

AI 小车实验报告及课程总结

陈洁婷 2016202134

一、 AI 小车实验报告

1. 实验情况概述

本次实验，我们的目的是以小车为载体，实践课堂上所学的人工智能知识。我们小组在购买了材料后，共同组装、调试小车，保证了硬件的支持。随后，我们共同搭建小车的红外、超声波、蓝牙模块，分别实现了手机和电脑对小车的自如控制以及小车的自动避障，为后续的人工智能奠定了基础。在最后的阶段，我们综合运用课堂所学及自有经验，各尽其能，从图像和声音两个角度实现了小车的智能。

从图像的角度，我们利用深度神经网络，利用公开数据集，结合自制数据集训练模型，实现了物体识别，小车能实时地对所见物体做出反应。从声音的角度，我们下载了 google 提供的 ESC 环境场景数据集和自制的数据集，利用了支持向量机、高斯混合模型、循环神经网络三种方法，借助贝叶斯信息准则、EM 算法、K 折交叉验证、网格搜索法的超参数调优，实现了小车的语音控制、说话人识别及音频场景识别功能。

2、个人主要贡献

在本次实验中，我表现积极，全程参与。除与大家组装小车、沟通合作外，主要负责手机蓝牙通信、数据集采集、基于支持向量机的音频场景识别实现及参数调优工作。同时，我也参与了说话人识别的 GMM 模型设计、语音处理工具配置以及 ppt 制作工作。

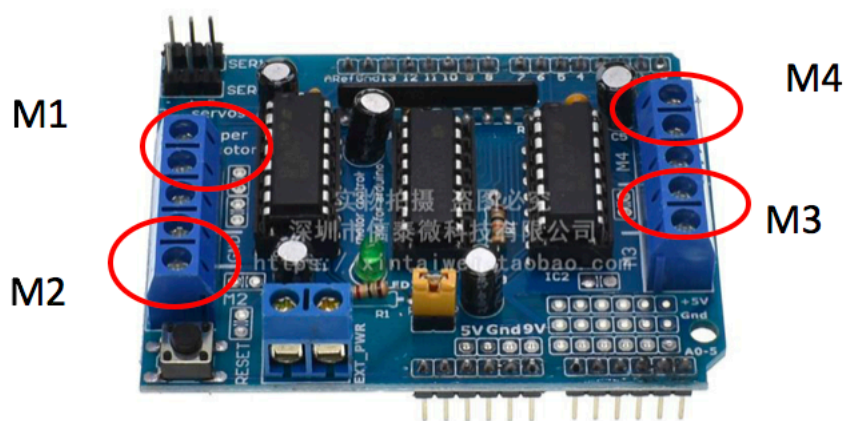
3、实验过程

3.1 硬件组装

在硬件组装过程中，我们所用的材料有

- (1) UNO R3 单片机
- (2) 电机驱动板
- (3) 蓝牙模块
- (4) 超声波及红外避障模块

- (5) 3.7V 航模电池 2 块
- (6) 电机
- (7) 轮子、面包板、杜邦线、电烙铁等



组装的方法在此不再赘述, 要点是将电机驱动板和 UNO R3 单片机正确连接, 以及上图驱动板的 4 个端口需与电机正确对应, 才能顺利控制小车行动。此外, 各模块的连接都应按照说明书进行, 否则会出现电路问题, 造成异常。

组装过程中我们可以说是历经坎坷, 花了大量的时间, 也多次更新了硬件。下面分享我们遇到的问题及解决方案。

(1) 小车电机、蓝牙异常

该问题可能出现在电池上, 对于我们的小车, 无法用充电宝替代电池, 其电压过低, 会导致行动的无力以及其他硬件故障。因此, 我们使用了 3.7V 的航模

电池替代，两节串接后使用。

该问题也可能出现在蓝牙上，蓝牙模块很容易装反，装反后会被电路烧热，可能导致部件损坏，蓝牙失效。

(2) 小车两轮速度不均衡

这问题是硬件本身的误差所致，我们可以通过 arduino 程序中设置一个速度比率，人工测试调整，实现两轮平衡。

(3) 小车控制与实际行动不匹配

这个问题困扰了我们很久，其实是因为我们不了解驱动板与 UNO 板之间引脚的对应关系，网上也几乎没有这方面的教程。因此我们重新购置了另一种型号的驱动板，其配备有清晰的引脚说明，问题解决。这里要特别注意的是，我们烧入代码的控制方式一定要与驱动板匹配，否则也会出现控制失误。

(4) 蓝牙模块连不上

这里的问题主要是由于蓝牙被不正确的操作烧坏了，尽管能够连接，但其信号传递部分已悄悄损坏。此外，蓝牙连接电脑时，我们需要弄清蓝牙的虚拟端口，配置好电脑环境，才能实现蓝牙控制。

(5) 程序无法烧入小车

特别注意，烧入程序时，蓝牙不能插在小车上，且小车 UNO 板的开关应该打开，否则会出现问题。

3.2 通信搭建

在通信搭建的模块，我主要负责小车的蓝牙与手机互连。这一块并不难实现，但需要在手机上安装一款“蓝牙串口助手”的 app。该 app 可以实现给小车发送蓝牙，并查看小车蓝牙与手机蓝牙的交互信息。有了这款工具，我们就可以实现对小车的自如控制，十分方便。

3.3 数据集采集

本次实验，我负责采集了两个数据集，一是针对说话人识别的自制数据集，我们以 16000hz 的采样率采集了 3 位同学的 10 个音频，每个音频约 5s，内容为呼唤小车名字“畅畅”。

第二个数据集用于音频场景识别，来自 <https://doi.org/10.7910/DVN/YDEPUT> 的 ESC50 数据集。内含 50 种不同的场景声音，如犬吠、警笛、玻璃破碎、水滴、门铃、下雨声。每种声音含 40 个 5s 的音频可供训练。在本次实验中，我们选择了 43:siren, 9:sheep, 11:rain, 21:crying baby, 44:car horn 这几种声音进行分类。

数据集内容见下图。

Animals	Natural soundscapes & water sounds	Human, non-speech sounds	Interior/domestic sounds	Exterior/urban noises
Dog	Rain	Crying baby	Door knock	Helicopter
Rooster	Sea waves	Sneezing	Mouse click	Chainsaw
Pig	Crackling fire	Clapping	Keyboard typing	Siren
Cow	Crickets	Breathing	Door, wood creaks	Car horn
Frog	Chirping birds	Coughing	Can opening	Engine
Cat	Water drops	Footsteps	Washing machine	Train
Hen	Wind	Laughing	Vacuum cleaner	Church bells
Insects (flying)	Pouring water	Brushing teeth	Clock alarm	Airplane
Sheep	Toilet flush	Snoring	Clock tick	Fireworks
Crow	Thunderstorm	Drinking, sipping	Glass breaking	Hand saw

3.4 SVM 音频场景识别实现

3.4.1 方法概述

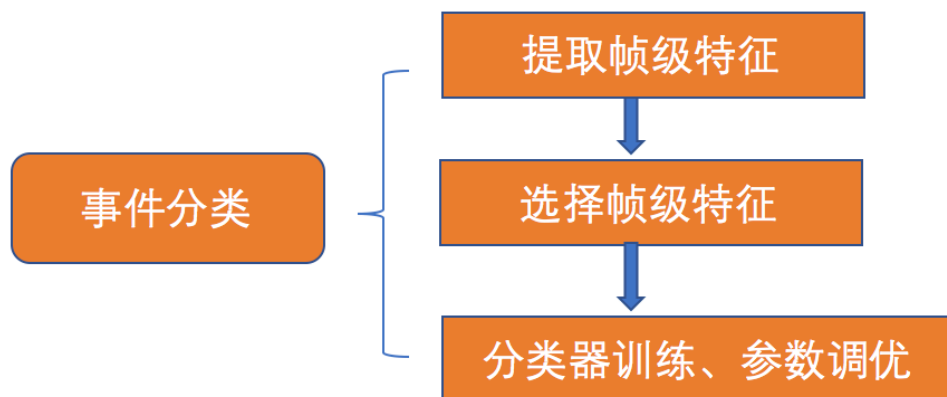
由于音频是一段连续的信号，我们将其以 0.25ms 的帧长、0.1ms 的帧移分帧处理，之后的训练和预测，也以帧为单位进行。该长度的窗所提取到的特征比较稳定。此外，我们选用海明窗对每一帧进行加窗处理，顺滑了分帧边界，避免了频谱泄露。

对于音频分类，特征十分关键。在特征提取的步骤中，我们首先利用 python 工具包 pyaudioanaly 及 opensmile 软件进行了特征提取的尝试。接着，通过声学含义分析和文献参考，我们对特征进行了筛选，去除了冗余，留下了最利于分类的特征。

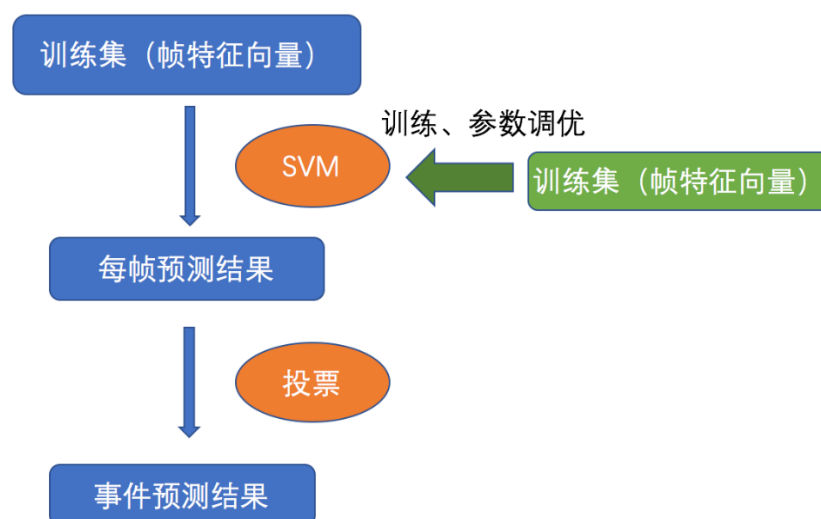
在场景分类的过程中，我们使用了 SVM 作为分类器，并通过网格分析法和三折交叉验证对其惩罚参数、核函数等进行调优，确定了我们能力范围内的最优结果。分类确定后，我们还采用了投票法，得到了对一段音频的事件预测。

3.4.2 流程图

训练流程图



测试流程图



3.4.3 特征选择

特征的选取是模式识别的关键问题，不同的特征分类的效果也是不同的。为更好地对音频进行分类，我们阅读了大量文献，初步筛选出了有利特征，并提取出来进行多次分类比较，最终确定了最有利于分类的特征。

阅读大量音频分类文献后，我们筛选出的 11 种特征名称及内涵如下

(1) 短时过零率

短时平均过零率，即每帧信号内，信号过零点的次数，体现的是频率特性。一般来说，清音的过零率更高，浊音更低。

(2) 短时能量

短时能量，即每帧信号的平方和，体现的是信号能量的强弱。

(3) 短时能量熵

能量熵，跟频谱的谱熵（Spectral Entropy）有点类似，不过它描述的是信号的时域分布情况，体现的是连续性。

(4) 谱熵

谱熵，根据熵的特性可以知道，分布越均匀，熵越大，能量熵反应了每一帧信号的均匀程度，如说话人频谱由于共振峰存在显得不均匀，而白噪声的频谱就更加均匀，借此进行 VAD 便是应用之一。

(5) 带宽

带宽反映的是音频频域分布范围的特征

(6) 子带能量比

子带能量比是描述音频信号频率分布的特征，它指的不同子带能量占整个频带能量的比例。

(7) 频谱质心

频谱质心反映的是声音信号的亮度特征，是指一个音频帧频谱能量分布的平均点。

(8) 13 维 MFCC 特征（12 维+energy）

MFCC 特征的原理在课堂上已经了解得十分清楚，在此不做赘述。MFCC 是目前人们在音频分类中，模拟人耳感知特性最好打参数，同时也一定程度地反映了发声原理，具有良好的识别性和抗噪性，在语音技术中十分常用。

(9) 13 维 chroma 特征（12 维+标准方差）

对于这个特征，我们的了解比较少，在文献中也没有看到，只知 12 个参数对应着西方的 12 个音阶。但由于 pyaudioanalysis 可以顺便提出该特征，我们就将其尝试使用，后发现该特征有利于分类。

(9) 频谱延展度

该特征又称为频谱二阶中心矩，它描述了信号在频谱中心周围的分布状况。这个特征在文献中没有出现，也是由于它是 pyaudioanalysis 中的经典特征之一，我们将其提出。

(10) 频谱通量

描述的是相邻帧频谱的变化情况，这是提取上述特征产生的附带产品，因此我们一同提出。

(11) 频谱滚降点

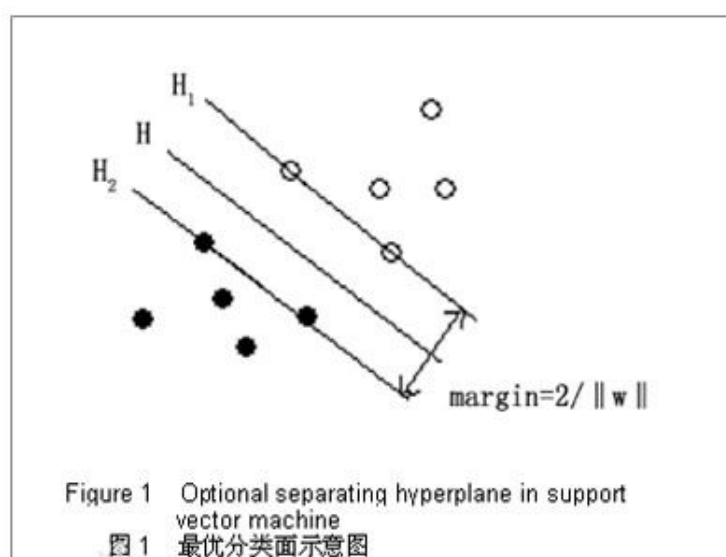
频谱滚降点表现的是某个频率点对应的幅度值对于整个频谱和的比率，是描述频谱波形的一种方式，被广泛用于语音和音乐的分类中。

除上述选出的特征外，在帧级还有许多特征如 LPCC、基频等，甚至在段级还有静音比率、高过零率比、低能量比、能量均值方差等特征。之所以排除掉某些帧级特征，有三方面的考虑：一是从声学含义来看，某些特征不契合本次分类的需求；二是许多文献表明，在这类实验中，某些特质不如另一种相似特征的效率高，如 LPCC 与 MFCC，MFCC 更胜一筹；三是由于时间限制，我们还没有时间顾及那些文献中已经证明对分类影响相对没那么大的段级特征。

为提取这些特征，我们以 25ms 的帧长，10ms 的帧移分帧，加汉明窗，通过 pyaudio+opensmile 进行提取，转存在 csv 文件中，使用时直接读取。

3.4.3 分类器选择理由

支持向量机是老师在课堂上详细讲过的一个模型，我们在音频分类的相关文献中也频繁看到，因此决定选用这个方法。支持向量机 SVM (Support Vector Machines) 是一种重要的机器学习方法，常用于模式识别中。它性能好，较稳定，特别适合复杂，但是为中小规模数据的分析。最原始的支持向量机是线性的二分类模型，它试图判定边界，将两种类别分开并保持一条尽可能宽敞的“街道”，位于“街道”边缘的样本点，就称为支持向量。



现在的支持向量机通过将二分类组合，非线性数据的投影，已经能够实现对非线性数据的多分类处理。

3.4.4 支持向量机的重要超参数

惩罚系数 C：C 越小，容错度越大

gamma：当模型为非线性时，需要将数据投影到高维变成线性可分，gamma 就与投影的过程有关，它的值越小，每个样本的影响范围越大，“街道”边界将更平滑。

核函数 kernel：常见的核有线性核函数 linear、高斯径向函数 rbf、多项式函数 poly。一般来说我们选择线性核函数，但当数据本身不服从线性规律时，不能强行使用线性核函数，需要使用 rbf 或 poly 进行实验。

3.4.5 分类器实现

我们采用 sklearn 库的 svm 模块实现了该算法，代码如下

```
def trainSVM(type,trD):    #SVM分类器
    print("Is training now.....")
    T,label=trainData(type,trD)
    clf=svm.SVC(C=1,gamma=0.1,kernel='rbf')    #参数已调优
    clf.fit(T,label)
    save_path="modelSVM.m"
    joblib.dump(clf,save_path)
```

```
vote=np.zeros(51)    #每帧投一次票，统计用于预测
model_path = "modelSVM.m"
clf=joblib.load(model_path)
for frameMFCC in test:
    P = clf.predict([frameMFCC])[0]
    vote[P]+=1
result=np.argmax(vote)
print('the number is:', t)
print('the predict number is:', result)
#计算命中率
if result == t:
    hit = hit + 1
hitrate = hit / test_size
print('the hit rate is:', hitrate)
```


效果如下（测试音频共 10 条，来自于 5 种类型）

```
the number is: 9
the predict number is: 9
the number is: 9
the predict number is: 9
the number is: 11
the predict number is: 11
the number is: 11
the predict number is: 11
the number is: 21
the predict number is: 21
the number is: 21
the predict number is: 21
the number is: 43
the predict number is: 43
the number is: 43
the predict number is: 21
the number is: 44
the predict number is: 44
the number is: 44
the predict number is: 44
the hit rate is: 0.9
```

3.4.6 参数调优（网格搜索算法+3 折交叉验证）

在默认情况下，sklearn 中的 svm.svc 函数采用的是 kernel=rbf，C=1，gamma=1/样本点数。我们在 10 个测试事件上进行试验，每个事件采用了十个音频，准确率为 0.9，图示如上一部分所示。

接着我们利用网格搜索算法，遍历不同的参数组合，并采用 3 折交叉验证的方法获得每种参数组合的评分。

```
def findBestPara(type, trD): #分类器调优
    T, label=trainData(type, trD)
    tuned_parameters = [{ 'kernel': ['rbf'], 'gamma': [1,0.1,0.01], 'C': [0.1,1,10]},
                        { 'kernel': ['linear'], 'C': [0.1,1,10]},
                        { 'kernel': ['poly'], 'C': [0.1,1,10], 'gamma': [1,0.1,0.01]}]
    grid = GridSearchCV(svm.SVC(), tuned_parameters, cv=4)
    # 用训练集训练这个学习器 clf
    grid.fit(T, label)
```

线性核函数 linear

```
0.638 (+/-0.111) for {'C': 0.1, 'kernel': 'linear'}
0.652 (+/-0.085) for {'C': 1, 'kernel': 'linear'}
0.659 (+/-0.073) for {'C': 10, 'kernel': 'linear'}
```

高斯径向核函数 rbf

```
0.507 (+/-0.133) for {'C': 0.1, 'gamma': 1, 'kernel': 'rbf'}
0.670 (+/-0.114) for {'C': 0.1, 'gamma': 0.1, 'kernel': 'rbf'}
0.601 (+/-0.145) for {'C': 0.1, 'gamma': 0.01, 'kernel': 'rbf'}
0.599 (+/-0.158) for {'C': 1, 'gamma': 1, 'kernel': 'rbf'}
0.696 (+/-0.111) for {'C': 1, 'gamma': 0.1, 'kernel': 'rbf'}
0.661 (+/-0.109) for {'C': 1, 'gamma': 0.01, 'kernel': 'rbf'}
0.599 (+/-0.159) for {'C': 10, 'gamma': 1, 'kernel': 'rbf'}
0.706 (+/-0.087) for {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
0.696 (+/-0.090) for {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}
```

多项式核函数 poly

经实验，我们的 $10 \times 500 \times 5 = 25000$ 帧放入 poly 核中耗时巨大，以至于无法做出遍历比较结果。因此，我将训练帧数降到 7500 帧，每个类别仅取 3 个音频进行参数选择比较。这导致 rbf 核以及 linear 核的最优参数发生了一些变化，有失严谨，但 poly 核对于控制变量下的输入数据训练耗时实在过长，我们决定采用这样的方式感受一下三种核的差别。可以比较出，rbf 核此时表现依然较为稳定和优秀。

Grid scores on development set:

```
0.500 (+/-0.183) for {'C': 0.1, 'kernel': 'linear'}
0.516 (+/-0.214) for {'C': 1, 'kernel': 'linear'}
0.529 (+/-0.243) for {'C': 10, 'kernel': 'linear'}
0.372 (+/-0.095) for {'C': 0.1, 'gamma': 1, 'kernel': 'rbf'}
0.624 (+/-0.164) for {'C': 0.1, 'gamma': 0.1, 'kernel': 'rbf'}
0.530 (+/-0.121) for {'C': 0.1, 'gamma': 0.01, 'kernel': 'rbf'}
0.474 (+/-0.091) for {'C': 1, 'gamma': 1, 'kernel': 'rbf'}
0.635 (+/-0.201) for {'C': 1, 'gamma': 0.1, 'kernel': 'rbf'}
0.596 (+/-0.162) for {'C': 1, 'gamma': 0.01, 'kernel': 'rbf'}
0.479 (+/-0.117) for {'C': 10, 'gamma': 1, 'kernel': 'rbf'}
0.645 (+/-0.287) for {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
0.613 (+/-0.209) for {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}
0.621 (+/-0.301) for {'C': 0.1, 'gamma': 1, 'kernel': 'poly'}
0.602 (+/-0.273) for {'C': 0.1, 'gamma': 0.1, 'kernel': 'poly'}
0.516 (+/-0.179) for {'C': 0.1, 'gamma': 0.01, 'kernel': 'poly'}
0.621 (+/-0.301) for {'C': 1, 'gamma': 1, 'kernel': 'poly'}
0.621 (+/-0.301) for {'C': 1, 'gamma': 0.1, 'kernel': 'poly'}
0.540 (+/-0.220) for {'C': 1, 'gamma': 0.01, 'kernel': 'poly'}
0.621 (+/-0.301) for {'C': 10, 'gamma': 1, 'kernel': 'poly'}
0.621 (+/-0.301) for {'C': 10, 'gamma': 0.1, 'kernel': 'poly'}
0.554 (+/-0.250) for {'C': 10, 'gamma': 0.01, 'kernel': 'poly'}
```

The best parameters are {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'} with a score of 0.65

经过上述实验，综合考虑时间复杂度，我们决定继续选择 rbf 核函数，并且确定最优参数如下

参数：kernel=rbf C=10 gamma=0.1

命中率：0.8

```
number: 11, predict: 9
the number is: 9
the predict number is: 9
the number is: 9
the predict number is: 9
the number is: 11
the predict number is: 11
the number is: 11
the predict number is: 11
the number is: 21
the predict number is: 21
the number is: 21
the predict number is: 21
the number is: 43
the predict number is: 44
the number is: 43
the predict number is: 21
the number is: 44
the predict number is: 44
the number is: 44
the predict number is: 44
the hit rate is: 0.8
```

令人震惊的是，命中率较调优前竟没有提高反而下降，这说明我们的调优参数范围存在局限性，网格搜索法遍历，无法尝试完所有可能的参数。因此，我们最终还是选用了 sklearn 库的默认参数，但通过这个调参的过程，我还是学到了很多新的知识，也更深入地理解了 svm 内部的原理及各种参数的效果。

3， 实验心得

本次实验是我第一次接触硬件，组装小车的过程非常有吸引力，也很考验我们的动手能力。最终，当我们把课堂所学的算法，学过的数学知识，运用到程序中，程序能够在实物上运行并作出反应时，真的令人欣喜，仿佛有种母亲看着儿子长大的喜悦。

在赋予小车智慧的过程中，我接触到了各种各样的机器学习算法，实践了课堂上那些看似抽象的理论，感受到了数学与算法的神奇力量。并且，由于涉及到图像与音频的处理，我们对于特征处理、特征工程，以及最终分类器的参数调优，都有了比较深刻的认识。

最后，令我颇为感动的是我们的组员，我们的合作非常地愉快，每个人都认真负责，全程参与，目标一致。在人工智能实现的部分，更是各尽所能，用心地

设计我们的实验，并用自己所能，广泛学习，努力实现。大家都积极参与每一次小组活动，无一人划水，交流融洽，十分令人感动。同时，我们也感谢其他小组同学给我们的帮助与分享。最后，我们也特别感谢胡老师，用简洁清晰的语言，教会我们人工智能的基本知识，并在我们遇到困难时，耐心地给予支持和解答。我们最终的成果与收获，与大家都帮助是分不开的。

二、课程总结

本学期的课程，给了我很大的收获。尽管在此前也接触过一些机器学习算法，但毕竟没有过系统的认识，总觉得零散，不成体系。经过这一学期的学习，我们了解到了最早的 AI 技术、经典的机器学习方法、神经网络的机器学习方法以及一些具体的有趣项目实践。

此外，对于机器学习的处理过程，如数据处理、特征处理、算法原理、参数调优等，我也有了更深刻的认识。过去总觉得人工智能、机器学习以及相关的一些名词高深不可触摸，学习之后，先是觉之精妙，用上之后，便是十分有成就感，着实是一个令人快乐的学习过程。

下面是我在本学期每节课上以及在网络上查找资料所得的课堂笔记，在此分享。一是便于自己复习，二也是分享给后来的同学，希望有利于大家共同进步。