简介

以各个大厂为维度，整理经典算法面试题，仅供参考，如有错误，欢迎指正！！ by 猫先生

目录

1.携程推荐算法面试题8道

问题1：讲一讲推荐系统包含哪些流程?

推荐系统的流程通常包括以下几个步骤：

- **数据收集**：收集用户行为数据（如浏览记录、购买记录、点击记录等）和物品数据（如物品特征、分类、标签等）。
- **数据预处理**：对数据进行清洗、归一化、特征提取等预处理操作。
- **特征工程**：构建用户画像和物品画像，提取有助于推荐的特征。
- **模型选择**：选择合适的推荐算法，如基于内容的推荐、协同过滤、矩阵分解、深度学习等。
- **模型训练**：使用历史数据训练推荐模型。
- **推荐生成**：根据训练好的模型生成推荐列表。
- **评估与调优**：使用评价指标（如准确率、召回率、F1-score等）评估推荐效果，并进行模型调优。
- **上线与更新**：将推荐系统上线，并定期更新模型和数据。

问题2：Transformer 位置编码是什么?

Transformer中的位置编码（Positional Encoding）是为了弥补自注意力机制中缺乏顺序信息的缺陷。位置编码有两种常见方式：

- **固定位置编码**：如原始Transformer论文中使用的正弦和余弦函数。每个位置的编码是一个固定的向量，不随训练变化。
- **可训练位置编码**：将位置编码作为可训练的参数，与模型其他参数一起训练。

问题3：QKV 注意力公式为什么除以根号 d

除以根号 $(\sqrt{d_k})$ 的原因是为了防止内积值过大导致softmax函数的梯度消失问题。由于Q和K的维度较高，其内积的值会随 (d_k) 增加而增大， 从而导致softmax的输出极端化（接近0或1）。除以根号 $(\sqrt{d_k})$ 可以使得内积值的方差接近1，保持梯度稳定。

问题4：简单讲讲 GCN

GCN（Graph Convolutional Network，图卷积网络）是一种用于图数据的神经网络。GCN通过在图的节点上进行卷积操作来提取节点的特征。基本的GCN操作步骤包括：

- **邻接矩阵**：用邻接矩阵（A）表示图结构。
- **节点特征矩阵**：用矩阵（X）表示节点特征。
- **卷积操作**：将节点特征与邻接矩阵进行卷积。

问题5：简单讲讲RNN

RNN（Recurrent Neural Network，循环神经网络）是一类用于处理序列数据的神经网络。RNN通过循环结构使得网络可以在序列的每个时间步共享参数， 从而记忆和处理序列中的上下文信息。RNN的基本结构包括：

- **输入层**：接收序列的当前时间步输入。
- **隐藏层**：通过循环连接，将前一时间步的隐藏状态和当前时间步的输入一起处理，生成当前时间步的隐藏状态。
- **输出层**：生成当前时间步的输出。

问题6：RNN 里的参数有什么特点?

RNN的参数具有共享性，即在序列的每个时间步使用相同的一组参数。这使得RNN能够有效处理不同长度的序列，并在参数数量固定的情况下学习序列中的时间依赖关系。

问题7：Dropout 是怎么做的？有什么作用？推理和训练时 Dropout 的区别？如果推理也用 dropout 会怎么样?

Dropout是一种正则化技术，通过在训练过程中随机丢弃（即设置为零）一部分神经元来防止过拟合。具体步骤为：

1. 在每一层的输出中，以一定的概率 (p) 随机丢弃一些神经元。
2. 对保留的神经元进行放大 $(\frac{1}{1-p})$，以保持总的激活值不变。

作用：通过随机丢弃神经元，减少节点之间的相互依赖，从而提高模型的泛化能力。

推理和训练时 Dropout 的区别

- **训练时**：应用Dropout，即随机丢弃神经元，并对保留的神经元进行放大。
- **推理时**：不应用Dropout，使用完整的网络。

如果推理时也用Dropout，模型的输出将变得不稳定，因为每次推理时网络结构都不同，导致结果不可预测且精度下降。

问题8：讲讲 BN？BN 训练和推理什么区别？有什么用?

Batch Normalization（BN）是一种加速深层神经网络训练并提高其稳定性的方法。BN通过对每一层的输入进行标准化，使得输入具有零均值和单位方差， 同时允许网络学习最优的均值和方差。

训练和推理的区别

- **训练时**：使用mini-batch的均值和方差，并更新全局均值和方差的移动平均值。
- **推理时**：使用训练过程中计算的全局均值和方差。

作用

- 减少内部协变量偏移（Internal Covariate Shift）。
- 加快训练速度。
- 提高模型的泛化能力。

2. 趣玩科技推荐算法面试题9道

问题1：二分类的分类损失函数?

二分类的分类损失函数一般采用交叉熵（Cross Entropy）损失函数，即 CE 损失函数。二分类问题的 CE 损失函数可以写成：

$$-y\log(p)-(1-y)\log(1-p)$$

其中， y 是真实标签， p 是预测标签，取值为0或1。

问题2：多分类的分类损失函数(Softmax)?

多分类问题一般采用交叉熵损失函数与Softmax激活函数结合使用。多分类问题的交叉熵损失函数可以写成：

$$-\sum_{i=1}^{N}y_i\log(p_i)$$

其中， N 是类别的数量， y_i 是第 i 类的真实标签， p_i 是第 i 类的预测概率。

问题3：关于梯度下降的sgdm,adagrad，介绍一下。

SGD（Stochastic Gradient Descent）是最基础的梯度下降算法，每次迭代随机选取一个样本计算梯度并更新模型参数。 SGDM（Stochastic Gradient Descent with Momentum）在 SGD 的基础上增加了动量项，可以加速收敛。 Adagrad（Adaptive Gradient）是一种自适应学习率的梯度下降算法，它根据每个参数的梯度历史信息调整学习率，可以更好地适应不同参数的变化范围。

问题4：为什么不用 MSE 分类用交叉熵?

MSE （ Mean Squared Error ）损失函数对离群点敏感，而交叉熵（Cross Entropy）损失函数在分类问题中表现更好， 因为它能更好地刻画分类任务中标签概率分布与模型输出概率分布之间的差异。

问题5：yolov5 相比于之前增加的特性有哪些?

YOLOv5 相比于之前版本增加了一些特性，包括：使用 CSP（Cross Stage Partial）架构加速模型训练和推理；采用 Swish 激活函数代替 ReLU； 引入多尺度训练和测试，以提高目标检测的精度和召回率；引入 AutoML 技术，自动调整超参数以优化模型性能。

问题6：可以介绍一下 attention 机制吗?

Attention 机制是一种用于序列建模的技术，它可以自适应地对序列中的不同部分赋予不同的权重，以实现更好的特征表示。 在 Attention 机制中，通过计算查询向量与一组键值对之间的相似度，来确定每个键值对的权重，最终通过加权平均的方式得到 Attention 向量。

问题7：关于 attention 机制，三个矩阵 Q,K,V的作用是什么?

在 Attention 机制中，KQV 是一组与序列中每个元素对应的三个矩阵，其中 K 和 V 分别代表键和值，用于计算对应元素的权重，Q 代表查询向量， 用于确定权重分配的方式。三个矩阵 K、Q、V 在 Attention 机制中的具体作用如下：

- K（Key）矩阵：K 矩阵用于计算每个元素的权重，是一个与输入序列相同大小的矩阵。通过计算查询向量 Q 与每个元素的相似度， 确定每个元素在加权平均中所占的比例。
- Q（Query）向量：Q 向量是用来确定权重分配方式的向量，与输入序列中的每个元素都有一个对应的相似度，可以看作是一个加权的向量。

- V（Value）矩阵：V 矩阵是与输入序列相同大小的矩阵，用于给每个元素赋予一个对应的特征向量。在 Attention 机制中， 加权平均后的向量就是 V 矩阵的加权平均向量。

通过K、Q、V三个矩阵的计算，Attention机制可以自适应地为输入序列中的每个元素分配一个权重，以实现更好的特征表示。

问题8：介绍一下文本检测 EAST?

EAST（Efficient and Accurate Scene Text）是一种用于文本检测的神经网络模型。EAST 通过以文本行为单位直接预测文本的位置、方向和尺度， 避免了传统方法中需要多次检测和合并的过程，从而提高了文本检测的速度和精度。EAST 采用了一种新的训练方式，即以真实文本行作为训练样本， 以减少模型对背景噪声的干扰，并在测试阶段通过非极大值抑制（NMS）算法进行文本框的合并。

问题9：编程题(讲思路)：给定两个字符串 s,t，在 s 字符串中找到包含 t 字符串的最小字符串。

给定两个字符串 s、t，可以采用滑动窗口的方式在 s 中找到包含 t 的最小子串。具体做法如下：

- 定义两个指针 left 和 right，分别指向滑动窗口的左右边界。
- 先移动 right 指针，扩展滑动窗口，直到包含了 t 中的所有字符。
- 移动 left 指针，缩小滑动窗口，直到无法再包含 t 中的所有字符。
- 记录当前滑动窗口的长度，如果小于之前记录的长度，则更新最小长度和最小子串。
- 重复（2）到（4）步骤，直到 right 指针到达 s 的末尾为止。

3.快手推荐算法面试题7道

问题1：为什么 self-attention 可以堆叠多层，有什么作用?

Self-attention 能够捕捉输入序列中的长距离依赖关系，通过堆叠多层 self-attention，模型可以学习序列中更深层次的模式和依赖关系。 多层 self-attention 就像神经网络中的多个隐藏层一样，使模型能够学习和表示更复杂的函数。

问题2：多头有什么作用？如果想让不同头之间有交互，可以怎么做?

多头注意力（Multi-head attention）的设计是为了让模型同时学习到输入序列的不同表示。每个“头”都有自己的参数，可以学习到不同的注意力分布， 这样可以让模型同时关注不同的特征或信息。至于不同头之间的交互，这通常在所有头的输出被拼接和线性转换之后自然实现。如果你希望在这之前增加交互， 你可能需要设计新的结构或者机制。

问题3：讲一讲多目标优化，MMoE 怎么设计？如果权重为 1,0,0 这样全部集中在某一个专家上该怎么办?

多目标优化是指优化多个目标函数，通常需要在不同目标间找到一个权衡。多门专家混合网络（MMoE, Multi-gate Mixture-of-Experts） 是一种处理多目标优化的方法，其中每个目标都由一个专家网络来处理，而门网络则决定每个专家对最终输出的贡献。如果权重全部集中在某一个专家上， 那么模型的输出就完全由那个专家决定。这可能在某些情况下是合理的，但在大多数情况下，你可能希望各个专家都能对输出有所贡献， 这需要通过训练和调整权重来实现。

问题4：介绍一下神经网络的优化器有哪些。

常见的神经网络优化器有：

- 梯度下降（GD）
- 随机梯度下降（SGD）
- 带动量的随机梯度下降（Momentum SGD）
- Adagrad
- RMSProp
- Adam
- Adadelta
- Nadam 等

问题5：介绍一下推荐算法的链路流程。

推荐系统通常包括以下步骤：

- 数据收集（用户行为、物品信息等）
- 特征工程
- 模型选择和训练
- 推荐列表生成
- 排序等

问题6：介绍一下神经网络的初始化方法。

常见的神经网络初始化方法有：

- 零初始化（所有权重设为 0）
- 随机初始化（权重随机设定，如高斯初始化或均匀分布初始化）
- Xavier/Glorot 初始化（权重初始化为均值为 0，方差为 $(\frac{1}{n})$ 的正态分布或均匀分布，其中 (n) 为输入神经元的数量）
- He 初始化（类似于 Xavier，但方差为 $(\frac{2}{n})$，适用于 ReLU 激活函数）

问题7：讲一讲推荐算法序列建模的模型。

推荐算法中的序列建模通常使用序列模型来捕捉用户行为的时间依赖性。常见的序列模型有：

- RNN（如 LSTM 和 GRU）
- 序列到序列模型（Seq2Seq）
- 注意力模型（如 Transformer）
- 预训练模型（如 BERT、GPT 等）

这些模型可以处理用户行为序列，学习用户的历史行为对他们未来行为的影响，并据此进行推荐。

4.华为NLP算法面试题9道

问题1. **NLP中常见的分词方法有哪些?**

常见的中文分词方法包括基于规则、基于统计和基于深度学习的方法。 其中，基于规则的方法根据预先定义的规则对文本进行切分；基于统计的方法通过统计某个词 在语料库中出现的概率来进行分词；基于深度学习的方法则利用深度神经网络模型从大规模语料中学习分词模型。

问题2. **讲一下BERT的结构?**

BERT（Bidirectional Encoder Representations from Transformers）是**一种基于Transformer模型的预训练语言模型。** 其结构由多层 Transformer编码器组成，其中每层包含多头自注意力机制和前馈神经网络。**BERT还采用了双向训练策略，使得模型能够在不同层次、 不同粒度下理解输入序列中的上下文信息。**

问题3. **自然语言处理有哪些任务?**

自然语言处理任务包括**文本分类、命名实体识别、情感分析、机器翻译、文本生成、问答系统等。**

问题4. **L1，L2正则化的区别，岭回归是L1正则化还是L2正则化?**

L1正则化和L2正则化都是用来约束模型的复杂度，以避免过拟合。 L1正则化通过将模型参数的**绝对值之和**作为正则项，使得一些参数变为0， 从而达到特征选择的目的；L2正则化通过将模型参数的**平方和**作为正则项，使得参数的值都变小，从而使模型更加稳定。岭回归是一种使用L2正则化的线性回归模型。

问题5. **怎么处理类别不平衡?**

类别不平衡问题可以通过过采样、欠采样、生成新样本、集成学习等方法来解决。 过采样方法包括随机过采样、SMOTE等；欠采样方法包括随机欠采样、 TomekLinks等；生成新样本方法包括GAN、VAE等；集成学习方法包括Bagging、Boosting等。

问题6. **模型提速的方法有哪些?**

模型提速的方法包括**模型压缩、剪枝、量化、分布式训练等。** 模型压缩包括权重共享、低秩近似、深度可分离卷积等方法； 剪枝包括通道剪枝、结构剪枝、权重剪枝等；量化包括权重量化、激活量化等；分布式训练则是利用多个计算节点共同完成模型训练任务，加速训练过程。

问题7. **了解数据挖掘的方法嘛?**

数据挖掘包括**数据预处理、特征工程、模型构建和模型评估等步骤。** 其中，数据预处理包括数据清洗、数据集成、数据变换和数据规约； 特征工程包括特征提取、特征选择和特征构建；模型构建包括选择合适的模型和模型参数调优；模型评估包括模型效果评估和模型泛化能力评估。

问题8. **了解对比学习嘛?**

对比学习是一种无监督学习方法， 通过训练模型使得相同样本的表示更接近，不同样本的表示更远离，从而学习到更好的表示。 对比学习通常使用对比损失函数，例如Siamese网络、Triplet网络等，用于学习数据之间的相似性和差异性。

问题9. **说一下广度优先遍历和深度优先遍历?**

- **广度优先遍历（BFS）** 是一种图形搜索算法，从起点开始，依次访问与起点相邻的所有节点，再访问与这些节点相邻的所有未访问过的节点， 直到找到目标节点或者所有节点都被访问。广度优先遍历使用队列来保存访问过的节点。
- **深度优先遍历（DFS）** 是一种图形搜索算法，从起点开始，一直访问相邻节点，直到达到最深的节点，再返回上一级节点，继续访问其他未访问过的节点， 直到找到目标节点或者所有节点都被访问。深度优先遍历使用栈来保存访问过的节点。与广度优先遍历相比，深度优先遍历更适用于搜索深度较深的图形。

5.联想算法面试题9道

问题1：分类问题的交叉熵是什么？

分类问题的交叉熵（cross-entropy） 是一种用来衡量分类模型输出与真实标签之间差异的指标。在二分类问题中，交叉熵可以表示为以下公式：

$$H(p,q) = -\sum_{c=1}^{2} p(c)\log(q(c))$$

其中， p 表示真实标签， q 表示模型预测的标签， N 表示样本数量。

问题2：分类问题是否可以用MSE？

分类问题通常不能使用均方误差（MSE）作为损失函数， 因为分类问题中的标签通常是离散的，而MSE适用于连续变量的回归问题。 当使用MSE作为损失函数时，模型的输出可能会超过1或小于0，这是因为MSE的计算方式不适用于概率值的范围。

问题3：推荐系统中，相比于余弦相似度，是否可以用欧几里得距离判断相似度？

在推荐系统中，通常使用余弦相似度来度量用户或物品之间的相似度。 因为余弦相似度考虑的是向量之间的夹角，而不是向量的长度， 因此它对于不同大小的向量比较稳健。而欧几里得距离则是考虑向量之间的长度，因此对于不同大小的向量比较敏感。

问题4：过拟合怎么处理？

过拟合是指模型在训练集上表现良好，但在测试集上表现较差的现象。过拟合的常见处理方法包括：

- **增加数据量**：通过增加训练数据来降低模型对于训练集的过度拟合。
- **简化模型**：减少模型的参数量，简化模型的结构，降低模型的复杂度。
- **正则化**：通过在损失函数中添加正则化项，限制模型参数的大小，从而避免模型过度拟合。常见的正则化方法包括L1正则化和L2正则化。

问题5：L1、L2正则化的效果、区别、原理？

L1正则化和L2正则化都是正则化方法，目的是通过限制模型参数的大小，降低模型的复杂度，防止过拟合。

- **L1正则化**：

  $$(\text{L1 Regularization} = \lambda \sum |w_i|)$$

  ，会使一部分参数变为0，从而实现特征选择的效果， 适合处理稀疏数据。
- **L2正则化**：

  $$(\text{L2 Regularization} = \lambda \sum w_i^2)$$

  ，会让所有参数都趋向于较小的值，但不会使参数为0。

问题6：Dropout的原理、在训练和测试时的区别？

Dropout是一种常用的正则化方法， 其原理是在每次迭代中随机将一部分神经元的输出置为0，从而减少神经元之间的共适应性，防止过拟合。 在训练时，Dropout会随机将一定比例的神经元的输出置为0，而在测试时，为了保持网络的稳定性，Dropout通常被关闭。

问题7：SGD、Adam、动量优化的SGD?

SGD（Stochastic Gradient Descent）是一种基本的梯度下降算法，使用单个样本的梯度来更新模型参数。 动量优化的SGD（Momentum SGD） 在SGD的基础上引入了动量，使得更新方向更加平稳，收敛速度更快。 Adam是一种基于梯度的优化算法，结合了RMSProp和动量优化的思想， 通过维护一个梯度的指数加权移动平均和梯度平方的指数加权移动平均来自适应地调整每个参数的学习率。

问题8：Adam和动量优化的SGD效率上的区别?

Adam相对于动量优化的SGD有自适应性、速度快和可靠性高的优点。 Adam能够自适应地调整每个参数的学习率，从而在不同的参数空间下能够更加快速地收敛， 并且对超参数的选择不太敏感。

问题9：推荐系统中，如何进行负采样?

在推荐系统中，**负采样是一种重要的技术，用于构造负样本，以便训练推荐模型。** 负采样的过程通常包括计算每个物品的权重、根据权重进行采样、 去除已有的正样本和控制负采样比例等步骤。

6.字节电商CV算法实习岗面试题8道

问题1：如何解决类别极度不平衡的问题?

- 在机器学习中，类别不平衡是指数据集中各类样本的数量差异较大，这可能导致模型对多数类过度拟合，而对少数类泛化能力差。

问题2：说下 Transformer 模型

- **Transformer 模型** 是一种基于自注意力机制的神经网络架构，广泛应用于序列到序列的任务中。 Transformer 本身是一个典型的encoder-decoder模型， Encoder端和Decoder端均有6个Block， Encoder端的Block 包括两个模块，多头self-attention模块以及一个前馈神经网络模块；Decoder端的 Block 包括三个模块，多头self-attention模块，多头Encoder-Decoder attention交互模块，以及一个前 馈神经网络模块； 需要注意：Encoder端和Decoder端中的每个模块都有残差层和LayerNormalization层。

问题3：说下 Focal Loss

- **Focal loss 是目标检测中解决正负样本严重不平衡的方法**，在标准交叉熵损失基础上修改得到的。这个函数可以通过减少易分类样本的权重， 使得模型在训练时更专注于稀疏的难分类的样本；防止大量易分类负样本在loss 中占主导地位

问题4：介绍下深度可分离卷积和传统卷积的区别

- **深度可分离卷积首先在每个通道上独立进行空间卷积，然后使用 1×1 的卷积来组合通道特征，与传统卷积相比，它大大减少了参数数量和计算量。**
- 传统的卷积是各个通道上采用不同的卷积核，然后不同的卷积核用于提取不同方面的特征。
- 深度可分离卷积先在各个通道上采用不同的卷积核提取不同的特征，但是这样对于某个通道来说，就只提取了一方面的特征，因此在此基础上加入点卷积， 用1*1的卷积对提取特征后的特征图再次提取不同方面的 特征，最终产生和普通卷积相同的输出特征图。

问题5：如何防止过拟合

- 防止过拟合的方法包括：获取更多数据、数据增强、正则化（如L1、L2正则化）、Dropout、Early Stopping等。

问题6：BN 在训练和测试的时候的区别？可以防止过拟合吗？

- **Batch Normalization (BN) 在训练时对每个mini-batch的数据进行归一化，而在测试时使用整个训练集的统计量。**
- **BN有助于防止过拟合，因为它使得模型对于输入数据的分布变化更加鲁棒。** BN算法防止过拟合：在网络的训练中，BN的使用使得一个minibatch中所有样本都被关联在了一起， 因此网络不会从某一个训练样本中生成确定的结果，即同样一个样本的输出不再仅仅取决于样本的本身，也取 决于跟这个样本同属一个batch的其他样本， 而每次网络都是随机取batch，这样就会使得整个网络不会朝这一个方向使劲学习。一定程度上避免了过拟合。

问题7：什么是 AUC?

- **AUC（Area Under the Curve）是ROC曲线下的面积，用来衡量分类器的排序能力。** AUC可以解读为从所有正例中随机选取一个样本A， 再从所有负例中随机选取一个样本B，分类器将A判为正例的概率比将B判为正例的概率大的可能性。AUC反映的是分类器对样本的排序能力。 AUC越大，自然排序能力越好，即分类器将越多的正例排在负例之前。

问题8：卷积核计算公式

- 卷积层的输出特征图可以通过以下公式计算：

$$Output = (W - K + 2P)/S + 1$$

其中 W 是输入特征图的大小， K 是卷积核的大小， P 是padding的大小， S 是步长。

- 如果想保持卷积前后的特征图大小相同，通常会设定padding为：

$$P = \frac{K - 1}{2}$$

7.快手多模态算法岗面试题7道

问题1：PPO 和 DPO 有什么区别?

PPO (Proximal Policy Optimization)

PPO 是一种强化学习算法，它通过策略优化方法实现稳定训练。PPO 的核心是限制策略更新的幅度，以避免策略的剧烈变化，从而减少策略崩溃的风险。 它通过剪裁损失函数，确保策略变化在一个较小的范围内。PPO 引入了近端目标函数，利用优势函数更新策略，以兼顾策略的探索和收敛。

DPO (Direct Policy Optimization)

DPO 是一种较新的算法，旨在简化强化学习中的策略优化问题。它通过直接最小化目标函数来优化策略，而不是通过PPO的对数比率和剪裁损失函数。 DPO 采用了更直接的优化方式，简化了策略更新过程。

主要区别

- **策略更新**: PPO 通过限制策略变化幅度来实现稳定训练，而 DPO 直接优化目标函数。
- **稳定性和效率**: PPO 通常更稳定，但训练效率可能较低；DPO 更高效，但可能牺牲一些稳定性。

问题2：DPO训练可能遇到的问题

- **梯度爆炸或消失**: DPO 直接优化策略目标函数，可能导致策略更新过快或过剧，引发梯度爆炸或消失问题。
- **收敛性问题**: 缺少限制策略更新的机制，可能导致训练过程中的不稳定或策略崩溃。
- **探索与利用的平衡**: DPO 可能过早地利用已有策略，导致探索不足，从而无法找到全局最优解。

问题3：位置编码介绍及ROPE（相对位置编码）的优势

Transformer 需要位置编码以捕捉序列中的位置信息，因为其缺乏循环结构。位置编码分为：

- **绝对位置编码 (APE)**: 为每个位置赋予固定的唯一标识。
- **相对位置编码 (ROPE)**: 动态捕捉序列中的相对位置信息，提供更好的外推能力。在处理长度变化的序列时，ROPE 相比 APE 表现更优， 因其不依赖于绝对位置，能更好地处理序列中元素之间的相对关系。

问题4：Transformer 的结构整体介绍一下

Transformer 是一种基于自注意力机制的深度学习模型，广泛应用于自然语言处理任务中。其核心结构包括：

- **Encoder和Decoder**: Encoder 负责对输入序列进行编码，生成一组高维特征表示；Decoder 则根据这些表示和目标序列进行解码，生成最终输出。
- **多头自注意力机制 (Multi-Head Self-Attention)**: 通过多个注意力头并行处理序列数据，捕捉不同位置之间的依赖关系。
- **前馈神经网络 (Feed-Forward Network, FFN)**: 在注意力层后面，通过两层全连接网络对每个位置进行独立的非线性变换。
- **残差连接和LayerNorm**: 通过残差连接和LayerNorm，避免梯度消失问题，同时加快模型的收敛速度。

问题5：BatchNorm与LayerNorm的区别及Transformer的选择

- **BatchNorm**: 主要用于卷积神经网络，通过标准化每个维度来减少内部协变量偏移。
- **LayerNorm**: 主要用于序列模型，如RNN和Transformer，对整个层进行标准化，独立于mini-batch。

问题6：PostNorm与PreNorm的优缺点

- **PostNorm**: 在注意力层和前馈网络后进行归一化，有助于后期训练效果，但前期训练可能不稳定。
- **PreNorm**: 在注意力层和前馈网络前进行归一化，增强模型初期稳定性，但可能影响深层网络的表现能力。

问题7：PostNorm与PreNorm都需要warm up吗?

- **Warm-up**: 用于缓解训练初期梯度过大或过小的问题，帮助模型平稳过渡到正常训练。
- **PreNorm**: 通常不需要warm-up，因为归一化使得初期梯度更新稳定。
- **PostNorm**: 通常需要warm-up，因为初期训练时梯度可能不稳定，warm-up能有效缓解这个问题。

8.平安产险科技中心算法一面面试题9道

问题1：大模型有哪些可选参数?

在训练和调优大模型时，**有几个关键参数可以选择和调整：**

- **学习率（Learning Rate）**：影响模型收敛速度和稳定性。

- **批量大小（Batch Size）**：影响训练稳定性和内存占用，较大的批量大小通常可以加快训练速度，但也需要更多的内存。
- **模型层数（Number of Layers）**：决定模型的深度，层数越多，模型的表达能力越强，但训练难度也增加。
- **隐藏层维度（Hidden Size）**：影响模型的参数量和计算复杂度。
- **头数（Number of Attention Heads）**：影响自注意力机制的表现，更多的头数可以使模型更好地捕捉多种不同的关系。
- **正则化参数（如Dropout率）**：防止过拟合的重要因素。
- **温度（Temperature）**：在生成阶段影响输出的多样性和随机性。

问题2：Transformer的mask是如何操作的?

Transformer模型使用mask主要有两个目的：

- **防止信息泄露**：在训练自回归模型（如GPT系列）时，模型在生成每个单词时只能看到它前面的单词。为了实现这一点，使用了"前向mask"。 这种mask通常是一个上三角矩阵，其中位置(i, j)的值为1（表示masked），如果i < j，否则为0。这样，在计算自注意力时，模型只关注当前单词及其之前的单词。
- **填充mask**：在处理不同长度的输入序列时，通常会对较短的序列进行填充（padding），以使它们具有相同的长度。 为了防止模型在计算时关注填充部分，使用填充mask。填充mask通常会为填充的部分设置为1（masked），而非填充的部分为0。这样，模型在计算自注意力时就会忽略填充的部分。

问题3：Self attention的公式是什么，为什么要除以sqrt

Self Attention的公式为：

$$[\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V]$$

其中，(d_k) 是键向量的维度。除以 $(\sqrt{d_k})$ 是为了防止在高维空间中，点积的值过大，从而导致softmax函数的梯度消失，确保梯度更平稳，提升训练效果。

问题4：Temperature的大小如何影响分布?

温度参数在模型生成阶段起着调节作用：

- **低温度（如0.5）**：会使概率分布更加集中，生成的输出更加确定，通常会导致重复和常见的输出，因为模型倾向于选择高概率的词。
- **高温度（如1.5）**：会使概率分布变得更平坦，增加生成输出的随机性，使得模型更可能选择低概率的词，从而增加多样性和创新性， 但也可能导致输出的质量下降，生成不相关或无意义的内容。

总的来说，温度的选择依赖于具体应用的需求：如果需要更具创新性和多样性的输出，可以选择较高的温度；如果需要更稳定和可靠的输出，可以选择较低的温度。

问题5：Attention的复杂度是多少?

Attention机制的时间复杂度为 $O(n^2 * d)$（n为输入序列长度，d为特征维度），因为对于每个输入元素都需要计算与所有其他元素的相似度。

问题6：什么情况下，MAP的损失函数可以用NMSE来计算?

在一些特定的场景下，如高斯先验和高斯似然的情况下，MAP估计的损失函数可以用均方误差（NMSE）来计算，因为这时优化的目标与最小化均方误差等价。

问题7：MAP（最大后验概率）和似然函数有什么关系？

MAP（最大后验概率）估计是基于贝叶斯理论的，结合了似然函数和先验分布。MAP目标是最大化后验概率，公式为：

$$[P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}]$$

其中，$(P(X|\theta))$ 是似然函数。

问题8：vector扩容方法是什么？

`std::vector` 在其容量不足以容纳新元素时，会进行扩容。扩容通常是倍增策略，即当元素数量达到当前容量时，新的容量会设置为原来容量的两倍。 这样做可以保持摊销的时间复杂度为 $O(1)$ 。扩容时，vector会分配新的内存，将旧元素复制到新内存中，然后释放旧内存。

问题9：智能指针、虚函数？

- **智能指针**：是一种自动管理内存的指针类型，常见于C++，如 `std::unique_ptr` 和 `std::shared_ptr`。它们可以防止内存泄漏，通过引用计数和作用域控制自动释放内存。
- **虚函数**：在基类中声明为 `virtual` 的成员函数，用于实现多态性。通过虚函数，派生类可以重写基类的方法，从而在运行时决定调用哪个版本。