**人工智能课程报告**

***欧阳妍妍 2015201353***

**一、学习成果概述**

在本课程的学习中，我主要收获了一下几方面的知识：

1、能够利用Docker、Github、等开源平台实现信息的交互。能够利用Arduino

实现计算机和机器的交互。

2、系统梳理学习了线性回归（岭回归、lasso）、支持向量机、决策树、神经网络的知识，并用python实现相关的操作。

3、具体在神经网络的部分，掌握了卷积神经网络、循环神经网络的原理，能够利用tensorflow进行模型训练。

4、通过小组作业具体实践，了解硬件的组装、环境的配置、以及模型的训练。

5、了解了人工智能发展现状以及主要的应用。

**二、小组作业**

1、小组作业实现情况

在本学期的人工智能课程中，我们共计完成了三次实验。我们组三次实验的成果分别为：

1）完成小车硬件的搭建和随机游走的实现。

实现小车能够自动前进，不原地打转；遇到障碍物时，能够通过超声波规避障碍物。

2）完成小车蓝牙的交互，实现蓝牙控制小车。

能够使用电脑或者手机蓝牙遥控小车，使其完成前进、左右转弯、后退、停止、启动。

能够向电脑传输拍摄的图像以及视频。

3）完成视频的抓取和分析。

利用卷积神经网络训练实现数字的识别，小车能够模拟红绿灯计时实现特定行走动作。

小车识别跟踪红色物体。

2、小组作业参与情况

从小组作业的积极参与中，我掌握了小车硬件的搭建方法，实现了计算机蓝牙、Wifi和小车通讯的环境搭配，掌握了Arduino的编写方法和使用Tensorflow训练模型的方法。

在三次的小组作业中，我们平均一周进行一次讨论，我负责了第一次小组ppt的制作，第二次小组报告、ppt制作以及展示。还负责了第三次实验中小车完成特定动作的策划。

3、小组作业遇到的问题以及收获

1）在完成小组作业的过程中，我熟悉掌握了Arduino的编写方法。在训练模型的过程中，我学会了抓取图片特征以及卷积神经网络的构建。

2）在实现的过程中，由于硬件频繁出现问题，我们组多次调试，使得我掌握了硬件的调试方法、以及代码的调试方法。了解了蓝牙模块以及Arduino板的构造以及电路。

3）另外，由于非专业的原因，遇到了环境配置的问题，导致计算机无法和小车建立蓝牙连接以及图像的捕捉，因此利用终端研究了环境配置，更加熟悉python。

**三、课后代码实现**

1、我尝试使用本课程教授的lasso和回归决策树的方法，进行书价预测的模型建构。选取多种函数变换后的自变量以及交叉项，利用lasso进行降维，实现自变量的挑选。

1）准备工作

首先是开始前的准备工作，在创建回归模型的过程中使用以下几个库文件，他们分别为sklearn库，numpy库，pandas库。sklearn库用于计算线性回归模型中的关键参数，并对模型进行检验。numpy和pandas库用于数据导入，创建数据表和一些基础的计算工作。

2）读取并查看数据表

准备工作完成后，开始读取数据，使用了一组每日广告成本和点击量的数据。将这组数据读取到python中并取名为data。通过head函数查看数据表中前5行的内容。

3）设置模型的自变量和因变量

在data数据表中，我们将书价相关自变量，页数、字数、纸质、出版年份设置为自变量X，将书价设置为因变量Y。希望通过回归模型发现广告成本对于广告点击量的影响，以及两者间的关系。将X设置为cost，Y设置为price。并通过shape函数查看了两个变量的行数，共1232行。

4）将数据分割为训练集和测试集

原始数据中共记录到1232本书的数据。将这1232条数据分为两部分，一部分作为训练集创建回归模型，另一部分作为测试集对回归模型的结果进行检验。

5）代入线性回归模型，计算模型的关键参数

将训练集数据代入到线性回归模型中，以计算关键的参数值。

建立多元回归模型。并增加交叉项。

6）使用测试集评估模型

评估回归模型的方法就是使用测试集数据，将测试集中的自变量代入到回归模型中，将结果与测试集的因变量进行对比，评估模型的准确性。模型预测结果，MSE=243.4。

2、TensorFlow部分

在学习了TensorFlow计算模型——计算图，TensorFlow数据模型——张量，TensorFlow运行模型——会话等基本概念之后，我用TensorFlow实现神经网络，主要分为以下3个步骤。

1）定义神经网络的结构和前向传播的输出结果

2）定义损失函数以及选择方向传播优化算法

3）生成会话并在训练数据上反复运行反向传播优化算法

3、参考代码实现了卷积神经网络手写数字训练

使用deepnn函数来构造升经网络，构建卷积层、池化层、全连接层、训练神经网络。交叉熵作为误差函数，使用Adam算法优化器

**卷积层：**

**全连接层：**

**池化层：**

**训练神经网络：**

**保存和恢复参数：**

在此过程中学习了如下几点：

1）python导入图片并进行数字化和大小的处理。将图片调为四维。

2）构建卷积层时，申明了两个变量，一个是保存权重，一个保存偏执，每一个变量的添加也是一个操作。然后使用tf.nn.relu创建另一个操作。

3）了解并尝试了多种优化器，如：

class tf.train.GradientDescentOptimizer 梯度下降算法

class tf.train.AdadeltaOptimizer 使用adadelta算法

class tf.train.AdagradOptimizer 使用adagradOptimizer算法

class tf.train.MomentumOptimizer 使用Momentum算法

4)学习了训练结果参数的保存和调用。

5）训练数据和测试数据的选择。

**四、学习内容梳理**

1、对各方法的整理理解

整体上看，机器学习就是模仿人识别事物的过程，即：学习、提取特征、识别、分类。目前我们课上学习的都是回归与分类的机器学习方法。

1）线性回归：拟合自变量和因变量线性关系的统计分析方法，常用最小二乘法来求解参数

2）SVM：把数据映射到多维空间中以点的形式存在，然后找到能够分类的最优超平面，最后根据这个平面来分类。SVM能对训练集之外的数据做很好的预测，但其对参数调节和核函数的参数过于敏感。SVM是二分类的最好的方法，但也仅限于二分类。

3）决策树：直观运用概率的图解方法，按特征来生成决策树，使目标期望达到最大，实际使用过程特征选择方式和决策树的修剪是关键。

4）神经网络：模仿人类大脑的神经突触结构，从而完成信息的传递处理

**步骤：**

a）初始化模型参数：在第一次运行的时候，初始化模型参数。通常使用随机数来代替，或者全部设置为0。

b）输入训练数据：对每个数据样本进行训练，通常会随机的打乱样本数据顺序。

c）在训练数据上验证模型：在训练数据上执行模型，根据当前的模型参数计算每次训练的输出。

d）计算损失：这个主要是用于表明我们采用的模型与实际模型的偏差有多大。对于不同的模型有不同的损失函数。在分类问题中，因为是将数据分成有限类，因此想要通过概率来判断属于哪一类，概率最大的就是结果。针对概率，一般是用交叉熵来作损失函数；由于回归问题是对具体数值的预测，即预测的是任意实数。所以对于回归问题，最常用的是均方差损失函数。

e）调整模型参数：这是在学习过程中发生的。根据损失函数，调整模型的参数，通常使用梯度下降算法来学习。梯度下降算法的参数优化方法在机器学习基础里已经总结。

2、机器学习思想

1）数据抽象

将数据集和具体问题抽象成数学语言，以恰当的数学符号表示。这样做自然是为了方便表述和求解问题，而且也更加直观。

例如图像识别的抽象。

2）最小化损失函数和泛化能力的矛盾

训练模型目标：经验风险最小化，即以训练样本误差最小化来衡量模型的好坏，从而无限增加模型的复杂度。

考虑的因素：模型的泛化能力（“过拟合”现象）

解决方案：使用类似均方误差一类的函数，使用正则化的方法。

3）设定性能度量指标

机器学习是产生模型的算法，一般来说模型都有误差。一般来说机器学习面对的主要问题都是过拟合。那么为了保证模型的泛化能力足够强，必须要有衡量模型泛化能力的评价标准，也就是性能度量的设定。

然后基于统计假设检验来做效果判定，挑选出更优的指标。

4）训练及优化

- 如果要评估训练集和验证集的划分效果，常用的有留出法、交叉验证法、自助法、模型调参等

- 如果模型计算时间太长，可以考虑剪枝

- 如果是过拟合，则可通过引入正则化项来抑制（补偿原理）