Batch Normalization

조희철

2018년 2월 20일

1 Batch Normalization

Batch Normalization Layer는 Activation Function 앞에 놓여진다. Batch Normalization의 backward propagation은 Computational Graph를 통해서 계산 할 수 있다.

Forward Propagation			Backward Propagation		
		X	$dX = dX_c + \frac{1}{N}d\mu$		
	↙	↓	↑	↖	
$\mu = \frac{1}{N}\sum X$					$d\mu = -\sum dX_c$
	↘	↓	↑	↗	
		$X_c = X - \mu$	$dX_c = \frac{1}{\sigma}dX_n + \frac{2X_c}{N}dV$		
	↙	↓	↑	↖	
$V = \frac{X_c^2}{N}$					$dV = \frac{d\sigma}{2\sigma}$
↓		↓	↑		↑
$\sigma = \sqrt{V}$					$d\sigma = -\sum \frac{X_c \circ dX_n}{\sigma^2}$
	↘	↓	↑	↗	
γ, β		$X_n = \frac{X_c}{\sigma}$	$dX_n = \gamma \circ dY$		$d\gamma = \sum X_n \circ dY$ $d\beta = \sum dY$
	↘	↓	↑	↗	
		$Y = \gamma X_n + \beta$	dY		

♠ **Back Propagation Speed Up 구현** [1]

N개의 data $x_1, \cdots, x_n (x_i \in \mathbb{R}^m)$에 대하여

$$\begin{aligned}
\mu &= \frac{1}{N}\sum_{i=1}^{N} x_i, \\
\sigma^2 &= \frac{1}{N}\sum_{i=1}^{n}(x_i - \mu)^2, \\
\bar{x_i} &= \frac{x_i - \mu}{\sigma}, \\
y_i &= \gamma \bar{x_i} + \beta.
\end{aligned}$$

이제 Loss Function L에 대한 gradient $\frac{\partial L}{\partial x_i}$를 계산해 보자.

$$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial \bar{x_i}}\frac{\partial \bar{x_i}}{\partial x_i} \quad + \quad \frac{\partial L}{\partial \sigma^2}\frac{\partial \sigma^2}{\partial x_i} \quad + \quad \frac{\partial L}{\partial \mu}\frac{\partial \mu}{\partial x_i}$$

$\frac{\partial L}{\partial x_i}$는 3개 식의 합으로 표현되는데, 각각을 나누어서 계산해 보자.

[1] `https://costapt.github.io/2016/07/09/batch-norm-alt/`

- 첫번째 식 $\frac{\partial L}{\partial \bar{x}_i}\ \frac{\partial \bar{x}_i}{\partial x_i}$ 의 두 식은 각각 다음과 같이 계산된다.

$$\begin{aligned}
\frac{\partial L}{\partial \bar{x}_i} &= \frac{\partial L}{\partial y_i}\ \frac{\partial y_i}{\partial \bar{x}_i} \\
&= \frac{\partial L}{\partial y_i}\ \gamma, \\
\frac{\partial \bar{x}_i}{\partial x_i} &= \frac{1}{\sigma}, \\
\frac{\partial L}{\partial \bar{x}_i}\ \frac{\partial \bar{x}_i}{\partial x_i} &= \frac{\gamma}{\sigma}\ \frac{\partial L}{\partial y_i}.
\end{aligned}$$

- 두번째 식 $\frac{\partial L}{\partial \sigma^2}\ \frac{\partial \sigma^2}{\partial x_i}$:

$$\begin{aligned}
\frac{\partial L}{\partial \sigma^2} &= \sum_i^N \frac{\partial L}{\partial \bar{x}_i}\ \frac{\partial \bar{x}_i}{\partial \sigma^2} \\
&= \sum_i^N \frac{\partial L}{\partial y_i}\ \gamma\ \ (x_i - \mu)\big(-\frac{1}{2\sigma^3}\big) \\
&= -\frac{\gamma}{2\sigma^3} \sum_i^N \frac{\partial L}{\partial y_i}\ (x_i - \mu), \\
\frac{\partial L}{\partial \sigma^2}\frac{\partial \sigma^2}{\partial x_i} &= -\frac{\gamma}{2\sigma^3}\Big[\sum_j^N \frac{\partial L}{\partial y_j}\ (x_j - \mu)\Big]\ \Big(\frac{2}{N}(x_i - \mu)\Big) \\
&= -\frac{\gamma}{N\sigma}\big[\sum_j^N \frac{\partial L}{\partial y_j}\bar{x}_j\big]\bar{x}_i \quad \leftarrow \frac{\partial L}{\partial \gamma} = \sum_j^N \frac{\partial L}{\partial y_j}\bar{x}_j \\
&= -\frac{\gamma}{N\sigma}\frac{\partial L}{\partial \gamma}\bar{x}_i.
\end{aligned}$$

- 마지막 세번째 식 $\frac{\partial L}{\partial \mu}\ \frac{\partial \mu}{\partial x_i}$:

$$\begin{aligned}
\frac{\partial L}{\partial \mu} &= \sum_{i=1}^N \frac{\partial L}{\partial \bar{x}_i}\frac{\partial \bar{x}_i}{\partial \mu}\ +\ \frac{\partial L}{\partial \sigma^2}\ \frac{\partial \sigma^2}{\partial \mu} \\
&= \sum_{i=1}^N \frac{\partial L}{\partial y_i}\ \gamma\ \big(-\frac{1}{\sigma}\big) + \frac{\partial L}{\partial \sigma^2}\sum_{i=1}^N -\frac{2}{N}(x_i - \mu) \\
&= -\frac{\gamma}{\sigma}\sum_{i=1}^N \frac{\partial L}{\partial y_i}, \\
\frac{\partial L}{\partial \mu}\ \frac{\partial \mu}{\partial x_i} &= -\frac{\gamma}{N\sigma}\sum_{i=1}^N \frac{\partial L}{\partial y_i} \\
&= -\frac{\gamma}{N\sigma}\frac{\partial L}{\partial \beta} \quad \leftarrow \frac{L}{\beta} = \sum_{i=1}^N \frac{\partial L}{\partial y_i}.
\end{aligned}$$

- 이제 세 식을 합치면

$$\frac{\partial L}{\partial x_i} = \frac{\gamma}{N\sigma}\Big(N\frac{\partial L}{\partial y_i} - \frac{\partial L}{\partial \gamma}\bar{x}_i - \frac{\partial L}{\partial \beta}\Big)$$

```
def batchnorm_backward_alt(dout, cache):
    gamma, xhat, istd = cache # istd = 1/std, xhat = Xn
    N, _ = dout.shape

    dbeta = np.sum(dout, axis=0)
    dgamma = np.sum(xhat * dout, axis=0)
    dx = (gamma*istd/N) * (N*dout - xhat*dgamma - dbeta)

    return dx, dgamma, dbeta
```