**作业2：**

**一、对于给定的模型，比如二阶模型y=w1\*x2 + w2\*x + b，尝试找出该model中的最佳function。**

分别验证Data Scaling技术、学习率lr、BGD及其两种变体SGD和MBGD、不同的梯度优化方法、训练数据量，对损失函数L(f)的收敛速度和准确度的影响。其中，

* + 收敛速度：指，是否能很快找到最佳function？即，是否能很快找到Loss的极小值？
  + 收敛准确度：指，找到的最佳function是否正确？即是否能找到Loss函数的全局极小值？（若损失函数L(f)为凸函数，极小值只有一个）

（对应的代码为：AI\_StuAsssign\_OrderMore.py）

**1、验证数据归一化技术的效果。**采用BGD。超参数设置如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| training\_epochs | Learning Rate | N | TestingDataRatio | OrderNum | Seed | BatchSize |
| int(1000) | 1\*1e-4 | 50 | 0.2 | 2 | 16225151 | N\*(1-TestingDataRatio) |

1. 设置DataScaleFlag=0（表示未使用数据归一化技术）
2. 设置DataScaleFlag=1（表示使用数据归一化技术）

对比观察tensorboard中以上两种情况在训练数据集上的Total\_cost的learning Curve图（即观察Loss在不同epoch时有何不同），并尝试归纳总结出数据归一化技术对Total\_cost的收敛速度的影响。

**2、学习率对收敛速度的影响。**

2.1）采用BGD。超参数设置如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| training\_epochs | BatchSize | N | TestingDataRatio | OrderNum | Seed | DataScaleFlag |
| int(2000) | N\*(1-TestingDataRatio) | 500 | 0.2 | 2 | 16225151 | 0 |

当学习率lr分别为以下4种取值时，观察tensorboard中训练数据集上的Total\_cost的learning curve图。尝试总结：不同学习率lr，对Total\_cost的收敛速度的影响。

* 1. 设置：lr = 0.00001
  2. 设置：lr = 0.0001
  3. 设置：lr = 0.001
  4. 设置：lr = 0.01

2.2）采用SGD。超参数设置如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| training\_epochs | BatchSize | N | TestingDataRatio | OrderNum | Seed | DataScaleFlag |
| int(1000) | 1 | 500 | 0.2 | 2 | 16225151 | 0 |

当学习率lr分别是以下4种取值，观察tensorboard中训练数据集上的Batch\_Cost和Total\_cost的learning curve（Batch\_Cost的learning curve，是指，Loss在不同Batch时的变化）。尝试总结：不同学习率lr，对于Total\_cost的收敛速度和准确度的影响。

Loss

1. 设置：lr = 0.00001
2. 设置：lr = 0.0001
3. 设置：lr = 0.001
4. 设置：lr = 0.01

**3、对比BGD，SGD和MBGD，并进一步观察分析MBGD中BatchSize对收敛稳定性和收敛速度的影响。使用基本的GD方法，**超参数设置如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| training\_epochs | Learning Rate | N | TestingDataRatio | OrderNum | Seed | DataScaleFlag |
| int(2000) | 1\*1e-3 | 500 | 0.2 | 2 | 16225151 | 0 |

1. 设置BatchSize = N\*(1-TestingDataRatio)（表示使用BGD）
2. 设置BatchSize =1（表示使用SGD）
3. 设置BatchSize = 10（表示使用MBGD，且一个echo中包含40个batch）
4. 设置BatchSize = 40（表示使用MBGD，且一个echo中包含10个batch）
5. 设置BatchSize = 80（表示使用MBGD，且一个echo中包含5个batch）

在tensorboard中对比以上5种情况在训练数据集上的Total\_cost和Batch\_Cost的learning curve图，观察不同BatchSize取值时对Total\_cost收敛速度和Batch\_Cost稳定性（出现震荡，即意味着不稳定）的影响。运行代码，将观察到如下现象：

1. Batchsize=80时，Total Loss收敛最快。
2. BGD下，Total Loss稳定的下降，不断朝极小值接近，但收敛速度慢。
3. SGD和Batchsize=10这两种情况下，很早就出现了Total Loss值不再下降的现象，但实际上此时并没有找到Total Loss的极小值。

Loss

1. MBGD下，随着Batchsize的取值不同，Batch\_Cost出现了不同程度的震荡。

Q1：现象3）中，为何会出现Total Loss值不再更新的现象呢？

Q2：现象4）中，为何会出现震荡？

Q3：你觉得以上5种不同的BatchSize中，哪种取值最好？为什么？

**4、验证常用梯度优化算法的效果。**

超参数设置如下：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| training\_epochs | Learning Rate | BatchSize | N | TestingDataRatio | OrderNum | Seed | DataScaleFlag |
| int(2000) | 1e-3 | N\*(1-TestingDataRatio) | 500 | 0.2 | 2 | 16225151 | 0 |

采用不同的梯度优化方法，比如AdaGrad,Adam,Moment算法，在tensorboard中对比这3类优化方法在训练数据集上的Total\_cost的learning curve图，并对观察到的结果进行解释。**注意，**对于AdaGrad和Adam算法，可以增加Learning Rate，将Learning Rate设置为0.1，将可加快收敛速度。

Q：通过观察分析，你认为，对于当前代码，哪种梯度优化算法最好？为什么？

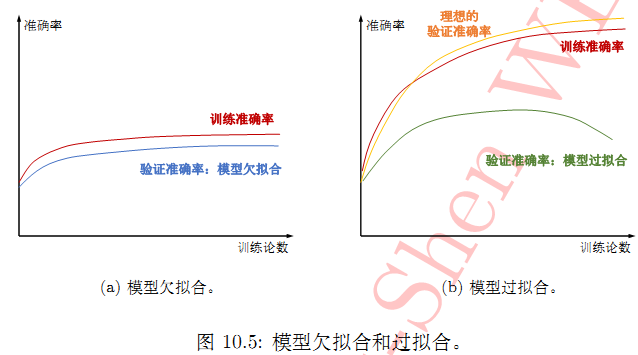
**5、数据集和验证集CV划分对“过拟合”的影响。**采用BGD作为梯度下降方法。

超参数设置如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| training\_epochs | Learning Rate | BatchSize | N | OrderNum | Seed | DataScaleFlag |
| int(2000) | 0.001 | N\*(1-TestingDataRatio) | 50 | 3 | 16225151 | 1 |

对于以下三种CV情况，在tensorboard中分别对比观察训练数据和测试数据上Total\_cost的Learning Curve图。请指出哪种情况下出现了过拟合（模型过拟合的判断依据），并尝试总结训练数据集的数据量对过拟合的影响。

1. 设置：TestingDataRatio = 0.2
2. 设置：TestingDataRatio = 0.5
3. 设置：TestingDataRatio = 0.8



**二、跨越不同模型进行比较，从而找出终极的最佳function。**（对应的代码为：AI\_StuAsssign\_CrossModel.py）

使用Adam作为梯度下降优化方法。

超参数设置如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| training\_epochs | BatchSize | N | TestingDataRatio | Learning Rate | Seed | DataScaleFlag |
| int(1e20) | N\*(1-TestingDataRatio) | 500 | 0.2 | 0.1 | 用自己的学号的数字部分 | 1 |

1、分别定义1阶、2阶、3阶、4阶和5阶模型，找出各个模型下的最佳function，并按照以下示例的表格模板来记录最佳参数组合、最佳function在训练数据和测试数据上的损失值。

注意，请使用自己学号的数字部分作为随机数生成器的seed，从而保证每位同学所使用的参数W和b的初值不同，以保证每个人最终记录的表结果不一样。

示例：若采用seed=16225151，则可得到如下结果：

|  |  |  |
| --- | --- | --- |
| Model & Optimized function | Training Loss | Testing Loss |
| 1阶: y=2.01x-8.06 | 22.646 | 18.9569 |
| 2阶：y=0.99\*x2-3.92\*x-3.15 | 1.11558 | 0.996076 |
| 3阶：y=0.01\*x3+0.9x2-3.71\*x-3.16 | 1.09718 | 0.997507 |
| 4阶：y=1.01e-4\*x4+8.8e-3\*x3+0.9x2-3.72\*x-3.16 | 1.09707 | 0.997443 |
| 5阶：  y=3.96e-4\*x5-6.33e-3\*x4+3.93e-2\*x3+0.87x2-3.78\*x-3.11 | 1.09216 | 1.00331 |

2、观察上表中你的记录结果，试找出跨模型的最佳function，并说出你的选择理由。