**一、线性回归**

* 1. **课题说明**

**手写实现线性回归模型进行波士顿房价的预测并进行优化探索**

**组员分工：**

**陈奕澄：模型搭建，Optimizer实现，SGD优化，设计实验探索优化效果和参数变量的关系**

**王均涛：数据集处理，Normalization，Feature Scaling，PCA数据降维，多项式数据升维，结果可视化**

* 1. **课题目标（示例）**

本次实验的目标为：

1.通过手写实现从数据预处理到模型构建再到绘制结果和优化的线性回归机器学习模型搭建全流程，了解如何从数学原理转变为实际的训练代码。

2.通过在线性回归模型训练中调整学习率、正则化参数等研究这些超参数对于训练结果的影响，包括积极影响和消极影响。

3.探究输入变量参数之间的关联性，探索数据升维、数据降维、特征归一化的一些常用方法，并且根据它们造成的影响探讨哪些方法更具有普适效应，哪些限制更多。

* 1. **课题数据集**

波士顿房价数据集（数据来源UCI Machine Learning Repository，https://archive.ics.uci.edu/ml/machine-learning-databases/housing/housing.data）是一个经典的数据集，由Harrison和Rubinfeld在1978年发布，用于研究房屋价格的预测问题。数据集中的数据来自于1970年代中期波士顿周边地区的房屋销售记录，包括了506个样本，每个样本有13个特征，没有明显的缺失值，目标是预测房屋价格中位数，是一个连续变量。我们小组之所以选择这个数据集，是因为它的特征和目标之间的关系比较复杂，而且数据集的大小适中，不会过于复杂。这既允许我们两个初学者能够很好地上手回归问题，实现从数据预处理、模型构建到绘制结果、优化和审查全流程的手写完成，又具备一定的复杂性和很强的可优化性，让我们有探索归一化、主成分分析等预处理方法和Feature Scaling、Optimzer等优化方法的空间。这个模型还允许我们与现有的预测结果之间进行参考，让我们可以了解所做出来的结果相对于优秀结果之间的差距，激励我们尝试更多优化方法。

具体特征如下：

- CRIM：城镇人均犯罪率。

- ZN：住宅用地超过 25,000 sq.ft. 的占地面积比例。

- INDUS：城镇非零售商业用地比例。

- CHAS：是否靠近 Charles River 。

- NOX：一氧化氮浓度（每千万分之一）。

- RM：每个住宅的平均房间数。

- AGE：1940 年之前建成的自用房屋比例。

- DIS：距离 5 个波士顿就业中心的加权距离。

- RAD：距离高速公路的便利指数。

- TAX：每 10,000 美元的全额物业税率。

- PTRATIO：城镇师生比例。

- B：1000(Bk - 0.63)^2 其中 Bk 是城镇黑人比例。

- LSTAT：人口中地位低下者的比例。

**二、实验报告设计**

* 1. **数据准备**

1. 下载波士顿房价数据集 housing.data
2. 将其转化为csv格式
3. 读取：将每个样本的13个特征和需要预测的price分别存入x和y变量中返回
   1. **数据预处理**

本实验采用的数据预处理包括：1）特征归一化Feature Scaling，2）主成分分析Principal Components Analysis (PCA)，3）多项式回归Polynomial regression

1. 特征归一化

Feature Scaling可以将特征的取值范围缩放到相似的尺度上，以便更好的拟合模型，即消除不同特征之间的尺度差异，使得每个特征对模型的贡献相对均衡。本次实验使用的是标准化，其公式如下：

1. 主成分分析

主成分分析可以将原始的高维特征空间降维到一个低维的特征空间，以更好的拟合模型和提高模型的性能。通过降维，可以减少特征数量，降低计算复杂度和减少存储空间，同时可以提高模型的鲁棒性和泛化能力。在进行PCA之前，需要对数据进行标准化处理，避免特征之间的数量级差异对PCA结果的影响。

PCA主要是通过线性变换将原始特征空间中的数据映射到一个新的地位空间中，使得新空间中的数据保留原始数据中的主要信息。

在本实验中，PCA的主要步骤是：

1. 计算协方差矩阵：协方差矩阵可以反映不同特征之间的相关性。
2. 计算协方差矩阵的特征值和特征向量：特征向量可以表示协方差矩阵的主要方向，而特征值表示在该方向上的方差大小。可以通过特征值分解的方法，得到协方差矩阵的特征值和特征向量。
3. 选择主成分：选择在新的地位空间中保留的特征数量，通常可以选择保留协方差矩阵的前k个特征向量。这些向量构成了一个转换矩阵W，可以将原始数据集X通过矩阵乘法映射到新的低维空间中去。
4. 映射：公式如下，其中，W为m x k的矩阵，X为 n x m的矩阵（n为样本数量，m为特征数量）
5. 多项式回归

数据升维是一种处理非线性关系的方法，通过将原始数据进行转换，将地位特征映射到更高维空间中，从而使得模型可以更好的拟合数据。

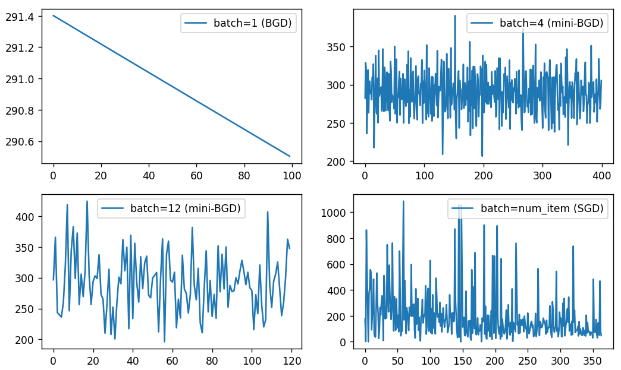
在本实验中，数据升维可以通过多项式回归来实现。多项式回归是一种特殊的线性回归算法，将特征向量转化成高位向量的形式，来处理非线性关系的问题。即用更高阶的函数来拟合数据，从而提高模型的拟合度和预测性能。

但是其具有更复杂的模型，可能导致过拟合等问题

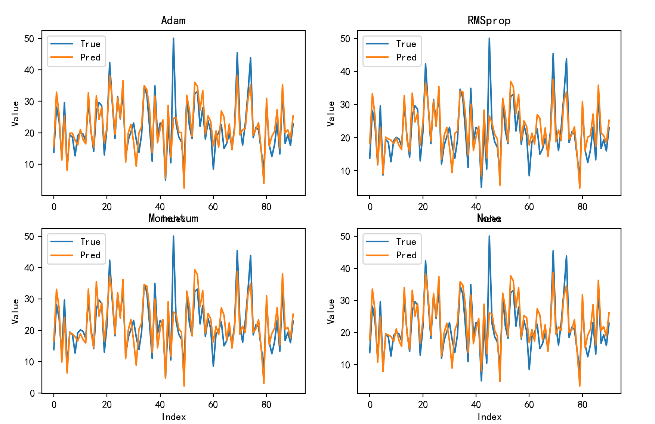
* 1. **模型搭建**

本次实验主要是搭建线性回归模型，封装在项目的regression\_model.py中的LR类中，主要包含1）初始化，2）训练，3）预测，4）评测功能

1. 初始化：将线性回归所需要的数据和参数进行初始化。将数据的数目和特征数分别存入num\_item和num\_feature中。包含的参数：权重矩阵theta（维数：num\_feature + 1），训练轮数epoch，学习率lr，正则化参数alpha，优化器参数gamma和beta，batch个数，优化器方法optimizer。需要特别注意的是：类中存储的训练数据x比实际的数据多一个特征，恒为1，使得x与theta的乘积包含bias。
2. 训练：
   1. 根据参数batch计算batch\_size
   2. 为优化器Momentum，RMSprop，Adam初始化初始化向量m, v, M
   3. 循环epoch
      1. 随机化x，y
      2. 根据batch\_size从x，y中取当前batch
      3. 计算梯度grad
      4. 根据不同优化器的方法对权重矩阵进行更新
      5. 对优化器向量进行更新
      6. 计算cost函数，输出或存储
3. 预测：使用权重矩阵与输入参数x点乘，得到预测值y\_pred
4. 评测：包含MSE，RMSE，MAE，R\_Squared四种评测函数
   1. **模型训练测试**
5. 模型训练：主要过程参考2.3第2部分，训练过程中，将每个epoch的cost进行记录，观察并绘图。



1. 模型测试：主要过程参考2.3第4部分，使用MSE，RMSE，MAE和R\_Squared四个方法进行评测。
   1. MSE 是平均误差的平方，是用于测量预测值与实际值之间的差异的指标。MSE 的值越小，模型的性能越好。
   2. RMSE 是 MSE 的平方根，与 MSE 相比，它对误差大的数据点的惩罚更严厉。RMSE 的值越小，模型的性能越好。
   3. MAE 是预测值和真实值之间的绝对差异的平均值。MAE 对所有误差的大小都给予相等的权重，因此对于误差大的数据点的惩罚比 MSE 和 RMSE 要小。MAE 的值越小，模型的性能越好。
   4. R\_Squared 是用于衡量模型的拟合程度的指标。它表示预测值与实际值之间的变异程度。R\_Squared 的值越接近 1，模型的性能越好。



* 1. **结果可视化**

可视化结果放在2.6节中与结果分析和优化探讨一起呈现，实验结果的部分放在了结论部分呈现，故而这里不再另外展示。本次实验所使用的可视化工具主要为matplotlib库。

* 1. **分析和优化**

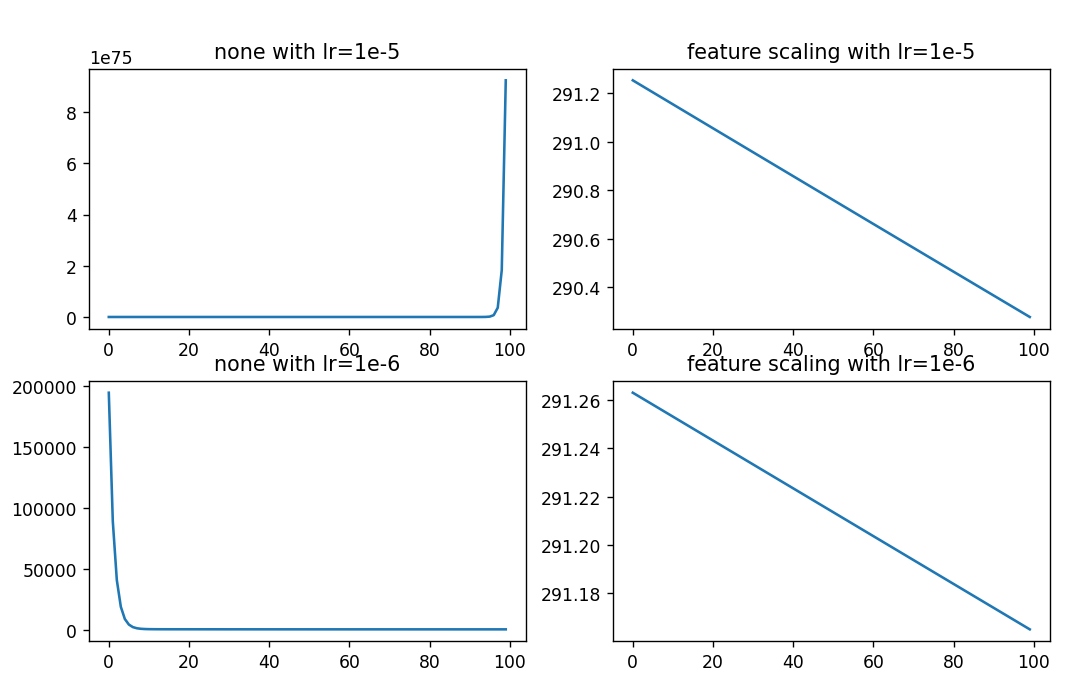
本实验在基础的线性回归模型上，主要做了如下优化：1）Feature Scaling，2）Batch，3）Normalization，4）PCA，5）Optimizer。并分别对其进行了实验进行分析。

1. Feature Scaling

原理参考2.2第一部分

在进行实验过程，出现了梯度爆炸（cost不断增加或者cost巨大）的情况。可能是由于学习率过大，导致模型参数更新过大，导致模型在优化过程中跳过最优点，从而使损失函数的值不断增大；或者模型过拟合，即当模型过于复杂或者训练集中样本数量不足，可能会导致模型过拟合训练数据，从而使得模型在测试数据上的表现不佳，损失函数的值不断增大。

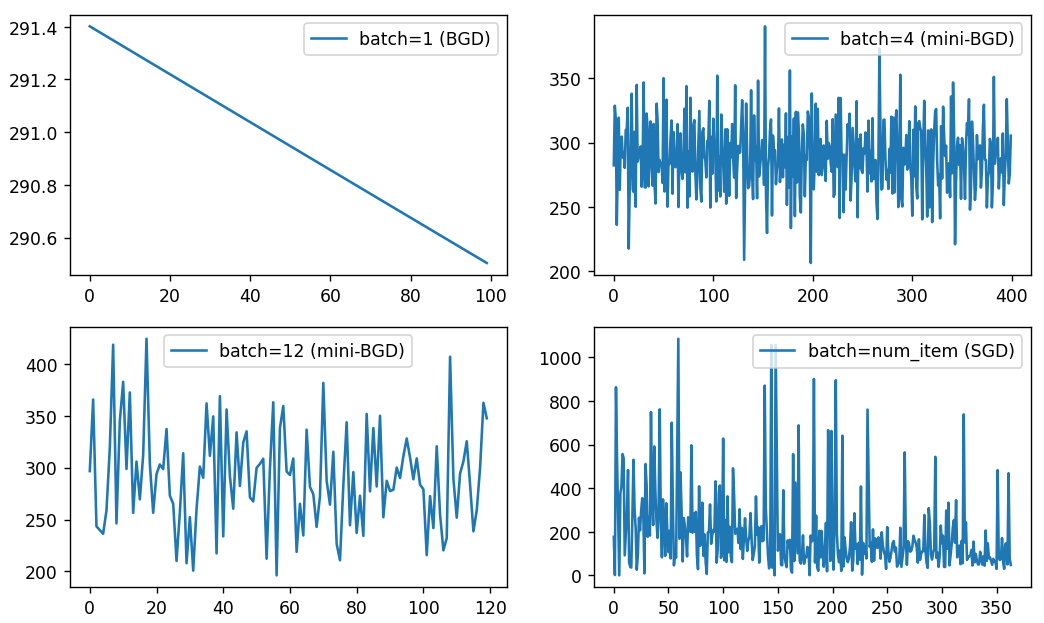
Feature Scaling可以帮助解决上述问题的出现，通过避免特征值对模型训练的影响不均和加速模型收敛（如果特征值的范围差异较大，会导致梯度下降算法在更新模型参数时需要更多的迭代次数才能收敛。通过将特征值缩放到相似的范围，可以加速梯度下降算法的收敛速度）来解决上述问题。



根据实验，可以发现当合适的学习率（不会发生梯度爆炸）为1e-6的数量级。对学习率1e-5和1e-6，是否使用Feature Scaling进行了实验，可以发现，其可以有效的避免在较大学习率出现的梯度爆炸的问题，同时具有更快的收敛速度。

1. Batch

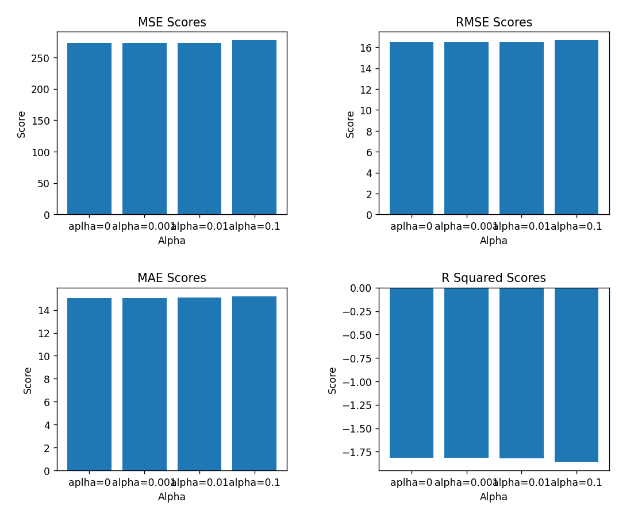
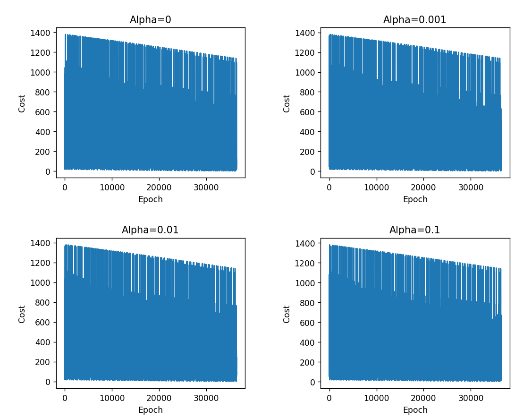
相比较于BGD，SGD有更快的收敛速度：SGD每次只更新一个样本的参数，因此每次迭代计算的速度比BGD要快的多；更少的内存消耗：由于SGD只需要考虑一个样本的梯度，因此只需要存储一个样本的参数和梯度，而BGD需要存储整个训练集的参数和梯度；更好的泛化能力：由于SGD更新参数时只考虑一个样本的梯度，因此SGD可以避免陷入局部最优解，并且可以更好的泛化到未见过的数据。



根据如图的实验结果，cost值下降到稳定值所用的时间SGD比mini-BGD和BGD要快，证明了SGD的收敛速度更快。同时根据验证数据test上MSE，RMSE，MAR等数据的比较，SGD的效果比mini-BGD和BGD更好。但是，由于单个样本的梯度时不稳定的，因此可以看到SGD的优化路径比较嘈杂。

1. Normalization

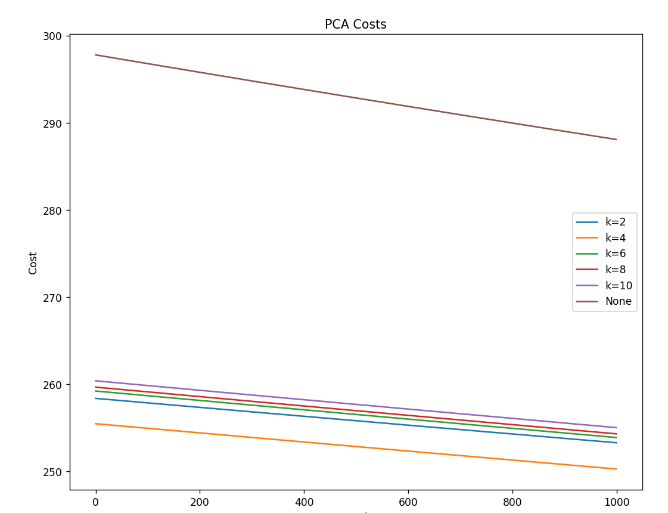
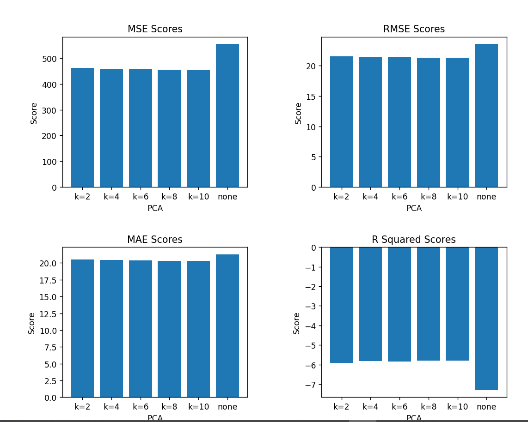
正则化可以用来减少模型的过拟合现象，提高模型的泛化能力。正则化通过添加一个惩罚项来限制模型的复杂度，从而降低模型对训练数据的过拟合程度。本实验主要采用的时L2正则化：惩罚项是模型参数的平方和，达到缩小模型参数，改善模型的鲁棒性的效果。

根据如图实验结果，正则化的效果不明显。可以是因为数据集比较小，所含的信息优先，过拟合的可能性相对较小；或者因为特征之间的相关性较小，过拟合的可能性也比较小，正则化的作用也相对较小。

1. PCA

原理参考2.2第二部分

根据如图实验结果，经过PCA降维后，在测试集上有较小的MSE，RMSE，MAE等，说明PCA降维可以有效的提高实验效果。主要原因可能为PCA提高了数据的线性可分性，即PCA将数据投影到了主成分上，使得数据在坐标系下更加线性可分；去除了数据噪声，PCA可以识别和去除数据中的噪声，从而提高模型的鲁棒性和泛化能力。

1. Optimizer

本次实验所采用的优化器为Momentum，RMSprop和Adam。

对于学习率的优化是必要的：如果learning rate选的过小，收敛速度会很慢，如果过大，loss function就会在极小值处不停地震荡甚至偏离。对于非凸函数，还要避免陷于局部极小值处。

若采取措施：先设定较大的学习率，当两次迭代之间的变化低于某个阈值后，就减小learning rate。其可以在一定程度上缓解上面问题，但是需要提前设定阈值，不能够适应数据集的特点。

因之需要引入学习率优化器：

（1）Momentum通过下面公式：

使得在梯度不变的维度上速度变快，梯度方向有所改变的维度上的更新速度变慢，达到加快收敛并减少震荡的效果。

（2）RMSprop

利用了二阶导数信息，解决学习率的设置导致的梯度爆炸问题。它的更新规则是基于当前梯度的平方的指数加权移动平均数来跟新权重，同时除以平方根。这个操作可以让权重在更新时的缩放更为平滑，因此更新的步长会更加稳定，即减少随机梯度下降算法中的震荡。

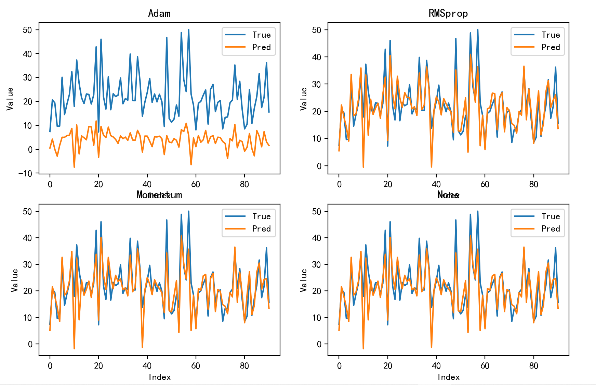
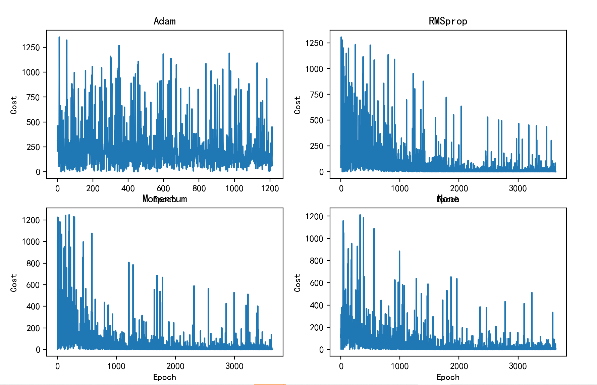
（3）Adam

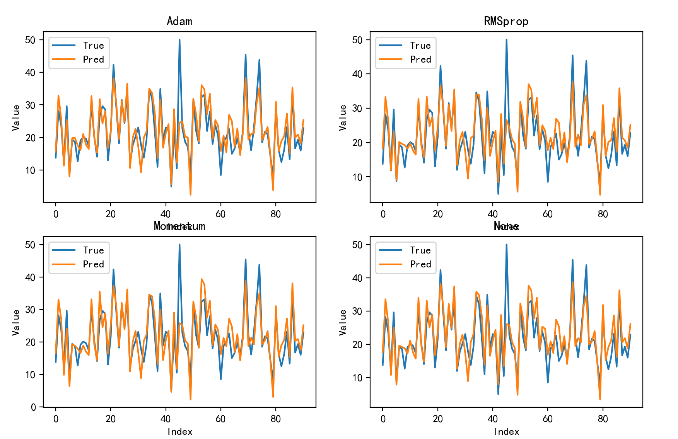
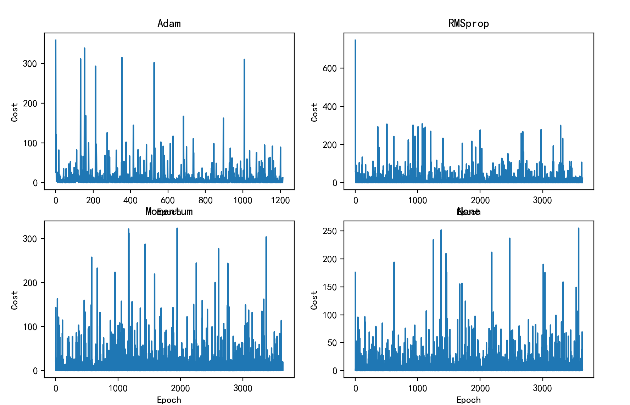
Adam为自适应矩估计Adaptive Moment Estimation，即对于梯度进行自适应调整，使得每个参数的学习率都可以独立自适应地调整。根据每个参数的历史梯度信息和梯度平方的移动平均值，自适应地调整每个参数的学习率，使得不同参数具有不同的学习率；同时在更新参数时，处理使用梯度信息，还引入了动量概念，即加入了梯度的指数衰减平均值，以便使更新过程更加平滑；在更少的训练迭代次数内获得更高的精度。

初始化：m，v表示梯度的一阶矩估计和二阶矩估计

偏差校正

梯度更新





根据上述实验结果，1）当学习率较小（第一张图lr=1e-5）时，Adam和RMSprop下降的速度比Momentum要慢。可能的原因分析：初始的学习率不当，Adam的性能对于初始学习率很敏感，与Momentum相比，Adam的默认学习率通常较低；Adam的性能在小数据上可能不如Momentum，Adam使用了一些二阶矩估计，需要大量的数据来准确估计这些矩阵。当数据集过小时，Adam的估计可能不准确；2）但当学习率较大（第二张图lr=1e-2）时，Adam和RMSprop下降地更加平缓，同时，减少了发生损失爆炸的可能性。

1. **总结**

本次线性回归任务，目标是建立一个模型来探究变量之间的关系。数据采用了经典线性回归波士顿预测房价数据集。采用了课程中所提到的特征归一化，正则化，随机梯度下降，主成分分析，学习率优化器等法师进行优化。并通过使用对比试验，验证了这些优化的作用。整个过程只调用numpy等基础的包，对线性回归模型进行实现，得到了与sklearn封装的线性回归模型相近的结果。

动手实现线性回归模型，让我们对于机器学习的过程掌握的更加熟练。同时，在实验中，出现了一些出乎意料的结果。具体来说，在真实数据集上，出现了损失爆炸的情况；在小数据集上出现了Adam的学习速率比Momentum慢的情况。对于这些特殊情况的分析，加强了我们对于模型和各种优化的理解。

线性回归模型是一种常见的机器学习模型，用于预测连续变量。虽然线性回归模型具有许多优点，例如可解释性强、易于实现和计算效率高等，但是也存在一些缺陷。1）对非线性关系拟合效果不佳：线性回归模型的主要假设是自变量与因变量之间的关系是线性的，即每个自变量对因变量的影响是恒定的。而实际的数据大部分是非线性的，因此线性模型不能进行很好的处理。2）对异常值和噪声敏感：线性回归模型对于异常值和噪声非常敏感，即使是一个小的异常值或噪声也可能导致模型出现很大的偏差。因为线性回归模型假设自变量与因变量之间的关系是线性的，当数据中存在异常值或噪声时，会影响模型的拟合效果，从而导致模型产生偏差。3）无法处理非数值型特征：线性回归模型只能处理数值型特征，对于非数值型特征，需要进行特殊的处理，例如独热编码等。这会增加特征工程的复杂度和计算成本。具体来说，在本次实验中，我们开始采用了Medical Cost Personal Dataset数据集，该数据集的特征包含age, sex, bmi, children, smoker, region, 需要预测医疗花费。其中，sex、smoker和region就是非数值特征，尤其是smoker与医疗花费有强相关。即使采用了独热编码，最后的效果也不佳，因为数据集本身非线性。

在小组学习的过程中，每个人有不同的想法和见解，相互学习，使得我们对于线性回归模型更深入的理解。同时，实验过程中，我们也有明确的分工，加强了我们团队协作的能力。

线性回归模型是一种用于预测连续变量的监督学习算法，其一些基本概念和技术可以用于分类问题中以及未来的机器学习课程。