**深度机器 输出**

Author: 智能2111 沈超凡

Description: 修复一些已知问题

Date: 2023-02-11 00:51:29

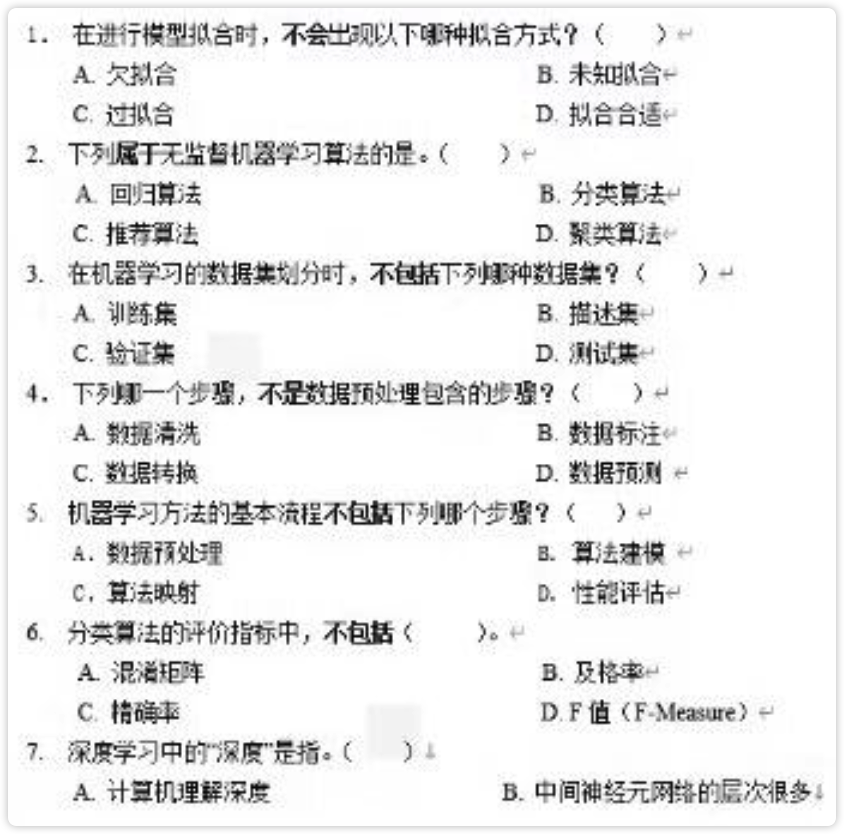
LastEditors: jarrycf 1354602179@qq.com

LastEditTime: 2023-02-20 21:45:07

**单选 15\*2**

BDBDC（考到）

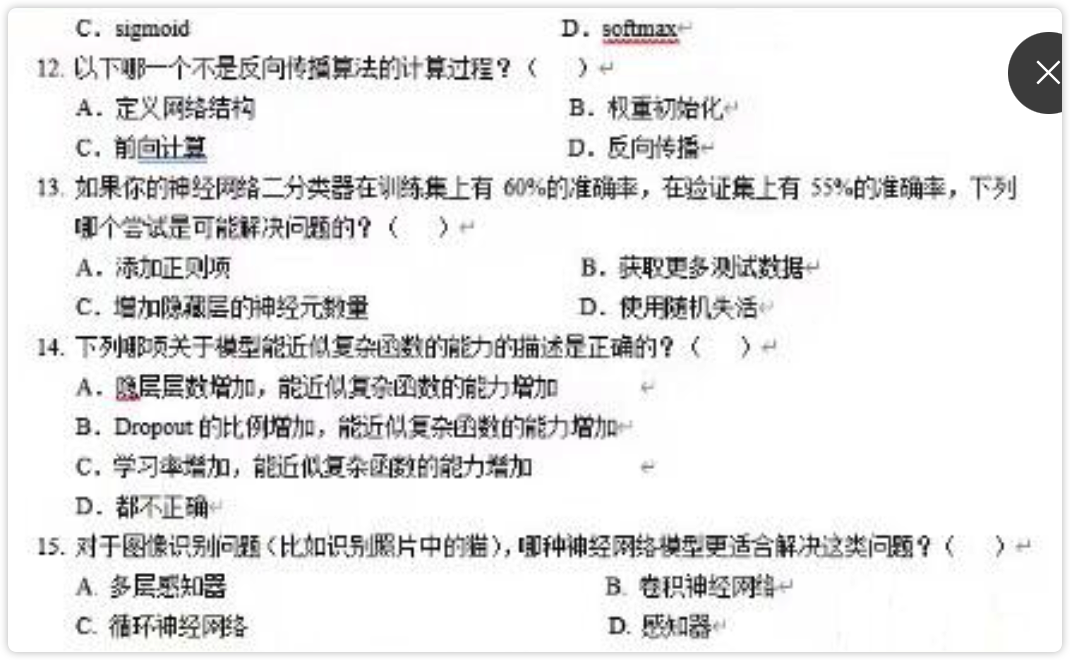
BC



C. 计算机建立模型的深度 D. 数据输入层与输出层之间的层数多

CABDB（考到）

11.激活函数用于二分类任务的是



**选择2（可多选）（全没考到）**

BBDC(ABCD)

1、深度学习中的“深度”是指

A. 计算机理解深度

B. 中间神经元网络的层次很多

C. 计算机的求解更加精确

D. 计算机对问题的处理更加灵活

2、下列哪一项在神经网络中引入了非线性

A. 随机梯度下降

B. 修正线性单元(ReLU)

C. 卷积函数

D. 以上都不对

3、下列哪个神经网络结构会发生权重共享

A. 卷积神经网络

B. 循环神经网络

C. 全连接神经网络

D. 选项A和B

4、关于句子向量表示，下列说法正确的是

A. 只能通过有监督学习获得

B. 只能通过无监督学习获得

C. 有监督和无监督学习都可以获得

D. 以上都不对

5、在神经网络中，下列哪种技术用于解决过拟合

A. Dropout

B. 正则化

C. early stop

D. Batch Normalizaiton

ABDAB

6、以下哪种不是自适应学习率方法

A. Mini-batch SGD

B. Adagrad

C. RMSprop

7、哪种策略可以加速词向量训练

A. para2vect

B. 层级softmax

C. 最大似然估计

D. 以上都不对

8、关于梯度下降算法，以下说法正确的是

A. 随机梯度下降算法是每次考虑单个样本进行权重更新

B. Mini-Batch梯度下降算法是批量梯度下降和随机梯度下降的折中

C. 批量梯度下降算法是每次考虑整个训练集进行权重更新

D. 以上都对

9、与传统机器学习方法相比，深度学习的优势在于

A. 深度学习可以自动学习特征

B. 深度学习完全不需要做数据预处理

C. 深度学习完全不提取底层特征，如图像边缘、纹理等

D. 深度学习不需要调参

10、下列哪一项在神经网络中引入了非线性

A. 随机梯度下降

B. Sigmoid激活函数

C. 增大权重和偏置的初始化值

D. 以上都不对

D(BD)ACC #svm高斯核函数比线性核函数模型更复杂，容易过拟合

11、在其他条件不变的前提下，以下哪些做法容易引起机器学习中的过拟合问题

A. 增加训练集量

B. 减少神经网络隐藏节点数

C. 在损失函数中增加正则项

D. SVM算法中使用高斯核/RBF核代替核性核

12、下列模型，属于判别式模型的有

A. 隐马尔可夫模型

B. 支持向量机

C. 朴素贝叶斯

D. 神经网络

13、关于集成学习，下列说法错误的是

A. 集成学习一定能提升个体学习器的性能

B. Bagging方法中，个体学习器之间彼此独立

C. Boosting是一种重视错误样本的学习方法

D. Boosting方法中，个体学习器存在强依赖

14、如果使用的学习率太大，会导致

A. 网络收敛的快

B. 网络收敛的快

C. 网络无法收敛

D. 不确定

15、在一个神经网络里，知道每一个神经元的权值和偏差值是最重要的一步。如果以某种方法知道了神经元准确的权重和偏差，就可以近似任何函数。实现这个最佳的办法是

A. 随机赋值，祈祷他们是正确的

B. 搜索所有权重和偏差的组合，直到得到最佳值

C. 赋予一个初始值，通过检查跟最佳值的差值，然后迭代更新权重

D. 以上说法都不正确

CC(ABC)C(ABCD)

16、下列目标检测网络中，哪个是一阶段的网络

A. Faster-rcnn

B. RFCN

C. YOLOV3

D. SPP-net

17、SSD主要通过哪种方法来解决检测不同大小目标的问题

A. 设置更多的anchor尺寸

B. 设置更多的anchor纵横化

C. 在不同的特征图上进行预测

D. 使用图像金字塔作为输入

18、当图像分类的准确率不高时，可以考虑以下哪种方法提高准确率

A. 数据增强

B. 调整超参数

C. 使用预训练网络参数

D. 减少数据集

19、文字监测网络TextBoxes基于下列哪个网络

A. Fast-rcnn

B. Faster r-cnn

C. SSD

D. Y0L0

20、下面哪些情况可能导致神经网络训练失败

A. 梯度消失

B. 梯度爆炸

C. 激活单元死亡

D. 鞍点

(AC)BCCB

21、假定你现在在解决一个有着非常不平衡的分类问题，即主要类别占据了训练数据的99%。现在你的模型在测试集上表现为99%的准确率。下面表述正确的是

A. 准确度并不适合于衡量不平衡类别问题

B. 准确度适合于衡量不平衡类别问题

C. 精确率和召回率适合于衡量不平衡类别问题

D. 精确率和召回率不适合于衡量不平衡类别问题

22、假定在神经网络中的隐藏层中使用激活函数X。在特定神经元给定任意输入，会得到输出[-0.0001]。X可能是以下哪一个激活函数

A. ReLU

B. tanh

C. Sigmoid

D. 以上都不是

23、如果增加神经网络的宽度，精确度会增加到一个阈值，然后开始降低。造成这一现象的原因可能是

A. 只有一部分核被用于预测

B. 当核数量增加，神经网络的预测能力降低

C. 当核数量增加，其相关性增加，导致过拟合

D. 以上都不对

24、假设只有少量数据来解决某个具体问题，但有有个预先训练好的神经网络来解决类似问题。可以用下面哪些方法来利用这个预先训练好的网络

A. 把除了最后一层外所有的层都冻结，重新训练最一层

B. 重新训练整个模型

C. 只对最后几层进行微调

D. 对每一层模型进行评估，只使用少数层

25、下列哪个不属于CRF模型对于HMM模型的优势

A. 特征灵活

B. 速度快

C. 可以容纳较多的上下文信息

D. 能够做到全局最优

(ABCD)BAC

26、下列方法中，可以用于特征降维的方法包括

A. 主成分分析

B. 线性判别分析

C. 自编码器

D. 矩阵奇异值分解

27、以下说法中正确的是

A. SVM对噪声鲁棒

B. 在AdaBoost算法中，所有被分错的样本的权重更新比列相同

C. Boosting和Bagging都是组合多个分类器投票的方法，二者都是根据单个分类器的正确率决定其权重

D. 分类器集成一定能提升单分类器的性能

28、关于集成学习，以下说法错误的是

A. 集成学习一定能提升个体学习器的性能

B. Bagging方法中，个体学习器之间彼此独立

C. Boosting是一种重视错误样本的学习方法

D. Boosting方法中，个体学习器存在强依赖

29、比较基于深度学习的在线更新跟踪算法与在线不更新的跟踪算法，下列说法错误的是

A. 一般来说，在线不更新的跟踪算法速度更快

B. 一般来说，在线更新的跟踪算法，可以适应目标的变化，和背景信息的变化，对特征的要求较低

C. 一般来说，在线更新的跟踪算法，在发生丢失和遮挡时，很容易找回

D. 一般来说，在线不更新的跟踪算法，对特征的要求比较高，要求特征的表示能力鲁棒性更强

**选择3**

逻辑回归是什么算法？（考到）

分类算法

注：

基本思想是利用回归分析的方法，将自变量和因变量之间的关系建立为一个逻辑函数，从而将连续的输出值转化为离散的分类值

深度学习框架有哪四个？（考到）

MXnet

Pytorch

Tensorflow

PaddlePaddle

DBscan 算法应用的场合？（考到的k-mean是应用场景）

数据集中有噪声或异常点

数据集中的簇数量不确定

数据集中的簇形状不规则或复杂

数据集的维度不太高

支持向量机有哪几种核函数？（考到代码线性）

线性

多项式

高斯

拉普拉斯（Laplace kernel function）

sigmoid核函数

CNN，FCN，GAN，可以用来干什么？（考到，考后总结）

CNN：图片识别 图像分类

FCN：图像语义分割

GAN：图像生成 图像修复

注：

图像分割是指将一张图像分成若干个互不重叠的区域，每个区域代表一个语义信息。而图像语义分割是指将图像分割成具有语义的区域，即将图像中的每个像素分配到其对应的语义类别中，通常是通过像素级别的分类实现的。因此，可以说图像语义分割是图像分割的一种特殊形式，强调了对像素级别的语义理解和分类。

上采样的方法有哪些（不考）？

双线性差值

反卷积

反池化

转置卷积

注：

上采样（upsampling）：将低分辨率图像恢复为高分辨率图像

激活函数 softmax、sigmoid、tanh、rule 分别用于什么任务，值域是多少（考到图像）

softmax 多分类 (0, 1)

sigmoid 二分类 (0, 1)

tanh 回归 (-1, 1)

rule 非线性任务如识别 [0, +♾️)

注：

1. Softmax 函数：适用于多分类任务，将神经网络的输出映射到一个概率分布上，值域为 (0, 1)。
2. Sigmoid 函数：适用于二分类任务和概率估计问题，将神经网络的输出映射到一个概率上，值域为 (0, 1)。
3. Tanh 函数：适用于回归任务，将神经网络的输出映射到 (-1, 1) 的范围内，可以解决输入数据存在负值的问题。
4. ReLU 函数：适用于处理非线性关系的任务，通常用于图像识别和语音识别任务，值域为 [0, ∞)。

值得注意的是，ReLU 函数在输入小于等于 0 的时候输出为 0，因此有一些改进的 ReLU 函数被提出来，如 LeakyReLU 和 ELU，可以缓解 ReLU 函数的一些问题。

**填空 15\*2**

1. 在机器学习中，完整的机器学习流程通常包括：\_、\_、\_、\_四个流程（考到）
2. 深度学习的三个关键步骤\_、\_、\_
3. 卷积层的输出通常由卷积核的长、宽、\_、\_、\_、決定
4. 随机梯度下降（stochastic gradient descent, SGD）未来减少每次迭代的计算开销，通常在每次迭代中，随机均匀采样\_个样本计算梯度，更新一次参数（考到，bitch—size能否改变这里的1）
5. 假设有一个 63×63×16 的输入，并使用大小为 5 x 5 的 32 个过滤器进行卷积，步幅为3，无填充，见输出形状是\_，再经过一个 2x2，步幅为 2 的最大池化后的输出的形状是\_。（考到）
6. 简单 RNN 模型存在\_问题，所以无法处理很长的输入序列，\_模型可以较好的解决这个问题

数据清洗 特征工程 模型训练 模型评估

设计网络结构和参数 定义损失函数和优化方法 使用训练集和测试集调整模型性能

输入通道数、输出通道数（卷积核个数）、填充和步长

(63+2\*0-5)/3 + 1 = 20 \* 20 \* 32

(20 - 2) / 2 + 1 = 10 \* 10 \* 32

梯度消失（短期记忆） LSTM

**填空2**

随机森林的基础算法是什么？

决策树

交叉验证有哪几种方法？

Hold-Out Cross Validation

K-Fold Cross Validation

Leave-One-Out Cross Validation

Bootstrap Cross Validation

注：  
交叉验证是机器学习模型评估的重要方法之一，它通过将数据集划分成若干份来评估模型的泛化能力。常见的交叉验证方法包括：

简单交叉验证（或留出交叉验证 Hold-Out Cross Validation）：通常将数据集的70%~80%作为训练集

K折交叉验证（K-Fold Cross Validation）：将数据集平均分成K份，每次选取其中的一份作为验证集，其余的K-1份作为训练集，进行K次交叉验证，最终得到K个验证结果的平均值。

留一交叉验证（Leave-One-Out Cross Validation）：将数据集中的每个样本都作为验证集，其余的样本作为训练集，进行N次交叉验证，N为数据集的大小。

自助法交叉验证（Bootstrap Cross Validation）：从数据集中有放回地抽样，得到一个新的训练集，将剩余的样本作为测试集，进行多次交叉验证，最终得到平均值。

分层交叉验证（Stratified Cross Validation）：在K折交叉验证的基础上，确保每一份中都包含不同类别的样本，以避免数据集划分不平衡的问题。

决策树算法中生成决策树的三种算法？（考到）

ID3算法：使用信息增益作为特征选择的标准，以熵为度量构建多叉树

C4.5算法：使用信息增益率作为特征选择的标准，以熵为度量构建多叉树，并进行剪枝处理

CART算法：使用基尼（GINI）系数作为特征选择的标准，以基尼不纯度（Gini Impurity）构建二叉树，并进行剪枝处理

KNN算法的名称？

KNN(k-nearest neighbor classification) 最近k邻居分类

贝叶斯算法的核心思想？

通过考虑逆向概率（先验概率和后验概率）来预测分类

贝叶斯定理

|  |
| --- |
| Python p(A|B) = (p(B|A) \* p(A)) / p(B) # 后验概率 = （似然度 \* 先验概率）/ 证据 |

通过先前的经验给出一个先验分布P(θ)，没有经验则随意指定。然后观察数据D，不断修正P(θ)，最终得到后验分布P(θ|D)，这就是逆向概率。然后选择具有最大后验概率的类别作为分类结果。

层次聚类的两种类别？

自下而上 （凝聚型层次聚类 agglomerative）

自上而下 （裂型层次聚类 divisive）

a点的坐标是（1,2,6），b点的左边是（2,3,4），求它的欧式距离和曼哈顿距离？

sqrt(1 + 1 + 2\*\*2) = sqrt(6)

4

Adam（自适应矩估计优化算法 Adaptive Moment Estimation）是基于哪两种方法的优化算法？

自适应梯度（AdaGrad）

均方根传递（RMSprop）

20个数据 batch\_size = 4 需要几次更新（epoch）？

5

残差网络是什么，用来解决什么问题？（考到简答）

通过跳层的方式将高层的梯度直接传递到低层

用于解决深度神经网络训练时，随着网络层数的增加而产生的梯度消失（网络退化问题）问题，使模型可以更深，也可以解决梯度爆炸的问题。

**填空3**

神经网络的网络结构是由哪四部分组成的，卷积神经网络由哪几部分组成？（考到填空）

输入层 隐藏层 输出层 连接权重

卷积层 池化层 全连接层 激活函数

机器学习的基本流程？

数据采集

数据预处理

特征工程

模型选择

模型训练和预测

模型评估

模型优化

模型应用

注

1. 数据采集：选择合适的数据源，收集数据样本，确定特征值和目标值
2. 数据预处理：清洗、转换、规范化、缺失值处理等，提高数据质量
3. 特征工程：选择、构造、降维等，提取有效的特征，减少冗余和噪声
4. 模型选择：根据问题类型和数据特点，选择合适的机器学习算法，如监督学习、非监督学习或强化学习等
5. 模型训练和预测：利用训练集对模型进行参数优化，利用测试集或新数据对模型进行预测或分
6. 模型评估：使用各种指标和方法，如准确率、召回率、F1分数、ROC曲线等，评估模型的性能和泛化能力
7. 模型优化：根据评估结果，对模型进行调参、优化等处理，以提高模型的性能
8. 模型应用：将模型应用于实际问题中

具体实现方式和步骤可能因具体的应用场景而有所不同

常见的梯度下降方法有哪些？

BGD 每次更新参数时使用所有的数据样本计算梯度 稳定但慢

SGD 随机选一个 快但不稳定

MBGD 小部分数据 稳定兼速度但要调整

SAG SGD的改进 更快收敛

注

BGD：批量梯度下降：每次更新参数时使用所有的数据样本计算梯度，优点是稳定，缺点是速度慢。

SGD：随机梯度下降：每次更新参数时随机选择一个数据样本计算梯度，优点是速度快，缺点是不稳定。

MBGD：小批量梯度下降：每次更新参数时使用一小部分数据样本计算梯度，优点是兼顾速度和稳定性，缺点是需要调整批量大小。

SAG：随机平均梯度下降：SGD的一种改进，它可以加快收敛速度

写出下列学习率衰减各个参数的含义？（考到最后一题）

reduce\_lr = ReduceLROnPlateau(monitor='accuracy', mode = 'max', factor = 0.5, patience = 5, min\_lr=0.0001, verbose=1)

监测模型训练时的准确率变化

当准确率不再上升（监测指标不再改善）时，学习率每次的降低因子为0.5

经过5个epoch而准确率不再上升则停止训练

学习率降低下限为0.0001

记录输出过程

写出下列早停各个参数的含义？

early\_stoping = EarlyStopping(monitor= 'loss', mode= 'min', patience=10, verbose=1)

监测模型训练时的loss函数变化

经过10个epoch而loss函数不再降低则停止训练

记录输出过程

**简答 4\*5**

1. 简述机器学习中划分训练集、验证集、测试集的作用？（考到简答）

训练集：用于模型训练阶段，用于拟合模型的参数，如权重、偏置等

验证集：用于模型预测阶段，用于调整模型超参数，如正则化系数、学习率等

测试集：用于模型评估阶段，用于最终测试，即最终评估模型泛化能力且只能使用一次（巧计：期末测试）

1. 简述什么是数据标准化？（考到选择）

将不同数据间的数值范围映射到相同的数值范围，从而消除数据异常值、规模、特征、纲量单位、分布差异对模型的影响，以至提升收敛速度和鲁棒性，避免特征依赖。

常见的标准化如：

规范化（standardization）：如：z-score标准化

归一化（normalization）：如：min-max标准化

1. 简述 FCN 与 U-Net 的区别？

网络结构不同 ：UNet是encoder和decoder完全对称，FCN的池化层和上采样层并不对称。

跳跃连接不同：UNet的跳跃连接是一种叠操作（连接对应的上采样层和下采样层），FCN的跳跃连接是一种加操作（加和对应的池化层与上采样层）

输入输出不同 ：UNet适用于输入输出大小一致的情况，即图像分割任务。FCN适用于输出大小不一致的情况，即图像语义分割任务

1. 请简述三种用于解决神经网络欠拟合和过拟合问题的方法？

欠拟合：

1. 通过数据增强：如对图片进行平移、旋转、翻转生成新的训练数据
2. 通过增加模型复杂性：如增加网络层数、节点数、卷积核大小
3. 通过SGD、MSGD减少每次的训练量
4. 通过动量法（Momentum）手动调整学习率
5. 通过Adam自适应学习率
6. 迁移学习

过拟合

1. L1、L2正则化
2. 早停（Early Stopping）
3. 随机失活（Dropout）
4. 批归一化（Batch Normalization）
5. 集成学习，如Bagging、Boosting等
6. 交叉验证

**简答2**

机器学习和深度学习的含义和区别是什么？

含义：

机器学习是从数据中自动分析构建模型，并利用模型对未知数据进行预测

它利用统计学习方法来构建模型。常见的机器学习技术有：线性回归、决策树、神经网络

深度学习是机器学习技术的一种

它利用多层神经网络来构建模型，如：多层感知机、CNN、RNN

区别：

深度学习需要大量数据

深度学习不需要特征工程手动提取特征

深度学习需要大量计算资源

深度学习的结果不太具有可解释性

线性回归算法的基本步骤？

确定回归问题：确定特征和目标值的特点

确定loss函数：最小二乘法、最大似然估计、交叉熵等

求解loss函数，使误差最小：正规方程，常见的优化方法、随机梯度下降、Adam等

梯度下降：

1. 确定loss函数
2. 选择起始点
3. 计算当前点梯度
4. 按学习率前进
5. 重复3、4步骤，直到找到最低点（损失函数最优解）

集成学习算法的含义是什么？（考到简答）

集成算法是一种机器学习模型，它通过构建多个学习器，并使用一定策略（如平均，加权平均、投票）组合他们，来完成特定的任务，并且常常获得比单一学习器更好的效果

常见的集成学习算法有三种类别：

Bagging 如：随机森林

Boosting 如：AdaBoost、Gradient Boosting

Stacking 如：如：GBDT（梯度提升树）

注：

Bagging（Bootstrap Aggregating）：采用自助采样法（bootstrap）来获得多个子样本，每个子样本使用相同的算法进行训练，最终的预测结果通过对子样本的预测结果进行投票、平均等方式得到。常见的 Bagging 算法有随机森林（Random Forest）

Boosting：在训练过程中，加强或调整那些已经被弱分类器错分的样本的权值，弱分类器学习的时候会将更多的注意力放在错分的样本上，最终组合多个弱分类器来提高模型的性能。常见的 Boosting 算法包括 AdaBoost、Gradient Boosting。

Stacking：使用多个不同的模型进行预测，将每个模型的预测结果作为新的特征输入到最终的模型中，以提高模型性能。常见的 Stacking 算法有 GBDT（梯度提升树）等。

机器学习模型产生欠拟合过拟合，分别可以采取什么方法？

欠拟合：

1. 通过数据增强：如对图片进行平移、旋转、翻转生成新的训练数据
2. 通过增加模型复杂性：如增加网络层数、节点数、卷积核大小
3. 通过SGD、MSGD减少每次的训练量
4. 通过动量法（Momentum）手动调整学习率
5. 通过Adam自适应学习率
6. 迁移学习

过拟合

1. L1、L2正则化
2. 早停（Early Stopping）
3. 随机失活（Dropout）
4. 批归一化（Batch Normalization）
5. 集成学习，如Bagging、Boosting等
6. 交叉验证

**计算 4\*5**

1. 假设有数据集D，D=(4, 12, 2, 8)。请使用 Min-Max 标准化方法，对数据集D进行标准化，计算经过 Min-Max 标准化方法后的数据集D。
2. 现在有100张图片，其中猫25只，狗75只，小明做了一个分类算法，预测猫：28只，狗62只。预测为猫的结果中，只有20个为真猫。画出猫狗分类器预混淆矩阵A，请计算出混淆矩阵A的准确率，精确率，召回率，F1（考准确率）

|  |  |
| --- | --- |
| 1. 请计算下图的灰度图片经过 3×3 的卷积后的特征图，步长=1， padding=1（考后总结） |  |

|  |  |
| --- | --- |
| 1. 下图是一个 RNN 网络结构，假设所有的 weight = 2， bias = 1，激活函数 = 线性函数，输入数据如下，请计算对应的RNN层的输出结果（考到bias为-1） |  |

|  |  |
| --- | --- |
| 1. 求集合A的熵和集合B的熵，什么是熵，公式是什么？（看考后总结） |  |

答案：

1. (4-2)/(12-2) = 1/5 (0.2, 1, 0, 0.6)

准确率：(20+57) / (20 + 5 + 18 +57) = 0.77

Accuracy = (TP + TN) / (TP + FP + TN + FN) = (20 + 57) / 100 = 0.77

精确率：20 / (20+18) = 20/38 = 0.53

Precision = TP / (TP + FP) = 20 / (20 + 18) = 0.53

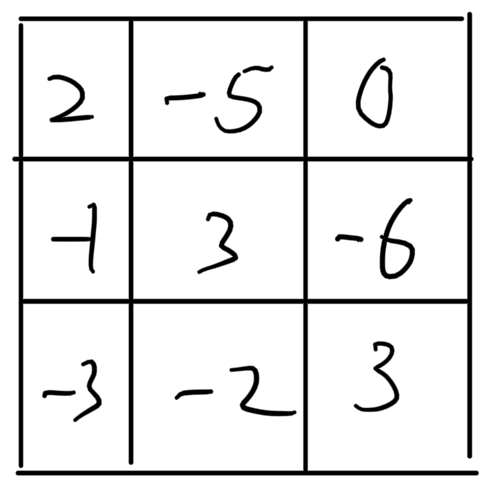
召回率：20 / 25 = 20/25 = 0.8

Recall = TP / (TP + FN) = 20 / (20 + 5) = 0.8

F1：2\*20/38\*20/25 / 20/38 + 20/25 = 0.64

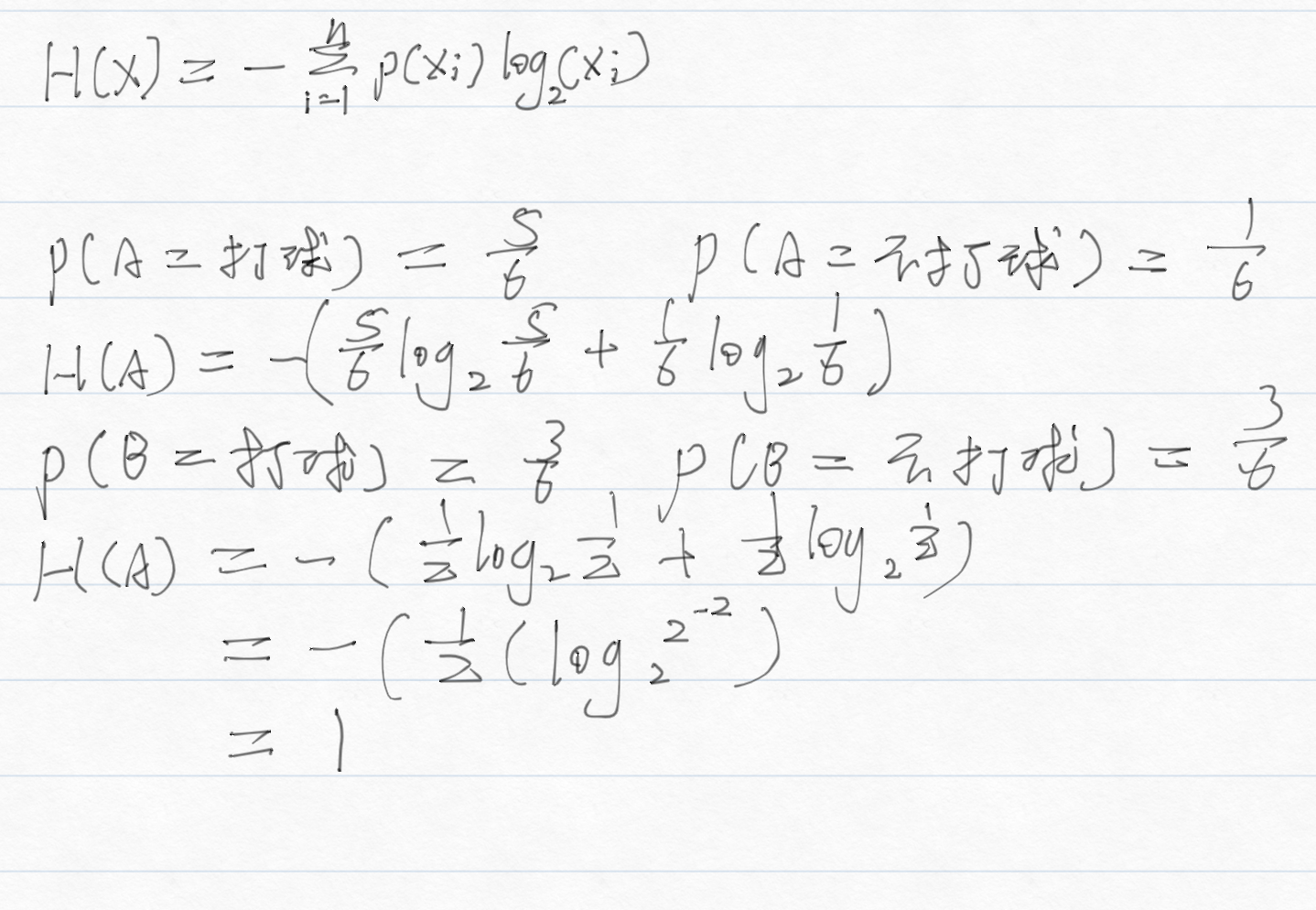
F1 = 2 \* Precision \* Recall / (Precision + Recall) = 2 \* 0.53 \* 0.8 / (0.53 + 0.8) = 0.64





|  |  |
| --- | --- |
|  |  |

1. 熵：熵是表示随机变量不确定性的度量



**案例分析 15\*3**

1. 阅读以下程序，回答如下问题

|  |
| --- |
| Python import tensorflow as tf from keras.layers import Conv2D, concatenate, MaxPooling2D, Flatten, Dense def get\_model(input\_layer):  conv1\_1 = Conv2D(8, (3, 3), activation = 'relu', padding = 'same')(input\_layer)  conv2\_1 = Conv2D(8, (1, 1), activation = 'relu', padding = 'same')(conv1\_1)  conv2\_2 = Conv2D(8, (1, 1), activation = 'relu', padding = 'same')(conv1\_1)  conv2\_2 = Conv2D(8, (3, 3), activation = 'relu', padding = 'same')(conv2\_2)  conv2\_3 = Conv2D(8, (1, 1), activation = 'relu', padding = 'same')(conv1\_1)  conv2\_3 = Conv2D(8, (5, 5), activation = 'relu', padding = 'same')(conv2\_3)  conv2\_4 = concatenate([conv2\_3, conv2\_2, conv2\_1])   conv1\_2 = Conv2D(32, (3, 3), activation = None, padding = 'same')(conv2\_4)  conv1\_2 = MaxPooling2D(pool\_size=(2,2))(conv1\_2)    conv1\_4 = Flatten()(con1\_3)  conv1\_7 = Dense(128, activation='relu')(conv1\_6)  output\_layer = Dense(10, activation='softmax')(conv1\_7)  return output\_layer |

输入为 128\*128\*3 的图像，根据代码画出网络结构图

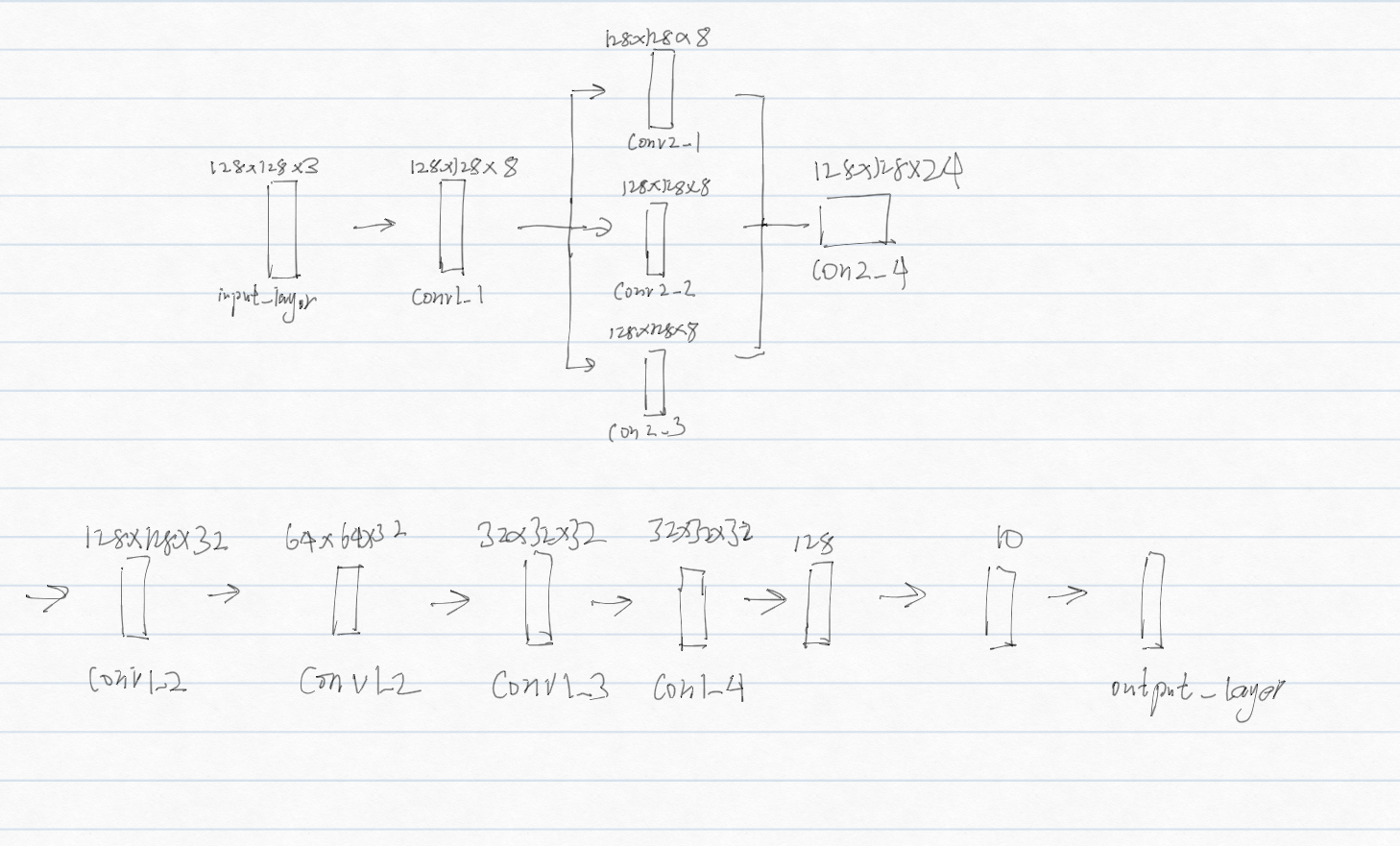
1. 请阅读以下程序，回答如下问题（考到首次输入通道数为3）

|  |
| --- |
| Python # 构建一个简单的CNN模型  model = Sequential() model.add(Conv2D(filters = 10, kernel\_size = (3, 3), strides = 1, input\_shape = (32, 32, 1), activation = 'relu')) model.add(MaxPooling2D(pool\_size = (2, 2))) model.add(Flatten()) model.add(Dense(1, activation = 'sigmoid')) |

1. 请计算 Conv2D 层需要训练的参数的数量，写出计算过程。
2. 请计算 输出层（全连接层）需要训练的参数的数量，写出计算过程。
3. 假设所需的包均已引用，请完成代码填空（5个空）

|  |
| --- |
| Python # 构建一个简单的回归的深度学习模型  # 加载数据集 (x\_train, y\_train), (x\_test, y\_test) = # 数据加载代码  # 定义模型输入 model\_input = \_1\_(shape=(# 输入形状))  # 将模型的输入通过单层神经网络 output = \_\_2(1, \_3\_)(model\_input)  # 创建模型 model = \_4\_(inputs=model\_input, outputs=output)  # 编译模型 model.compile(optimizer='adam', loss='mean\_squared\_error')  # 训练模型 model.fit(x\_train, y\_train, epochs=100, batch\_size=32, \_5\_=(x\_test, y\_test)) |

答案



same：输入输出形状相同

(卷积核高 \* 卷积核宽 \* 通道数 + 1) \* 卷积核个数

(3\*3+1)\*10 = 100

(上一层神经元个数 + 1) \* 当前层神经元个数

32\*32\*1 --> (32+0-3)/1 + 1 (30 \* 30 \* 10) --> (15\*15\*10 + 1) \* 1 = 2251

keras.layers.Input

keras.layers.Dense

activation = 'relu'

keras.models.Model

validation\_data

**案例分析2**

补全下列代码(4空)

|  |
| --- |
| Python # 建立网络结构 model = tf.\_1\_  # 建立输入层，将图片拉成了一维数组作为网络的输入 model.add(tf.keras.layers.\_2\_(input\_shape=(28, 28)))  # 建立第二层:隐含层(神经元节点个数;激活函数) model.add(tf.keras.layers.Dense(512,activation='relu'))  # 建立第三层:隐含层 model.\_3\_(tf.keras.layers.Dense(256,activation='relu')) model.add(tf.keras.layers.Dense(256,activation='relu'))  # 建立第四层(输出层，节点个数为10，激活函数用的softmax) model.add(tf.keras.layers.Dense(128, activation='relu')) model.add(tf.keras.layers.Dense(10, activation='softmax'))  # 建立第五层(输出层，节点个数为10，激活函数用的softmax) model.add(tf.keras.layers.Dense(128, activation='relu')) model.add(tf.keras.layers.Dense(10, activation='softmax'))  # 建立第六层(输出层，节点个数为10，激活函数用的softmax) model.add(tf.keras.layers.Dense(128, activation='relu')) model.add(tf.keras.layers.Dense(10, activation='softmax'))   # 设置网络参数(优化器，l0SS,评价指标) model.\_4\_ model.\_5\_(optimizer='adam',  loss='sparse\_categorical\_crossentropy',  metrics=['acc']  ) |

keras.Sequential()

Flatten

add

summary()

compile

补全下列线性回归的代码(7 空）

|  |
| --- |
| Python from sklearn import \_1\_ #调用sklearn中的linear\_model模块进行线性回归。 import numpy as np #导入numpy计算库 model = \_2\_ #调用线性回归方法 model.\_3\_(X, y) #模型训练 print("截距为: ",model.\_\_4\_\_) #输出截距 print("权重值为: ", model.\_\_5\_\_) #输出权重值 a = model.\_6\_([[12]]) #模型预测 # 检查训练的效果 print(f"训练集R^2：{model.\_7\_(train\_X, train\_y)}") |

linear\_model

linear\_model.LinearRegression()

fit

intercept\_

coef\_

predict

score

补全下列k-means的代码(8空)

|  |
| --- |
| Python import numpy as np from \_1\_ import \_2\_3\_ from sklearn.datasets import make\_blobs # 生成样本数据 centers = [[1, 1], [-1, -1], [1, -1]] X, \_ = make\_blobs(n\_samples=10000, centers=centers, cluster\_std=0.5) bandwidth = \_4\_(X, quantile=0.2, n\_samples=500) ms = \_5\_(bandwidth=bandwidth, bin\_seeding=True) ms.\_6\_(X) labels = ms.\_7\_ cluster\_centers = ms.\_8\_ labels\_unique = np.unique(labels) n\_clusters\_ = len(labels\_unique) print("聚类个数为 : %d" % n\_clusters\_) |

sklearn.cluster

MeanShift,

estimate\_bandwidth

estimate\_bandwidth

MeanShift

fit

labels\_

cluster\_centers\_

补全下列决策树的代码( 10 空)

|  |
| --- |
| Python import numpy as np #导入 np数据集合 from sklearn.datasets import load\_iris #导入鸢尾花数据集 from sklearn.\_1\_ import \_2\_ #导入数据集的划分 from sklearn.\_3\_ import \_4\_ #导入sklearn自带的决策树算法 from sklearn.\_5\_ import \_6\_ #导入算法的评分指标 from sklearn import tree # 导入数据  iris\_dataset=load\_iris() # from sklearn.model\_selection import train\_test\_split #设置训练集为70%数据，测试集为30%的数据 X\_train, X\_test, y\_train, y\_test = train\_test\_split(iris\_dataset['data'], iris\_dataset['target'],random\_state=0,test\_size=0.3) #criterion = gini/entropy 可以用来选择用基尼指数或者熵来做损失函数。 #splitter = best/random 用来确定每个节点的分裂策略。支持“最佳”或者“随机”。 #max\_depth = int 用来控制决策树的最大深度，防止模型出现过拟合。 #min\_samples\_leaf = int 用来设置叶节点上的最少样本数量，用于对树进行修剪。 model = \_7\_(criterion = 'entropy') model.\_8\_(X\_train,y\_train) #模型训练 result = model.\_9\_(X\_test) #模型预测 print('预测正确率：',np.round(\_10\_(result, y\_test),2)\*100,'%') #输出模型的预测结果和评价 |

model\_selection

train\_test\_split

tree

DecisionTreeClassifier

metrics

accuracy\_score

DecisionTreeClassifier

fit

predict

accuracy\_score

补全下列贝叶斯的代码(4空)

|  |
| --- |
| Python from \_1\_ import \_5\_ mlt = \_3\_() # 引入多项式类型 pred = mlt.\_4\_(data, target) # 对模型进行训练 y\_pred = pred.\_5\_([[0, 0, 1]]) # 因为数据特征的顺序是 打游戏，逛街，学习，因此不打游戏，也不逛街，而是学习对于的特征值就是 [0, 0, 1] print(y\_pred) y\_pred = pred.predict([[1, 1, 0]]) print(y\_pred) |

sklearn.naive\_bayes MultinomialNB

MultinomialNB()

fit

predict

**补充**

什么是权重共享——不同的神经网络层使用相同的权重参数

梯度消失是指在反向传播时，由于激活函数的导数和权重矩阵的乘积过小，导致远距离的梯度趋于零，从而使得网络难以学习到长期依赖关系。

LSTM 是一种特殊的 RNN 结构，它引入了一个记忆单元和三个门控制机制，可以有效地避免梯度消失和爆炸，并且能够存储和利用长期信息。

**考后总结**

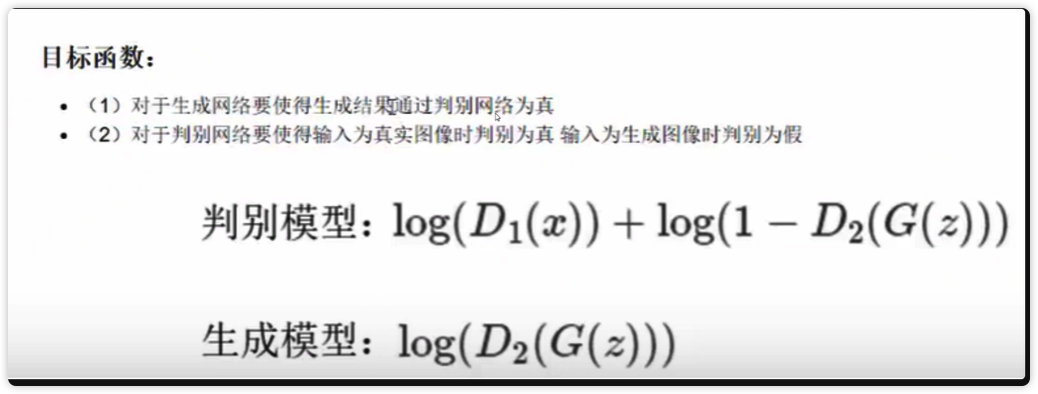
1. 选择：AlaxNet没有使用sigmoid激活函数，使用到了GPU计算，随机失活
2. 选择：风格迁移的网络是——Pix2Pix生成器  
   FCN和Deeplab都是全连接的





1. 简答：生成对抗神经网络是什么？





1. 计算题 池化是怎么计算的  
   MaxPooling 求最大值 而不是求和

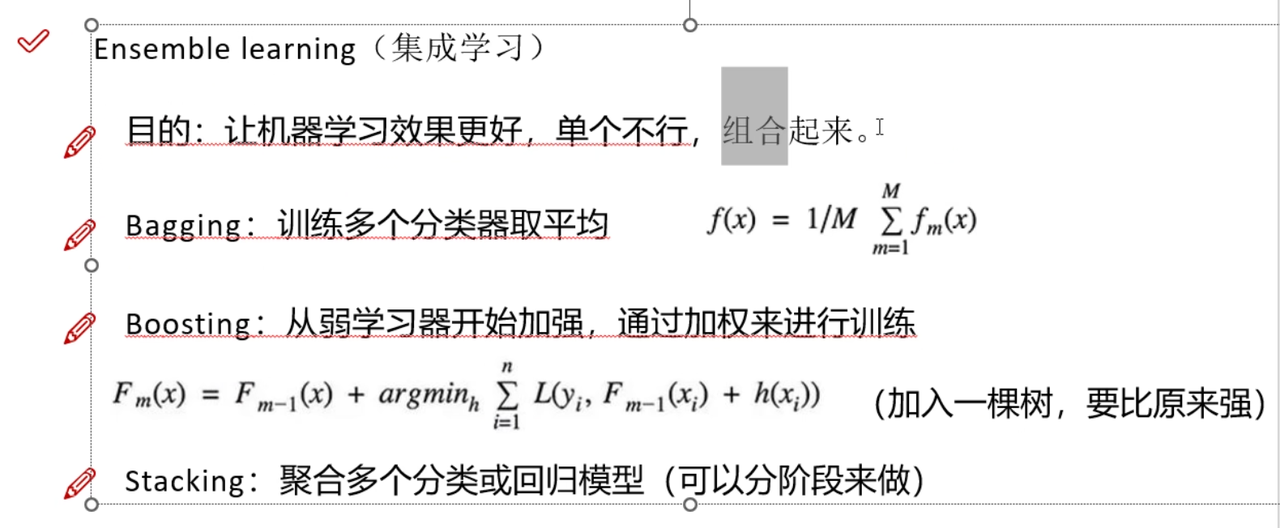
池化是一种常用的降采样方法，用于减少图像数据的维度和大小。通常情况下，池化是对每个图像区域计算池化值，从而得到更小的输出特征图。

池化操作的计算过程如下：

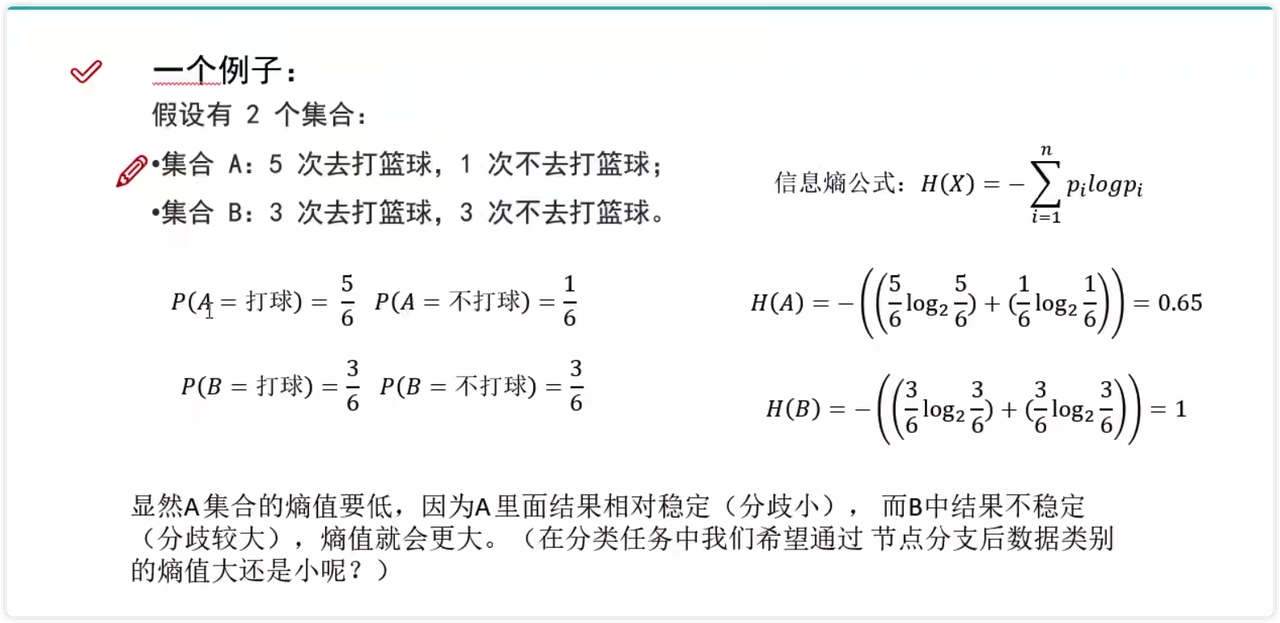
1. 选择一个池化窗口大小和步幅。池化窗口是指在输入图像上沿着行和列滑动的矩形框。步幅是指池化窗口在滑动时的跨度。
2. 对于每个池化窗口，根据池化操作的类型（最大池化、平均池化等）计算池化值。例如，对于最大池化操作，池化值是窗口内像素的最大值；对于平均池化操作，池化值是窗口内像素的平均值。
3. 将池化值作为输出特征图的像素值，从而得到更小的特征图。

池化的主要作用是减少图像数据的维度和大小，从而提高后续神经网络的计算效率。此外，池化还可以帮助提取输入特征的主要信息，从而提高分类、识别等任务的准确性。

1. 简答题考到来集成学习的含义



1. 计算熵——原题答案0.65为什么可以写等于



1. 代码题还有一个svm和plt结合起来预测的函数名 什么svm\_什么(model, plt)