Set  NMSE(dB) | Valid Set  NMSE(dB) |
| Original | --- | -26.743 | -26.794 |
| FCN(M=5) | -35.315 | -35.276 | -35.324 |
| LSTM(M=5) | -42.341 | -38.654 | -38.929 |
| Bi-LSTM(M=5) | -44.254 | -33.932 | -33.205 |
| GRU(M=5) | -47.393 | -43.417 | -43.694 |
| CNN-LSTM(M=5) | -38.850 | -36.615 | -36.823 |
| Multi-layer FCN (M=5) | -46.003 | -44.859 | -45.179 |
| TCN(M=4) | -42.507 | -38.627 | -37.308 |
| Transformer(M=4) |  |  |  |
| Timedelay(M=4) | -44.655 | -30.378 | -30.719 |
| Volterra(M=100, N=13) | -34.707 | -32.241 | -32.349 |

注：所有的网络中，仅Timedelay使用了Levenberg-Marquardt优化算法，其他网络均使用Adam优化算法。

对比之后发现，Multi-layer FCN的性能表现最好，过拟合程度最浅（在表现较好的模型中），而且训练时间也适中。于是我们决定使用Multi-layer FCN作为接下来建立更复杂模型的主要方法，其他方法作为补充。

# The selection of optimization algorithm

在进一步的摸索中，我们发现——选择不同的优化算法，带来的结果也会相差较大。我们将以Multi-layer FCN为例，探究不同的优化算法对DPD建模带来的影响。

## 3.1 Adaptive Moment Estimation (adam)

Adam是神经网络优化中常用的自适应学习率优化算法，它结合了动量法（Momentum）和RMSprop算法的优点：既可以通过指数加权平均积累历史梯度方向，加速收敛；又对不同参数自适应调整学习率，缓解梯度消失/爆炸问题，在数据量大、参数多的场景中表现优异。

使用adam优化算法的结果同2.7，不再赘述。

## 3.2 Stochastic Gradient Descent with Momentum (sgdm)

sgdm为经典的随机梯度下降（SGD）结合动量（Momentum），通过累积历史梯度方向加速收敛，减少震荡。动量因子可以理解为对过去更新方向的“记忆程度”，较高的动量因子会使优化过程在一定程度上保持之前的方向，避免在优化过程中频繁地改变方向，从而加快收敛速度。

我们选择sgdm作为优化算法，设置Memory Depth(M)=5，学习率由0.005调整为0.01，动量因子取0.9，最大训练次数为5000轮，构建了同2.7的Multi-layer FCN网络，其拟合NMSE结果为-42.238dB。

经过验证，使用sgdm作为优化算法的Multi-layer FCN网络在Train Set上的NMSE为

-41.339dB，在Valid Set上的NMSE为-41.572dB。





得到的AM-AM图像如下：

|  |  |
| --- | --- |
|  |  |

可以看到，使用sgdm时，其迭代更快，需要的迭代次数更少，但相应地，迭代过程更不稳定，需要宽松其验证条件，避免提前结束训练。

## 3.3 Root-Mean-Square Propagation (rmsprop)

RMSProp 是一种自适应学习率优化算法，主要思想是对梯度进行归一化，以解决梯度下降过程中学习率固定带来的问题。其核心是通过对梯度的平方进行指数加权移动平均来调整学习率。

我们选择rmsprop作为优化算法，设置Memory Depth(M)=5，学习率由0.005调整为0.001，动量因子取0.9，最大训练次数为15000轮（在11900轮左右满足验证条件停止），构建了同2.7的Multi-layer FCN网络，其拟合NMSE结果为-31.994dB。

经过验证，使用rmsprop作为优化算法的Multi-layer FCN网络在Train Set上的NMSE为

-33.357dB，在Valid Set上的NMSE为-33.364dB。





得到的AM-AM图像如下：

|  |  |
| --- | --- |
|  |  |

可以看到，使用rmsprop时，其迭代更慢，需要的迭代次数也更多，在相似条件下，比起其他优化算法需要更多的资源和时间，同时效果也并不理想，不是一个很好的选择。

## 3.4 Levenberg-Marquardt (trainlm)

Levenberg - Marquardt 算法是一种用于解决非线性最小二乘问题的优化算法。它结合了梯度下降法和高斯 - 牛顿法的优点。当参数离最优解较远时，它表现出类似梯度下降的特性，能够稳定地朝着误差减小的方向前进；而当参数接近最优解时，它又表现出类似高斯 - 牛顿法的特性，可以快速收敛。且Levenberg - Marquardt 算法在处理中小规模的神经网络训练任务时非常有效，尤其是在函数逼近、模式识别和分类等任务中，恰好适合DPD的处理场景。

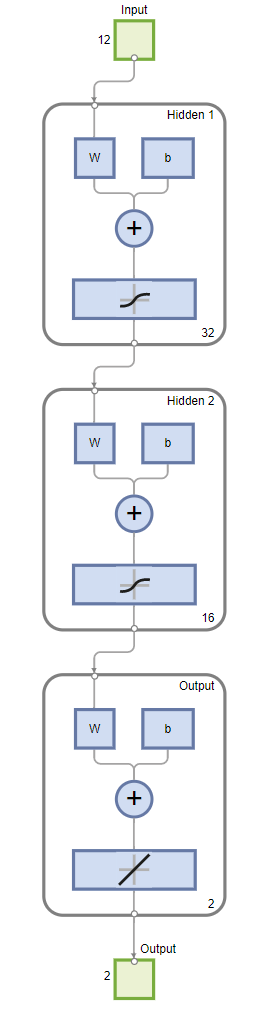
由于MATLAB的Deep Network Designer不支持使用trainlm作为优化算法，故而需切换到传统神经网络工具箱。

我们选择trainlm作为优化算法，设置Memory Depth(M)=5，无需设置学习率，构建了同2.7的Multi-layer FCN网络，其拟合NMSE结果为-54.500dB。经过验证，使用trainlm作为优化算法的Multi-layer FCN网络在Train Set上的NMSE为-45.662dB，在Valid Set上的NMSE为-46.046dB。





可以看到，trainlm算法迭代非常快，性能优异，也导致了其过拟合问题较为严重。于是我们减少输入数据量，并精简隐藏层数量，设计网络如下：



其拟合NMSE结果为-54.067dB。经过验证，使用trainlm作为优化算法的Multi-layer FCN网络在Train Set上的NMSE为-49.004dB，在Valid Set上的NMSE为-46.381dB。





得到的AM-AM图像如下：

|  |  |
| --- | --- |
|  |  |

相比于其他优化算法，trainlm稳定性很好，训练时间更快，同时性能表现也更好。唯一可能比较遗憾的是，在其训练过程中使用到了雅各比矩阵，并不支持在GPU上计算。

## 3.5 Scaled Conjugate Gradient (trainscg)

MATLAB传统神经网络工具箱还支持缩放共轭梯度法和贝叶斯正则化等优化算法，我们将进一步探索。

Scaled Conjugate Gradient算法是一种用于训练神经网络的优化算法，由 Martin F. Moller 于1993年提出。它结合了共轭梯度法的高效性和自适应步长调整策略，特别适合解决中等规模的非线性优化问题（如神经网络的参数训练）。

选择trainscg作为优化算法，设置Memory Depth(M)=5，无需设置学习率，构建了同3.3的simple Multi-layer FCN网络，其拟合NMSE结果为-41.556dB。经过验证，使用trainlm作为优化算法的Multi-layer FCN网络在Train Set上的NMSE为-41.540dB，在Valid Set上的NMSE为-41.833dB。





得到的AM-AM图像如下：

|  |  |
| --- | --- |
|  |  |

## 3.6 Bayesian Regularization (trainbr)

贝叶斯正则化算法是神经网络训练中用于防止过拟合的一种高阶优化方法，结合了Levenberg-Marquardt算法与贝叶斯正则化技术。能够通过正则化抑制模型复杂度，提升泛化性能，防止过拟合；而且适用于小规模数据，在训练数据较少时仍能保持较高的预测精度。这与我们DPD的优化场景十分吻合。

选择trainscg作为优化算法，设置Memory Depth(M)=4，无需设置学习率，构建了同3.3的simple Multi-layer FCN网络，其拟合NMSE结果为-52.952dB。经过验证，使用trainlm作为优化算法的Multi-layer FCN网络在Train Set上的NMSE为-49.211dB，在Valid Set上的NMSE为-49.750dB。





得到的AM-AM图像如下：

|  |  |
| --- | --- |
|  |  |

可以看到贝叶斯正则化算法比LM算法的性能表现稍好一点点，且过拟合问题得到缓解，美中不足的是其训练时间比LM算法长不少。

## 3.7 Comparison and Summary

|  |  |  |  |
| --- | --- | --- | --- |
| 优化算法 | 拟合NMSE(dB) | Train Set  NMSE(dB) | Valid Set  NMSE(dB) |
| adam | -46.003 | -44.859 | -45.179 |
| sgdm | -42.238 | -41.339 | -41.572 |
| rmsprop | -31.994 | -33.357 | -33.364 |
| trainlm with  complex network | -54.500 | -45.662 | -46.046 |
| trainlm with  simple network | -54.067 | -49.004 | -49.381 |
| trainscg with  simple network | -41.556 | -41.540 | -41.833 |
| trainbr with  simple network | -52.952 | -49.211 | -49.750 |

# 4. Other Models

# 5. The Best Model

# 6. Conclusion

**7. A brief summary of this experiment**

**7.1 Member 1：04022212钟源**

In this lab, my partner and I use Neural Network and Volterra model to build a DPD model. We’ve tried FCN, LSTM, Bi-LSTM, GRU, CNN-LSTM, TCN, Transformer and so on. After so many tries, we finally realized that “the less is the more”: simple network might bring better output. In fact, adjusting the parameters of a network is a very painful process, because when you solve one problem, another problem emerges, which will affect performance of the entire model. Anyway, this lab assignment is a valuable experience for us to learn system modelling and neural network.

**7.2 Member 2：04022114贾岩森**