

Welcome

Brian Wu



No problem

Hey,big brother,can I move my company to your “silent sheep” hole ? I’ve heard they can work 996 extensively at very low price without complaint

Welcome,my good friend,but for safety reasons,data center should be in my country and source code open for checking



Wake up!!!



Hey!,We are 草泥马

The motivation

- Help people who have the same trouble as me and save their time
- It is necessary when you want to add layers to existing models in darknet framework



Angel plays with angel



,devil with devil

Give to Caesar what is Caesar's,to God what is God's.

Agenda

- Back-prop at convolution layer
 - Intuitive way
 - Derivation from math
- Batch normalization
 - Intro to Batch normalization
 - Derivation from math



How come? You got super skills

The company AI is moving to “silent sheep hole”, they are smart and cheap, even work like human robots,

Damn global economy with devil joining the game.
Capital and consumers tend to be short sighted

Dad, I lost my job



Cross-correlation vs Convolution

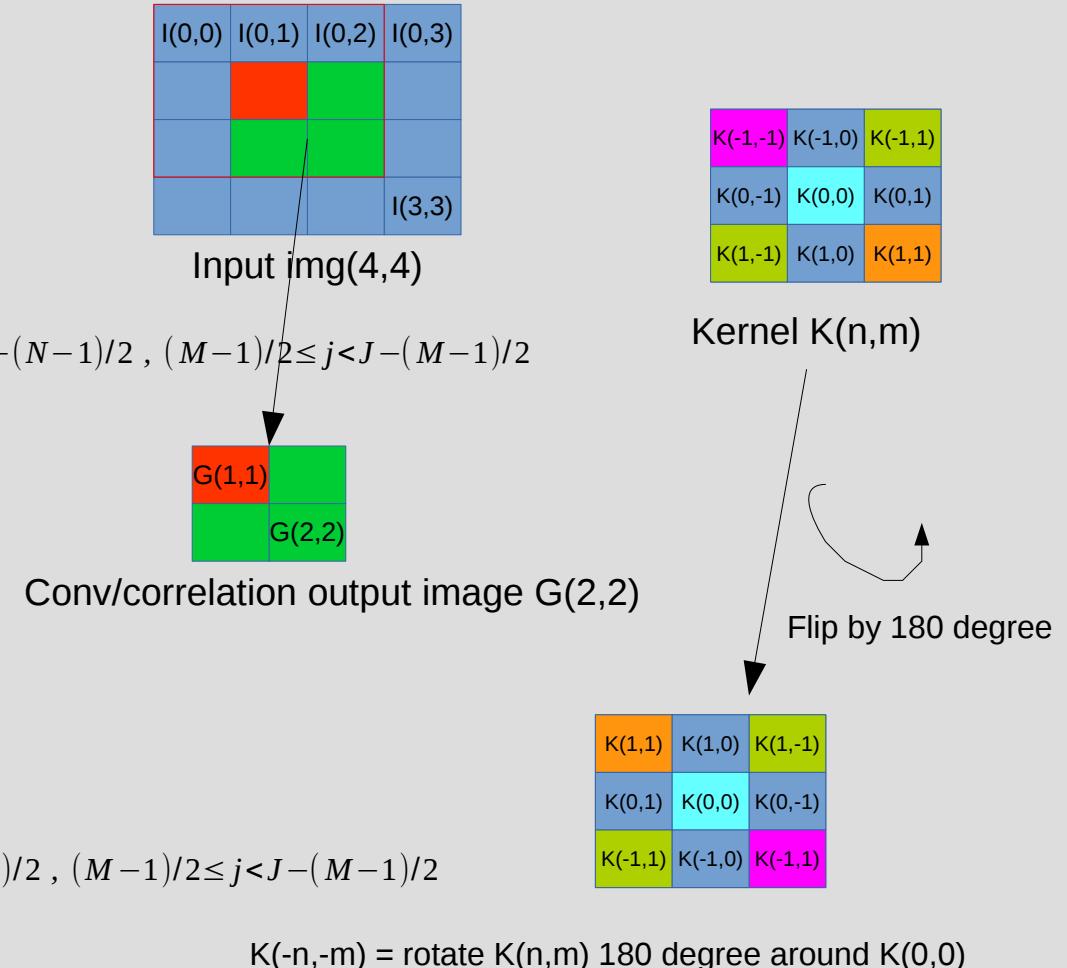
- Given image $\text{img}(I, J)$, and kernel $K(N, M), N, M$ is odd, $N < I, M < J$.
- Cross-Correlation : finding similarity

$$G(i, j) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} \text{img}(i+n, j+m) * K(n, m) \quad \text{where } (N-1)/2 \leq i < I - (N-1)/2, (M-1)/2 \leq j < J - (M-1)/2$$

- Convolution : filtering operation

$$G(i, j) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} \text{img}(i-n, j-m) * K(n, m)$$

$$= \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} \text{img}(i+n, j+m) * K(-n, -m) \quad \text{where } (N-1)/2 \leq i < I - (N-1)/2, (M-1)/2 \leq j < J - (M-1)/2$$



Flip the kernel, convolution becomes correlation

$$C_{k,i,j}^l = \sum_{c=0}^{inCHs-1} \sum_{n=-(N-1)/2}^{(N-1)/2} \sum_{m=-(M-1)/2}^{(M-1)/2} a^{l-1}(c, i+n, j+m) * K_k'(c, -n, -m) + b_k$$

where $0 \leq c < inputChannels$, k is k -th output channel index, $(N-1)/2 \leq i < I - (N-1)/2$, $(M-1)/2 \leq j < J - (M-1)/2$

Flipping the kernel, and reindexing, let $K_k'(c, -n, -m) = K_k(c, n, m)$

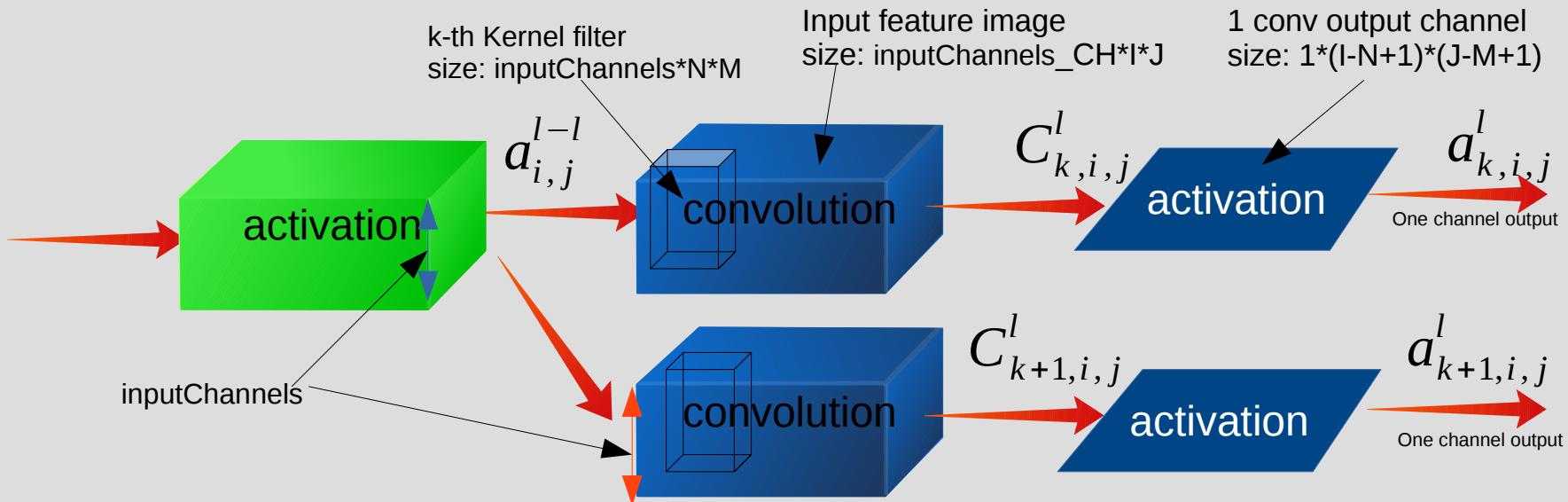
$$C_{k,i,j}^l = \sum_{c=0}^{inCHs-1} \sum_{n=-(N-1)/2}^{(N-1)/2} \sum_{m=-(M-1)/2}^{(M-1)/2} a^{l-1}(c, i+n, j+m) * K_k(c, n, m) + b_k$$

where $0 \leq c < inputChannels$, k is k -th output channel index, $(N-1)/2 \leq i < I - (N-1)/2$, $(M-1)/2 \leq j < J - (M-1)/2$

c is the c -th input channel number.

k is the k -th filter for the k -th output, l is the layer index

Convolution layer & formula



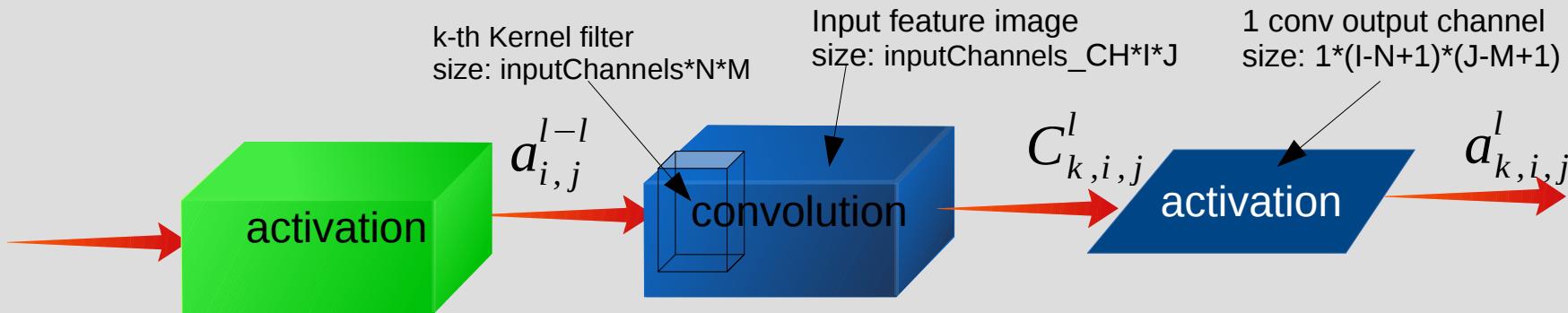
$$C_{k,i,j}^l = \sum_{c=0}^{inCHs-1} \sum_{n=-\frac{(N-1)}{2}}^{\frac{(N-1)}{2}} \sum_{m=-\frac{(M-1)}{2}}^{\frac{(M-1)}{2}} a^{l-1}(c, i+n, j+m) * K_k(c, n, m) + b_k$$

where $0 \leq c < \text{inputChannels}$, k is k-th output channel index, $(N-1)/2 \leq i < I - (N-1)/2$, $(M-1)/2 \leq j < J - (M-1)/2$

c is the c-th input channel number.

k is the k-th filter for the k-th output, l is the layer index

Back-pro at conv layer, objective & given



Objective:

- Derivative w.r.t conv inputs : $\frac{\partial \text{Loss}}{\partial a^{l-1}(cin, i', j')}$ where $0 \leq i' < I, 0 \leq j' < J$ → Prepare for back-pro for previous layer
- Derivative w.r.t weights : $\frac{\partial \text{Loss}}{\partial K^l(cin, n, m)}$ → $K(n, m) = K(n, m) + \text{learning_rate} * \frac{\partial \text{Loss}}{\partial K(n, m)}$

Given:

- Derivative w.r.t conv outputs, the output sensitive map :

$$\delta_{k,i,j}^l = \frac{\partial \text{Loss}}{\partial C^l(k, i, j)} \quad k \text{ is } k\text{-th output channel index. } 0 \leq i < I-1-N+1, 0 \leq j < J-1-M+1$$

Derivative w.r.t conv input $\frac{\partial Loss}{\partial a^{l-1}(cin, i', j')}$

$$C_{k,i,j}^l = \sum_{c=0}^{inCHs-1} \sum_{n=-\frac{(N-1)}{2}}^{\frac{(N-1)}{2}} \sum_{m=-\frac{(M-1)}{2}}^{\frac{(M-1)}{2}} a^{l-1}(c, i+n, j+m) * K_k(c, n, m) + b_k$$

where $0 \leq c < inputChannels$, k is k -th output channel index, $(N-1)/2 \leq i < I-1-(N-1)/2$, $(M-1)/2 \leq j < J-1-(M-1)/2$

By chain rule: $\frac{\partial Loss}{\partial a^{l-1}(cin, i', j')} = \frac{\partial Loss}{\partial C^l(k, i, j)} * \frac{\partial C^l(k, i, j)}{\partial a^{l-1}(cin, i', j')}$

$$\frac{\partial Loss}{\partial a^{l-1}(cin, i', j')} = \sum_{k=0}^{outputChannels-1} \sum_{i=\frac{N-1}{2}}^{I-1-\frac{N-1}{2}} \sum_{j=\frac{M-1}{2}}^{J-1-\frac{M-1}{2}} \frac{\partial Loss}{\partial C^l(k, i, j)} * \frac{\partial C^l(k, i, j)}{\partial a^{l-1}(cin, i', j')}$$

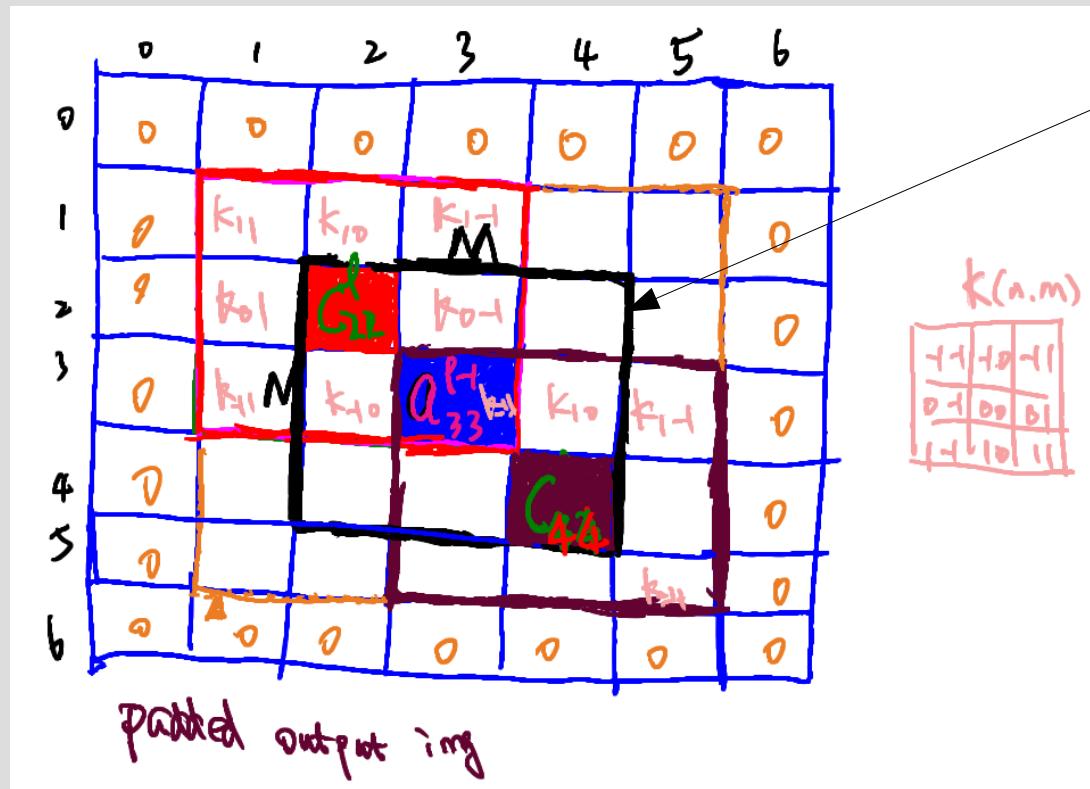
Note that: $(N-1)/2 \leq i < I-1-(N-1)/2$, $(M-1)/2 \leq j < J-1-(M-1)/2$, i is the output index, $0 \leq i' < l$, $0 \leq j' < J$, j' is the output index

But not all output pixel $C^l_{(k,i,j)}$ is related to input pixel $a^{l-1}_{(cin, i', j')}$ only these within the $M \times N$ filter bounding box are, as illustrated below:

$$\partial Loss$$

$$\frac{\partial Loss}{\partial a^{l-1}(cin, i', j')}$$

Derivative w.r.t conv input cont.



Only convolution output $C^l_{(k,i,j)}$ inside the $N \times M$ box are related to input $a^{l-1}_{(cin, 3, 3)}$

In general, $C^l_{(k,i' \pm \frac{N-1}{2}, j' \pm \frac{M-1}{2})}$ is related to input pixel $a^{l-1}_{(cin, i', j')}$

Derivative w.r.t conv input $\frac{\partial Loss}{\partial a^{l-1}(cin, i', j')}$ cont.

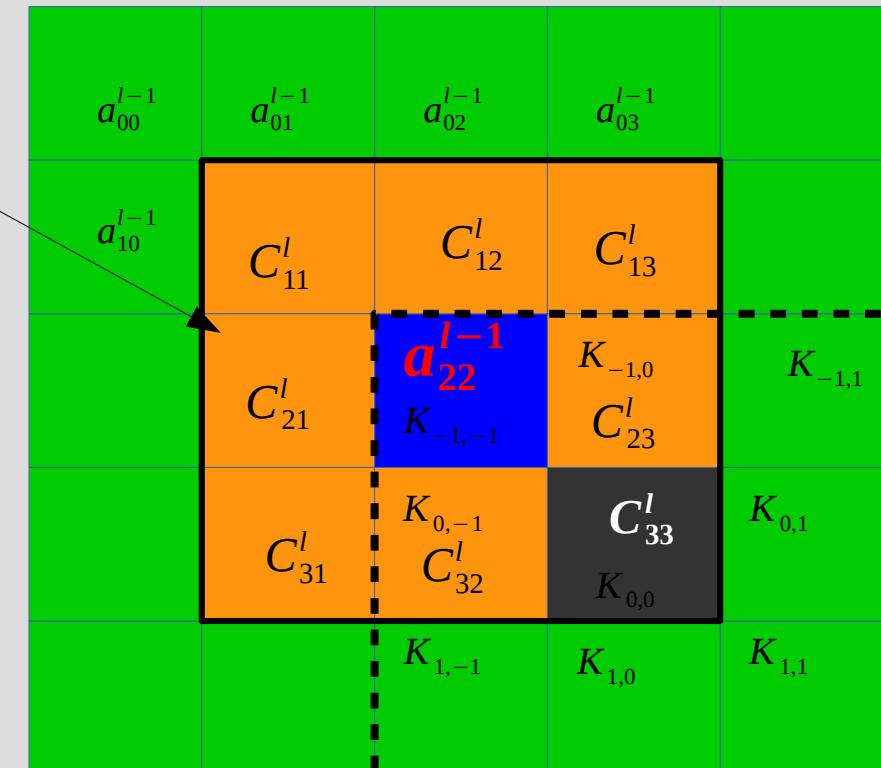
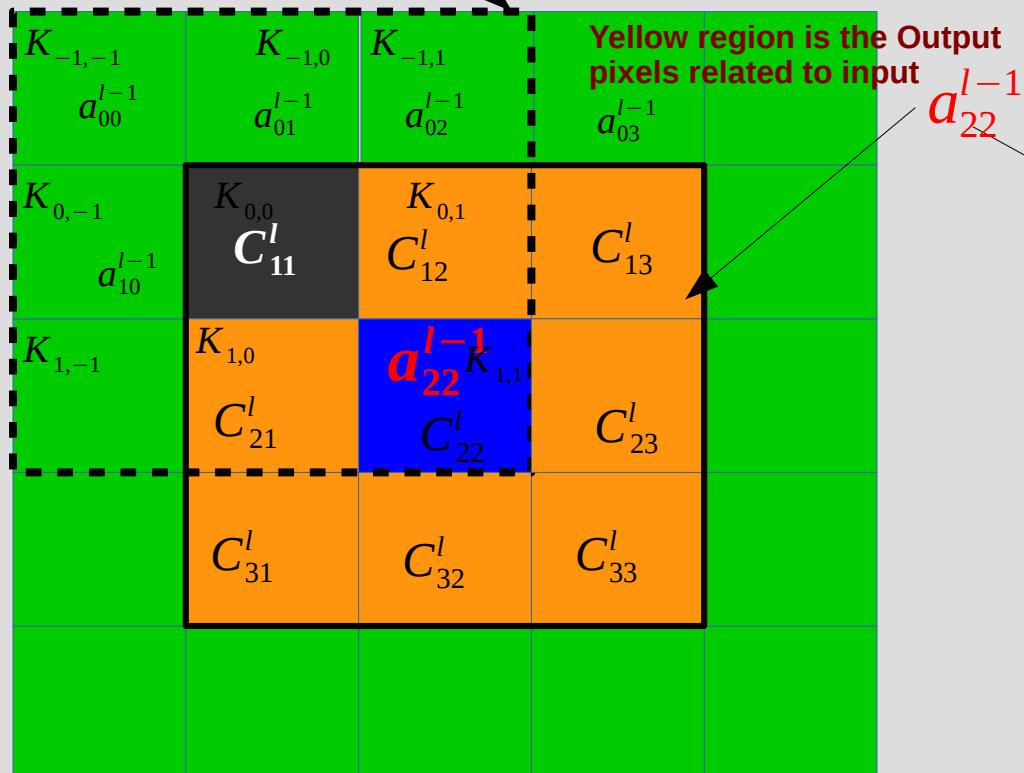
Let's start with simple case, one channel in, one channel out. Input 5*5, kernel 3*3, output

5*5 padded with 0

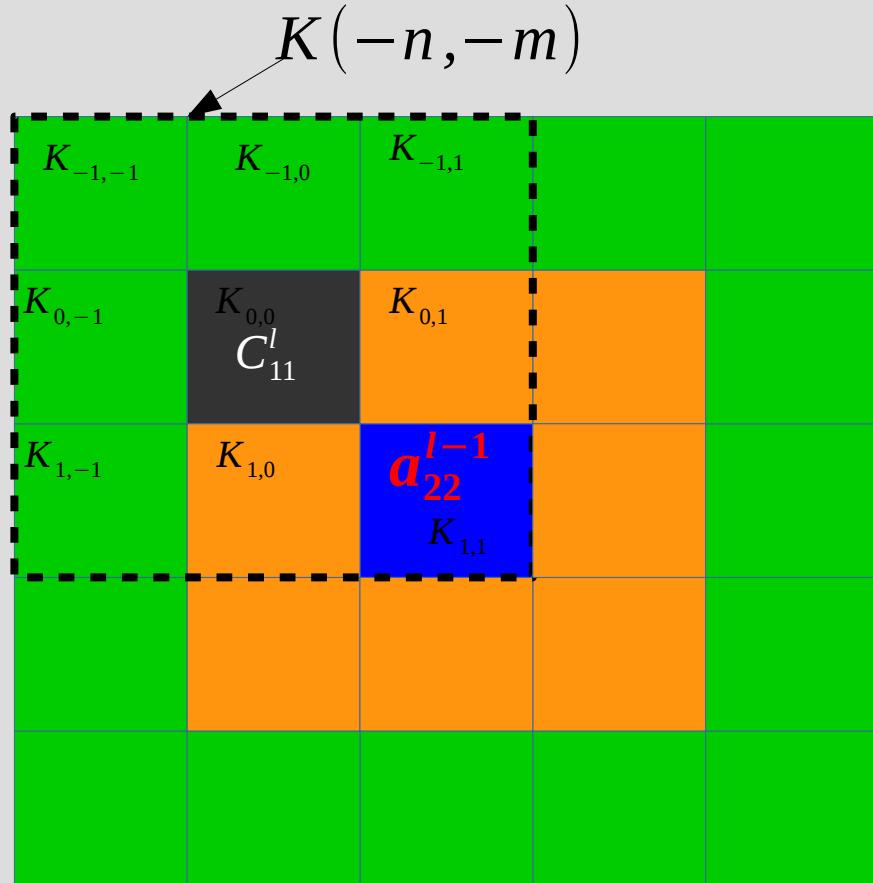
$$\frac{\partial Loss}{\partial a^{l-1}(2,2)} = \frac{\partial Loss}{\partial C_{11}^l} * \frac{\partial C_{11}^l}{\partial a_{22}^{l-1}} + \frac{\partial Loss}{\partial C_{12}^l} * \frac{\partial C_{12}^l}{\partial a_{22}^{l-1}} \\ \dots \\ + \dots + \frac{\partial Loss}{\partial C_{33}^l} * \frac{\partial C_{33}^l}{\partial a_{22}^{l-1}}$$

- █ green is the input data region
- █ yellow is the conv outputs related to input
- █ Black is the current output pixel generated by current inputs inside dash box

Sliding flipped kernel



Derivative w.r.t conv input $\frac{\partial Loss}{\partial a^{l-1}(cin, i', j')}$ cont.

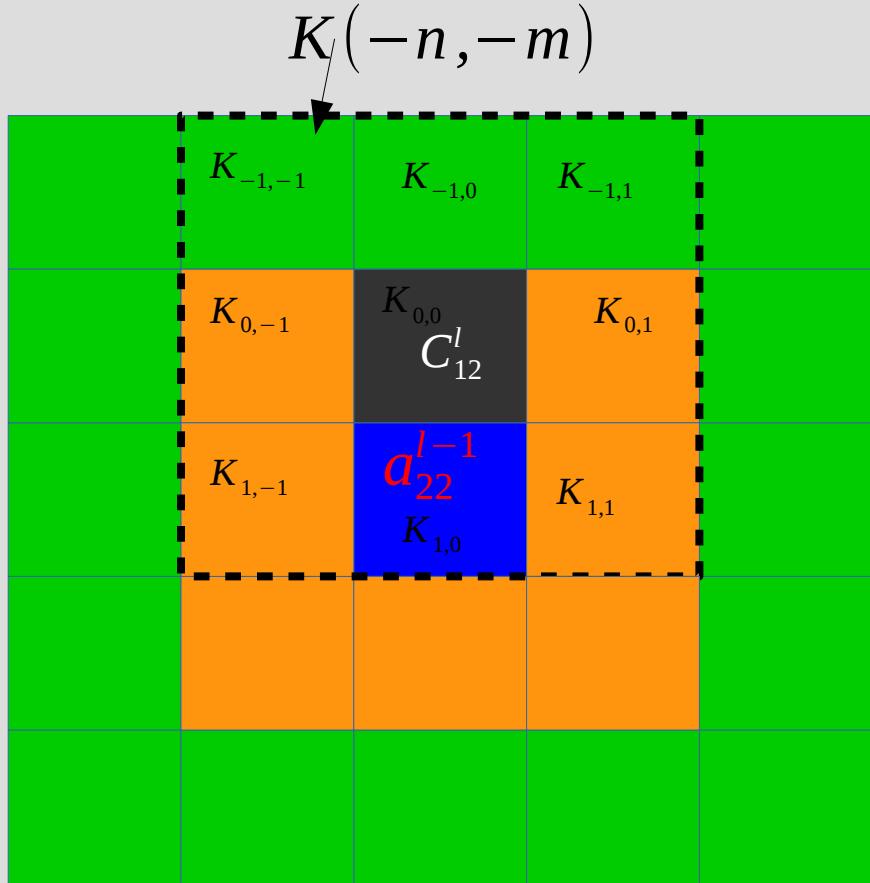


$$\frac{\partial Loss}{\partial a_{22}^{l-1}} = \boxed{\frac{\partial Loss}{\partial C_{11}^l} * \frac{\partial C_{11}^l}{\partial a_{22}^{l-1}}} + \frac{\partial Loss}{\partial C_{12}^l} * \frac{\partial C_{12}^l}{\partial a_{22}^{l-1}} + \dots + \frac{\partial Loss}{\partial C_{33}^l} * \frac{\partial C_{33}^l}{\partial a_{22}^{l-1}}$$

\downarrow

$$\frac{\partial Loss}{\partial C_{11}^l} * \frac{\partial C_{11}^l}{\partial a_{22}^{l-1}} = \delta^l(1,1) * K(1,1)$$

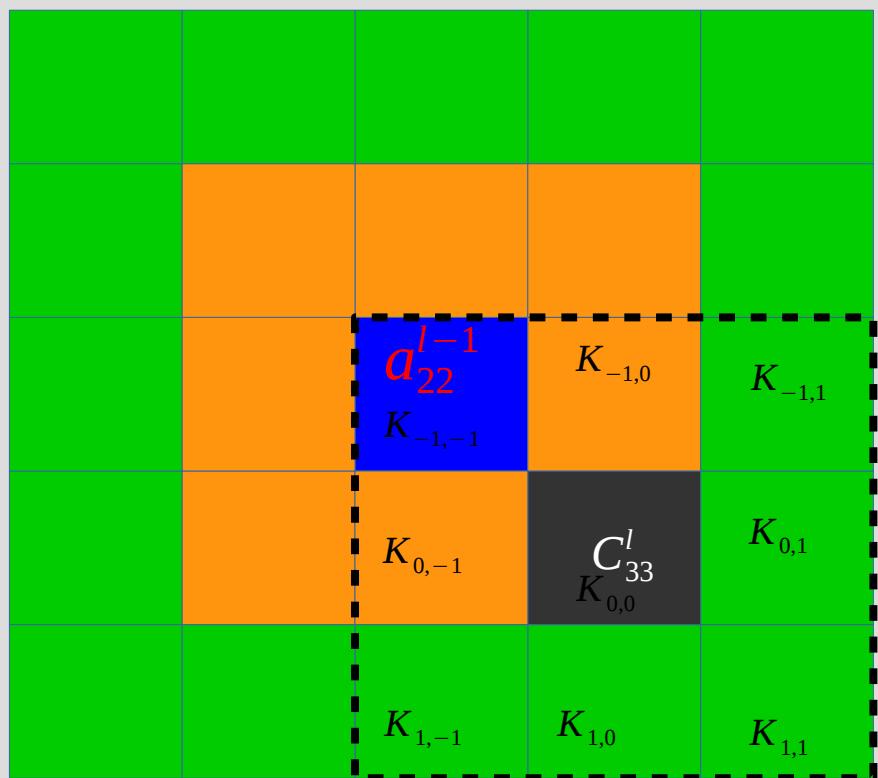
Derivative w.r.t conv input $\frac{\partial Loss}{\partial a^{l-1}(cin, i', j')}$ cont.



$$\frac{\partial Loss}{\partial a_{22}^{l-1}} = \frac{\partial Loss}{\partial C_{11}^l} * \frac{\partial C_{11}^l}{\partial a_{22}^{l-1}} + \boxed{\frac{\partial Loss}{\partial C_{12}^l} * \frac{\partial C_{12}^l}{\partial a_{22}^{l-1}} + \dots + \frac{\partial Loss}{\partial C_{33}^l} * \frac{\partial C_{33}^l}{\partial a_{22}^{l-1}}}$$

$$\frac{\partial Loss}{\partial C_{11}^l} * \frac{\partial C_{11}^l}{\partial a_{22}^{l-1}} = \delta^l(1,1) * K(1,0)$$

Derivative w.r.t conv input $\frac{\partial Loss}{\partial a^{l-1}(cin, i', j')}$ cont.



$$\frac{\partial Loss}{\partial a^{l-1}_{22}} = \frac{\partial Loss}{\partial C_{11}^l} * \frac{\partial C_{11}^l}{\partial a^{l-1}_{22}} + \frac{\partial Loss}{\partial C_{12}^l} * \frac{\partial C_{12}^l}{\partial a^{l-1}_{22}} + \dots + \frac{\partial Loss}{\partial C_{33}^l} * \frac{\partial C_{33}^l}{\partial a^{l-1}_{22}}$$

$$\frac{\partial Loss}{\partial C_{11}^l} * \frac{\partial C_{11}^l}{\partial a^{l-1}_{22}} = \delta^l(1,1) * K(-1,-1)$$

Derivative w.r.t conv input $\frac{\partial Loss}{\partial a^{l-1}(cin, i', j')}$ **cont.**

$$\begin{aligned}\frac{\partial Loss}{\partial a_{22}^{l-1}} &= \frac{\partial Loss}{\partial C_{11}^l} * \frac{\partial C_{11}^l}{\partial a_{22}^{l-1}} + \frac{\partial Loss}{\partial C_{12}^l} * \frac{\partial C_{12}^l}{\partial a_{22}^{l-1}} + \dots + \frac{\partial Loss}{\partial C_{33}^l} * \frac{\partial C_{33}^l}{\partial a_{22}^{l-1}} \\ &= \delta^l(1,1)*K(1,1) + \delta^l(1,2)*K(1,0) + \dots + \delta^l(3,3)*K(-1,-1)\end{aligned}$$



$$\frac{\partial Loss}{\partial a^{l-1}(i', j')} = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} \delta^l(i'+n, j'+m)*K(-n, -m)$$

Next step : derivation from math, see if they agree with each other

Heavy math coming next

C nation's foreign currency pool



we've heard you print good money !

But not \$, by law you must turn in your \$ to our nation's foreign currency pool !

By law that's our private property !

In exchange, we print you same amount of our own currency, plus some rewards in cash to reduce your tax ! Sounds good ?

The money you are printing? Hell no ! don't fool us like idiots !

Which you are supposed to be educated as !

Luckily we are 草泥马 !

Derivative w.r.t conv input $\frac{\partial Loss}{\partial a^{l-1}(cin, i', j')}$ cont.

$$\frac{\partial Loss}{\partial a^{l-1}(cin, i', j')} = \sum_{k=0}^{outputChannels-1} \sum_{i=\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{j=\frac{M-1}{2}}^{\frac{M-1}{2}} \frac{\partial Loss}{\partial C^l(k, i, j)} * \frac{\partial C^l(k, i, j)}{\partial a^{l-1}(cin, i', j')}$$

$$= \sum_{k=0}^{outputChannels-1} \sum_{i=i'-\frac{N-1}{2}}^{i'+\frac{N-1}{2}} \sum_{j=j'-\frac{M-1}{2}}^{j'+\frac{M-1}{2}} \frac{\partial Loss}{\partial C^l(k, i, j)} * \frac{\partial C^l(k, i, j)}{\partial a^{l-1}(cin, i', j')}$$

Narrow down to the only outputs it related to.

$$C_{k, i, j}^l = \sum_{c=0}^{inCHs-1} \sum_{n=-(N-1)/2}^{(N-1)/2} \sum_{m=-(M-1)/2}^{(M-1)/2} a^{l-1}(c, i+n, j+m) * K_k(c, n, m) + b_k$$

Substitute output formula in

where $0 \leq c < inputChannels$, k is k-th output channel index, $(N-1)/2 \leq i < I - (N-1)/2$, $(M-1)/2 \leq j < J - (M-1)/2$

$$= \sum_{k=0}^{outputChannels-1} \sum_{i=i'-\frac{N-1}{2}}^{i'+\frac{N-1}{2}} \sum_{j=j'-\frac{M-1}{2}}^{j'+\frac{M-1}{2}} \frac{\partial Loss}{\partial C^l(k, i, j)} * \frac{\partial \sum_{c=0}^{inCHs-1} \sum_{n=-(N-1)/2}^{(N-1)/2} \sum_{m=-(M-1)/2}^{(M-1)/2} a^{l-1}(c, i+n, j+m) * K_k(c, n, m) + b_k}{\partial a^{l-1}(cin, i', j')}$$

Only when

$cin=c, i'=i+n, j'=j+m,$

$cin=c, i-i'=-n, j-j'=-m,$

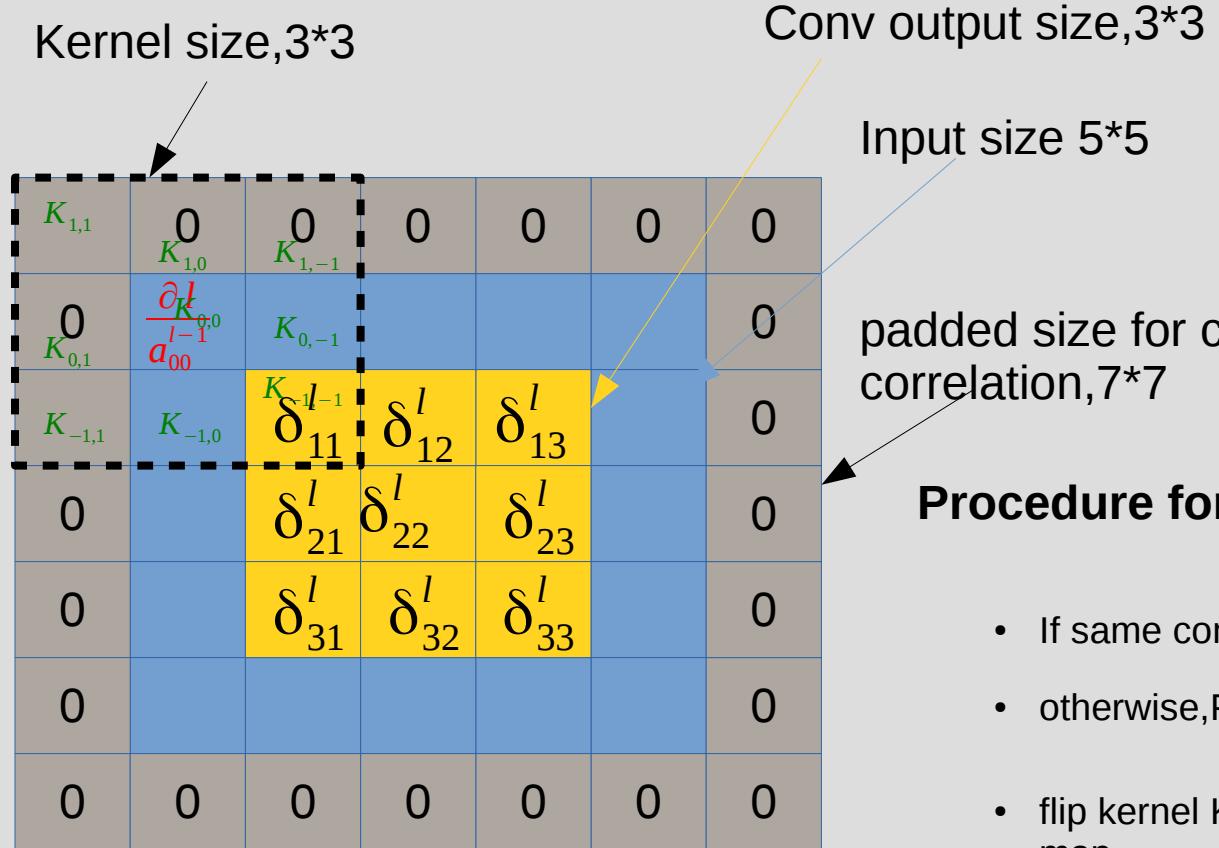
$$= \sum_{k=0}^{outputChannels-1} \sum_{i=i'-\frac{N-1}{2}}^{i'+\frac{N-1}{2}} \sum_{j=j'-\frac{M-1}{2}}^{j'+\frac{M-1}{2}} \delta^l(k, i, j) * K_k(cin, i-i', j-j')$$

$$= \sum_{k=0}^{outputChannels-1} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} \delta^l(k, i+n, j+m) * K_k(cin, -n, -m)$$

The two methods matched

$$\frac{\partial Loss}{\partial a^{l-1}(i', j')} = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} \delta^l(i'+n, j'+m) * K(-n, -m)$$

Geometric interpretation for $\frac{\partial Loss}{\partial a^{l-1}(cin, i', j')}$



- If same convolution, Padding 1 zero around sensitive map $\delta_{i,j}^l$
- otherwise, Padding 2 zeros around sensitive map $\delta_{i,j}^l$
- flip kernel $K(n,m)$ to $K(-n,-m)$, sliding along the padded map

$$\frac{\partial Loss}{\partial a^{l-1}(cin, i', j')} = \sum_{k=0}^{outputChannels-1} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} \delta^l(k, i+n, j+m) * K_k(cin, -n, -m)$$

Derivative w.r.t weights

$$\frac{\partial Loss}{\partial K^l(cin, n, m)}$$

$$C_k^l(i, j) = \sum_{c=0}^{inCHs-1} \sum_{n'=-\frac{(N-1)}{2}}^{\frac{(N-1)}{2}} \sum_{m'=-\frac{(M-1)}{2}}^{\frac{(M-1)}{2}} a^{l-1}(c, i+n', j+m') * K_k(c, n', m') + b_k$$

where $0 \leq c < inputChannels$, k is k -th output channel index, $\frac{(N-1)}{2} \leq i < I - \frac{(N-1)}{2}$, $\frac{(M-1)}{2} \leq j < J - \frac{(M-1)}{2}$

$$\frac{\partial Loss}{\partial K_k^l(cin, n, m)} = \sum_{i=\frac{N-1}{2}}^{I-1-\frac{N-1}{2}} \sum_{j=\frac{M-1}{2}}^{J-1-\frac{M-1}{2}} \frac{\partial Loss}{\partial C_k^l(i, j)} * \frac{\partial C_k^l(i, j)}{\partial K_k^l(cin, n, m)} \quad cin=c, n=n', m=m',$$

$$= \sum_{i=\frac{N-1}{2}}^{I-1-\frac{N-1}{2}} \sum_{j=\frac{M-1}{2}}^{J-1-\frac{M-1}{2}} \delta^l(cin, i, j) * a^{l-1}(cin, i+n, j+m)$$

Correlation operation

$$\begin{aligned} -\frac{N-1}{2} \leq n \leq \frac{N-1}{2} \quad -\frac{M-1}{2} \leq m \leq \frac{M-1}{2} \end{aligned}$$

Geometric interpretation for

$$\frac{\partial Loss}{\partial K^l(cin, n, m)}$$

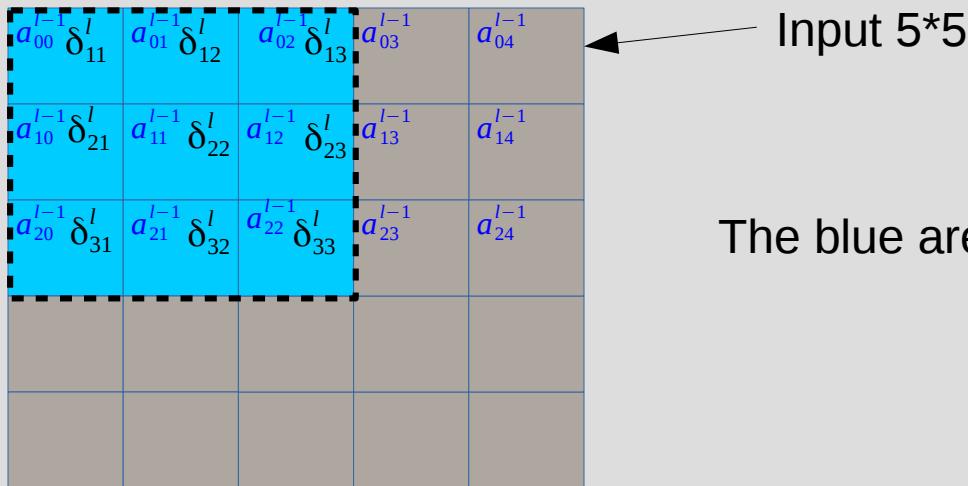
$$\frac{\partial Loss}{\partial K_k^l(cin, n', m')} \quad \text{let } n + \frac{N-1}{2} = n' \quad m + \frac{M-1}{2} = m'$$

$$= \sum_{\substack{i=\frac{N-1}{2} \\ 0 \leq n' \leq N-1}}^{I-1-\frac{N-1}{2}} \sum_{\substack{j=\frac{M-1}{2} \\ 0 \leq m' \leq M-1}}^{J-1-\frac{M-1}{2}} \delta^l(cin, i, j) * a^{l-1}(cin, i+n' - \frac{N-1}{2}, j+m' - \frac{M-1}{2})$$

Geometric interpretation for

$$\frac{\partial Loss}{\partial K^l(cin, n, m)}$$

$$\frac{\partial Loss}{\partial K_k^l(cin, n', m')} = \sum_{\substack{i=\frac{N-1}{2} \\ 0 \leq n' \leq N-1}}^{I-1-\frac{N-1}{2}} \sum_{\substack{j=\frac{M-1}{2} \\ 0 \leq m' \leq m-1}}^{J-1-\frac{M-1}{2}} \delta^l(cin, i, j) * a^{l-1}(cin, i+n' - \frac{N-1}{2}, j+m' - \frac{M-1}{2})$$



The blue area is the pixels that $K(0,0)$ can touch

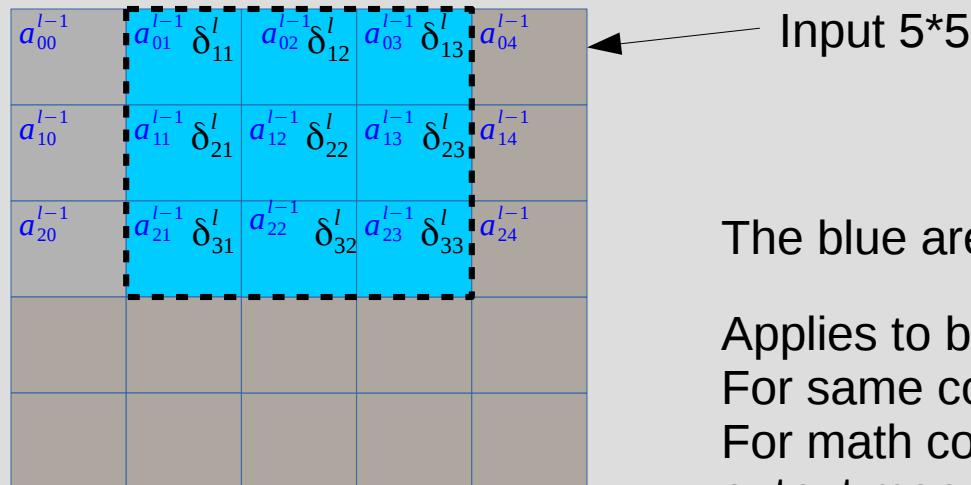
Case $n=0, m=0$

$$\begin{aligned} \frac{\partial Loss}{\partial K^l(0,0)} &= \delta^l(1,1)*a^{l-1}(0,0) + \delta^l(1,2)*a^{l-1}(0,1) + \delta^l(1,3)*a^{l-1}(0,2) \\ &\quad \dots \\ &\quad \dots + \delta^l(3,3)*a^{l-1}(2,2) \end{aligned}$$

Geometric interpretation for

$$\frac{\partial Loss}{\partial K^l(cin, n, m)}$$

$$\frac{\partial Loss}{\partial K_k^l(cin, n', m')} = \sum_{\substack{i=\frac{N-1}{2} \\ 0 \leq n' \leq N-1}}^{I-1-\frac{N-1}{2}} \sum_{\substack{j=\frac{M-1}{2} \\ 0 \leq m' \leq M-1}}^{J-1-\frac{M-1}{2}} \delta^l(cin, i, j) * a^{l-1}(cin, i+n' - \frac{N-1}{2}, j+m' - \frac{M-1}{2})$$



The blue area is the pixels that $K(0,1)$ can touch

Applies to both same convolution and reduced convolution
 For same convolution, we need to pad 1 zeros around input map
 For math convenience purpose we always pad virtual zeros around output map, indexing from $[(N-1)/2, (M-1)/2]$ to $[I-1-(N-1)/2, J-1-(M-1)/2]$, which is the actual output region

Case $n=0, m=1$

$$\frac{\partial Loss}{\partial K^l(0,1)} = \delta^l(1,1)*a^{l-1}(0,1) + \delta^l(1,2)*a^{l-1}(0,2) + \delta^l(1,3)*a^{l-1}(0,3) \\ \dots \\ \dots + \delta^l(3,3)*a^{l-1}(2,3)$$

Coming next: Batch Normalization

韭菜岭
君拥韭菜岭，
长发王八卷。
定眼吹大泡，
天上建人间。
一波抓一波，
只需一坨屎。



谁的财富

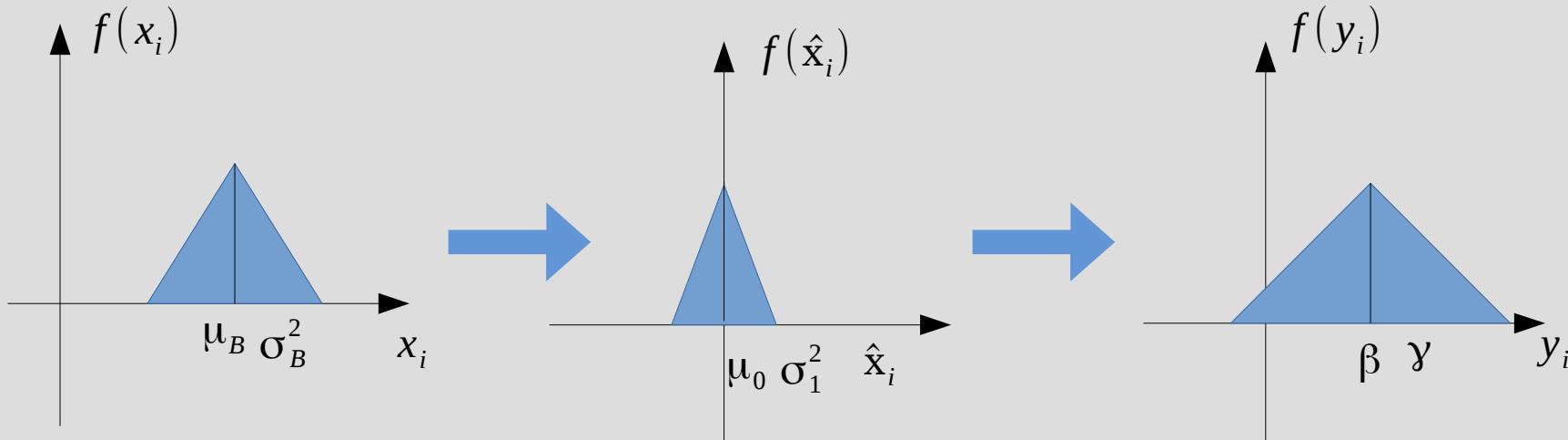
能自由带出去的
是真的属于你的财富
剩下的
也许可能大概
是真的纸币
却大多用来填了房坑
天马行空的建筑
就像是一座座昂贵的坟墓
埋葬了我们的青春、梦想、甚至灵魂
一家几代人在里面
骄傲、苟且地活着

Batch normalization Intro.

$$\mu_B \leftarrow \frac{1}{m} \sum_{i=0}^{M-1} x_i \quad \sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=0}^{M-1} (x_i - \mu_B)^2$$

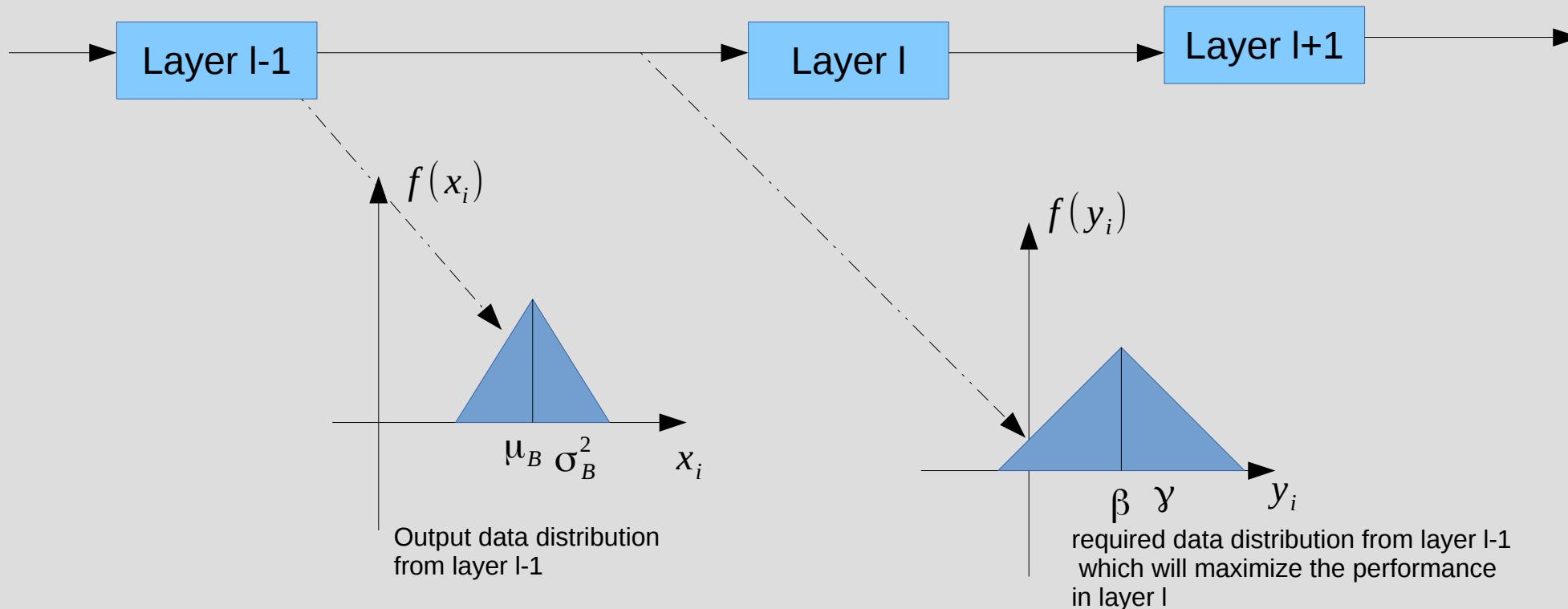
$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

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



Distribution transformation

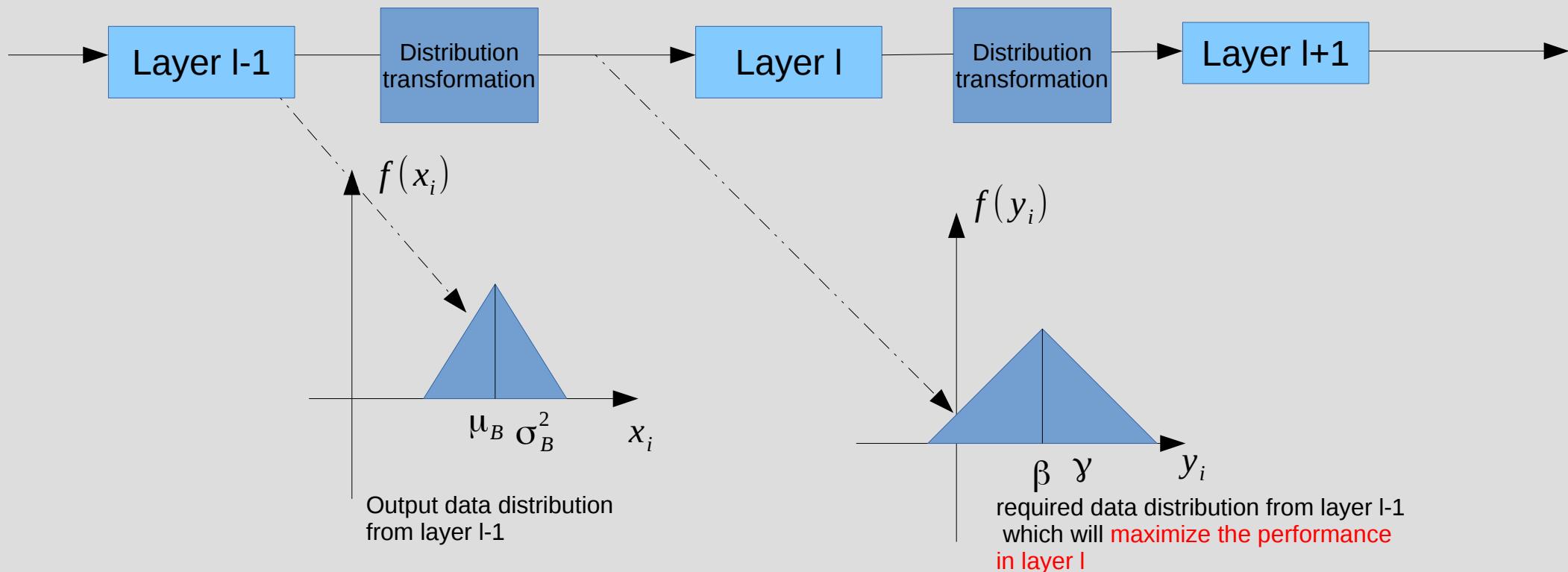
Why we need Batch normalization ?



There is a mismatch between what is given and what is required.

To solve the mismatch and ease the training pain,better to add two specific tunable parameter to do the distribution transformation

Why we need Batch normalization ? Cont.



To solve the mismatch and ease the training pain,better to add two specific tunable parameter to do the distribution transformation

Why BN makes training faster ?

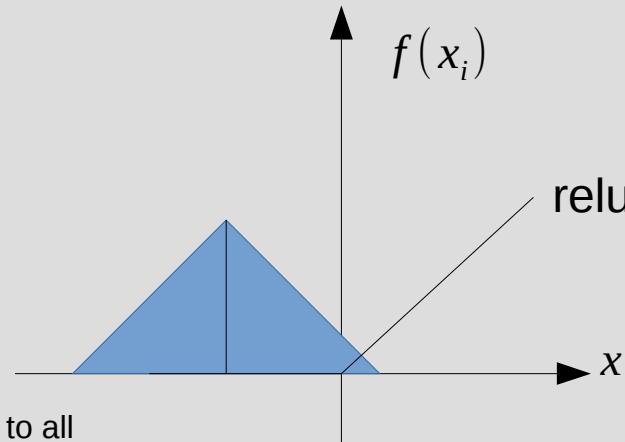
When training DNN with relu activation, if the initial learning rate is set to be too big it will probably cause data distribution in some layer end up like that shown in right picture.

Too many zeros in the neuron

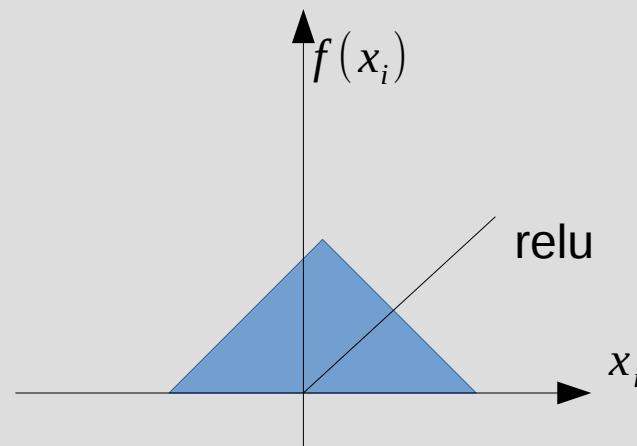
Not enough information for the following layer to extract



Tend to lead to all zero outcome, training dead !



When BN added, the mean of data distribution tend to be around zero, which could prevent training die from all zero outcomes



Would it make training even faster ?

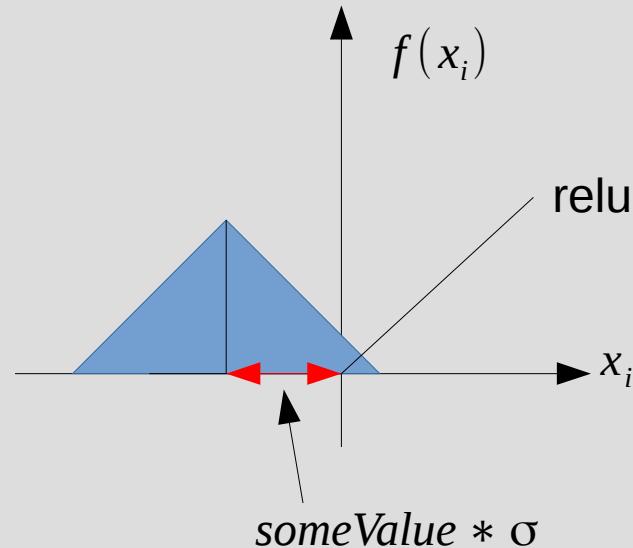
If we add a constrain to β update during training,

$$\beta = \beta + d\beta$$

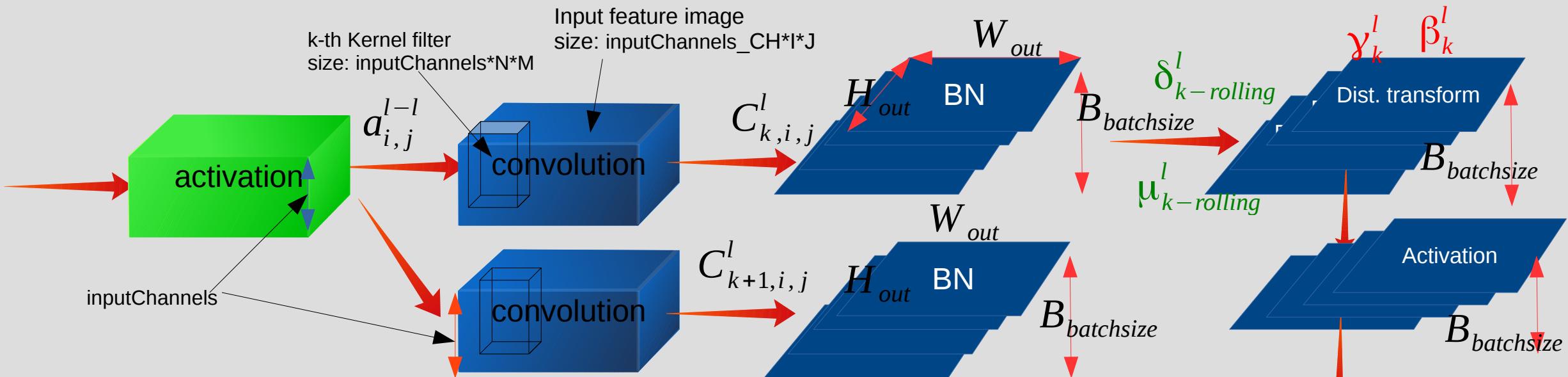
If $\beta < -someValue * \sigma$

$$\beta = -someValue * \sigma$$

This could avoid data being shifted too negative. **With this we can set the initial learning rate even bigger ???**



Batch normalization following conv.



$$\mu_k = \frac{1}{B * H * W} \sum_{b=0}^{B-1} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} C_k^l(b, i, j) \quad K\text{-th Channel mean}$$

$$\sigma_k^2 = \frac{1}{B * H * W} \sum_{b=0}^{B-1} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (C_k^l(b, i, j) - \mu_k)^2 \quad K\text{-th Channel variance}$$

$$\hat{C}_{k,b,i,j}^l = \frac{C_{k,b,i,j}^l - \mu_k}{\sqrt{\sigma_k^2 + \epsilon}}$$

$$y_{k,b,i,j} = \gamma_k * \hat{C}_{k,b,i,j}^l + \beta_k$$

Back-pro at BN

$$\mu_k = \frac{1}{B*H*W} \sum_{b=0}^{B-1} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} C_k^l(b, i, j)$$

$$\sigma_k^2 = \frac{1}{B*H*W} \sum_{b=0}^{B-1} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (C_k^l(b, i, j) - \mu_k)^2$$

$$\hat{C}_{k,b,i,j}^l = \frac{C_{k,b,i,j}^l - \mu_k}{\sqrt{\sigma^2 + e}}$$

$$y_{k,b,i,j} = \gamma_k * \hat{C}_{k,b,i,j}^l + \beta_k$$

Given: $\delta_{k,b,i,j}^l = \frac{\partial Loss}{\partial y_{k,b,i,j}^l}$

Objective: $\frac{\partial Loss}{\partial \gamma_k^l}$  Easy, skip this part

$$\gamma_k^l = \gamma_k^l + d\gamma_k^l$$

$$\frac{\partial Loss}{\partial \beta_k^l}$$
  Easy, skip this part

$$\beta_k^l = \beta_k^l + d\beta_k^l$$

$$\frac{\partial Loss}{\partial C_{k,b,i,j}^l}$$
  Only do this part here

For back-pro in conv layer

The subscript k could be ignored because channels don't cross talk with each other

Back-pro at BN

$$\frac{\partial Loss}{\partial C_{b,i,j}^l}$$

$$\mu_k = \frac{1}{B*H*W} \sum_{b=0}^{B-1} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} C_k^l(b, i, j)$$

$$\sigma_k^2 = \frac{1}{B*H*W} \sum_{b=0}^{B-1} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (C_k^l(b, i, j) - \mu_k)^2$$

$$\frac{dl}{dC_{b,i,j}^l} = \sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \frac{d}{d\hat{C}_{b',i',j'}^l} * \frac{d\frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma_k^2 + e}}}{dC_{b,i,j}^l}$$

$$= \sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \frac{dl}{dy_{b',i',j'}^l} * \frac{dy_{b',i',j'}^l}{d\hat{C}_{b',i',j'}^l} * \frac{d\frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma_k^2 + e}}}{dC_{b,i,j}^l}$$

$$= \sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \delta^l(b', i', j') * \gamma_k * \frac{d\frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma_k^2 + e}}}{dC_{b,i,j}^l}$$

$$\hat{C}_{k,b,i,j}^l = \frac{C_{k,b,i,j}^l - \mu_k}{\sqrt{\sigma_k^2 + e}}$$

$$y_{k,b,i,j} = \gamma_k * \hat{C}_{k,b,i,j}^l + \beta_k$$

Substitute y in and expand a little bit

Using chain rule we can expand the formula even further, see next slide

Back-pro at BN

$$\frac{\partial Loss}{\partial C_{b,i,j}^l}$$

$$\mu_k = \frac{1}{B*H*W} \sum_{b=0}^{B-1} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} C_k^l(b, i, j)$$

$$\sigma_k^2 = \frac{1}{B*H*W} \sum_{b=0}^{B-1} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (C_k^l(b, i, j) - \mu_k)^2$$

$$\hat{C}_{k,b,i,j}^l = \frac{C_{k,b,i,j}^l - \mu_k}{\sqrt{\sigma^2 + e}}$$

$$y_{k,b,i,j} = \gamma_k * \hat{C}_{k,b,i,j}^l + \beta_k$$

$$\frac{\partial Loss}{\partial C_{b,i,j}^l} = \sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \delta^l(b', i', j') * \gamma_k *$$

$$\boxed{\frac{d \frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma^2 + e}}}{d C_{b',i',j'}^l} * \frac{d C_{b',i',j'}^l}{d C_{b,i,j}^l} + \frac{d \frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma^2 + e}}}{d \mu_k} * \frac{d \mu_k}{d C_{b,i,j}^l} + \frac{d \frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma^2 + e}}}{d \sigma^2} * \frac{d \sigma^2}{d C_{b,i,j}^l}}$$

$$\sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \delta^l(b', i', j') * \gamma * \frac{d \frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma^2 + e}}}{d C_{b',i',j'}^l} * \frac{d C_{b',i',j'}^l}{d C_{b,i,j}^l}$$

$$= \delta^l(b, i, j) * \gamma * \frac{1}{\sqrt{\sigma^2 + e}} \quad \text{only when } b=b', i=i', j=j'$$

Solve these 3 components one by one

Back-pro at BN

$$\frac{\partial Loss}{\partial C_{b,i,j}^l}$$

$$= \sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \delta^l(b', i', j') * \gamma_k * \left(\frac{d \frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma^2 + e}}}{d C_{b',i',j'}^l} * \frac{d C_{b',i',j'}^l}{d C_{b,i,j}^l} + \boxed{\frac{d \frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma^2 + e}}}{d \mu_k} * \frac{d \mu_k}{d C_{b,i,j}^l} + \frac{d \frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma^2 + e}}}{d \sigma^2} * \frac{d \sigma^2}{d C_{b,i,j}^l}} \right)$$

$$\sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \delta^l(b', i', j') * \gamma * \frac{d \frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma^2 + e}}}{d \mu_k} * \frac{d \mu_k}{d C_{b,i,j}^l}$$

$$= \gamma * \frac{-1}{\sqrt{\sigma^2 + e}} * \frac{1}{B * N * M} * \sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \delta^l(b', i', j')$$

$$\boxed{\frac{d \mu_k}{d C_{b,i,j}^l} = d \frac{\frac{1}{B * N * M} \sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} C_{b',i',j'}^l}{d C_{b,i,j}^l} = \frac{1}{B * N * M} \delta_{b',b} \delta_{i',i} \delta_{j',j} = \frac{1}{B * N * M} \quad \delta_{b',b} = 1 \text{ if } b' = b \text{ else } \delta_{b',b} = 0}$$

Back-pro at BN

$$\frac{\partial Loss}{\partial C_{b,i,j}^l}$$

$$= \sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \delta^l(b', i', j') * \gamma_k * \left(\frac{d \frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma^2 + e}}}{d C_{b',i',j'}^l} * \frac{d C_{b',i',j'}^l}{d C_{b,i,j}^l} + \frac{d \frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma^2 + e}}}{d \mu_k} * \frac{d \mu_k}{d C_{b,i,j}^l} + \boxed{\frac{d \frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma^2 + e}}}{d \sigma^2} * \frac{d \sigma^2}{d C_{b,i,j}^l}} \right)$$

$$\sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \delta^l(b', i', j') * \gamma * \frac{d \frac{C_{b',i',j'}^l - \mu_k}{\sqrt{\sigma^2 + e}}}{d \sigma^2} * \frac{d \sigma^2}{d C_{b,i,j}^l}$$

$$= \sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \frac{\delta^l(b', i', j') * \gamma_k * \frac{1}{2} (C_{b',i',j'}^l - \mu_k)}{(\sigma^2 + e)^{\frac{-3}{2}}} * \frac{d \sigma^2}{d C_{b,i,j}^l}$$

$$= \frac{1}{B * N * M} * 2 * (C_{b,i,j}^l - \mu_{ch}) * \gamma * \frac{0.5}{(\sigma^2 + e)^{\frac{-3}{2}}} * \sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \delta^l(b', i', j') * (C_{b',i',j'}^l - \mu_k)$$

$$\begin{aligned} \frac{d \sigma^2}{d C_{b,i,j}^l} &= \frac{1}{B * N * M} * d \frac{\sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} (C_{b',i',j'}^l - \mu_{ch})^2}{d C_{b,i,j}^l} \\ &= \frac{1}{B * N * M} * 2 * (C_{b,n,m}^l - \mu_k) \end{aligned}$$

Back-pro at BN

$$\frac{\partial Loss}{\partial C_{b,i,j}^l}$$

$$\frac{\partial Loss}{\partial C_{b,i,j}^l} = \delta^l(b,i,j) * \gamma * \frac{1}{\sqrt{\sigma^2 + e}}$$

The final formula for derivative w.r.t input

$$+ \frac{1}{B*N*M} * \sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \gamma * \delta^l(b',i',j') * \frac{-1}{\sqrt{\sigma^2 + e}}$$

$$+ \frac{1}{B*N*M} * 2 * (C_{b,i,j}^l - \mu_{ch}) * \frac{0.5}{(\sigma^2 + e)^{\frac{-3}{2}}} * \sum_{b'=0}^{B-1} \sum_{i'=0}^{H-1} \sum_{j'=0}^{W-1} \gamma * \delta^l(b',i',j') * (C_{b',i',j'}^l - \mu_k)$$

Thank you

Discussion

夏监人

孔雀东南归，
红纸漫天飞。
千人争骨头，
赛过一群猴。
刘邦赐斗粮，
犬儒辱庙堂。
勤学数十载，
终成夏监人。

Email: brianwchh@gmail.com

Linkedin : <https://www.linkedin.com/in/brianwchh>