第一部分 神经网络部分 吴枫

神经网络的输入输出：

|  |  |  |
| --- | --- | --- |
|  | 编号 | 含义 |
| 输入 | 1 | 速度的大小(m/s) |
| 2 | 期望方向与向量(1, 0)的夹角(弧度) |
| 3-7 | 与周围5个人（3m内）的x方向的相对距离 |
| 8-12 | 与周围5个人的y方向的相对距离 |
| 13-17 | 与周围5个人的x方向的相对速度 |
| 18-22 | 与周围5个人的y方向的相对速度 |
| 输出 | 1 | 下一时刻的x方向的速度 |
| 2 | 下一时刻的y方向的速度 |

1. 第一个输入是速度的大小，不包括方向，因为要把自己的速度逆时针旋转到x轴正方向上（向量(1,0)），旋转的角度记为，。
2. 期望方向逆时针旋转后得到，得到，与向量(1,0)的夹角作为第2个输入。
3. 得到第3-12个输入：
4. 先用下式计算相对距离：

对应第3-7个输入，对应第8-12个输入。(x, y)是待计算下一时刻速度的人（P）的位置坐标，(xi, yj)是P周围5个人的位置坐标。

，

是P考虑的半径，这里设成3m。就是说只考虑P周围3m内的距P最近的5个人对P的影响。

和在[0,1)之间取值。周围的人距离P越近，w值越接近与1。所以当周围的人数不足5人的时候和取0。

1. 把1)得到的相对距离逆时针旋转，作为第3-12个输入。
2. 得到第13-22个输入：

按下式计算相对速度。

，

同样逆时针旋转后，作为第13-22个输入。

1. 通过神经网络得到的输出是旋转过后的。所以再顺时针旋转得到P的下一时刻速度。

神经网络部分需要实现下面的方法：

next\_vel(第1-22个输入)：返回下一时刻的速度。

神经网络部分接收的输入和输出都是经过旋转后的，不用考虑旋转的问题。

神经网络部分调用get\_data()得到数据集，需要自己划分训练/测试集。

第二部分 扩展Circle 张哲维

在原来的Circle的基础上，需要新增一些方法来产生神经网络的输入。

compute\_next\_by\_nn()：这个方法的功能需要与原有的compute\_next ()相同，也就是计算这个人下一个速度和位置。这个方法需要产生神经网络的输入，并调用next\_vel(第1-22个输入)（得到的速度需要旋转）得到某个人下一时刻的速度。

第三部分 数据处理 蒋锋

数据先用#00101-#00103吧.

一个文件夹下面有26个文件，就是26个人的位置情况。一个文件是一个人的位置情况，文件里的一行是一个时刻的位置坐标。时间间隔是1/15s。

1. 先计算速度。
2. 按前面所说的进行旋转得到训练样本。其中，第二个输入需要得到期望方向，这里需要创建一个Scene对象，并且像之前那样调用QuickPathFinder的get\_direction(scene, source)方法得到期望方向。具体操作：
3. 创建scene对象
4. 调用QuickPathFinder.path\_finder\_init(scene)
5. 调用QuickPathFinder的get\_direction(scene, source)，source是需要计算期望方向的那个行人。

旋转的部分调用Vector2D的get\_rotate\_angle()得到需要旋转的角度，调用rotate(angle)进行旋转（返回一个旋转后的Vector2D，原来的向量不改变）。

数据处理部分需要实现下面的方法：

get\_data()：这个方法可以返回这样的矩阵，矩阵是n\*m，每一列是神经网络的一个输入(22个数)和输出(2个数)，因此n=24，矩阵有m列，每一列都是一个训练/测试样本。

数据处理部分不需要考虑划分训练/测试集。

第四部分 陈治齐

完成Vector的旋转的2个方法。