基本知识 什么是 AI 人工智能 是利用数字计算机或者数字计算机控制的机器模拟、延伸和扩展人的智能，感知环境、获取知识并使用知识获得最佳结果的理论、方法、技术及应用系统。**AI 的四大主流流派** 1.符号主义（Symbolism）例：知识图谱；2.连接主义（Connectionism）例：深度神经网络；3.行为主义（Behaviourism）例：机器人；4.统计主义（Statisticsism）例：机器学习。
AI、机器学习、深度学习三者之间的异同和关联 1. 三者属于包含关系：AI 包含机器学习，机器学习包含深度学习，深度学习主要指机器学习中的深度神经网络；2. 在机器学习中，需要先进行人工特征提取，再定义 Model；而大部分深度学习只构建一个端对端的模型，没有人工特征提取；3. 在深度学习中，有时需要先进行数据清洗、格式转换、特征提取（卷积）等操作，再将数据喂给全连接神经网络。**机器学习三步曲**：1 定义 Model； 定义 Loss/cost/error/objective function；2 找到 Model 中的最佳 function：利用梯度下降来迭代调整参数，以使得损失函数达到最小值。3 预测：验证集和测试集。前向传播：预测输出，计算 Loss；通过验证集可以用于超参数调优，模型经过训练集训练，和验证集调优，然后交给测试集测试性能。**Deep Learning 三步曲** (1) 定义一个 Model，深度学习里 Model 是 Neural Network Structure；(2) 定义这个 Model 好坏，使用合适的 Loss Function 来衡量损失；(3) 找出最佳参数，使用反向传播不断优化参数，从而找出最佳参数。
机器学习项目的通用工作流程 1 定义问题：软件架构设计、确定评价指标 2 获取数据：自动化方式 3 研究数据：可视化方式，相关性研究等 4 准备数据：数据清理、特征选择及处理 5 研究模型：确定评估方法、列出可能的模型并训练，选择最有希望的 3 到 5 个模型 6 微调模型：寻找最佳超参数，模型融合，评估泛化性能 7 展示解决方案：将工作进行文档化总结展示 8 启动、监视、维护系统：投入使用。**神经网络的通用工作流程** 1 定义问题、收集数据集 2 选择衡量成功的指标 3 确定评估方法留出验证？ K 折验证？你应该将哪一部分数据用于验证 4 准备数据 5 开发比基准更好的模型 6 扩大模型规模：开发过拟合的模型 7 模型正则化与调节超参数：基于模型在验证数据上的性能。**深度学习和机器学习有什么不同** 机器学习：利用计算机、概率论、统计学等知识，输入数据，让计算机学会新知识。机器学习的过程，就是训练数据去优化目标函数。深度学习：是一种特殊的机器学习，具有强大的能力和灵活性。它通过学习将世界表示为嵌套的层次结构，每个表示都与更简单的特征相关，而抽象的表示则用于计算更抽象的表示。传统的机器学习需要定义一些手工特征，从而有目的的去提取目标信息， 非常依赖任务的特异性以及设计特征的专家经验。而深度学习可以从大数据中先学习简单的特征，并从其逐渐学习到更为复杂抽象的深层特征，不依赖人工的特征工程，这也是深度学习在大数据时代受欢迎的一大原因。**机器学习的定义**对于某类任务 T 和性能度量 P，一个计算机程序被认为可以从经验 E 中学习是指，通过经验 E 改进后，它在任务 T 上由性能度量 P 衡量的性能有所提升。任务 T：分类，翻译等机器学习的目标任务 性能度量 P：最常用的有准确率， 召回率，F 值等，根据任务 T 不同，P 也不尽相同。 经验 E：训练集。**什么场合下需要使用机器学习？** 需要进行大量手工调整或需要拥有长串规则才能解决的问题：机器学习算法通常可以简化代码、提高性能。问题复杂，传统方法难以解决：最好的机器学习方法可以找到解决方案。环境有波动：机器学习算法可以适应新数据。洞察复杂问题和大量数据。**模型选择的三个维度**：Scenario ： 数据集是否带标签？是否是强化学习 Task ： 要预测的值（即输出值）是否是数值类型？是否连续？Method ： 选择的模型。**机器学习的分类及差异。1 监督学习**的目标是建立一个学习过程，将预测结果与“训练数据”（即输入数据）的实际结果进行比较，不断的调整预测模型，直到模型的预测结果达到一个预期的准确率。例：手写字符识别、肿瘤分类、预测天气、支持向量机、线性判别。特点：训练样本数据和待分类的类别已知，且训练样本数据皆为标签数据。**2 非监督学习**从无标记的训练数据中推断结论，它可以在探索性数据分析阶段用于发现隐藏的模式或者对数据进行分组。例：聚类分析、主成分分析。特点：训练样本数据和待分类的类别已知，但训练样本数据皆为非标签数据。**3 半监督学习**的训练数据通常是少量有标记数据及大量未标记数据，介于无监督学习和监督学习之间。例：聚类假设、流形假设。特点：训练样本数据和待分类的类别已知，然而训练样本既有标签数据，也有非标签数据。**4 强化学习**的输入数据作为对模型的反馈，强调如何基于环境而行动，以取得最大化的预期利益。与监督式学习之间的区别在于，它并不需要出现正确的输入/输出对，也不需要精确校正次优化的行为。强化学习更加专注于在线规划，需要在探索（在未知的领域）和遵从（现有知识）之间找到平衡。例：学习下围棋、打星际争霸、DotA2、双人德州扑克。（全部都是 1v1 的场景，目前强化学习领域对于多人博弈研究的很少）特点：决策流程，激励系统，学习一系列的行动。**5 迁移学习**是通过从已学习的相关任务中迁移其知识来对需要学习的新任务进行提高。例：牛津的 VGG 模型、谷歌的 Inception 模型和 word2vec 模型、微软的 ResNet 模型。特点：需求的训练数据集合较小、训练时间较小，可以方便的进行迁移以满足个性化。**6 弱监督学习**，特点：弱监督学习可以看做是有多个标记的数据集合，次集合可以是空集，单个元素，或包含多种情况（没有标记，有一个标记，和有多个标记）的多个元素。 数据集的标签是不可靠的，这里的不可靠可以是标记不正确，多种标记，标记不充分，局部标记等。已知数据和其一一对应的弱标签，训练一个智能算法，将输入数据映射到一组更强的标签的过程。标签的强弱指的是标签蕴含的信息量的多少，比如相对于分割的标签来说，分类的标签就是弱标签。算法举例：举例，给出一张包含气球的图片，需要得出气球在图片中的位置及气球和背景的分割线，这就是已知弱标签学习强标签的问题。**机器学习三类数据集**：训练集：利用训练集上的误差来训练模型，学习参数。验证集：利用验证集上的误差来挑出备选模型，确定各个备选模型的最佳超参数，并从中挑选出最佳模型。测试集：用于估计泛化误差，评估模型的表现。**机器学习两个计算过程**：1 学习 训练 优化：目标是找到最佳函数来表达输入输出之间的关系，理想的最佳函数，应在优化和泛化的平衡点处，应刚好在欠拟合和过拟合的界线上。2 推理 预测 泛化：将找到的最佳函数应用到测试数据 真实数据上，做出推理 预测。**机器学习两类函数**：最佳函数：（模型训练的终极目标是找到这个函数）成本/代价/损失函数：用来定义训练过程中找到的函数的好坏程度。**机器学习两类参数**：参数：模型训练过程中自动找到。超参数：程序员来选择/指定 手自动：网格搜索、随机搜索。手动：程序员设定。**训练过程中，模型的状态**：欠拟合过拟合正好。**机器学习两个评估指标**：模型评估标准（在训练数据集和验证数据集上）：优化指标，即损失函数。通常用于模型的训练中。损失函数的取值 越小，则越好。模型测试标准（在测试集上）：验收指标。用于测试模型的泛化性能。比如：样本平衡的分类问题中常用的验收指标 ROC AUC 但是， ROC/AUC 不能直接被优化，故不能作为损失函数，而是将 ROC AUC 的替代指标 交叉熵作为优化指标，即损失函数。**机器学习评估方法**：留出验证集，交叉验证 (CrossValidation)。**Epoch**：对整个数据集进行一次 forward 和 backward 过程，被称为一个 Epoch。 **Batch_size**：每一批的大小。在一次前向传播/反向传播中，批量的大小。**Batch**：使用训练集中一小部分样本对模型权重进行一次反向传播的参数更新，这一小部分样本被称为一个 Batch。**Iteration**：梯度更新的次数，也就是总过已经使用了多少个 batch。**Batch_num**：每次使用 Batch_size 个数据，需要 batch_num 次迭代完一个 epoch。举个栗子：假设我的训练集中总样本数为 2048、Batch_size=128，那么我需要迭代(Iterations)16 次才能完成一个 Epoch。**误差来源分析** 1 偏差 bias 期望预测与真实标记的误差，偏差越大偏离理论值越大。在一个训练集 D 上模型 f 对测试样本 x 预测输出为 f(x;D)，那么学习算法 f 对测试样本 x 的期望预测为：f(x) =E_D[f(x; D)] 这里用偏差的平方来表示偏差的计算公式 Bias^2(x) = (f(x) − y)^2。2 方差 variance 预测模型的离散程度，方差越大离散程度越大。使用样本数相同的不同训练集产生的方差为:var(x) = E_D[(f(x; D)−f(x))^2]泛化误差：期望误差和经验误差之间的差异。用来表示泛化性能的优劣。训练集趋向于无穷大时，泛化误差趋向于 0。**什么是交叉验证** 将数据集 D 划分成 k 个大小相似的互斥子集，每次用 k-1 个子集作为训练集，余下的子集做测试集，最终返回 k 个训练结果的平均值。交叉验证法评估结果的稳定性和保真性很大程度上取决于 k 的取值。适用于数据集不是特别大时。**几种正则化** L1-Norm：参数绝对值之和。$L'(\theta)=L(\theta)+\lambda||\theta||$1，其中$||\theta||_1=\Sigma_{i=1}^{n}|\theta_i|$作用：特征筛选，把得最终的参数为稀疏性向量 L2-Norm：参数平方的和。$L'(\theta)=L(\theta)+\lambda||\theta||^2$，其中$||\theta||_2=\Sigma_{i=1}^{n}\theta_i^2$作用：通过限制模型复杂度，从而避免过拟合，提高泛化能力。Elastic Net:(L1+L2):$L'(\theta)=L(\theta)+\lambda[\rho||\theta||_1+(1-\rho)||\theta||_2]$。
如何区分回归问题与分类问题 输出值可以有限枚举出来就是分类问题，否则就是回归问题。**为什么不可以用线性回归的模型解决分类问题** 当样本分布比较复杂时，线性回归无法做到准确的分类。有些问题不是线性可分的（异或问题）。可以使用线性回归解决二分类问题，但是无法解决多分类的问题。原因是：需要人为的划定区间，引入误差。在损失函数上，MSE 无法很好地评估模型。举例：假如有 5 元分类问题，class0 class4，如果正确分类是 class4，但是预测到了 class1，和预测到了 class3。根据 MSE，损失函数是不同的。这就无法很好地区分了。**在逻辑回归中，为何使用 Cross Entropy 作为损失函数而不使用 MSE**，如果是交叉熵，距离 target 越远，微分值就越大，就可以做到距离 target 越远，更新参数越快。而平方误差在距离 target 很远的时候，微分值非常小，会造成移动的速度非常慢，这就是很差的效果了。离目标离目标很近的时候，$\partial L/\partial w$ 很小导致更新很慢。逻辑回归公式$\sigma(z)=1/1+e^{-z}$，$z(x)=wx+b$，分别对 w 和 b 求导，有$\partial\sigma(z)/\partial w=d\sigma(z)/dz\cdot\partial z/\partial w=\sigma(z)(1-\sigma(z))*x$和$\partial\sigma(z)/\partial b=d\sigma(z)/dz\cdot\partial z/\partial b=\sigma(z)(1-\sigma(z))$如果使用 MSE 作为损失函数的话，那写出来是这样的$L(w,b)=1/2m\sum_{i=1}^{m}(\sigma_{w,b}(x^{(i)})-y^{(i)})$使用梯度下降来进行优化时，需要计算 L(w,b)对 w 和 b 的偏导数：$\partial L(w,b)/\partial w=1/m\sum_{i=1}^{m}(\sigma_{w,b}(x^{(i)})-y^{(i)})\sigma_{w,b}(x^{(i)})(1-\sigma_{w,b}(x^{(i)}))x^{(i)}$和（把前式的 w 换成 b，最后的 x 不要），所以在 σ(x)接近于 1 或者 0 的时候，也就是预测的结果和真实结果很相近或者很不相近的时候，σ(x)和 1− σ(x)中总有一个会特别小，从而使得优化速度大大减缓。而当使用 Cross Entropy 作为损失函数时，损失函数为：$L(w,b)=1/m\sum_{i=1}^{m}(-y^{(i)}\log(\sigma_{w,b}(x^{(i)}))-(1-y^{(i)})\log(1-\sigma_{w,b}(x^{(i)})))$L(w,b)分别对 w 和 b 求偏导，结果如下$\partial L(w,b)/\partial w=1/m\sum_{i=1}^{m}(\sigma_{w,b}(x^{(i)})-y^{(i)})x^{(i)}$ $\partial L(w,b)/\partial b=1/m\sum_{i=1}^{m}(\sigma_{w,b}(x^{(i)})-y^{(i)})$这样梯度始终和预测值与真实值之差挂钩，预测值与真实值偏离很大时，梯度也会很大，偏离很小时，梯度会很小。所以我们更倾向于使用 Cross Entropy 而不使用 MSE。4. 级联逻辑回归(Cascading logistic regression model)模型是啥？为什么要引入这个概念？级联逻辑回归模型是将很多的逻辑回归接到一起，以进行特征转换再用一个逻辑回归来进行分类。级联逻辑回归模型是神经网络的雏形。5. Loss Function - 损失函数 回归任务假设函数：$h_{w,b}(x)=wx+b$分类任务假设函数：$h_{w,b}(x)=\sigma(wx+b)$在回归任务中，多使用均方误差作为损失函数：上面有。在分类任务中，多使用交叉熵作为损失函数，上面有。6. Sigmoid 和 Softmax Sigmoid Function 不具体表示哪一个函数，而是表示一类 S 型函数，常用的有逻辑函数 σ(z)上面有。而 Softmax，或称归一化指数函数，是逻辑函数的一种推广，二分类情况下，Softmax 退化为逻辑函数。该函数的形式通常按下面的式子给出：$\sigma(z)_j=e^{z_j}/\sum_{k=1}^{K}e^{z_k}\quad j=1,\dots,K$。**Logistic 回归与 Softmax 回归联系与区别** Logistic 回归与 Softmax 回归是两个基础的分类模型，虽然听名字像是回归模型，实际上并非如此。Logistic 回归，Softmax 回归以及线性回归都是基于线性模型。其实 Softmax 就是 Logistic 的推广，Logistic 一般用于二分类，而 softmax 是多分类。softmax 是 lr 的升级版，可以用来多类问题划分，lr 是二分问题。softmax 的权值 w 是矩阵，lr 的是向量。softmax 输出是属于每一个类的概率值，概率最大的就是其对应类。lr 就两个值，是和不是（0 - 1 ‖ - 1 - 1 一般是这样）。只是应用场景不同，其他的没啥区别。
分类 分类器的性能评估指标 True Positive （真正 TP ）被模型预测为正的正样本。True Negative （真负 ，TN ）被模型预测为负的负样本。False Positive （假正 ，FP ）被模型预测为正的负样本。False Negative （假负 ，FN ）被模型预测为负的正样本。混淆矩阵竖轴为预测类别，横轴为实际类别。**评价指标** 1）准确率或者正确率（accuracy）正确率是我们最常见的评价指标，accuracy = (TP+TN)/(P+N)，正确率是被分对的样本数在所有样本数中的占比，通常来说，正确率越高，分类器越好。2)错误率（error rate)错误率则与正确率相反，描述被分类器错分的比例，error rate = (FP+FN)/(P+N)，对某一个实例来说，分对与分错是互斥事件，所以 accuracy =1 - error rate。3) 灵敏度（sensitivity）sensitivity = TP/P，表示的是所有正例中被分对的比例，衡量了分类器对正例的识别能力。4) 特异性（specificity)specificity = TN/N，表示的是所有负例中被分对的比例，衡量了分类器对负例的识别能力。5) 精确率或者查准率或者精度（precision）precision=TP/(TP+FP)，精度是精确性的度量，表示被分为正例的示例中实际为正例的比例。6) 查全率或者召回率（recall）召回率是覆盖面的度量，度量有多个正例被分为正例，recall=TP/(TP+FN)=TP/P=sensitivity，可以看到召回率与灵敏度是一样的。7) 其他评价指标：计算速度：分类器训练和预测需要的时间；鲁棒性：处理缺失值和异常值的能力；可扩展性：处理大数据集的能力；可解释性：分类器的预测标准的可理解性，像决策树产生的规则就是很容易理解的，而神经网络的一堆参数就不好理解，我们只好把它看成一个黑盒子。8）精度和召回率反映了分类器分类性能的两个方面。如果综合考虑查准率与查全率，可以得到新的评价指标 F1-score，也称为综合分类率：**F1**=2×precision×recall/（precision+recall）。为了综合多个类别的分类情况，评测系统整体性能，经常采用的还有微平均 F1（micro-averaging）和宏平均 F1（macro-averaging ）两种指标。（1）宏平均 F1 与微平均 F1 是以两种不同的平均方式求的全局 F1 指标。（2）宏平均 F1 的计算方法先对每个类别单独计算 F1 值，再取这些 F1 值的算术平均值作为全局指标。（3）微平均 F1 的计算方法是先累加计算各个类别的 a、b、c、d 的值，再由这些值求出 F1 值。（4）由两种平均 F1 的计算方式不难看出，宏平均 F1 平等对待每一个类别，所以它的值主要受到稀有类别的影响，而微平均 F1 平等考虑文档集中的每一个文档，所以它的值受到常见类别的影响比较大。**ROC 曲线 AUC 面积和 PRC** ROC (Receiver Operating Characteristic)受试者工作特征曲线，在不同的分类阈值下，分别计算其对应的“真正例率” TPR 和“假正例率 FPR，然后以 TPR 为 Y 轴，以 FPR 为 X 轴，画出不同分类阈值下（FPR，TPR）值对。$TPR=TP/(TP+FN)$ ，$FPR=FP/(TN+FP)$。ROC 曲线越靠近左上角，说明其对应模型越可靠。若两个学习器的 ROC 曲线发生交叉，则比较 ROC 曲线下的面积（Area Under Curve，AUC）来评价模型，AUC 越大，模型越可靠。PRC 曲线是 Precision Recall Curve 的简称，描述的是 precision 和 recall 之间的关系，以 recall 为横坐标，precision 为纵坐标画出所有阈值情况下的（R P）值对。该曲线下面积实际上是目标检测中常用的评价指标平均精度（Average Precision，AP）。AP 越高，说明模型性能越好。实践中，引入 “平衡点 ”（BEP）来度量，BEP 表示“查准率=查全率”时的取值，值越大表明分类器性能越好。**多分类任务的常用评价指标有哪些？** mAP (Mean Average Percision))：平均 AP 所有 类别 的 AP 的算数平均。Kappa 系数：kappa 系数是用在统计学中评估一致性的一种方法，取值范围是[-1,1]实际应用中，一般是 [0,1]，与 ROC 曲线中一般不会出现下凸形曲线的原理类似。这个系数的值越高，则代表模型实现的分类准确度越高。铰链损失 （Hinge loss）：一般用来使“边缘最大化” （ maximal margin ）。损失取值在 0~1 之间，当取值为 0 ，表示多分类模型分类完全准确，取值为 1 表明完全不起作用。**多标签分类的评价指标**：1.Hamming loss(汉明损失)，表示所有 label 中错误样本的比例，所以该值越小则网络的分类能力 越强。计算公式：$\text{HammingLoss}(x_i,y_i)=\frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|x_i\oplus y_i|}{|L|}$其中：|D|表示样本总数，|L|表示标签总数，xi 和 yi 分别表示预测结果和 ground truth。xor 表示异或运算。2. Jaccard index(杰卡德指数），概念挺陌生，公式是再熟悉不过了。其中：T 表示 ground truth，P 表示预测结果。再观察这个公式，和检测算法中的 IOU 多么相近。$|T\cap P|/|T\cup P|$。3. 精度、召回率和 F1 值。其中精度计算公式为$|T\cap P|/|P|$，召回率计算公式为$|T\cap P|/|T|$， F1 值的 计算为精度和召回率的调和平均数。4.准确匹配。这个是最严格的标准了，是预测结果和 ground truth 完全一致时的样本数与总的样 本数的比值。**正确率能很好的评估分类算法吗**不同算法有不同特点，在不同数据集上有不同的表现效果，根据特定的任务选择不同的算法。如何评价分类算法的好坏，要做具体任务具体分析。对于决策树，主要用正确率去评估，但是其他算法，只用正确率能很好的评估吗？答案是否定的。正确率确实是一个很直观很好的评价指标，但是有时候正确率高并不能完全代表一个算法就好。比如对某个地区进行地震预测，地震分类属性分为 0：不发生地震、1 发生地震。我们都知道，不发生的概率是极大的，对于分类器而言，如果分类器不加思考，对每一个测试样例的类别都划分为 0，达到 99%的正确率，但是，问题来了，如果真的发生地震时，这个分类器毫无察觉，那带来的后果将是巨大的。很显然，99%正确率的分类器并不是我们想要的。出现这种现象的原因主要是数据分布不均衡，类别为 1 的数据太少，错分了类别 1 但达到了很高的正确率缺忽视了研究者本身最为关注的情况。**生成模型和判别模型的区别**生成模型：由数据学习联合概率密度分布 P(X,Y)，然后求出条件概率分布 P(Y|X)作为预测的模型，即生成模型：P(Y|X)= P(X,Y)/ P(X)（贝叶斯概率）。基本思想是首先建立样本的联合概率概率密度模型 P(X, Y)，然后再得到后验概率 P(Y|X)，再利用它进行分类。典型的生成模型有朴素贝叶斯，隐马尔科夫模型等。判别模型：由数据直接学习决策函数 Y=f(X)或者条件概率分布 P(Y|X)作为预测的模型，即判别模型。基本思想是有限样本条件下建立判别函数，不考虑样本的产生模型，直接研究预测模型。典型的判别模型包括 k 近邻，感知级，决策树，支持向量机等。这些模型的特点都是输入属性 X 可以直接得到后验概率 P(Y|X)，输出条件概率最大的作为最终的类别（对于二分类任务来说，实际得到一个 score，当 score 大于 threshold 时则为正类，否则为负类）。
集成学习 什么是模型融合集成学习？ 将一组弱 基预测器的预测结果进行融合集成，以实现一个强预测器，从而获得比单个预测器更好的泛化能力。**模型融合的聚合策略？** 1 平均法：一般用于回归预测模型 中。平均法包括 一般的平均和加权平均融合。投票回归器 (Voting Regressor Boosting 系列融合模型 2.投票法 ：一般用于分类模型 。具体可分为绝对多数投票（得票超过一半），相对多数投票（得票最多），加权投票 。例如 bagging 模型投票分类器(Voting Classifier) 3.学习法：通过另一个预测器称为混合器或元学习器来实现聚合。常见的有 Stacking 和 Blending 两种。stacking 一般使用交叉验证的方式,Blending 是建立一个 Holdout 集。**模型融合的分类**：按照弱预测器之间的依赖关系，模型融合方法可分为两类。1 弱预测器间不存在强依赖关系、可同时学习的并行化方法，如果弱预测器只有一种，即所有的弱预测器都是基于同一类模型。代表是 Bagging 和 “随机森林”。如果弱预测器的种类多。比如投票分类器，投票回归器， Stacking 。2 弱预测器间存在强依赖关系、必须串行学习的序列化方法， 代表是 Boosting 方法有下面几种变体 AdaBoost，Gradient Boosting：GBRT，XGBoost ，LightGBM。注意区分：强依赖≠强相关性，弱预测器之间存在强依赖，是指单个弱预测器的训练存在必需的先后关系；弱预测器 之间若存在强相关性，会影响融合后的性能。**模型融合集成学习一定有效吗？** 可以 通过数学证明：随着弱预测器数目 T 的增大，集成后得到的强预测器的错误率将呈指数 级下降最终趋向于零 。弱预测器数目 T 过大，可能导致强预测器出现过拟合。集成学习有效，必须：弱预测器之间的性能表现差不多，不能差距太大。同时弱预测器之间的相关性要尽可能的小。
误差、偏差和方差有什么区别和联系对于 Error：误差（error）：一般地，我们把学习器的实际预测输出与样本的真是输出之间的差异称为“误差”。 Error = Bias + Variance + Noise，Error 反映的是整个模型的准确度。噪声：描述了在当前任务上任何学习算法所能达到的期望泛化误差的下界，即刻画了学习问题本身的难度。对于 Bias：Bias 衡量模型拟合训练数据的能力（训练数据不一定是整个 training dataset，而是只用于训练它的那一部分数据，例如：mini-batch），Bias 反映的是模型在样本上的输出与真实值之间的误差，即模型本身的精准度。 Bias 越小，拟合能力越高（可能产生 overfitting）；反之，拟合能力越低（可能产生 underfitting）。偏差越大，越偏离真实数据，如下图第二行所示。对于 Variance：方差公式：$S_N^2=1/N\sum^N(x_i-\bar{x})^2$ 。Variance 描述的是预测值的变化范围，离散程度，也就是离其期望值的距离。方差越大，数据的分布越分散，模型的稳定程度越差。Variance 反映的是模型每一次输出结果与模型输出期望之间的误差，即模型的稳定性。Variance 越小，模型的泛化的能力越高；反之，模型的泛化的能力越低。如果模型在训练集上拟合效果比较优秀，但是在测试集上拟合效果比较差劣，则方差较大，说明模型的稳定程度较差，出现这种现象可能是由于模型对训练集过拟合造成的。**如何判断欠拟合和过拟合？** 1 利用训练误差和验证误差，训练误差和验证误差都很大，并且很接近，此时模型存在欠拟合问题：当训练误差很小，但验证误差还是很大时，此时模型存在过拟合问题。2 利用训练集和验证集上的学习曲线。欠拟合：高 bias error，低 variance error 过拟合：低 bias error，高 variance error。好：低 bias error，低 variance error。学习曲线就是不同训练集大小时训练集和交叉验证的准确率。当训练集和测试集的误差收敛但却很高时，为高偏差。 训练集和验证集的准确率都很低，很可能是欠拟合。当训练集和测试集的误差之间有大的差距时，为高方差。 当训练集的准确率比其他独立数据集上的测试结果的准确率要高时，一般都是过拟合。**如何解决欠拟合** 1 添加其他特征项。组合、泛化、相关性、上下文特征、平台特征等特征是特征添加的重要手段，有时候特征项不够会导致模型欠拟合。2 添加多项式特征。例如将线性模型添加二次项或三次项使模型泛化能力更强。例如，FM（Factorization Machine）模型、FFM（Field-aware Factorization Machine）模型，其实就是线性模型，增加了二阶多项式，保证了模型一定的拟合程度。3 可以增加模型的复杂程度。4 减小正则化系数。正则化的目的是用来防止过拟合的，但是现在模型出现了欠拟合，则需要减少正则化参数。**如何解决过拟合**：1 重新清洗数据，数据不纯会导致过拟合，此类情况需要重新清洗数据。 2 增加训练样本数量。 3 降低模型复杂程度。 4 增大正则项系数。 5 采用 dropout 方法，dropout 方法，通俗的讲就是在训练的时候让神经元以一定的概率不工作。 6. early stopping。 7. 减少迭代次数。 8. 增大学习率。 9. 添加噪声数据。 10. 树结构中，可以对树进行剪枝。 11. 减少特征项。**如何加快模型的训练 1 特征缩放/标准化** (1) Feature Scaling - 特征缩放/归一化 输入值减去样本中最小值，然后除以样本范围，这样会使得结果永远在 0~1 的范围内，所有特征参数差不多一个速度优化到最低点。式子如下：$x'=(x-\min(x))/[max(x)-min(x)]$(2) Mean Normalization - 均值归一化 输入值减去样本均值，然后除以样本范围。式子如下：$x'=x-avg(x)/[max(x)-min(x)]$(3) Standardization 标准化：让输入的值减去样本平均数 μ，再除以样本标准差 σ。经过这样的处理，数据符合标准正态分布，即均值为 0，标准差为 1。$x'=(x-\mu)/\sigma$。**2 梯度下降** (1) Gradient Descent - 梯度下降 如果需要找到一个函数的局部极小值，必须朝着函数上当前点所对应梯度（或者是近似梯度）的反方向，前进规定步长的距离进行迭代搜索。$\theta^{t+1}=\theta^t-\eta\nabla L(\theta^t)$ 为什么要以梯度的反方向为更新方向？因为梯度方向是函数方向导数最大的方向，所以沿着梯度方向反方向更新的话，函数下降变化幅最大。优点：能保证朝梯度下降的方向去优化，易于并行实现。缺点：样本数很多时，会很慢。(2) Stochastic Gradient Descent - 随机梯度下降每次更新只用到了一个样本，如果这个训练集有 m 个样本，那么梯度下降更新一次参数，随机梯度下降已经更新了 m 次参数了。优点：快。BGD 更新一次的时候，SGD 能更新的次数=Batch_size。缺点：引入了随机噪声，可能会出现震荡的现象。当目标函数非凸时，反而可以使其逃离局部最优点。无法充分利用计算机的并行能力。(3) Mini-batch Gradient Descent Mini-batch 梯度下降是梯度下降和随机梯度下降的中和版本，Mini-batch 梯度下降每次更新所考虑的样本是可以被指定的，如果总共有 m 个样本，那就可以在 1~m 中任意指定。如果每次更新时所参考的样本数合适，那么既兼顾了随机梯度下降更新速度快的特性，又兼顾了梯度下降更新的稳定性。优点：收敛快，计算开销小。**3. 调整学习率** 当我们在训练过程中，发现 loss 下降的很慢时，可以适当增大学习率；发现 loss 不降反增的时候，要降低学习率。(1) Adagrad 算法的学习率会根据迭代次数来放缓学习率，从而达到学习率越来越小的目的。通过公式可以看到，学习率是会一直除以前面所有梯度的平方和再开根号的，这一定是一个大于 0 的数，所以学习率会越来越小。但是防止一开始的时候梯度就是 0，如果让分母变为 0 会导致错误的，所以后面还要跟一个很小的正数 ε，最终的式子是这样的：$w^{t+1}\leftarrow w^t-\eta/\sqrt{\sum_{i=0}^{t}(g^i)^2+\epsilon}g^t$ Adagrad 算法也有很多不足：a) 如果初始的学习率设置过大的话，这个学习率要除以一个较大梯度，那么此算法会对梯度的调节太大；b) 在训练的中后期，分母上梯度平方的累加将会越来越大，使 gradient->0，使得训练提前结束。(2) RMSprop Adagrad 算法的改进版 RMSprop 算法：$w^{t+1}\leftarrow w^t-\eta g^t/\sigma^t\quad \sigma^t=\sqrt{\alpha(\sigma^{t-1})^2+(1-\alpha)(g^t)^2}$Adagrad 和 RMSprop 算法这两个算法很相近，不同之处在于 RMSprop 算法增加了一个衰减系数 α 来控制历史信息的获取多少。(3) SGD with Momentum (SGD-M) SGD 在遇到沟

壑时容易出现抖动现象。为此，可以为其引入动量 Momentum，加速 SGD 在正确方向的下降并抑制震荡。$v^0-0\quad v^t=\lambda v^{t-1}-\eta\nabla L(\theta^{t-1})\quad \theta^t=\theta^{t-1}+v^t$这里多了一个 m t，可以将其想象为动量或者惯性，意味着参数更新方向不仅由当前的梯度决定，也与此前累积的下降方向有关。如果上一次梯度和只一次同方向，m t 会越来越大，参数也会更新越来越快；如果方向不同，m t 会让上次更小，参数更新速度减慢。γ 是取上一次更新的动量大小，通常取 0.9 左右。这使得参数中那些梯度方向变化不大的维度可以加速更新，并减少梯度方向变化较大的维度上的更新幅度。由此产生了加速收敛和减小震荡的效果。(4) SGD with Nesterov(NAG)$v^t=\lambda v^{t-1}-\eta\nabla L(\theta^{t-1}+\lambda v^{t-1})\quad \theta^t=\theta^{t-1}+v^t$NAG 算法在 SGD-M 上进一步改进，计算 g`t 时有所不同。简单解释来说就是，在 SGD-M 算法中，更新参数要用到上一次更新的动量；于是考虑先进行λv^{t−1}的更新，再用新的θ计算梯度。$d_i=\beta d_{i-1}+g(\theta_{i-1})+\beta[g(\theta_{i-1})-g(\theta_{i-2})]\quad \theta_i=\theta_{i-1}-\alpha d_i$由等效，可以看出使用了二阶导信息，因此可以加速收敛。**给定实验结果，如何判断学习率是过大，还是过小？**当学习率设置的过小时，收敛过程将变得十分缓慢。而当学习率设置的过大时，梯度可能会在最小值附近来回震荡，甚至可能无法收敛。**学习率过大或过小，将带来什么影响？**学习率设置太小，网络收敛非常缓慢。学习率设置太大会造成网络不能收敛，在最优值附近徘徊。**类别不平衡问题解决方法** 1、扩大数据集，增加包含小类样本数据的数据，更多的数据能得到更多的分布信息。2、对大类数据欠采样减少大类数据样本个数，使与小样本个数接近。缺点：欠采样操作时若随机丢弃大类样本，可能会丢失重要信息。 代表算法：EasyEnsemble。其思想是利用集成学习机制，将大类划分为若干个集合供不同的学习器使用。相当于对每个学习器都进行欠采样，但对于全局则不会丢失重要信息。3、对小类数据过采样。过采样：对小类的数据样本进行采样来增加小类的数据样本个数。 代表算法：SMOTE 和 ADASYN。 SMOTE：通过对训练集中的小类数据进行插值来产生额外的小类样本数据。新的少数类样本产生的策略：对每个少数类样本 a，在 a 的最近邻中随机选一个样本 b，然后在 a、b 之间的连线上随机选一点作为新合成的少数类样本。 ADASYN：根据学习难度的不同，对不同的少数类别的样本使用加权分布，对于难以学习的少数类的样本，产生更多的综合数据。 通过减少类不平衡引入的偏差和将分类决策边界自适应地转移到困难的样本两种手段，改善了数据分布。4、使用新评价指标，如果当前评价指标不适用，则应寻找其他具有说服力的评价指标。比如准确度这个评价指标在类别不均衡的分类任务中并不适用，甚至进行误导。因此在类别不均衡分类任务中，需要使用更有说服力的评价指标来对分类器进行评价。5、选择新算法不同的算法适用于不同的任务与数据，应该使用不同的算法进行比较。6、数据代价加权，例如当分类任务是识别小类，那么可以对分类器的小类样本数据增加权值，降低大类样本的权值，从而使得分类器将重点集中在小类样本身上。7、转化问题思考角度，例如在分类问题时，把小类的样本作为异常点，将问题转化为异常点检测或变化趋势检测问题。 异常点检测即是对那些罕见事件进行识别。变化趋势检测区别于异常点检测在于其通过检测不寻常的变化趋势来识别。8、将问题细化分析，对问题进行分析与挖掘，将问题划分成多个更小的问题，看这些小问题是否更容易解决。

什么是卷积核，池化，特征图。卷积层中进行的处理叫卷积运算，相当于图像处理中的滤波器运算，用来作卷积运算的东西叫卷积核（也叫核、滤波器或者过滤器，每一次卷积操作则将卷积核上的每个元素和对应输入数据的元素相乘，最后把所有的结果相加，即为本次卷积的结果（一个绿色），然后将卷积核移动一个步长(stride)接着下一次卷积，直到卷积核遍历完输入数据即可。池化(pooling)是用来缩小数据尺寸的运算，池化操作也有一个类似卷积核一样东西在特征图上移动，书中叫它池化窗口，所以这个池化窗口也有大小，移动的时候有步长，池化前也有填充操作，每次移动池化窗口一般要计算的是窗口里的最大值(Max)或者平均值(Average)。有时将卷积层或者池化层的输入输出数据称为特征图(feature map)，输入数据称为输入特征图(input feature map)，输出数据称为输出特征图(output feature map) 。**为什么卷积核设计尺寸都是奇数？**保证像素点中心位置，避免位置信息偏移，填充边缘时能保证两边都能填充，原矩阵依然对称。**为什么要池化？**使用下采样的原因，一是减少需要处理的特征图的元素个数，二是通过让连续卷积层的观察窗口越来越大（即窗口覆盖原始输入的比例越来越大），从而引入空间过滤器的层级结构。**图像大小、步幅和卷积层的 Feature Map 大小的关系**$W_2=(W_1-F+2P)/S+1\backslash H_2=(H_1-F+2P)/S+1$。其中 W 2 ， 是卷积后 Feature Map 的宽度；W 1 是卷积前图像的宽度； F 是 filter 的宽度；P 是 Zero Padding 数量，Zero Padding 是指在原始图像周围补几圈 0，如果 P 的值是 1，那么就补 1 圈 0；S 是步幅；H 2 卷积后 Feature Map 的高度；H 1 是卷积前图像的宽度。举例：假设图像宽度 W 1 = 5 ，filter 宽度 F=3，Zero Padding P=0，步幅 S=2 则$W_2=(W_1-F+2P)/S+1=(5-3+0)/2+1=2$说明 Feature Map 宽度是 2。同样，我们也可以计算出 Feature Map 高度也是 2。**如何寻找超参数的最优值**超参数具体来讲比如算法中的学习率（learning rate）、梯度下降法迭代的数量（iterations）、隐藏层数目（hidden layers）、隐藏层单元数目、激活函数（ activation function）都需要根据实际情况来设置，这些数字实际上控制了最后的参数和的值，所以它们被称作超参数。在使用机器学习算法时，总有一些难调的超参数。例如权重衰减大小，高斯核宽度等等。这些参数需要人为设置，设置的值对结果产生较大影响。常见设置超参数的方法有：1. 猜测和检查：根据经验或直觉，选择参数，一直迭代。2. 网格搜索：让计算机尝试在一定范围内均匀分布的一组值。3. 随机搜索：让计算机随机挑选一组值。4. 贝叶斯优化：使用贝叶斯优化超参数，会遇到贝叶斯优化算法本身就需要很多的参数的困难。5. MITIE 方法，好初始猜测的前提下进行局部优化。它使用 BOBYQA 算法，并有一个精心选择的起始点。由于 BOBYQA 只寻找最近的局部最优解，所以这个方法是否成功很大程度上取决于是否有一个好的起点。在 MITIE 的情况下，我们知道一个好的起点，但这不是一个普遍的解决方案，因为通常你不会知道好的起点在哪里。从好的方面来说，这种方法非常适合寻找局部最优解。稍后我会再讨论这一点。6. 最新提出的 LIPO 的全局优化方法。这个方法没有参数，而且经验证比随机搜索方法好。**为什么要归一化** 1. 为了后面数据处理的方便，归一化的确可以避免一些不必要的数值问题。2. 为了程序运行时收敛加快。 3. 同一量纲。样本数据的评价标准不一样，需要对其量纲化，统一评价标准。这算是应用层面的需求。4. 避免神经元饱和。啥意思？就是当神经元的激活在接近 0 或者 1 时会饱和，在这些区域，梯度几乎为 0，这样，在反向传播过程中，局部梯度就会接近 0，这会有效地“杀死”梯度。5. 保证输出数据中数值小的不被吞食。**什么是批归一化（Batch Normalization）**以前在神经网络训练中，只是对输入层数据进行归一化处理，却没有在中间层进行归一化处理。要知道，虽然我们对输入数据进行了归一化处理，但是输入数据经过$\sigma(WX+b)$这样的矩阵乘法以及非线性运算之后，其数据分布很可能被改变，而随着深度网络的多层运算之后，数据分布的变化将越来越大。如果我们能在网络的中间也进行归一化处理，是否对网络的训练起到改进作用呢？答案是肯定的。 这种在神经网络中间层也进行归一化处理，使训练效果更好的方法，就是批归一化 Batch Normalization（BN）。**BN 算法的优点：** 1. 减少了人为选择参数。在某些情况下可以取消 dropout 和 L2 正则项参数，或者采取更小的 L2 正则项约束参数； 2. 减少了对学习率的要求。现在我们可以使用初始很大的学习率或者选择了较小的学习率，算法也能够快速训练收敛； 3. 可以不再使用局部响应归一化。BN 本身就是归一化网络(局部响应归一化在 AlexNet 网络中存在) 4. 破坏原来的数据分布，一定程度上缓解过拟合（防止每批训练中某一个样本经常被挑选到，文献说这个可以提高 1% 的精度）。 5. 减少梯度消失，加快收敛速度，提高训练精度。**为什么要正则化？** 1. 深度学习可能存在过拟合问题——高方差，有两个解决方法，一个是正则化，另一个是准备更多的数据，这是非常可靠的方法，但你可能无法时时刻刻准备足够多的训练数据或者获取更多数据的成本很高，但正则化通常有助于避免过拟合或减少你的网络误差。 2. 如果你怀疑神经网络过度拟合了数据，即存在高方差问题，那么最先想到的方法可能是正则化，另一个解决高方差的方法就是准备更多数据，这也是非常可靠的办法，但你可能无法时时准备足够多的训练数据，或者，获取更多数据的成本很高，但正则化有助于避免过度拟合，或者减少网络误差。**正则化注意事项**：正则项只有在训练过程中才会被加到损失函数，即训练过程中使用 L’（θ）来计算梯度，以更新参数。一旦训练完成，应该使用没有正则化的损失函数 L（θ）去测量评价模型的表现。偏置项（即常数项）没有正则化。**三种正则化方法：L2 正则化**，也叫岭回归（Ridge）$L(\theta)=L(\theta)+\lambda\cdot|\theta|_2,|\theta|_2=(w_1)^2+(w_2)^2+K$。**L1 正则化**，也叫 Lasso 回归$L(\theta)=L(\theta)+\lambda\cdot|\theta|_1,|\theta|_1=|w_1|+|w_2|+K$。**ElasticNet（ L1 + L2 ）**$L(\theta)=L(\theta)+\lambda\cdot[\rho\cdot|\theta|_1+(1-\rho)\cdot|\theta|_2]$，超参数：正则化因子 λ 。**什么是 dropout 正则化** 直观上理解：不要依赖于任何一个特征，因为该单元的输入可能随时被清除，因此该单元通过这种方式传播下去，并为单元的四个输入增加一点权重，通过传播所有权重，dropout 将产生收缩权重的平方范数的效果，和之前讲的 L2 正则化类似；实施 dropout 的结果实它会压缩权重，并完成一些预防过拟合的外层正则化；L2 对不同权重的衰减是不同的，它取决于激活函数倍增的大小。**dropout 有什么缺点？** dropout 一大缺点就是代价函数 J 不再被明确定义，每次迭代，都会随机移除一些节点，如果再三检查梯度下降的性能，实际上是很难进行复查的。定义明确的代价函数 J 每次迭代后都会下降，因为我们所优化的代价函数 J 实际上并没有明确定义，或者说在某种程度上很难计算，所以我们失去了调试工具来绘制这样的图片。我通常会关闭 dropout 函数，将 keep-prob 的值设为 1，运行代码，确保 J 函数单调递减。然后打开 dropout 函数，希望在 dropout 过程中，代码并未引入 bug。我觉得你也可以尝试其它方法，虽然我们并没有关于这些方法性能的数据统计，但你可以把它们与 dropout 方法一起使用。**图像可以用 CNN 的原因 1**pattern 具有局部性 2 图像的不同区域可能会出现的同样的 pattern3 对图像进行向下采样，对整体特征的影响不大，卷积层用到 a 和 b 两个原因，池化层用到 c 原因。同时，利用这些原理可以减少计算量。且，CNN 是稀疏连接，且共享参数，大大减少了计算量，CNN 可以很好地进行特征提取，再搭配全连接网络和 softmax 进行图像识别(分类),由于图像的特征具有局部性、重复性，因此可以通过卷积核进行特征提取。另外，降低图像分辨率对图像整体特征的影响较小，因此通过池化进行向下取样，减少计算量。综上，对比全连接，CNN 是稀疏连接，且共享参数，大大减少了计算量，又有很好的识别效果。**什么是序列数据?举例说明**。有时间维度的数据称为时间序列数据。如文本段落、语音输入、视频流、DNA 序列分析等。**1×1 的卷积的作用** 1.加入非线性。卷积层之后经过激励层，1×1 的卷积在前一层的学习表示上添加了非线性激励（ non-linear activation ），提升网络的表达能力； 实现跨通道的交互和信息整合 2. 进行卷积核通道数的降维和升维。如果卷积的输出输入都只是一个平面，那么 1x1 卷积核并没有什么意义，它是完全不考虑像素与周边其他像素关系。 但卷积的输出输入是长方体，所以 1x1 卷积实际上是对每个像素点，在不同的 channels 上进行线性组合（信息整合），且保留了图片的原有平面结构，调控 depth，从而完成升维或降维的功能。3 跨通道信息交互（channal 的变换）例子：使用 1*1 卷积核，实现降维和升维的操作其实就是 channel 间信息的线性组合变化，3*3，64channels 的卷积核后面添加一个 1*1，28channels 的卷积核，就变成了 3*3，28channels 的卷积核，原来的 64 个 channels 就可以理解为跨通道线性组合变成了 28channels，这就是通道间的信息交互。**RNN 特点**，当前的预测值考虑到了之前的运行结果。**即使只有一层的 RNN 模型仍可能出现梯度消失和梯度爆炸，为什么？** 对 RNN 进行优化需要用到 BPTT，用来表示 RNN 的记忆状态，权值的偏导中存在累乘，如果每一项都小于 1，那么乘多了就变 0 了，如果每一项都大于 1，那么乘多了又会很大，所以 RNN 存在梯度消失和爆炸的原因。**LSTM 与一般的 RNN 相比，优势在哪？**可以避免梯度消失（无法消除梯度爆炸）在 LSTM 中，也有和 RNN 一样的记忆单元，叫做细胞状态(LSTM Cell)。LSTM 的单元状态更新公式中是一个加法而不是乘法，因为是加法，所以不容易导致接近于 0 的情况。LSTM 可以保持长时记忆，LSTM 的记忆门可以控制记忆存放多久。不过 LSTM 可以保持长时间记忆根本原因也是因为 LSTM 解决了梯度消失的问题吧。训练 LSTM 相当于是训练每个 block 的三个门的输入权值。**对于给定问题，能判断出是否该使用 RNN 模型？**当输入和输出有一个是序列数据时使用 RNN 模型。典型 RNN 任务：语音识别、音乐生成、语义分析、机器翻译。**神经网络参数量的预估**，对于卷积神经网络，卷积需要参数=卷积核的三维相乘×卷积核个数+偏置（每个卷积核都有一个偏置）比如输入 32×32×3，卷积核为 5×5×3，有 6 个卷积核，那么参数为 5×5×3×6+6。池化层没有参数。全连接层 dense 层参数=（上一层输入的个数+1）×输出的个数。

计算题：梯度的计算反向传播算法损失函数，$L=\frac{1}{2}||a^L-y||_2^2$，我们约定，第 l-1 层有 m 个神经元，第 l 层有 n 个神经元，第 l+1 有 p 个神经元。 $a^l=\sigma(z^l)=\sigma(W^la^{l-1}+b^l)$。$\frac{\partial J(W,b,x,y)}{\partial w_{jk}^l}=\frac{\partial z^l}{\partial w_{jk}^l}\frac{\partial J(W,b,x,y)}{\partial z^l}$和$\frac{\partial J(W,b,x,y)}{\partial b^l}=\frac{\partial z^l}{\partial b^l}\frac{\partial J(W,b,x,y)}{\partial z^l}$

于是$\frac{\partial z^l}{\partial w_{jk}^l}=\left[\frac{\partial z_1^l}{\partial w_{jk}^l},\cdots,\frac{\partial z_j^l}{\partial w_{jk}^l},\cdots,\frac{\partial z_n^l}{\partial w_{jk}^l}\right]=\left[0,\cdots,\frac{\partial(w_j^la^{l-1}+b_j)}{\partial w_{jk}^l},\cdots,0\right]=[0,\cdots,a_k^{l-1},\cdots,0]$和$\frac{\partial z^l}{\partial b}=I_n$。我们注意到$\frac{\partial J(W,b,x,y)}{\partial z^l}=\frac{\partial z^{l+1}}{\partial z^l}\frac{\partial J(W,b,x,y)}{\partial z^{l+1}}=\frac{\partial z^{l+1}}{\partial z^l}\delta^{l+1}$。还有$\frac{\partial z^{l+1}}{\partial z^l}=\frac{\partial(W^{l+1}a^l+b^{l+1})}{\partial z^l}=diag\left(\sigma'(z^l)\right)(W^{l+1})^T$。于是$\frac{\partial J(W,b,x,y)}{\partial w_{jk}^l}=\frac{\partial z^l}{\partial w_{jk}^l}\frac{\partial J(W,b,x,y)}{\partial z^l}=$

$[0,\cdots,a_k^{l-1},\cdots,0][\delta_1^l,\cdots,\delta_j^l,\cdots,\delta_n^l]^T=\delta_j^la_k^{l-1}$。前式可进一步写为$\frac{\partial J(W,b,x,y)}{\partial W^l}\Big|_{jk}=\left[\delta^l(a^{l-1})^T\right]_{jk}$。所以最终可得$\frac{\partial J(W,b,x,y)}{\partial W^l}=\delta^l(a^{l-1})^T$和$\frac{\partial J(W,b,x,y)}{\partial b^l}=\delta^l$

给定项目案例，针对具体问题，进行相应解答。创建测试集时，通常有哪两类抽样方式？分层抽样和随机抽样。**如何判断哪种抽样方法更好？**评判依据：需避免抽样偏差，即测试集中数据的分布规律应与完整数据集是一致的。**我们需要对测试集中的样本进行研究吗？**不需要，避免透视偏差。**怎么探寻属性之间的相关性？**用相关性矩阵或者散点图矩阵。**怎么处理缺失值？** 1. 缺失值较多，直接将该特征舍弃掉，否则可能反倒会带入较大的 noise，对结果造成不良影响。2. 缺失值较少，其余的特征缺失值都在 10%以内，我们可以采取很多的方式来处理：1) 把 NaN 直接作为一个特征，假设用 0 表示。只要 0 不是一个有意义的值。网络能够从数据中学到 0 意味着缺失数据，并且会忽略这个值；2) 用均值填充；3) 用随机森林等算法预测填充。**为什么要将属性的文本类型转换为数值类型的？**便于计算机处理。**样本不均衡时，评估指标是什么？**样本不均衡时，使用 AUC 作为评估指标，评估性能非常接近其实际性能，评估很准确！经验法则：当正例很少，或者当你关注假正例多于假反例的时候，优先使用 PRC 其他情况使用 ROC 曲线。**影响模型复杂度的超参数？**也就是模型定义中自带的超参数，正则化因子；修正了损失函数的定义。注意； 正则化模型的学习曲线的误差计算是基于未添加正则项的损失函数。**影响自动化训练过程的超参数？**1 学习率 2 训练前是否需要对特征进行标准化或归一化 3 SGD/BGD/MBGD。**如何选择衡量成功的指标？**对于平衡分类问题（每个类别的可能性相同），精度和 AUC 是常用的指标 。对于类别不平衡的问题，你可以使用准确率和召回率。准确率和召回率，对样本不均衡敏感。对于排序问题或多标签分类，你可以使用平均准确率均值（ mean average precision）。**三种常见的评估方法：** 留出验证集 。数据量很大时可以采用这种方法 。K 折 交叉验证 。如果留出验证的样本量太少，无法保证可靠性，那么应该选择这种方法。重复的 K 折验证 ：多次使用 K 折交叉验证。如果可用的数据很少， 同时模型评估又需要非常准确，那么应该使用重复的 K 折验证。**为什么要进行 Batch Normalization** 在网络训练过程中 后一层网络的输入是前一层的输出 因此前一层网络参数的变化将导致后一层输入数据分布的改变且这种改变会在训练过程中向后传递并被逐步放大；这种在训练过程中数据分布的改变称为 Internal Covariate Shift 神经网络的训练目标是在输出层得到原始输入数据数据分布的一个映射，即在训练过程中应该力图保证数据分布的映射关系：若数据分布在训练过程中发生了偏移则会降低网络的泛化能力在 Batch based Training 中 每个 Batch 的分布各不相同网络需在每个 Batch 的训练中适应不同的分布从而大大降低训练速度。针对每层网络的每个 Batch 进行 Batch Normalization. 可有效提高训练速度和 效果。通常在激活函数之前，先进行 Batch Normalization。**如何解决训练样本少的问题**目前大部分的深度学习模型仍然需要海量的数据支持。例如 ImageNet 数据就拥有 1400 多万的图片。而现实生产环境中，数据集通常较小，只有几万甚至几百个样本。这时候，如何在这种情况下应用深度学习呢？（1）利用预训练模型进行迁移微调（fine-tuning），预训练模型通常在特征上拥有很好的语义表达。此时，只需将模型在小数据集上进行微调就能取得不错的效果。这也是目前大部分小数据集常用的训练方式。视觉领域内，通常会 ImageNet 上训练完成的模型。自然语言处理领域，也有 BERT 模型等预训练模型可以使用。（2）单样本或者少样本学习（one-shot，few-shot learning），这种方式适用于样本类别远远大于样本数量的情况等极端数据集。例如有 1000 个类别，每个类别只提供 1-5 个样本。少样本学习同样也需要借助预训练模型，但有别于微调的在于，微调通常仍然在学习不同类别的语义，而少样本学习通常需要学习样本之间的距离度量。例如孪生网络（Siamese Neural Networks）就是通过训练两个同种结构的网络来判别输入的两张图片是否属于同一类。上述两种是常用训练小样本数据集的方式。此外，也有些常用的手段，例如数据集增强、正则或者半监督学习等方式来解决小样本数据集的训练问题。**自动化超参数搜索方法有哪些？**目前自动化搜索主要包含网格搜索，随机搜索，基于模型的超参优化。网格搜索：通常当超参数量较少的时候，可以使用网格搜索法。即列出每个超参数的大致候选集合。利用这些集合进行逐项组合优化。在条件允许的情况下，重复进行网格搜索会当优秀，当然每次重复需要根据上一步得到的最优参数组合，进行进一步的细粒度的调整。网格搜索最大的问题就在于计算时间会随着超参数的数量指数级的增长。随机搜索：随机搜索，是一种用来替代网格搜索的搜索方式。随机搜索有别于网格搜索的一点在于，我们不需要设定一个离散的超参数集合，而是对每个超参数定义一个分布函数来生成随机超参数。随机搜索相比于网格搜索在一些不敏感超参上拥有明显优势。例如网格搜索对于批样本数量（batch size），在[16,32,64]这些范围内进行逐项调试，这样的调试显然收益更低下。当然随机搜索也可以进行细粒度范围内的重复的搜索优化。**为什么需要激活函数？** 1. 激活函数对模型学习、理解非常复杂和非线性的函数具有重要作用。2. 激活函数可以引入非线性因素。如果不使用激活函数，则输出信号仅是一个简单的线性函数。线性函数一个一级多项式，线性方程的复杂度有限，从数据中学习复杂函数映射的能力很小。没有激活函数，神经网络将无法学习和模拟其他复杂类型的数据，例如图像、视频、音频、语音等。3. 激活函数可以把当前特征空间通过一定的线性映射转换到另一个空间，让数据能够更好的被分类。**为什么激活函数需要非线性函数？** 1. 假若网络中全部是线性部件，那么线性的组合还是线性，与单独一个线性分类器无异。这样就做不到用非线性来逼近任意函数。2. 使用非线性激活函数 ，以便使网络更加强大，增加它的能力，使它可以学习复杂的事物，复杂的表单数据，以及表示输入输出之间非线性的复杂的任意函数映射。使用非线性激活函数，能够从输入输出之间生成非线性映射。**常见的激活函数：**$f(x)=1/1+e^{-x}\quad f(x)=tanh(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}$。f(x) = max(0, x) softmax 函数 $\sigma(z)_i=e^{z_j}/\sum_{k=1}^K e^{z_k}$。**常见的优化器 1** 梯度下降法(Gradient Descent) 1. 标准梯度下降法(GD)2. 批量梯度下降法(BGD) 3.随机梯度下降法(SGD)。**2** 动量优化法 1. Momentum 使用动量(Momentum)的随机梯度下降法(SGD)，主要思想是引入一个积攒历史梯度信息动量来加速 SGD。2. NAG 加速梯度（NAG，Nesterov accelerated gradient）算法，是 Momentum 动量算法的变种。**3** 自适应学习率优化算法。1.AdaGrad 算法思想：AdaGrad 算法，独立地适应所有模型参数的学习率，缩放每个参数反比于其所有梯度历史平均值总和的平方根。具有代价函数最大梯度的参数相应地有个快速下降的学习率，而具有小梯度的参数在学习率上有相对较小的下降。$W_{t+1}=W_t-\eta_0/\sum_{i=0}^t\left(g_{t,i}\right)+\epsilon\odot g_{t,i}$。2.RMSProp 算法思想：RMSProp 算法修改了 AdaGrad 的梯度积累为指数加权的移动平均，使得其在非凸设定下效果更好。3. AdaDelta 算法思想：AdaGrad 算法和 RMSProp 算法都需要指定全局学习率，AdaDelta 算法结合两种算法每次参数的更新步长。4. Adam 算法思想：首先，Adam 中动量直接并入了梯度一阶矩（指数加权）的估计。其次，相比于缺少修正因子导致二阶矩估计可能在训练初期具有很高偏置的 RMSProp，Adam 包括偏置修正，修正从原点初始化的一阶矩（动量项）和（非中心的）二阶矩估计。**只要使用深度神经网络，就无须担心特征工程呢？**并不是这样，原因有两点。良好的特征仍然可以让你用更少的资源更优雅地解决问题。例如，使用卷积神经网络来读取钟面上的时间是非常可笑的。良好的特征可以让你用更少的数据解决问题。深度学习模型自主学习特征的能力依赖于大量的训练数据。如果只有很少的样本，那么特征的信息价值就变得非常重要。**长尾数据怎么处理？**回归问题中，可以对标签取对数，使标签接近于正太分布。**线性分类器与非线性分类器的区别以及优劣。**如果模型是参数的线性函数，并且存在线性分类面，那么就是线性分类器，否则不是。常见的线性分类器有：LR，贝叶斯分类，单层感知机、线性回归。常见的非线性分类器：决策树、RF、GBDT、多层感知机 SVM 两种都有（看线性核还是高斯核）线性分类器速度快、编程方便，但是可能拟合效果不会很好非线性分类器编程复杂，但是效果拟合能力强。**选择线性还是非线性分类器？**维度高的时候，数据一般在维度空间里面会比较稀疏，很有可能线性可分。因为低维空间可能很多特征都跑到一起了，导致线性不可分。**如何进行特征选择？**特征选择是一个重要的数据预处理过程，主要有两个原因：一是减少特征数量、降维，使模型泛化能力更强，减少过拟合；二是增强对特征和特征值之间的理解常见的特征选择方式：1 去除方差较小的特征 2 正则化。1 正则化能够生成稀疏的模型。L2 正则化的表现更加稳定，由于有用的特征往往对应系数非零。3 随机森林，对于分类问题，通常采用基尼不纯度或者信息增益，对于回归问题，通常采用的是方差或者最小二乘拟合。一般不需要 feature engineering、调参等繁琐的步骤。它的两个主要问题，1 是重要的特征有可能得分很低（关联特征问题），2 是这种方法对特征变量类别多的特征越有利（偏向问题）。4 稳定性选择。是一种基于二次抽样和选择算法相结合较新的方法，选择算法可以是回归、SVM 或其他类似的方法。它的主要思想是在不同的数据子集和特征子集上运行特征选择算法，不断的重复，最终汇总特征选择结果，比如可以统计某个特征被认为是重要特征的频率（被选为重要特征的次数除以它所在的子集被测试的次数）。理想情况下，重要特征的得分会接近 100%。稍微弱一点的特征得分会是非 0 的数，而最无用的特征得分将会接近于 0。**如何数据预处理** 1 缺失值，填充缺失值 fillna；离散：None,连续：均值。缺失值太多，则直接去除该列。2 连续值：离散化。有的模型（如决策树）需要离散值。3 对定量特征二值化。核心在于设定一个阈值，大于阈值的赋值为 1，小于等于阈值的赋值为 0。如图像操作。4 皮尔逊相关系数，去除高度相关的列。5 去除噪声点。通过 matplotlib.scatter 函数图示某一特征与预测特征的点分布图，明显看出噪声点，去除即可。6 标签编码，把字符类别特征编码成数值类型，如红绿蓝编码为 0、1、2。7 归一化。将数据按比例缩放，使这些数据落入到一个较小的特定的区间之内。最小最大缩放和 b.Z-score 标准化。让数据服从基于 μ=0 和 σ=1 的标准正态分布 8 OneHot 编码 get_dummies。有的离散特征如颜色需 OneHot 编码，编码后需特征对齐，因为进行 one-hot 编码后，会出现一种情况就是：某个特征的某一个取值只出现在训练集中，没有出现在测试集中，或者相反。9 产生 K 折交叉数据集：sklearn.cross_validation.KFold。10 填充均值前，通过对数运算修正数据基本符合正态分布。**梯度消失** $\partial C/\partial w_1=\partial C/\partial a_n\cdot\partial a_n/\partial z_n\cdot\partial z_n/\partial a_{n-1}\cdots\partial a_1/\partial z_1\cdot\partial z_1/\partial w_1=\partial C/\partial a_n\cdot s(z_n)\cdot w_n\cdot s(z_{n-1})\cdot w_{n-1}\cdots s(z_2)\cdot w_2\cdot s(z_1)\cdot x$。如果我们使用标准方法来初始化网络中的权重，那么会使用一个均值为 0 标准差为 1 的高斯分布。因此所有的权重通常会满足|w j|<1，而 s ‘是小于 0.25 的值，那么当神经网络特别深的时候，梯度呈指数级衰减，导数在每一层至少会被压缩为原来的 1/4，当 z 值绝对值特别大时，导数趋于 0，正是因为这两个原因，从输出层不断向输入层反向传播训练时，导数很容易逐渐变为 0，使得权重和偏差参数无法被更新，导致神经网络无法被优化，训练永远不会收敛到良好的解决方案。 这被称为梯度消失问题。**梯度爆炸**当我们将 w 初始化为一个较大的值时，例如>10 的值，那么从输出层到输入层每一层都会有一个 s ‘(zn)*wn 的增倍，当 s ‘(zn)为 0.25 时 s ‘(zn)*wn>2.5，同梯度消失类似，当神经网络很深时，梯度呈指数级增长，最后到输入时，梯度将会非常大，我们会得到一个非常大的权重更新，这就是梯度爆炸的问题，在循环神经网络中最为常见.**解决方案** 1 好的参数初始化方式，如 He 初始化 2 非饱和的激活函数（如 ReLU）3 批量规范化（Batch Normalization）4 梯度截断（Gradient Clipping）5 更快的优化器 6LSTM7 残差结构。