一、时间安排

《Computing With Spatial Trajectories》 8h

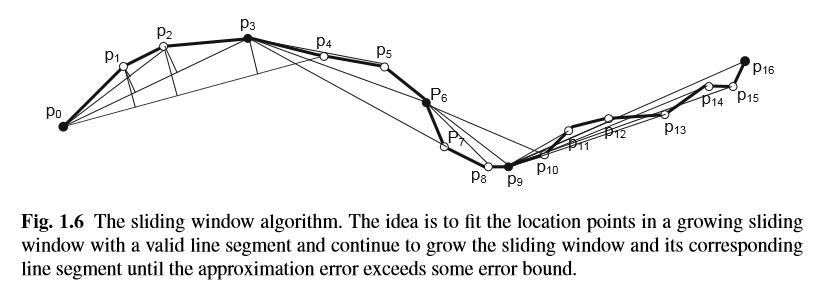
《神经网络与深度学习》 30h

二、心得

继续阅读《Computing With Spatial Trajectories》，并把之前跳过的介绍数据压缩在线处理方法的章节看完了。

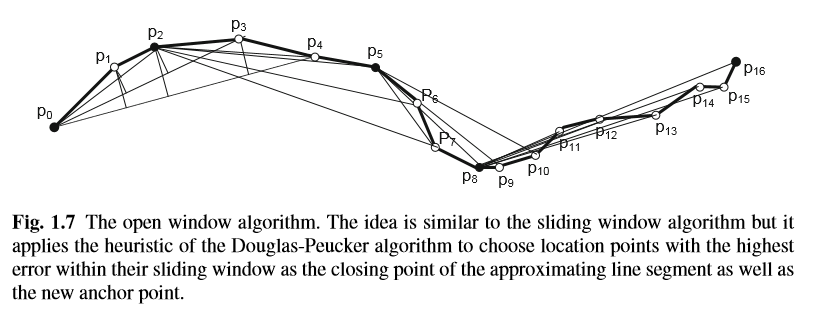
在线处理分别介绍了以下集中算法

1. reservoir sampling algorithm：固定压缩后的数据点数R（即reservoir蓄水池），对于第k个数据点（k>R），按照概率R/k判断是否加入蓄水池，若结果为加入，随机从蓄水池队列的R个数据点中替换一个，并加入队尾。
2. sliding window algorithm：



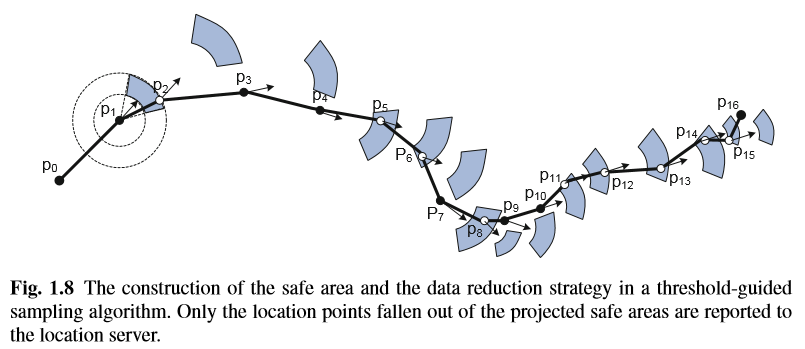
首先，设置一个阈值th，将数据点依次加入窗口，如{p0}，{p0,p1}，{p0,p1,p2}，{p0,p1,p2,p3}，{p0,p1,p2,p3,p4}；当向窗口加入一个点k后，窗口内的某个点对应的垂直欧氏距离大于th，如加入p4后，p2到p0p4的距离大于th，则将上一个加入窗口的点p3作为数据压缩的保留值（上图实心点所示），以及新窗口的起点；迭代重复上述步骤。

1. NOPW（Normal Opening Window）将sliding window和Douglas-Peucker algorithm结合：



与单纯的sliding window algorithm的区别在于，当向窗口加入p4后，p2对应的垂直欧氏距离大于th，取p2（第一个垂直欧氏距离大于th的点）作为数据压缩的保留值，并将p2作为新窗口的起点。

1. Threshold-guided sampling algorithm：仅由前两个点（数据压缩后的点）的时间、速度、方向，决定是否将当前数据点作为数据压缩的保留值。而非像前几种方法，关注于子集。



算法的本质是比较预测值和实际值。首先确定速度和方向的变化范围（阈值），根据前两个数据压缩的保留值（上图实现点）作计算速度和方向，根据新采集到的数据的时间点可以计算距离（速度\*时间），得到图上蓝色范围作为预测区域。若新采集到的点在预测区域内，则认为其携带的信息量少，丢弃；否则，保留。

然而这种方法会造成误差的累计，比如轨迹平滑的变化，但轨迹点都落在预测区域中，一种方法是将数据压缩后的点（相邻的实心点）的预测区域和数据压缩前的点（图上相邻两点）的预测区域取交集，作为新的预测区域。

1. STTrace algorithm：方法4中压缩率高，但在受限的内存上仍可能出问题。

先用方法4进行数据压缩至R个点（受限的内存），当又一个点通过了方法4要加入到内存中，计算替换每个点后的时间同步欧氏距离，取最小的方案。

神经网络的第一门课成看来一多半，目前的内容和之前《machine learning》课程主要有两点区别。第一，把原有来的θ替换成了{w,b}，主要造成了书写上的差异；第二、在下图推导逻辑回归logistic的梯度下降过程中，引入了神经网络中前向/反向传播的思路，这一点是之前在学习中没有考虑到的。

