# 1 xxx预测模型设计

机器学习（Machine Learning，ML）与深度学习（Deep Learning，DL）方法均基于统计方法寻找数据之间的关系，并在面对未知情况的数据时给出判断。这些学习算法根据输入数据和预期输出的结果示例，以及给定衡量算法效果的损失函数，经过不断训练和迭代优化，得到效果较好的预测模型。学习算法实现的假设前提是，训练与测试数据由相同的概率分布产生。给定训练数据集,称为输入地观测值，为输出示例。学习系统通过学习和训练得到模型，并进行性能度量，经验风险最小化的模型即为最优模型。在预测中，给定测试集样本的输入，并根据得到预测的输出结果，如图1.1所示。

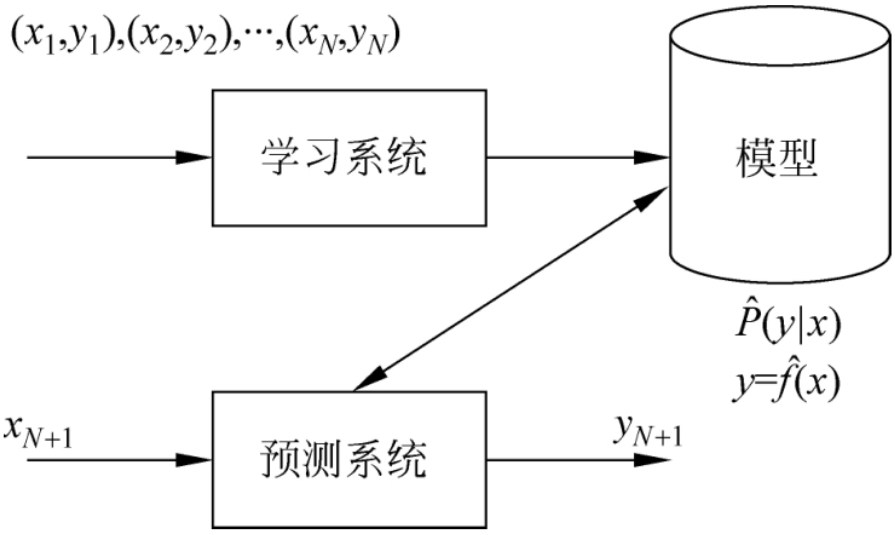


图1.1 学习算法的实现过程

传统的机器学习方法，也被称为“非深度”机器学习，更加依赖于人工介入进行学习。这类方法在各个领域得到广泛应用，种类繁多。通过学习数据的规律和模式，这些方法能够挖掘出一般性的数学规律，并用于对未知数据的预测。机器学习的核心思想是，计算机通过学习算法从复杂数据中提取规律。为了建立模型，需要为学习算法提供历史经验数据。在处理新的情况时，模型能够为我们提供相应的判断，从而实现对未知数据的预测或分类。预测的过程涉及从历史数据中学习系统演变的规律，以推测系统未来的发展趋势。机器学习方法主要分为监督学习、无监督学习和强化学习算法。监督学习是其中最常用的方法，它通过从标注数据中学习和预测模型，得到输入和输出的映射统计规律，可用于分类或回归任务。一些典型的监督学习算法包括线性回归、支持向量机（SVM）、K近邻（KNN）、决策树、朴素贝叶斯等算法。

神经网络也称为人工神经网络(ANN)，属于深度学习方法，由多个节点层组成，包含一个输入层、多个隐藏层和一个输出层。该模型通常比较复杂，从输入到输出的信息传递路径较长。每个节点也称为人工神经元，连接到其他节点，具有相关的权重和阈值。当任何单个节点的输出超过指定的阈值时，该节点被激活，并将数据传递到网络的下一层。反之，如果输出低于阈值，则该节点不传递任何数据。因此，神经网络可视为一种由一组参数控制的复杂函数，用于处理模式识别任务。训练神经网络是一个逐渐逼近目标函数的过程。

机器学习和深度学习方法在非线性问题的建模上有独特的优越性。有关研究表明，机器学习为预测电容xxx提供了有力帮助。如。。。。

以下对建立电容xxx预测模型所用的算法进行基本原理的介绍。

## KNN法

K近邻法（K-Nearest Neighbor，KNN）于1968年由Cover和Hart提出，是用于分类与回归的一种基本方法。在分类模型中，KNN法利用训练数据集对特征向量空间进行划分，计算训练样本到查询点（query point）的距离来分配标签。距离度量、k值的选择及分类决策规则是KNN法的三个基本要素。

KNN模型中，输入训练数据集，其中为实例的特征向量，为实例的类别；输出为示例所属的类。算法实现原理如下：

1. 根据给定的距离度量，在训练集中找出与最邻近的k个点，涵盖这k个点的的邻域记作；
2. 在中根据分类决策规则（如多数表决）决定的类别：

其中为指示函数，当时为1，否则为0。对于输入的实例点（特征向量），最近邻法将训练数据集中与最邻近点的类作为的类。

1. 模型的特征空间是维实数向量空间，使用的度量距离是欧氏距离，但也可以是更一般的距离（Lp-distance）：

k值的选择会对KNN法的结果产生重大影响，如图1.2所示。选择适当的k值是KNN算法中的重要任务，需要在模型的学习误差和估计误差之间找到平衡，以获得更好的预测性能。当k值较小时，相当于使用较小邻域中的训练实例进行预测，从而减小模型学习的近似误差。只有与输入实例较近的训练实例才会对预测结果产生影响，但这也带来了估计误差增大的缺点。此时，预测结果对近邻的实例点非常敏感，如果邻近实例点恰巧是噪声，可能导致错误的预测。较小的k值使整体模型变得复杂，容易发生过拟合。反之，选择较大的k值相当于使用较大邻域中的训练实例进行预测。这可以减少学习的估计误差，但缺点是学习的近似误差会增大。此时，与输入实例较远的训练实例也会对预测产生影响，可能导致错误的预测。较大的k值使整体模型变得简单，但也带来了一些缺点。

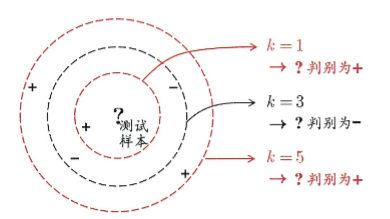


图1.2 KNN法

KNN法的分类决策通常为多数表决，即输入实例的k个邻近的训练实例中的多数类决定输入实例的类。

## 1.2决策树法

决策树法（Decision Tree，DT）最初于1986年提出，并经过不断改进衍生出了多种模型和算例。分类和回归树（Classification and Regression Tree，CART）算法是其中一种建立决策树模型的方法，使用单个特征和阈值将训练集划分为两个子集，最小化成本函数为。CART假设决策树是二叉树，内部结点特征的取值为“是”和“否”。这样的决策树等价于递归地二分每个特征，将输入空间即特征空间划分为有限个单元，并在这些单元上确定预测的概率分布，也就是在输入给定的条件下输出的条件概率分布。

CART是一种贪婪算法，从顶层开始对训练集进行搜索，将数据集分割为两部分。在每个层级上，CART算法重复这个过程，使用相同的逻辑对子集进行分割，然后继续分割子集，依此类推。一旦达到指定的最大深度或无法找到可减少不纯度的分割时，递归过程停止。CART算法包括两个主要步骤：决策树生成和决策树剪枝。在决策树生成阶段，算法基于训练数据集生成决策树，力图生成尽可能大的决策树；在决策树剪枝阶段，算法使用验证数据集对已生成的树进行剪枝，并选择最优的子树。此时，损失函数的最小化被作为剪枝的标准。决策树模型通常采用基尼（gini）不纯度作为度量方式。假设有K个类，样本点属于第k类的概率为  ，其定义为：

基于CART分类树的决策树算法实现原理如下。模型输入量为训练数据集和停止计算的条件；输出量为CART决策树模型；根据训练数据集，从根结点开始，递归地对每个结点进行以下操作，构建决策树：

1. 设结点的训练数据集为，计算现有特征对该数据集的基尼指数。此时，对每一个特征，对其可能取的每个值，根据样本点对的测试为“是”或“否”，将分割成和两部分，计算时的基尼指数；
2. 在所有可能的特征以及它们所有可能的切分点中，选择基尼指数最小的特征及其对应的切分点作为最优特征与最优切分点。依最优特征与最优切分点，从现结点生成两个子结点，将训练数据集依特征分配到两个子结点中去；
3. 对两个子结点递归地调用（1）（2）过程，直至满足停止条件。算法停止计算的条件是结点中的样本个数小于预定阈值，或样本集的基尼指数小于预定阈值（样本基本属于同一类），或者没有更多特征；
4. 生成CART决策树模型。

决策树算法易于理解和解释、用途广泛，同时也具有一些局限性，如对训练集分布较为敏感，泛化性和稳定性较差。为了避免可能存在的过拟合问题，可以根据需要对决策树进行剪枝，从已生成的树上剪掉一些叶结点或叶结点以上的子树，并将其父结点或根结点作为新的叶结点，从而简化生成的决策树。

## 1.3 BP神经网络

BP(back propagation)神经网络基于误差反向传播算法，将确定的样本输入和输出变成非线性优化问题，包含输入层、隐藏层（中间层）、输出层三部分，其基本结构如图1.2所示。

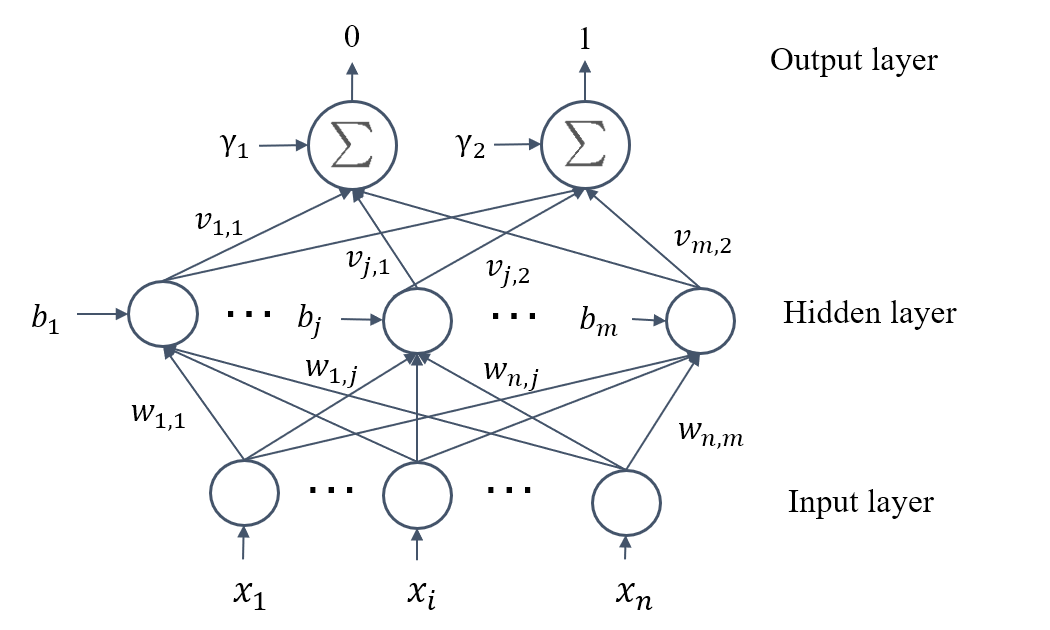


图1.2 BP神经网络拓扑结构

BP神经网络核心为反向传播算法，可以较好地解决各个输入贡献度分配的问题。传统的前向传播算法，按顺序从输入到输入层计算和存储神经网络中的每层结果。而反向传播算法计算神经网络的梯度参数，根据微积分中的链式法则，按相反的顺序从输出到输入层遍历网络。

当实际输出与期望误差较大时，选取梯度下降法修正神经元权重值，不断迭代更新模型参数，直至输出误差收敛到设定值时停止训练，导出模型。BP神经网络在初始训练时，对权重进行随机赋值，因此仅实现了一系列随机变换，其输出值自然与理想结果相去甚远，损失值也很大。随着模型训练循环的迭代次数（epoch）增加，权重值逐渐朝着正确的方向微调，并使得损失函数最小化。具有最小损失值的神经网络，其输出值与目标值尽可能地接近，即为训练效果最有的模型。其训练过程分为以下三步：在前向传播过程中，计算每一层的净输入和激活值，直到最后一层；在反向传播过程中，计算每一层的误差项；计算每一层参数的偏导数并更新参数。

基于反向传播的BP神经网络算法实现原理如下：

输入量：训练集*，*验证集，学习率，正则化系数，网络层数，神经元数量

输出量：神经网络的参数和偏置项

begin:

对训练集的样本随机重排序

for do

从中选取样本

前向传播计算每一层净输入和激活值，直至最后一层

反向传播计算每一层的误差

计算每一层参数的导数

更新参数

end

until 神经网络模型在中的误差值不再下降

输出

BP神经网络的第一个成功的实际应用来自于1989年的贝尔实验室，Yann LeCun将卷积神经网络的早期思想与反向传播算法相结合，并应用于手写数字分类问题，设计了名为LeNet的神经网络，在20世纪90年代被美国邮政署采用，用于自动读取信封上的邮政编码，取得了巨大的成功。

## 基于残差结构的神经网络

尽管传统的BP神经网络已经在大量分类任务上取得了较好的效果，然而深层的神经网络都存在梯度消失问题，即随着网络层数加深，梯度不断衰减直至消失，模型训练和优化变得十分困难。

He等人于2016年提出了一种名为残差网络（Residual Network, ResNet）的新型结构，并使用该网络赢得了ILSVRC 2015图像识别的挑战赛，其Top-5错误率仅为3.6%。ResNet这一简洁而高效的设计思想深刻影响了后续神经网络模型的发展和设计，残差网络通过给模型中间层加直连边（skip connection）的方式，提高了信息的传播效率。传统的多层神经网络结构与添加了残差网络的网络模块对比如图1.3所示。假设在深度神经网络中，期望一个非线性单元去逼近目标函数。将目标函数拆分为恒等函数和残差函数两部分，根据通用近似定理，一个由神经网络构成的非线性单元有足够的能力近似逼近原始目标函数或残差函数，但实际后者更容易学习，因此用非线性单元去逼近残差函数，并用去逼近目标函数。

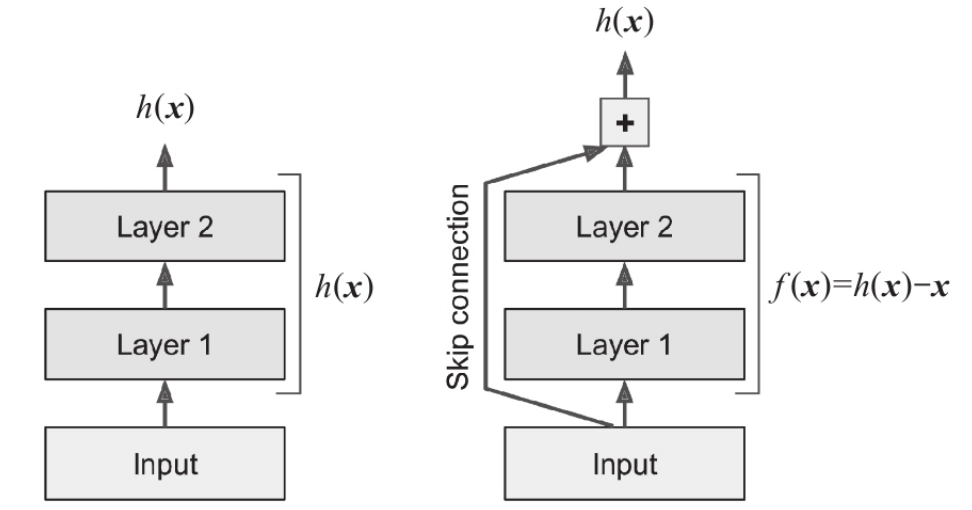


图1.3残差模块设计

残差块使得模型在训练时更易于捕捉恒等函数的细微波动，信号可以轻松地整个网络之间跨层传播。He提出的 ResNet50、ResNet101等模型便是由若干残差块堆叠完成，这一网络结构被广泛应用于深度学习领域，作为各种网络的主干特征提取层，取得了优异的效果。

# 2电容xxx数据集介绍

模型训练所需的数据来源于。。。。

数据集划分为训练集和测试集，其构成如表1.1所示。其中，训练集用于提供给模型训练，不断迭代并更新模型权重信息，测试集用于评估模型对新样本的分析能力，测验模型性能。

输入量共八项，分别为。。。。

输出量共包含十类，表示。。。输出量为有限个离散值，因此根据该数据集建立的学习算法为分类模型。

表1.1 数据集构成

|  |  |  |
| --- | --- | --- |
|  | 训练集 | 测试集 |
| 数据来源 |  |  |
| 数据量 |  | 20 |

# 3实验结果与分析

基于上述模型原理和数据集，采用VS code软件作为集成开发环境，学习算法设计和模型的训练环境如表1.2所示。

表1.2 模型训练环境

|  |  |
| --- | --- |
| 硬件设备 | 软件环境 |
| CPU： | Windows11 操作系统 |
| GPU：NVIDIA GeForce RTX3060 | 编程语言Python 3.8.18 |
| 显存： | 训练框架 torch 1.8.0  scikit-learn 1.3.2 |
| 内存：  存储： |

其中，scikit-learn机器学习库提供了大量机器学习模型和调用方法，有助于编写清晰统一的机器学习代码，并易于理解和复现实验结果；而深度学习模型的开发则采用主流的Pytorch框架进行模型的搭建、训练与测试。

在此，设计了七种不同类型和参数组合的模型：

1. KNN法。其中k值参数设置为5，表示数据点附近五个数据点中的多数类别作为预期输出；
2. 决策树法。模型参数使用基尼不纯度作为衡量模型性能的指标，从所有特征中选择最佳作为划分依据，不限制决策树的最大深度，设置划分所需最小的样本数量为2，叶子节点所需的最小数量为1，不设置随机种子，且不限制叶子节点数量。
3. BP-100神经网络。该网络基于BP神经网络结构，共三层，输入层共8个特征，输出节点个数为10，表明共分为10类，使用全连接层（Fully-Connected Layer, FC）作为中间隐藏层，其输出节点个数为100个，并在之后连接了ReLU激活函数，用于增强模型的非线性拟合效果，最大化发挥神经元的筛选能力。训练时采用梯度下降法作为优化策略，初始学习率为0.001，损失函数为分类交叉熵。
4. BP-200神经网络。该网络同样为三层结构，隐藏层的输出节点个数为200个，其他参数与（c）相同。
5. BP-100-5layers神经网络。该网络共包含五层结构，除了包含8个节点的输入层和10个节点的输出层外，在中间设置了三层全连接层，每层的输出节点个数均为100个，其他参数与（c）相同。
6. ResNet-128-128神经网络。该网络输入特征层共8个节点，输出层共10个节点，中间设计了一组残差块，包含两层全连接层，其输出节点个数均为128个，在各全连接层后均接入ReLU激活函数。最终，全连接层的输出部分与输入的第一个全连接层相连形成残差连接的结构，相加得到的结果汇总至输出层。训练时采用梯度下降法作为优化策略，初始学习率为0.001，损失函数为分类交叉熵。
7. ResNet-128-256神经网络。与（f）相比，模型的残差块中的第二全连接层的节点个数修改为256个，其他参数相同。

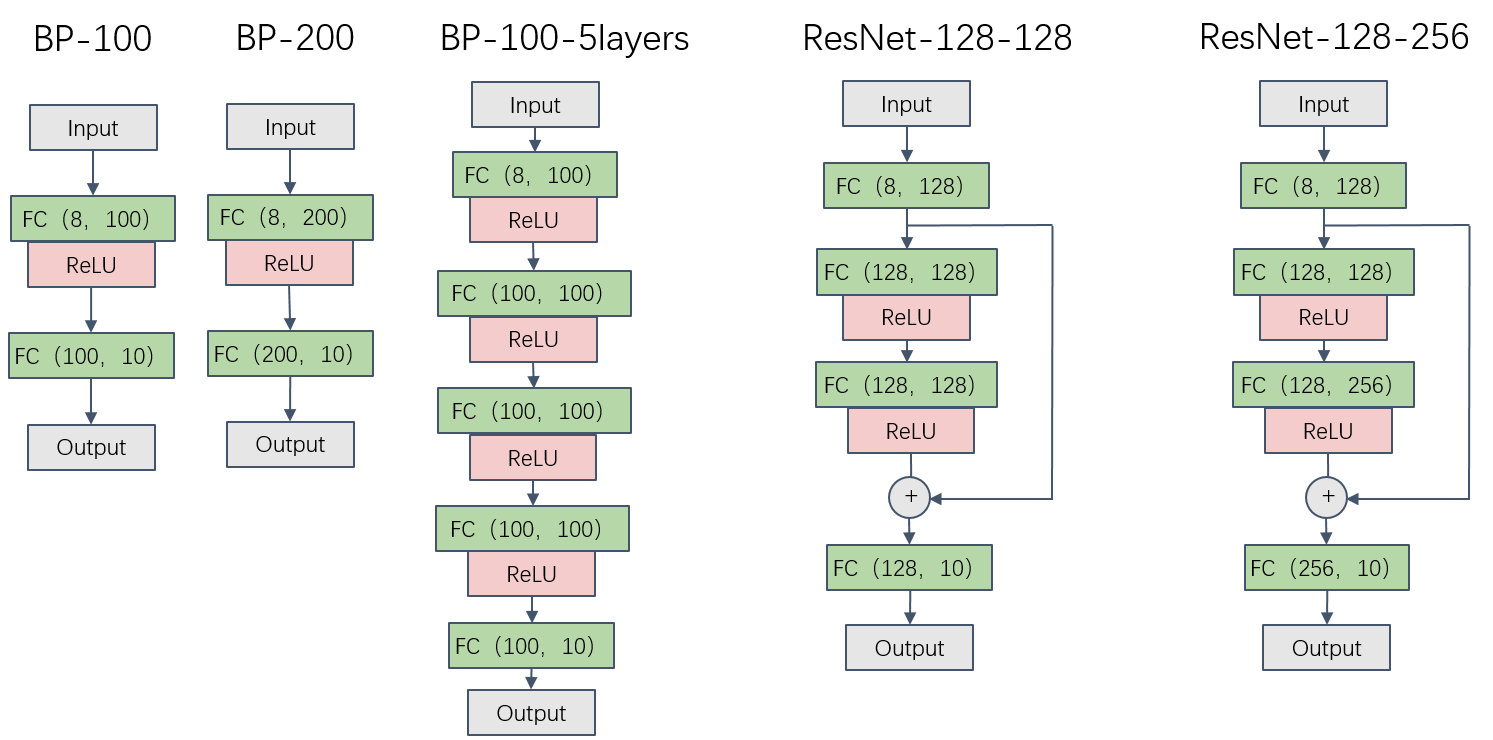


图1.4 模型结构图（括号内第一项数字表示该层的输入节点个数，

第二项表示该层输出节点个数）

模型（c）（d）（e）（f）（g）的结构如图1.4所示。模型取训练迭代1000次后误差值（用多分类交叉熵的方式计算得出）最小的保存为最优模型。模型训练的误差曲线如图1.4所示，横坐标表示迭代次数（epoch），纵坐标表示误差值（loss），随着迭代次数的增加，不同模型的参数均趋于收敛，反映了模型和数据集的可靠性；具有残差结构的ResNet网络相比BP神经网络的收敛速度更快，训练更为稳定，学习效果更好；而对于同一类型的模型，中间层节点个数增加，模型的训练和优化速度也能有所提升。

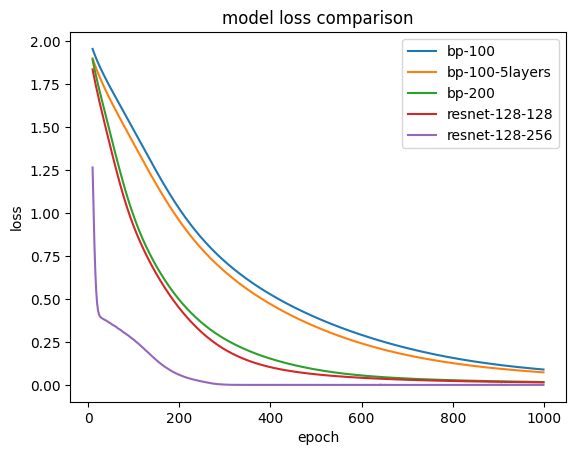


图1.4 模型训练的loss对比

在测试集中分析上述各模型的预测准确率，如表1.3所示，各模型预测结果对应的混淆矩阵如图1.5所示。部分预测结果与真实类别对比如表1.4所示。其中，序号a-g对应为上述各模型编号。

表1.3 模型预测结果对比

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 模型 | a | b | c | d | e | f | g |
| 准确率 | 65% | 75% | 75% | 80% | 75% | 85% | 90% |

|  |  |
| --- | --- |
|  | C:\Users\Rainy\AppData\Local\Microsoft\Windows\INetCache\Content.Word\bp100-75.00%.png |
| a | b |
|  | C:\Users\Rainy\AppData\Local\Microsoft\Windows\INetCache\Content.Word\80.00%.png |
| c | d |
| C:\Users\Rainy\AppData\Local\Microsoft\Windows\INetCache\Content.Word\决策树-75.00%.png |  |
| e | f |
|  |  |
| g |  |

图1.5 模型预测结果的混淆矩阵

表1.4 预测结果示例

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 输入值 | | | | | | | | 真实类别 | 模型预测结果 | | | |
| a | b | d | g |
| 1.01021 | 1.40091 | 0.86915 | 1.37571 | 1.07831 | 1.52874 | 1.14404 | 1.55035 | **ALO1** | **ALO1** | **ALO1** | **ALO1** | **ALO1** |
| 0.98385 | 1.3895 | 0.86635 | 1.3227 | 0.9724 | 1.66779 | 0.94614 | 1.33483 | **ALO2** | ALO1 | **ALO2** | ALO1 | **ALO2** |
| 0.99534 | 1.49055 | 0.84424 | 1.19134 | 0.42443 | 0.67503 | 0.71578 | 1.006 | **NBR** | **NBR** | ALO1 | **POM** | **NBR** |
| 0.428 | 0.48538 | 0.37603 | 0.4473 | 0.39856 | 0.4462 | 0.43259 | 0.48765 | **PMMA1** | PMMA2 | **PMMA1** | **PMMA1** | **PMMA1** |
| 3.03405 | 8.53989 | 2.6289 | 8.80827 | 3.03682 | 8.78481 | 3.01325 | 9.39864 | **C** | **C** | Cu | **C** | **C** |
| 4.86183 | 12.45943 | 3.13227 | 9.49289 | 4.2174 | 12.19278 | 5.76314 | 16.27556 | **Cu** | **Cu** | **Cu** | **Cu** | **Cu** |

由表1.3和图1.5可知：

KNN法的检测准确率最低，仅为65%，反映出该模型适用于简单的二类或三类的分类问题，在区分类别较多的高维度数据时，容易将近距离的同类样本混淆，分类效果较差；

决策树法与BP神经网络模型的分类效果相近，在75%至80%之间，且出现预测错误和混淆的情况大部分在第0类（ALO1）和第1类（ALO2）、第8类（C）和第9类（Cu）之间，产生这一结果的原因可能是这些类别。。。。；

模型（f）（g）的分类效果最佳，这是因为这两种模型引入了残差块结构，其侧向连接处使得模型在不产生额外参数且不增加计算复杂度的情况下，相当于简单地执行了恒等映射，防止在反向传播计算导数值时出现梯度消失现象，提升了对输入数据深层和浅层特征的提取效果。

最终结果表明，ResNet-128-256神经网络是最优模型，在测试集上达到了90%的准确率，考虑到该模型为十分类问题，已经属于相当不错的表现，初步表明基于深度学习方法实现电容xxx模型预测具有可行性。