基于ELM主动学习在Spark上的实现

1. 初始阶段，将有类标的数据集保存在trainHasLabelInputDataRDD中，将隐藏类标的数据集保存在trainHideLabelInputDataLineRDD中；
2. 将有类标的数据集放在广播trainHasLabelList1中；
3. 对读入的隐藏类标数据集trainHideLabelInputDataLineRDD执行mapPartition操作，具体的操作是：将广播中的有类标数据集用ELM模型训练，再用训练得到的参数计算分区内部各个样例的信息熵，并以<熵值，样例>的形式保存在HashMap中，根据熵值将HashMap对象重新排序，并将前L个熵值最大的HashMap封装成iterator对象保存在resulttrainHideLabelRDD中；
4. 对trainHasLabelInputDataRDD和resulttrainHideLabelRDD执行union操作，生成updateTrainHasLabelInputDataLineRDD。此时，实现了将隐藏数据集中熵值最大的前L个数据放入到有类标数据集的操作；
5. 将更新后的有类标的数据集保存成文件存储在HDFS上；
6. 将之前生成的广播删除，生成新的广播，广播的内容为更新后的有类标的数据集；
7. 循环执行2-6步，循环50次，默认从隐藏类标数据集中选取50L个样例放置到有类标的数据集中。