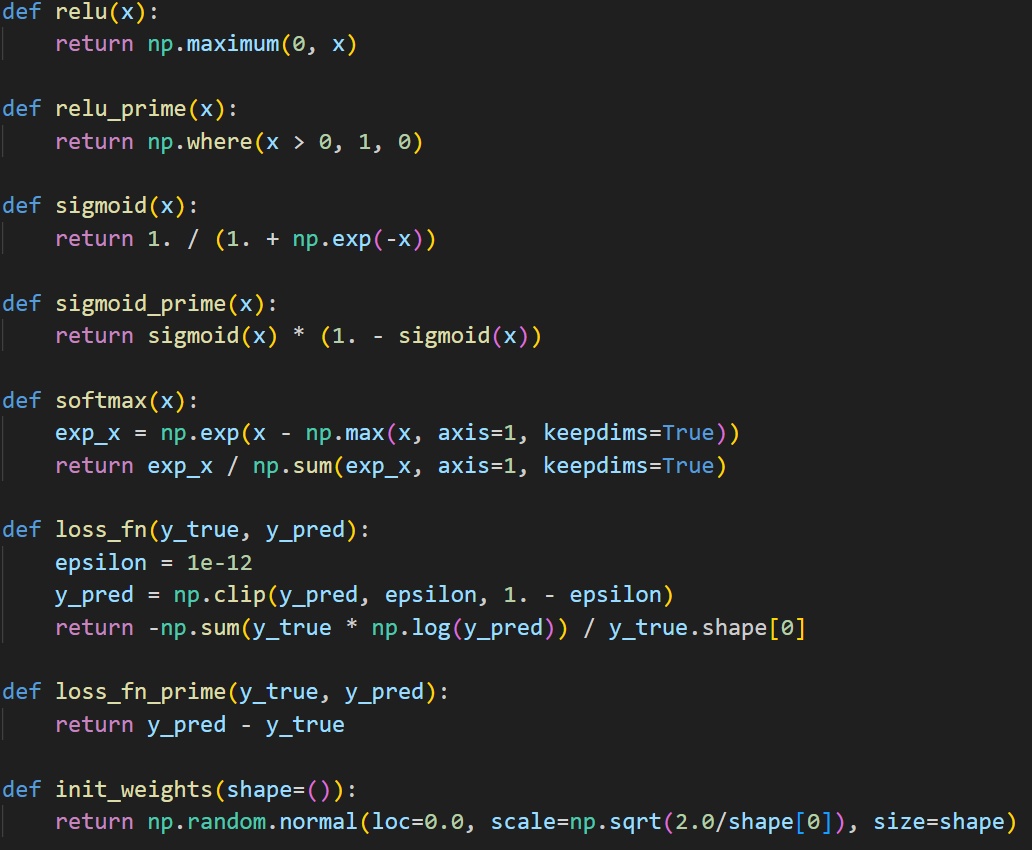
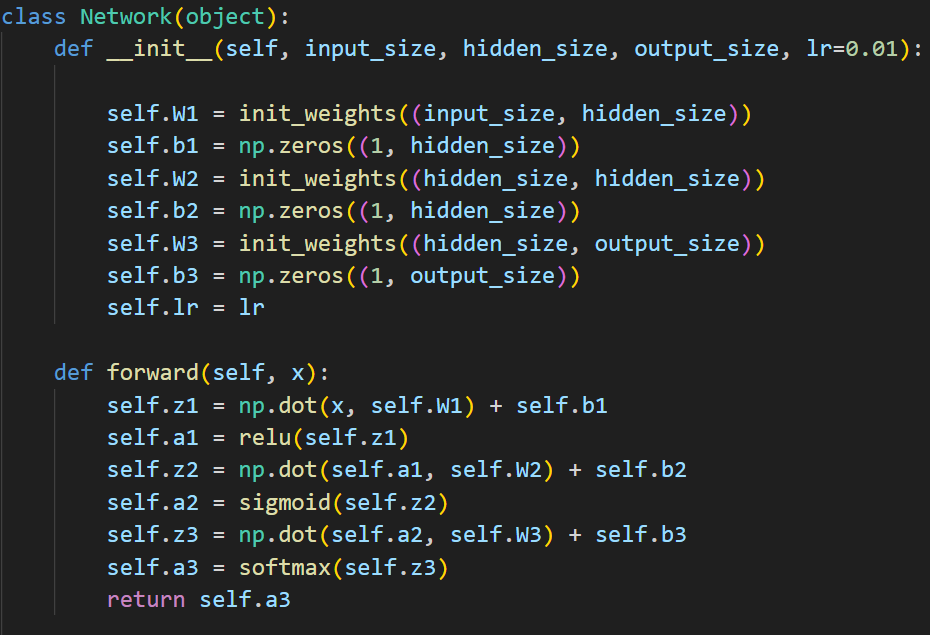
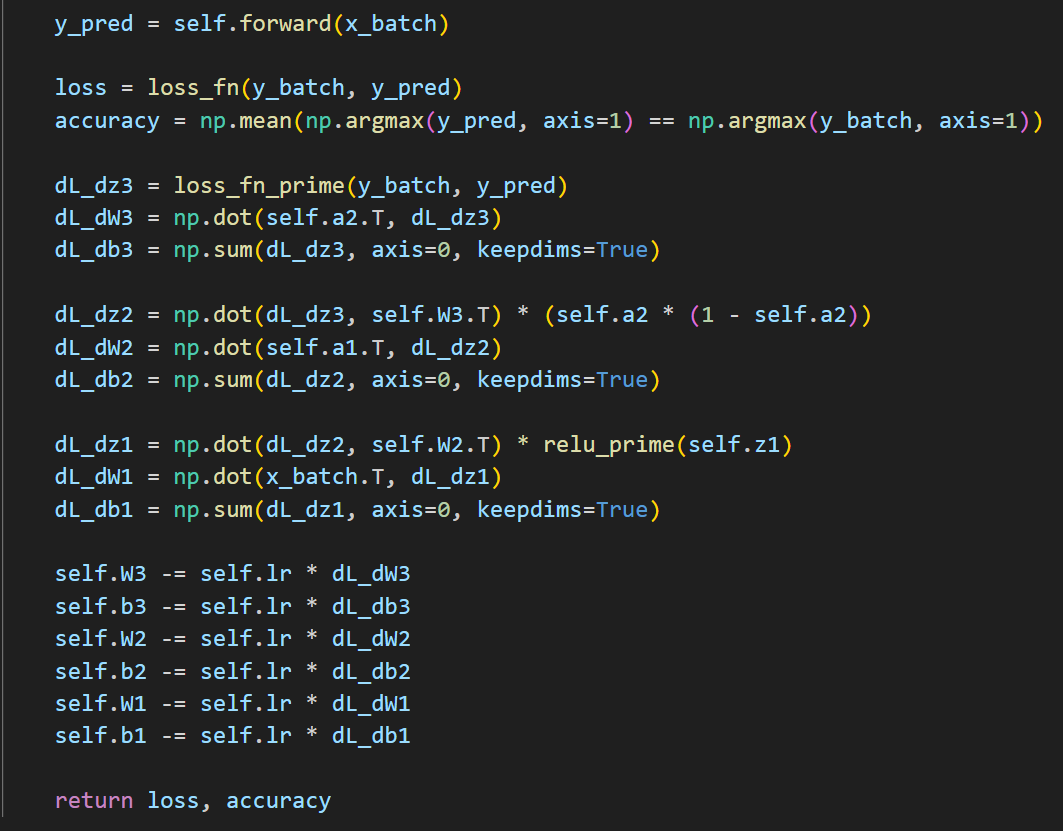
20250313 np\_mnist实验报告



函数填空部分，relu、sigmoid、softmax及其求导比较常规。损失函数采用的是多分类的逻辑回归函数。此处设定一个小数e，并将数据压缩到(e,1-e)的目的是不让0出现在求对数运算之中。注意在此后的神经网络的输出层采用的是softmax，将softmax与逻辑回归复合后，得到的求导式子简单，适合作为此处的计算函数使用。

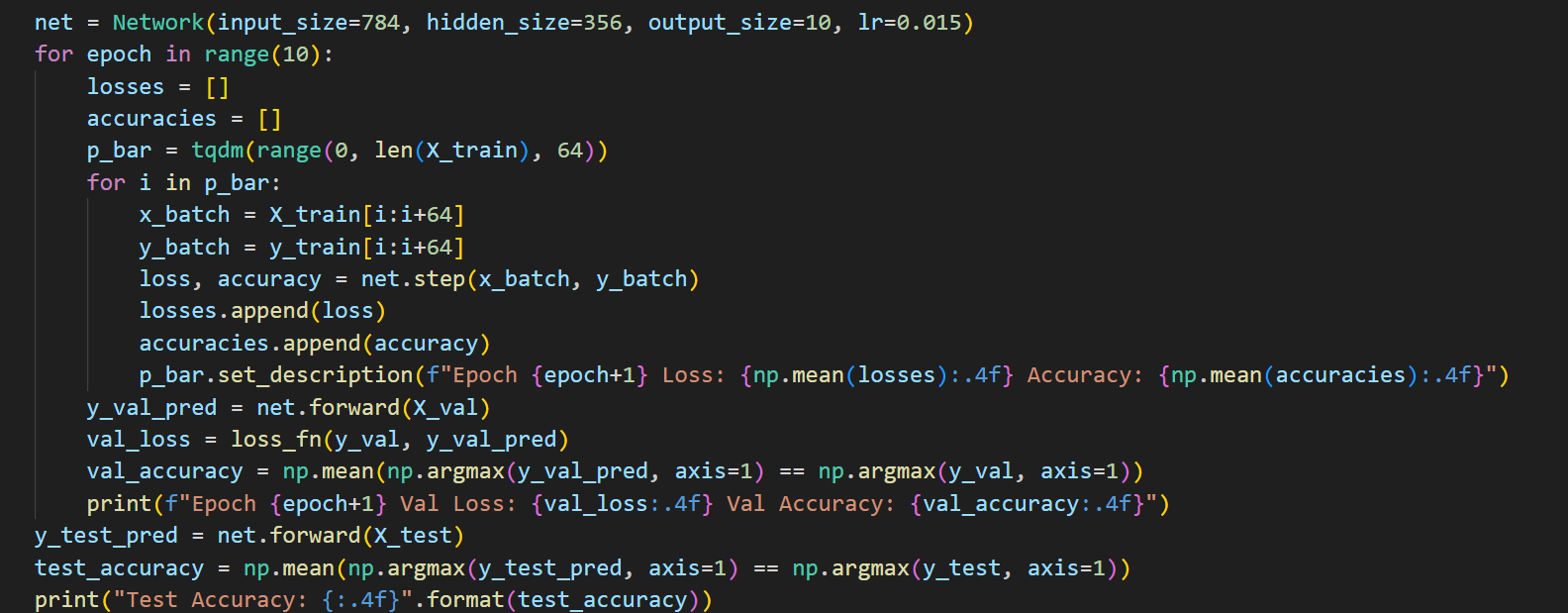
随后是网络部分：



此处采用的是四层神经网络，输入层到第一个隐藏层采用的是ReLU，第一隐藏层到第二隐藏层是sigmoid函数，第二隐藏层到输出层则是softmax。

反向传播过程，先用正向传播forward计算出y的预测值，随后开始对变量求偏导：此处所有的dy\_dx格式命名的变量都意为y对x求偏导，例如dL\_dW3就是损失函数对w3求偏导。对z求偏导参照ppt中的求导式子，而对w和b的偏导则是在对z的偏导求出后，再乘以z对w或b的偏导。对w偏导即将上一层的输出和对z的偏导相乘，对b偏导则是乘以1。最后根据梯度下降原则更新参数。

Main函数中修改的参数主要有两个，学习率和隐藏层的神经元数量。根据多次调整，神经元数量超过512后便有了明显的边际递减效应，甚至超过1024后，计算速度大大减慢，拟合程度反而还下降了。选用356个神经元作为隐藏层的规模较为合适。学习率改为了0.015，可以更快的接近局部最小值。

同时，在打印的内容中还增加了验证集的准确率实时变化，以观察是否有过拟合。

如此得到的神经网络，可以大部分时间保持在98%以上的测试准确率，同时验证集准确率也单调递增，说明没有出现过拟合。终端截图如图。

