COMPSCI 682 Additional Session

A closer look at the math inside batch normalization

Hang Su
10/04/2019

Today's Plan

- Recap of the motivation and design considerations of batch normalization
- Detailed derivation of the backward pass derivatives

Read the paper (URL)

> *Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift."* arXiv preprint arXiv:1502.03167 *(2015).*

Internal Covariate Shift

- When the input distribution to a learning system changes, it is said to experience **covariate shift** (Shimodaira, 2000).
- Internal covariate shift: "*the change in the distribution of network activations due to the change in network parameters during training*"
- Solution: whiten the inputs (to internal layers)

Benefits

- Faster training: higher learning rate can be afforded

- Less sensitive to careful initialization
- Some extent of regularization
- Less prone to saturated modes (only relevant with saturating nonlinearities)

Two Simplifications

1. Batch **Normalization**

Instead of whitening, normalize each scalar element individually:

$$\hat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

2. **Batch** Normalization

Estimate $\mathrm{E}[x^{(k)}]$ and $\mathrm{Var}[x^{(k)}]$ with mini-batch statistics

Another detail ...

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)}$$

Foward Pass

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

$\frac{\partial x}{\partial y}$

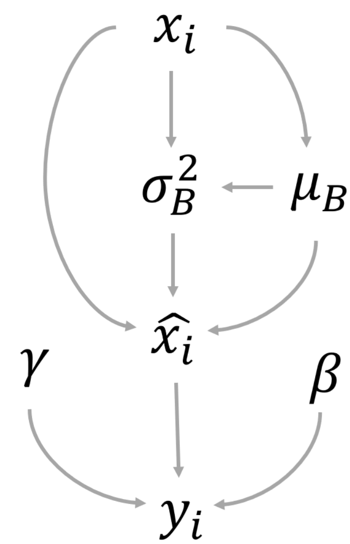
- "gradient of x with regard to y"
- *how much x will change in proportion to y*
- **... at current position**

- More on board ...

Backward Pass

- Input: $\frac{\partial l}{\partial y_i}$,
  and all things cached in forward pass (x_i, μ_B, σ_B^2
  ...)

- Output: $\frac{\partial l}{\partial x_i}, \frac{\partial l}{\partial \gamma}, \frac{\partial l}{\partial \beta}$



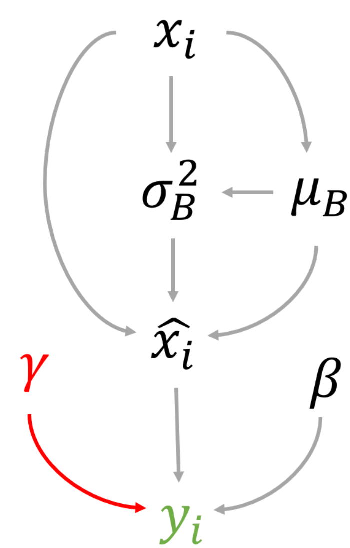
Foward pass:

> $\mu_B = \frac{1}{m}\sum x_i$
> $\sigma_B^2 = \frac{1}{m}\sum (x_i - \mu_B)^2$
> $\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$
> $y_i = \gamma \hat{x}_i + \beta$

Computing $\frac{\partial l}{\partial \gamma}$

> $\mu_B = \frac{1}{m}\sum x_i\,,\quad \sigma_B^2 = \frac{1}{m}\sum (x_i - \mu_B)^2\,,\quad \hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}\,,\quad y_i = \gamma \hat{x}_i + \beta$

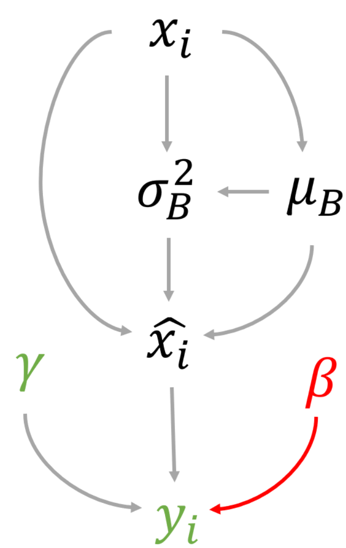
$$\frac{\partial l}{\partial \gamma} = \sum \frac{\partial l}{\partial y_i} \hat{x}_i$$



Computing $\frac{\partial l}{\partial \beta}$

> $\mu_B = \frac{1}{m}\sum x_i\,,\quad \sigma_B^2 = \frac{1}{m}\sum (x_i - \mu_B)^2\,,\quad \hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}\,,\quad y_i = \gamma \hat{x}_i + \beta$

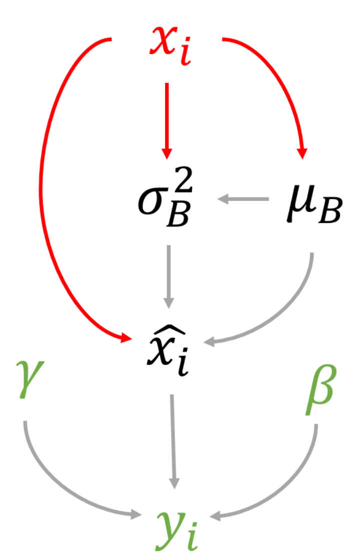
$$\frac{\partial l}{\partial \beta} = \sum \frac{\partial l}{\partial y_i}$$



Computing $\frac{\partial l}{\partial x_i}$

> $\mu_B = \frac{1}{m}\sum x_i\,,\quad \sigma_B^2 = \frac{1}{m}\sum (x_i - \mu_B)^2\,,\quad \hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}\,,\quad y_i = \gamma \hat{x}_i + \beta$

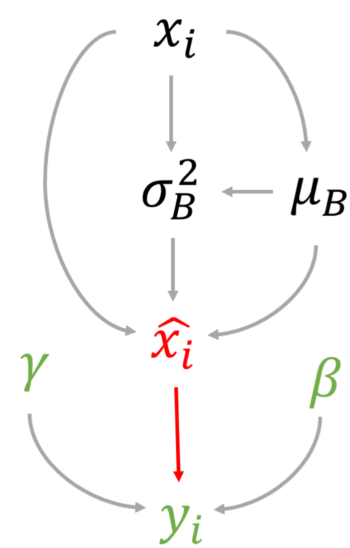
Not ready yet ...



Computing $\frac{\partial l}{\partial \hat{x}_i}$

> $\mu_B = \frac{1}{m}\sum x_i\,,\quad \sigma_B^2 = \frac{1}{m}\sum (x_i - \mu_B)^2\,,\quad \hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}\,,\quad y_i = \gamma \hat{x}_i + \beta$

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

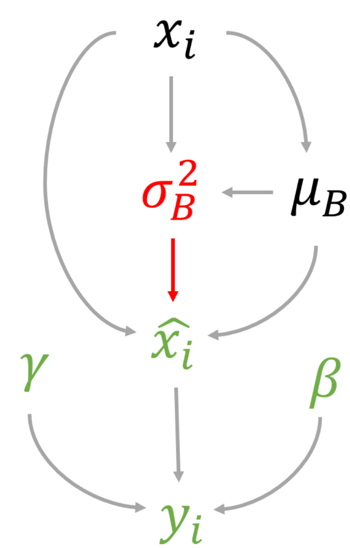


Computing $\frac{\partial l}{\partial \sigma_B^2}$

> $\mu_B = \frac{1}{m}\sum x_i\,,\quad \sigma_B^2 = \frac{1}{m}\sum(x_i-\mu_B)^2\,,\quad \hat{x}_i = \frac{x_i-\mu_B}{\sqrt{\sigma_B^2+\epsilon}}\,,\quad y_i = \gamma\hat{x}_i+\beta$

$$\frac{\partial l}{\partial \sigma_B^2} = \sum \frac{\partial l}{\partial \hat{x}_i}\frac{\partial \hat{x}_i}{\partial \sigma_B^2}$$

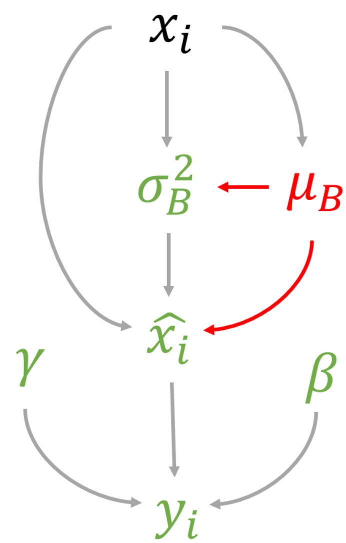
$$= \sum \frac{\partial l}{\partial \hat{x}_i}\frac{-1}{2}(x_i-\mu_B)(\sigma_B^2+\epsilon)^{-3/2}$$



Computing $\frac{\partial l}{\partial \mu_B}$

> $\mu_B = \frac{1}{m}\sum x_i\,,\quad \sigma_B^2 = \frac{1}{m}\sum(x_i-\mu_B)^2\,,\quad \hat{x}_i = \frac{x_i-\mu_B}{\sqrt{\sigma_B^2+\epsilon}}\,,\quad y_i = \gamma\hat{x}_i+\beta$

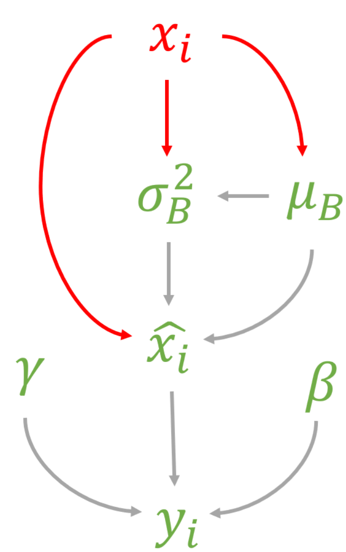
$$\frac{\partial l}{\partial \mu_B} = \frac{\partial l}{\partial \sigma_B^2}\frac{\partial \sigma_B^2}{\partial \mu_B} + \sum \frac{\partial l}{\partial \hat{x}_i}\frac{\partial}{\partial \mu_B}\left(\frac{x_i-\mu_B}{\sqrt{\sigma_B^2+\epsilon}}\right)$$
$$= \frac{\partial l}{\partial \sigma_B^2}\frac{\sum -2(x_i-\mu_B)}{m} + \sum \frac{\partial l}{\partial \hat{x}_i}\frac{-1}{\sqrt{\sigma_B^2+\epsilon}}$$



Computing $\frac{\partial l}{\partial x_i}$

> $\mu_B = \frac{1}{m}\sum x_i$, $\sigma_B^2 = \frac{1}{m}\sum (x_i-\mu_B)^2$, $\hat{x}_i = \frac{x_i-\mu_B}{\sqrt{\sigma_B^2+\epsilon}}$, $y_i = \gamma \hat{x}_i + \beta$

$$\frac{\partial l}{\partial x_i} = \frac{\partial l}{\partial \hat{x}_i}\frac{\partial}{\partial x_i}\left(\frac{x_i-\mu_B}{\sqrt{\sigma_B^2+\epsilon}}\right)$$
$$+\frac{\partial l}{\partial \sigma_B^2}\frac{\partial}{\partial x_i}\left(\frac{1}{m}\sum (x_i-\mu_B)^2\right) + \frac{\partial l}{\partial \mu_B}\frac{\partial \mu_B}{\partial x_i}$$
$$= \frac{\partial l}{\partial \hat{x}_i}\frac{1}{\sqrt{\sigma_B^2+\epsilon}} + \frac{\partial l}{\partial \sigma_B^2}\frac{2(x_i-\mu_B)}{m} + \frac{\partial l}{\partial \mu_B}\frac{1}{m}$$



Backward Pass: Summary

$$\frac{\partial l}{\partial \gamma} = \sum \frac{\partial l}{\partial y_i}\hat{x}_i$$
$$\frac{\partial l}{\partial \beta} = \sum \frac{\partial l}{\partial y_i}$$
$$\frac{\partial l}{\partial \hat{x}_i} = \frac{\partial l}{\partial y_i}\gamma$$
$$\frac{\partial l}{\partial \sigma_B^2} = \sum \frac{\partial l}{\partial \hat{x}_i}\frac{-1}{2}(x_i-\mu_B)(\sigma_B^2+\epsilon)^{-3/2}$$

$$\frac{\partial l}{\partial \mu_B} = \frac{\partial l}{\partial \sigma_B^2} \frac{\sum -2(x_i - \mu_B)}{m} + \sum \frac{\partial l}{\partial \hat{x}_i} \frac{-1}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$\frac{\partial l}{\partial x_i} = \frac{\partial l}{\partial \hat{x}_i} \frac{1}{\sqrt{\sigma_B^2 + \epsilon}} + \frac{\partial l}{\partial \sigma_B^2} \frac{2(x_i - \mu_B)}{m} + \frac{\partial l}{\partial \mu_B} \frac{1}{m}$$

← further simplify (on board)

A formula for designing new layers

1. Brainstorm an operation that might be useful
2. Design its forward pass behavior
3. Derive its backward pass derivatives
4. Optimize derivatives computation (simplification, native GPU impl. etc.)

- (3) is no longer necessary with most NN libraries
- (4) is typically only done after initial positive results