生活·技术·探索

# 借助AI模型预测障碍物方向

谢作如　浙江省温州中学
王博伦　上海人工智能实验室

**摘要：**用机器学习来解决数学回归问题，是人工智能教育中的基本实验之一。本文以创客活动中障碍物方向测量这一问题为例，介绍了使用深层神经拟合复杂三角函数公式的过程。作者提出，学生在不了解三角函数知识的情况下，也能够用神经网络来解决真实世界中的数学问题，这为学生开展科创活动提供了新的思路。

**关键词：**BaseNN；深度学习；AI科创活动

**中图分类号：**G434　**文献标识码：**A　**论文编号：**1674—2117（2023）05—0098—03

## 问题的提出

在2017年时，笔者曾和诸暨的李琦老师一起做过一个跨学科学习案例，研究用超声波传感器来识别前方障碍物的方向，并且写了一篇名为《基于距离差的方向跟踪》的文章（刊登在本刊2017年第7期）。这一研究的原理，是利用三角函数的知识，通过障碍物和不同传感器之间的距离来计算出角度。而根据超声波传感器来测量障碍物的方向，可以应用于无人驾驶、目标跟踪等场景。

近年来，有不少义务教育阶段的创客教师与笔者讨论，认为文章列举的案例是很好的学习案例，可惜小学和初中的学生没有学过三角函数，无法进行研究。前段时间复习AI知识，想起神经网络在理论上可以拟合各种曲线，那么，是否可以用一个深层的神经网络来模拟这个计算公式呢？如果可以，即使学生没有学过三角函数，也能用训练模型的方式来完成障碍物角度的计算。有了这个想法后，笔者立刻开始着手验证。

## 实验设计和可行性分析

测量障碍物方向的方法有很多。2017年时采用的方法是通过距离差，根据平面几何原理计算出被测物体与观察者正前方视线的水平夹角。如图1所示，障碍物和超声波传感器1、2形成了一个三角形，只要测量出障碍物和超声波传感器1、2的距离（D1和D2这两条边），就能计算障碍物和超声波传感器之间的中点C的夹角α（即方向）。

从图1中可以看出，决定夹角α的变量有三个，除了D1和D2，还有两个超声波之间的距离。为了减少数据量，可以先固定这一距离，那么此问题就可以转换为一个典型的数学回归问题：找出两个输入量和一个输出量之间的关系。我们可以用机器学习的方式，借助现有的真实数据，训练一个AI模型来“推测”出这三个量之间的关系。

## 数据的采集

现实中有多种方式采集数据。第一种是最真实的，即在障碍物和中点之间拉一条线，然后读取两个超声波传感器的数据，同时测量角

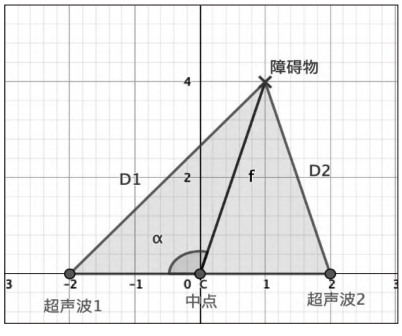


图1

度并记录；或者是拉三条线，因为超声波传感器的数值和真实长度误差是很小的。

当然，因为这一角度是可以用三角函数计算的，那么最方便的数据采集方式莫过于用Python写一段代码，然后将一组数据输出到CSV文件中，或者使用Excel的公式来计算，再导出关键数据，如图2所示。

除了做实地测量外，还有没有其他简便的办法呢？答案当然是有，可以借助geogebra来获取数据。如图3所示，只要画出这个图，用鼠标拖动两个滚动条的圆点，即可看到D1和D2，以及角α的数据在不断变化。可以用100条数据作为训练集，再用20条作为验证集，这样就完成了数据集的整理工作。

这里需要强调一点，因为最后输出的角度是在0~180之间（真实情况会小很多），这和输入的三组数据在数据大小上差距较大，容易导致训练时Loss值不正常，甚至无法收敛，因此最好使用弧度值来训练。通过实际测试可知，只要除以18，把数据范围控制在0~10之间，训练时就很正常。

### 网络搭建和模型训练

笔者选择用BaseNN来搭建神经网络。BaseNN是神经网络Python库，延续了MMEdu极简的训练流程和风格，聚焦于基础原理的探究和模型的快速应用。BaseNN类似Keras，只不过底层是Pytorch，初学者用简短的代码即可搭建出各种网络模型，调整神经元的个数、层数、激活方式等。

1.载入数据集（如图4）

2.网络搭建

笔者准备用一个最基础的全连接神经网络来解决这一问题。BaseNN的代码很简洁，几乎类似伪代码。下面的数行代码搭建了一个三层的神经网络，输入维度是3（3列数据），最后输出维度是1（1列数据），隐藏层是30和6，激活函数使用ReLU（如下页图5）。这里的隐藏层的层数和宽度仅用于参考，可

J2 =DEGREES(I2)

| | A | B | C | D | F | H | I | J |
|---|---|---|---|---|---|---|---|---|
| 1 | D1 | D2 | 距离 | 顶角（弧度） | 左角（弧度） | 中线长 | 方向（弧度） | 方向（角度） |
| 2 | 4 | 4 | 4 | 1.047197551 | 1.047197551 | 3.464101615 | 1.570796327 | 90 |
| 3 | 3 | 3 | 4 | 1.459455312 | 0.841068671 | 2.236067977 | 1.570796327 | 90 |
| 4 | 4 | 5.6569 | 4 | 0.785390076 | 1.570812502 | 4.47216489 | 1.107135778 | 63.4342074 |
| 5 | 5.6569 | 4 | 4 | 0.785390076 | 0.785390076 | 4.47216489 | 2.034456876 | 116.5657926 |
| 6 | 4.1231 | 5 | 4 | 0.888480113 | 1.325818772 | 4.123102813 | 1.325816043 | 75.9636637 |
| 7 | 4.0112 | 5.8728 | 4 | 0.746669957 | 1.645658514 | 4.614081993 | 1.048957007 | 60.10080937 |
| 8 | 2.8284 | 2.8284 | 4 | 1.570815507 | 0.785388573 | 1.99996164 | 1.570796327 | 90 |
| 9 | 5 | 5 | 4 | 0.823033692 | 1.159279481 | 4.582575695 | 1.570796327 | 90 |
| 10 | 4.1231 | 5 | 4 | 0.888480113 | 1.325818772 | 4.123102813 | 1.325816043 | 75.9636637 |
| 11 | 2.3 | 2.7 | 4.5 | 2.236423234 | 0.491464858 | 1.10792599 | 1.368851545 | 78.42941628 |
| 12 | 2.5 | 2.8 | 4.5 | 2.026395 | 0.592882832 | 1.408012784 | 1.444992402 | 82.79196605 |
| 13 | 2.7 | 2.9 | 4.5 | 1.865597044 | 0.664471488 | 1.669580786 | 1.496190795 | 85.72541791 |
| 14 | 2.9 | 3 | 4.5 | 1.734748242 | 0.717796731 | 1.908533468 | 1.536440915 | 88.0315799 |

图2

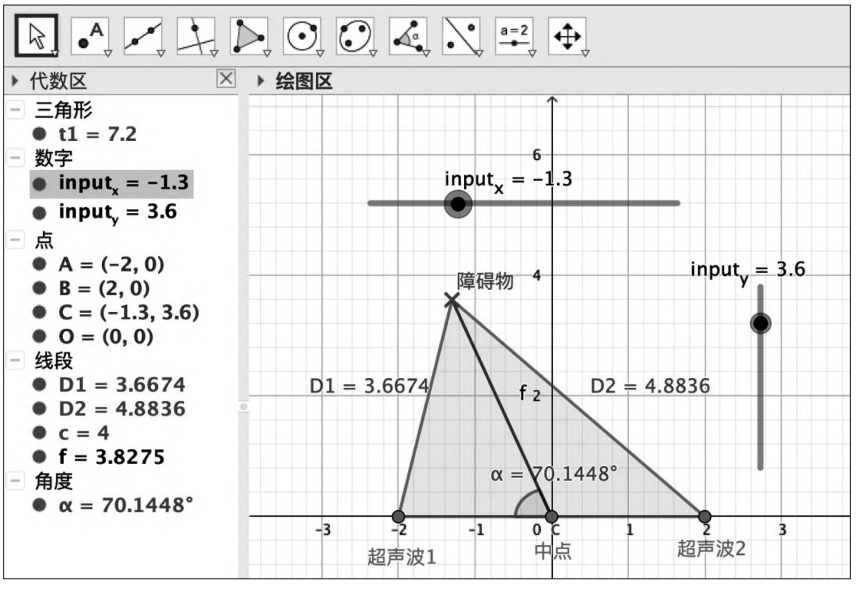


图3

```
import numpy as np
train_path = './data/train.csv'
x = np.loadtxt(train_path, dtype=float,
delimiter=',',skiprows=1,usecols=range(0,3)) # 读取前 3 列
y = np.loadtxt(train_path, dtype=int, delimiter=',',skiprows=1,usecols=[3]) # 读取 4 列
```

图4

以根据具体的情况修改。

笔者绘制了一个神经网络示意图，由于30层画起来密密麻麻，所以仅画出了15层，如图6所示。

**3.模型训练**

经过多次测试，200轮训练就能够得到较好的效果，程序如图7所示。

**4.数据测试**

接下来用model.inference函数进行推理，即可得到模型预测的结果，程序如图8所示。

为直观呈现结果，笔者画出了一张对比图（如图9），可以看出，预测结果和真实值几乎一致。

本项目访问地址是https://www.openinnolab.org.cn/pjlab/project?id=639d352b3791ab1c3aa8b987，“克隆”后即可体验。

## 反思与总结

通过这一实验，证明了用多层神经网络果然可以训练出一个能够拟合三角函数曲线的AI模型，能够根据距离差来准确预测方向，或者说，验证了“神经网络能拟合任意曲线”这一结论。但是有AI专家表示担忧：这会不会导致神经网络被滥用，明明可以用三角函数精确计算，却偏偏用神经网络来拟合？

这一担心不无道理。但任何一项新技术最开始总会被指责“滥用”，而之后才会发现其实是常态。既然神经网络能够让数学较差或者没学过较复杂公式的孩子，利用采集数据的方式进行“粗略计算”，那么就等同于他们掌握了一种万能工具，不应该以“滥用”的名义进行否认。

当然，真实世界中还存在很多有关联却无法用精确的数学公式来表示的物理量，这在科学研究和工程应用中经常会遇到。那么，用采集数据加上训练模型的方式来“拟合”多种物理量之间的关系，显然是一种有效的解决办法，希望有更多的教师去探索这一领域，让AI成为学生开展科创活动的好帮手，真正培养他们的数据意识和AI思维。

```
from BaseNN import nn
model = nn() #声明模型
model.load_dataset(x, y) # 载入数据
model.add('Linear', size=(3, 30), activation='ReLU')
model.add('Linear', size=(30, 5), activation='ReLU')
model.add('Linear', size=(5, 1))
model.add(optimizer='SGD')
```

图5

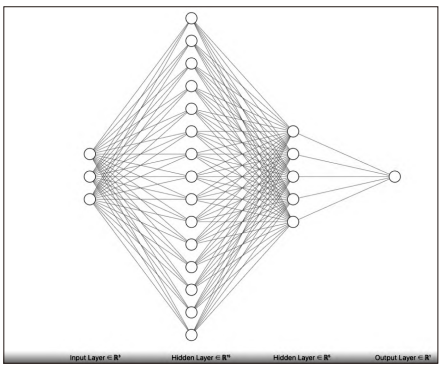

图6

```
model.save_fold = 'checkpoints/ckpt'
model.train(lr=0.001, epochs=200, loss="MSELoss",metrics=["mae"])
```

图7

```
    test_path = './data/test.csv'
    test_x          =          np.loadtxt(test_path,          dtype=float,
delimiter=',',skiprows=1,usecols=[0,1,2])
    test_y          =          np.loadtxt(test_path,          dtype=float,
delimiter=',',skiprows=1,usecols=[3])
    result = model.inference(data=test_x) # 对数据 test_x 进行预测
```

图8

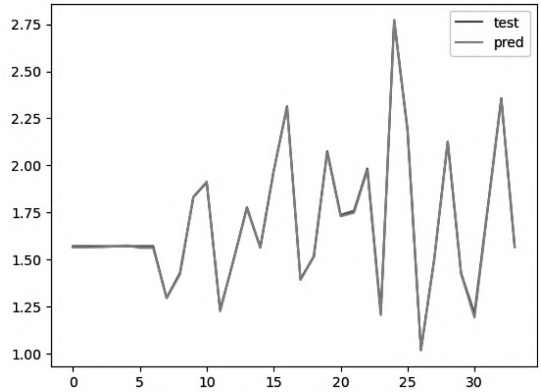


图9