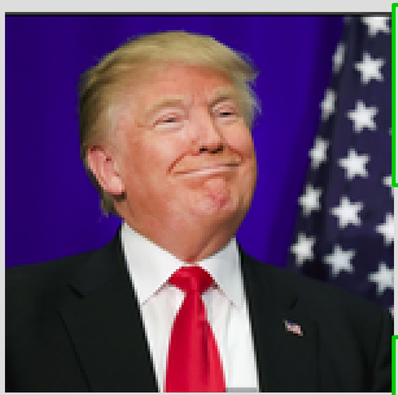
Welcome

Brian Wu



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

No problem

Hey!,We are 草泥马

Wake up!!!

The motivation

- Help people having the same trouble and save their time
- It is necessary when you want to add layers to existing models

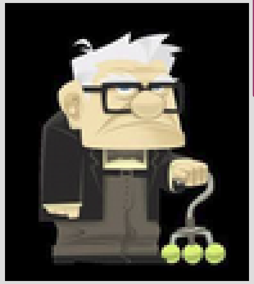
Give to Caesar what is Caesar’s,to God what is God’s.

Agenda

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

Dad,I lost my job

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

Cross-correlation vs Convolution

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

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

- Convolution : filtering operation

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

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

I(0,0)	I(0,1)	I(0,2)	I(0,3)
			I(3,3)

img(4,4)

K(-1,-1)	K(-1,0)	K(-1,1)
K(0,-1)	K(0,0)	K(0,1)
K(1,-1)	K(1,0)	K(1,1)

K(3,3)

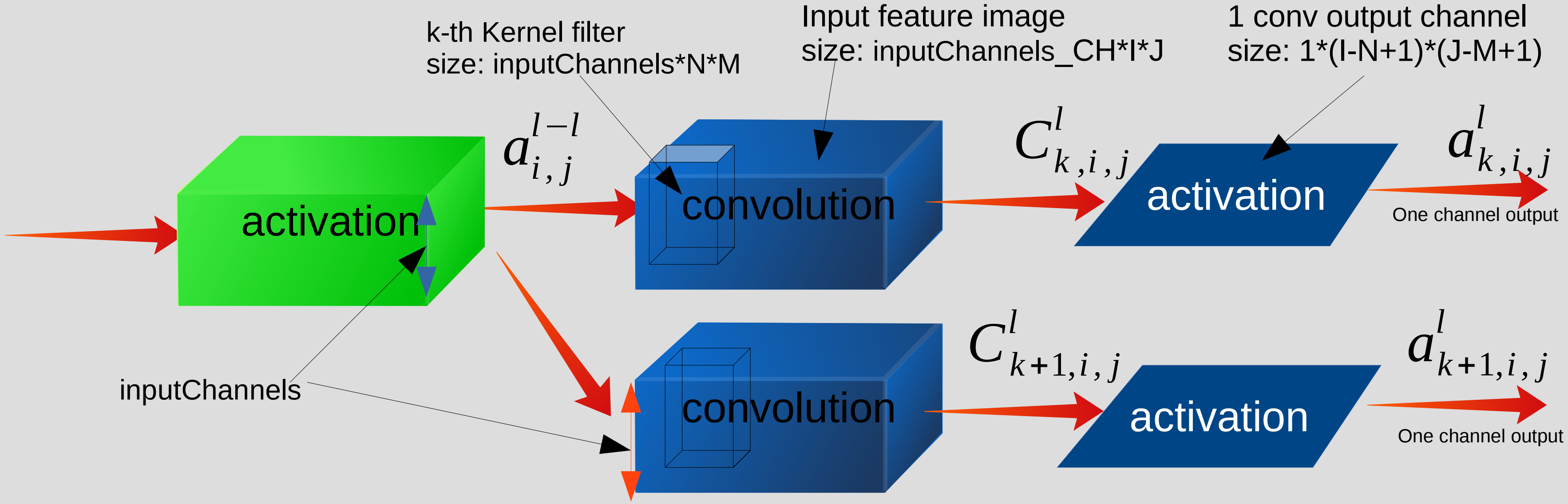
G(1,1)	
	G(2,2)

G(2,2)

K(1,1)	K(1,0)	K(1,-1)
K(0,1)	K(0,0)	K(0,-1)
K(-1,1)	K(-1,0)	K(-1,-1)

K(-n,-m) = rotate K(n,m) 180 degree around K(0,0)

Convolution layer & formula

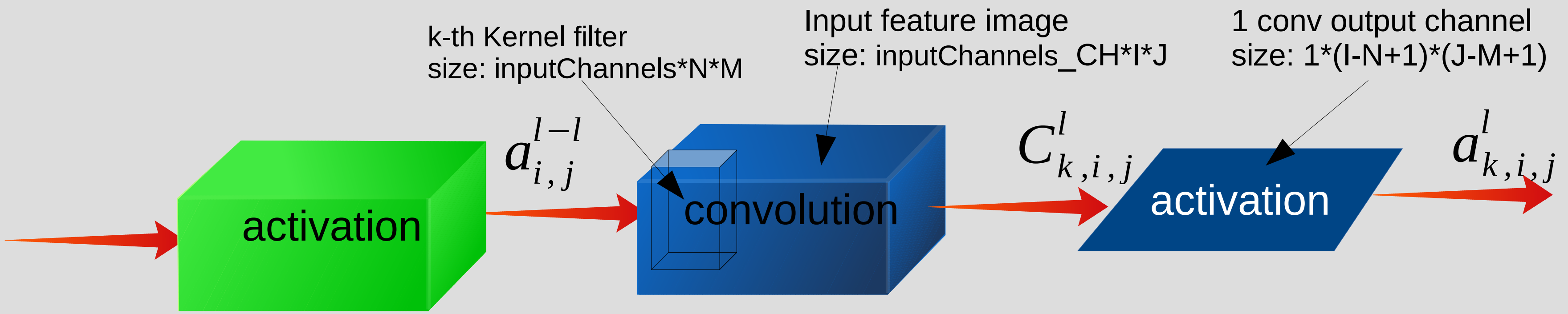


$$C^{l}_{k,i,j} = \sum_{c=0}^{inCHs-1} \sum_{n=-(N-1)/2}^{(N-1)/2} \sum_{m=-(M-1)/2}^{(M-1)/2} a^{l-1}(c, i+n, j+m) * K_k(c, -n, -m) + b_k$$

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

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

Given:

- Derivative w.r.t conv outputs, the output sensitive map : $\delta^{l}_{k,i,j} = \dfrac{\partial Loss}{\partial C^{l}(k,i,j)}$ $k\ \ is\ \ \text{k-th}\ \ output\ \ channel\ \ index.\ \ 0 \le i < I-1-N+1,\ 0 \le j < J-1-M+1$

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

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

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

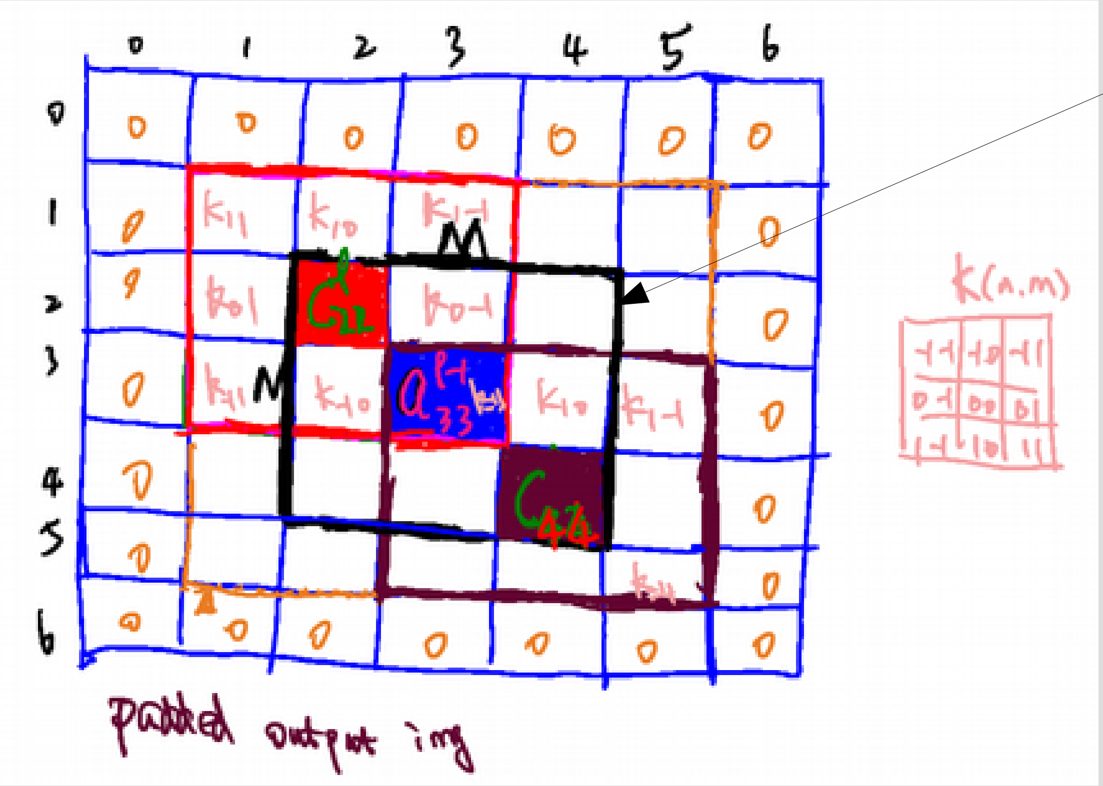
By chain rule: *partial Loss/ partial_all_output * partial_all_output / partial_input*

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

Note that: (N-1)/2<=i<I-1-(N-1)/2, (M-1)/2<=j<J-1-(M-1)/2,is the output index,0<=i’<I, 0<=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*N filter bounding box are, as illustrated below:

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



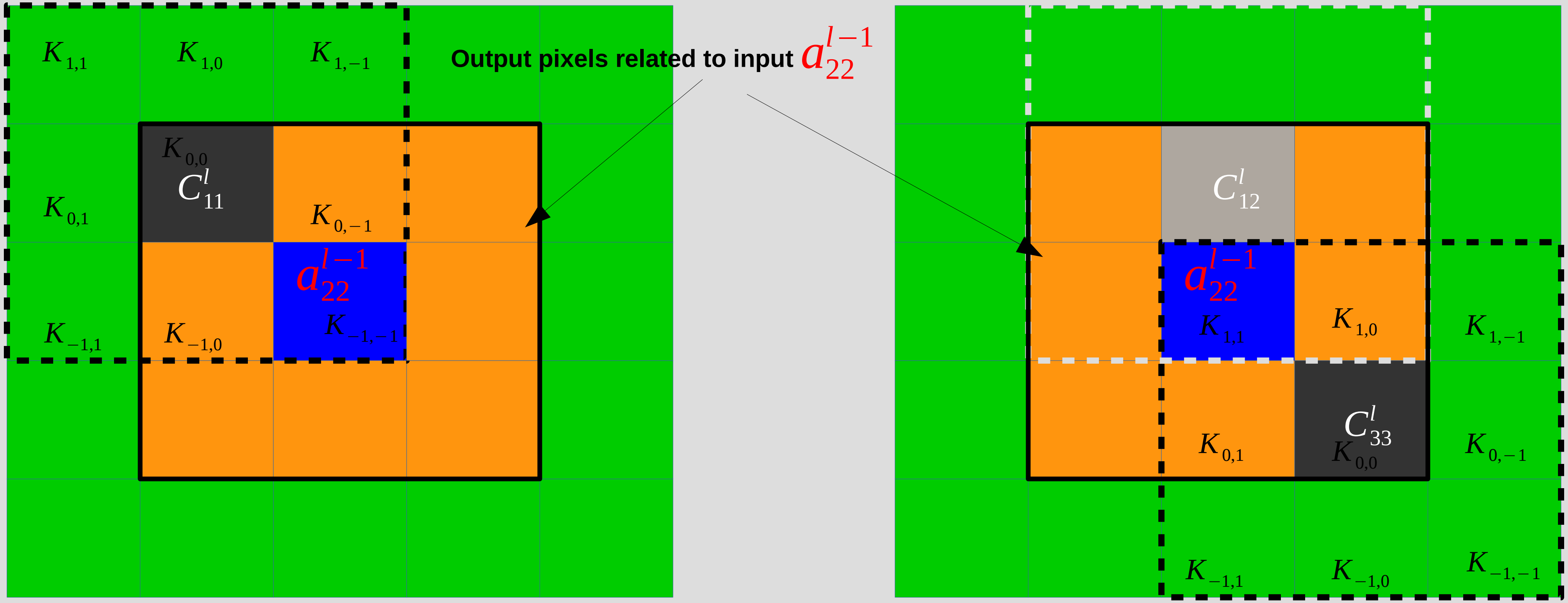
Only convolution output $C^{l}{}_{(k,i,j)}$ inside the N*M box are related to input $a^{l-1}{}_{(cin,3,3)}$

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

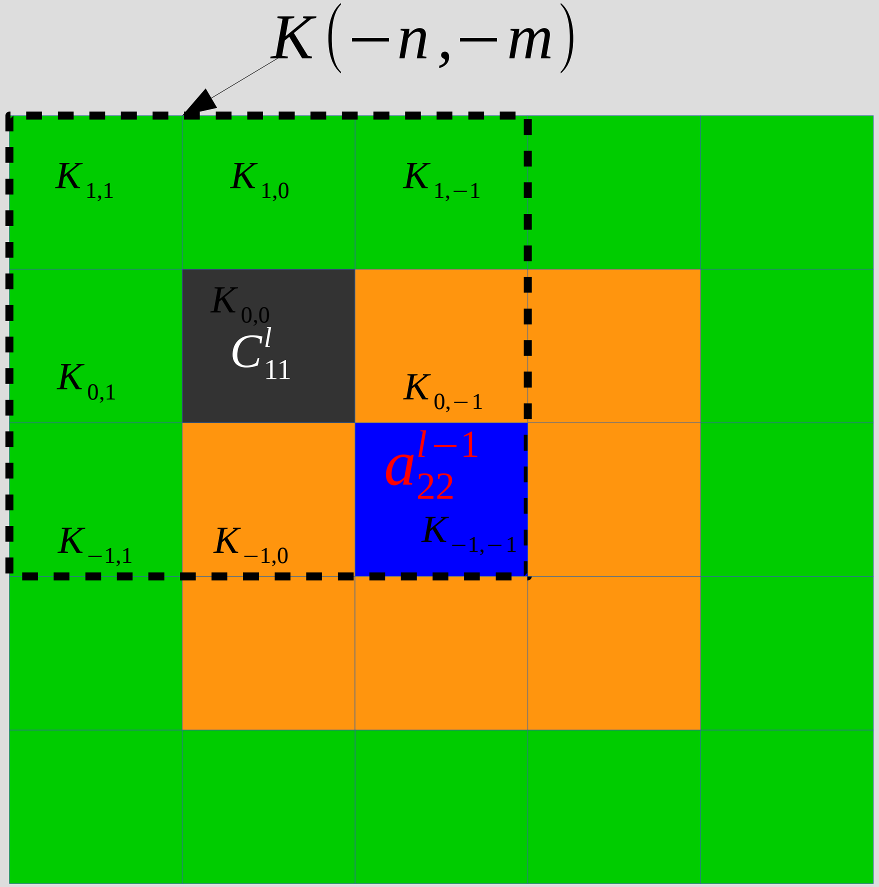
Derivative w.r.t conv input $\dfrac{\partial\, Loss}{\partial\, a^{l-1}(cin, i', j')}$ cont.

Let's start with simple case, one channel in,one channel out. Input 5*5, kernel 3*3, output 5*5 padded with 0

$$\frac{\partial\, Loss}{\partial\, a^{l-1}(2,2)} = \frac{\partial\, Loss}{\partial\, C^l_{11}} * \frac{\partial\, C^l_{11}}{\partial\, a^{l-1}_{22}} + \frac{\partial\, Loss}{\partial\, C^l_{12}} * \frac{\partial\, C^l_{12}}{\partial\, a^{l-1}_{22}} + \ldots + \frac{\partial\, Loss}{\partial\, C^l_{33}} * \frac{\partial\, C^l_{33}}{\partial\, a^{l-1}_{22}}$$



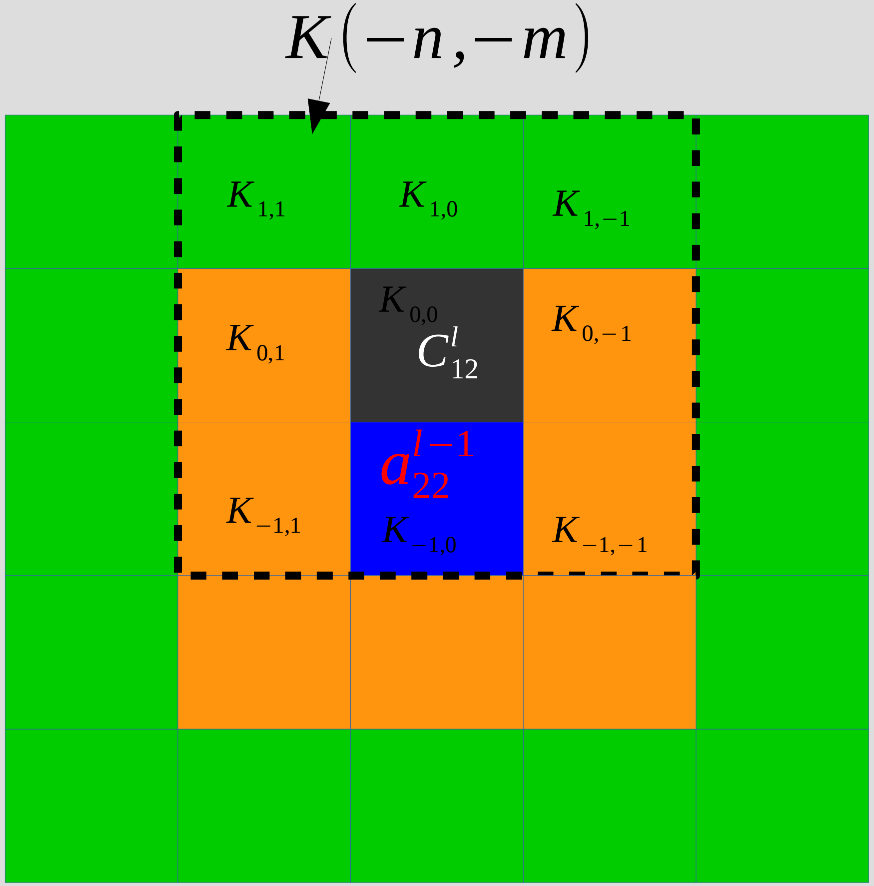
Derivative w.r.t conv input $\dfrac{\partial Loss}{\partial a^{l-1}(cin, i', j')}$ cont.



$$\frac{\partial Loss}{\partial a^{l-1}(2{,}2)} = \boxed{\frac{\partial Loss}{\partial C^l_{11}} * \frac{\partial C^l_{11}}{\partial a^{l-1}_{22}}} + \frac{\partial Loss}{\partial C^l_{12}} * \frac{\partial C^l_{12}}{\partial a^{l-1}_{22}} + \ldots . + \frac{\partial Loss}{\partial C^l_{33}} * \frac{\partial C^l_{33}}{\partial a^{l-1}_{22}}$$

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

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



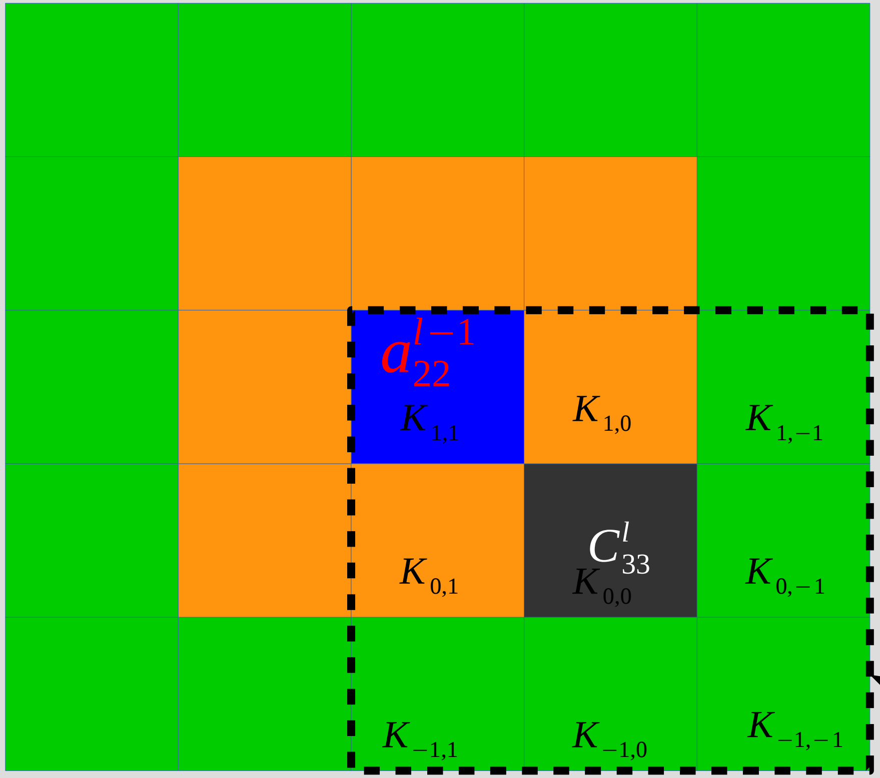
$$\frac{\partial Loss}{\partial a^{l-1}(2,2)} = \frac{\partial Loss}{\partial C^l_{11}} * \frac{\partial C^l_{11}}{\partial a^{l-1}_{22}} + \boxed{\frac{\partial Loss}{\partial C^l_{12}} * \frac{\partial C^l_{12}}{\partial a^{l-1}_{22}}} + \dots + \frac{\partial Loss}{\partial C^l_{33}} * \frac{\partial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

$$\frac{\partial Loss}{\partial C^l_{12}} * \frac{\partial C^l_{12}}{\partial a^{l-1}_{22}} = \delta^l(1,2) * K(-1,0)$$

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

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

$$\frac{\partial Loss}{\partial C^l_{33}} * \frac{\partial C^l_{33}}{\partial a^{l-1}_{22}} = \delta^l(3,3) * K(1,1)$$



$K(-n, -m)$

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

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

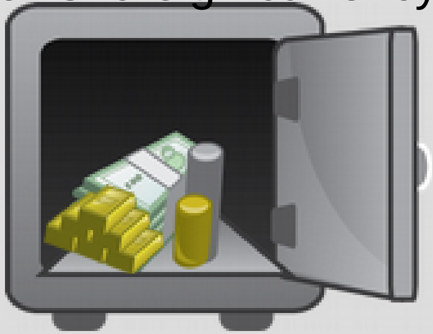
$$= \delta^{l}(1,1) * K(-1,-1) + \delta^{l}(1,2) * K(-1,0) + \dots + \delta^{l}(3,3) * K(1,1)$$

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

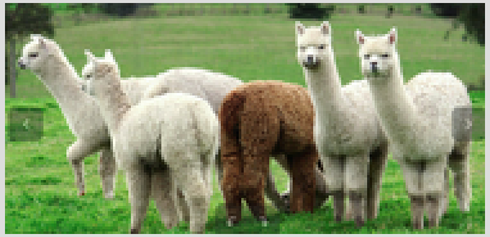
Next step : derivation from math, see if they agree with each other

Heavy math coming next

C nation’s foreign currency pool



we’ve heard you do good foreign business !



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

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

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

Narrow down to the only outputs it related to.

Substitute output formula in

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

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

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

Only when

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

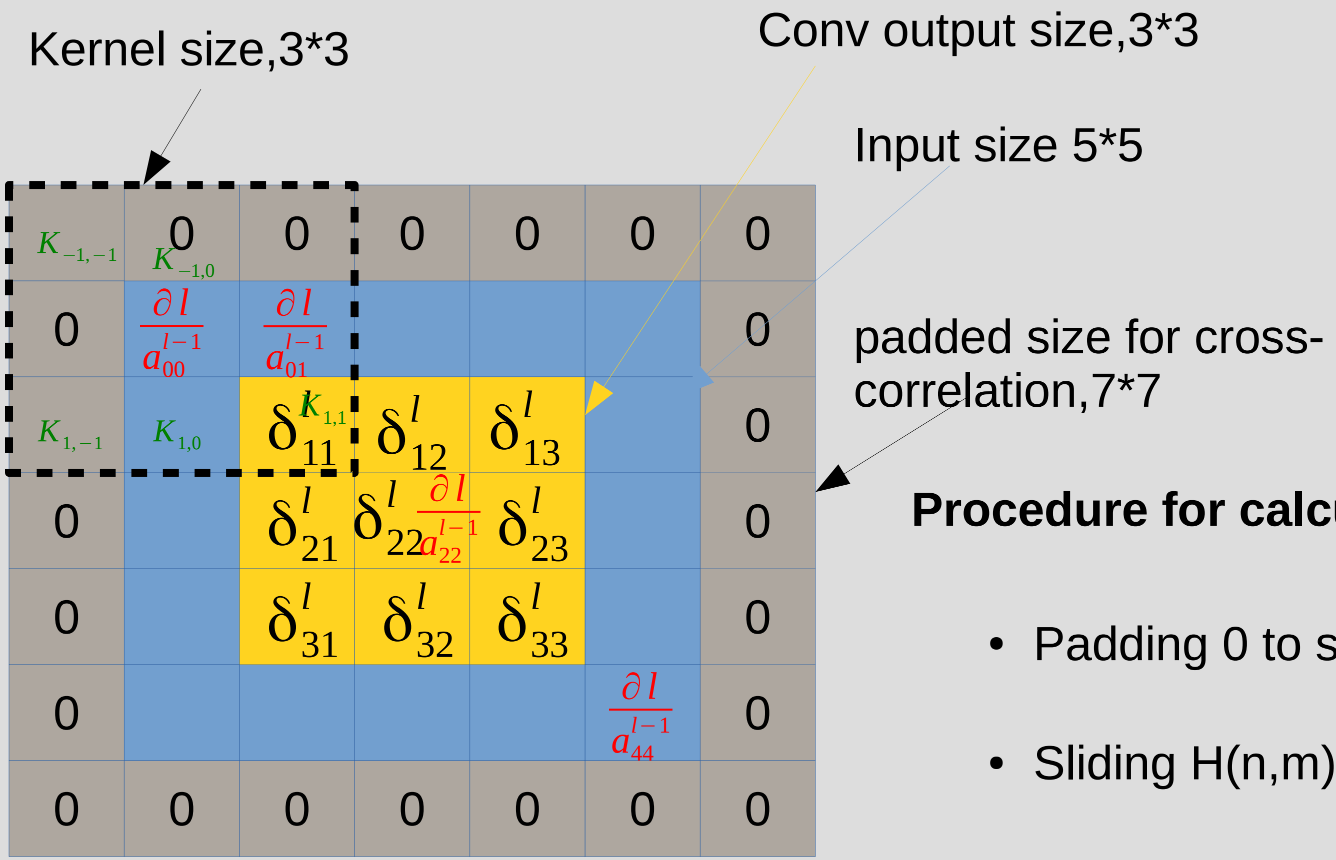
$cin=c,i-i'=-n,\ j-j'=-m,$

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

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



$$\frac{\partial Loss}{\partial a_{00}^{l-1}} = \delta^{l}(1,1) * K(1,1)$$

Procedure for calculating $\dfrac{\partial Loss}{\partial a^{l-1}(cin, i', j')}$

- Padding 0 to sensitive map $\delta_{i,j}^{l}$
- Sliding H(n,m) along the padded map

Derivative w.r.t weights $\dfrac{\partial Loss}{\partial K^l(cin,n,m)}$

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

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

$$\frac{\partial Loss}{\partial K^l_k(cin,n,m)} = \sum_{i=\frac{N-1}{2}}^{I-1-\frac{N-1}{2}} \sum_{j=\frac{M-1}{2}}^{J-1-\frac{M-1}{2}} \frac{\partial Loss}{\partial C^l_k(cin,i,j)} * \frac{\partial C^l_k(cin,i,j)}{\partial K^l_k(cin,n,m)}$$

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

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

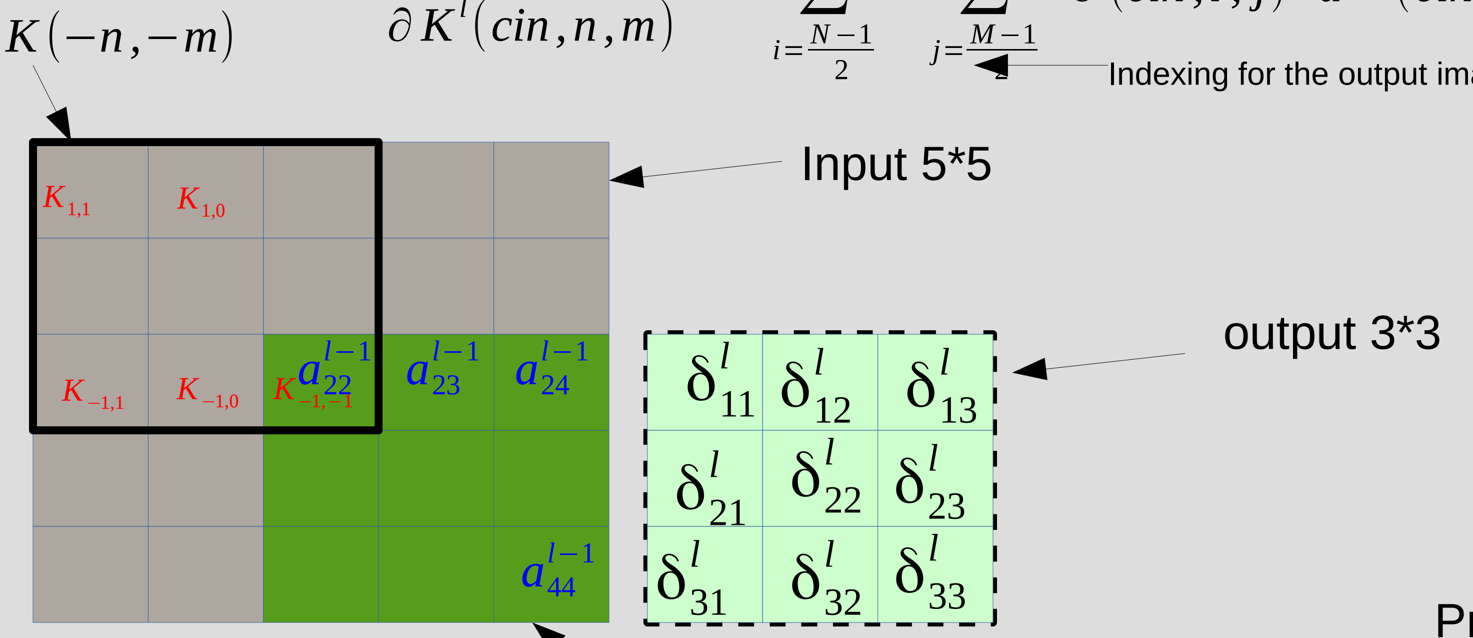
$$G(i,j) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} img(i-n, j-m) * K(n,m)$$

Looks like convolution but index is not the same

Geometric interpretation for $\dfrac{\partial Loss}{\partial K^l(cin,n,m)}$

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

Indexing for the output image



Case n=-1,m=-1

This is the input elements K(-1,-1) has multiplied to generate the output

Procedure: sliding from bottom right to top left and do cross-correlation

$$\frac{\partial Loss}{\partial K^l(-1,-1)} = \delta^l(1,1)*a^{l-1}(2,2) + \delta^l(1,2)*a^{l-1}(2,3) + \delta^l(1,3)*a^{l-1}(2,4) \ \dots\ + \delta^l(3,3)*a^{l-1}(4,4)$$

Coming next: Batch Normalization

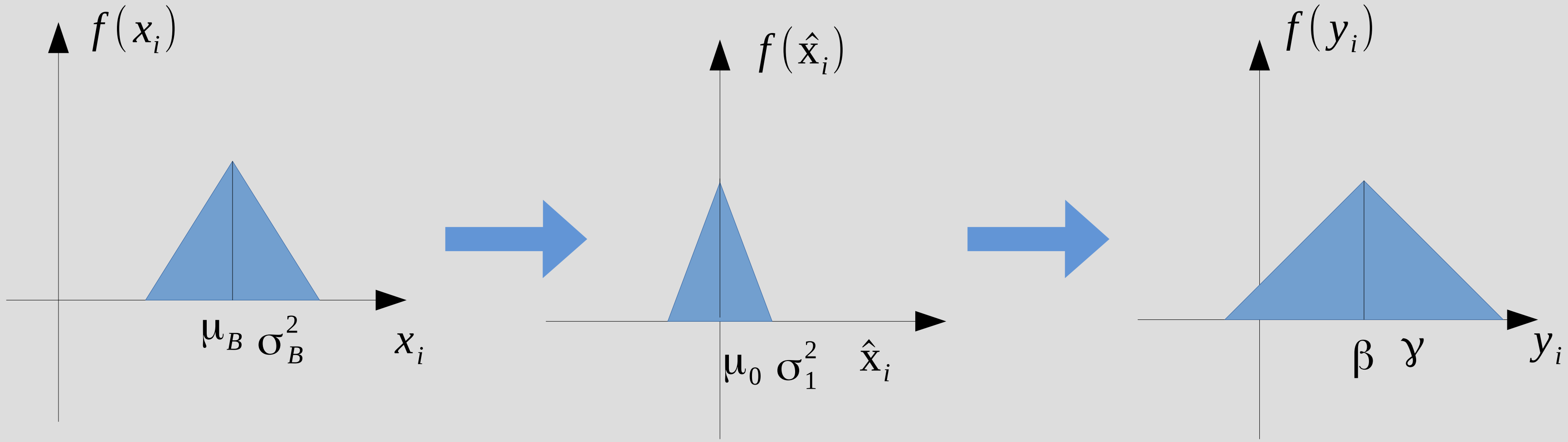
韭菜岭
君拥韭菜岭，
长发王八卷。
定眼吹大泡，
天上建人间。
一波抓一波，
只需一坨屎。

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

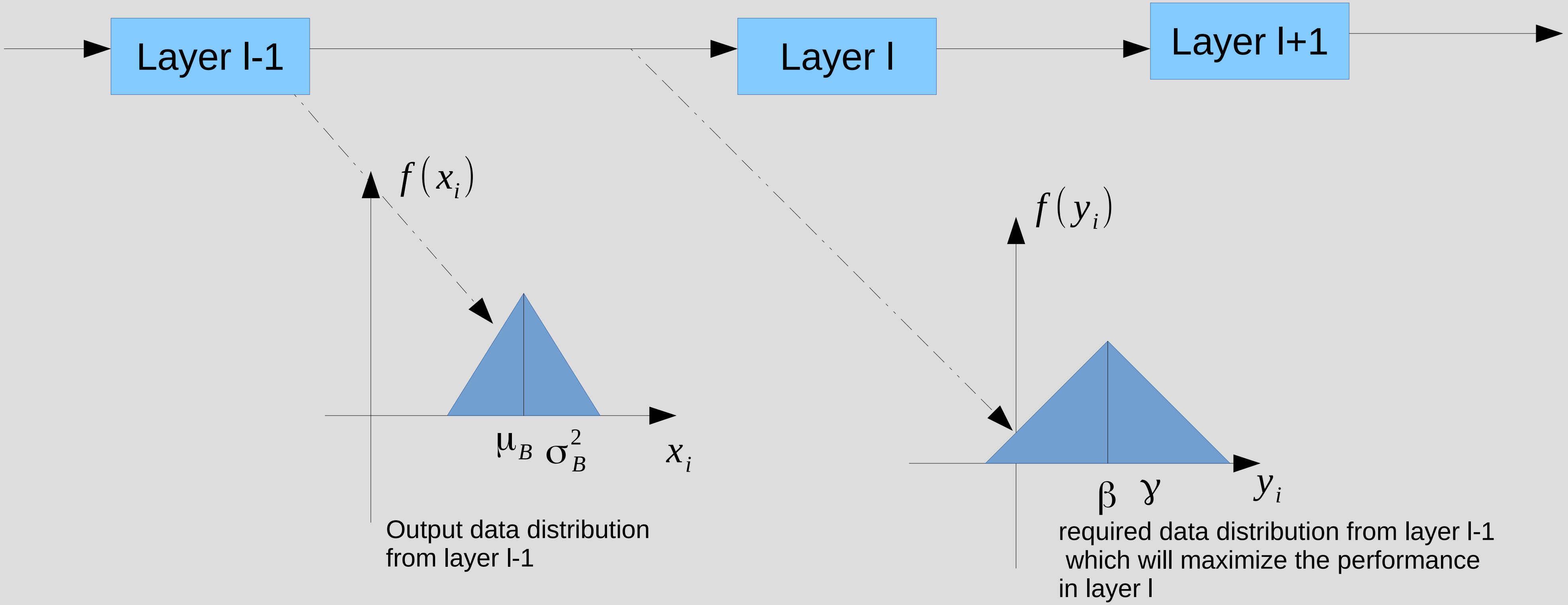
Batch normalization Intro.

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=0}^{M-1} x_i \qquad \sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=0}^{M-1}(x_i-\mu_B)^2 \qquad \hat{x}_i \leftarrow \frac{x_i-\mu_B}{\sqrt{\sigma_B^2+e}} \qquad y_i \leftarrow \gamma\,\hat{x}_i+\beta$$



Distribution transformation

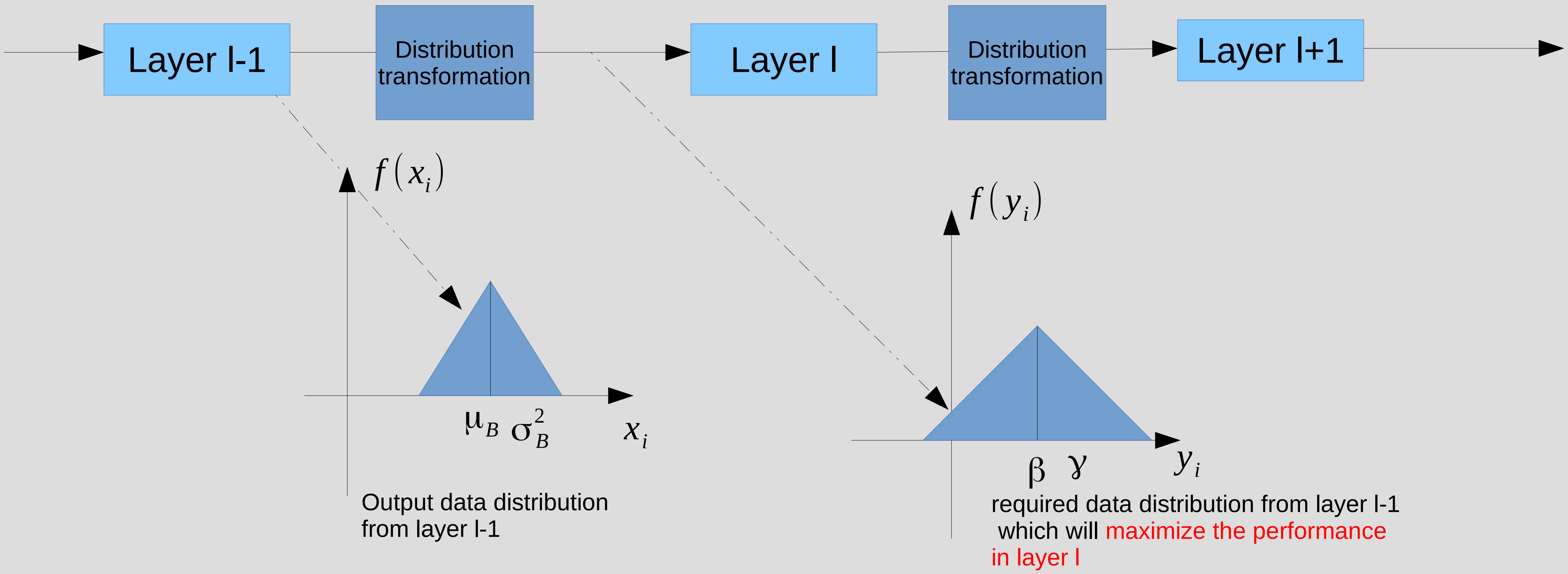
Why we need Batch normalization ?



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

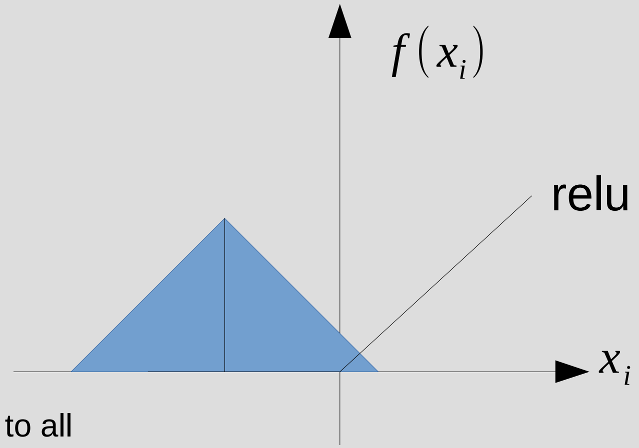
Why we need Batch normalization ? Cont.



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

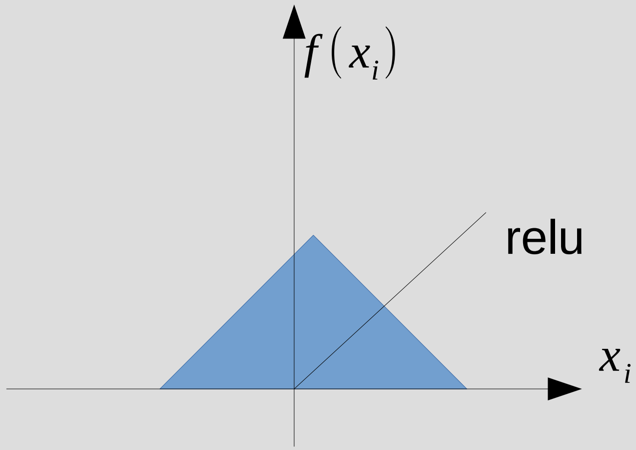
Why BN makes training faster ?

When training DNN with relu activation, if the initial learning rate is set to be too big if 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 ?

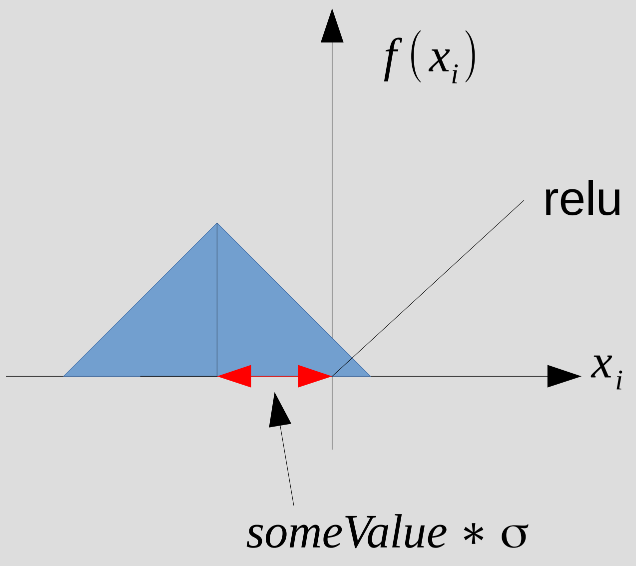
If we add a constrain to β update during training,

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

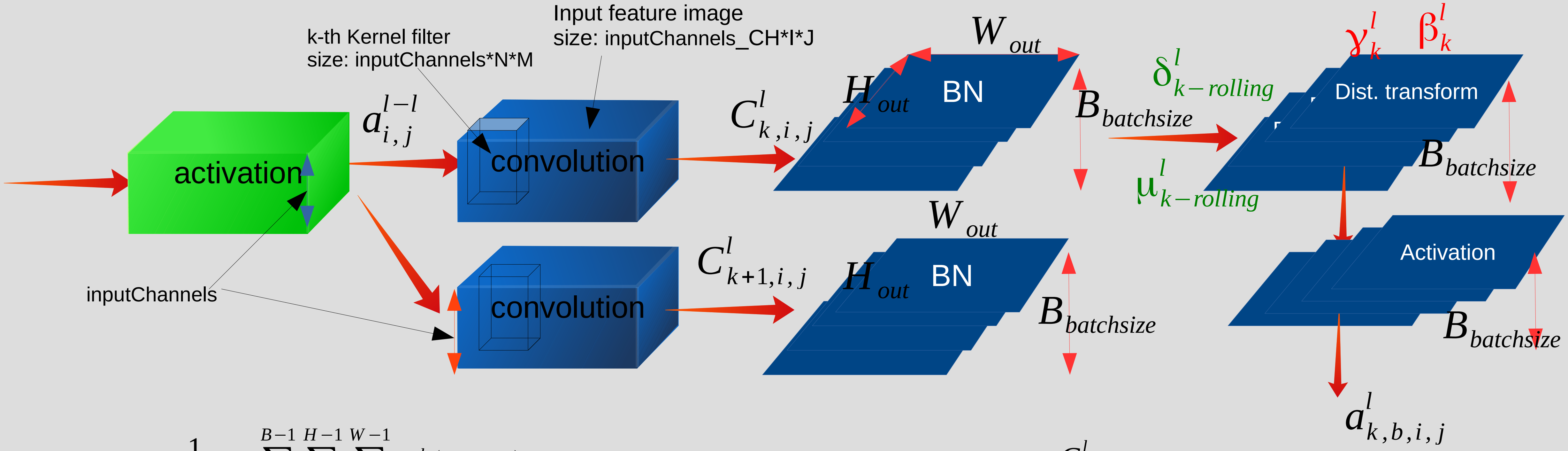
If $\beta < -someValue * \sigma$

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

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



Batch normalization following conv.



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

K-th Channel mean

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

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

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

K-th Channel variance

Back-pro at BN

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

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

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

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

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 $\frac{\partial Loss}{\partial C^l_{b,i,j}}$

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

$$\sigma^2_k = \frac{1}{B*H*W}\sum_{b=0}^{B-1}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1} (C^l_k(b,i,j)-\mu_k)^2$$

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

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

$$\frac{dl}{dC^l_{b,i,j}} = \sum_{b'=0}^{B-1}\sum_{i'=0}^{H-1}\sum_{j'=0}^{W-1} \frac{dl}{d\hat{C}^l_{b',i',j'}} * \frac{d\frac{C^l_{b',i',j'}-\mu_k}{\sqrt{\sigma^2_k+e}}}{dC^l_{b,i,j}}$$

Substitute y in and expand a little bit

$$= \sum_{b'=0}^{B-1}\sum_{i'=0}^{H-1}\sum_{j'=0}^{W-1} \frac{dl}{d y^l_{b',i',j'}} * \frac{dy^l_{b',i',j'}}{d\hat{C}^l_{b',i',j'}} * \frac{d\frac{C^l_{b',i',j'}-\mu_k}{\sqrt{\sigma^2+e}}}{dC^l_{b,i,j}}$$

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

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

Back-pro at BN $\frac{\partial Loss}{\partial C^l_{b,i,j}}$

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

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

$$\sigma^2_k = \frac{1}{B*H*W}\sum_{b=0}^{B-1}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1} (C^l_k(b,i,j)-\mu_k)^2$$

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

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

Solve these 3 components one by one

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

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

Back-pro at BN $\frac{\partial Loss}{\partial C^l_{b,i,j}}$

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

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

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

$$\frac{d\mu_k}{dC^l_{b,i,j}}=d\frac{\frac{1}{B*N*M}\sum_{b'=0}^{B-1}\sum_{i'=0}^{H-1}\sum_{j'=0}^{W-1}C^l_{b',i',j'}}{d\,C^l_{b,i,j}}=\frac{1}{B*N*M}\delta_{b',b}\delta_{i',i}\delta_{j',j}=\frac{1}{B*N*M}$$

$$\delta_{b',b}=1\ if\ b'=b\ else\ \delta_{b',b}=0$$

Back-pro at BN $\frac{\partial Loss}{\partial C^l_{b,i,j}}$

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

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

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

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

$$\frac{d\sigma^2}{dC^l_{b,i,j}}=\frac{1}{B*N*M}*d\frac{\sum_{b'=0}^{B-1}\sum_{i'=0}^{N-1}\sum_{j'=0}^{M}(C^l_{b',i',j'}-\mu_{ch})^2}{dC^l_{b,i,j}}$$

$$=\frac{1}{B*N*M}*2*(C^l_{b,n,m}-\mu_k)$$

Back-pro at BN $\frac{\partial Loss}{\partial C^l_{b,i,j}}$

The final formula for derivative w.r.t input

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

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

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

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