

补充

不同任务的损失函数

回归任务：

- **目标：**预测连续值，比如房价、温度等。
- **常用损失函数：**均方误差（Mean Squared Error, MSE）或均方根误差（Root Mean Squared Error, RMSE）。这些损失函数计算预测值与实际值之间的平方差，鼓励模型输出接近真实值的连续数值。
- **均方差（Mean Squared Error）：**
均方差是回归任务中常用的损失函数，用于衡量预测值与实际值之间的差异。它的计算公式如下：

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

其中：

n 是样本数量。这是一行文字，其中包含变量 y_i 。

y_i 是第 i 个样本的实际值。

\hat{y}_i 是第 i 个样本的预测值。

分类任务：

- **目标：**预测离散的类别标签，比如垃圾邮件检测、图像识别等。
- **常用损失函数：**
 - **交叉熵损失（Cross-Entropy Loss）：**也称为对数损失，用于多分类问题。它衡量的是模型输出的概率分布与真实标签的概率分布之间的差异。
 - **二元交叉熵损失（Binary Cross-Entropy Loss）：**用于二分类问题，计算的是模型输出为正类的概率与实际标签之间的交叉熵。
 - **交叉熵损失（Cross-Entropy Loss）：**
交叉熵损失是分类任务中常用的损失函数，用于衡量模型输出的概率分布与真实标签的概率分布之间的差异。对于多分类问题，交叉熵损失的计算公式如下：

$$\text{Cross-Entropy Loss} = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m y_{ij} \log(\hat{y}_{ij})$$

其中：

n 是样本数量。

m 是类别数量。

y_{ij} 是第 i 个样本的真实标签，如果第 j 个类别是正确的类别，则 $y_{ij} = 1$ ，否则 $y_{ij} = 0$ 。

\hat{y}_{ij} 是第 i 个样本预测为第 j 个类别的概率。

对于二分类问题，交叉熵损失的计算公式可以简化为：

$$\text{Binary Cross-Entropy Loss} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

其中：

y_i 是第 i 个样本的实际标签，通常取值为0或1。

\hat{y}_i 是第 i 个样本的预测概率。

不同的激活函数

激活函数在神经网络中扮演着非常重要的角色，它们引入非线性因素，使得神经网络能够学习和模拟复杂的函数。不同的激活函数适用于不同的场景和需求。以下是一些常见的激活函数及其应用：

1. Sigmoid:

- 定义:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- 用途: 早期的神经网络中常用于二分类问题，输出值在0和1之间，可以解释为概率。
- 缺点: 梯度消失问题，当输入值很大或很小的时候，梯度接近0，导致网络训练困难。

2. Tanh (双曲正切函数):

- 定义:

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

- 用途: 输出值在-1和1之间，常用于隐藏层，因为它的输出值中心化在0。
- 缺点: 同样存在梯度消失问题。

3. ReLU (Rectified Linear Unit):

- 定义:

$$f(x) = \max(0, x)$$

- 用途: 现代神经网络中最常用的激活函数，因为它简单且计算效率高，减少了梯度消失的问题。
- 缺点: 存在“死亡ReLU”问题，即当输入小于0时，梯度为0，导致部分神经元不再更新。

4. Leaky ReLU:

- 定义:

$$f(x) = \max(0.01x, x)$$

- 用途: 改进ReLU，为负输入提供一个小的非零斜率，以解决死亡ReLU问题。

5. Parametric ReLU (PReLU):

- 定义:

$$f(x) = \max(ax, x)$$

- 用途: Leaky ReLU的变体，其中a是一个可学习的参数。

6. ELU (Exponential Linear Unit):

- 定义:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases}$$

- **用途**：类似于ReLU，但当输入小于0时，输出是负的，有助于提高模型的表达能力。

7. SELU (Scaled Exponential Linear Units) :

- **定义**：

$$f(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha e^x - \alpha & \text{if } x \leq 0 \end{cases}$$

- **用途**：自带归一化效果，有助于网络训练的稳定性。

8. Softmax:

- **定义**：

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

- **用途**：多分类问题的输出层，将输出转换为概率分布。