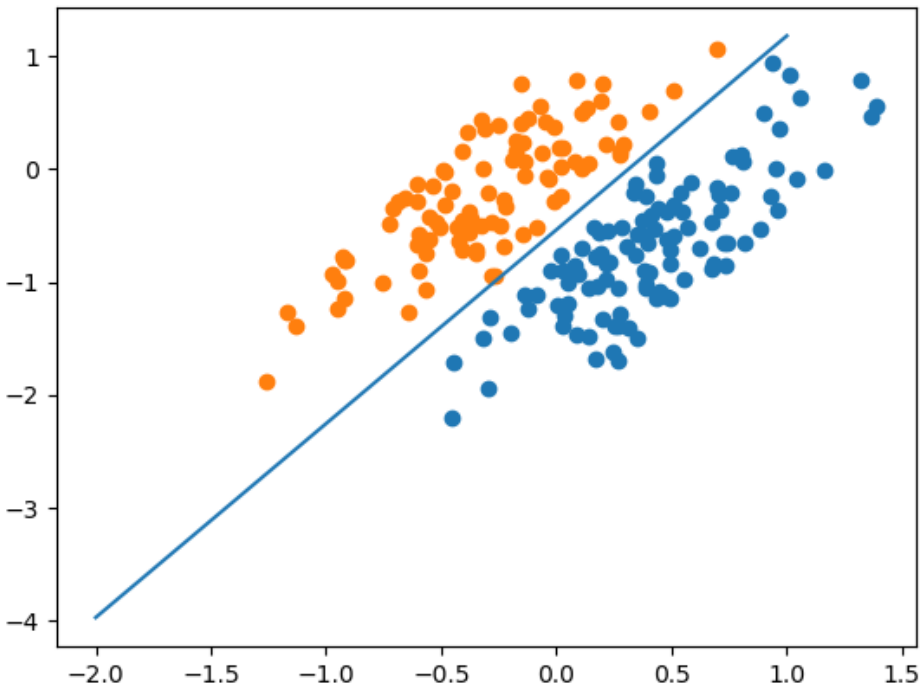
Assignment 2 报告

Task 1：

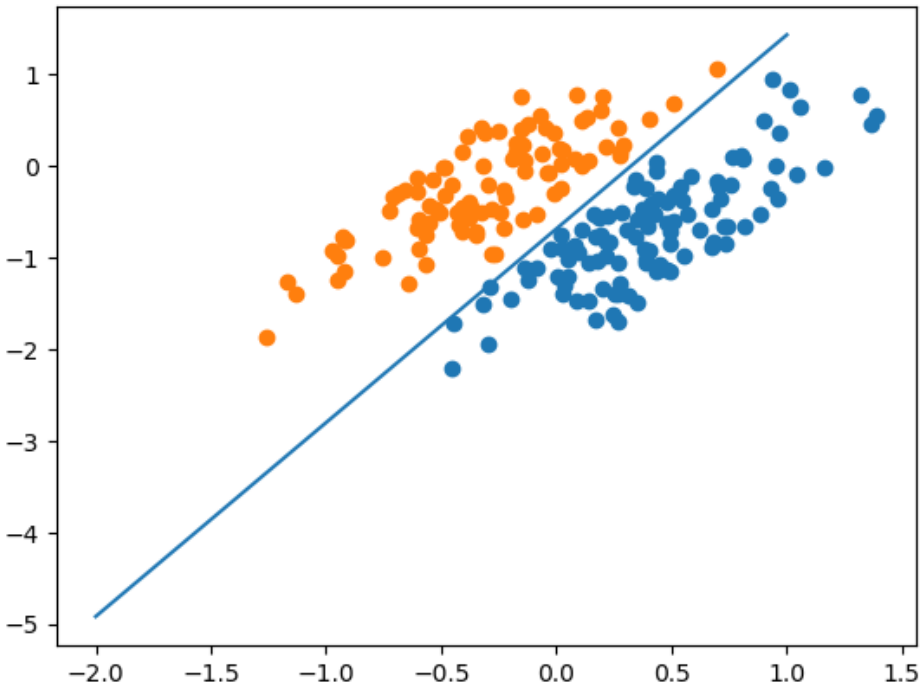
1. Least Square Method

准确率 accuracy = 100%



2. Perceptron Algorithm

准确率 accuracy = 100%



Task 2:

1. Preprocessing

预处理主要分为三个部分：串处理、串转矢量（即 multi-hot）、以及目标类转成矢量形式（即 one-hot）

One-hot 的提取是简单的。由于文章存在 sklearn 的 datasets 类中，直接访问 target 属性可以提取出类的 one-hot 形式。

Multi-hot 的提取稍微复杂一些。首先将串进行处理：小写化、去掉多余的字符。之后用 hash map（python 中的 dict）存下所有处理后的串，筛选出现次数大于等于 mincount 的串（此次 pj 中设为 10）建立字典。

建立字典之后，每一个单词 s 表示为长为 D 的 01 向量 X（D 是字典大小），第 i 位为 1 当且仅当 s 是字典里的第 i 个单词。

2. Loss function & logistic regression

首先令向量 $\mathrm{x}_\mathrm{n}' = [1, x_n^T]^T, W_n'^T = [1, W_n^T]^T$ ，其中 n 是数据集大小

损失函数

$$\mathrm{L} = -\frac{1}{N}\sum y_n^T \log softmax\left(W'^T x'_n\right) + \lambda\|W\|^2$$

通过链式法则求导，易得：

$$\frac{\mathrm{dL}}{\mathrm{dW}'} = \frac{1}{N} + (y' - y)x^T + 2 * \lambda * \mathrm{W}$$

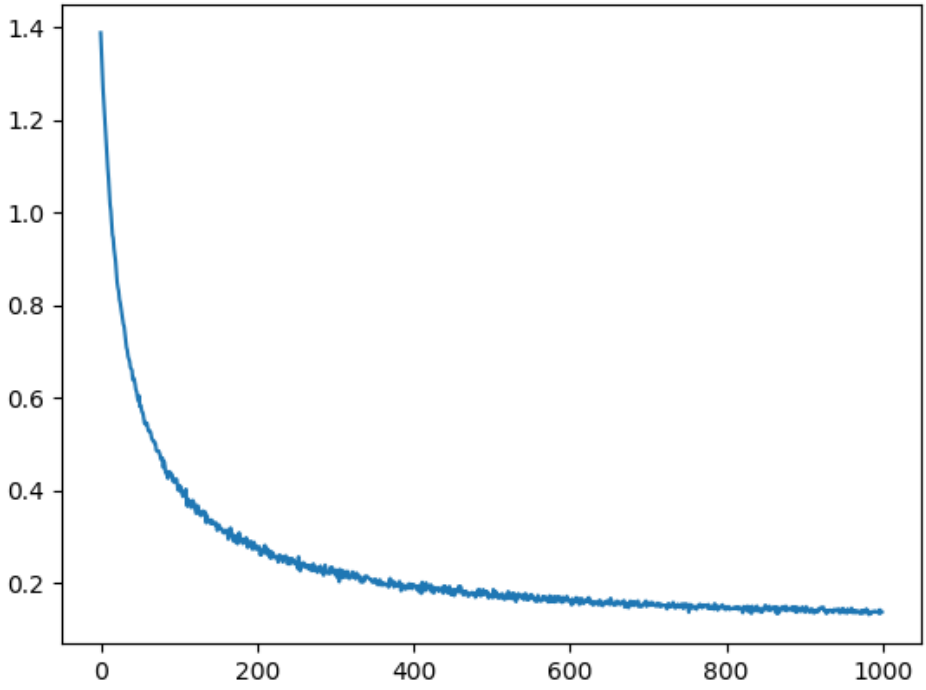
注意此处的 W′是扩容之后的 W 向量，包含了偏移量 bias。

（1） bias 不用正则化惩罚。因为对 bias 惩罚，说明希望 bias 项尽可能为 0，而 bias 项只控制模型的位置，并不控制模型的整体形状。因此不存在过拟合的情况，对 bias 进行正则化反而很可能导致欠拟合的出现

（2） 使用数值梯度计算方法，在某个点左右加减一个很小的范围ε（取ε =1e-6），计算 $\frac{F(x+\varepsilon)-F(x-\varepsilon)}{2*\varepsilon}$，检查计算结果和解析法梯度算出来的结果是否相同。

3. Training process

取 learning rate = 0.1, λ = 0.001, 迭代次数 iter_num = 1000，采用 Full Batch Gradient Descen, 得出 loss curve 如下：



(1) 适合的学习率对训练非常重要。一般来说，它需要在训练的开始能够让 loss 值迅速的下

降，同时在训练的后期并不会因为过大从而越过最优点，来回振荡导致 loss 值的上升。通过经验，本次选取的 learning_rate = 0.1，效果不错。如果有需要，后续可以通过自动调节学习率来优化训练的结果。

(2) 通过一定数量的梯度下降之后，loss 值将趋于稳定。此时继续训练意义不大，就可以终止训练过程了。

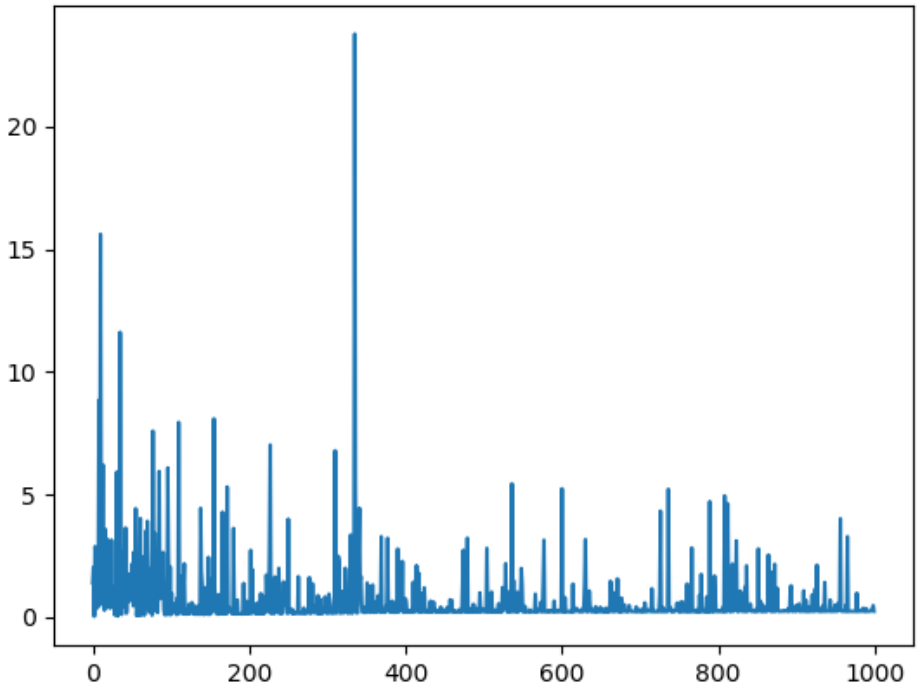
所以一种方法是设定合适的迭代次数，迭代一定次数后结束训练过程。本次试验中采取的就是这种方法，设定的迭代次数为 1000，从上图可以看到，1000 次之后 loss 值基本稳定。

另外一种方法是设定一个阈值，在 loss 值小于阈值的时候停止训练。相比于设立迭代次数，这种方法精度更高，但是对初始阈值的要求也更高。如果没设立好很有可能导致学习的结果不行或者是达不到给定阈值而导致程序死掉。

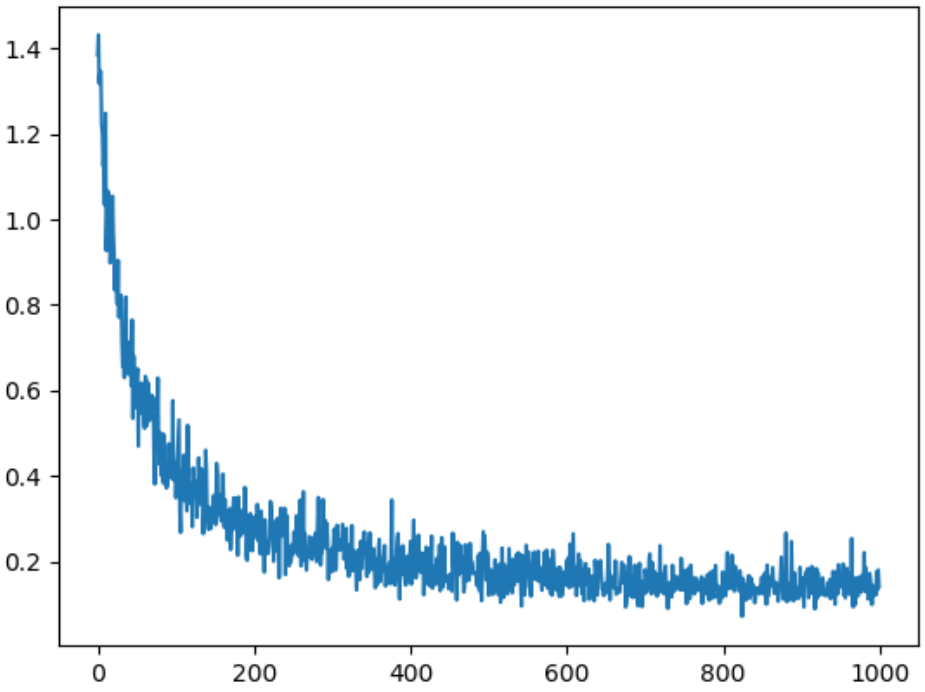
4. Different gradient update strategies

三种策略的参数相同： learning rate = 0.1, λ = 0.001, 迭代次数 iter_num = 1000，采用 Full Batch Gradient Descen。 除此之外，batched gradient descent 中的 batch_size=32。下面是三种策略的 loss curve：

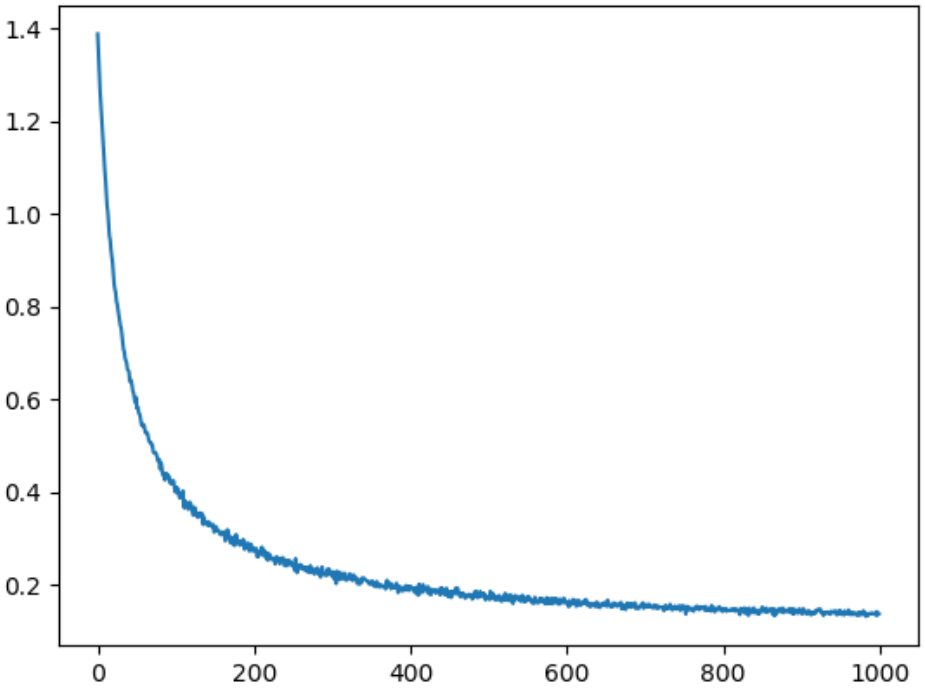
1) Stochastic Gradient Descent



2) Batched Gradient Descent



3) Full-batched Gradient Descent



整体来看， Stochastic 下降最快，到达极值点的速度也最快。但是曲线振荡幅度太大，虽然最后也把 loss 降了下来，但是几乎看不出下降的趋势。Batched 和 Full-batched 曲线趋

势接近，但 Full-batched 相比于 Batched 更加平滑，说明下降更加稳定

2) 对优缺点进行分析：

1. 从效率来看，Stochastic 的效率最高，batched 次之，full-batched 最慢。这也是由他们每次更新数据的量来决定的。
2. 从噪音的角度来看，Stochastic 的噪音过大，已经严重影响训练本身，从训练过程来看看不出训练的趋势。Full-batched 噪音最小，训练过程稳定。
3. Batched 方法整体上是 Full-batched 和 Stochastic 二者的折中，在保证较高效率的同时能够获得不错的训练结果。

5. Results

三种方法的 loss curve 如上可见。测试的准确率分别为：

Stochastic: 86.8%
Batched: 92.6%
Full-batched 92.8%

可以发现，full-batched 准确率最高，但 batched 和 full-batched 准确率相差不大。