

课程报告

课程 人工智能

姓名 吴红薇

学号 2015201928

日期 2017-12-14

1. **人工智能基本介绍**

人工智能（Artificial intelligence）亦称机器智能，是指由人制造出来的机器所表现出来的智能。通常人工智能指“智能主体的研究与设计”，智能主体是指一个可以观察周遭环境并作出行动以达到目标的系统。人工智能的研究可以分为几个技术问题。其分支领域主要集中在解决具体问题，其中之一是，如何使用各种不同的工具完成特定的应用程序。AI的核心问题包括推理、知识、规划、学习、交流、感知、移动和操作物体的能力等。如何让机器拥有人类的学习、推理、解决问题的能力，一直是人工智能领域研究的核心。

人工智能的覆盖范围实际上是随着时间推移不断变化的，某一类问题一旦达到较高的解决程度，人们就不认为它是一个需要“智能”才能解决的问题了，这一类问题就被剔除出人工智能的覆盖范围。人工智能的最大目标是实现general intelligence即一般化智能，像人一样解决各种各样的问题。但这个目标太大短期内无法一蹴而就，就把它分成许多子目标，包括推理、知识、规划、学习、交流、感知、移动和操作物体的能力等。

人工智能的方法可以分成几大类。

1）符号处理：20世纪50年代，数字计算机研制成功，研究者开始探索人类智能是否能简化成符号处理。60年代，符号方法在小型证明程序上模拟高级思考有很大的成就，基于控制论或神经网络的方法则置于次要。60-70年代的研究者确信符号方法最终可以成功创造强人工智能的机器，同时这也是他们的目标；

2）子符号方法：1980年代符号人工智能停滞不前，很多人认为符号系统永远不可能模仿人类所有的认知过程，特别是感知、机器人、机器学习和模式识别。很多研究者开始关注子符号方法解决特定的人工智能问题；

3）统计学方法：90年代，人工智能发展出复杂的数学工具来解决特定的分支问题。这些工具是真正的科学方法，即这些方法的结果是可测量和可验证的，同时也是近期人工智能成功的原因。这类方法的优点是非常优雅，严谨；但也有人批评这些技术太专注于特定的问题，而没有考虑长远的强人工智能目标。

从学科范畴来看，人工智能是一门边缘学科，属于自然科学和社会科学的交叉。涉及到物理学、哲学和认知科学、逻辑学、数学、心理学、计算机科学、控制论、决定论、不确定性原理、社会学、犯罪学。该学科的建立基于“人的行为、思维可以被精确刻画”这一一种假设，但目前还不能严格证明general intelligence能严格实现。同时随着人工智能的兴起，伴生了许多伦理性争议，如人造人是否享有与人类相等的权利等。

21世纪以来，AI技术，包括传统技术和新的技术都取得了很大的发展，人们把它归纳为几个原因：①计算机的发展，计算能力提升；②数据集规模增大，催生出智能的算法；③算法上的进步，理论上取得很大突破。

**二、学习情况**

这学期我们用的教材是《Hands-on Machine Learning with Scikit-Learn & TensorFlow》。这本书是17年新推出来的关于机器学习的书，分为两部分。第一部分主要讲经典的机器学习方法，主要用到python的开源工具库Scikit-Leart。这个库是目前在机器学习方面用到的最流行的库，常见数据集，包括数据预处理、经典机器学习算法、后期交叉验证、准确率计算，都完全地集成在里面，用起来很方便。第二部分主要介绍深度学习和神经网络，用到的是Google推出的TensorFlow库。下面就我个人通过学习的理解，简单总结人工智能、机器学习、神经网络、深度学习的关系。



图 1 关系图

如前面所述，人工智能（Artificial Intelligence）是一个大的概念，它是机器学习的父类。此处我将机器学习之外的人工智能归纳成了“逻辑/算法编程”，也就是通过编程将人类所知的知识和逻辑告诉机器，从而借助机器的高速计算和海量存储等能力实现一些人类才能做的“弱智能”工作。另一类就是基于数据的自我学习，即机器学习，把大量的数据告诉机器由机器自己去分析这些数据从而总结得出某种规律/逻辑，然后利用这种逻辑来处理新的数据。

机器学习是人工智能的一个分支，是实现人工智能的一个途径，即以机器学习为手段解决人工智能中的问题。机器学习是人工智能中最为火热和最有前途的方向，由于机器运算性能的提高，由机器取代人自主学习复杂逻辑成为了可能，这种“省力”的操作成了人工智能机器学习领域进步的动力。机器学习的一个定义如下：

*“Machine learning is the idea that there are generic algorithms that can tell you something interesting about a set of data without you having to write any custom code specific to the problem. Instead of writing code, you feed data to the generic algorithm and it builds its own logic based on the data.”*

这里面有几个重要的关键词，就是你不用写专门的业务逻辑代码而是通过输入大量的数据给机器，由机器通过一个通用的机制来建立它自己的业务逻辑，也就是机器“自我学习”了业务的逻辑，当然这种学习后的逻辑可以用来处理新的数据。这和人类的学习过程有些类似，但又不是完全一样。举个简单的例子，机器学习是不断的训练过程，其模型是在连续的优化调整中，随着训练数据额越多其模型越准确，但是人类的学习不仅仅是一个连续学习过程，还有一种跳跃式学习，也就是常说的“顿悟”，这点是机器学习目前所没有的。同时我们也可以看到，机器学习是通过大量的数据来训练得到模型的，因此其模型考虑的是相关性逻辑，而不是因果性逻辑，这和大数据分析的概念是一致的。总结来说，机器学习就是“利用已有数据，得出某种模型，并利用此模型预测未来的方法”。

机器学习主要包括有监督学习和无监督学习两种学习方法。所谓有监督学习，就是训练用历史数据是既有问题又有答案，而无监督学习就是训练用历史数据只有问题没有答案（正式的说法一般是把答案称之为标签label）。

在无监督学习中，主要是发现数据中未知的结构或者是趋势。虽然原数据不含任何的标签，但我们希望可以对数据进行整合（分组或者聚类），或是简化数据（降维、移除不必要的变量或者检测异常值）。因此无监督算法主要的分类包含：

-聚类算法（代表：K均值聚类，系统聚类）

-降维算法（代表：主成分分析PCA，线性判断分析LDA）

有监督学习，可以根据预测变量的类型再细分。如果预测变量是连续的，那这就属于回归问题。而如果预测变量是独立类别（定性或是定类的离散值），那这就属于分类问题了。因此有监督学习主要的分类包含：

-回归算法（线性回归，最小二乘回归，LOESS局部回归，神经网络，深度学习）

-分类算法（决策树，支持向量机，贝叶斯，K-近邻算法，逻辑回归，随机森林）

**三、课程实验情况**

人工智能不等于机器学习/深度学习。人工智能关注智能机器代理，关注这方面技术，为了不偏离主题，本门课程设计了课程项目AI CAR。我们小组的研究内容是利用人工智能小车进行深度学习算法的实践。我们的实验内容分为三个阶段，各阶段的功能实现目标如下：

（1）Randon Walker

小车能够按照随机方向行走，并且能够根据超声波传感器传回的信息智能规避障碍物。（2）Sensored Walker

利用蓝牙串口模块，通过AI CAR与手机的蓝牙连接，实现实时信息交互以及操作控制。小车的超声波传感器接收到的数据将会传输到手机上，而手机将可以控制小车的运行方向、速度，实现对小车的遥感控制。

（3）DeepAI Walker

利用小车装载的手机摄像头传输的视频图像，依托深度神经网络模型进行物体图像识别，同时通过小车与电脑的蓝牙连接，实现信息交互和操作控制，最终实现“小车寻物”的功能。

**3.1 第一阶段：Random Walker**

3.1.1 材料准备

我们按照老师的课件给出的推荐，进行小车制作材料的购买，主要材料如下：

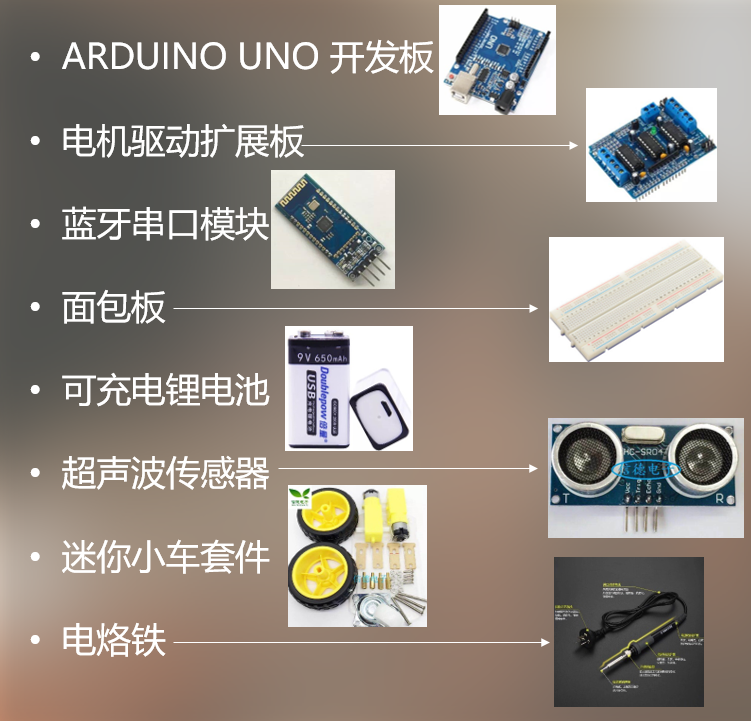


图 2 智能小车材料

3.1.2 小车拼装

开始拼装前，我们首先研究了Arduino板的各接口功能。其中我们用到的接口有：

1. digital pin 0 (RX):接收来自计算机的指令信息，与蓝牙串口模块上的TXD相连。
2. digital pin 1 (TX)：发送指令信息给计算机，与蓝牙串口模块上的RXD相连。
3. VCC：电源接口，相当于正极。
4. GND：电源接口，相当于负极。

其中1、2接口用作手机蓝牙与Arduino交换信息；3、4接口用作与电池相连，为Arduino板及电机供电。

根据以上接口功能，并借鉴老师的AI Car拼装方法，对小车进行了拼装。

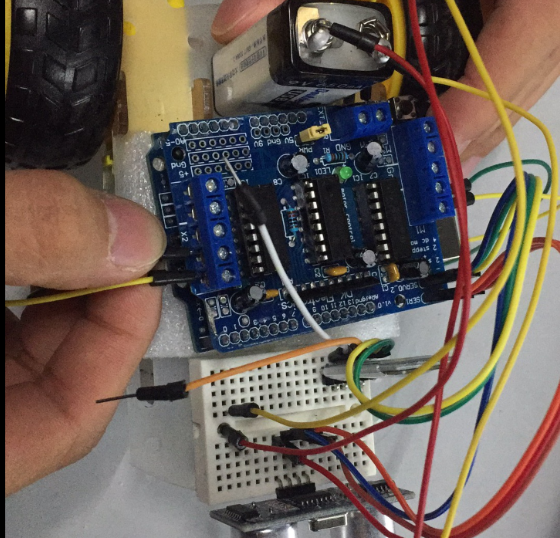
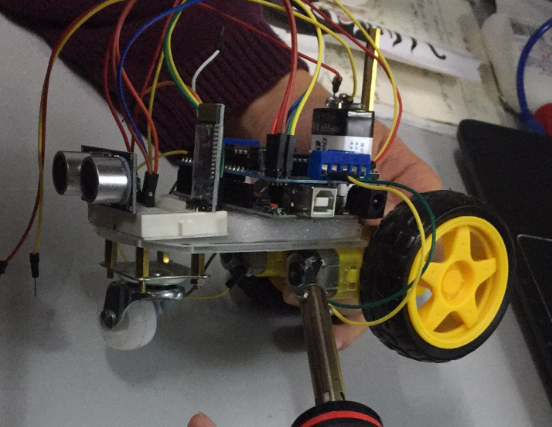
首先是拼装马达、车轮，再将相应的接口用线连接起来。如图3所示。

图 3 连线 图 4 接口焊接

然后是用电烙铁加固接口处连线。如图4所示。

3.1.3 代码导入

我们以老师上传到github上的aicar code为基础，为我们的小车设定了适当的参数。

在代码编译时我们发现了头文件”AFMotor.h”缺失的问题，通过上网查找相关资料我们发现我们需要自行下载“AFMotor.h“以及”AFMotor.cpp”，并将其放在与本文件相同的目录下，从而解决了该问题。

在代码中，控制左右马达转速的参数RLRatio的初始值为1.0，而我们进行预实验后发现右轮转速明显慢于左轮，因此我们尝试将RLRatio的值设为1.5，再进行实验时两轮的转速相当。可见1.5的转速比是适于本车的参数值。

在开头的若干次实验中，我们发现小车常常往后倒退。我们的第一个想法是更改代码，改变马达的转向。但后来梁晓周同学指出可以通过调换接线顺序来达到相同的目的。在更换插线顺序后小车顺利向前行驶。

此外，我们认为小车的车速过慢，这是由于代码中控制最大车速的参数maxSpeed的初始值只有80，因此我们改为了160并进行实验，却发现车速过快，小车有撞毁风险。因此，我们将该值重设为100，发现效果良好。

**3.2 第二阶段：Sensored Walker**

3.2.1 解决现存问题

上一阶段遗留了两个未解决问题：一是小车卡住后无法通过后退来解决困局；二是小车遇到障碍后只会右转，导致了“秦王绕柱”的局面，即当前方有柱状障碍物阻挡住小车去路时，它会一直右转，表现为绕着柱子转圈。分析小车卡住的原因，不难想到，由于小车只在前方安装有超声波测距仪，因此存在较大的视觉盲区，在障碍物较多的情况下，躲闪不及就进入了死角；此外，由于电池存在损耗问题，在遇到较大阻力时小车前进动力不足，自然就卡住了。为此，我们在每次LOOP循环中加入对距离变化的判断，在一个时间间隔前后两次测得的距离差小于某个阈值，则视为卡住，执行倒车操作，并为保证倒车后不会再次沿着原方向前进而再次卡住，在倒车后执行自动转弯函数调换方向。但新的问题又出现了，小车有时会一直倒车直至车尾卡在障碍物处，这种情形下，由于每次loop中测量的距离差都小于阈值，因此小车会无休止地执行倒车操作。为解决此问题，我们对“卡住”情形的处理增加了新的措施：小车有可能选择倒车，也有可能选择继续前进，这就有效地避免了这一问题。

第一个问题是小车卡住后无法通过后退来解决困局。

对于小车“卡住”的判断，我们首先想到采取以下办法：在每次 loop 循环都会执行的函数ultrasonicCar()中加入了对距离变化的判断，在时间间隔 time2judgebackward前后分别通过getdistance()函数测量距离 s1、 s2，若二者相等则视为小车卡住，执行倒车操作；否则正常执行 ultrasonicCar(s2)函数。

在多次试验后我们发现，小车卡住后并不会执行倒车操作。梁晓周同学提出，这有可能是因为我们对于距离差|s1-s2|的要求过于严格了，而超声波传感器本身的精度是有限的，因此可能会出现小车卡住了但多次测量的距离不同的情形，因此我们应设置阈值 deltadistance2backward ，若 |s1-s2|<=deltadistance2backward 则视为卡住。这一修改得到了良好的成效。

但新的问题又出现了，小车有时会一直倒车直至车尾卡在障碍物处，这种情形下，由于每次 loop中测量的|s1-s2|都小于阈值，因此小车会无休止地执行倒车操作。为解决此问题，我们对“卡住”情形的处理增加了新的措施：小车有可能倒车，也有可能前进，这就有效地避免了这一问题。

但实验结果却差强人意，小车有时在停住后甚至停止了工作，马达也不动了。我在观察代码后发现了问题：我们将 time2judgebackward 声明为了 int 变量，并将其作为 delay()函数的参数，导致其无限长地执行 delay()函数，故而小车停止运转。在将其类型改为 float 后问题得到了解决。

虽然我们保证了小车不会“卡死”，但是与此同时，我们发现小车经常撞到障碍物，这是由于测量距离差的用时 time2judgebackward 过长，而这段时间里是不执行障碍物判断并避开的操作的，因此会撞上障碍物。为此，我们将 time2judgebackward缩减到 0.3s。

然而，多次试验后我们又发现小车经常做“不必要的”倒车操作，当它并没有卡住甚至前方都没有障碍物时，它有时也会倒车。王辉同学指出这可能是因为 time2judgebackward 的缩短导致了所测量的距离差|s1-s2|也会相应缩短，当小车行驶速度较为缓慢时，距离差就有可能小于阈值而判断为卡住。我们首先想到的是将阈值deltadistance2backward 设为小车速度 maxspeed 的线性函数，但收效甚微。我们最后将其设为 1cm，发现小车在不同的速度 maxspeed 下都发挥良好。

此外，我们在倒车操作后又执行了函数randomright()调换方向，保证倒车后不会再次沿着原方向前进而再次卡住。我们还将障碍物判断的阈值从 40cm 降到了 20cm，得到了更好的表现。第二个问题是小车遇到障碍后只会右转，导致了“秦王绕柱”的局面（即小车绕着一根杆子一直绕圈）。

我们将 randomright()改为了 randomleftorright()，当判断距离 s<20cm 时， 我们执行右转或左转操作，这就有效地避免了同方向环绕的问题。在对代码进行了若干次修改后，我们发现小车前进时总是会向右倾斜，我们认为当前控制左右马达相对转速的参数 RLRatio 已不再适合小车的行驶。基于对该参数名字字面的认识，我们想当然地认为 RLRatio 值越高，则右轮相对转速越快，因此我们尝试着将 RLRatio 置为 2.0，结果发现右倾现象不见好转。当我们为这顽固的右倾分子烦恼时，我发现 forward()函数中 run()函数的参数为 BACKWARD，同时 backward()函数中的 run()的参数为 FORWARD，虽然我们对于为什么这样传参不甚理解，但这也引发了我们的猜想：RLRatio值增大时，右轮BACKWARD的程度加剧，因而更加慢于左轮。因此我们又尝试着置该值为 1.7，收效良好。

3.2.2 蓝牙功能加载

老师所给的代码中已有下面几项控制选项：前进、倒车、左打转、右打转、向左倾斜、向右倾斜、停车、获得距离值、打开自动行驶、关闭自动行驶。

我们首先是用iPhone安装了蓝牙助手但探测到了许多蓝牙设备，我们并不清楚小车蓝牙的名称，在多次连接尝试后都没能找到正确的设备。在参阅使用文献后我们发现该蓝牙只支持与安卓手机连接，因此我们用安卓手机安装了“蓝牙<->串口”程序，并成功找到了名称为“JDY-30”的蓝牙设备与其连接，顺利实现了蓝牙交互。

在多次试验后我们发现蓝牙交互操作的功能和我们设想的有所出入：在执行某个操作如倒车后，我们发现小车转变为了手动驾驶模式。而我们的设想是在操作后小车仍然保持自动驾驶。因此，我们将每个操作下的“control=0”语句删除，这样当小车执行完某个操作后驾驶模式不会发生切换。只有当我们选择打开或关闭自动驾驶时才会切换模式。

但是试验中我们发现小车对于蓝牙发送的操作无动于衷，自顾自地在自动驾驶。我们于是在每个操作下都加入了delay()语句，从而使得操作得以显现出来。

3.2.3 倒车警示功能加载

为了尽力模拟实际车辆的情形，我们考虑加入蜂鸣器作为喇叭以及提示器。我们声明了新的AF\_DCMotor变量Sounder来表示蜂鸣器，并为它分配了2号端口。我们设计了call()和rest()函数分别控制其发出响声和停止发声。

现实生活中，货车倒车时会发出“倒车请注意”的声响以提醒车后方的车辆或行人，于是我们在backward()函数中加入了call()语句，使得小车在倒车时蜂鸣器会长鸣。

但实验中我们发现蜂鸣器一旦发出声响就不会停止，无论小车倒车与否。这是因为虽然backward()函数执行结束了，但是由于call()函数设定了蜂鸣器的值，因此蜂鸣器不会随着函数的结束而停止发声。因此我们在所有其他的操作下加入了rest()语句，以使得小车不会胡乱地报出“倒车请注意”给车尾人群造成不必要的恐慌。

此外，我们还为蓝牙交互设置了按喇叭以及关闭喇叭的操作，实现了对小车喇叭的人为控制。

3.2.4 车灯功能加载

说到模拟实际车辆，车灯是必不可少的：转弯打灯、倒车打灯、危险报警闪光灯……因此，出于端口数量限制，为了加上两个LED灯，我们不得不将蜂鸣器去除，而“倒车请注意”则用双闪灯来代替。其实这样的效果更好，因为马达的噪音在一定程度上掩盖住了蜂鸣器的警示声，而光的形式则不受干扰。

与call()、rest()类似地，我们为两个小灯分别设计了函数Ltwinkle()、Lstop()和Rtwinkle()、Rstop()。left()函数中加入了Ltwinkle()函数，right()函数中加入了Rtwinkle()函数，而backward()函数中则加入了Ltwinkle()和Rtwinkle()函数。此外，为了避免类似于蜂鸣器长鸣不止的情形发生，我们在相应操作中加入了Lstop()或Rstop()函数。

在实验一开始时两个小灯都在发光后瞬间熄灭了，并且在后续的操作中都不再亮起。在直接通电检测后我们发现两个小灯烧了。王辉同学反应过来由于我们将小灯和蜂鸣器同等对待的缘故，小灯被加上的电压等同于马达所加上的电压，因此电压过高就烧了。我们特意为小灯设置了电压水平lightRate以保证其正常工作。在新购置的小灯到货后我们顺利完成了这些功能。

此外，我们还为蓝牙交互设置了长亮小灯以及关闭小灯的操作，同时为了防止对小灯的人工操作和小车行驶时小灯的闪灭之间的冲突，我们设置了标志值twinkle\_flag，当它被置为1时小灯长亮且不受小车行驶状态的影响。

我们还设计了危险报警闪光灯函数Blink()：以同样的时间间隔blink\_interval对两个小灯执行亮灭操作，并将函数放到loop中，保证闪光的持续性。

我们还为蓝牙交互设置了打开和关闭危险报警闪光灯、增加和降低闪光频率的操作。

3.2.5 蓝牙操作手册

0：关闭自动驾驶 < : 左倾 l : 左转向

1：打开自动驾驶 >：右倾 r：右转向

! : 双灯长亮 s：停车 q：双闪

? : 关闭长亮 f：前进 p：关闭双闪

+: 加快闪烁 b：后退 d：测距

- : 减慢闪烁

**3.3 第三阶段：DeepAI Walker**

3.3.1 图片物体识别

我们利用Tensorflow提供的Object Detection API，选用其中的ssd\_mobilenet\_v1\_coco模型训练出可以识别人、车、椅子等90种物体的深度神经网络模型。对于图片对象，该模型的物体识别函数能够识别图片中的物体，并重新渲染图片，显示出物体的名称以及预测可能性指数。

图 5 图片物体识别结果

3.3.2 视频物体识别

接下来要将模型应用到视频的物体识别中，我们利用MoviePy提供的VideoFileClip接口读入视频片段，并以物体识别函数为参数调用fl\_image()函数，这样就可以用每一个重新渲染后的帧替换原帧，得到一个动态识别物体的视频片段。



图 6 视频物体识别结果

3.3.3 实时视频物体识别

为了实现实时视频的物体识别，我们先在手机端下载了可以通过wifi传输实时图像的软件“IP Webcam”，并得到视频流的URL地址。然后我们利用OpenCV提供的VideoCapture接口读入该URL的视频流内容，再对其进行视频物体识别。

实时视频物体识别最大的问题是延迟太严重，由于wifi传输速度以及深度神经网络模型识别速度的限制，往往会产生长达十余秒的渲染图像延迟。

3.3.4 “小车寻物“功能加载

我们利用PyBluez提供的BluetoothSocket接口进行电脑与小车蓝牙之间的连接，从而实现电脑对小车的遥感控制。电脑端输入目标物体名称，如chair，那么装载着手机摄像头的小车就会开始按照随机方向行走，电脑端根据进行实时视频物体识别，若识别到的物体名称与目标物体名称匹配，则通过蓝牙发送命令让小车停止运动并长鸣喇叭，表示已找到目标物体。

图 7 实时视频物体识别结果

**3.4 课程实验总结**

本学期的实验任务循序渐进，既增强了我们的动手能力，又使我们一步步将课堂上所学的工具真正吸收为自己能掌握的知识。同时要感谢两位优秀的队友，在一次次的“头脑风暴”中提出了许多新奇的想法，令我受益匪浅。

**四、展望**

人工智能近年来发展迅速，这是有目共睹的。从可以自行充电的拟人机器人到无人驾驶飞行器集群的出现，也将有更多的更高级的人工智能产品，并使之在越来越多的领域在某种程度上超越人类智能。人工智能广泛应用与发展，说明其发展空间巨大。人工智能将在农业、医学、心理学、数学、天文学等多种学科做出更大的贡献，更好地服务人们

人工智能已经进入到了一个新的加速发展的时期，就它的意义来说，远远超过蒸汽机和电的出现而产生的工业革命。任何学科的成长不可能是一帆风顺的，人工智能在其发展过程中总会遇到很多困难和挫折，但是我们必将克服一切困难，找到“和谐”的方法去协调优劣势，让人工智能这一技术尽可能地造福社会。在我国，人工智能学科已经迎来了它的大好时期，人工智能将为社会发展、经济建设和人们的生活做出巨大的贡献。