Programming：手动实现 MNIST 上的 KNN 分类器。

1.分别用 100, 300, 1000, 3000, 10000, 30000, 60000 个训练样例进行训练，比较分类器的性能。

从 http://yann.lecun.com/exdb/mnist/下载原始数据集，并处理成 csv 格式，训练集和测试集大小分别为(60000,785), (10000,785), 标签放在(:,0), 28*28 的像素信息放在(:,1:785). 在 k=1,L2 范数距离的条件下，比较不同数目的训练样例对分类器性能的影响，数据如表 1 所示。

表 1：k=1, L2 范数时样本规模对准确率的影响

样本规模	100	300	1000	3000	10000	30000	60000
准确率%	67.94	79.23	86.90	91.91	94.63	96.18	96.91

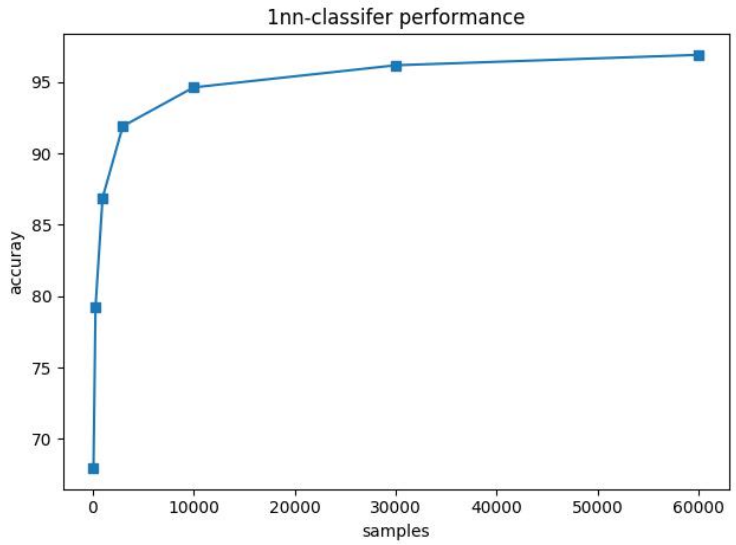

图 1：k=1, L2 范数时样本规模与准确率关系

从图中可以看出，在样本数较少时准确率不理想；随着样本数增大，准确率显著上升；样本数增大到万级后，准确率增长不再明显，达到饱和。

2.用不同的 k 比较分类器的性能。

在样本规模为 100，L2 范数距离的条件下，比较不同参数 k 对分类器性能的影响，如下表。

表 2：不同 k, L2 范数时样本规模对准确率的影响

样本规模	100	300	1000	3000	10000	30000	60000
k=3 准确率%	64.76	75.90	86.22	91.68	94.63	96.24	97.05
k=5 准确率%	62.32	74.46	85.82	91.68	94.42	96.27	96.88
k=7 准确率%	60.56	72.96	85.22	91.29	94.43	96.20	96.94
k=9 准确率%	58.38	71.46	84.61	91.05	94.08	96.04	96.59

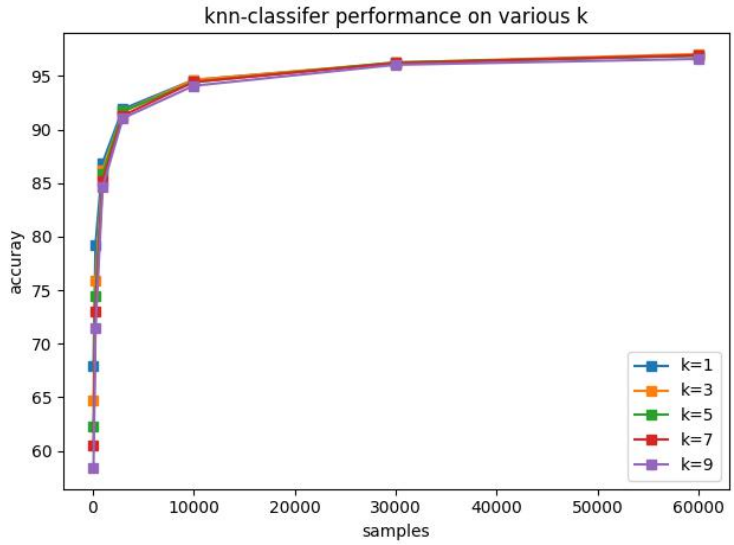

图 2：不同 k, L2 范数时样本规模与准确率关系

由图可知，k 对准确率的影响很小，5 条曲线基本上重合。在样本数较少时差异较明显，另外，随着 k 增大，准确率小幅下降，所以不是 k 越大越好，需要与样本数和具体问题相结合，选择一个最优的 k。

3.分别用 L0,L1,L2 范数距离来比较分类器的性能。
在样本规模为 100，k=3 的条件下，比较不同距离度量对分类器性能的影响，如下表：

表 3：k=1，不同范数时样本规模对准确率的影响

样本规模	100	300	1000	3000	10000	30000	60000
L2 准确率%	67.94	79.23	86.90	91.91	94.63	96.18	96.91
L1 准确率%	66.17	77.55	85.60	90.61	93.67	95.42	96.31
L0 准确率%	44.94	54.88	64.41	69.92	76.30	80.54	82.79

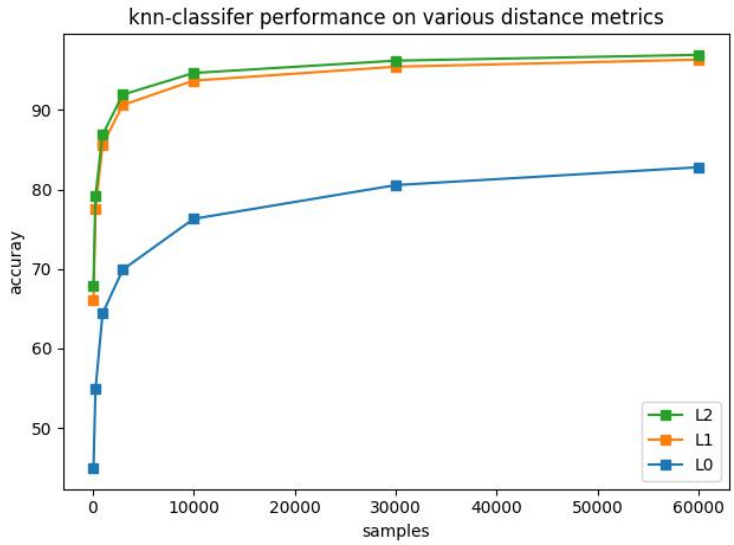

图 3：k=1，不同范数时样本规模与准确率关系

由图可知，L2 的准确率最高，L1 与 L2 相差不大，L0 则显著低于 L1 和 L2 范数距离，所以准确率与距离度量也有很大关系，需要综合考虑样本分布的属性。

总结：

本次实验使用了 cpu numpy，cpu tensor，和 gpu tensor。第一个 cpu numpy 效率最低，运行时间为几个小时；cpu tensor 效率提升巨大，运行时间为几分钟；gpu tensor 则只需要几秒钟。

但是 gpu 会报 cuda out of memory 错误，可能是(10000,60000)距离矩阵没有办法一次放入显存，可以分 batch 计算。