1. **有哪些解决回归问题的方法？**

回归问题的目标是预测一个连续值，常见方法包括：

* 传统回归方法
  + 线性回归（Linear Regression）：适用于线性关系，易解释但表达能力有限。
  + 岭回归（Ridge Regression）和Lasso回归：在普通线性回归基础上加入正则项，防止过拟合。
  + 多项式回归：通过引入高阶项拟合非线性关系。
* 树模型
  + 决策树回归（Decision Tree Regression）
  + 随机森林回归（Random Forest Regression）
  + 梯度提升树（GBDT、XGBoost、LightGBM、CatBoost）：在广告算法中非常常见，能处理高维稀疏特征，支持特征重要性分析。

1. **决策树**

图示

AI 生成的内容可能不正确。

常见决策树：CART（分类和回归树）

**（1）CART**在分类和回归问题中的异同

* 相同：
  + 都是二叉树
  + 所有落在同一片叶子中的输入都有相同的输出
* 差异：
  + 分类问题中：
    - 使用**基尼系数**作为选择特征和划分的依据
    - 每一片叶子都代表一个class
  + 回归问题中：
    - 使用**MSE均方误差**和**MAE绝对平均误差**作为选择特征和划分的依据
      * **MSE均方误差：**能更敏感地捕捉大的偏差；在数学上具有良好的性质：一阶导数和二阶导数都存在，便于优化；适合连续值预测。
      * **MAE绝对平均误差**：对离群点不敏感，鲁棒性更强；更适合数据中存在异常值的情况。
    - 每一片叶子表示一个预测值，取值连续

**信息增益**：在给定某个特征的条件下，信息的不确定性减少的程度（越大越好）

**基尼系数**：从数据集随机抽取两个样本，其类别标记不同的概率（越小纯度越高）

**CART过程：**

* 对每个特征的每个可能取值进行划分。
* 计算划分后左右子集的误差。
* 找出使总误差（例如 left\_MSE + right\_MSE）最小的划分点。
* 递归进行，直到满足停止条件（如最大深度、最小样本数等）
* 完整构建CART树之后，再进行剪枝（后剪枝）

**（2）ID3算法**：使用信息增益来选择最优特征，使用贪心算法一直选择当前最优的特征进行分割，直到数据集分割完成为止

**ID3构建步骤：**

* 计算信息熵 Ent(D)：衡量当前数据集的纯度
* 计算特征A的条件熵 Ent\_A(D)：将数据集按特征A的取值划分为若干子集，计算每个子集的熵并加权平均：
* 计算信息增益 Gain(D, A)：表示使用特征A划分后信息不确定性减少的程度。
* 选择信息增益最大的特征作为划分节点
* 递归构建子树：对每个子集重复上述过程，直到满足停止条件（如所有样本属于同一类或特征用尽）。

**（3）C4.5算法：**使用信息增益比来选择最优特征，能处理连续值和缺失值

* + 连续值处理：使用信息增益率（Gain Ratio）进行二分分裂，先找到最佳分裂点再进行计算，并在计算中修正增益以避免连续特征的偏好；
  + 缺失值处理：对于缺失值，C4.5为其分配概率权重，参与到信息增益的计算中

**C4.5构建步骤：**

* 处理连续特征：将连续特征按值排序，计算所有可能的划分点（中间值）。每个划分点将连续特征转化为布尔特征（如 x ≤ t）。
* 计算信息增益率 GainRatio(D, A)：首先计算信息增益 Gain(D, A)，然后计算分裂信息 SplitInfo(A)，最后计算信息增益率：
* 选择信息增益率最大的特征作为划分节点
* 剪枝处理：使用悲观剪枝（pessimistic pruning）策略。使用统计方法评估子树在验证集上的准确率，若剪枝后准确率提升，则将子树替换为叶节点。
* 递归构建子树，直到满足停止条件

悲观剪枝和乐观剪枝

| **特性** | **悲观剪枝（PEP）** | **乐观剪枝** |
| --- | --- | --- |
| 是否需要验证集 | 否 | 是（或交叉验证） |
| 剪枝标准 | 统计估计误差 | 实际误差比较 |
| 剪枝倾向 | 保守 | 激进 |
| 适用场景 | 数据量小、无验证集 | 数据量大、有验证集 |

表格

AI 生成的内容可能不正确。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

1. **决策树回归优缺点**

* 优点：
  + 易于理解和解释
  + 能处理分线性关系，无需对数据进行线性假设
  + 对数据的缺失值不敏感
* 缺点：
  + 特别是在数据量较小或树的深度较大时，容易过拟合
  + 不稳定性，树的小变化会导致树的结构显著改变
  + 难以处理连续性特征，需进行离散化处理，损失部分信息

1. **决策树剪枝（剪枝时考虑全局最优）**

* 预剪枝：构建过程中
* 后剪枝：先完整构建、后剪枝

1. **梯度提升树GBDT**

每一棵树学的是之前所有树的结论和残差，残差加一个预测之后能得真实值的累加量

损失函数：平均误差损失函数

图示

AI 生成的内容可能不正确。

1. **XGBoost**

使用二阶泰勒展开近似损失函数，引入正则项控制模型复杂度，支持缺失值处理，使用列采样和行采样提升泛化能力，支持并行构建树结构。

目标函数：误差函数+L2正则化。 误差函数用来说明模型有多拟合数据，正则化用来惩罚复杂模型

* L1正则化：在损失函数中加入**权重向量的绝对值之和**作为惩罚项。
  + 产生稀疏矩阵，**不重要的特征权重会变为0**；
  + 可用于特征选择，帮助提升模型可解释性。
  + 为模型权重设置了拉普拉斯分布的先验，在0点附近的概率密度较大
* L2正则化：在损失函数中加入**权重向量的平方和**作为惩罚项。
  + 使较大的权重值变小，防止模型过拟合
  + 模型权重趋于平滑，但不是完全为0
  + 为模型权重设置了高斯分布的先验，**倾向于产生较小的、非零的权重值**

**7. 广告算法中回归模型的典型应用场景**

（1）广告投放与销售额预测

目标：通过历史广告投放费用预测未来销售额。

方法：使用线性回归模型建立广告费用（如电视、广播、报纸）与销售额之间的关系。实际应用：某公司通过分析200个市场的广告费用与销售额数据，发现电视广告对销售额影响最大（相关系数0.78），报纸广告影响最弱（0.23），从而优化预算分配。

（2）广告点击率（CTR）预测

目标：预测用户点击广告的概率。

方法：使用逻辑回归或广义线性模型（GLM）对CTR进行建模。

特征：用户行为、广告位、时间段、设备类型等。

实际应用：在推荐系统中，CTR预测是排序模型的重要组成部分，常与深度学习模型（如Wide & Deep、DeepFM）结合使用。

（3）视频观看时长建模

目标：预测用户观看视频的时长，用于广告插入、内容推荐。

方法：传统回归模型（如Lasso、GBDT）预测连续值。分段建模（CREAD模型）将时长离散化为区间，分类预测后恢复连续值。因果建模（D2Q模型）解决duration bias问题。

实际应用：Kuaishou在KDD 2022提出D2Q模型，通过因果图识别曝光路径和观看路径的双重影响，提升推荐效果。

（4）广告预算分配与ROI预测

目标：预测不同渠道广告投放的ROI（投资回报率）。

方法：多元线性回归或加权回归模型。

1. **MSE前提一定是要满足正态分布吗**

MSE（均方误差，Mean Squared Error），用于衡量模型预测值与真实值之间的差异。

MSE的使用不依赖正态分布，是一种纯粹的误差度量方法，它本身不依赖于数据分布的任何假设。

1. **长尾数据分布，多峰分布怎么应对？**
2. **长尾数据**

长尾分布指的是少数类别（头部）拥有大量样本，而多数类别（尾部）样本稀少，导致模型偏向头部类别，忽略尾部信息。

* 数据层面的处理方法
  + 重采样（Resampling）：
    - 过采样：增加尾部类别样本数量，如使用 SMOTE、ADASYN 等合成方法。
    - 欠采样：减少头部类别样本数量，避免模型过拟合头部。
  + 数据增强（Data Augmentation）：
    - 图像领域可用旋转、裁剪、翻转等方式；
    - 文本领域可用同义词替换、句式重组等。
  + 迁移学习（Transfer Learning）：利用在大规模数据集上预训练的模型，在尾部类别上微调，提升泛化能力。
  + 主动学习（Active Learning）：优先标注尾部类别样本，提高其在训练中的权重。
* 模型层面的处理方法
  + 类别重加权（Class Reweighting）：在损失函数中对尾部类别赋予更高权重，提升其学习效果。
  + 混合模型（Hybrid Models）：使用多个模型分别处理头部和尾部类别，再融合结果。
  + 特征工程与空间扩展：通过 PCA、LDA 等方法提取更具区分性的特征，或使用核方法、深度学习扩展特征空间。

**（2）多峰分布**

多峰分布指的是数据在多个区域集中，可能对应多个子群体或行为模式。

* **分布建模**：使用混合高斯模型（GMM）或变分自编码器（VAE）建模多峰结构。
* **分群建模（Cluster-wise Modeling）**：先聚类，再对每个簇单独建模，适用于用户行为或商品属性明显分层的场景。

1. **梯度爆炸和梯度消失**

* 原因：
  + 不合适的网络结构：链式求导中，对激活函数求导结果若>1，则梯度的更新将以指数形式增长，发生梯度爆炸；否则，随着层数的增加求导的信息会以指数形式递减，发生梯度消失。
  + 不合适的激活函数：如Logistic函数和Tanh函数

图表, 直方图

AI 生成的内容可能不正确。

* 解决方法
  + 改变激活函数﻿：使用ReLU、Leaky ReLU、ELU等非饱和激活函数，其在正区间的导数为1，可以有效缓解梯度消失。
  + 改进模型结构
    - 残差连接 (ResNet)：残差网络通过“shortcut”（捷径）连接，允许梯度直接流过层，确保一个常数梯度，有效解决梯度消失。﻿
    - 批量归一化 (Batch Normalization)：对每层激活函数的输入进行缩放和偏移，使梯度值更加稳定，从而缓解梯度消失和爆炸。﻿
    - 门控循环单元(如LSTM)：在循环神经网络中引入遗忘、输入、输出等门控机制，控制信息的流动，有效解决长期依赖问题中的梯度消失和爆炸。﻿
  + 优化参数和训练过程
    - 正则化：L1、L2正则化，可以通过约束权重的取值范围，防止权重过大，从而限制梯度爆炸。﻿
    - 梯度裁剪：为梯度设置一个阈值，当梯度值超过该阈值时，将其强制缩放到指定范围内，直接防止梯度爆炸。﻿
    - 调整学习率：对于梯度消失问题，可以适当增大学习率；对于梯度爆炸，可以减小学习率。﻿
  + 预训练加微调﻿
    - 预训练（Pre-training）：先对网络进行逐层无监督训练，找到一个较好的局部最优解。
    - 微调（Fine-tuning）：之后再对整个网络进行有监督的微调，整合预训练阶段的学习结果，可以更好地解决梯度问题。

1. **激活函数**

线性转为非线性，缓解梯度消失问题，将特征输入映射到新的特征空间以及加速模型收敛。

图示

AI 生成的内容可能不正确。

* **Sigmoid**
  + 优点
    - 压缩输入特征
    - 适用于将预测概率作为输出的模型
  + 缺点
    - 输入非常大/小时，输出基本为常数，导致梯度接近0——梯度消失
    - 输出不是0均值，进而导致后一层神经元将得到上一层输出的非0均值的信号作为输入。随着网络的加深，会改变原始数据的分布趋势；
* **Tanh**
  + 优点
    - 导数取值在0~1，一定程度缓解梯度消失
    - 在原点附近与y=x函数形式相近，当输入的激活值较低时，可以直接进行矩阵运算，训练相对容易
  + 缺点
    - 梯度消失、幂运算
* **Relu**
  + 优点
    - 在输入为正时，Relu函数不存在饱和问题，即解决了gradient vanishing问题，使得深层网络可训练；
    - 计算速度非常快，只需要判断输入是否大于0值；
    - 收敛速度远快于sigmoid以及Tanh函数；
    - Relu输出会使一部分神经元为0值，在带来网络稀疏性的同时，也减少了参数之间的关联性，一定程度上缓解了过拟合的问题
  + 缺点
    - 某些神经元可能永远不会被激活，进而导致相应参数一直得不到更新，产生该问题主要原因包括参数初始化问题以及学习率设置过大问题；
    - 当输入为正值，导数为1，在“链式反应”中，不会出现梯度消失，但梯度下降的强度则完全取决于权值的乘积，如此可能会导致梯度爆炸问题；
* **Leaky Relu**
  + 优点
    - 针对Relu函数中存在的Dead Relu Problem，Leaky Relu函数在输入为负值时，给予输入值一个很小的斜率，在解决了负输入情况下的0梯度问题的基础上，也很好的缓解了Dead Relu问题；
    - 该函数的输出为负无穷到正无穷，即leaky扩大了Relu函数的范围，其中α的值一般设置为一个较小值，如0.01；
  + 缺点：
    - 效果不稳定
    - 由于在不同区间应用的不同的函数所带来的不一致结果，将导致无法为正负输入值提供一致的关系预测。
* **ELU**
  + 优点
    - ELU具有Relu的大多数优点，不存在Dead Relu问题，输出的均值也接近为0值；
    - 该函数通过减少偏置偏移的影响，使正常梯度更接近于单位自然梯度，从而使均值向0加速学习；
    - 该函数在负数域存在饱和区域，从而对噪声具有一定的鲁棒性；
  + 缺点
    - 效果并不稳定
    - 幂运算，计算度高

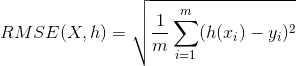
1. **优化器**

更新模型参数以最小化训练误差的算法。它可以将损失函数的梯度作为输入，并根据该梯度调整模型参数的值

* **SGD**
* **Adam**
* **AdamW**
* **AdaGrad**

1. **回归的评价指标有哪些？**

* **RMSE（Root Mean Square Error）均方根误差**：衡量观测值与真实值之间的偏差



* **MSE（Mean Square Error）均方误差**：真实值与预测值的差值的平方然后求和平均。通过平方的形式便于求导，所以常被用作线性回归的损失函数

在这里插入图片描述

* **MAE（Mean Absolute Error）平均绝对误差**：是绝对误差的平均值。可以更好地反映预测值误差的实际情况。

在这里插入图片描述

* **R-squared可决系数**(coefficient of determination)也叫拟合优度，反映的是自变量x对因变量y的变动的解释的程度.越接近于1,说明模型拟合得越好

衡量观测值与真实值之间的偏差。

1. **ROC，AUC含义？**

**（1）ROC** 曲线是以假正例率（FPR）为横轴，真正例率（TPR）为纵轴绘制的曲线。

* TPR（True Positive Rate）：真正例率，也叫召回率

* FPR（False Positive Rate）：假正例率

ROC 曲线展示了模型在不同阈值下的分类能力。每一个点对应一个分类阈值。

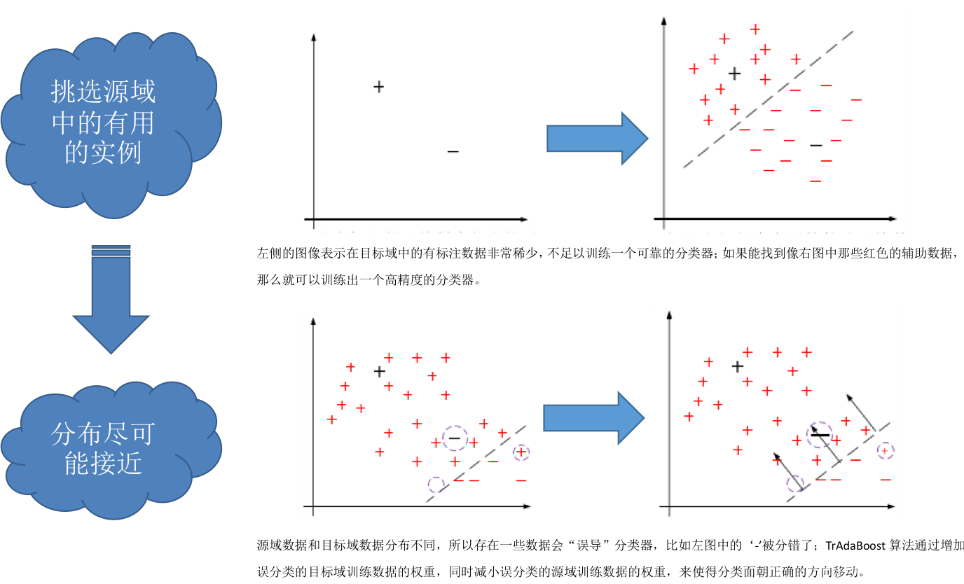
**（2）AUC** 是 ROC 曲线下的面积，取值范围为 [0, 1]，表示模型将正样本排在负样本前的概率。

* AUC = 1：完美分类器
* AUC = 0.5：随机猜测
* AUC < 0.5：模型表现比随机还差（可能是正负样本标签反了）

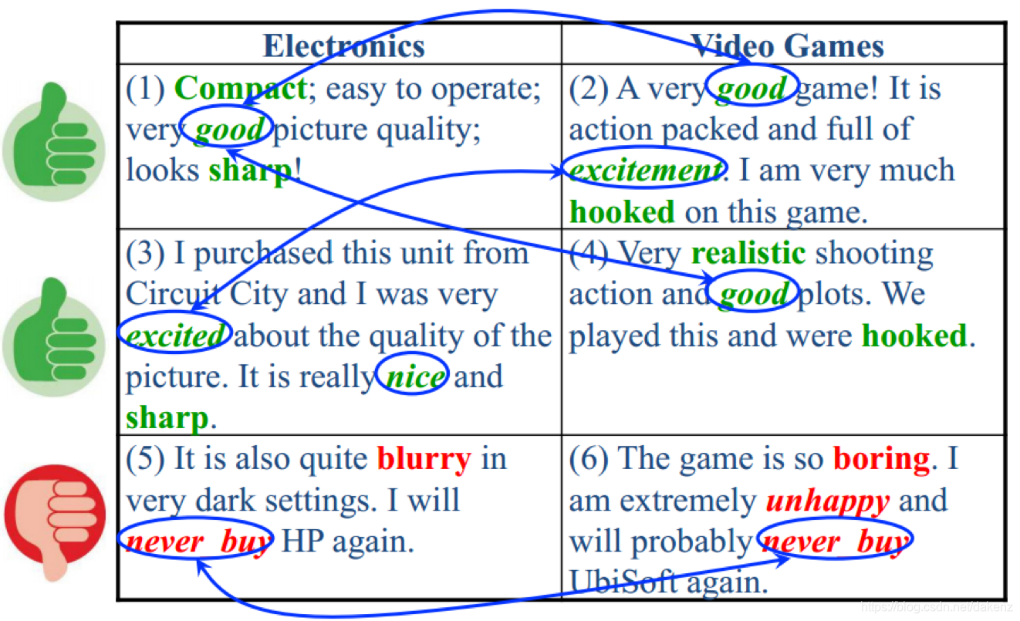
| **优点** | **说明** |
| --- | --- |
| 与阈值无关 | 不依赖于具体分类阈值，适合评估模型整体性能 |
| 抗样本不均衡 | 不受正负样本比例影响，适合长尾数据 |
| 可视化直观 | ROC 曲线越靠近左上角，模型越好 |

1. **迁移学习**

* 目标：如何充分利用之前标注好的数据（废物利用），同时又保证在新的任务上的模型精度。将某个领域或任务上学习到的知识或模式应用到不同但相关的领域或问题中。
* 两个关键概念：域（某个时刻的某个特定领域）和任务
* 用什么迁移：不同领域之间有哪些共有知识可以迁移
* 如何迁移：基于实例、基于特征、基于共享参数
  + **基于实例的迁移**：从源领域中选出对目标领域的训练有用的实例进行有效权重分配，使其接近目标域的实例分配。
    - TrAdaBoost：增加误分类的目标域训练的权重，同时减小误分类的源训练域的数据权重，来使分类面向正确方向移动



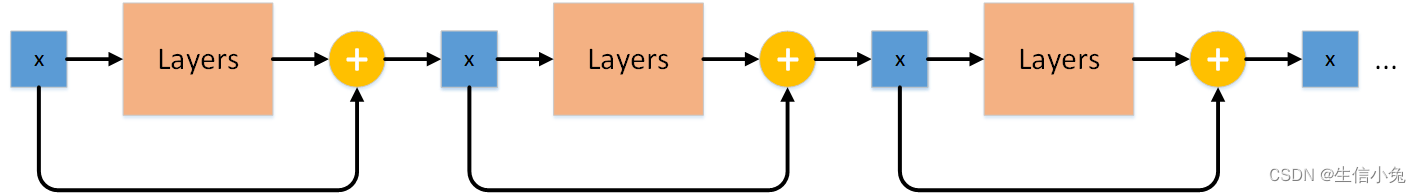
* + **基于特征的迁移**：关注的是如何将源领域和目标领域的数据从原始特征空间映射到新的特征空间中去。在该空间中，源领域数据与的目标领域的数据分布相同，从而可以在新的空间中，更好地利用源领域已有的有标记数据样本进行分类训练，最终对目标领域的数据进行分类测试。



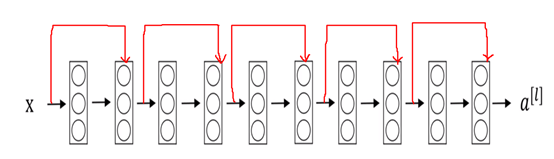
* + **基于共享参数的迁移**：如何找到源数据和目标数据的空间模型之间的共同参数或者先验分布，从而可以通过进一步处理，达到知识迁移的目的，假设前提是，学习任务中的的每个相关模型会共享一些相同的参数或者先验分布。
* 何时迁移：领域间有公共知识结构

1. **残差网络：**缓解神经网络中增加深度带来的梯度消失和梯度爆炸的问题，可以训练很深的网络。

残差块：把某层输入跳跃到下一层或者更深层的激活层之前，同本层输出一起经过激活函数输出。输出 = 网络的输出 + 输入



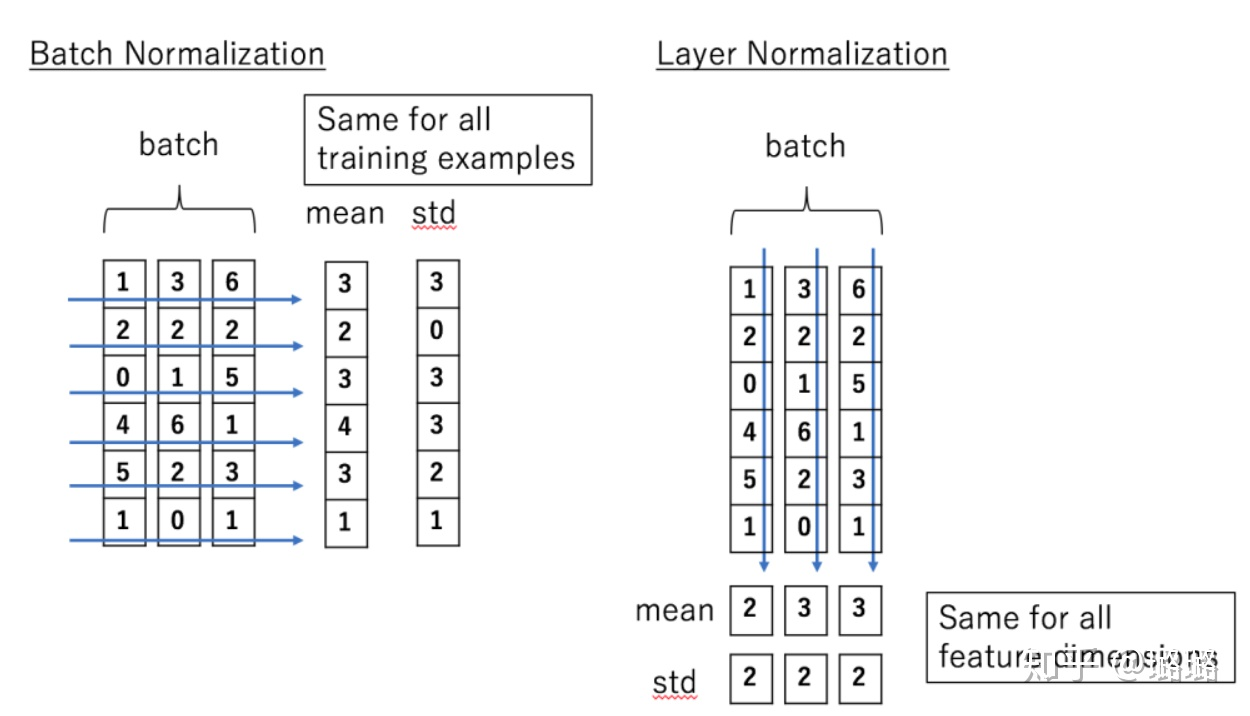
跳跃连接，可以有效缓解梯度消失的问题：



1. **Batch Normalization 和 Layer Nomination**

核心作用是加速训练和收敛，也可以防止过拟合

因为每一个mini-batch的内部数据的均值和方差都有一定差别，相当于在train过程中引入了噪声和随机性；但是如果batch\_size足够大，方差和均值几乎不变，就无法防止过拟合。一啊不能是8的倍数，不会低于16。太小难收敛



Layer Normalization适合NLP任务：归一化处理后，一句话内每个字之间的联系并没有被破坏

Batch Normalization适合CV任务，但不能和Dropout一起使用。

原因：

* 当网络的状态从训练转移到测试时，Dropout转移了特定神经单元的方差。但是，在测试阶段，BN保持了它的统计方差，这是在整个学习过程中积累的。Dropout和BN中方差的不一致性(命名为“方差偏移”)导致推断中不稳定的数值行为，最终导致错误的预测。
* 对于下图，展示了BN和Dropout的训练和测试区别，Dropout在训练直接以p的概率进行失活，而在测试，对每个神经元进行尺度放缩(乘p)。另一种等价的表现形式，在训练阶段乘1/p，而测试阶段不需要做任何改动，所以，训练：X=a\*(1/p)\*X，测试：X=X，存在方差偏移。

