**Ex2 report**

自硕21 崔晏菲 2021210976

**1. KNN**

(1) 我使用了sklearn中的KNN。这个算法的原理就是对于给定测试样本，基于距离度量找出训练集中与其最靠近的K个训练样本，然后基于这K个“邻居”的信息来进行预测。

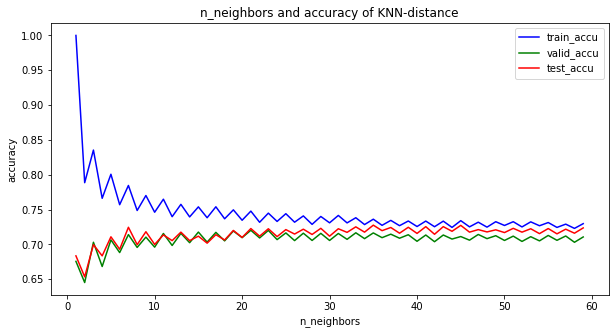
其中有一些参数需要人为设定：

n\_neighbors: 这是KNN最关键的参数，也就是“最靠近的K个训练样本”中的那个K

weights: 这是个权重参数。有”uniform”和”distance”两个选项。”uniform”的意思是，对于给定样本，训练集中的其余样本对该给定样本的影响力是相等的；而”distance”的意思是，对于给定样本，训练集中的其余样本对该给定样本的影响力并不是相等的，距离远的样本影响小，距离近的样本影响大。

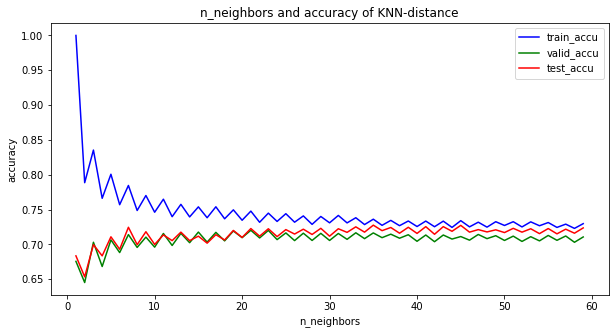
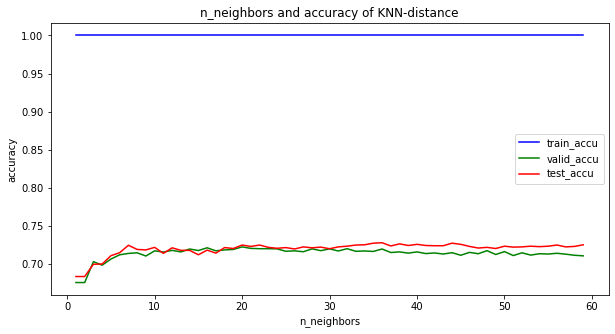
metric: 这是度量参数。我们可以选择不同的度量方式，常见的有曼哈顿距离（1范数），闵可夫斯基距离（p-范数），余弦距离等。

(2) 我使用3折交叉验证来进行训练。首先选取n\_neighbors的大小。



可以发现，n\_neighbors = 45是较优的。

接着对比weights参数：

可以发现”distance”过拟合现象非常严重，因此”uniform”参数较优。

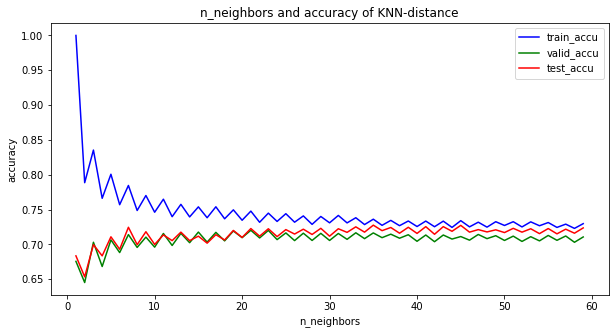
接着对比metric，取n\_neighbors=45，weights = “uniform”，结果如下：

metric = "minkowski", train\_accu = 0.7402, test\_accu = 0.7329

metric = "manhattan", train\_accu = 0.7454, test\_accu = 0.7329

metric = "chebyshev", train\_accu = 0.6540, test\_accu = 0.6253

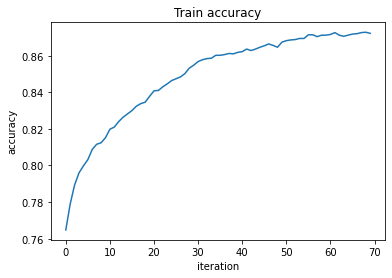
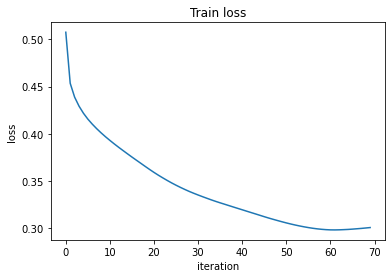
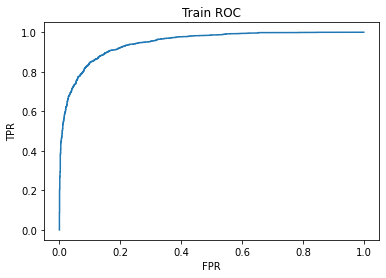
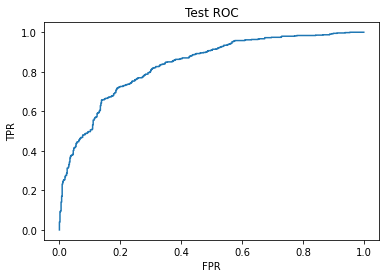
可以看到1范数和2范数相差不大，无穷范数表现较差。这也是很自然地，因为无穷范数只比较绝对值最大的维度。

(3) 

根据此图，我们可以发现，当k越小的时候，越容易对训练集过拟合，k越大，和实际情况的分割就越接近。但是k过大就会失去分辨能力。

**2. MLP**

(1) 我的MLP有两个全连接层、一个relu激活函数以及一个softmax交叉熵计算层。训练时的损失情况如下：

(2) 我采取了Batch normalization的策略，并且使用了动量加速的minibatch梯度下降算法，不论是收敛速度还是最后的准确率都有比较好的提升。