

AIGC算法工程师面经—公式理解篇（下）

原创 喜欢瓦力的卷卷 瓦力算法学研所 2024年04月25日 15:40 广东

面试经验专栏

本篇总结了AIGC面经中公式理解类问题与相应答案。

本篇开始填上一篇 AIGC算法工程师面经—公式理解篇（上） 的坑，没看过（上）的小伙伴们可以看看。

再重复一下~ 在实际面试中，这类问题很大概率需要手写，或者需要很清晰地讲出公式含义及原理，这个过程中可能会遭到反复拷打，甚至手撕代码。

在本篇中，为了便于辅助记忆与掌握原理，公众号文章中尽量用简洁的文字介绍，并提供了附有问题类型总结、详细公式代码与原理讲解的完整版ppt版本便于大家后续复习。

下面是一个问题的快捷目录，需要注意答案中每个问题下方可能有引申的更深入的问题。本文将对剩余的6个问题进行解答。

面试题

1. 手写softmax公式，手写BN公式，softmax层的label 是什么？
2. 交叉熵公式，分类为什么用交叉熵不用平方差？
3. 手推lr梯度，交叉熵损失为什么有log项？为什么取负？
4. NER任务的损失函数是什么，写出来并解释一下？
5. 为什么逻辑回归用sigmoid激活函数？多分类逻辑回归是否也是sigmoid？
6. KL loss 公式是什么？一般什么情况下用的？
7. 请解释ReLU激活函数的公式以及它的作用？
8. 在卷积神经网络中，卷积操作的数学表达式是什么？请解释卷积核、步长和填充在其中的作用？
9. 请解释残差连接的公式和原理，并说明它在深度神经网络中的作用？
10. 请解释LSTM中遗忘门的计算公式以及其作用？
11. 请解释L1和L2正则化的公式，以及它们在深度学习中的作用。
12. 什么是贝叶斯定理？请写出其数学表达式，并解释每个部分的含义。

答案

7.请解释ReLU激活函数的公式以及它的作用？

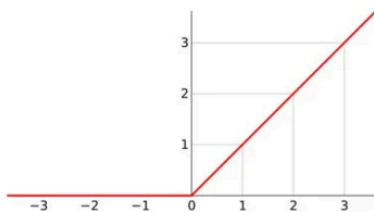
- ReLU函数的定义很简单：对于输入的任何负值，输出都是0；对于输入的任何非负值，输出等于输入。这种简单的非线性函数使得它的导数在大多数情况下都是易于计算的，并且具有使网络更快收敛的能力。
- 手写ReLU激活函数公式

$$\text{ReLU}(x) = \max(0, x)$$

备注：其实还可能会有其他延申的问题，包括有哪些基于ReLU改进的激活函数，具体推荐大家去看看ppt。

✦ ReLU激活函数面经问题 ✦

问题2：请画出ReLU图像，还知道有哪些基于ReLU改进的激活函数



ReLU图像如上所示

以下是一些常见的改进的ReLU函数：

1. Leaky ReLU (LReLU): Leaky ReLU在输入为负时不是严格地输出零，而是输出一个非零的小值，通常是输入的一个小比例。

2. Parametric ReLU (PReLU): Parametric ReLU是Leaky ReLU的扩展，其负数区间的斜率是可以通过网络学习得到的，而不是固定的。这使得网络可以自适应地学习每个神经元的激活函数形状。

3. Exponential Linear Units (ELU): ELU函数在负数区间不仅允许斜率变化，而且能够保持平滑，避免了ReLU可能带来的不连续性问题。

公众号 · 瓦力算法学研所

8.在卷积神经网络中，卷积操作的数学表达式是什么？请解释卷积核、步长和填充在其中的作用？

- 卷积操作的数学表达式：

$$(f * g)(i, j) = \sum_m \sum_n f(m, n) \cdot g(i - m, j - n)$$

其中， $(f * g)(i, j)$ 表示输出特征图中的像素值， $f(m, n)$ 是输入图像中的像素值， $g(i - m, j - n)$ 是卷积核中的权重值， m 和 n 是卷积核的坐标， i 和 j 是输出特征图中的坐标。

- 卷积核作用：

卷积核是卷积操作中的一个小的矩阵，它包含了一组权重参数，用于对输入数据的局部区域进行滤波操作。卷积核的大小和形状可以根据任务和需求进行设置，常见的大小为3x3或5x5。

- **步长的作用：**

步长是卷积核在输入数据上滑动的步长，它决定了输出特征图的尺寸。1.较大的步长会导致输出特征图的尺寸缩小，而较小的步长会导致输出特征图的尺寸保持较大。

填充的作用：填充是在输入数据的边缘上添加额外的像素，以便在卷积操作中保持输出特征图的尺寸不变或者减小的速度降低。填充可以是零填充（zero-padding），即在边缘上添加零值像素；也可以是其他值的填充。

9.请解释残差连接的公式和原理，并说明它在深度神经网络中的作用？

- **残差连接原理：**

残差连接的核心思想是引入一个“快捷连接”（shortcut connection）或“跳跃连接”（skip connection），允许数据绕过一些层直接传播。

- **残差连接公式：**

$$\text{output} = F(\text{input}) + \text{input}$$

- **残差连接作用：**

- 缓解梯度消失问题：**通过将输入直接添加到网络的中间层输出中，残差连接使得梯度能够更容易地在网络中传播回较早的层，从而缓解了深层网络训练过程中的梯度消失问题。
- 提高网络的收敛速度：**由于残差连接使得网络更容易训练和优化，因此可以加速网络的收敛速度，减少训练时间和计算资源的消耗。
- 构建更深的网络结构：**残差连接使得网络可以更容易地构建更深的网络结构，而不会出现梯度消失或梯度爆炸等问题，从而提高了网络的表达能力和性能。

备注：其它还有例如代码手撕残差连接结构模型，请参考ppt

✦ 残差连接面经问题

问题2：请用代码实现一个残差连接结构模型

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class ResidualBlock(nn.Module):
    def __init__(self, in_channels, out_channels, stride=1):
        super(ResidualBlock, self).__init__()
        self.conv1 = nn.Conv2d(in_channels, out_channels, kernel_size=1, stride=stride)
        self.bn1 = nn.BatchNorm2d(out_channels)
        self.conv2 = nn.Conv2d(out_channels, out_channels, kernel_size=1, stride=1)
        self.bn2 = nn.BatchNorm2d(out_channels)
        self.relu = nn.ReLU(inplace=True)
        self.downsample = nn.Sequential()
        if stride != 1 or in_channels != out_channels:
            self.downsample = nn.Sequential(
                nn.Conv2d(in_channels, out_channels, kernel_size=1, stride=stride, bias=False),
                nn.BatchNorm2d(out_channels)
            )

    def forward(self, x):
        identity = x
        out = self.conv1(x)
        out = self.bn1(out)
        out = self.relu(out)
        out = self.conv2(out)
        out = self.bn2(out)
        out += self.downsample(identity)
        out = self.relu(out)
        return out
```

10.请解释LSTM中遗忘门的计算公式以及其作用？

• 遗忘门

决定我们从神经元中丢弃什么信息，这个操作由一个忘记门层来完成。输出结果1表示“完全保留”，0表示“完全舍弃”。公式如下所示：

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

公众号·瓦力算法学研所

• 输入门

sigmoid层作为“输入门层”，决定我们将要更新的值，确定神经元状态所存放的新信息，这一步由两层组成。同时，需要更新状态信息。公式如下所示：

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned} \longrightarrow C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

公众号·瓦力算法学研所

• 输出门

输出将会基于更新后的神经元状态，更像是一个过滤后的版本。公式如下所示：

$$\begin{aligned} o_t &= \sigma(W_o [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned}$$

公众号·瓦力算法学研所

11.请解释L1和L2正则化的公式，以及它们在深度学习中的作用。

- L1正则化是通过模型参数的绝对值之和来惩罚模型复杂度，其公式如下：

$$L_{L1} = \lambda \sum_{i=1}^n |w_i|$$

- L2正则化是通过模型参数的平方和来惩罚模型复杂度，其公式如下：

$$L_{L2} = \lambda \sum_{i=1}^n w_i^2$$

- **L1和L2正则化具有：**

- 防止过拟合：** L1和L2正则化通过限制模型参数的范围，降低了模型的复杂度，减少了模型对训练数据的拟合程度，从而可以有效地防止模型过拟合。
- 稀疏性（L1正则化）：** L1正则化倾向于使得部分模型参数变为零，从而实现了特征的稀疏性，有助于特征选择和模型的解释性。
- 参数平滑（L2正则化）：** L2正则化对模型参数进行平滑处理，使得模型参数不会出现剧烈的变化，有助于提高模型的稳定性和泛化能力。


12.什么是贝叶斯定理？请写出其数学表达式，并解释每个部分的含义。

贝叶斯定理是一种用于计算在给定先验知识的情况下更新观测数据后的后验概率的方法。它是统计学中的重要理论之一，也是贝叶斯统计推断的基础。数学表达式如下：

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

- $P(A|B)$ 是在观测到事件 B 发生后，事件 A 发生的概率，称为后验概率。
- $P(B|A)$ 是在事件 A 发生的情况下，事件 B 发生的概率，称为似然度。
- $P(A)$ 是事件 A 发生的先验概率，即在观测到任何数据之前，我们对事件 A 发生的信念或估计。
- $P(B)$ 是事件 B 发生的边际概率，即在任何数据情况下，事件 B 发生的概率。

想要获取资料的同学请欢迎关注公众号，并且回复“公式理解”获取完整版本ppt。



喜欢卷卷的瓦力

扫一扫上面的二维码图案，加我为朋友。

添加瓦力微信

算法交流群 · 面试群

大咖分享 · 学习打卡

👤 公众号 · 瓦力算法学研所



瓦力算法学研所

我们是一个致力于分享人工智能、机器学习和数据科学方面理论与应用知识的公众号。我...
117篇原创内容

公众号

面试干货 70

[面试干货 · 目录](#)

[上一篇](#)

[垂直领域大模型微调实践经验最全总结](#)

[下一篇](#)

[大模型面经——从prefix-decoder、casual-decoder、encoder-decoder角度深入聊聊...](#)