> 逐字稿学习法，一定要自己经常复述

- 《跟着大壮一起机器学习【亮】》
- 《深度学习500问》
- 《百面机器学习》
- 《百面深度学习》
- 《 轻松学算法 互联网算法面试宝典 ,赵烨》
- 面试宝典——自总：算法、数据结构、数据库、python 所有算法可能得面试问题 一天整理50个 很快就能结束 可问题是 天天杂事很多 算咯
- Transformer常见问题
- 概率算法、李航
- 原理、code、数学表达式 理论结合实践

时间线

1. 2024年10月20日开始 第一本《跟着大壮一起机器学习【亮】》 集成学习

九 集成学习

问题目录：

1. 什么是集成学习，以及它的基本思想是什么?
2. 如何理解集成学习中 “多样性”的概念，为什么多样性对集成模型的性能至关重要?

3. 集成学习中bagging和boosting有什么区别？举例说明。
4. 集成学习中常见基本算法有哪些?
5. 什么是随机森林？它是如何工作的，以及如何处理过拟合问题?
6. adaboost算法如何改进弱分类器性能呢？基本原理是什么?
7. 集成学习中，如何选择合适的基础学习器 或 弱分类器?
8. 集成学习模型的预测是如何组合多个基础学习器的输出的？请描述一些常见的组合策略。
9. 什么是袋外误差（out of bag error）和交叉验证（cross validation）在集成学习中的作用?
0. 集成学习在哪些领域和任务中表现出色？它有什么局限性?

1 什么是集成学习，以及它的基本思想是什么?

集成学习的基本思想是组合多个学习算法或模型的预测来提高整体性能和泛化能力

汇总多个模型的意见，减小单个模型的方差和偏差，提高模型的鲁棒性和准确性

背后的基本思想包括：

1 组合多个模型：集成学习 同时使用 多个不同的学习算法 或者 同一个算法的不同设置 创建多个基本模型 这些模型被称为 弱学习器

2 多样性和独立性：为了保证集成模型的多样性 基本模型是在不同的子样本或特征子集上训练的 来捕捉不同方面的模式，多样性和独立性是集成学习成功的关键因素

3 投票或加权平均：一旦生成了这些基本模型，集成学习使用不同的策略来组合它们的预测结果。对于分类问题，可以采用多数投票来决定最终的分类结果；对于回归问题，采用加权平均来获得最终的预测值

4 性能提升：通过组合多个模型，集成学习旨在提高整体性能，减少模型过拟合风险，以便在新数据上更好地泛化

实际案例 说明集成学习的应用

下面以股票价格预测 说明集成学习的应用：假设你是股票投资者，使用集成学习方法提高预测的准确性

1. 数据收集：收集大量与股票有关的数据，包括历史股价、成交量、市场指数、公司财务数据
2. 基本模型选择：选择几种不同的时间序列预测模型，如ARIMA、Prophet、神经网络 作为基本模型
3. 数据随机化：为确保基本模型的多样化，对数据进行随机化和分割，以确保每个模型使用不同的训练数据子集
4. 模型训练：每个基本模型使用其分配的数据子集进行训练，以学习如何预测股票价格。
5. 预测集成：一旦基本模型训练完成，将来自所有模型的结果进行集成。可以采用加权平均的方法，其中更有经验的模型可能被赋予更大的权重
6. 性能评估：使用历史数据的验证集来评估集成模型的性能，以确定其对未来股票走势的预测准确性

通过这种方式，利用了不同模型的多样性和独立性，提高了股票价格预测的准确性。

[code 补充]

2 如何理解集成学习中“多样性”的概念，为什么多样性对集成模型的性能至关重要?

多样性指的是：集成学习的基本模型，也就是弱学习器之间的差异和多样性。多样性对集成模型的性能至关重要，好处：

1. 降低过拟合的风险：多样性有助于减小集成模型的方差，意味着模型对训练数据不会过于敏感，减少过拟合的风险。如果所有基本模型都过于相似，那它们可能在训练数据上都表现得很好，但在新数据上的泛化性能可能会很差。
2. 增加鲁棒性：即使其中一些基本模型出现错误或者不适用于特定数据分布，其他模型仍可以提供有用的信息。鲁棒性对于 噪声数据 或 异常情况非常重要
3. 提高整体性能：多样性有助于集成模型捕捉数据中不同方面的模式和信息。如果基本模型之间存在多样性，集成模型能够更全面地考虑不同模型的预测结果，从而提高整体性能
4. 减小估计偏差：集成模型的一个关键思想 减小估计偏差，多样性 可以增加 模型之间的差异，从而减小集成模型的 偏差。意味着 集成模型 更可以适应数据的 真实模式，不受单个模型的限制

实现多样性的方法：

1. 使用不同的基本模型：选择 不同类型 的机器学习算法 或者模型，确保学习方式不同
2. 数据随机化：对训练数据进行随机子采样或 引入噪声，确保每个基本模型的数据略有不同
3. 特征选择：对特征进行不同的选择或排除，使基本模型使用的特征集不同

4. 参数调整：对每个基本模型使用不同的超参数设置，使得学习的模型不同

集成学习关键词：多样性和独立性提高集成模型的性能和鲁棒性，使其能够更好地应对不同类型的数据和问题；减少过拟合，提高泛化能力。角度：模型种类、模型参数、数据量、数据特征

3 集成学习中的Bagging和Boosting有什么区别？举例说明

Bagging：Bootstrap Aggregating

Bagging和Boosting是两种常见的集成学习技术

Bagging（bootstrap aggregatting）

工作原理：Bagging通过随机有放回地从训练数据集中抽取多个子样本（Bootstrap样本），然后使用这些子样本来训练多个基本模型，每个基本模型都是在不同的训练子集上训练

模型训练：每个基本模型独立的学习训练数据的不同子集，因此它们之间相互独立，可以并行训练。

集成策略：bagging的集成策略是通过平均或者投票来集成基本模型的预测结果，获得最终的集成模型

Boosting

工作原理：Boosting通过迭代地训练一系列基本模型，每个基本模型都试图修正前一个模型的错误，给每一个样本分配一个权重，使前一个模型错误分类的样本在下一轮训练中获得更高的权重

（迭代训练，前一个模型错误分类的样本下一轮训练中获得更高的权重）

模型训练：Boosting的基本模型是依次训练的，每个模型都在前一个模型的基础上学习，尝试减小前一个模型的错误

（依次训练）

集成策略：Boosting的集成策略是通过加权平均基本模型的预测结果来获得最终的集成模型。权重通常是根据模型的性能分配的，性能好的模型通常有更高的权重

（性能好的模型 权重高；分类错误的样本权重高）

举例说明区别：

假设有一个二分类问题，我们正在使用决策树作用基本模型，进行Bagging和Boosting的对比：

bagging示例：

1. Bagging首先随机抽取多个Bootstrap样本，每个样本都包含一部分训练数据
2. 针对每个bootstrap样本，独立地训练一个决策树模型
3. 最后，Bagging将所有决策树的预测结果进行投票，以决定最终的分类结果

关键词：自助采样样本、投票

boosting例子：

1. boosting首先使用原始训练数据训练一个决策树模型
2. 根据第一个模型的错误分类情况，调整训练模型中的样本权重，将更多关注被错误分类的样本（错误分类的样本权重高）
3. 接下来，使用调整后的数据训练第二个决策树模型，修正第一个模型的错误

4. 迭代这个过程，依次训练更多的模型，每个模型都试图修正前一个模型的错误（迭代训练，修正上一个的错误）
5. 最后，Boosting将所有模型的预测结果进行加权平均，得到最终的分类结果。（加权平均）

记忆点：性能好的模型权重高；分类错误的样本权重高；每次使用所有模型；加权平均输出最终结果

bagging：并行训练多个独立地基本模型；预测结果平均化，减小方差；

boosting：迭代训练一系列基本模型，每个模型都试图修正前一个模型的错误，提高整体性能；

==》boosting 性能更强大，且更容易 过拟合

4 集成学习的常见算法有哪些？随机森林，Adaboost，gradient boosting

集成学习中 有很多常见的基本算法，每个算法都有特定的方式组合多个模型或学习器的预测结果

1. bagging（bootstrap aggregating）：对训练数据进行有放回的随机抽取创建多个子样本，使用这些子样本训练多个基本模型。最终的预测结果通过对这些模型的预测结果进行平均（回归问题）或投票（分类问题）得到。随机森林是Bagging的一个典型应用，使用决策树作为基本模型。

   bagging 自主采样样本 训练、投票（分类），平均（回归）、并行训练

   1. random forest：随机森林是一种基于bagging的集成学习算法，使用多个决策树作为基本模型，并且在每个决策树的训练过程中 引入了 随机性。随机森林的特点是能够降低过拟合风险，同时保持较高的预测性能。

2. adaboost || adaptive boosting:Adaboost 是一种boosting算法，通过迭代的训练多个弱学习器，每个学习器纠正签一个学习器的错误。每轮迭代中，adaboost会增加错误分类样本的权重，使得下一个学习器更加关注这些样本。最终通过对弱学习器进行加权组合，得到一个强学习器。（迭代训练，纠正前一个学习器的错误，错误分类的样本 更多的关注；弱学习器的加权组合）
3. gradient boosting：gradient boosting也是一种boosting算法，通过梯度下降的方法迭代训练每个基本模型。每一轮迭代中，gradient boosting试图拟合上一轮的残差（误差的负梯度），从而逐渐改进模型的性能。XGBoost、LightGBM、CatBoost是gradient boosting的改进版本。
4. stacking：stacking是一种元集成学习方法，不仅使用多个基本模型，还包括一个元模型，用于组合基本模型的预测结果。基本模型的预测结果作为原模型的输入，元模型通过学习如何组合这些结果来产生最终的预测，stacking通常需要更复杂的模型组织和训练过程（基本模型的输出 作为元模型的输入，元模型组合基本模型的输出预测结果）
5. voting：简单且有效的集成策略，用于分类问题。可以基于硬投票（基于多数票）或软投票（基于预测的概率值），将多个基本模型的预测结果结合起来，选择最终的标签。（三个臭皮匠 顶个诸葛亮）

5 什么是随机森林（random forest），如何工作的，以及如何处理过拟合的问题

随机森林是一种集成学习方法，用于解决分类和回归问题，由多个决策树组成，每个决策树都是独立训练的，通过投票或平均预测结果来进行最终的预测。（用自助采样样本训练 bagging）

工作原理：

1. 数据的随机抽样：从原始数据集中随机选择一个子集（有放回地抽样），称为自主采样（bootstrap sampling），目的 生成 多个不同的训练数据子集（采样的是 特征子集？）
2. 构建决策树：对于每个子集，使用决策树算法构建一个独立的决策树模型。决策树递归的将数据集分割为更小的子集，并且根据特征的值进行判断。特征的选择过程 在每个节点是随机的
3. 集成预测：当要进行预测时，将测试样本输入到每个决策树中，得到各自的预测结果。最后根据投票（分类问题）或者平均（回归问题）的方式输出最终的预测结果

（在自主采样样本上 独立的训练 然后测试样本 输入到每颗决策树 通过投票 或者 平均 得到最终的预测结果）

随机森林处理过拟合问题，具体方法：

1. 自主采样（bootstrap sampling）：通过自主采样，随机森林生成多个不同的训练数据子集。重复抽样的过程减少模型对特定数据的敏感性，减少过拟合的风险
2. 随机特征选择：每个决策树节点划分时，随机森林只考虑一个随机选择的特征子集。有助于减少特征之间的相关性，避免某些特征对结果的过度依赖
3. 多个模型的平均：随机森林采用多个决策树进行集成预测。对每个决策树的预测结果进行投票或平均，减少单个决策树带来的噪声和错误，提高整体的泛化能力

这些技术共同作用使得随机森林具有更好的鲁棒性和泛化能力，有效处理过拟合问题。

随机森林的特点：泛化性、鲁棒性，有效地处理过拟合问题

6

十、神经网络

问题目录

1．什么是神经网络（neural network）？它的基本原理是什么?

1 什么是神经网络（neural network）？它的基本原理是什么?

神经网络是一种机器学习算法，模仿人脑的神经系统结构和工作原理。由多个称为神经元的节点（或单元）组成，这些节点通过连接（权重）相互传递信息，并输入数据上执行复杂的非线性计算。

基本原理

1．输入层（input layer）接收外部输入数据，并传递给下一层

2. 隐藏层（hidden layer）位于输入层和输出层之间的一系列层，负责处理和转换输入数据
3. 输出层（output layer）产生最终结果或预测
4. 每个连接都有一个相关的权重，用于调整输入值的重要性
5. 神经元应用激活函数（activation Function）处理加权总和，并产生输出

对于一个简单的神经网络，用公式表示：

输入层到隐藏层的计算：

$$h_i = \sigma(\sum_{j=1}^{n} \omega_{ij} x_j + b_i)$$

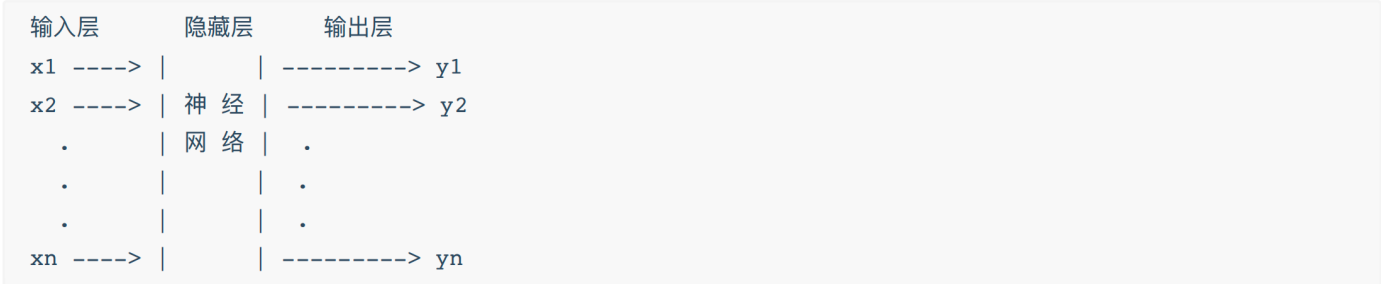
- h_i 隐藏层中 第i个神经元的输出
- σ 激活函数
- $\sum_{j=1}^{n} \omega_{ij} x_j$ 输入层到隐藏层的加权总和
- ω_{ij} 连接输入层第j个神经元与隐藏层第i个神经元的权重
- x_j 输入层第j个神经元的输出
- b_i 偏置(bias)，用于调整 加权总和的阈值

隐藏层到输出层的计算

$$o_k = \sigma(\sum_{i=1}^{m} \omega_{ki} h_i + b_k)$$

- o_k 输出层中第k个神经元的输出
- $\sum_{i=1}^{m} \omega_{ki} h_i$ 隐藏层到输出层的加权总和
- ω_{ki} 连接隐藏层第i个神经元与输出层第k个神经元的权重
- b_k 输出层的偏置

图示：



实际应用中，神经网络可以有多个隐藏层，每个隐藏层可以有不同数量的神经元，这种多层结构称为深度神经网络（deep neural network）

2 描述前馈神经网络的结构和工作原理

神经网络是一种计算模型，受到人脑的神经系统的启发而设计。由多个连接的处理单元组成，这些神经元之间通过权重来传递和处理信息

基本原理：

1. 神经元：每个神经元接收输入信号，并根据权重对输入信号进行加权求和，通过激活函数将结果转换为输出信号
2. 权重：每个输入信号与对应的权重相乘，权重决定了输入对神经元输出的影响程度
3. 激活函数：激活函数对加权和进行非线性变换，使得神经网络能够更好地拟合复杂的数据模型，常见的激活函数：sigmoid、relu、tanh等

神经网络公式：

输入层到隐藏层：(明确固定的数字是隐藏层)

$$a_1 = \sum_{i=1}^{n}(\omega_{1i}x_i) + b_1$$

$$a_2 = \sum_{i=1}^{n}(\omega_{2i}x_i) + b_2$$

$$h_1 = f(a_1)$$

$$h_2 = f(a_2)$$

其中，x_i 表示 输入值 ω 表示权重，b 表示偏差（偏置项），h_i 表示隐藏层的输出，y 表示神经网络的最终输出，$f(\cdot)$ 表示激活函数

这个公式描述了一个具有一个隐藏层和一个输出层的简单前馈神经网络。通过调整权重和偏差，神经网络可以学习输入和输出之间的复杂映射关系，从而解决可以解决，如分类、回归、图像识别和自然语言处理等任务。

3 神经网络中的激活函数 activation function 有什么作用？常见的激活函数有哪些?

激活函数（activation function）在神经网络中 起到 非线性变换的作用，对神经元的加权和进行非线性映射，使神经网络能够更好地拟合非线性数据和复杂模式。激活函数引入非线性性质，增加神经网络的表达能力，并且通过梯度传播算法（如反向传播）可以有效地进行训练

①sigmoid②relu③leaky relu④tanh双曲正切函数

sigmoid： $f(x) = \frac{1}{1+e^{-x}}$

sigmoid函数将输入映射到0到1之间，具有平滑的S型曲线，将任意实数映射到一个概率值，适合二分类问题和输出层的概率估计

2、**ReLU函数（Rectified Linear Unit）：**

$$f(x) = \max(0, x)$$

ReLU函数在$x > 0$时返回x，在$x \leq 0$时返回0。ReLU函数简单、易于计算，并且在处理大规模神经网络时效果良好。它可以缓解梯度消失问题，但可能会导致一些神经元“死亡”（输出始终为0）问题。

3、**Leaky ReLU函数：**

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{if } x \leq 0 \end{cases}$$

Leaky ReLU函数是ReLU函数的改进版本，当$x \leq 0$时引入一个小的斜率α，以解决ReLU函数中神经元“死亡”问题。

4 深度学习和深度神经网络之间的关系

5 前向传播 反向传播

反向传播计算损失关于权重和偏置的梯度、计算梯度，更新参数；

前向传播计算预测结果；反向传播 计算梯度 更新参数；最小化损失函数

6 神经网络的优化方法是什么？有哪些常见的正则化技术和防止过拟合的策略?

QKV的理解

你有一个问题Q，然后去搜索引擎里面搜，搜索引擎里面有好多文章，每个文章V有一个能代表其正文内容的标题K，然后搜索引擎用你的问题Q和那些文章V的标题K进行一个匹配，看看相关度（QK --->attention值），然后你想用这些检索到的不同相关度的文章V来表示你的问题，就用这些相关度将检索的文章V做一个加权和，那么你就得到了一个新的Q'，这个Q'融合了相关性强的文章V更多信息，而融合了相关性弱的文章V较少的信息。这就是注意力机制，注意力度不同，重点关注（权值大）与你想要的东西相关性强的部分，稍微关注（权值小）相关性弱的部分。

作者：李图图
链接：https://www.zhihu.com/question/427629601/answer/1558216827