



---

人工智能

---

课程报告



学生

孙佳佳 2016202138

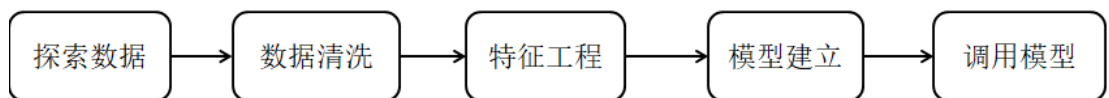
2018-12-25

## 1 学习目标

- (1) 理解并掌握机器学习的基础算法、神经网络与深度学习算法。并能熟练运用到具体的机器学习项目中。
- (2) 能熟练应用 Arduino、树莓派等一些开源硬件。
- (3) 了解机器学习领域最新算法。
- (4) 理解掌握 hands-on-ml 书上的示例代码；完成基于 arduino 板的智能小车搭建。

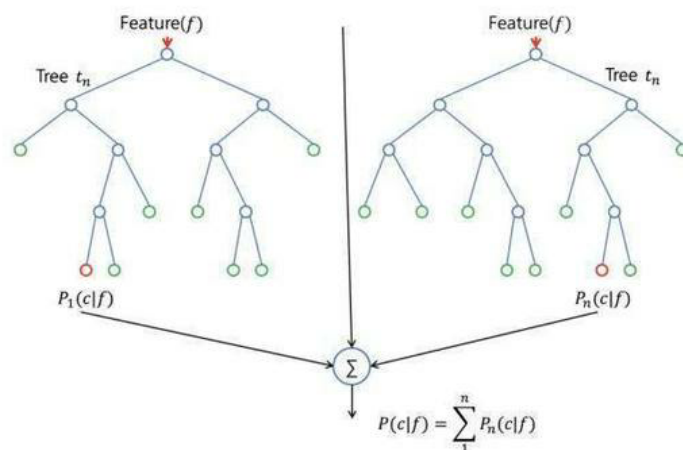
## 2 学习内容总结

- (1) 一个完整机器学习项目的流程：



在机器学习项目中，大部分的工作都在数据清洗和特征工程上，需要把反应现实的数据处理得更加适合模型。

- (2) 几个概念：  
训练集、测试集、交叉验证、准确率、召回率
- (3) 回归算法
  - 线性回归  
数学上使用最小二乘法的思想：拟合出一条直线使得所有误差的平方和最小。  
计算机中：梯度下降法和牛顿法是数值计算中的经典算法。
  - 逻辑回归  
逻辑回归于线性回归是一种非常相似的算法。但是从本质上看，线性回归处理的是数值问题，逻辑回归处理的是分类问题。
- (4) 支持向量机  
支持向量机（support vector machines）是一种二分类模型，它的目的是寻找一个超平面来对样本进行分割，分割的原则是间隔最大化，最终转化为一个凸二次规划问题来求解。由简至繁的模型包括：
  - 当训练样本线性可分时，通过硬间隔最大化，学习一个线性可分支持向量机；
  - 当训练样本近似线性可分时，通过软间隔最大化，学习一个线性支持向量机；
  - 当训练样本线性不可分时，通过核技巧和软间隔最大化，学习一个非线性支持向量机；通过跟高斯“核”的结合，SVM 可以表达出非常复杂的分类界线，从而达到很好的分类效果。“核”实际上是一种特殊的函数，最典型的特征就是可以将低维的空间映射到高维的空间。
- (5) 随机森林  
随机森林是指利用多棵决策树对样本进行训练并预测的一种算法。输出的类别是由个别决策树输出的类别的众数来决定的。



随机森林对于大部分的数据分类效果比较好，能处理高维的特征，不容易产生过拟合，且对数据集的适应能力强，既能处理离散数据，也能处理连续型数据

#### (6) 降维

降维算法是一种无监督学习算法，主要特征就是将数据从高维降低到低维层次。降维算法的主要作用是压缩数据与提升机器学习其他算法的效率。通过降维算法，可以将具有几千个特征的数据压缩至若干个特征。

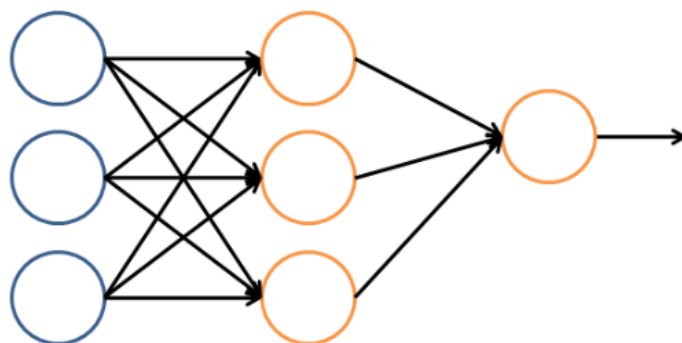
降维算法的主要代表是 PCA 算法（即主成分分析算法）

#### (7) 人工神经网络

神经网络的逻辑架构为：输入层、隐藏层、输出层。

输入层负责接收信号，隐藏层负责对数据的分解与处理，最后的结果被整合到输出层。

如图，每层的一个圆代表一个处理单元，可以认为模拟了一个神经元，若干个处理单元组成了一个层，若干层再组成了一个网络，就是“神经网络”。



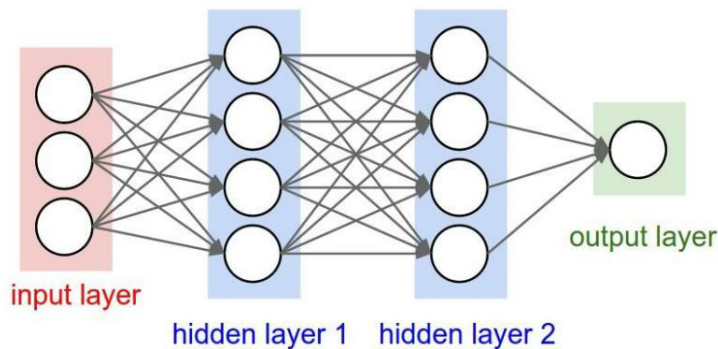
在神经网络中，每个处理单元事实上就是一个逻辑回归模型，逻辑回归模型接收上层的输入，把模型的预测结果作为输出传输到下一层。通过这样的过程，神经网络可以完成非常复杂的非线性分类

#### (8) 卷积神经网络

卷积神经网络的层级结构：

➤ 数据输入层/Input layer: 对原始图像数据进行预处理（去均值、归一化、PCA）

- 卷积计算层/CONV layer:  
局部关联。每个神经元只关注一个特性，神经元就是图像处理中滤波器 (filter)；  
窗口滑动：通过窗口滑动实现 filter 对局部数据的计算
- ReLU 激励层/ReLU layer: CNN 采用的激励函数一般为 ReLU
- 池化层/Pooling layer:  
池化层夹在连续的卷积层中间，用于压缩数据和参数的两年，减小过拟合。比如：如果输入时图像的话，那么池化层的最主要作用就是压缩图像。
- 全连接层/FC layer:  
两层之间的所有神经元都有权重连接，也就是跟传统的神经网络神经元的连接方式是一样的：



优点：

- 共享卷积核，对高维数据处理无压力
- 无需手动选取特征，训练好权重，即特征分类效果好

缺点：

- 需要大量样本，需要使用 GPU
- 物理含义不明确，我们并不知道每个卷积层到底提取到的是什么特征，而且神经网络本身就是一种难以解释的黑箱模型

卷积神经网络常用框架：caffe、TensorFlow

### 3 Aicar 大作业总结

(1) 总结三次实验成果

- 第一次展示
  - 1) 完成了小车的硬件连接,能把程序烧如 arduino 板中,控制小车的前进后退;
  - 2) 通过蓝牙给小车发送特定字符控制小车的前进后退等动作
  - 3) 使用了手机 app: OpenCV, 实现了小车跟踪颜色鲜艳的物体。
- 第二次展示
  - 1) 语音识别：电脑识别人发出的语音，并发送相应控制指令给小车，最终实现了语音控制小车的功能
  - 2) 人脸识别：当电脑识别出特定人脸时，给小车发送前进指令
- 第三次展示
  - 1) “自动驾驶功能”：搭建简易跑道，拍下训练集照片，并打上相应标签：前进、后退、左转、右转。通过人工神经网络训练模型。运行模型最终实现小车在跑道上的自动驾驶

2) 识别交通标志：小车能识别特定交通标志并作出相应动作

3) 在手机上实现集成 APP

## (2) 小组参与度

### ➤ 第一次实验

在此次实验中，我主要负责实现“跟踪颜色鲜艳物体”的功能。

实验过程为：

✧ 通过 serial.read 获得手机 APP：OpenCV 发送的物体位置

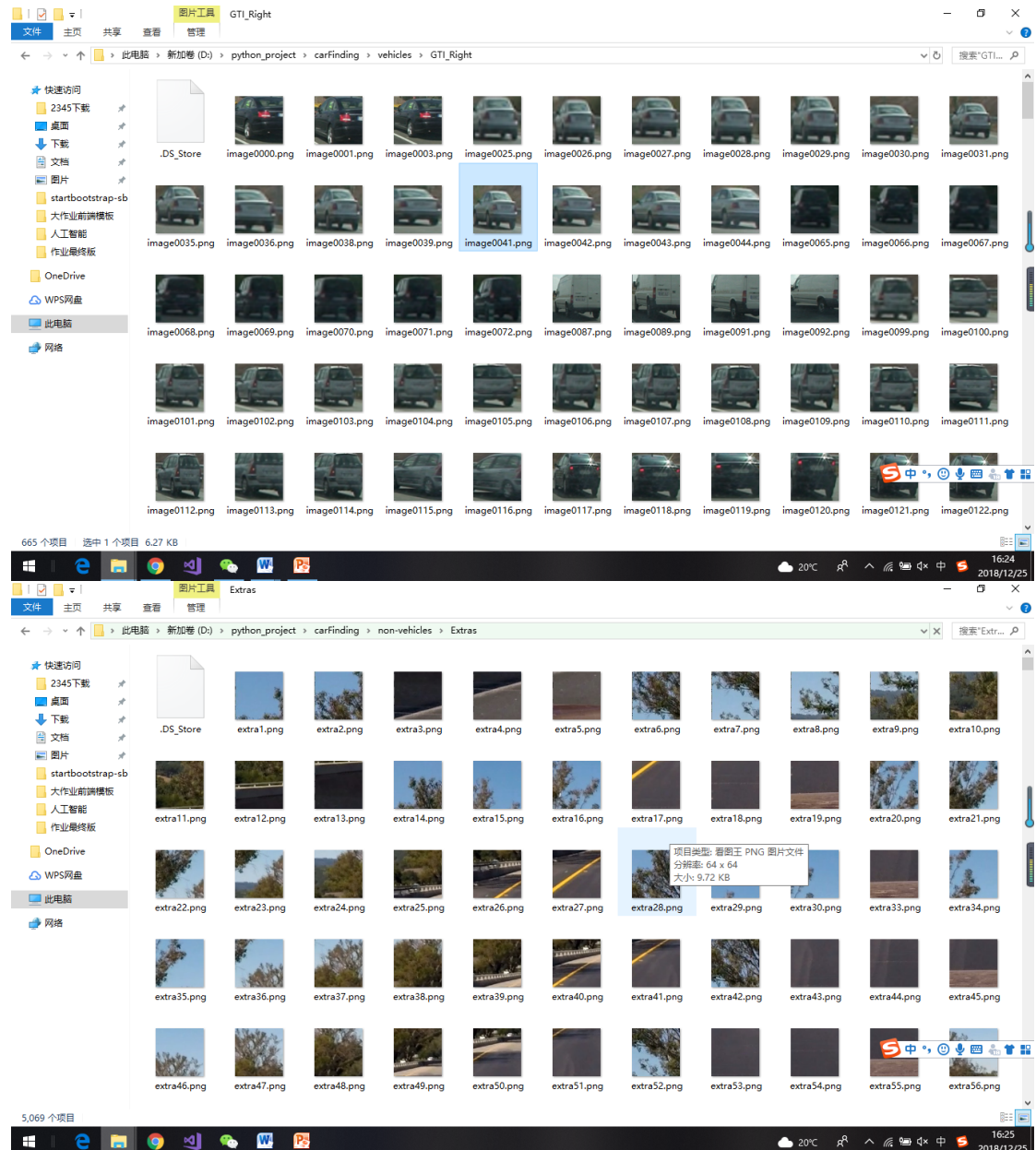
✧ 用 if...else...语句判断物体在手机屏幕中的相对位置，并控制小车的电机，让小车跟踪手机屏幕中的物体

### ➤ 第二次实验

在此次实验中，我主要负责实现自动驾驶功能。很可惜一开始的方向设得有点大没有能最终实现，只实现了：识别交通视频中车辆的功能

实验过程：

✧ 数据集：车辆照片、没有车辆的道路照片



- ✧ 编写特征提取函数
- ✧ 使用 SVM 和提取到的特征训练一个二分类器，区分汽车和非汽车
- ✧ 编写滑动窗口，并把每一个窗口用 SVM 判断是否有汽车
- ✧ 编写热力图函数，把多个判断为汽车的窗口合并为一个窗口
- ✧ opencv 读取视频和图片转换为视频
- ✧ 最终结果：



### ➤ 第三次实验

在此次实验中我主要负责“自动驾驶”模块，让小车能在跑道上自动驾驶

实验内容

- ✧ 数据采集：编写 collect\_data 函数：通过电脑键盘的前进后退向左向右键控制小车在跑道上行驶，每按下一个控制键，就自动通过手机 IP 摄像头保存一帧图片，并且把当前的按键动作一并保存。  
这样就得到数据包括 picture + label 并保存为 npz 文件
- ✧ 使用 ANN 训练模型，使用两层隐藏层，并把训练好的模型保存为 xml 文件  
得到准确率和召回率分别为 87.63% 86.90%

```
Run: train_moudle x
Image array shape: (278, 1036800)
Label array shape: (278, 4)
Loading data duration: 19.53s
Training ...
Training duration: 58422.65s
Train accuracy: 87.63%
Validation accuracy: 86.90%
Model saved to: 'saved_model/nn_model.xml'

Process finished with exit code 0
```

- ✧ 运用模型判断小车的前进方向实现自动驾驶



### (3) 收获

- 在后两次实验中，我意识到训练集一般性的重要性。在机器学习中训练集的使用要具有一般性和代表性。  
在第二次实验中，我使用的数据集是从网上下载的数据集，使用 SVM 模型的准确率和召回率都很高，但当我使用自己拍摄的视频进行测试时，出现了很多识别错误。  
在第三次实验中，当我们换一个场地搭建跑道时，小车不能正确地沿着跑道走。
- 在模型的选择上，要结合每个模型的特点和特定场景的要求作出选择，在不能确定时可以多做测试，根据交叉验证评估各个模型的好坏。

## 4 课后练习

Kaggle 竞赛网站上“预测房价”的竞赛：

### (1) 练习目的

巩固机器学习项目的实验流程

运用不同的机器学习算法，如 LinearRegression、Ridge、Lasso、Random Forrest、Gradient Boosting Tree、Support Vector Regression、Linear Support Vector Regression 等算法

### (2) 主要内容

- 数据清洗：去掉离群值、缺失值填充
- 特征工程：使用 LabelEncoder 对不连续的数字或文本进行编号；使用 PCA 进行降维：有些特征是高度相关的，并且它导致多重共线性。PCA 可以去相关这些特征。因此，将在 PCA 中使用与原始数据大致相同的维度
- 选择模型；参数调优

```
In [50]: models = [LinearRegression(), Ridge(), Lasso(alpha=0.01, max_iter=10000), RandomForestRegressor(), GradientBoostingRegressor(), SVR(), LinearSVR(),
                  ElasticNet(alpha=0.001, max_iter=10000), SGDRegressor(max_iter=1000, tol=1e-3), BayesianRidge(), KernelRidge(alpha=0.6, kernel='polynomial',
                  ExtraTreesRegressor(), XGBRegressor())

In [51]: names = ["LR", "Ridge", "Lasso", "RF", "GBR", "SVR", "LinSVR", "Ela", "SGD", "Bay", "Ker", "Extra", "Xgb"]
          for name, model in zip(names, models):
              score = rmse_cv(model, X_scaled, y_log)
              print("{} {:.6f}, {:.4f}".format(name, score.mean(), score.std()))

LR: 1451643144.490545, 678059351.3682
Ridge: 0.117881, 0.0092
Lasso: 0.123088, 0.0060
RF: 0.138287, 0.0057
GBR: 0.119341, 0.0046
SVR: 0.113260, 0.0053
LinSVR: 0.125086, 0.0104
Ela: 0.109403, 0.0059
SGD: 0.331926, 0.0235
Bay: 0.111316, 0.0066
Ker: 0.110461, 0.0057
Extra: 0.136049, 0.0073
Xgb: 0.120769, 0.0064
```

- 最终选择 SVM 建立模型进行预测

## 六、预测

```
In [65]: weight_avg.fit(X_scaled, y_log)

Out[65]: AverageWeight(mod=[SVR(C=13, cache_size=200, coef0=0.0, degree=3, epsilon=0.009, gamma=0.0004,
kernel='rbf', max_iter=1, shrinking=True, tol=0.001, verbose=False), KernelRidge(alpha=0.2, coef0=1.2, degree=3, gamma=None, kernel='polynomial',
kernel_params=None)],
weight=[0.5, 0.5])

In [66]: pred = np.exp(weight_avg.predict(test_X_scaled))

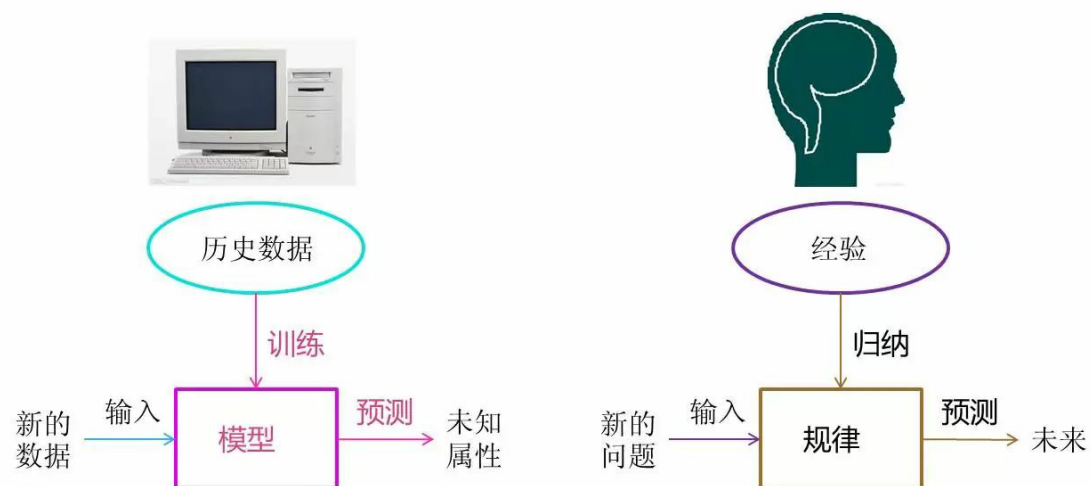
In [67]: result=pd.DataFrame({'Id':test.Id, 'SalePrice':pred})
          result.to_csv("submission_11.csv",index=False)
```

## 5 课程学习心得

传统上如果我们想让计算机工作就给它一串指令，让它按照这个指令一步步执行下去，最终得到结果。而机器学习不接受输入的指令而接受输入的数据，也就是说机器学习是一种让计算机利用数据而不是指令来进行各种工作的方法。

机器学习的一个主要目的就是把人类思考归纳经验的这一过程转化为家算计通过对数据的处理计算得出模型的过程。经过计算机得出的模型能够近似于人的方式解决很多灵活复杂的问题。

(机器学习与人类思考的类比图:)



人工智能是机器学习的父类，机器学习的出现让人工智能找到的一个重要方向。基于机器学习的图像识别和语音识别在某些领域已经到了跟人相媲美的程度。也许在不远的将来，一个人具有人类智能的计算机真的有可能实现。