进行过采样, 首先为每个少数类样本选出几个近邻的样本, 将每个样本与其近邻的几个样本相连接, 然后在连接线上随机取点作为新生成的样本, 这些新样本与原始样本并不相同, 从而扩大数据量少的类别, 使得不同类别的恒星光谱具有大致相同的数据量. SMOTE 算法也是可以直接调用 SMOTE 库来实现数据的平衡.

核心代码如算法 2.

算法 2. 过采样 SMOTE 算法

```
from imblearn.over_sampling import SMOTE #导入 SMOTE 库
print(sorted(Counter(y_train).items()) #查看#原始数据中不同类别恒星光谱的数据量
X_train,Y_train=SMOTE().fit_sample(x_train,y_train)
#对原始数据进行过采样处理
print(sorted(Counter(y_train).items()) #得#到不同类别的恒星光谱具有相同的数据量
```

对数据的处理完成后, 就需要采用数据挖掘中的分类算法对数据进行分类. 之所以采用 BP 神经网络算法, 是因为其具有其他算法所不具有的优点. BP 神经网络实现了输入数据经过网络的传输, 最终输出的过程, 在这个过程中, BP 神经网络能够以任意精度逼近非线性连续函数, 具有很强的非线性映射能力; BP 神经网络在训练数据的过程中, 能够提取出输入数据与输出数据之间的“规则”, 并自适应的将“规则”记忆于网络的权值之中, 所以 BP 神经网络具有自适应能力; BP 神经网络具有着较为复杂的网络结构, 当其中的部分神经元受损后, 不会对全局的 BP 神经网络结构造成太大的影响, 甚至仍然可以正常工作, 具有一定的容错能力. BP 神经网络算法适合处理内部机制复杂的问题, 而且具有较强的自适应能力与容错能力, 而恒星光谱数据量大, 数据关系复杂, 所以采用 BP 神经网络算法对恒星光谱数据进行分类是很合适的选择.

3 BP 神经网络算法研究

3.1 BP 神经网络原理

神经网络, 顾名思义, 模仿人脑中的传输过程, 首先是输入信号进行输入, 经过神经元的层层传输以及处理, 最终输出反馈, 即得到输出结果. BP 神经网络是一种包括 3 层或 3 层以上的阶层型神经网络, 标准的 BP 神经网络模型有 3 层, 包括输入层、隐含层、输出层, 如图 1 所示. BP 神经网络具有一层输入层, 一层输出层, 可包含多层隐含层, 具有很强的记忆与泛化能力[8], 通过对网络中的权值和阈值不断地进行调整, 来降低预测误差平方和.

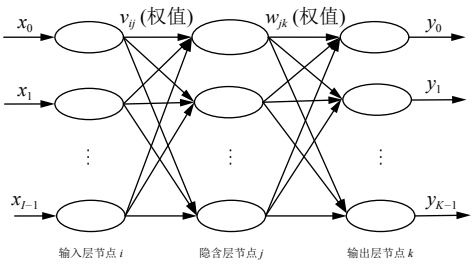

图 1 3 层 BP 神经网络结构示意图

利用 BP 神经网络系统实现对恒星的分类, 其过程类似于上述中信号在神经网络中的传输过程. 可以把目标源的轮廓信息、颜色信息、光谱能量分布信息作为输入数据, 输入到神经网络系统的输入层, 神经网络系统按照一定的规则、特定函数来对这些信息进行判断、处理、分析, 输出层就会给出目标源信息的特征量, 然后分析、总结输出层输出的数据信息, 判断、总结目标源的特点, 从而可以实现对目标源的分类. BP 神经网络包括两个阶段, 第一个阶段称为正向传播过程, 输入数据从输入层输入神经网络, 经过层层网络的计算、处理, 得到输出层各个单元的输出值. 记 BP 神经网络的输入层节点数为 I, 输入向量记为 $X_p=(x_0, x_1, x_2, \cdots, x_{I-1})$. 输出层的节点数为 K, 隐含层节点数为 J, 输入层节点数一般表示数据集中对于数据的属性描述有 I 个, 将属性描述对应的属性值作为 BP 神经网络的输入值, 而数据集中每个样本的分类类别作为 BP 神经网络的期望输出值, 输出值为 K 个, 隐含层第 j 个神经元的阈值 α_j 来表示, 输出层第 k 个神经元的阈值用 β_k 来表示, 输入层第 i 个神经元与隐含层第 j 个神经元间的权值用 v_{ij} 来表示, 隐含层第 j 个神经元与输出层第 k 个神经元间的权值用 w_{jk} 来表示. 对于 BP 神经网络, 总会有一个理想化的输出向量, 称为期望输出向量, 表示为 $D_p=(d_1, d_2, d_3, \cdots, d_{K-1})$

输入层数值表示:

$$O_i = x_i,\ \ i = 0,1,2,\cdots,I-1 \tag{2}$$

输入层的每个神经元的输入值分别与其连接的隐含层的第 j 个神经元对应的权值相乘, 然后求和, 得到隐含层中第 j 个神经元的输入值, 表示为:

$$net_j = \sum_{i=0}^{I} v_{ij} O_i,\ \ i = 0,1,2,\cdots,I-1 \tag{3}$$

隐含层的输出: 隐含层的第 j 个神经元输入值 net_j 经过激活函数的处理得到其输出值:

$$O_j = f\left(net_j\right),\ \ j = 0,1,2,\cdots,J-1 \tag{4}$$

其中, 激活函数一般选取 Sigmoid 函数[9]:

$$f(net) = \frac{1}{1+e^{-net}} \tag{5}$$

隐含层的每个神经元的输出值分别与其连接的输出层的第 k 个神经元对应的权值相乘, 然后求和, 作为输出层中第 k 个神经元的输入值, 表示为:

$$net_k = \sum_{j=0}^{J} w_{jk} O_j,\ \ i,j = 0,1,2,\cdots,J-1 \tag{6}$$

输出层的输出: 输出层的第 k 个神经元输入值 net_k 经过激活函数的处理得到其输出值:

$$O_k = f(net_k),\ \ k = 0,1,2,\cdots,K-1 \tag{7}$$

BP 神经网络的实际输出与期望输出之间总会有一定的差别, 把差别称为误差, 对于每个样本的误差函数的计算公式为:

$$E_p = 1/2 \sum_{k=0}^{K-1} \left(d_k^p - O_k^p\right)^2 \tag{8}$$

如果数据集中有 N 个样本, 则 N 个样本的总误差计算公式为:

$$E = \frac{1}{2N} \sum_{p=0}^{N-1} \sum_{k=0}^{K-1} \left(d_k^p - O_k^p\right)^2 \tag{9}$$

其中, E_p 表示 p 的输出误差, d_k^p 表示样本 p 的期望输出, O_k^p 表示 BP 神经网络的实际输出.

第二个阶段为反向传播过程, 经输出层输出的数据得到输出误差, 输出误差反向向前传输经过各隐含层, 计算隐含层各单元的误差, 再根据此误差来对前层的权值进行修正. 具体过程为:

首先对 BP 神经网络中的输出层与隐含层之间的连接权值进行调整, 输出层的误差表示为[10]:

$$\delta_k = (d_k - O_k) f'(net_k) \tag{10}$$

对于输出层与隐含层之间的权值调整为[10]:

$$\Delta w_{jk} = -\eta O_j (d_k - O_k) O_k (1 - O_k) \tag{11}$$

其中, η 为学习步长, 取值区间为 (0,1).

然后对 BP 神经网络中的隐含层与输入层之间的连接权值进行调整, 隐含层的误差表示为[10]:

$$\delta_j = f'\left(net_j\right) \sum_{k=0}^{K-1} \delta_k w_{jk} \tag{12}$$

对于隐含层与输入层之间的权值调整为[10]:

$$\Delta v_{ij} = \eta O_j \left(1 - O_j\right) \sum_{k=0}^{K-1} \delta_k w_{jk} O_i \tag{13}$$

BP 神经网络算法的流程图如图 2 所示.

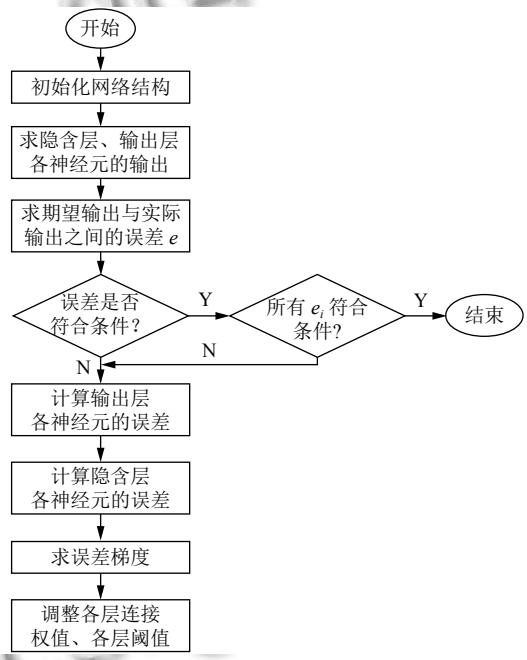

图 2 BP 神经网络算法流程图

3.2 BP 神经网络实现

读取存储恒星光谱的光谱文件, 提取出在不同波长处的光谱波长信息以及光谱类别信息, 每个文件中共有 3909 个波长数据, 提取出光谱信息后, 将其中不是恒星以及恒星子类别不明确的光谱数据进行剔除, 将光谱波长数据中的冗余数据错误数据进行删除, 然后对数据进行降维操作, 选择将光谱波长数据降维成 22 维数据, 既提取出了其中的重要数据信息, 又减少了数据量, 方便对数据的操作处理. 然后对数据进行归一化处理, 消除数据之间的数量级差距, 同时也减小数据, 降低对数据的计算量. 还要对数据进行平衡处理, 通过采样操作使不同类别的恒星光谱数据具有相似的数

据量.

将恒星光谱的波长信息作为 BP 神经网络的输入, 类别信息作为 BP 神经网络的期望输出, 输入数据经过 BP 神经网络层层传递以及反馈, 经过 BP 神经网络权值、阈值的不断修正, 使输入数据经过 BP 神经网络的传输后, 实际的输出结果能最大限度的接近期望输出, 从而使误差达到最下, 即较高的分类正确率.

BP 神经网络隐含层的确定可根据输入层与输出层节点数目来确定, 假设 BP 神经网络的输入层节点为 m 个, 输出层节点为 n 个, 则隐含层节点的选取范围为$(m+n)^{\frac{1}{2}}+a$, $a\in[1,10)$, 当设置 BP 升级网络的输入层节点为 22, 隐含层节点为 13, 输出层为 5, 基于交叉验证的 BP 神经网络得到的结果有时为 70.34%, 而有时又可达 76.66%, 其中的正确率还是有一定差距的, 因为每次选取的训练集、测试集是不同的, 所以导致最终的预测结果也是不同的.

4 基于交叉验证的 BP 神经网络

4.1 交叉验证

交叉验证是一种估计泛化误差的模型选择方法, 在进行误差估计之前不需要任何假设条件, 操作简单、方便, 交叉验证在各种类型的模型选择中都可适用, 所以其应用非常广泛[11]. 当分类算法在对数据中的训练集进行训练后得到训练模型, 如果用此训练模型再去对与训练集有交集数据的测试集进行误差估计, 会导致预测误差非常低, 得到错误的预测结果.

经常用的交叉验证方法为 K-折交叉验证方法 (K-folder Cross Validation, K-CV). K-折交叉验证是将数据集随机分成 k 份, 其中 k–1 份作为训练集, 1 份作为测试集, 在训练的过程中, 依次从 k 份中选择一份作为测试集, 剩余的 k–1 份为训练集, 每次用训练集进行模型训练, 用训练出的模型对测试集进行测试, 进行结果预测, 得到预测的正确率与误差[12]. K 份数据就需要进行 k 次训练与测试, 最后将得到的 k 个结果求平均值作为最终预测结果. K-折交叉验证的优点是数据集中的所有数据都作为了训练集和测试集, 每个数据样本都被验证过, 这样可使预测结果更具有客观性, 对预测结果求平均值也降低了偶然性误差.

在数据需要进行过采样时, 需要注意与交叉验证过程的顺序问题. 如果在交叉验证之前进行过采样同样会导致过拟合的问题, 因为在对数量小的类别进行过采样后, 对数据集进行随机划分, 不能保证测试集与训练集中的数据没有交集. 用一个例子来说明, 如图 3, 最左边一列为原始数据, 假设包含 3 个类别, 其中两个类别的数据量较小; 接下来如果首先对少数类别的数据进行过采样操作, 得到图中第 2 列数据集; 然后进行交叉验证中的数据划分过程, 会发现训练集与测试集中包含着同样的数据, 会导致最终结果产生过拟合的问题.

所以, 需要在交叉验证后对数据进行过采样操作, 如图 4 所示, 首先将验证样本从数据集中挑选出来, 剩余数据作为训练集; 然后对训练集中数量小的类别进行过采样, 得到如图中第 3 列所示, 此时训练集与测试集中的数据是没有重复的, 会防止预测结果产生过拟合. 在进行 10 折交叉验证的过程中, 同样是在交叉验证的每次循环中做过采样, 即每次在训练集中插入与测试集无交集的数据来保证数据平衡.

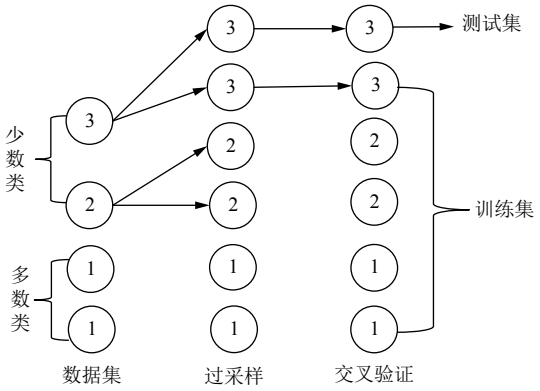

图 3 先过采样再交叉验证

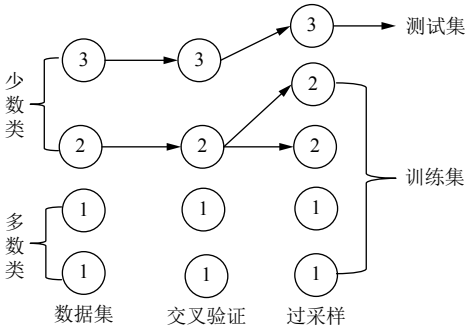

图 4 先过采样再交叉验证

4.2 基于交叉验证的 BP 神经网络

BP 神经网络中存储着诸多信息, 主要有两方面的信息, 一方面是网络的结构, 包括网络的输入层、隐含层以及输出层的节点个数以及隐含层的层数; 另一方面网络中连接每一层之间的权值. 网络中的输入层、输出层节点个数一般是可以确定的, 而隐含层层数与节点个数则由用户凭经验而设定, 设定的过小或过大都会造成不好的影响; 网络中的权值开始是在一个范围内进行初始化的, 在反向传播过程中, 会根据误差来进行调整. 用 BP 神经网络算法对数据进行训练、测试时, 测试的只是其中的一部分数据, 对这部分数据的测试结果并不代表对其他数据具有相似的测试结果, 而采用交叉验证的 BP 神经网络, 可使所有的数据都用于训练模型, 最终得到的结果会稳定、可靠; 实验过程中消除了数据分配的随机性而带来的误差影响, 确保实验的可重复性.

利用 10 折交叉验证进行 BP 神经网络, 将数据集平均分成 10 份, 每次从中选出 1 份作为测试集, 剩余 9 份作为训练集, BP 神经网络用训练集来训练出网络模型, 得到网络中的权值、阈值等, 确定了 BP 神经网络模型, 再用训练好的模型对测试集进行结果预测. 具体思想过程如图 5 所示, 将数据集平均分成 10 份, 分别用编号 1, 2, 3, …, 10 来表示, 第 1 次运行过程是将编号为 1 的那部分数据作为测试集, 剩余编号为 2~10 的 9 份数据作为训练集, 得到第 1 个测试结果 1, 第 2 次运行过程是将编号为 2 的那部分数据作为测试集, 剩余的编号为 1、3~10 的 9 份数据作为训练集, 得到测试结果 3, 以此类推, 最后一次训练过程是将编号为 10 的那部分数据作为测试集, 剩余的编号为 1~9 的 9 份数据作为训练集, 得到测试结果 10. 然后对 10 次的测试结果求平均值, 得多最终的测试结果.

4.3 实验结果

利用同样的数据作为实验的输入数据, 类别信息作为期望输出数据, 输入层节点为 22 个, 输出层节点为 5, 设置隐含层节点为 12, 基于交叉验证的 BP 神经网络得到的结果为: [72.92%, 79.12%, 79.17%, 67.7%, 65.6%, 78.1%, 80.2%, 69.8%, 74%, 76.7%]. 从 10 次预测的 10 个结果中可以发现, 最小的预测正确率为 65.6%, 而最大的预测正确率为 79.17%, 最大值与最小值之间还是有一定的差距的, 所以利用交叉验证, 进行 10 次训练与预测, 再求得 10 次预测结果的平均值, 会得到一个较为平稳、准确的数据结果为 74.331%. 对预测结果的影响因素有输入数据的质量、隐含层节点数的定义等, 所以在确定 BP 神经网络输入层节点、输出层节点后, 还需对隐含层节点的数量进行不同的设定来寻找较高的预测结果[13]. 基于大量实验, 发现当设定隐含层节点数为 14 时, 得到的较高的预测结果为: [82.24%, 78.5%, 80.37%, 83.18,%, 79%, 84.11%, 81.3%, 83.18%, 74.3%, 78.28%]. 从数据中同样可以发现, 最小值为 74.3%, 最大值为 84.11%, 最大值与最小值差距较大, 所以对 10 次预测结果求得均值为 80.446%.

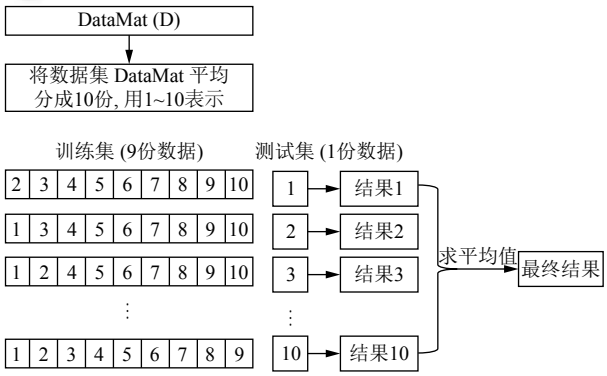

图 5 10 折交叉验证过程示意图

5 结语

在对数据进行分类前, 需要对数据进行噪声剔除、数据降维、数据规范化、数据平衡预处理, 可保证数据在训练过程中, 减少训练时间, 并且可使训练得到的 BP 神经网络更为准确, 从而可以得到较高的预测结果. BP 神经网络结构中的各层节点都是由数据来控制的, 但隐含层的节点的设定在一定范围内是随机的, 不同的隐含层节点的设定 BP 神经网络的预测结果是有一定的影响的, 所以需要基于大量的实验来确定使 BP 什么网络具有最高预测性能的隐含层节点数目. 基于交叉验证的 BP 神经网络, 几乎所有的数据都经过了 BP 神经网络算法的训练, 既可以在对数据进行过采样的过程中防止产生过拟合, 训练出的 BP 神经网络更为稳定, 降低了随机性, 而且多次训练得到的预测结果再求均值, 可以避免偶然的误差对结果造成的影响, 使测试结果更接近正确的、真实的结果.

参考文献

1 毕立鹏. LAMOST 低质量光谱交互式分析平台的设计与实现[硕士学位论文]. 济南: 山东大学, 2016.

2 赵永恒. 大规模天文光谱巡天. 中国科学: 物理学力学 天文学, 2014, 44(10): 1041–1048.

3 林雪梅. ANN 在天体光谱分类及恒星大气参数测量中的应用[硕士学位论文]. 济南: 山东大学, 2012.

4 艾丽雅. 天体光谱的分类算法研究[硕士学位论文]. 鞍山: 辽宁科技大学, 2016.

5 韦鹏. LAMOST 一维光谱自动处理[硕士学位论文]. 济南: 山东大学, 2011.

6 任利敬, 赵正旭, 陶智. 信息传承与长期保存技术策略研究. 兰台世界, 2016, (13): 25–27.

7 陈淑鑫, 孙伟民, 孔啸. LAMOST 恒星分类模板间相似性度量分析. 光谱学与光谱分析, 2018, 38(6): 1922–1925.

8 李老三, 辛军饷. 基于 BP 神经网络富水岩层围岩变形量预测. 重庆建筑, 2019, 18(4): 49–53. [doi: 10.3969/j.issn.1671-9107.2019.04.49]

9 贾伟, 赵雪芬. 改进量子粒子群 BP 神经网络参数优化及应用. 软件导刊, 2019, 18(10): 30–35.

10 王振武, 徐慧. 数据挖掘算法原理与实现. 北京: 清华大学出版社, 2015. 127–129.

11 阎少宏, 吴宇航. 基于交叉验证的级联 BP 神经网络的焦炭质量预测模型. 信息记录材料, 2018, 19(10): 223–224.

12 林悦, 夏厚培. 交叉验证的 GRNN 神经网络雷达目标识别方法研究. 现代防御技术, 2018, 46(4): 113–119. [doi: 10.3969/j.issn.1009-086x.2018.04.018]

13 丁常富, 王亮. 基于交叉验证法的 BP 神经网络在汽轮机故障诊断中的应用. 电力科学与工程, 2008, 24(3): 31–34. [doi: 10.3969/j.issn.1672-0792.2008.03.009]