**一、课题综述**

* 1. **课题说明**

小组成员：何诗锟、刘子牛、彭坤宇、张祖豪

任务划分：

-Data Prepare：

-Data Preprocess：

-Model Construct：

-Train & Test：

-Plot Result：

-Optimize & Review：张祖豪

* 1. **课题目标**

本课题选取abalone和Wine Quality作为数据集，均来自于UCI MACHINE LEARNING，广泛应用于机器学习的教育和实践。数据集abalone用于预测鲍鱼的年龄，Wine Quality用于评估葡萄酒的质量。

课题的目标为手写实现回归算法模型，利用数据特征标准化、Adam优化以及bp算法对其进行优化，完成基于机器学习的线性回归任务，取得较优的实验结果。并对不同学习率下的损失函数进行讨论，探究学习率对损失函数的影响。

* 1. **课题数据集**

本课题使用abalone和Wine Quality两个数据集。

其中abalone来自UCI Machine Learning Repository（https://archive.ics.uci.edu/dataset/1/abalone），用于预测鲍鱼的年龄，包含Sex、Length、Diam、Height、Whole、Shucked、Viscerat、Shell 共8个特征。目标变量为Rings，可作为年龄的判断标准。

Wine Quality同样来自UCI Machine Learning Repository（https://archive.ics.uci.edu/dataset/186/wine+quality），用于评估葡萄酒的质量。包含了红葡萄酒和白葡萄酒的11个特征：Fixed Acidity、Volatile Acidity、Citric Acid、Residual Sugar、ChloridesFree、 Sulfur Dioxide、Total Sulfur Dioxide、Density、pH、Sulphates、Alcohol。目标变量为Quality，以整数值表示，取值范围从3到9，其中较高的评分表示更高质量的葡萄酒。

**二、实验报告设计**

* 1. **数据准备**

经过充分的调研，最终决定选取abalone，Wine Quality两个经典的用于回归任务的数据集。数据集均来源于UCI Machine Learning Repository，充分尊重其引用要求。

* 1. **数据预处理**

·从文件中读取数据：读取方式为使用 Python 的 argparse 库来解析命令行参数，默认值为abalone.data。

·将数据集中的类别标签转换为数值标签：abalone数据集中性别特征值取值为‘M’、‘F’、‘I’，为便于后续回归的实现，将其分别转换为数值0、1、2。

·将数据集分为自变量（特征，x）和因变量（目标，y）：借助train\_test\_split函数完成测试集和训练集的划分。

·标准化：使用标准化（StandardScaler）对特征进行标准化，以确保它们具有相同的尺度，加速梯度下降的过程，使模型权重更具可解释性，降低特征的量纲影响，防止数值的不稳定性，同时提高模型的泛化能力。

* 1. **模型搭建**

本次实验实现了一个用于回归任务的机器学习模型，构建和训练一神经网络模型，支持反向传播和参数优化。

bp过程：

·前向传播（Forward Propagation）：在前向传播阶段，通过调用每个层节点的 \_forward 方法，从输入节点开始，逐步计算神经网络的输出。\_forward 方法的输出作为下一个层节点的输入，沿着计算图向前传播，直到计算出模型的预测值和损失值。

·损失计算：在前向传播的最后，通过计算损失函数（例如，均方损失或softmax损失）来评估模型的性能。

·反向传播：反向传播是通过计算图中节点的反向传递梯度来实现的。首先，初始化梯度为1.0，从损失节点开始反向传播。对于每个节点，调用其 \_backward 方法，该方法接收梯度并计算相对于输入的梯度。这些梯度通过链式法则传播回每个节点的父节点，以计算它们的梯度。最终，梯度被传播回模型的参数节点（常量节点），从而可以更新这些参数以最小化损失。

·参数更新：在梯度计算完成后，可以使用梯度下降等优化算法来更新模型的参数。在本实验中中，利用Adam优化算法更新参数。这是一种基于梯度的优化算法，具有动态学习率。

Adam优化：

Adam是一种基于梯度的优化算法，结合了动量和自适应学习率的概念，主要优点是它自适应学习率，并且能够在训练过程中有效地调整学习率，因此适用于不同参数的更新速度不同的情况。它结合了动量项和二阶动量项，使得在训练初期可以快速收敛，并在接近最优解时更加稳定。

过程：

1. 初始化参数：

- 初始化模型参数，包括权重和偏置。

- 初始化两个动量变量m（一阶动量）和v（二阶动量）为零。

- 初始化时间步数t为1。

2. 计算梯度：

- 对于每个训练批次，计算损失函数对于模型参数的梯度。通过反向传播来实现。

3. 更新动量：

- 计算一阶动量m和二阶动量v的估计值：

- 更新一阶动量m：

- 更新二阶动量v：

这里的梯度²表示每个元素的平方

4. 修正偏差：

- 由于m和v在初始时被初始化为零，它们会有偏差。为了减轻这种偏差，需要对它们进行修正：

- 修正一阶动量m：

- 修正二阶动量v：

5. 更新参数：

- 使用修正后的一阶动量m`和二阶动量v`来更新模型参数：

- 参数更新：

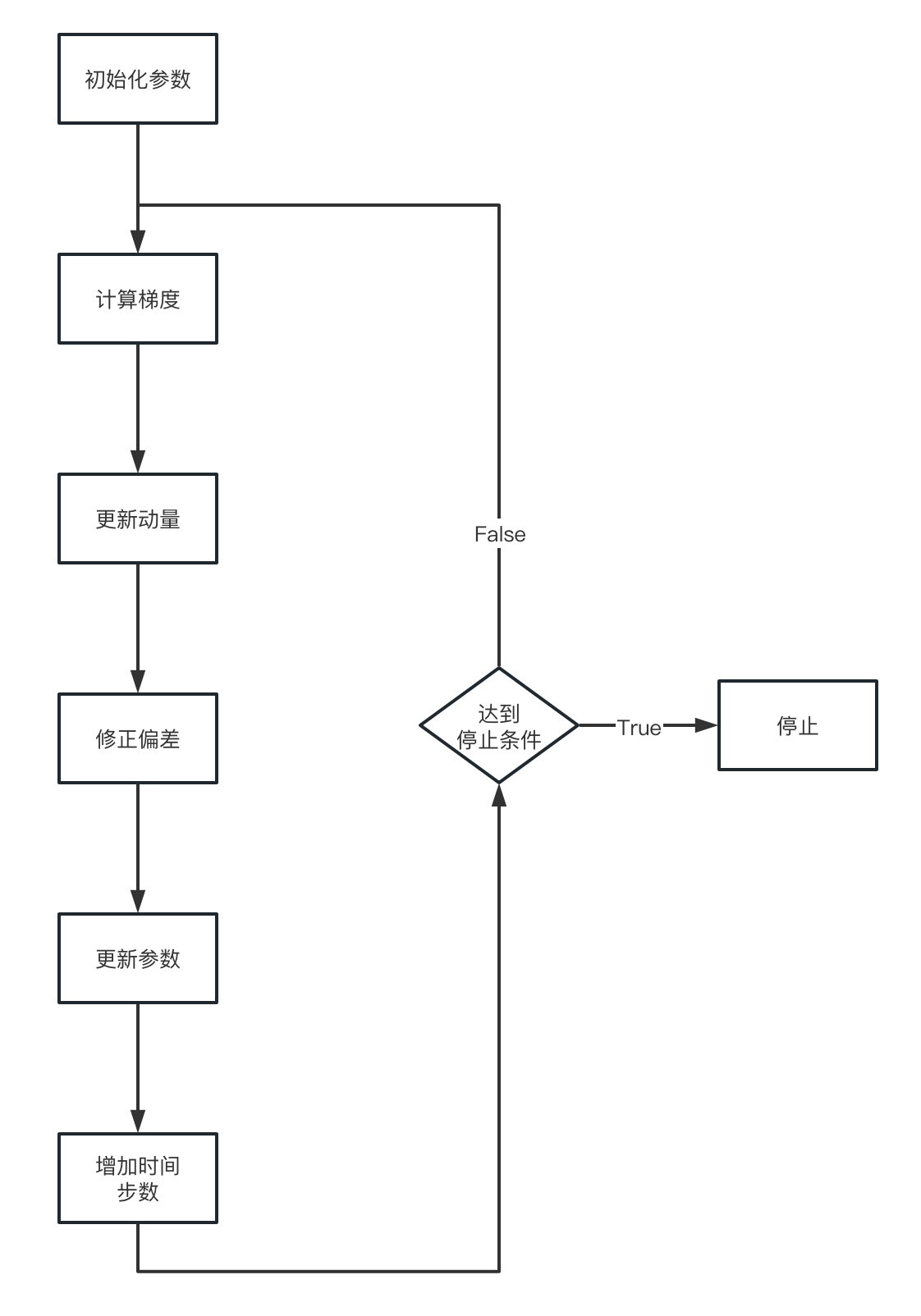
其中ε是一个很小的常数，用于防止除以零。

6. 增加时间步数：

- 将时间步数t增加1，用于下一次迭代。

7. 重复迭代：

- 重复步骤2到步骤6，直到达到停止条件（达到最大迭代次数或损失足够小）。



* 1. **模型训练测试**

·abalone数据集

学习率：0.03



·Wine Quality数据集

winequalityWhite

学习率：0.0003



winequalityRed

学习率：0.0003

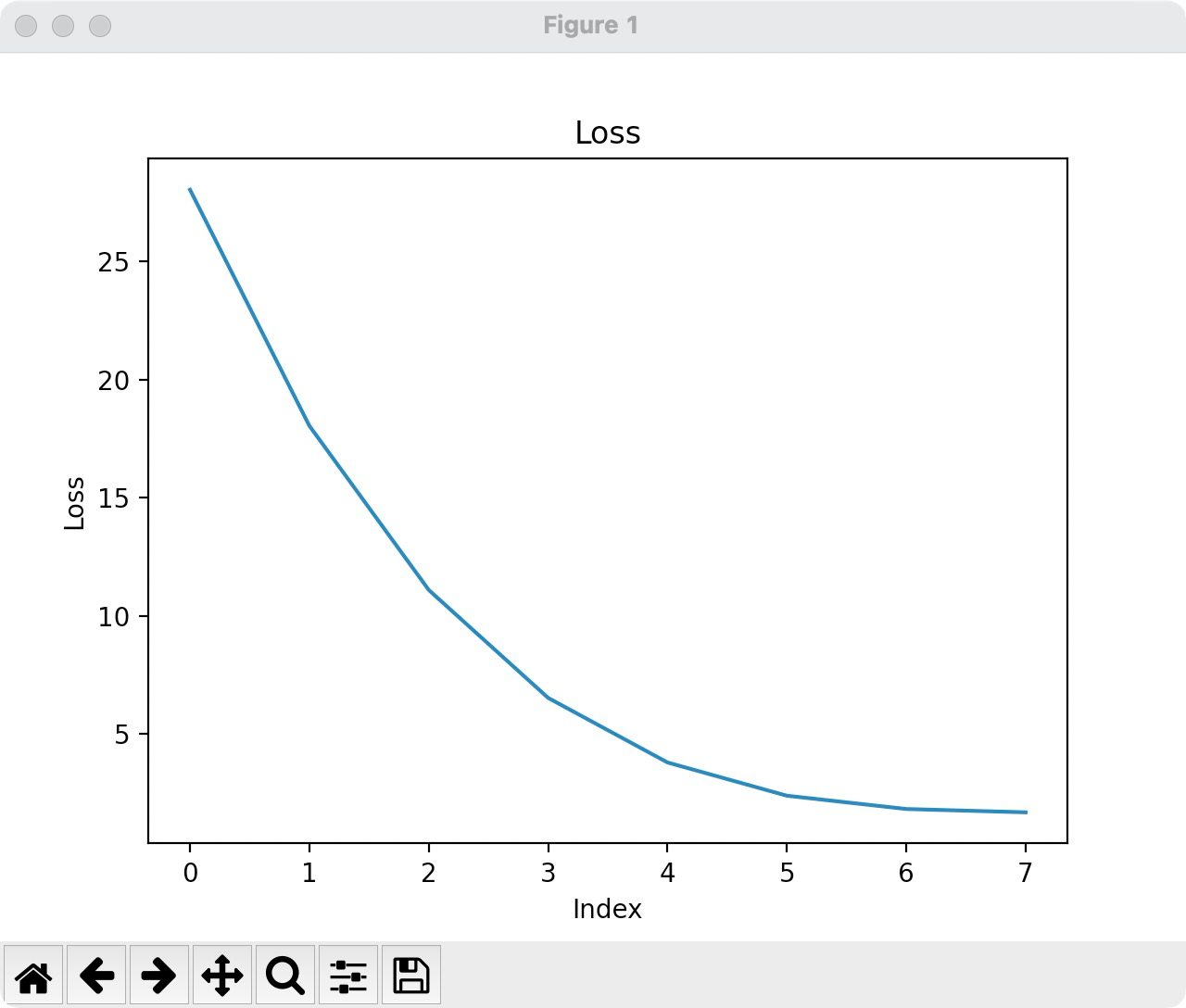


* 1. **结果可视化**

利用使用 Matplotlib 绘制训练过程中的损失曲线。x 轴表示训练批次的索引，y 轴表示损失值。帮助了解模型的训练进展和收敛情况。

·abalone数据集

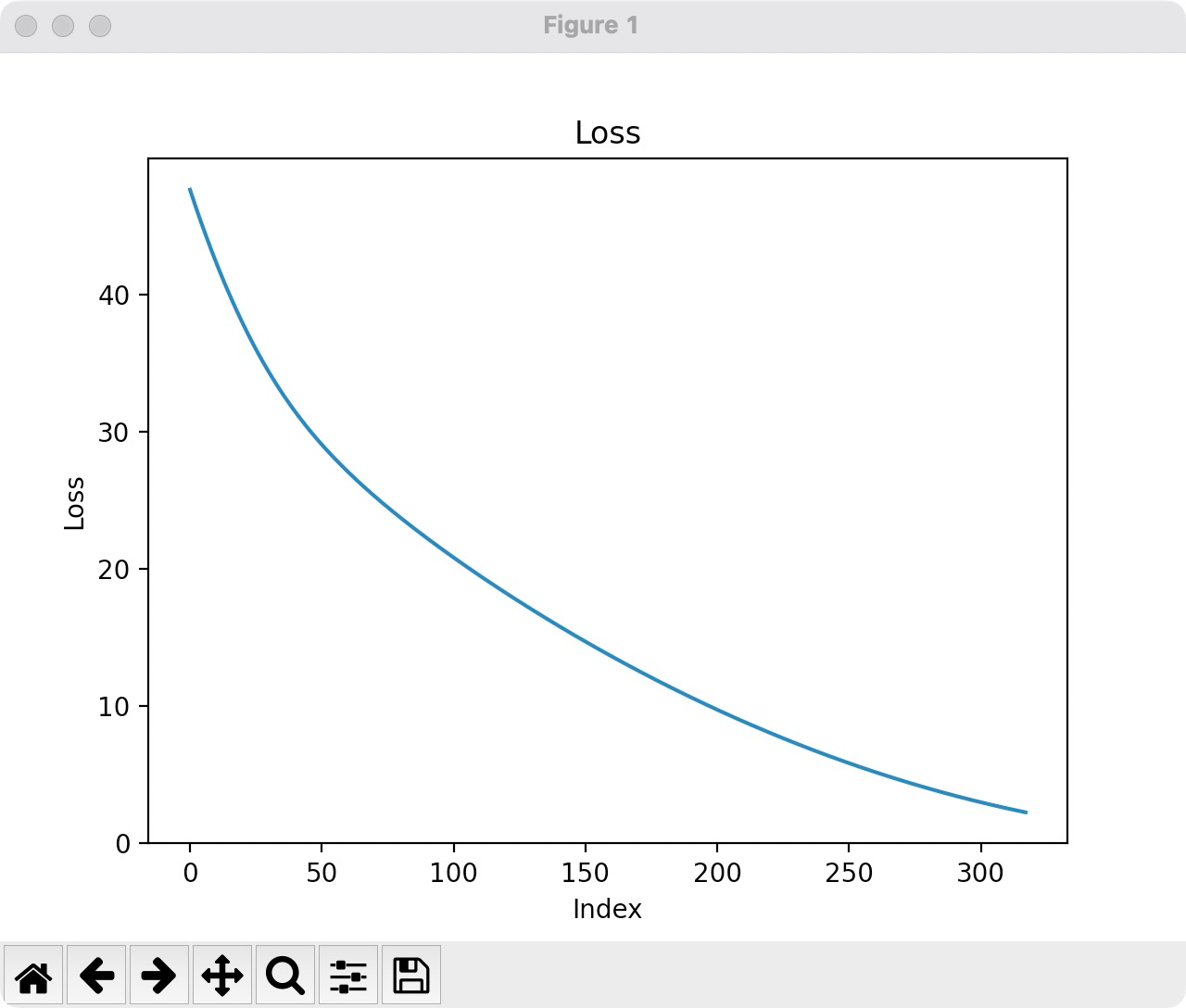
学习率：0.03



·Wine Quality数据集

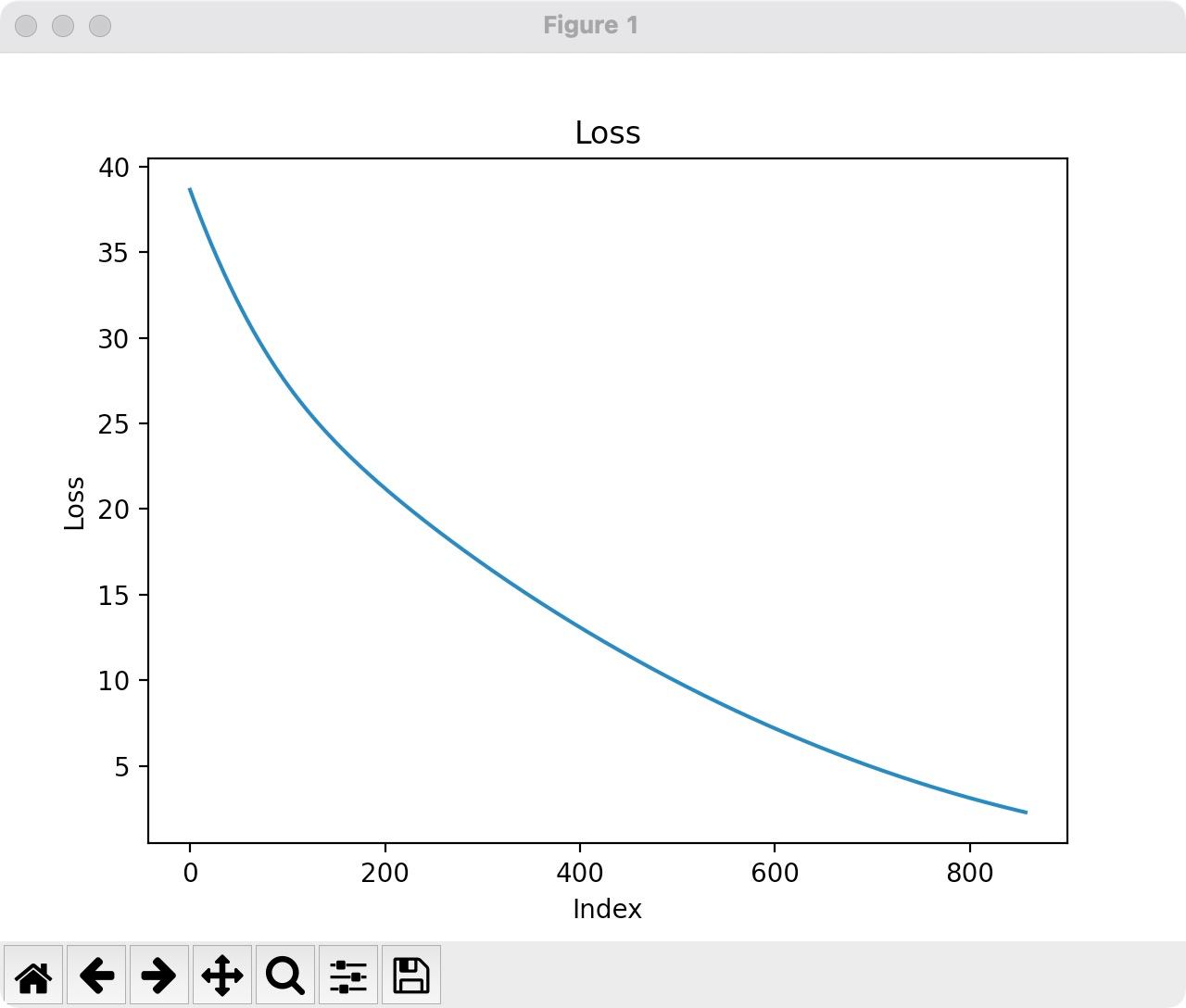
winequalityWhite

学习率：0.0003



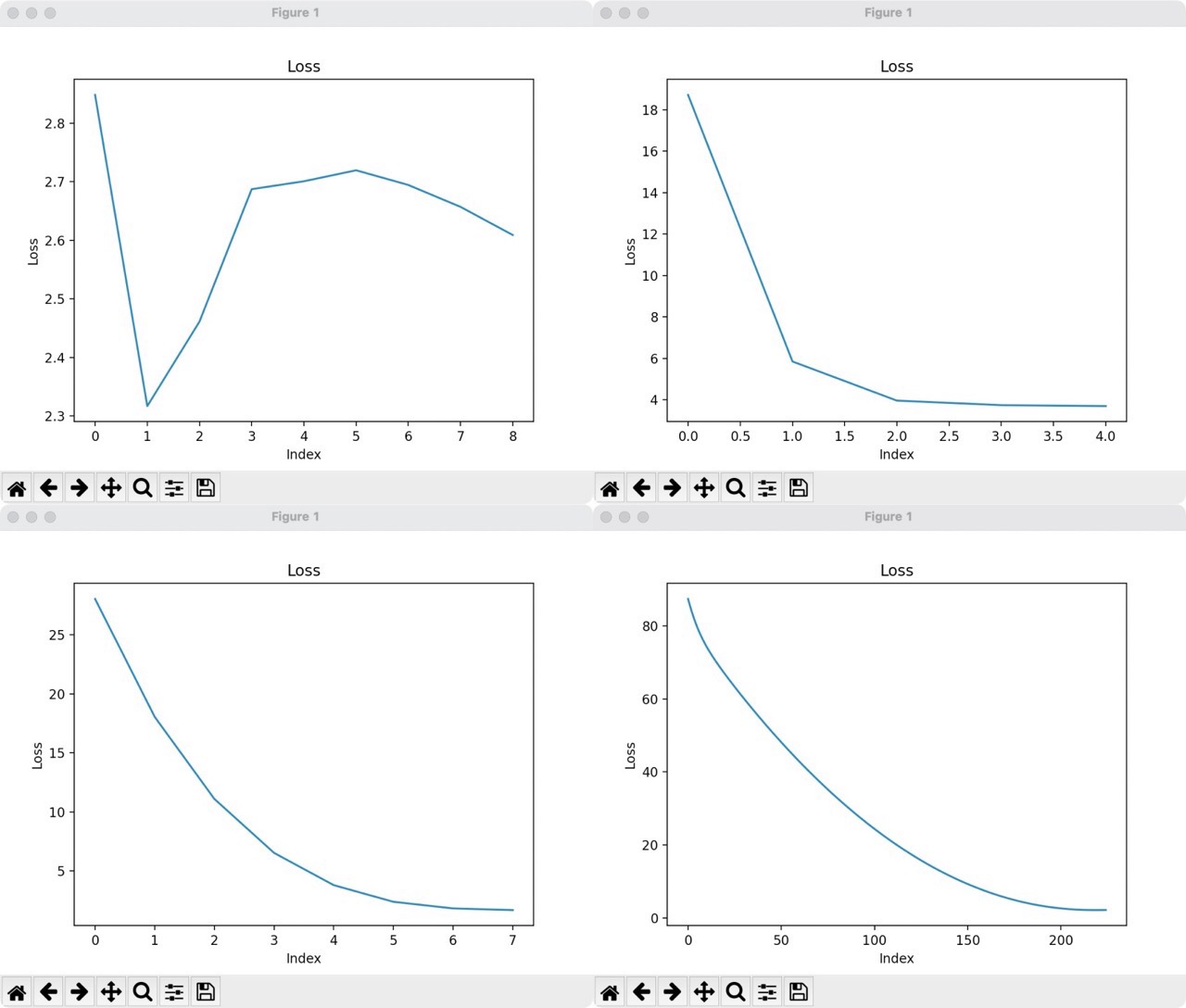
winequalityRed

学习率：0.0003



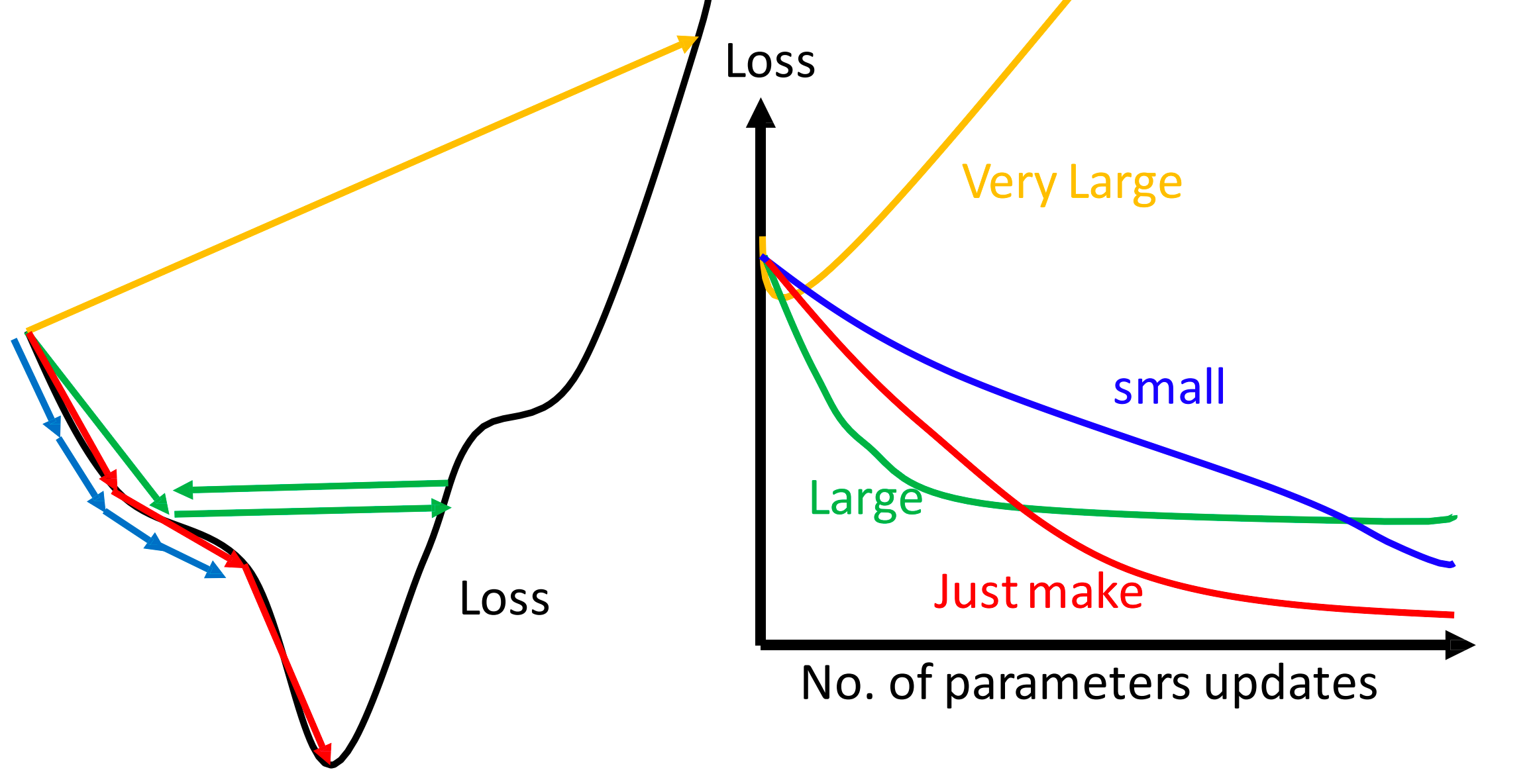
* 1. **分析和优化**

探讨同一数据集同一模型下学习率对损失函数的影响，以abalone数据集为例展开讨论。



（学习率：左上：1，右上0.1，左下：0.03，右下：0.01）

通过观察，学习率的大小直接影响损失函数，学习率过大时会导致损失函数发散而无法收敛，略大时可能会出现未收敛于极小值，但这并不意味着学习率越小越好，过小时会导致收敛过于缓慢，因此选取恰当的学习率至关重要。



1. **总结**

本次实验为课程的第一次实验，对机器学习的回归任务有了初步的体会。整个实验流程大致分为Data Prepare、Data Preprocess、Model Construct、Train & Test、Plot Result、Optimize & Review六个步骤，每个步骤都至关重要不可或缺。

回归任务的关键是通过大量的数据去找到合适的系数，从而达到准确预测目标值的任务。系数是否合适的判断依据是代价函数是否到达了极小值。这一过程需要不断的迭代参数，可采用梯度下降、Adam等方式训练。评估通常以均方根误差（RMSE）、平均绝对误差（MAE）、R平方等为标准。对学习率的探究也让我们对回归任务有了更深的理解，收获很多。