

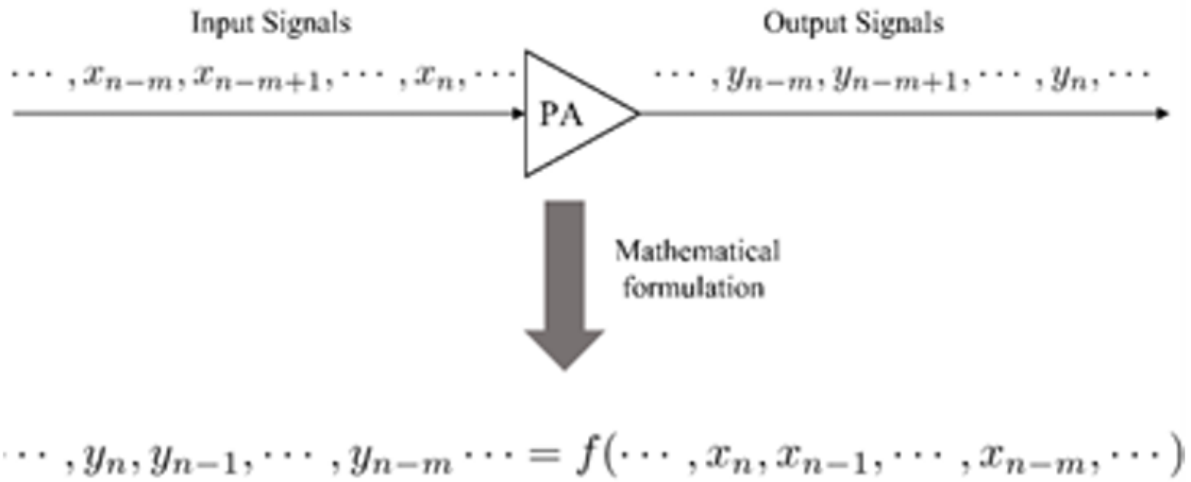
Mathematics for Computer Science Project 2

赵楷越 522031910803

2024 年 5 月 17 日

1 问题定义的描述和分析

任务 1 的问题主要描述了一个带有时间记忆特性的功率放大器。这个功率放大器会根据先前记忆深度 m 个输入状态、来决定当前时刻输出的结果（如下图所示）。我们所拥有的数据是在一个时间序列中、某功率放大器的一个输入信号的序列和其对应的输出信号序列。同时，其数据都以复数形式存储。要求建立模型、该模型能够对该功率放大器的另一个输入信号序列，作出对应的输出信号序列的预测。

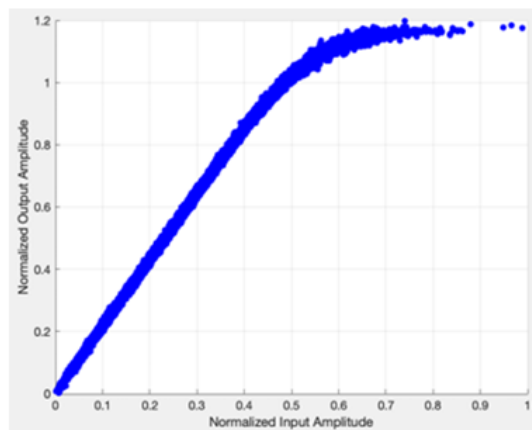
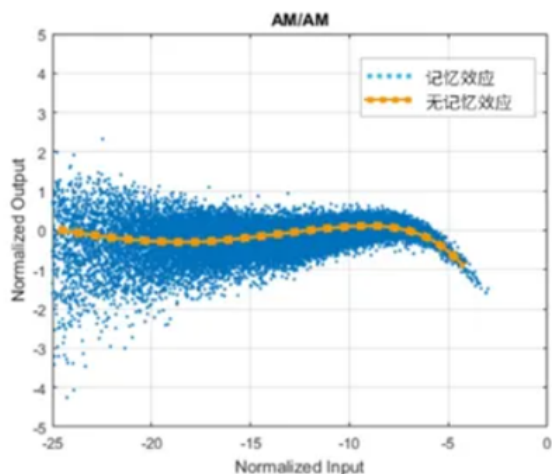


根据功率放大器的时间记忆特性、一个真实的功率放大器的输出和输入的关系可以表示为如下公式、其中 y_n 是根据 $x_n, x_{n-1}, \dots, x_{n-m}$ 的序列所决定的、其中 m 代表的是功率放大器的记忆深度。本次实验便基于如下公式、同时通过神经网络的方法、对功率放大器的输出信号序列作出了有效预测。

$$y_n = f(x_n, x_{n-1}, \dots, x_{n-m})$$

2 数据预处理过程的介绍

通过查阅资料 [1]，我了解到了功率放大器 PA 的记忆效应对其信号的放大所产生的影响。如下图所示，未发生记忆效应时，AM/AM 曲线为一条平滑曲线，代表同样输入功率时不同时刻 PA 的特性保持相同。发生记忆效应时，曲线表现出一组发散分布的点，点的分散的离散与否代表了记忆效应的强弱。如果 AM/AM 曲线表现分散，表示对于一个同样的输入功率，PA 有不同的增益或相位变化。因此在现实中、PA 的状态不止以此时刻的输入有关，还与其他时刻的状态有关，即 PA 表现出了“记忆效应”。同样地、在本次实验中，根据对训练集的 AM/AM 图的有效观测，功率放大器一样具有类似的“记忆效应”。



同时，本次实验中的信号数据都是由复数形式进行存储的，因此本次实验将一个复数的实部和虚部拆解开、从而转化为一个二维向量的形式进行处理和预测。为了体现功率放大器的记忆效应对模型的影响，本次实验还将训练集的输入信号以记忆深度（memory depth）为个数一组组合后进行训练，共计 $22680 - \text{memory depth}$ 组输入信号数据；类似地，在排除掉前 memory depth 个输出信号数据后、将剩余 $22680 - \text{memory depth}$ 个输出信号数据作为可靠值进行模型的训练。类似地、我们对测试集里的输入输出信号的数据也做了同样的处理，并作了同样的分组。相关预处理训练数据的部分代码如下图所示：

```

24 # 构建输入序列（考虑记忆深度）
25 def build_input_sequence(data_real, data_imag, memory_depth):
26     input_sequence = []
27     for i in range(memory_depth, len(data_real)):
28         input_sequence.append(np.column_stack((data_real[i-memory_depth:i], data_imag[i-memory_depth:i])))
29     return np.array(input_sequence)
30
31 # 创建训练数据的输入序列和输出序列
32 train_input_sequence = build_input_sequence(train_input_real, train_input_imag, memory_depth)
33 train_output = np.column_stack((train_output_real[memory_depth:], train_output_imag[memory_depth:]))

```

此外、本次实验的过程中、我也对数据的预处理过程进行了进一步的思考。首先、我思考了是否要对输入的数据进行适当的数据清洗，不过考虑到输入和输出具有时间上的状态依赖性质，同时因为通过 AM/AM 图观测到并没有显著的离群点、因此在本次实验中，我假设所给到的信号数据都是可信的。进一步地、我又思考到是否需要信号数据进行预先的归一化，使得神经网络模型能更快地收敛，但因为原有复数数据的实部和虚部范围就在一个较小的范围中（-1 到 1），当我进行归一化操作后、预测准确度反而不如未进行归一化前的效果。

3 所使用的方法和实现过程的详细解释

3.1 所使用的方法

通过对所给实验数据的观察、在训练集的 AM/AM 图中、我发现该输入和输出幅度归一化比值的变化是先在一定范围内为线性增长、最终经过非线性的变化、饱和至某一个数值（最大输出信号幅度小于 1.2）。同时、由于功率放大器记忆效应的影响，实验数据表现出在一个理论的效应曲线旁的一组发散分布的点。考虑到训练集的数据共有 22680 组，数据量充足。因此、为了良好处理这类含有时间影响因素的非线性预测问题，本实验采取神经网络模型对该功率放大器的放大效应进行预测、其中也运用了 LSTM 层对功率放大器的记忆效应进行处理。

3.2 实现过程的详细解释

3.2.1 确定记忆深度

首先、为了确定该功率放大器的记忆深度、我查阅了对应的参考文献 [2]，其中提到、一个常规的功率放大器的记忆深度 m 可以取在 5-10 的范围之内。同时、我也通过神经网络方法、在固定网络参数（第一层为 LSTM 层、共 128 个神经元、第二层为全连接层、共 128 个神经元、激活函数为 relu、第三层为输出层）的情况下，调整功率放大器的记忆深度 $memory\ depth$ 、对 NMSE 值进行了测定。如下表所示、发现当记忆深度大于 3 时、其最终预测结果 NMSE 均表现良好。最终、为了训练效率和训练效果的平衡，选择将本功率放大器的记忆深度 $memory\ depth$ 固定为整数值 6。

memory_depth	1	2	3	4	5	6	7	8	9	10
NMSE	-11.13	-22.33	-30.90	-31.52	-32.01	-30.69	-31.29	-31.23	-31.54	-31.55
memory_depth	11	12	13	14	15	16	17	18	19	20
NMSE	-32.39	-32.10	-31.80	-30.36	-30.63	-28.70	-32.35	-32.40	-31.36	-28.44

3.2.2 构建神经网络模型

在进行功率放大器的预测时，使用了基于序列 (Sequential) 的模型，因为功率放大器的输入通常是一个时间序列，其中每个时间步的输入特征与之前的时间步有关。这个模型首先将 LSTM 作为第一层、LSTM 层可以捕捉到时间序列数据中的长期依赖关系，用于处理功率放大器的记忆特性。其中， $inputshape$ 设定为先前预先处理好的输入信号数据组的规模，并设定 $return\ sequences$ 参数为 False，只利用当前序列最后一个输出值。接着添加了两个全连接层、激活函数选取了 relu、用于解决该信号预测的非线性特征问题。最终是一个二维的输出层、输出值分别代表预测输出信号的实部与虚部。在经过多次实验测试后，采取了递减数量的神经元个数设定，因为后续层只需通过较少数量的神经元个数便能学习到先前层中已经提取出的数据特征。最终，得到最优的节点个数由前至后依次设定为递减的 128、64、32 个。对应部分代码如下图所示。

```
35  ## 构建神经网络模型
36  model = tf.keras.Sequential([
37      tf.keras.layers.LSTM(128, input_shape=(memory_depth, 2), return_sequences=False),
38      tf.keras.layers.Dense(64, activation='relu'),
39      tf.keras.layers.Dense(32, activation='relu'),
40      tf.keras.layers.Dense(2),
41  ])
```

3.2.3 自定义损失函数

损失函数中主要的部分是代表二维向量的预测值与训练值之间的均方误差 (Mean Square Error)，该值用于将预测值与真实值尽可能地拟合。同时、通过对实际数据的观察、我发现该功率放大器在现实的实际情况下、输出信号的幅度不会超过 1.2。因此，在自定义模型的损失函数中、我也加上了一个惩罚项、如果当前预测的信号幅度超过了 1.2，则添加上相应的损失，使得模型能够尽可能地学习到输出信号的幅度不能超过 1.2 这一约束条件。在同时考虑这两个因素下 (拟合、约束)，模型就会往使这个损失函数的值最小的方向进行学习，进而达到了数据预测和非线性拟合的效果。具体自定义损失函数如下图所示。

```
43  def custom_loss(y_true, y_pred):
44      mse = tf.keras.losses.MeanSquaredError()(y_true, y_pred)
45      # Split the output into real and imaginary parts
46      y_pred_real = y_pred[:, 0]
47      y_pred_imag = y_pred[:, 1]
48      # Calculate the magnitude of the complex numbers
49      y_pred_mag = tf.sqrt(tf.square(y_pred_real) + tf.square(y_pred_imag))
50      # 如果预测的幅度大于1.2，则增加惩罚
51      penalty = tf.reduce_mean(tf.square(tf.maximum(0.0, y_pred_mag - 1.2)))
52      return mse + penalty
53  # 编译模型
54  model.compile(optimizer='adam', loss=custom_loss)
55  # 训练模型
56  model.fit(train_input_sequence, train_output, epochs=50, batch_size=16)
```

同时、因为训练集和测试集都体现了同一个功率放大器的工作曲线（AM/AM 图线趋势相同，且在本模型的初步验证下 NMSE 结果良好），因此不需要担心模型过拟合的情况、因此为了使得模型对于本功率放大器的输出信号预测准确、最终采取训练参数 epochs=50，batch size=16，使模型进行充分训练。对应代码如上图所示。

最终、本模型的预测值中依旧可能存在超过幅度超过 1.2 的情况、因此、在模型输出完毕后、也对模型的预测结果进行了人为修正干预、当预测数据的幅度超过 1.2 时，规定将该复数等比例地放缩至 1.2 的幅度，以满足该功率放大器的现实情况，同时也让预测的数据更为可靠。

4 测试结果

4.1 NMSE 分析

性能通过预测信号与实际功率放大器输出信号之间的归一化均方误差（NMSE）来评估。由于信号矩阵是一个复杂矩阵，在计算 NMSE 时，需要将实部和虚部分开为实数，分别计算，然后将它们相加。计算公式如下，其中， $\hat{I}_{out}(n)$ 代表了预测信号的实部， $\hat{Q}_{out}(n)$ 代表了预测信号的虚部， $I_{out}(n)$ 代表了实际值的实部， $Q_{out}(n)$ 代表了实际值的虚部：

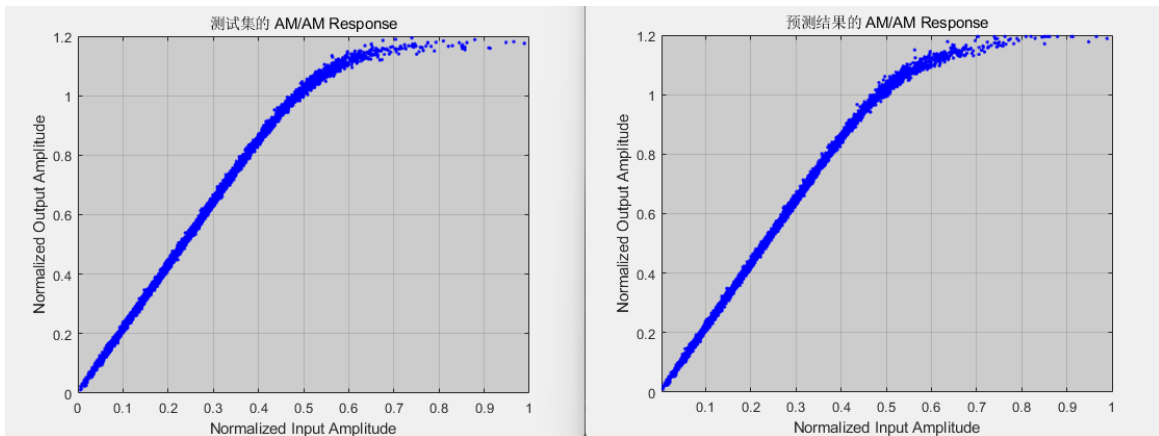
$$NMSE = 10lg(\frac{\sum_{n=1}^N(I_{out}(n) - \hat{I}_{out}(n))^2 + (Q_{out}(n) - \hat{Q}_{out}(n))^2}{\sum_{n=1}^N(I_{out}(n))^2 + (Q_{out}(n))^2})$$

在选用参数为第一层 LSTM 神经元个数为 128 个，第二次全连接层神经元个数为 64 个、第三层全连接层神经元个数为 32 个、训练参数 epochs=50、batch size=16 的情况下、程序能够正确运行得到模型在测试集上的结果如下图所示、得到了最终 NMSE 的值约为-37.93，表明该模型对于此次实验中的功率放大器的输入输出信号特性的预测性能十分出色，预测误差很小，精准度高。

```
1418/1418 ————— 3s 2ms/step - loss: 3.1928e-05
Epoch 49/50
1418/1418 ————— 3s 2ms/step - loss: 3.3297e-05
Epoch 50/50
1418/1418 ————— 5s 2ms/step - loss: 3.0520e-05
304/304 ————— 1s 2ms/step
Test NMSE: -37.93128388462286
```

4.2 归一化 AM/AM 图分析

仿照任务中对训练集绘制出的 AM/AM 图，我同样对测试集以及我的预测结果画出了对应的 AM/AM 图（测试集为左图、预测结果为右图），结果如下图所示。该模型的预测结果的 AM/AM 图与测试集的 AM/AM 图趋势相近、分布类似。因此、本模型除了在 NMSE 上表现的性能良好，在其输出信号幅度的预测上也较为准确。得出结论：本模型在预测该功率放大器输出信号的应用上表现十分良好。



5 参考资料

[1] 知乎文章 5G PA “记忆效应” 的现象、形成与消除

[2]F. M. Ghannouchi and O. Hammi, "Behavioral modeling and predistortion," in IEEE Microwave Magazine, vol. 10, no. 7, pp. 52-64, Dec. 2009, doi: 10.1109/MMM.2009.934516. keywords: Predistortion;Radio frequency;Multiaccess communication;OFDM;Power generation;Linearity;Linearization techniques;Mathematical model;Circuit testing;Peak to average power ratio,