编程和数学基础不佳如何入门人工智能？

一、人工智能的发展现状

1.1 概念

根据维基百科的解释，人工智能是被机器展示的智力，与人类和其他动物的自然智能相反，在计算机科学中AI研究被定义为“代理人软件程序”：任何能够感受周围环境并且能最大化它成功机会的设备。

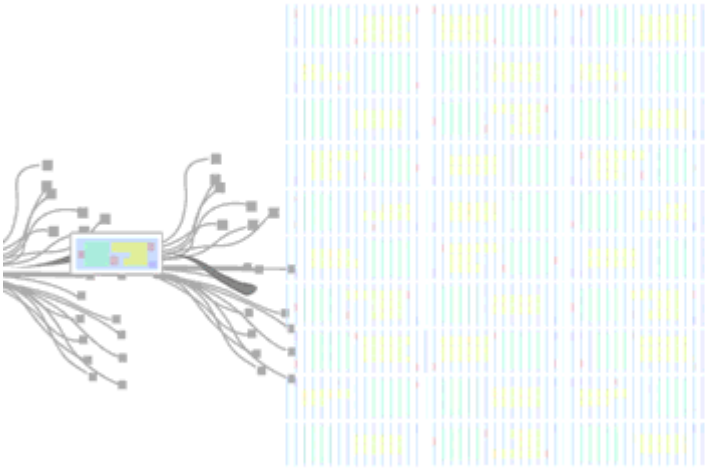
1.2 重大事件

- **2016**年**3**月，AlphaGo与当时世界排名第四、职业九段棋手李世石，进行围棋人机大战，以4:1总比分获胜。

- **2016** 年 **10** 月，美国白宫发布了《为未来人工智能做好准备》和《美国国家人工智能研究与发展策略规划》两份重磅报告，详细阐述了美国未来的人工智能发展规划以及人工智能给政府工作带来的挑战与机遇。

  VentureBeat 对这两份报告进行了总结，得出了 7 个浅显易懂的要点：1. 人工智能应当被用于造福人类、2. 政府应该拥抱人工智能、3. 需要对自动汽车和无人机进行管制、4. 要让所有孩子都跟上技术的发展、5. 使用人工智能补充而非取代人类工作者、6. 消除数据中的偏见或不要使用有偏见的数据、7. 考虑安全和全球影响。

- **2016**年双十一，鲁班首次服务双十一，制作了1.7亿章商品展示广告，提升商品点击率100%。如果全靠设计师人手来完成，假设每张图需要耗时20分钟，满打满算需要100个设计师连续做300年。2017年，鲁班的设计水平显著提升，目前已经学习百万级的设计师创意内容，拥有演变出上亿级的设计能力。此外，鲁班已经实现一天制作4000万张海报能力，没有一张会完全一样。



- **2017**年**5**月，AlphaGo Master战胜世界冠军柯洁。
- **2017**年**10**月**18**日，DeepMind团队公布了最强版本AlphaGo，代号AlphaGo Zero。
- **2017**年**10**月**25**日，在沙特举行的未来投资计划大会上，沙特阿拉伯授予美国汉森机器人公司生产的“女性”机器人索菲亚公民身份。

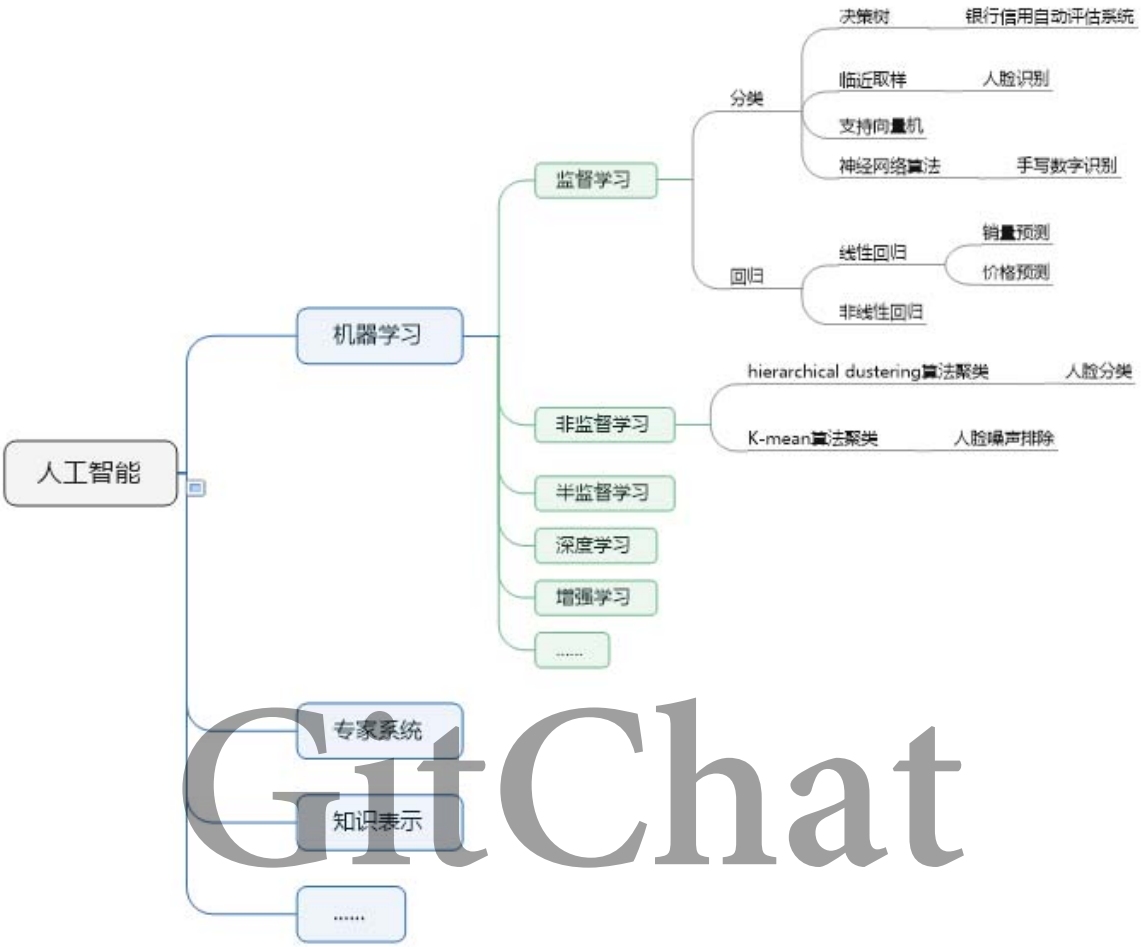
  作为世界上首个获得公民身份的机器人，索菲亚当天说，“她”希望用人工智能“帮助人类过上更好的生活”，同时对支持“AI威胁论”的马斯克说“人不犯我，我不犯人”!

  会后，马斯克在推特上说：“把电影《教父》输入了人工智能系统，还能有什么比这个更糟的？”教父是好莱坞经典电影，剧情充满了背叛和谋杀。

  索菲亚被授予公民身份后所产生的伦理问题也是人们不得不考虑的

- 近几年人工智能领域的大新闻太多，这里不一一列举

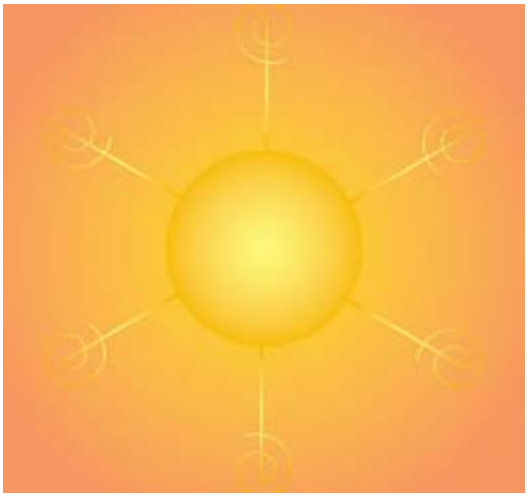
二、人工智能、深度学习、机器学习、增强学习之间的关系是怎样的



如图所示，人工智能是一个大类，包括专家系统、知识表示、机器学习等等，其中机器学习是目前最火也是发展最好的一个分支，机器学习中又包括监督学习、非监督学习、深度学习，增强学习等等。

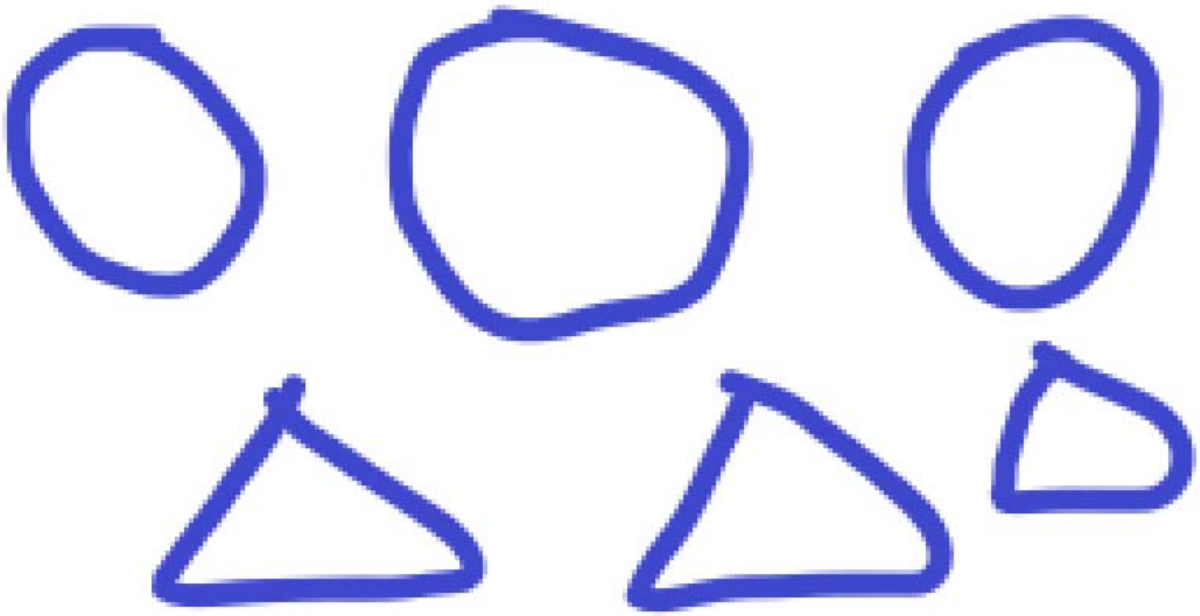
监督学习，就是人们常说的分类，通过已有的训练样本（即已知数据以及其对应的输出）去训练得到一个最优模型（这个模型属于某个函数的集合，最优则表示在某个评价准则下是最佳的），再利用这个模型将所有的输入映射为相应的输出，对输出进行简单的判断从而实现分类的目的，也就具有了对未知数据进行分类的能力。

举例来说，我们上幼儿园的时候经常做的一个活动叫看图识字，如上图所示，老师会给我们看很多图片，下面配了文字，时间长了之后，我们大脑中会形成抽象的概念，两个犄角，一条短尾巴，胖胖的（特征）...这样的动物是牛；圆的，黄的，发光的，挂在天上的...是太阳；人长这样。等再看到类似的东西时我们便能认出来，即使跟以前看到的不完全一样，但是符合在我们大脑中形成的概念，如下图所示。

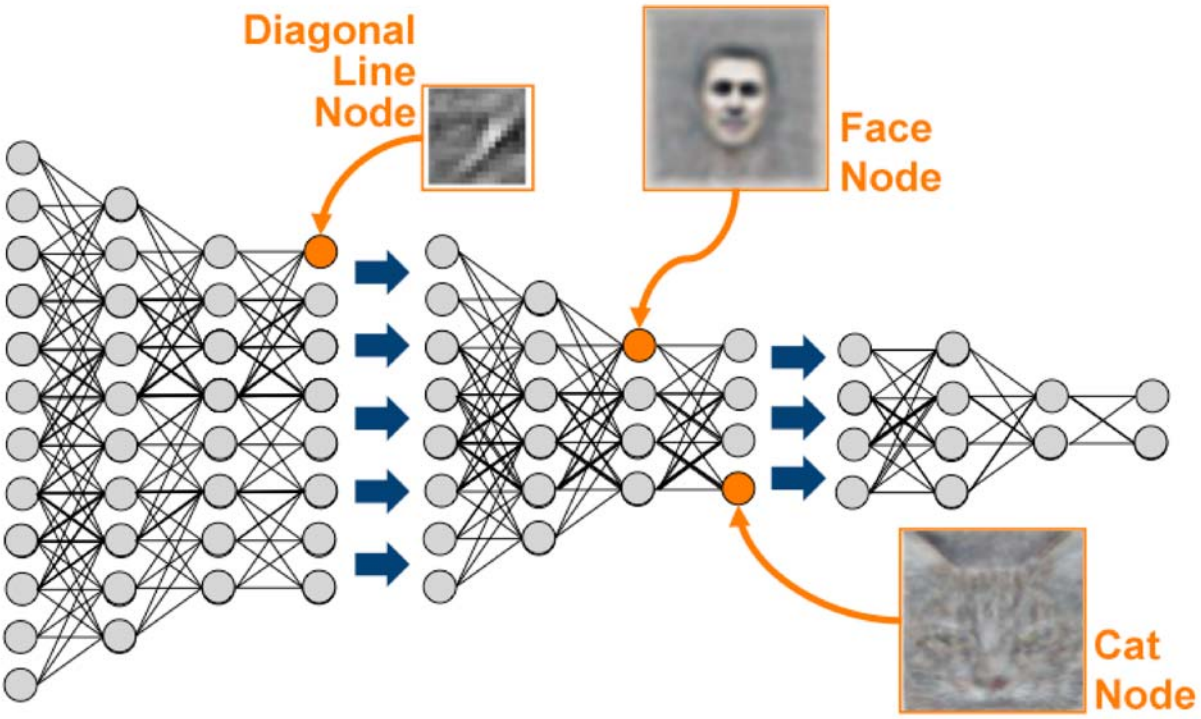


非监督学学习则是另一种研究的比较多的学习方法，它与监督学习的不同之处，在于我们事先没有任何训练样本，而需要直接对数据进行建模。

举个例子，如图所示，在没有任何提示（无训练集）的情况下，需要把下列六个图形分成两类，你会怎么分呢，当然是第一排一类，第二排一类，因为第一排形状更接近，第二排形状更接近。非监督学习就是在实现不知道数据集分类的情况下在数据中寻找特征。



深度学习是基于机器学习延伸出来的一个新的领域，由以人大脑结构为启发的神经网络算法为起源加之模型结构深度的增加发展，并伴随大数据和计算能力的提高而产生的一系列新的算法。



深度学习概念由著名科学家Geoffrey Hinton等人在2006年和2007年在《Sciences》等上发表的文章被提出和兴起。

深度学习，作为机器学习中延伸出来的一个领域，被应用在图像处理与计算机视觉，自然语言处理以及语音识别等领域。自2006年至今，学术界和工业界合作在深度学习方面的研究与应用在以上领域取得了突破性的进展。以ImageNet为数据库的经典图像中的物体识别竞赛为例，击败了所有传统算法，取得了前所未有的精确度。

增强学习也是机器学习一个重要的分支，是通过观察来学习做成如何的动作。每个动作都会对环境有所影响，学习对象根据观察到的周围环境的反馈来做出判断。

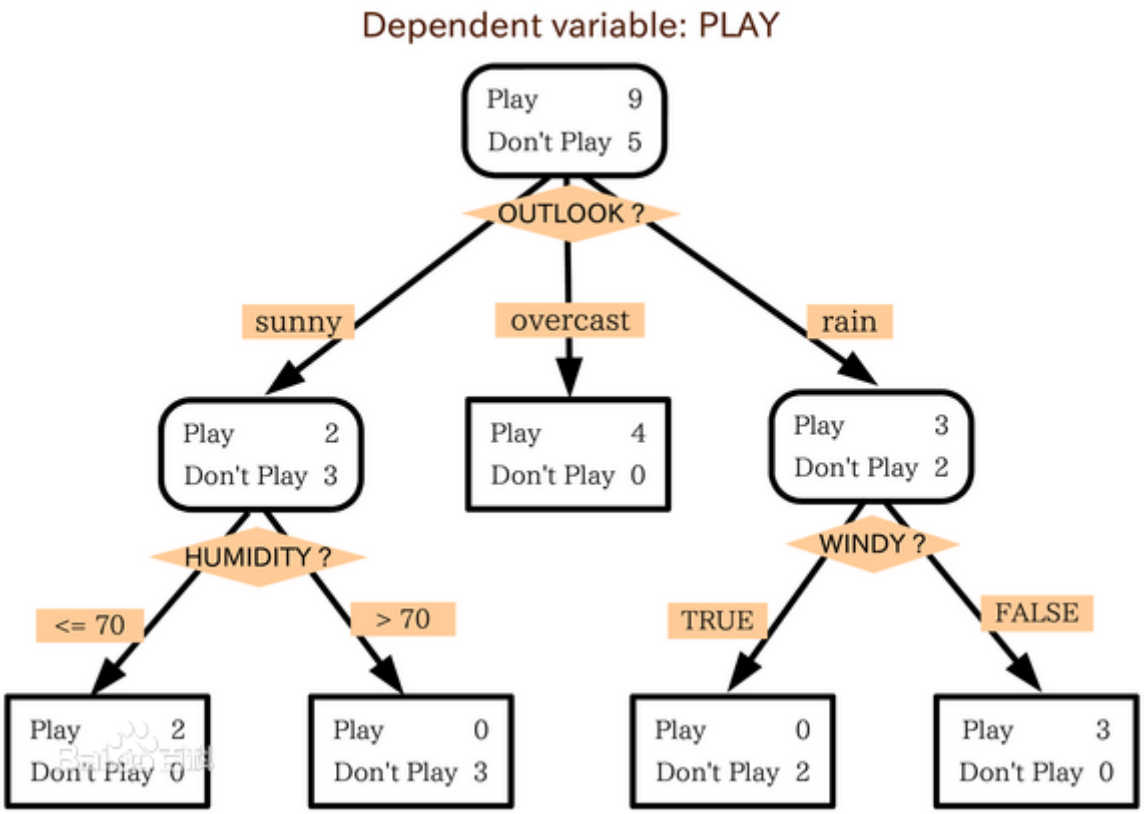
三、数学基础有多重要

对于数学基础知识，需要高中数学知识加上高数、线性代数、统计学、概率论，即使掌握的不是很完善，但是至少要知道概念，在用到的时候知道去哪查。如果基础不好，可以先看看吴军的《数学之美》，讲的比较通俗易懂。也可以边做边学，实践是检验真理的唯一标准，毕竟大多数人还是以工程实践为主，如果你想做研究理论的科学家，并不适合看本文。

四、入门级机器学习算法

4.1 决策树

判定树是一个类似于流程图的树结构：其中，每个内部结点表示在一个属性上的测试，每个分支代表一个属性输出，而每个树叶结点代表类或类分布。树的最顶层是根结点。



例：现有一个数据集，表示一些的人的年龄、收入、是否是学生、信用、是否会买电脑。年龄有年轻，中年，老年三种；收入有高中低；信用有一般和很好。数据及保存在AllElectronics.csv中。现在在有一个新的人（数据），要判断这个人是否会买电脑。


```
allElectronicsData = open(r'D:\deeplearning\AllElectronics.csv', 'rb')
reader = csv.reader(allElectronicsData)
headers = reader.next()

print(headers)
featureList = []
labelList = [] #最后一列

for row in reader:
    #print(row)
    labelList.append(row[len(row)-1])  #在元祖末尾添加元素
    rowDict = {}
    for i in range(1,len(row)-1):
        rowDict[headers[i]] = row[i]
    featureList.append(rowDict)
print(featureList)
print(labelList)

vec = DictVectorizer()
dummyX = vec.fit_transform(featureList).toarray()
print("dummyX:" + str(dummyX))
print(vec.get_feature_names())

lb = preprocessing.LabelBinarizer()
dummyY = lb.fit_transform(labelList)
print("dummyY:" + str(dummyY))
```

```
clf = tree.DecisionTreeClassifier(criterion='entropy')
clf = clf.fit(dummyX,dummyY)
print("clf: "+ str(clf))

with open("allElectronicInformationGainDri.dot",'w') as f:
    f =
tree.export_graphviz(clf,feature_names=vec.get_feature_names(),out_file
= f)  #在当前工作目录生成  .dot 文件

oneRowX = dummyX[0, :]
print("oneRowx: " + str(oneRowX))

newRowX = oneRowX

newRowX[0] = 1
newRowX[2] = 0
print("newRowX: " + str(newRowX))


predictedY = clf.predict(newRowX)
print("predictedY:" + str(predictedY))
```


4.2 最临近取样

最临近取样就是把已有数据分成几类，对新输入的数据计算与已知数据的距离，距离哪一个近，就把新数据分到哪一类，例如下图所示的电影分类，对于最后一行未知电影类型的电影，根据打斗次数和接吻次数，距离浪漫型更近，应该被归类为浪漫型电影。

电影名称	打斗次数	接吻次数	电影类型
California Man	3	104	Romance
He's Not Really into Dudes	2	100	Romance
Beautiful Woman	1	81	Romance
Kevin Longblade	101	10	Action
Robo Slayer 3000	99	5	Action
Amped II	98	2	Action
未知	18	90	Unknown

例：irisdata.txt实在网上下载的鸢尾属植物数据集，根据数据集合，对新的数据进行分类

```
# coding:utf-8

#不调用库，自己实现knn算法
import csv    #读取CSV文件用的模块，读取数据用的
import random #随机数计算
import math  #数学计算
import operator
from bokeh.util.session_id import random
from boto.beanstalk import response
from dask.array.learn import predict



# 装载数据集  filename:数据集文件名 split：以数据集中某个位置为结点，把数据集分为trainingSet和testSet
def loadDataSet(filename, split, trainingSet=[], testSet=[]):
    with open(filename, 'rb') as csvfile:
        lines = csv.reader(csvfile)  #把所有行存入lines
        dataset = list(lines) #把数据转换为list格式
        for x in range(len(dataset)-1):
            for y in range(4):
                dataset[x][y] = float(dataset[x][y])
            if random.random() < split:   #如果随机值小于split
                trainingSet.append(dataset[x])  #则加到trainingSet
            else:
                testSet.append(dataset[x])


#欧几里德距离 ：坐标差的平方的和再开根号      还有曼哈顿距离
def euclideanDistance(instance1, instance2, length):
    distance = 0
    for x in range(length):
        distance += pow((instance1[x] -instance2[x]), 2)
    return math.sqrt(distance)

#返回距离testInstance最近trainingSet的K个邻居
def getNeighbours(trainingSet, testInstance, k):
    distances = []
    length =len(testInstance) - 1
    for x in range(len(trainingSet)):
        dist = euclideanDistance(testInstance, trainingSet[x], length)
#每一个训练集数据和实例数据之间的距离
        distances.append((trainingSet[x],dist))
    distances.sort(key=operator.itemgetter(1)) #sort 排序为从小到大
#取前k个最近的neighbors
    neighbors = []
    for x in range(k):
        neighbors.append(distances[x][0])
    return neighbors
```

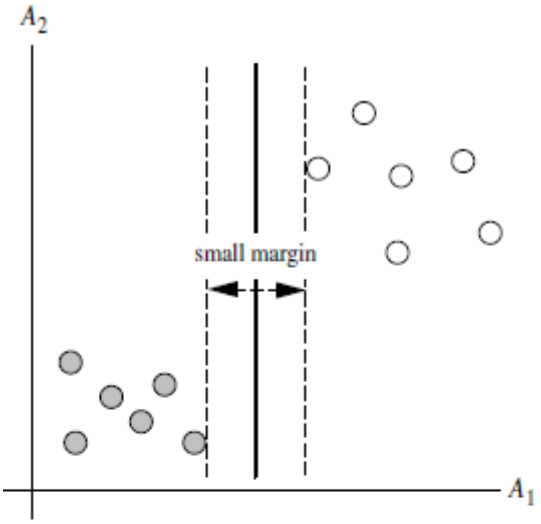
```
#根据少数服从多数的原则判断要预测实例属于哪一类。计算testInstance到trainingSet
距离最近的个数，返回最多的那一类
def getResponse(neighbors):
    classVotes = {}
    for x in range(len(neighbors)):
        response = neighbors[x][-1]
        if response in classVotes:
            classVotes[response] += 1
        else:
            classVotes[response] = 1
    sortedVotes = sorted(classVotes.iteritems(),
key=operator.itemgetter(1), reverse=True)
    return sortedVotes[0][0]

#获取预测的准确率testSet:测试数据集  predictions：用代码预测的类别集合
def getAccuracy(testSet, predictions):
    correct = 0
    for x in range(len(testSet)):
        if testSet[x][-1] == predictions[x]:  #-1表示数组的最后一个值。
            correct += 1
    return(correct/float(len(testSet))) * 100.0

def main():
    trainingSet=[]
    testSet=[]
    split = 0.67  #三分之二为训练集 ，  三分之一为数据集
    loadDataSet(r'C:\Users\ning\workspace\KNNdata\irisdata.txt', split,
trainingSet, testSet)
    print 'Train Set:  ' + repr(len(trainingSet)) #repr 转化为字符串
    print 'Test Set:  ' + repr(len(testSet))

    predictions = []
    k = 3
    for x in range(len(testSet)):
        neighbors = getNeighbours(trainingSet, testSet[x], k)
        result = getResponse(neighbors)
        predictions.append(result)
        print("> predicted=" + repr(result) + ', actual=' +
repr(testSet[x][-1]))
    accuarcy = getAccuracy(testSet, predictions)
    print('Accuracy: ' + repr(accuarcy) + '%')
main()
```

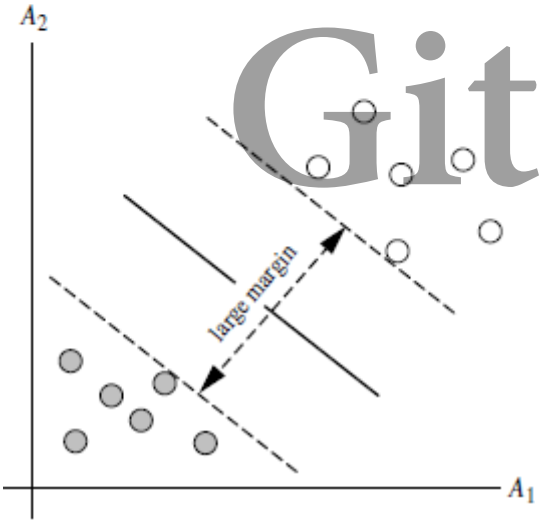

4.3 支持向量机

支持向量机（SVM）是从线性可分情况下的最优分类面发展而来。最优分类面就是要求分类线不但能将两类正确分开(训练错误率为0),且使分类间隔最大。SVM考虑寻找一个满

足分类要求的超平面,并且使训练集中的点距离分类面尽可能的远,也就是寻找一个分类面使它两侧的空白区域(margin)最大。

这两类样本中离分类面最近的点且平行于最优分类面的超平面上H1,H2的训练样本就叫做支持向量。





例：使用sklearn库实现svm算法， 俗称调库，实际上调库是一个很简单的过程，初级阶段甚至都不需要知道原理。

```
# coding:utf-8
from sklearn import svm
X = [[2,0], [1,1], [2,3]]
y = [0,0,1]
clf = svm.SVC(kernel = 'linear')
clf.fit(X,y)  #通过.fit 函数已经可以算出支持向量机的所有参数并保存在clf中

print clf

# get support vectors
print clf.support_vectors_
```

```
#get index of support vectors
print clf.support_

#get number of support vectors for each class
print clf.n_support_

#predict data ,参数是二维数组
print clf.predict([[2, 0], [10,10]])
```

五、书单推荐

- 《数学之美》吴军
- 《机器学习》 周志华
- 《漫谈人工智能》 集智俱乐部
- 《机器学习实战》 Peter Harrington
- 《TensorFlow技术解析与实战》 李嘉璇
- 《统计学习方法》 李航

六、学习人工智能的误区-人工智能又是一个泡沫？

人工智能很大程度上被一些科技巨头公司夸大了，为了拿到资本的钱，这也在情理之中，但是普通大众一定要有自己的鉴别能力，客观地分析自己到底是否适合做这一行。纵观互联网发展史，人工智能这种发展态势并不是首例，像2014年爆红的O2O模式，那时候不懂点O2O都不敢说自己是互联网圈的人，到现在，一批又一批的创业大军倒下去，当然也会留下像亚马逊、阿里巴巴这样的巨头，每个行业都有它的金字塔。

我上大二的时候可以说3D打印、VR技术处在风口浪尖，各种3D打印创业公司、VR创业公司层出不穷，大四就已经开始倒了一家又一家，包括我也做过3D打印方面的项目，实际上做的东西也不过时改进一些边边角角的东西，最核心的框架早已被大牛们设计好了。盲目追随科技的潮流，我们永远只能在潮流的后面。 最近看CCTV上都已经有撒贝宁主持的人工智能综艺节目了，这说明人工智能早已成了一片红海，与现在的移动互联网技术并没有本质上的区别。自从谷歌开源tensorflow框架（还有很多其他优秀的框架），写机器学习的代码很多都是调调参数，有的甚至都不用知道原理，当然大牛肯定是有，还是那句话，每个行业都有的它的金字塔，只不过到达塔尖的路径不同，在我看来，调用tensorflow的框架进行人工智能的开发与调用android的API开发app并没有本质的区别，真正伟大的是谷歌公司，后来者只不过是追随者。

题外话，不知道大家是否听过**21**世纪是生物的世纪，这一概念兴起之时，众多高考生选择生物相关的专业。之前有个对国内某著名高校生物专业毕业生的就业去向调查，其中一个结论是生物专业学生最好的出路就是离开这个专业。当然我们不得不说生物技术跟我们每个人的生活息息相关，但是其发展周期之长，又怎是一个人等得起的？如何把个人认同与社会认同，自我价值与社会价值协调统一，也是我们需要思考的问题。

人工智能是否是个泡沫？这个概念还能火多久？

第六部分内容纯属个人观点，仅供参考。

老罗给您吟诗一首