Embedding生成方面，编写了存储训练后的模型model，构造出的图graph，以及图中所有结点totalNodes的接口。对于相同数据的不同超参实验测试，可以复用存储的图graph，减少构造时间；对于ctd等真值发现算法，内部主要有两类操作：（1）图内容不变的情况下，计算权重时，需要计算两点间的距离，此时可以复用第一次训练存储的model，直接调用distance函数获取距离，解决了多层循环中可能存在的重复训练的问题；（2）ctd中计算了每轮迭代后的error（利用矩阵中不同元素个数表征的），然后以error为收敛条件，利用ctd中的算法更改数据源中的数值。此时图的内容发生变化，需要从头训练。

解决的问题：

（1）jvm内核的内存问题，现在可以支持任意大的内存了。目前实验用的1500M，测试用所有点60步长的随机游走，没有溢出现象。

（2）游走时随机dropout边引发的部分点采集不到的问题：在dropout过程中，添加正则项，保证每条路径有需要采集的信息（目前是保证了有数据源source\_i以及样本点）

存在的问题：

（1）是否需要保存游走的路径。游走路径在保存时出现了array的sublist序列化失败的error，未能解决。原来的embedding是将生成游走路径与训练分离了，现在由于无法保存路径，已经耦合在一起了。游走路径的作用是查看一些特殊类型的点是否游走到了（游走到才有embedding）。

（2）ctd中distance函数的特殊情况。目前采用的时，如果待比较的两个字符，如果任意一个为空，返回0，否则：（1）ctd算法（不太清晰）（2）采用embedding计算。

Ctd算法原文：

When the number of values provided by different sources is quite different, we can normalize the overall distance of each source by the number of its providing values.

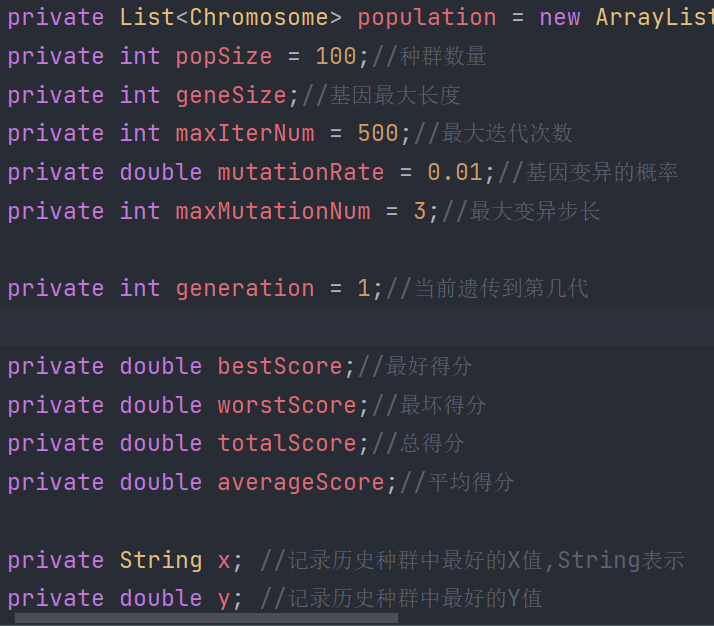
Distance内部实现也根据需要重新训练或读取现有模型，利用flag分成了两个部分。

遗传算法部分：

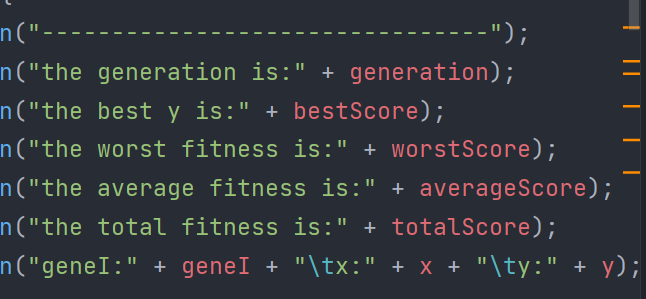
已经实现了遗传算法和embedding的嵌套：遗传算法分为=为染色体类和GA算法实现类。

染色体类包括基因序列的生成（目前是随机生成，待改进），基因克隆，初始化（主要是长度，染色体长度是确定的，需要人为设定），交叉遗传产生下一代（交叉的分割点需要人为设定，不能破坏染色体携带的超参结构），基因变异，基因表达（把基因编码还原为超参数值）。

GA算法实现类：首先超参有如下所示：



可计算的数值包括：



此部分最重要的是计算种群适应度，里面需要首先将基因序列显示表示为超参，然后把超参带入我们的embedding模型中，作为参数进行训练（注意，这里需要保存模型，文件吗，例如图之类的，需要捋一下逻辑进一步更正）这部分的显示工作可以通过命令行反馈提交给模型的信息，便于监督。

下一步，利用适应度函数计算种群中基因是否合适或被淘汰。这部分的输入信息是上一步模型反馈的结果，比如rmse等，然后带入咱们论文中的适应度函数，包括尾测试和那个Q的函数，计算出数值赋给染色体。最后优胜劣汰，迭代。

遗传算法套在ctd的distance中，用于寻找每轮最好的参数组合（？