参数

层参数

1. num_features
    - 作用：指定输入特征的数量
    - 不同维度的含义：
        - BatchNorm1d: 输入形状为 $[N, C]$, 或 $[N, C, L]$ 中的 C
        - BatchNorm2d: 输入形状为 $[N, C, H, W]$ 中的 C
        - BatchNorm3d: 输入形状为 $[N, C, D, H, W]$ 中的 C

超参数

1. momentum (默认 0.1)，超参数
    - 类型：浮点数，通常接近于 0, 比如 0.1
    - 用途：控制 running_mean 和 running_var 的更新速度。momentum 越大，新批次的数据对移动平均值影响越大，更新越快
    - 作用：平衡了训练时的批次波动与全局统计信息的稳定性
2. epsilon (ϵ)
    - 类型：浮点数，通常是一个很小的正数，比如 $\frac{1}{10^5}$
    - 用途：在方差分母上加上一个微小值
    - 作用：增加数值稳定性，防止方差为零而出现除以零的错误

可学习参数

可学习参数再模型训练过程中通过反向传播和梯度下降自动更新，它们通常初始化为特定值。

1. 缩放因子 (γ)
    - 形状: (C), 一个向量，大小等于输入张量的通道数
    - 用途: 用于缩放归一化后的数据。$y = \gamma \hat{x} + \beta$
    - 作用：恢复网络的表达能力。单纯的标准化会强制数据处于零均值和单位方差，这可能会限制非线性激活函数的最佳工作区间。γ 参数允许网络学习一个最优的缩放值，使得数据可以被缩放到任何想要的方差。
    - 初始化：通常初始化为 1.0
2. 偏移因子 (β)
    - 形状: (C), 一个向量，大小等于输入张量的通道数

- 用途：用于偏移归一化后的数据。$y = \gamma\hat{x} + \beta$
- 作用：恢复网络的表达能力。与 γ 类似，β 参数允许网络学习一个最优的偏移量，使得数据可以被偏移到任何想要的均值。
- 初始化： 通常初始化为 0.0

非可学习参数

这些参数是在训练过程中计算并更新，但不会通过反向传播进行优化。它们主要用于推理阶段。

1. 移动平均均值 (running_mean)
    - 形状: (C), 一个向量，大小等于输入张量的通道数
    - 用途： 在训练过程中，通过指数移动平均来估算整个训练集的均值
    - 更新公式：

    ```
    running_mean = (1-momentum)*running_mean + moementum * batch_mean
    ```

    - 作用：在推理阶段，由于无法获得 mini-batch 的均值，BatchNorm 层会使用这个 running_mean 作为全局均值来对数据进行标准化
2. 移动平均方差 (running_var)
    - 形状: (C), 一个向量，大小等于输入张量的通道数
    - 用途： 在训练过程中，通过指数移动平均来估算整个训练集的方差
    - 更新公式：

    ```
    running_var = (1-momentum)* running_var + momentum * batch_var
    ```

    - 作用： 在推理阶段，BatchNorm 层会使用这个 running_var 作为全局方差来对数据进行标准化
    - pytorch 中，running_var 需要乘以 $\frac{m}{(m-1)}$ 转化为无偏估计。其中 $m = N \cdot H \cdot W$, 假如是 BatchNorm2d 的话

数学定义

- 均值

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i$$

- 方差

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2$$

- 归一化

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

- 偏移和缩放

$$y_i = \gamma \hat{x}_i + \beta$$

计算方式

数学定义中的求和符号作用在除了通道 C 之外的其他所有维度上，比如 BatchNorm2d 的输入形状为 $[N, C, H, W]$，那么 $m = N \cdot H \cdot W$。也就是说，需要计算某个通道的统计量的时候，需要将不同批次的求和结果相加，作为最终结果输出。

前向传播

前向传播输入形状根据 BatchNorm1d, BatchNorm2d, BatchNorm3d 而不同。具体形状见 **[num_features]** 中的描述。

先举一个 BatchNorm2d 的示例：

假如输入形状为: $[2, 3, 2, 2]$, 表示有 2 个批次，通道数为 3 ，每个通道是一个长宽为 2 的图像：

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \begin{bmatrix} 9 & 10 \\ 11 & 12 \end{bmatrix}$$

$$\begin{bmatrix} 13 & 14 \\ 15 & 16 \end{bmatrix} \begin{bmatrix} 17 & 18 \\ 19 & 20 \end{bmatrix} \begin{bmatrix} 21 & 22 \\ 23 & 24 \end{bmatrix}$$

1. 首先计算通道均值 (batch_mean)
   - 这里有 3 个通道，所以，均值的形状为 $[3]$, 我们首先将输入 X 重塑为 $[6, 4]$:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 \end{bmatrix}$$

   - 对这个矩阵的每一行求和，得到一个 $[6]$ 的向量:

$$[10, 26, 42, 58, 74, 90]$$

- 再将上述的向量重塑为 $[2, 3]$ 的矩阵:

$$\begin{bmatrix} 10 & 26 & 42 \\ 58 & 74 & 90 \end{bmatrix}$$

- 再将上述 $[2, 3]$ 的矩阵转置为 $[3, 2]$ 的矩阵：

$$\begin{bmatrix} 10 & 58 \\ 26 & 74 \\ 42 & 90 \end{bmatrix}$$

- 对上述 $[3, 2]$ 的矩阵按行求和，得到最终结果：

$$[68, 100, 132]$$

- 再除以 $m = N \cdot H \cdot W = 2 \cdot 2 \cdot 2 = 8$,得:

$$\left[\frac{68}{8}, \frac{100}{8}, \frac{132}{8}\right]$$

2. 计算方差 (batch_var)
   - 同样按照通道进行计算，我们还用 $[6, 4]$ 的矩阵，减去均值之后：

$$\begin{bmatrix} 1 - mean_1 & 2 - mean_1 & 3 - mean_1 & 4 - mean_1 \\ 5 - mean_2 & 6 - mean_2 & 7 - mean_2 & 8 - mean_2 \\ 9 - mean_3 & 10 - mean_3 & 11 - mean_3 & 12 - mean_3 \\ 13 - mean_1 & 14 - mean_1 & 15 - mean_1 & 16 - mean_1 \\ 17 - mean_2 & 18 - mean_2 & 19 - mean_2 & 20 - mean_2 \\ 21 - mean_3 & 22 - mean_3 & 23 - mean_3 & 24 - mean_3 \end{bmatrix}$$

   - 再按照求均值的方式处理，记住先要平方。
   - 对最终的结果除以 m
3. 归一化，使用公式:

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

4. 仿射变换：

$$y_i = \gamma \hat{x}_i + \beta$$

其输出形状与输入相同。

如果是在训练过程中，我们需要缓存几个中间张量，以加速反向传播的计算：

- 均值: batch_mean (μ)

- 方差：batch_var $\left(\sigma^2\right)$
- 归一化结果： $\hat{x}$
- 标准差的倒数： $rstd = \frac{1}{\sqrt{\sigma^2+\epsilon}}$

反向传播时，可以直接用 rstd 进行计算即可。

在推理场景下，无需计算均值和方差，只需要用 running_mean 替代均值，running_var 替代方差，然后进行归一化，再接着仿射变换就得到输出。
所以，模型保存时，除了保存可学习参数 γ, β，还需要保存 running_mean 和 running_var

反向传播

假如已知梯度 $\frac{\partial L}{\partial y}$ 我们先推导反向传播公式。前向传播的公式：

$$y_i = \gamma \hat{x}_i + \beta$$

以下提到的 $m = N \cdot H \cdot W$

1. $\frac{\partial L}{\partial \gamma}$:
   γ 的形状为 $[C]$ 的向量，元素总数与输入通道数相同。即每个 γ_i 作用在输入中 m 个元素。这 m 个元素在输入中均属于第 i 个通道，因此：

   $$\frac{\partial L}{\partial \gamma_i} = \sum_{j=1}^{m} \frac{\partial L}{\partial y_j} \frac{\partial y_j}{\partial \gamma_i}$$

   其中 $\frac{\partial y_j}{\partial \gamma_i} = \hat{x}_j$
   所以，最终的公式：

   $$\frac{\partial L}{\partial \gamma_i} = \sum_{j=1}^{m} \frac{\partial L}{\partial y_j} \hat{x}_j$$

2. $\frac{\partial L}{\partial \beta}$
   β 的形状为 $[C]$ 的向量，它对输入的作用与 γ相似，β_i 影响了第 i 个输入通道中的所有元素，且因为 $\frac{\partial y_j}{\partial \beta_i} = 1$, 所以：

   $$\frac{\partial L}{\partial \beta_i} = \sum_{j=1}^{m} \frac{\partial L}{\partial y_j}$$

3. $\frac{\partial L}{\partial X}$
   我们有：

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

因此，我们得到如下的几个基本公式：

$$\frac{\partial \hat{x}_i}{\partial x_i} = \frac{1}{\sqrt{\sigma^2 + \epsilon}}$$

$$\frac{\partial \hat{x}_i}{\partial \mu} = -\frac{1}{\sqrt{\sigma^2 + \epsilon}}$$

$$\frac{\partial \hat{x}_i}{\partial \sigma^2} = \frac{-\frac{1}{2}\left(x_i - \mu\right)}{\left(\sigma^2 + \epsilon\right)^{\frac{3}{2}}}$$

因为 x_i 不仅直接影响 $\hat{x}_i$，而且还通过 μ 和 σ^2 间接影响 $\hat{x}_i$，所以：

$$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial \hat{x}_i}\frac{\partial \hat{x}_i}{\partial x_i} + \frac{\partial L}{\partial \mu}\frac{\partial \mu}{\partial x_i} + \frac{\partial L}{\partial \sigma^2}\frac{\partial \sigma^2}{\partial x_i}$$

- 计算 $\frac{\partial L}{\partial \hat{x}_i}$:

$$\frac{\partial L}{\partial \hat{x}_i} = \frac{\partial L}{\partial y_i}\gamma$$

- 计算 $\frac{\partial L}{\partial \mu}$:
  μ 的形状为 $[C]$ 的向量，其中 μ_i 是输入第 i 通道所有元素求和再取平均所得，所以，每个 μ_i 作用于第 i 个通道的所有的 $\hat{x}$，所以对 μ_i 的梯度有如下公式：

$$\frac{\partial L}{\partial \mu} = \sum_{j=1}^{m} \frac{\partial L}{\partial \hat{x}_j}\frac{\partial \hat{x}_j}{\partial \mu}$$

- 计算 $\frac{\partial \mu}{\partial x_i}$:
  根据 $\mu = \frac{1}{m}\sum_{i=1}^{m} x_i$，得到：

$$\frac{\partial \mu}{\partial x_i} = \frac{\partial\left(\frac{1}{m}\sum_{j=1}^{m} x_j\right)}{\partial x_i} = \frac{1}{m}$$

  上式求和符号中，$j \neq i$ 时，偏导数为零，$j = i$ 时，偏导数为 1。
- 计算 $\frac{\partial L}{\partial \sigma^2}$:
  σ^2 的形状为 $[C]$ 的向量，其中 σ_i^2 是输入第 i 通道所有元素的方差，所以，每个 σ_i^2 作用于第 i 个通道的所有的 $\hat{x}$，所以对 σ_i^2 的梯度有如下公式：

$$\frac{\partial L}{\partial \sigma^2} = \sum_{j=1}^{m} \frac{\partial L}{\partial \hat{x}_j}\frac{\partial \hat{x}_j}{\partial \sigma^2}$$

- 计算 $\frac{\partial \sigma^2}{\partial x_i}$:
  根据公式:

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2$$

  可得:

$$\frac{\partial \sigma^2}{\partial x_i} = \frac{2}{m}(x_i - \mu)$$

  上式中，当 $j = i$ 时，偏导数不为零。

我们将以上的分步骤推导的公式代入第一个公式，得到:

$$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial y_i}\gamma \cdot \frac{1}{\sqrt{\sigma^2+\epsilon}} + \sum_{j=1}^{m}\frac{\partial L}{\partial y_j}\gamma \cdot \frac{-1}{\sqrt{\sigma^2+\epsilon}} \cdot \frac{1}{m} + \sum_{j=1}^{m}\frac{\partial L}{\partial y_j}\gamma \cdot \frac{-\frac{1}{2}(x_j - \mu)}{(\sigma^2+\epsilon)^{\frac{3}{2}}} \cdot \frac{2}{m}(x_i - \mu)$$

经过整理，得到:

$$\frac{\partial L}{\partial x_i} = \frac{\gamma}{\sqrt{\sigma^2+\epsilon}}\left(\frac{\partial L}{\partial y_i} - \frac{1}{m}\sum_{j=1}^{m}\frac{\partial L}{\partial y_j} - \frac{1}{m} \cdot \hat{x}_i \cdot \sum_{j=1}^{m}\frac{\partial L}{\partial y_j} \cdot \hat{x}_j\right)$$

这里有个简化计算的办法:
因为:

$$\frac{\partial L}{\partial \gamma_i} = \sum_{j=1}^{m}\frac{\partial L}{\partial y_j}\hat{x}_j$$

以及:

$$\frac{\partial L}{\partial \beta_i} = \sum_{j=1}^{m}\frac{\partial L}{\partial y_j}$$

我们可以在计算: $\frac{\partial L}{\partial x_i}$,重复使用上述结果，代入即可得到:

$$\frac{\partial L}{\partial x_i} = \frac{\gamma}{\sqrt{\sigma^2+\epsilon}}\left(\frac{\partial L}{\partial y_i} - \frac{1}{m}\frac{\partial L}{\partial \beta_c} - \frac{1}{m} \cdot \hat{x}_i \cdot \frac{\partial L}{\partial \gamma_c}\right)$$

再考虑到 $\frac{1}{\sqrt{\sigma^2+\epsilon}}$ 在前向传播时已经计算完成且已经缓存，所以，计算 $\frac{\partial L}{\partial x_i}$ 显得不是那么复杂。