**基于WEKA的数据挖掘算法学习（三）**

# 决策树的理论知识

决策树是一类常用的机器学习方法，基于树的结构来进行决策，是面临决策问题是自然的一种处理机制。决策树的学习目的是为了产生一棵泛化能力强，即处理未见示例能力强的决策树，其生成是一个递归的过程。

## 1.1划分选择

如何选择最优划分属性是决策树学习的关键，随着划分过程不断进行，希望决策树的分支结点所包含的样本尽可能属于同一类别，即“纯度”越来越高。

### 1.1.1信息增益（ID3决策树学习算法）

“信息熵”是度量样本集合纯度最常用的指标，假定样本集合D中第k类样本所占比例为(k=1,2,…,|y|)，则D的信息熵定义为：

的值越小，则D的纯度越高。根据信息熵公式计算出某个属性a的样本的信息熵，再考虑不同分支结点所包含样本数不同，给其赋予权重，样本数越多的分支结点的影响越大，于是计算出“信息增益”：

信息增益越大，属性a来划分属性所获得的“纯度提升”越大，于是使用信息增益来进行决策树的划分属性选择。ID3算法即选择以信息增益为准则来选择划分属性。

### 1.1.2增益率（C4.5决策树学习算法）

信息增益准则对可取数目较多的属性有所偏好，为减少这种偏好带来的不利影响，使用“增益率”来选择最优划分属性，而不直接使用信息增益。由于增益率对数目较少的属性有所偏好，于是，著名的C4.5决策树算法的思想使用了启发式：先从候选划分属性中找出信息增益高于平均水平的属性，再从中选择增益率最高的。其中增益率的定义为：

,其中

### 1.1.3基尼指数（CART决策树学习算法）

CART决策树使用基尼指数来选择划分属性，数据集D的纯度可用基尼值来度量：

越小，数据集D的纯度越高。在候选属性集合A中，选择使得划分后基尼指数最小的属性作为最优划分属性。而属性a对D进行划分的基尼指数为

## 1.2剪枝处理

剪枝是决策树学习算法中对付“过拟合”的主要手段，即有时会造成把训练集自身的一些特点当做数据所具有的一般性质而导致过拟合。

### 1.2.1预剪枝

对每个结点在划分前进行估计，若当前结点的划分不能带来决策树泛化性的提升，则停止划分且将当前结点标记为叶结点。预剪枝基于“贪心”本质，给预剪枝决策树带来了欠拟合的风险。

### 1.2.2后剪枝

在训练集生成一棵完整的决策树后，自底向上对非叶结点进行考察，若将该结点对应的子树替换为叶结点能带来决策树泛化性能提升，则将该子树替换为叶结点。时间效率上低于预剪枝或未剪枝决策树。

## 1.3连续与缺失值

### 1.3.1连续值处理

连续值属性的可取值数目不再有限，连续属性离散化技术可作为解决方法，最简单是使用二分法，C4.5也是采用这种方案。

### 1.3.2缺失值处理

样本某些属性值缺失后，利用去掉这部分内容，进行无缺失值下的属性分类，相当于讨论每个属性时，去掉缺失的部分，再按照样本的比例来赋予权重，C4.5同样采用了这种方法。

# 2.神经网络的理论知识

## 2.1神经元模型

神经网络中最基本的成分是神经元模型，由生物神经网络抽象出来。神经元接收到来自n个其他神经元传递过来的输入信号，输入信号通过带权重的连接进行传递，神经元接收到的总输入值将与神经元的阈值进行比较，然后通过“激活函数”处理以产生神经元的输出。

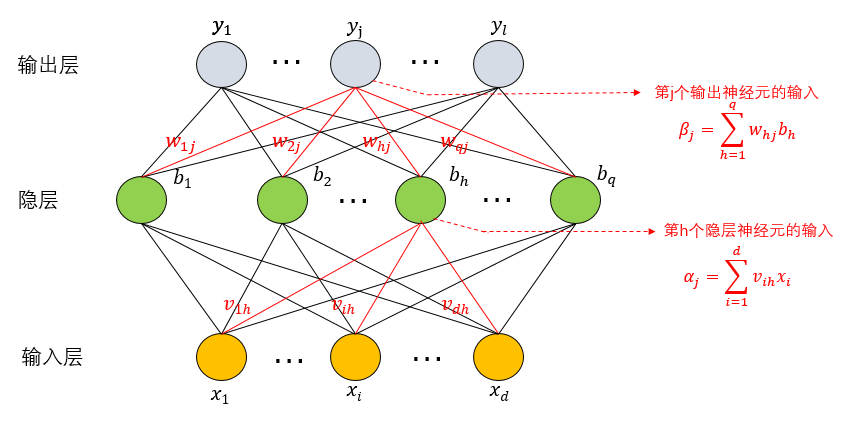
典型的神经元激活函数：

Sigmoid函数为，

把很多个的神经元按一定的层次结构连接起来，就得到了神经网络。

## 2.2误差逆传播算法（BP网络）

通过说“BP网络”时，一般指BP算法训练的多层前馈神经网络，包括输入层、隐层、输出层。



**图1 BP神经网络及算法中的变量符号**

神经网络中根据隐层神经元的误差来更新连接权与阈值是最重要的环节，而BP算法的目的是最小化训练集D上的累积误差：

隐层到输出层的连接权的推导：BP算法基于梯度下降策略，以目标的负梯度方向对参数进行调整，对误差，学习率η，有

其中，，又,

联立Sigmoid函数的性质：，可得

## 2.3其他常见神经网络

### 2.3.1RBF网络

全局逼近网络由于每次输入网络上每一个权值均需要调整，从而导致全局逼近网络的学习速度很慢，而RBF网络为局部逼近网络。通常采用两步过程来训练RBF网络：1、确定神经元中心，常用方式包括随机采样、聚类等；2、利用BP算法来确定参数。

### 2.3.2ART网络

竞争型学习是神经网络中常用的无监督学习策略，网络的输出神经元相互竞争，每一时刻仅有一个竞争获胜的神经元被激活，其他神经元的状态被抑制。明显，识别阈值就显得对ART网络的性能有重要影响，识别阈值较高的时候，输入样本将会被分成比较多、比较精细，反之则反。

### 2.3.4深度学习

计算能力的大幅度提高可缓解训练低效性，训练数据的大幅度增加则可降低过拟合风险，深度学习为代表的复杂模型就引入了进来。

典型的深度学习模型是很深层的神经网络，提高容量的一个简单方法：增加隐层数目，隐层数目多了，相应的神经元连接权、阈值等参数就会变多，且增加隐层数目显然比增加隐层神经元的数目更加有效。因为不仅增加了神经元数目，还增加了激活函数嵌套的层数。

策略1：预训练+微调，在预训练全部完成后，再对整个网络进行“微调”训练。

策略2：节省训练开销的“权共享”，让一组神经元使用相同的连接权，这个策略在卷积神经网络（CNN）中发挥了重要的作用。

这里重点谈一谈卷积神经网络：Deep Learning是全部深度学习算法的总称，而CNN是深度学习算法在图像处理领域的一个应用。CNN主要有两种网络结构，分别是卷积层和池化层。卷积层的作用是提取图像的各种特征；池化层的作用是对原始特征信号进行抽象，从而大幅度减少训练参数，另外还可以减轻模型过拟合的程度。

# 3.weka算法实现

## 3.1数据集的准备

1、UCI完备数据集：segment.csv

2、UCI不完备数据集：horse.csv

3、UCI不平衡数据集： glass.csv

## 3.2 J48

选择五折交叉数据验证，不断调整minNumObj参数进行数据测试，结果如下：

M=2时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.880 | 0.881 | 0.880 | 0.934 | 88.095 % | 0.12s |
| horse | 0.675 | 0.683 | 0.678 | 0.742 | 68.33% | 0.03s |
| glass | 0.973 | 0.972 | 0.972 | 0.983 | 97.196% | 0s |

M=3时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.871 | 0.871 | 0.871 | 0.934 | 87.14 % | 0.3s |
| horse | 0.712 | 0.720 | 0.715 | 0.777 | 72% | 0.05s |
| glass | 0.973 | 0.972 | 0.972 | 0.983 | 97.196% | 0s |

M=4时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.867 | 0.867 | 0.866 | 0.942 | 88.67 % | 0.17s |
| horse | 0.711 | 0.720 | 0.714 | 0.781 | 72% | 0.03s |
| glass | 0.973 | 0.972 | 0.972 | 0.983 | 97.196% | 0s |

M=5时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.863 | 0.862 | 0.861 | 0.936 | 86.19% | 0.01s |
| horse | 0.712 | 0.723 | 0.714 | 0.761 | 72.33% | 0.06s |
| glass | 0.977 | 0.977 | 0.977 | 0.985 | 97.66% | 0s |

M=6时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.868 | 0.867 | 0.866 | 0.941 | 86.67% | 0.03s |
| horse | 0.703 | 0.710 | 0.705 | 0.755 | 71% | 0.59s |
| glass | 0.977 | 0.977 | 0.977 | 0.985 | 97.66% | 0 s |

根据对实验数据结果分析，数据集segment在M=2时效果最好，数据集horse在M=3时效果最好，而数据集glass在M=5,6效果最好。在J48下三者整体时间效率均很高。还可更改confidencefactor（置信因素）进行数据测试，这里选择默认值C=0.25。

## 3.3 SimpleCart

选择五折交叉数据验证，不断调整minNumObj参数进行数据测试，结果如下：

M=2时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.876 | 0.871 | 0.873 | 0.930 | 87.12 % | 0.68s |
| horse | 0.698 | 0.703 | 0.700 | 0.721 | 70.33% | 0.22s |
| glass | 0.982 | 0.981 | 0.981 | 0.989 | 98.13% | 0.03s |

M=3时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.862 | 0.857 | 0.858 | 0.933 | 85.714 % | 0.19s |
| horse | 0.682 | 0.690 | 0.685 | 0.701 | 69% | 0.22s |
| glass | 0.982 | 0.981 | 0.981 | 0.989 | 98.13% | 0.05s |

M=4时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.837 | 0.833 | 0.834 | 0.937 | 83.33 % | 0.11s |
| horse | 0.688 | 0.697 | 0.691 | 0.730 | 69.67% | 0.12s |
| glass | 0.982 | 0.981 | 0.981 | 0.989 | 98.13% | 0.01s |

M=5时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.828 | 0.829 | 0.828 | 0.939 | 82.82% | 0. 1s |
| horse | 0.690 | 0.693 | 0.692 | 0.718 | 69.33% | 1.91s |
| glass | 0.982 | 0.981 | 0.981 | 0.989 | 98.13% | 0.01s |

根据对实验数据结果分析，数据集segment和数据集horse在M=2时效果最好，而数据集glass在M改变时，在性能评估上改变不大，只是时间效率上有改变，从数据集本身分析，应该是数据集样本数目太少而导致测试结果差距不大。

## 3.4 Multilayerceptron

选择五折交叉数据验证，不断调整learncate（学习率）参数，（其在weka中默认值为0.3）进行数据测试，结果如下：

L=0.01时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.750 | 0.757 | 0.741 | 0.952 | 75.71 % | 1.93s |
| horse | 0.698 | 0.703 | 0.700 | 0.715 | 70.33% | 2.38s |
| glass | 0.703 | 0.804 | 0.769 | 0.961 | 80.37% | 0.83s |

L=0.1时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.917 | 0.914 | 0.914 | 0.997 | 91.43 % | 1.57s |
| horse | 0.707 | 0.700 | 0.703 | 0.707 | 70% | 2.36 s |
| glass | 0.961 | 0.958 | 0.958 | 0.911 | 95.79% | 0.83s |

L=0.2时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.907 | 0.905 | 0.905 | 0.975 | 90.48 % | 1.59s |
| horse | 0.643 | 0.647 | 0.645 | 0.695 | 64.67% | 2.28s |
| glass | 0.949 | 0.949 | 0.949 | 0.989 | 94.86% | 0.83s |

L=0.3时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.903 | 0.900 | 0.900 | 0.979 | 90% | 1.56s |
| horse | 0.673 | 0.672 | 0.672 | 0.696 | 67.33% | 2.35s |
| glass | 0.939 | 0.939 | 0.938 | 0.991 | 93.93% | 0.82s |

L=0.4时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.901 | 0.900 | 0.898 | 0.982 | 90% | 1.54s |
| horse | 0.688 | 0.690 | 0.689 | 0.691 | 69% | 2.34s |
| glass | 0.939 | 0.939 | 0.939 | 0.991 | 93.93% | 0.8s |

虽然学习率默认值为0.3，单根据实验结果来看，三种数据集跑出来的结果都是在L=0.1时最佳，因此调整数据集样本最适合的参数可以尝试在0.1附近的区间进行选择。但也不能过于小（如L=0.01时，效果非常的差）。

BP算法是基于误差-修正学习的，因此，学习率的大小对收敛速度和训练结果影响很大。如果学习率太小，学习速度太慢；如果学习率太大，可能导致振荡或发散。为实现快速而有效的学习收敛过程，必须调整得出最适合的学习率。

神经网络还可尝试更改的参数有Momentum（动量）、Weight Decay（权值衰减）、Learning Rate Decay（学习率衰减）。

# 4.本周学习心得体会

本周着重加强了算法理论知识的学习，学习了《机器学习 周志华版》中第4章 <决策树>以及第5章<神经网络>的算法理论。

对决策树的基本流程和划分选择引申的三种决策树学习算法：ID3、C4.5、CART进行了学习，以及一些处理过拟合的方法的理解；对神经网络详细理解了神经网络模型，并对误差传播算法其中的公式推导进行了理解。以及了解了其他常用的神经网络，如深度学习。