Derivative of BatchNorm and LayerNorm operation w.r.t X

Kadulin V.

October 17, 2022

Let $Y = \gamma\hat{X} + \beta$, where $\hat{X} \in \mathbb{R}^{N\times D}$, $\gamma \in \mathbb{R}^D$, $\beta \in \mathbb{R}^D$, $\hat{X} = \frac{X-\mu}{\sigma}$. Let's start from BatchNorm case, where $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$, $v = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$, $\sigma = \sqrt{v+\epsilon}$, $\epsilon \in \mathbb{R}$ is a small scalar to omit zero division.

Let $f : \mathbb{R}^{N\times D} \to \mathbb{R}$ is a differentiable scalar function. We also know $\frac{\partial f}{\partial Y}$. Derive $\frac{\partial f}{\partial X}$.

Notice that

- y_i depends on x_j, where $1 \le j \le N$
- there is only per-column dependency: y_{ij} depends on x_{kj}, where $1 \le k \le N$

Consider first column of X as x.

$$y = \gamma\hat{x} + \beta$$
$$y, x, \hat{x} \in \mathbb{R}^N$$
$$\gamma, \beta, \mu, v, \sigma \in \mathbb{R}$$

Let's consider j-th element of x as x_j.

$$\frac{\partial f}{\partial x_j} = \sum_{i=1}^{N} \frac{\partial f}{\partial y_i}\frac{\partial y_i}{\partial \hat{x}_i}\frac{\partial \hat{x}_i}{\partial x_j}$$

Let's derive expressions for building blocks:
$\frac{\partial y}{\partial \hat{x_i}} = \gamma$
$\frac{\partial \hat{x_i}}{\partial x_j} = \frac{\partial (x_i - \mu)}{\partial x_j}(v+\epsilon)^{-\frac{1}{2}} + (x_i - \mu)(-\frac{1}{2})(v+\epsilon)^{-\frac{3}{2}}\frac{\partial (v+\epsilon)}{\partial x_j}$
$\frac{\partial \mu}{\partial x_j} = \frac{1}{N}$
$\frac{\partial x_i}{\partial x_j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$
Let's derive $\frac{\partial v}{\partial x_j}$.

1 Forward-mode differentiation

Consider i-th component of sum in v as y_i.
$\frac{\partial y_i}{\partial x_j} = \begin{cases} 2(x_i - \mu)(1 - \frac{1}{N}) = -\frac{2}{N}(x_i - \mu) + 2(x_i - \mu) & i = j \\ -\frac{2}{N}(x_i - \mu) & i \neq j \end{cases}$
So, $\frac{\partial v}{\partial x_j} = \frac{1}{N}(-\frac{2}{N}\sum_{i=1}^{N}(x_i - \mu) + 2(x_j - \mu))$.
Note, that $\sum_{i=1}^{N}(x_i - \mu) = \sum_{i=1}^{N} x_i - N\mu = N\mu - N\mu = 0$.
So, $\frac{\partial v}{\partial x_j} = \frac{2}{N}(x_j - \mu)$.
Now we can expand $\frac{\partial \hat{x_i}}{\partial x_j}$. Let's do it in case $i \neq j$:

$$\frac{\partial \hat{x_i}}{\partial x_j} = -\frac{1}{N}(v+\epsilon)^{-\frac{1}{2}} + (x_i - \mu)(-\frac{1}{2})(v+\epsilon)^{-\frac{3}{2}}\frac{2}{N}(x_j - \mu) =$$
$$-\frac{1}{N}(v+\epsilon)^{-\frac{1}{2}}(1 + (x_i - \mu)(x_j - \mu)(v+\epsilon)^{-1})$$

Note that the only difference of case $i = j$ is the first multiplier of the first term: it's $1 - \frac{1}{N}$ instead of $-\frac{1}{N}$.

Now we have all to derive $\frac{\partial f}{\partial x_j}$:

$$\frac{\partial f}{\partial x_j} = \sum_{i=1}^{N} \frac{\partial f}{\partial y_i} \frac{\partial y_i}{\partial \hat{x}_i} \frac{\partial \hat{x}_i}{\partial x_j} = \gamma \sum_{i=1}^{N} \frac{\partial f}{\partial y_i} \frac{\partial \hat{x}_i}{\partial x_j} =$$

$$\gamma \left(\sum_{i=1}^{N} -\frac{1}{N} (v+\epsilon)^{-\frac{1}{2}} (1 + (x_i - \mu)(x_j - \mu)(v+\epsilon)^{-1}) \frac{\partial f}{\partial y_i} + (v+\epsilon)^{-\frac{1}{2}} \frac{\partial f}{\partial y_j} \right) =$$

$$\gamma (v+\epsilon)^{-\frac{1}{2}} \left(-\frac{1}{N} \sum_{i=1}^{N} (1 + (x_i - \mu)(x_j - \mu)(v+\epsilon)^{-1}) \frac{\partial f}{\partial y_i} + \frac{\partial f}{\partial y_j} \right) =$$

$$\gamma (v+\epsilon)^{-\frac{1}{2}} \left(-\frac{1}{N} \sum_{i=1}^{N} \frac{\partial f}{\partial y_i} - (x_j - \mu)(v+\epsilon)^{-1} \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu) \frac{\partial f}{\partial y_i} + \frac{\partial f}{\partial y_j} \right)$$

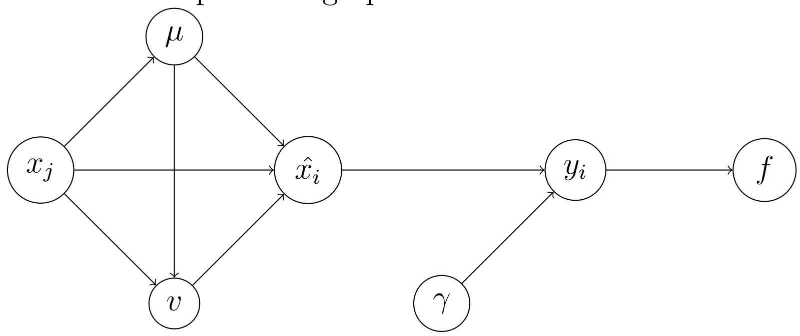
Knowing that, it's easy to write expression in matrix form:

$$\frac{\partial f}{\partial X} = \gamma (v+\epsilon)^{-\frac{1}{2}} \left(-\frac{1}{N} \sum_{i=1}^{N} \frac{\partial f}{\partial y_i} - (X - \mu)(v+\epsilon)^{-1} \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu) \frac{\partial f}{\partial y_i} + \frac{\partial f}{\partial Y} \right)$$

Here $x, y, v, \mu, \gamma \in \mathbb{R}^D$.

2 Backward-mode differentiation

Let's draw a computation graph:



Note that there are three paths from x_j and two paths from μ. So, derivative expressions through these nodes will consist of three and two terms respectively:

$$\frac{\partial f}{\partial x_j} = \sum_{i=1}^{N} \frac{\partial f}{\partial y_i} \frac{\partial y_i}{\partial \hat{x}_i} \left(\frac{\partial \hat{x}_i}{\partial x_j} + \frac{\partial \hat{x}_i}{\partial v} \frac{\partial v}{\partial x_j} + \left(\frac{\partial \hat{x}_i}{\partial \mu} + \frac{\partial \hat{x}_i}{\partial v} \frac{\partial v}{\partial \mu} \right) \frac{\partial \mu}{\partial x_j} \right)$$

$\frac{\partial y_i}{\partial \hat{x}_i} = \gamma$
$\frac{\partial \hat{x}_i}{\partial x_j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$
$\frac{\partial \hat{x}_i}{\partial v} = (x_i - \mu)(-\frac{1}{2})(v + \epsilon)^{-\frac{3}{2}}$
$\frac{\partial v}{\partial x_j} = \frac{2}{N}(x_j - \mu)$
$\frac{\partial \hat{x}_i}{\partial \mu} = -(v + \epsilon)^{-\frac{1}{2}}$
$\frac{\partial v}{\partial \mu} = -\frac{2}{N} \sum_{i=1}^{N}(x_i - \mu) = 0$
$\frac{\partial \mu}{\partial x_j} = \frac{1}{N}$
Now let's derive $\frac{\partial f}{\partial x_j}$:

$$\frac{\partial f}{\partial x_j} = \gamma \left(\sum_{i=1}^{N} \frac{\partial f}{\partial y_i} \left(-(x_i - \mu)(v + \epsilon)^{-\frac{3}{2}} \frac{1}{N}(x_j - \mu) - \frac{1}{N}(v + \epsilon)^{-\frac{1}{2}} \right) + (v + \epsilon)^{-\frac{1}{2}} \frac{\partial f}{\partial y_j} \right) =$$
$$\gamma(v + \epsilon)^{-\frac{1}{2}} \left(-\frac{1}{N} \sum_{i=1}^{N} \frac{\partial f}{\partial y_i} - (x_j - \mu)(v + \epsilon)^{-1} \frac{1}{N} \sum_{i=1}^{N}(x_i - \mu) \frac{\partial f}{\partial y_i} + \frac{\partial f}{\partial y_j} \right)$$

Note that result is the same as in previous approach.

3 LayerNorm

This operation differs from BatchNorm in a single aspect: μ and v are calculated by columns instead of rows. This implies the following changes:
$\frac{\partial y_i}{\partial \hat{x}_i} = \gamma_i$
$\frac{\partial \mu}{\partial x_j} = \frac{1}{D}$
So, the equation for LayerNorm operation looks as follows:

$$\frac{\partial f}{\partial X} = (v+\epsilon)^{-\frac{1}{2}} \left(-\frac{1}{D} \sum_{i=1}^{D} \gamma_i \frac{\partial f}{\partial y_i} - (X - \mu)(v + \epsilon)^{-1} \frac{1}{D} \sum_{i=1}^{D} \gamma_i (x_i - \mu) \frac{\partial f}{\partial y_i} + \gamma \frac{\partial f}{\partial Y} \right)$$

4 Other normalization methods

Now we know two normalization methods for 2-D case: BatchNorm and LayerNorm. But there are some details in case of images, which have a spatial structure. Consider a batch of images of shape $\mathbb{R}^{N \times C \times H \times W}$, where N - batch size, C - number of channels (or feature maps), H - height, W - width.

Here is a comparison of different normalization methods:

name	per	over	norm
BatchNorm	D	N	D
LayerNorm	N	D	D
Spatial BatchNorm	C	N, H, W	C
GroupNorm	N, G	C / G, H, W	C
InstanceNorm	N, C	H, W	C

- *per* - independent elements
- *over* - computing moments
- *norm* - scale and shift axis

Some notes about each method:

- BatchNorm – normalizes each feature independently. Quality of moments depends of batch size - higher is better.
- LayerNorm – computes statistics accross whole features. This fact makes it more preferable in case of small batches. Assumes equal contribution of each feature.
- Spatial BatchNorm – normalizes each feature map independently. Makes statistics consistent accross different images and image regions.
- GroupNorm – LayerNorm analogue for images, where aggregation is also done per channel groups: hypothesis is that feature maps are grouped by some factors like frequency, shapes, illumination, textures (examples from original paper). If so, each group might have different moments. Parametrized by G - number of groups of feature maps.
- InstanceNorm – special case of GroupNorm where $G = 1$.