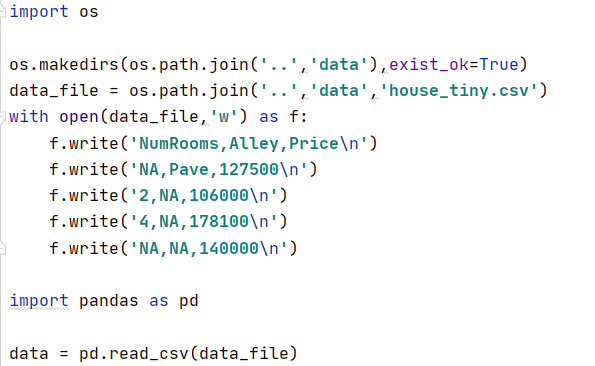
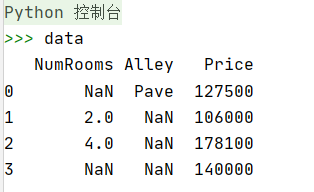
数据预处理

csv文件

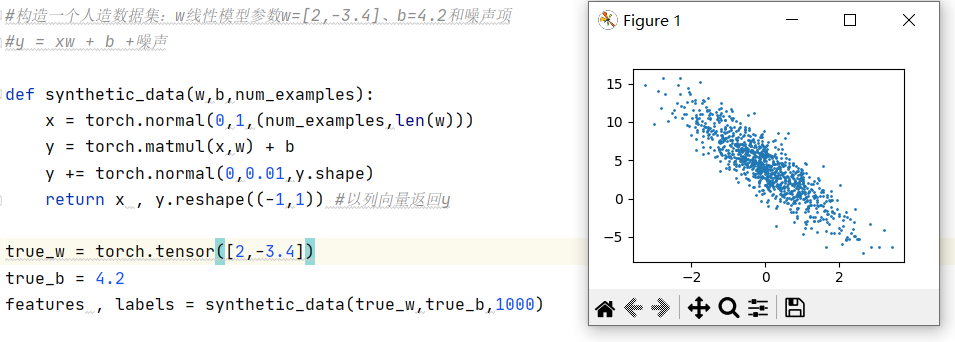
优点：纯文本格式、简单易懂；易于跨平台传输和处理；体积较少，不占空间

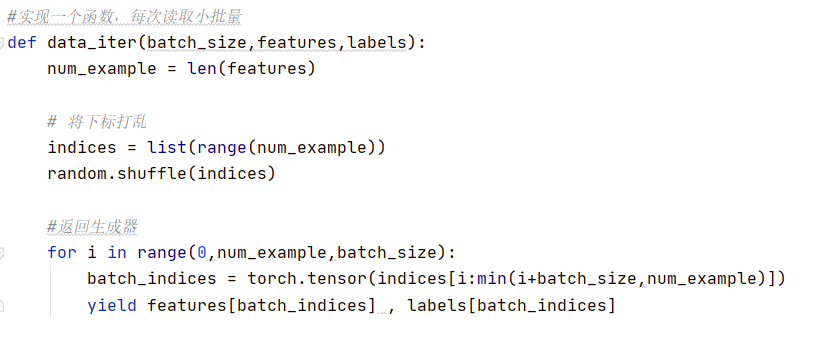
结构：每一行代表一条记录；逗号分隔；内容有逗号时，用引号把数据包围



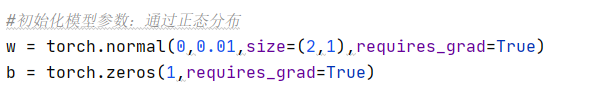
通过学习《动手学深度学习》复现代码

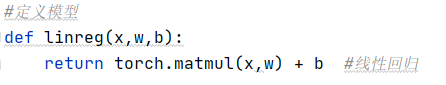
线性回归实现

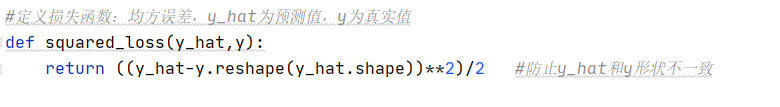
构造数据集，通过正态分布和随机数构造

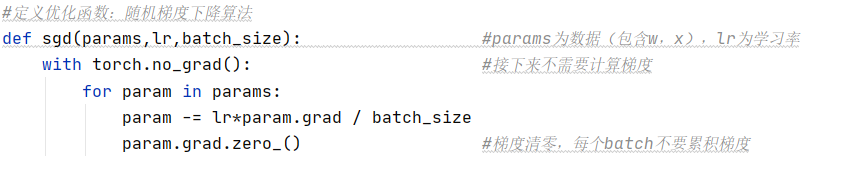


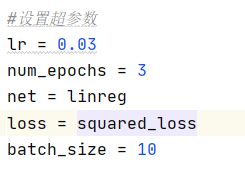
构造一个随机读取小批量数据，且能读完的函数

初始化w和b的值

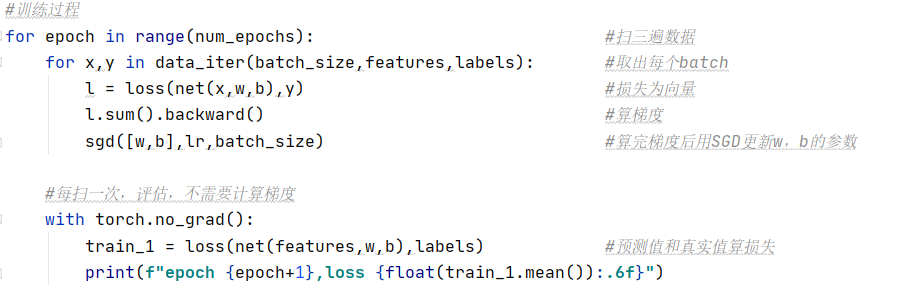
定义linear\_regression的模型

损失函数用均方误差

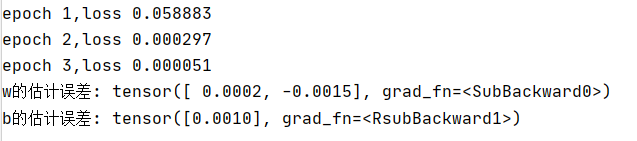
优化函数



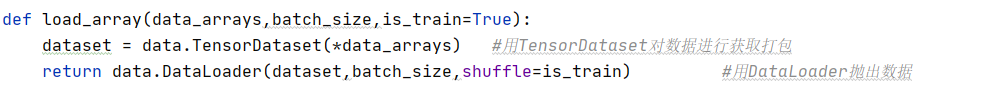
开始训练

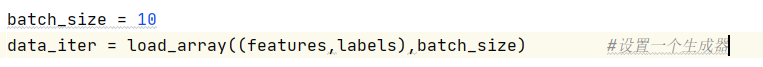


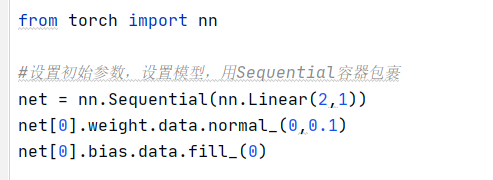
结果

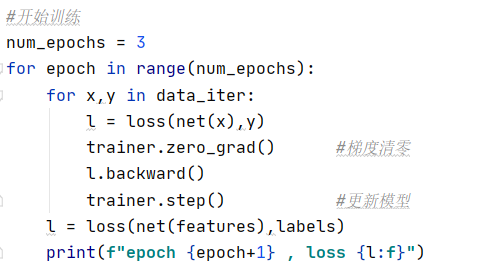


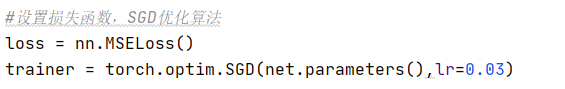
用pytoch方法更加简便



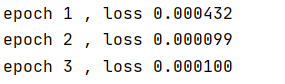




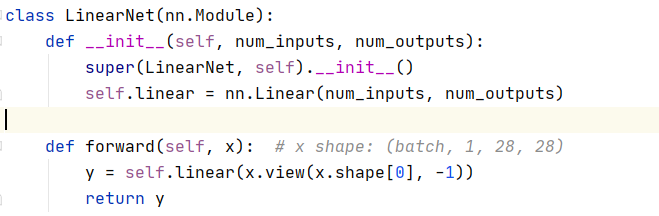


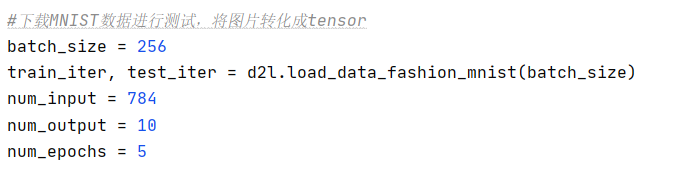


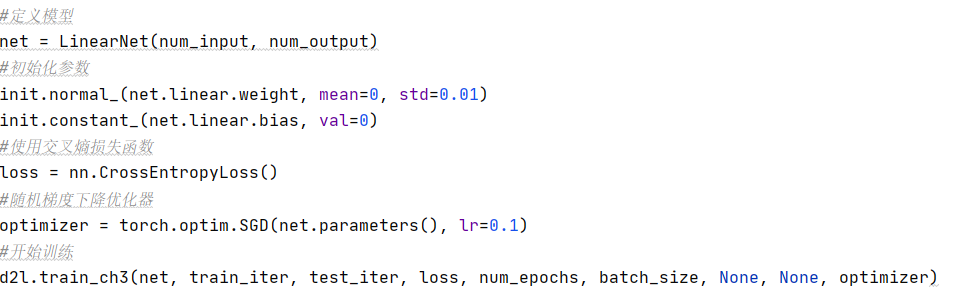
结果



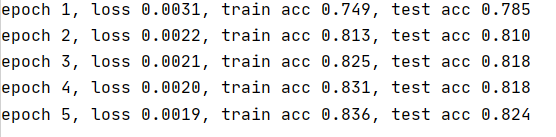
交叉熵损失：

softmax回归，使用fashion\_mnist数据集，进行图片分类





结果



train acc是训练精度，test acc是测试集的精度。

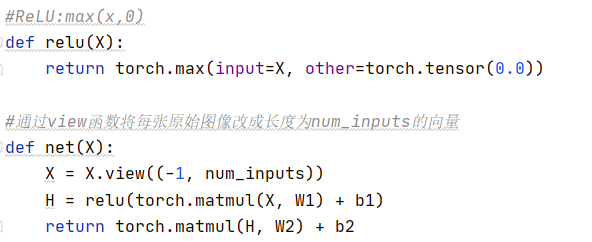
多层感知机学习，代码实现

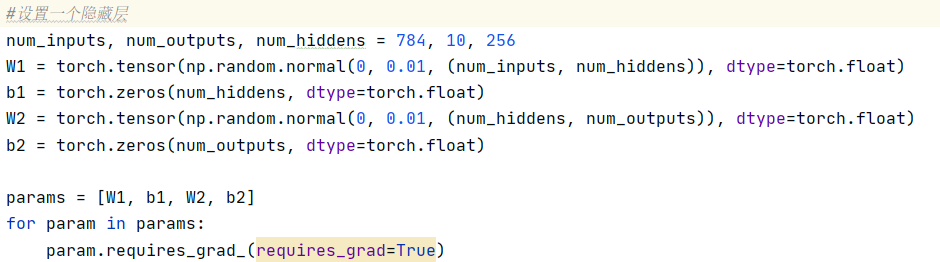
与感知机不同：隐藏层及其大小（超参数）+非线性激活函数（sigmoid，tanh，ReLU）

多隐藏层：

使用softmax来处理多分类

ReLU和网络模型

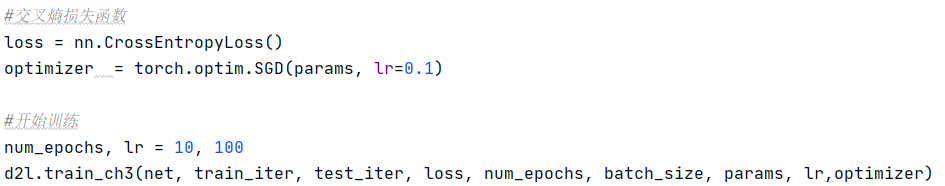




注意W1的行是784，列是256

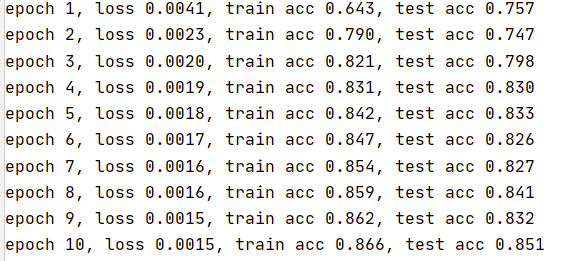
W2行为256，列为10

需要计算梯度



套用函数训练

结果



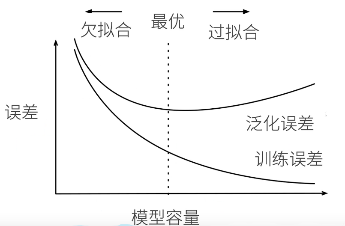
比softmax回归稍好一点

K-则交叉验证：

在没有足够多的数据时使用

算法：将数据分割成K块；使用其中一块作为验证数据集，其余作为训练集；报告K个验证集误差的平均；常用K = 5或10

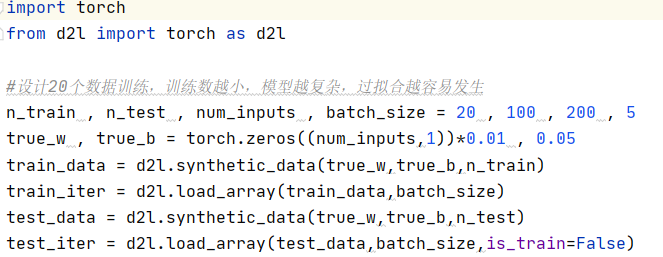
过拟合和欠拟合



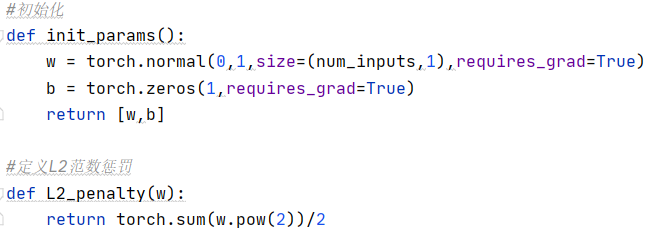
模型容量：拟合函数的能力

处理过拟合：权重衰退，增加L2正则项减小模型复杂度。（超参数lambda一般取1e-3、1e-4）

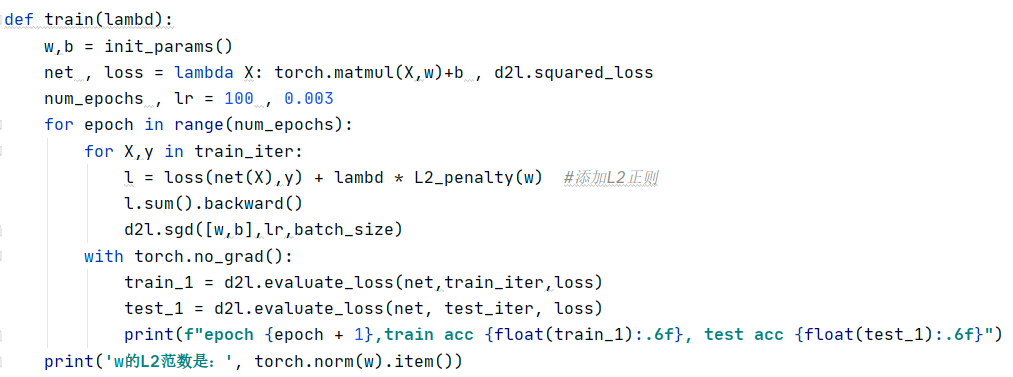
代码实现：

设置一个容易过拟合的数据集和模型

w为0.01



设置L2正则项

训练函数，添加L2正则，权重衰退

结果









范数在变小，过拟合程度减小了

若换成L1正则



结果：

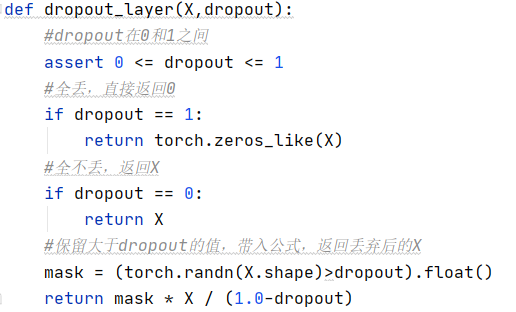




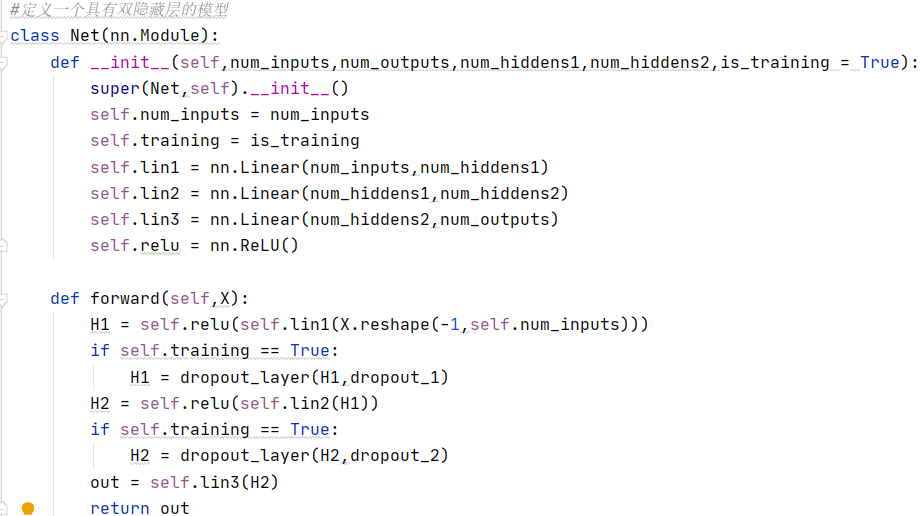
效果更好

丢弃法（Drop out）：无偏差的加入噪音

对x加入噪音成为x‘，E[x’]=x



代码，dropout只在训练时使用，相当于正则项



每层之后加一个dropout层

数值稳定性

带来两个问题：

A、梯度爆炸：

1、梯度爆炸可能会值超出值域（对于16位浮点数尤为严重（6e5-6e4））

2、对学习率敏感：过大学习率会得到更大梯度；过小会使学习毫无进展（需要不断调整学习率）

B、梯度消失：

1、梯度值变为0（学习无进展）

2、对底部层尤为严重：仅让顶部变得好；底部毫无进展

如何解决？如何让训练更加稳定？

答：合理的权重初始值和激活函数可以实现数值稳定性

或：1.乘法变加法：ResNet、LSTM

2．归一化：梯度归一化、梯度裁剪

权重初始化

1. 让每一层的反差变为一个常数：例Xavier初始（

激活函数

1. 使用ReLU比较理想
2. 调整sigmoid：4\*sigmoid(x)-2 (在x=0附近拟合y=x这个函数，会提高数值稳定性)