**问题描述：**

基于随机梯度法，求解MNIST数据集上的分类问题。

首先调用pytorch下载数据集并划分好batch，关键代码如下，重要指令均含有注释，因此不再赘述。

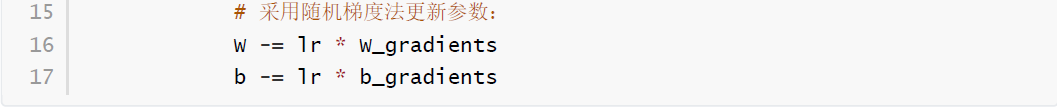


**算法设计：**

迭代公式：

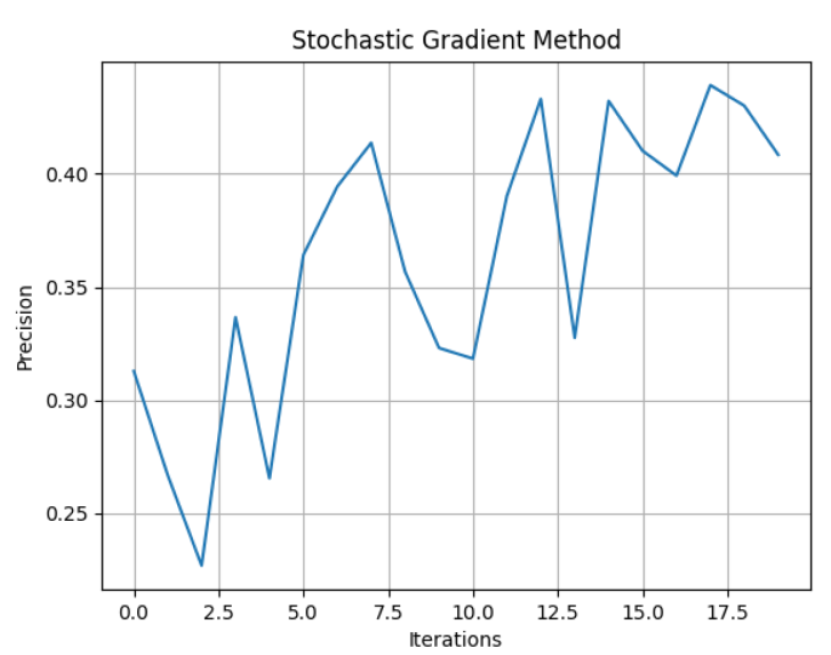
****

由公式可知，每次更新参数时，随机选择batch中某一个输入数据对应的梯度作为当前批次的梯度，从而对参数进行更新。对应的关键代码如下：

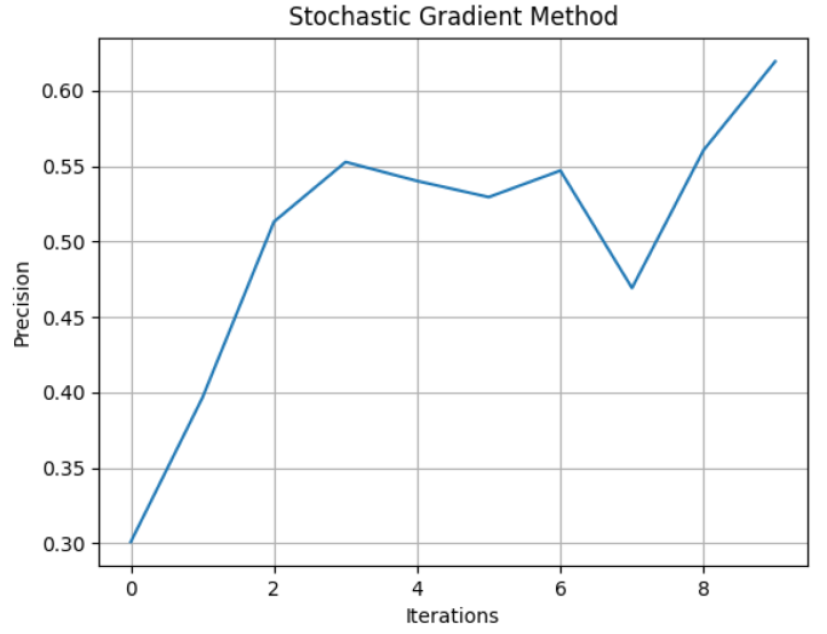
 

**数值实验：**

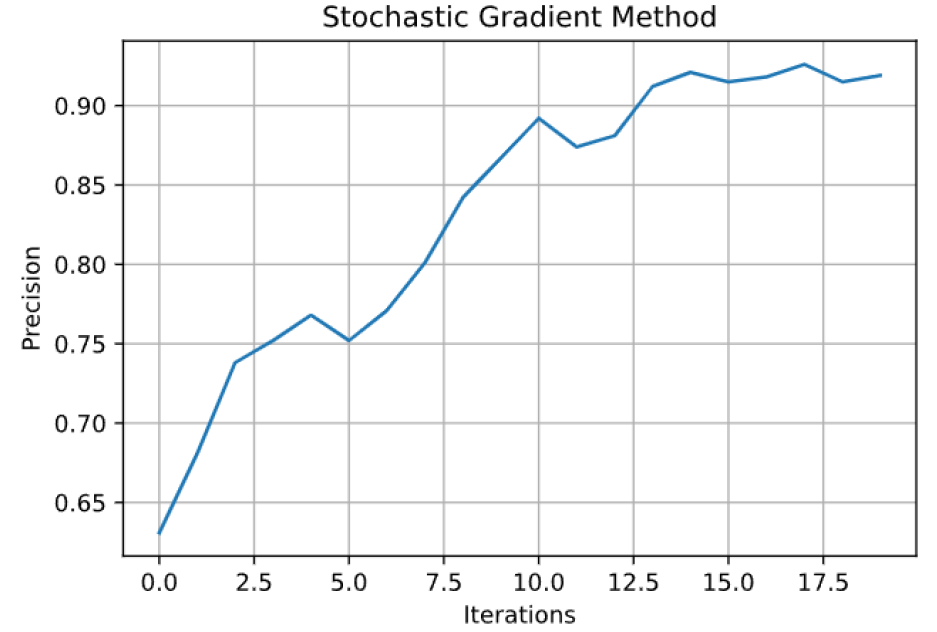
**2.1** batch size: 10000，epoch：20



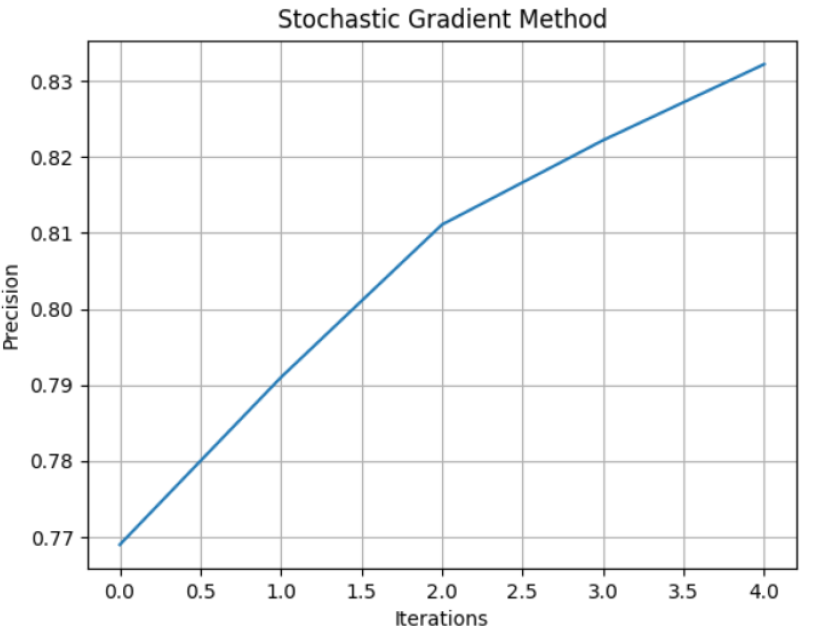
**2.2** batch size: 1000，epoch: 10



**2.3** batch size: 100, epoch: 20



**2.4** batch size: 10, epoch: 5



**2.5** 结论

在batch size: 100, epoch: 20的情况下，正确率可以到达91.58%。

**结果分析：**

随机梯度法的耗时更短，虽然算得的梯度不是十分精确，但依旧有助于模型的训练和收敛。在batch size: 100, epoch: 20的情况下，正确率可以到达91.58%。对于随机梯度法，mini-batch 大小对于结果有一定的影响：如数值实验的图像所示，mini-batch越大，每个epoch训练所需的时间越少，但是训练时模型预测正确率的波动程度越大，同时训练相同epoch的情况下，正确率也越低。