函数

Softmax 函数

Softmax 函数将任意实数向量 $z = [z_1, z_2, ..., z_K]$ 转换为一个概率分布 $p = [p_1, p2, ..., p_K]$,其中每个元素都在 $(0, 1)$ 范围内，并且所有元素之和为 1.

对于输入向量 z 中的第 k 个元素 z_k, 其中 Softmax 输出 p_k 定义为:

$$p_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

其中 K 是类别数量

LogSoftmax 函数

LogSoftmax 函数对 Softmax 的输出取自然对数。这样做的好处是增强数值稳定性，并且在计算交叉熵时能简化计算。

对于 Softmax 输出 p_k, LogSoftmax 输出 s_k 定义为:

$$s_k = \log(p_k) = \log\left(\frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}\right)$$

根据对数性质，我们可以进一步展开:

$$s_k = z_k - \log\left(\sum_{j=1}^{K} e^{z_j}\right)$$

交叉熵损失 (CrossEntroy Loss)

假设真实标签为 $y = [y_1, y_2, ..., y_K]$ 的 one-hot 编码，其中 $y_c = 1$,而其他 $y_j = 0$. 模型预测的对数概率为 $s = [s_1, s_2, ..., s_K]$.

交叉熵损失 J 定义为:

$$J = -\sum_{k=1}^{K} y_k \log(p_k)$$

由于 LogSoftmax 的输出 $s_k = \log(p_k)$, 我们可以直接用 s_k 代替 $\log(p_k)$:

$$J = -\sum_{k=1}^{K} y_k s_k$$

前向传播

直接计算 e^{z_i} 可能导致数值溢出，特别是当 z_i 特别大的时候。因此，一般 Softmax 使用如下的计算公式:

$$p_k = \frac{e^{z_k - z_{max}}}{\sum_{j=1}^{K} e^{z_j - z_{max}}}$$

其中

$$z_{max} = \max_j z_j$$

因此

$$\begin{aligned} s_k = \log(p_k) &= \log\left(\frac{e^{z_k - z_{max}}}{\sum_{j=1}^{K} e^{z_j - z_{max}}}\right) \\ &= z_k - z_{max} - \log\left(\sum_{j=1}^{K} e^{z_j - z_{max}}\right) \end{aligned}$$

则:

$$J = -\sum_{k=1}^{K} y_k \left[z_k - z_{max} - \log\left(\sum_{j=1}^{K} e^{z_j - z_{max}}\right)\right]$$

批量处理的扩展

对于批量输入 $\mathrm{z} \in^{(N,K)}$, 所有样本的损失总和为 J_{total}:

$$L = \frac{1}{N} J_{total}$$

表示当前批次上的平均损失。

反向传播公式推导

$$\frac{\partial J}{\partial z_i} = \sum_{k=1}^{K} \frac{\partial J}{\partial s_k} \frac{\partial s_k}{\partial z_i}$$

从 $J = -\sum_{k=1}^{K} y_k s_k$ 可知:

$$\frac{\partial J}{\partial s_k} = -\frac{\partial\left(\sum_{j=1}^{K} y_j s_j\right)}{\partial s_k} = -y_k$$

计算 LogSoftmax 输出对输入 z_i 的梯度 $\frac{\partial s_k}{\partial z_i}$:

我们有 $s_k = z_k - z_{max} - \log\left(\sum_{j=1}^{K} e^{z_j - z_{max}}\right)$:

$$\frac{\partial s_k}{\partial z_i} = \delta_{ki} - \frac{1}{\sum_{j=1}^{K} e^{z_j - z_{max}}} e^{z_i - z_{max}} = \delta_{ki} - p_i$$

则:

$$\frac{\partial J}{\partial z_i} = \sum_{k=1}^{K} \left[-y_k \left(\delta_{ki} - p_i \right) \right] = \sum_{k=1}^{K} \left(-y_k \delta_{ki} \right) + \sum_{k=1}^{K} y_k p_i$$

$$= -y_i + p_i \sum_{k=1}^{K} y_k$$

因为 y 是真实标签，也是概率分布，所以 $\sum_{k=1}^{K} y_k = 1$, 最终得到:

$$\frac{\partial J}{\partial z_i} = p_i - y_i$$

结合批次，最终的平均反向梯度输出为:

$$output_i = \frac{1}{N} \frac{\partial J}{\partial z_i}$$

补充

严格来说，z_{max} 也是 z_i 的函数，但在反向传播中，将其视为一个常数，不参数梯度计算，或者说梯度为零。

虽然 z_{max} 确实依赖于输入，但在最终的梯度计算中，所有与 z_{max} 相关的梯度项会相互抵消。

这可以通过以下方式来理解：

- Softmax 函数具有平移不变性：$softmax\left(z\right) = softmax\left(z + c\right)$, 其中 c 为任意常数.
- 减去 z_{max} 只是为了数值稳定性，不会概率最终的概率分布.

我们可以更加严格的推导 $\frac{\partial s_k}{\partial z_i}$,同时将 z_{max} 作为 z_i 的函数来考虑：

我们有 $s_k = z_k - z_{max} - \log\left(\sum_{j=1}^{K} e^{z_j - z_{max}}\right)$ ：

$$\frac{\partial s_k}{\partial z_i} = \delta_{ki} - \frac{\partial z_{max}}{\partial z_i} - \frac{1}{\sum_{k=1}^{K} e^{z_j - z_{max}}} \frac{\partial \left(\sum_{j=1}^{K} e^{z_i - z_{max}}\right)}{\partial z_i}$$

我们计算 $\frac{\partial \left(\sum_{j=1}^{K} e^{z_i - z_{max}}\right)}{\partial z_i}$ ：

$$\frac{\partial \left(\sum_{j=1}^{K} e^{z_i - z_{max}}\right)}{\partial z_i} = \sum_{j=1}^{K} \frac{\partial e^{z_j - z_{max}}}{\partial z_i}$$

对每一项:

$$\frac{\partial e^{z_j - z_{max}}}{\partial z_i} = e^{z_j - z_{max}} \frac{\partial \left(z_j - z_{max}\right)}{\partial z_i}$$

$$= e^{z_j - z_{max}} \left(\delta_{ij} - \frac{\partial z_{max}}{\partial z_i}\right)$$

因此：

$$\frac{\partial\left(\sum_{j=1}^{K}e^{z_i-z_{max}}\right)}{\partial z_i}=\sum_{j=1}^{K}e^{z_j-z_{max}}\left(\delta_{ij}-\frac{\partial z_{max}}{\partial z_i}\right)$$

$$=e^{z_i-z_{max}}-\frac{\partial z_{max}}{\partial z_i}\sum_{j=1}^{K}e^{z_j-z_{max}}$$

从而：

$$\frac{\partial s_k}{\partial z_i}=\delta_{ki}-\frac{\partial z_{max}}{\partial z_i}-\frac{e^{z_i-z_{max}}-\frac{\partial z_{max}}{\partial z_i}\sum_{j=1}^{K}e^{z_j-z_{max}}}{\sum_{j=1}^{K}e^{z_j-z_{max}}}$$

$$=\delta_{ki}-\frac{\partial z_{max}}{\partial z_i}-p_i+\frac{\partial z_{max}}{\partial z_i}=\delta_{ik}-p_i$$

可见 $\frac{\partial z_{max}}{\partial z_i}$ 被抵消了，将之视为常数，对求导结果没有影响。