

计算机科学的数学基础-Project2

王熠笑 522031910732

2024 年 5 月 20 日

1 问题定义的描述和分析

在此任务中，我们的目标是建立一个行为模型，用于理解和预测功率放大器（PA）的输入信号与输出信号之间的关系。功率放大器是无线通信系统中至关重要的组件，它的作用是放大输入信号以满足发送需求。然而，由于内在的非线性特性和记忆效应，功率放大器的行为预测变得复杂且具有挑战性。

1.1 问题描述

1.1.1 输入与输出信号

在此模型中，输入信号 X 和输出信号 Y 由下列形式表示：

$$X = [\dots, x_{n-m}, x_{n-m+1}, \dots, x_n, \dots]$$

$$Y = [\dots, y_{n-m}, y_{n-m+1}, \dots, y_n, \dots]$$

其中 n 表示当前时刻，而 m 是功率放大器的记忆深度。这意味着当前输出不仅取决于当前的输入，还受到过去 m 个时间点输入的影响。

1.2 问题分析

1.2.1 非线性和记忆效应

功率放大器的非线性和记忆效应是导致建模复杂性的主要原因。非线性意味着输出信号不是输入信号的线性函数，而记忆效应意味着输出不仅依赖于当前输入，还依赖于之前的输入。这些特性使得简单的线性模型无法准确预测放大器的行为。

1.2.2 模型的建立

为了捕捉这种复杂的动态关系，我们可以采用非线性回归或深度学习方法。这些方法能够处理输入数据的非线性关系，并能考虑到输入数据的历史信息，从而更准确地预测输出。

1.2.3 数据集和评价指标

我们将使用实验中收集的数据集，包括训练集和测试集，这些数据集中包含了不同时间点的输入输出信号。评价模型性能的指标为归一化均方误差 (NMSE)，它衡量的是预测输出与实际输出之间的误差。

通过上述分析，本项目旨在通过先进的建模技术，实现对功率放大器行为的精确预测。

2 数据预处理过程介绍

在本项目中，数据预处理是模型建立和性能评估的第一步。这个过程包括几个关键步骤：数据的加载、实部和虚部的分离、数据的标准化处理，以及为非线性回归模型准备的特征工程。

2.1 数据加载

首先，使用 `scipy.io.loadmat` 函数从 MAT 文件中加载训练集和测试集数据。输入信号和输出信号存储在变量 'paInput' 和 'paOutput' 中，分别表示功率放大器的输入和输出信号。

```
train_data = sio.loadmat('./dataset/task1/PA_data_train.mat')
test_data = sio.loadmat('./dataset/task1/PA_data_test.mat')
```

2.2 实部和虚部的分离

加载的数据是复数形式的，我们将输入和输出数据的实部和虚部分开处理，以便能够单独分析和建模这些部分。

```
x_train_real = x_train.real
x_train_imag = x_train.imag
y_train_real = y_train.real
```

```
y_train_imag = y_train.imag
```

2.3 数据标准化

使用 `sklearn.preprocessing.StandardScaler` 对数据进行标准化处理。这一步骤对于大多数机器学习模型来说是必要的，因为它可以帮助模型更好地学习和理解不同尺度的特征。

```
scaler_x_real = StandardScaler()
scaler_x_imag = StandardScaler()
scaler_y_real = StandardScaler()
scaler_y_imag = StandardScaler()

x_train_real_scaled = scaler_x_real.fit_transform(x_train_real.reshape(-1, 1))
                        .flatten()
x_train_imag_scaled = scaler_x_imag.fit_transform(x_train_imag.reshape(-1, 1))
                        .flatten()
y_train_real_scaled = scaler_y_real.fit_transform(y_train_real.reshape(-1, 1))
                        .flatten()
y_train_imag_scaled = scaler_y_imag.fit_transform(y_train_imag.reshape(-1, 1))
                        .flatten()
```

通过上述预处理步骤，数据被适当地清洗和转换，为接下来的模型建立和性能评估打下了坚实的基础。

3 所用方法和实施过程的详细说明

本部分详细介绍了在本项目中尝试的两种方法：矩阵变换与线性回归和深度学习模型，并详细说明了这些方法的实施过程。

3.1 矩阵变换与线性回归方法

在初步尝试中，我们采用线性回归模型来预测功率放大器的输出。该方法通过构建具有历史记忆深度的特征矩阵来实现，旨在使用历史输入数据来预测当前的输出。

3.1.1 特征工程

为每个数据点构建特征时，根据其前 10 个时间步的历史值创建特征矩阵。此特征工程通过以下函数实现：

```
def create_features(x, memory_depth):
    X = np.zeros((len(x), memory_depth))
    for i in range(memory_depth):
        X[i:, i] = x[:len(x) - i]
    return X[memory_depth - 1:]
```

3.1.2 模型训练与性能评估

使用 `LinearRegression` 模型对实部和虚部数据分别进行训练和测试。性能通过计算标准化均方误差（NMSE）来评估：

```
model_real = LinearRegression()
model_imag = LinearRegression()
model_real.fit(X_train_real, Y_train_real)
model_imag.fit(X_train_imag, Y_train_imag)
```

尽管进行了这些步骤，但由于线性模型未能充分捕捉数据中的非线性关系，导致预测精度不高。

3.2 深度学习方法

鉴于线性模型的局限性，我们转向使用深度学习模型，尤其是多层全连接神经网络，以更好地处理数据的非线性特性。

3.2.1 网络结构与训练

模型结构包括两个具有 128 个神经元的隐藏层，使用 ReLU 激活函数，以及直接输出预测的实部和虚部的输出层。使用 Adam 优化器和均方误差损失函数进行模型编译和训练：

```
model = Sequential([
    Dense(128, activation='relu', input_shape=(X_train.shape[1],)),
    Dense(128, activation='relu'),
```

```
Dense(y_train.shape[1])
])
model.compile(optimizer='adam', loss='mean_squared_error')
```

3.2.2 性能评估

模型在训练过程中展示了良好的性能，并在测试集上取得了显著的性能改进。这也证明了深度学习方法在处理具有非线性和复杂历史依赖的数据时的有效性。

3.3 结果可视化

为直观展示模型的性能，我们实现了为两种实现方法均绘制了预测结果与实际值的对比图，从而直观的表现验证模型是否能够学习到数据中的关键趋势和模式。

4 测试结果

在模型的实施和测试过程中，我们获得了以下的性能评估结果，使用 NMSE 作为主要的性能评价指标。这一指标能有效地反映出预测值与实际值之间的偏差大小。

4.1 矩阵变换与线性回归方法

使用线性回归模型处理复杂的非线性数据时，我们观察到以下的测试结果：

NMSE (Test): 1.445330463396422

此结果表明，尽管模型能够到达一定的预测准确度，但由于功率放大器数据的非线性和记忆特性，线性模型无法充分捕捉所有的数据动态。NMSE 值为正指出模型在预测中存在一定的准确性限制。

具体的结果如下图所示：

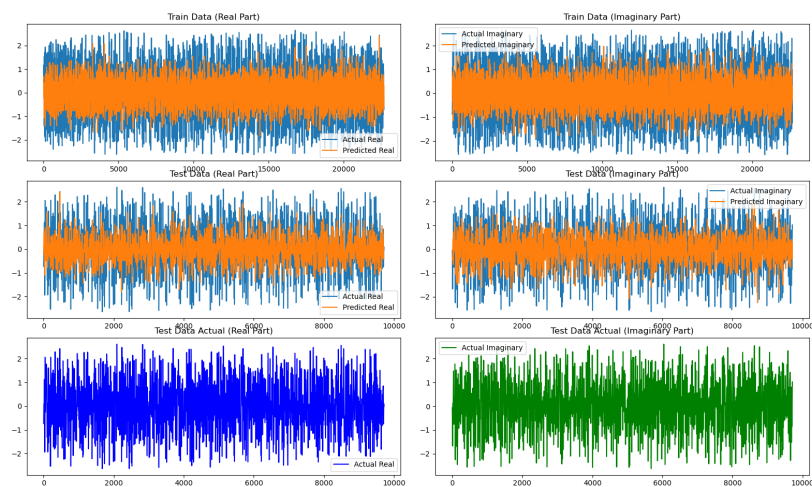


图 1: 矩阵变换与线性回归方法结果图

4.2 深度学习方法

转向使用深度学习方法后，我们获得了显著改善的测试结果：

NMSE (Test): -32.770314900066154

具体的结果如下图所示：

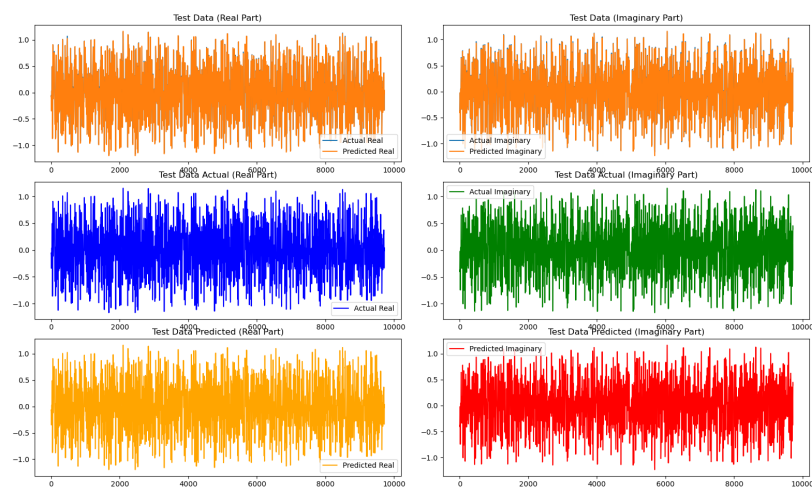


图 2: 深度学习方法结果图

这一极为负的 NMSE 值表明预测值与实际值之间的误差非常小，显示出深度学习模型在捕捉和建模复杂非线性关系方面的高效能力。深度学习模型通过其多层结构能够学习到数据中的复杂模式和动态变化，从而显著提高预测的精度和可靠性。

4.3 结果分析与讨论

从测试结果可以看出，深度学习方法相较于传统的线性回归方法有显著的优势。特别是在处理具有强烈非线性特性和高维数据的情况下，深度学习能够通过其多层次的非线性映射能力，显著提升模型的表现。这一结果强调了在面对复杂数据时，选择合适的建模方法的重要性。

这些发现不仅验证了我们选择使用深度学习方法的正确性，也为未来的研究提供了宝贵的经验和数据支持。未来工作可以进一步探索更多的深度学习架构和训练策略，以优化模型性能和泛化能力。

5 结论

本研究通过比较矩阵变换与线性回归方法和深度学习方法在预测功率放大器行为方面的表现，明确了深度学习在处理此类问题时的优越性。未来的研究可以继续深度学习领域深入，探索其与其他复杂系统建模中的应用潜力。