Batch Normalization: alternative backward

In class we talked about two different implementations for the sigmoid backward pass. One strategy is to write out a computation graph composed of simple operations and backprop through all intermediate values. Another strategy is to work out the derivatives on paper. For the sigmoid function, it turns out that you can derive a very simple formula for the backward pass by simplifying gradients on paper.

Surprisingly, it turns out that you can also derive a simple expression for the batch normalization backward pass if you work out derivatives on paper and simplify. After doing so, implement the simplified batch normalization backward pass in the function `batchnorm_backward_alt` and compare the two implementations by running the following. Your two implementations should compute nearly identical results, but the alternative implementation should be a bit faster.

Draft for the solution

So this, time we want to find $\frac{dL}{d\gamma}$, $\frac{dL}{d\beta}$ and $\frac{dL}{dx}$ with

$$y = \gamma\hat{x} - \beta$$

where

$$\hat{x} = \frac{x-\mu}{\sqrt{\sigma^2+\epsilon}} = (x-\mu)(\sigma^2+\epsilon)^{-1/2}$$

Therefore, we note for the following that

$$y_{kl} = \gamma_l \hat{x_{kl}} - \beta_l$$

and

$$\hat{x_{kl}} = (x_{kl} - \mu_l)(\sigma_l^2 + \epsilon)^{-1/2}$$

where

$$\mu_l = \frac{1}{N}\sum_p x_{pl}$$

and

$$\sigma_l^2 = \frac{1}{N}\sum_p (x_{pl} - \mu_l)^2$$

Let's begin by the easy one !

$$\begin{aligned} \frac{dL}{d\gamma_j} &= \sum_{kl} \frac{dL}{dy_{kl}} \frac{dy_{kl}}{d\gamma_j} && (1) \\ &= \sum_{kl} \frac{dL}{dy_{kl}} x_{kl}\delta_{lj} && (2) \\ &= \sum_{k} \frac{dL}{dy_{kj}} x_{kj} && (3) \end{aligned}$$

For β we have

$$\frac{dL}{d\beta_j} = \sum_{kl} \frac{dL}{dy_{kl}} \frac{dy_{kl}}{d\beta_j} \tag{4}$$

$$= \sum_{kl} \frac{dL}{dy_{kl}} \delta_{lj} \tag{5}$$

$$= \sum_{k} \frac{dL}{dy_{kj}} \tag{6}$$

Ok. Let's start the serious one.

$$\frac{dL}{dx_{ij}} = \sum_{kl} \frac{dL}{dy_{kl}} \frac{dy_{kl}}{dx_{ij}} \tag{7}$$

$$= \sum_{kl} \frac{dL}{dy_{kl}} \frac{dy_{kl}}{d\hat{x}_{kl}} \frac{d\hat{x}_{kl}}{dx_{ij}} \tag{8}$$

where

$$\hat{x_{kl}} = (x_{kl} - \mu_l)(\sigma_l^2 + \epsilon)^{-1/2}$$

. First, we have:

$$\frac{dy_{kl}}{d\hat{x}_{kl}} = \gamma_l$$

and

$$\frac{d\hat{x}_{kl}}{dx_{ij}} = (\delta_{ik}\delta_{jl} - \frac{1}{N}\delta_{jl})(\sigma_l^2 + \epsilon)^{-1/2} - \frac{1}{2}(x_{kl} - \mu_l)\frac{d\sigma_l^2}{dx_{ij}}(\sigma_l^2 + \epsilon)^{-3/2} \tag{9}$$

where

$$\sigma_l^2 = \frac{1}{N}\sum_p (x_{pl} - \mu_l)^2$$

and then,

$$\frac{d\sigma_l^2}{dx_{ij}} = \frac{1}{N}\sum_p 2\left(\delta_{ip}\delta_{jl} - \frac{1}{N}\delta_{jl}\right)(x_{pl} - \mu_l) \tag{10}$$

$$= \frac{2}{N}(x_{il} - \mu_l)\delta_{jl} - \frac{2}{N^2}\sum_p \delta_{jl}(x_{pl} - \mu_l) \tag{11}$$

$$= \frac{2}{N}(x_{il} - \mu_l)\delta_{jl} \tag{12}$$

Putting everything together we thus have

$$\frac{d\hat{x}_{kl}}{dx_{ij}} = (\delta_{ik}\delta_{jl} - \frac{1}{N}\delta_{jl})(\sigma_l^2 + \epsilon)^{-1/2} - \frac{1}{N}(x_{kl} - \mu_l)(x_{il} - \mu_l)\delta_{jl}(\sigma_l^2 + \epsilon)^{-3/2} \tag{13}$$

and therefore

$$\frac{dL}{dx_{ij}} = \sum_{kl} \frac{dL}{dy_{kl}} \frac{dy_{kl}}{d\hat{x}_{kl}} \frac{d\hat{x}_{kl}}{dx_{ij}} \tag{14}$$

$$= \sum_{kl} \frac{dL}{dy_{kl}} \gamma_l \delta_{jl} \left[(\delta_{ik} - \frac{1}{N})(\sigma_l^2 + \epsilon)^{-1/2} - \frac{1}{N}(x_{kl} - \mu_l)(x_{il} - \mu_l)(\sigma_l^2 + \epsilon)^{-3/2} \right] \tag{15}$$

$$= \sum_{k} \frac{dL}{dy_{kj}} \gamma_j \left[(\delta_{ik} - \frac{1}{N})(\sigma_j^2 + \epsilon)^{-1/2} \right] - \sum_{k} \frac{dL}{dy_{kj}} \gamma_j \left[\frac{1}{N}(x_{kj} - \mu_j)(x_{ij} - \mu_j)(\sigma_j^2 + \epsilon)^{-3/2} \right] \tag{16}$$

$$= \frac{dL}{dy_{ij}} \gamma_j (\sigma_j^2 + \epsilon)^{-1/2} - \frac{1}{N} \sum_{k} \frac{dL}{dy_{kj}} \gamma_j (\sigma_j^2 + \epsilon)^{-1/2} \tag{17}$$

$$- \frac{1}{N} \sum_{k} \frac{dL}{dy_{kj}} \gamma_j \left[(x_{kj} - \mu_j)(x_{ij} - \mu_j)(\sigma_j^2 + \epsilon)^{-3/2} \right] \tag{18}$$

$$= \frac{1}{N} \gamma_j (\sigma_j^2 + \epsilon)^{-1/2} \left[N \frac{dL}{dy_{ij}} - \sum_{k} \frac{dL}{dy_{kj}} - (x_{ij} - \mu_j)(\sigma_j^2 + \epsilon)^{-1} \sum_{k} \frac{dL}{dy_{kj}} (x_{kj} - \mu_j) \right] \tag{19}$$

Python implementation

Here is the simple python code for Batch-Normalization:

```
# forward propagation part
xhat = (x - mu) / np.sqrt(var + eps)
y = gamma * xhat + beta

# backward part
dgamma = np.sum(dy * xhat, axis=0)
dbeta = np.sum(dy, axis=0)
dx = (dy - np.mean(dy, axis=0) - (x - mu) / (var + eps) * np.mean(dy * (x - mu), axis=0))
   * gamma / np.sqrt(var + eps)
```

Issue about how `epsilon` is defined

If ϵ is defined the other way to avoid deviding by zero:

$$\hat{x_{kl}} = (x_{kl} - \mu_l)(\sigma_l + \epsilon)^{-1}$$

Then, for the back propagation part, `dx` should be revised as follows:

```
# backward part
std = np.sqrt(var)
dx = (dy - np.mean(dy, axis=0) - (x - mu) / (std * (std + eps))
   * np.mean(dy * (x - mu), axis=0)) * gamma / np.sqrt(std + eps)
```