Outline

- Binary Network and XNOR Network

Binary Network and XNOR Network

Reference:

XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

Innovation:use binary weight filter(only 1 and -1) to replace the traditional weight filter.

Binary Network

I_l : *input for the* l^{th} *layer*   *size* : $[\mathrm{c}, \mathrm{w}_{in}, \mathrm{h}_{in}]$

W_{lk} : *the* k^{th} *weight filter in the* l^{th} *layer*

K^l : *the number of weight filter in the* l^{th} *layer*

$\tilde{W}$: *binary weight filter*

W : *real – value weight filter*

convolutional filters do not have bias terms

Binary Network and XNOR Network

for each signal weight: $W \approx \alpha B$

$\alpha : a\ scaling\ \ factor \in R^{+}$ $B : binary\ \ filter \in \{+1, -1\}^{c \times w \times h}$

$$J(\mathrm{B}, \alpha) = \| W - \alpha B \|^{2}$$

$$\alpha^{*}, B^{*} = \arg\min J(\mathrm{B}, \alpha)$$

$\tilde{W}_{lk} \approx A_{lk} B_{lk}$ $A_{lk} = \alpha$ $B_{lk} = B$ $I * \tilde{W} \approx (I \oplus B)\alpha$

$\oplus$ *indicates a convolution without any multiplication* (XNOR)

Binary Network and XNOR Network

$$J(\mathrm{B},\alpha) = \| W - \alpha B \|^2$$

Example:

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$W^T W = \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} w_{11}^2 + w_{21}^2 & * \\ * & w_{12}^2 + w_{22}^2 \end{bmatrix}$$

$$\| A \|_2^2 = a_{11}^2 + a_{12}^2 + \cdots\cdots a_{nn}^2 = trac(A^T A) \triangleq A^T A \quad A \in R^{n \times n}$$

$$A^T B = a_{11} b_{11} + \cdots\cdots + a_{\mathrm{nn}} b_{nn}$$

Binary Network and XNOR Network

$$J(\mathrm{B},\alpha) = \| W - \alpha B \|^2 = \| \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} - \alpha \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \|^2$$

$$= (\sqrt{(w_{11} - ab_{11})^2 + (w_{12} - ab_{12})^2 + (w_{21} - ab_{21})^2 + (w_{22} - ab_{22})^2}\,)^2$$

$$= (w_{11}^2 - 2aw_{11}b_{11} + a^2 b_{11}^2) + \cdots\cdots + (w_{22}^2 - 2\alpha w_{22}b_{22} + \alpha^2 b_{22}^2)$$

$$= (w_{11}^2 + w_{12}^2 + w_{21}^2 + w_{22}^2) - 2\alpha(w_{11}b_{11} + w_{12}b_{12} + w_{21}b_{21} + w_{22}b_{22}) + \alpha^2 (b_{11}^2 + b_{12}^2 + b_{21}^2 + b_{22}^2)$$

$$= \| W \|^2 - 2\alpha W^T B + \alpha^2 \| B \|^2$$

$$= \alpha^2 B^T B - 2\alpha W^T B + W^T W$$

$$J(\mathrm{B},\alpha) = \alpha^2 B^T B - 2\alpha W^T B + W^T W$$

$$= \alpha^2 n - 2\alpha W^T B + c$$

$$B \in \{+1, -1\}^{n\times n}$$

$$\underline{B^T B = (\pm 1)_{11}^2 + (\pm 1)_{12}^2 + \cdots\cdots + (\pm 1)_{nn}^2 = n \times n}$$

Weight :[3, 3]   n=9

Binary Network and XNOR Network

$$B^* = \arg\max\{W^T B\} \; s.t. \; B \in \{+1,-1\}^n$$

Variational inference:
if B is a function with parameter λ
we can use this to find function B
to replace complexity function W!
Just in this demo,the B is simple +1,-1.-------Bayesian neural networks

$$B^* = sign(W) = \begin{cases} 1 & W \geq 0 \\ -1 & W < 0 \end{cases}$$

$$\frac{\partial J}{\partial \alpha} = 2n\alpha - 2W^T B = 0 \qquad \alpha^* = \frac{W^T B^*}{n} = \frac{W^T sign(W)}{n} = \frac{\sum |W_i|}{n} = \frac{1}{n}\|W\|_{l1}$$

a binary weight filter can be simply achieved by taking the sign of weight values. The optimal scaling factor is the average of absolute weight values.

Binary Network and XNOR Network

Binary Network block

Convolutionl

Batch Normalization

Active function

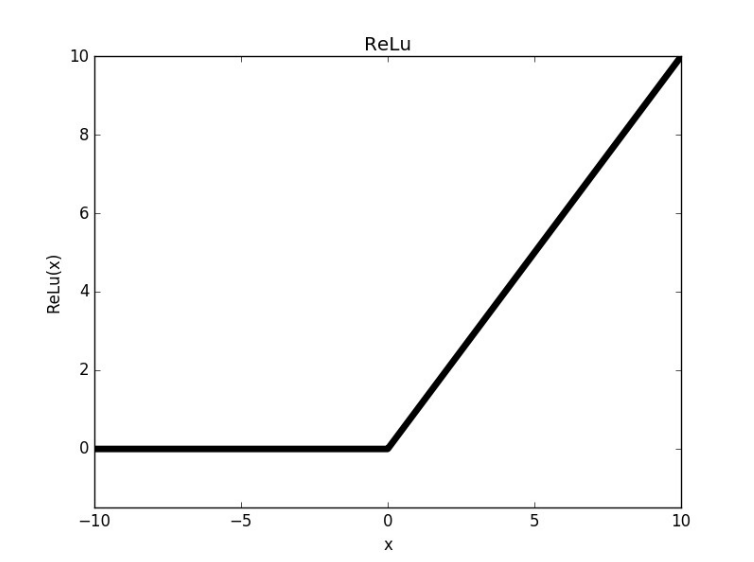
Pooling

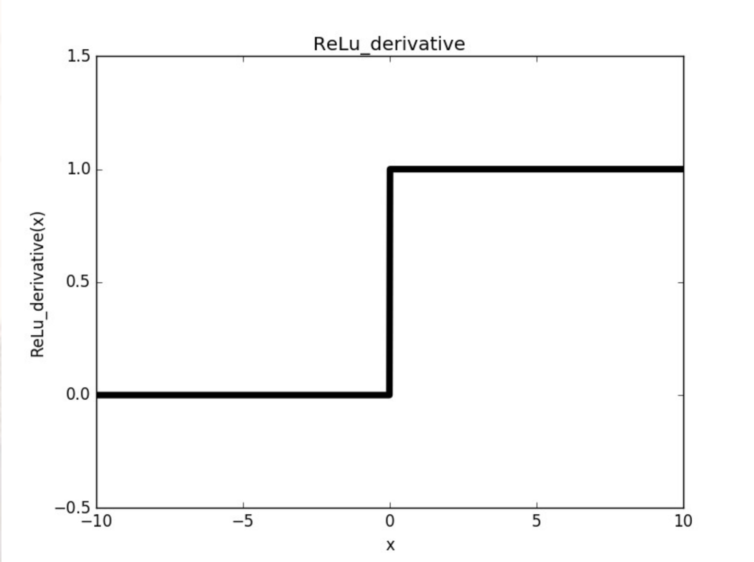
$$hard\_tanh(\mathrm{x}) = \max(-1, \min(1, x)) = \begin{cases} 1 & x > 1 \\ x & -1 \le x \le 1 \\ -1 & x < -1 \end{cases}$$

Binary Network and XNOR Network

$$\mathrm{ReLu(x)} = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

$$\mathrm{ReLu\_derivative(x)} = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

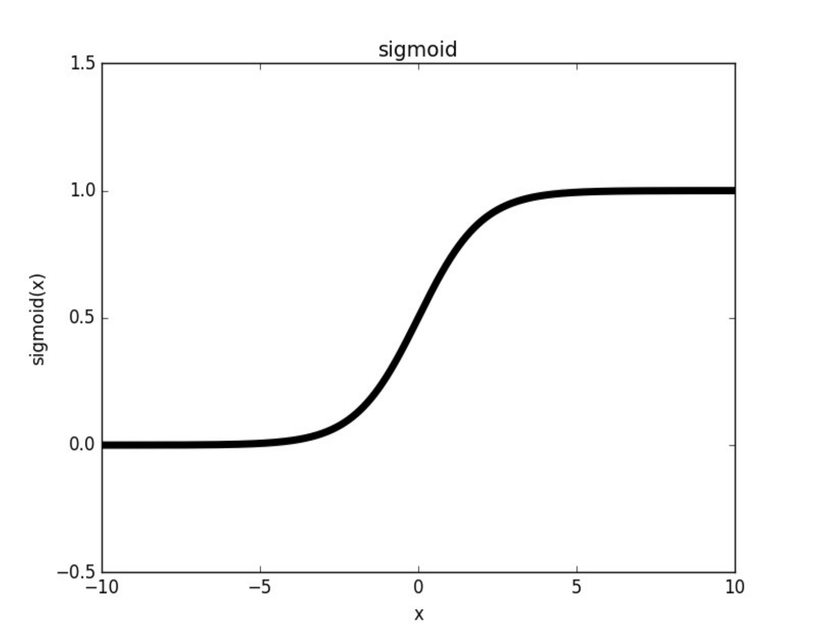


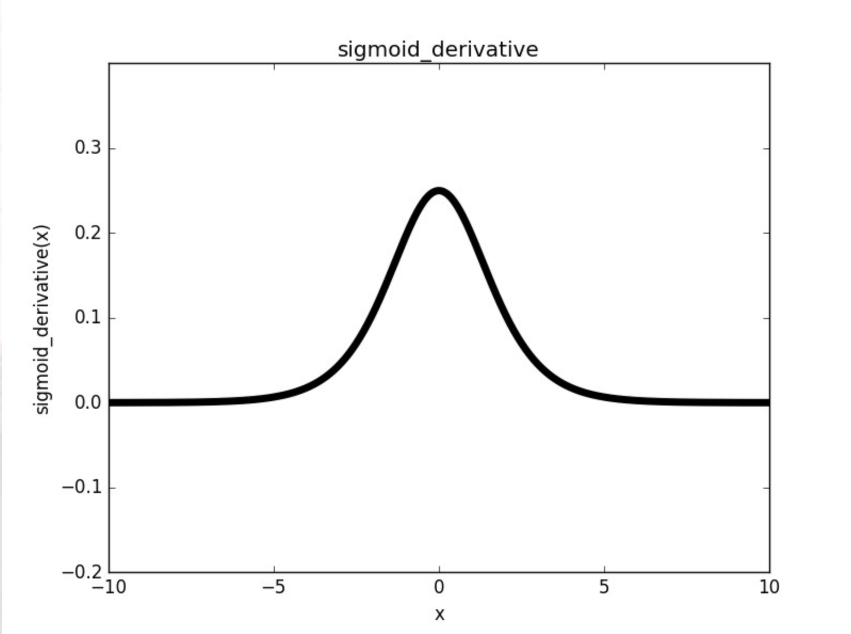


Binary Network and XNOR Network

$$sigmoid(\mathrm{x}) = \frac{1}{1+e^{-x}}$$

$$sigmoid\_\mathrm{derivative}(\mathrm{x}) = \frac{1}{1+e^{-x}}\left(1-\frac{1}{1+e^{-x}}\right)$$

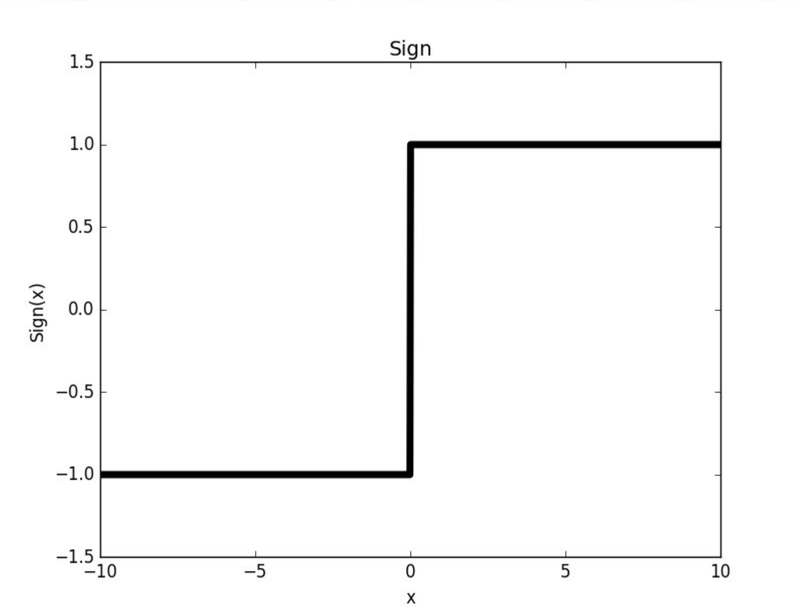


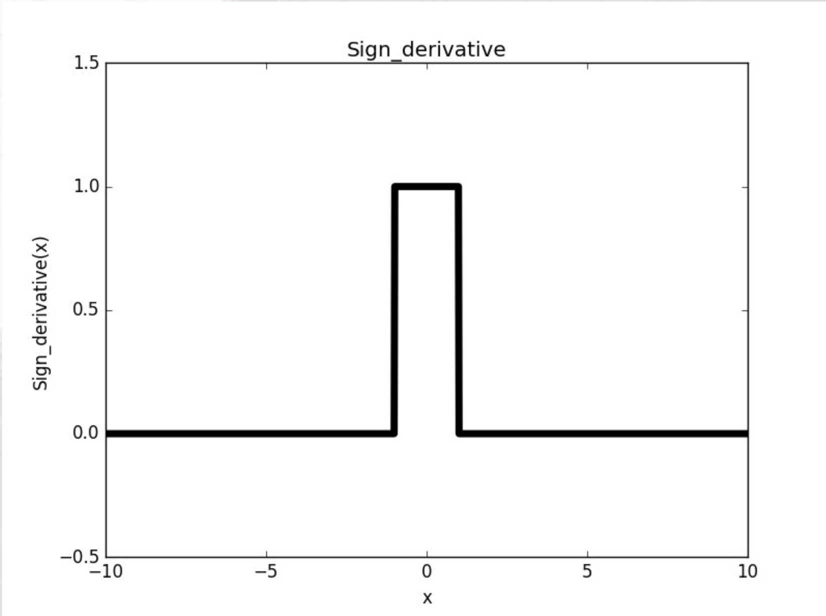


Binary Network and XNOR Network

$$Sign(\mathrm{x}) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

$$Sign\_derivative(\mathrm{x}) = \begin{cases} 1 & -1 \leq x \leq 1 \\ 0 & others \end{cases}$$

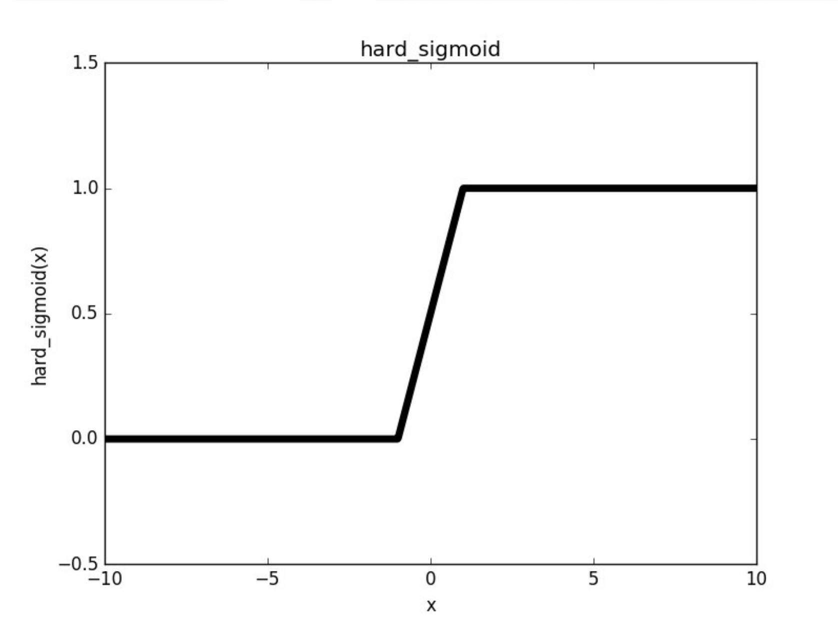


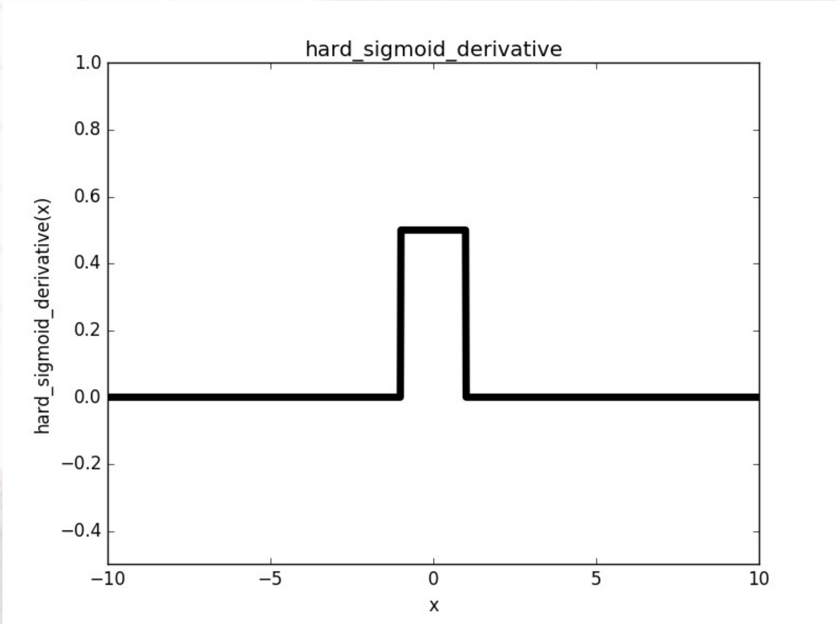


Binary Network and XNOR Network

$$hard\_sigmoid(\mathrm{x}) = \max(0, \min(1, \frac{x+1}{2})) = \begin{cases} 1 & x > 1 \\ \frac{x+1}{2} & -1 \le x \le 1 \\ 0 & x < -1 \end{cases}$$

$$hard\_sigmoid\_derivative(\mathrm{x}) = \begin{cases} 0.5 & -1 \le x \le 1 \\ 0 & others \end{cases}$$

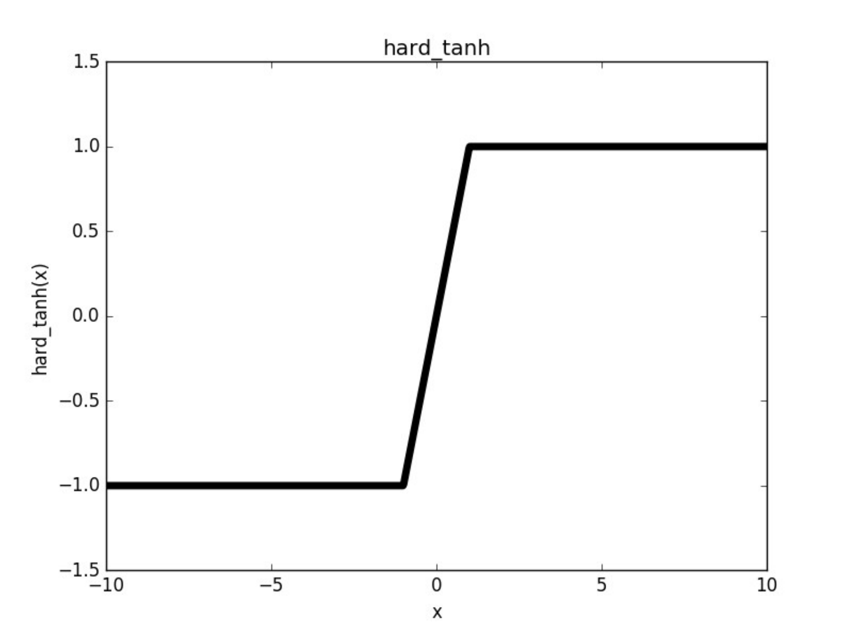


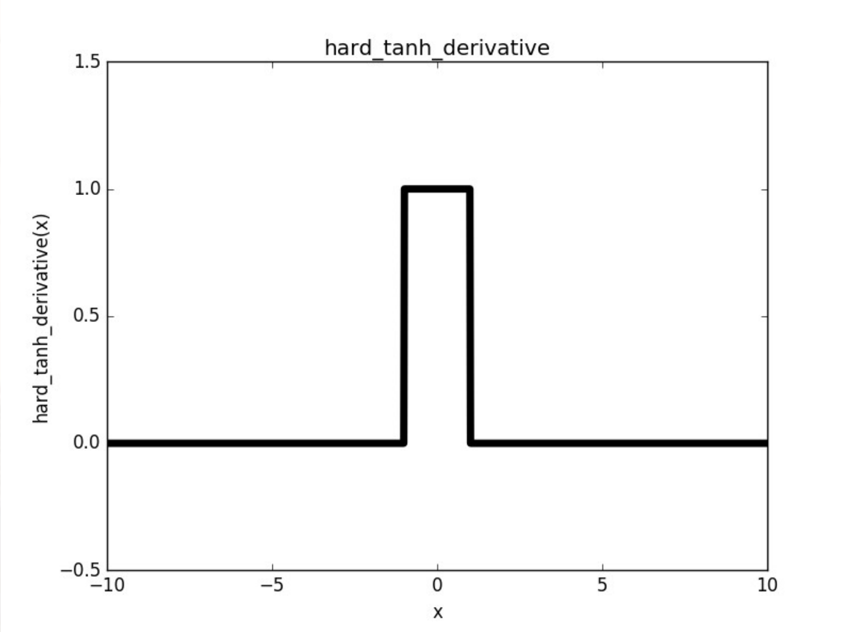


Binary Network and XNOR Network

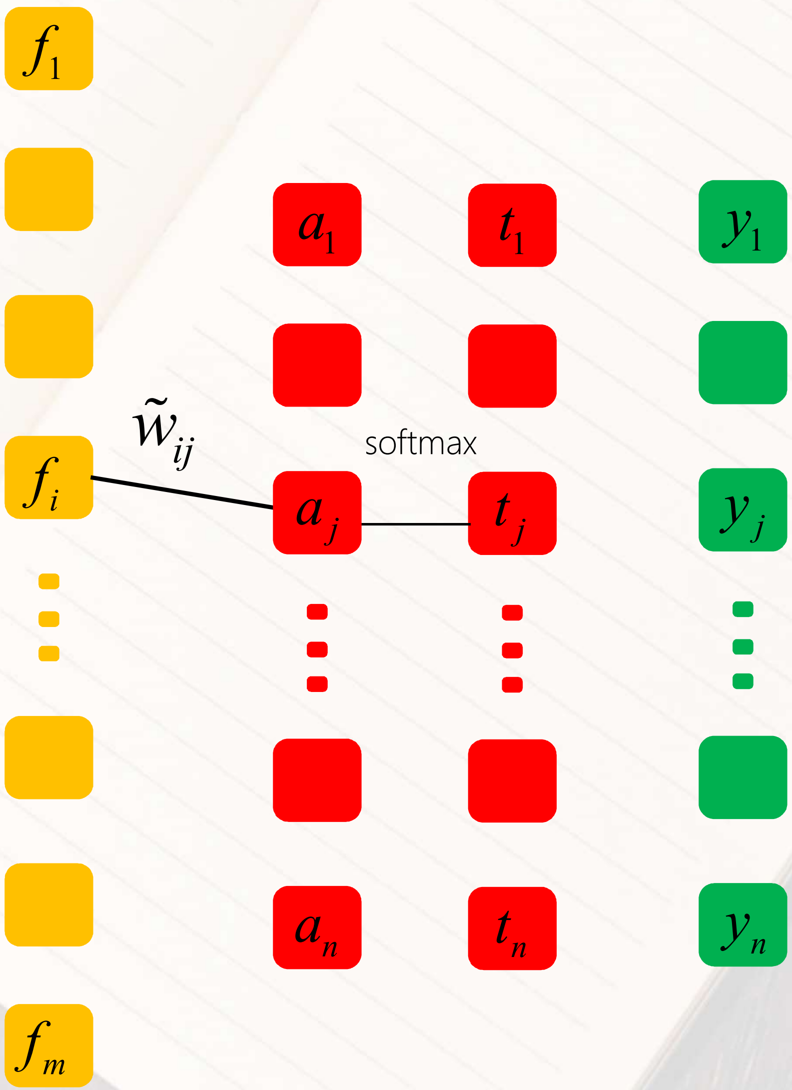
$$hard\_tanh(x) = \max(-1, \min(1, x)) = \begin{cases} 1 & x > 1 \\ x & -1 \le x \le 1 \\ -1 & x < -1 \end{cases}$$

$$hard\_tanh\_derivative(x) = \begin{cases} 1 & -1 \le x \le 1 \\ 0 & others \end{cases}$$





Binary Network and XNOR Network



$$t = image * \tilde{W}$$

$$C = \frac{1}{2} \| t - y \|^2 \qquad \delta = \frac{\partial C}{\partial t} = t - y$$

sigmoid

$$Softmax(x_k) = \frac{e^{x_k}}{\sum_{i=1}^{m} e^{x_i}}$$

$$\frac{\partial C}{\partial \tilde{W}} = \delta F \qquad \frac{\partial C}{\partial \tilde{W}} ---- \frac{\partial C}{\partial W}$$

Binary Network and XNOR Network

Cross entropy function:

$$\log(\frac{M}{N}) = \log M - \log N$$

$$t_j = \frac{e^{a_j}}{\sum_{i=1}^{n} e^{a_j}}$$

$$C = -\sum_i y_i \ln(t_i)$$

$$C = -\{y_1 \ln(\frac{e^{a_1}}{\sum_{i=1}^{n} e^{a_i}}) + y_2 \ln(\frac{e^{a_2}}{\sum_{i=1}^{n} e^{a_i}}) + \cdots + y_j \ln(\frac{e^{a_j}}{\sum_{i=1}^{n} e^{a_i}}) + \cdots + y_n \ln(\frac{e^{a_n}}{\sum_{i=1}^{n} e^{a_i}})\}$$

$$= -\{y_1[\ln(e^{a_1}) - \ln(\sum_{i=1}^{n} e^{a_i})] + \cdots + y_j[\ln(e^{a_j}) - \ln(\sum_{i=1}^{n} e^{a_j})] + \cdots\}$$

$$= -\{y_1[a_1 - \ln(\sum_{i=1}^{n} e^{a_i})] + \cdots y_j[a_j - \ln(\sum_{i=1}^{n} e^{a_i})] + \cdots\}$$

$$= y_1[\ln(\sum_{i=1}^{n} e^{a_i}) - a_1] + \cdots y_j[\ln(\sum_{i=1}^{n} e^{a_i}) - a_j] + \cdots$$

Binary Network and XNOR Network

$$(\log_a x)' = \frac{1}{x \ln a}$$

$$\frac{\partial C}{\partial a_j} = \frac{\partial}{\partial a_j}\{y_1[\ln(\sum_{i=1}^{n} e^{a_i}) - a_1] + \cdots y_j[\ln(\sum_{i=1}^{n} e^{a_i}) - a_j] + \cdots\}$$

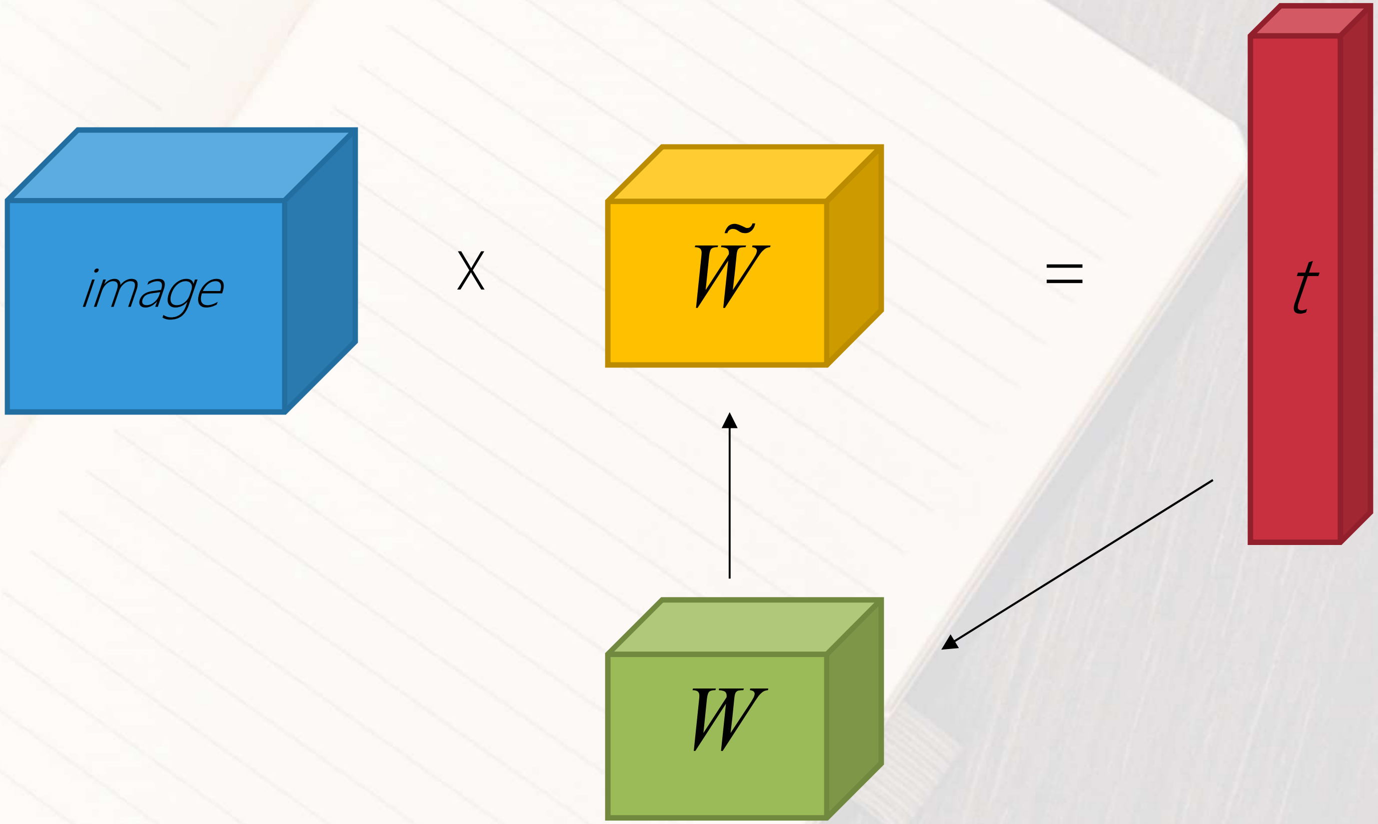
$$= y_1[\frac{\partial \ln(e^{a_1} + \cdots e^{a_j} + \cdots)}{\partial a_j} - 0] + \cdots y_j[\frac{\partial \ln(e^{a_1} + \cdots e^{a_j} + \cdots)}{\partial a_j} - 1] + \cdots$$

$$= y_1 \frac{1}{\sum_i e^{a_i}} e^{a_j} + y_2 \frac{1}{\sum_i e^{a_i}} e^{a_j} + \cdots y_j[\frac{1}{\sum_i e^{a_i}} e^{a_j} - 1] + \cdots y_n \frac{1}{\sum_i e^{a_i}} e^{a_j}$$

$$= \frac{1}{\sum_i e^{a_i}} e^{a_j} (y_1 + y_2 + \cdots y_n) - y_j$$

$$= Softmax(a_j)(y_1 + y_2 + \cdots y_n) - y_j$$

Binary Network and XNOR Network



Binary Network and XNOR Network

$$\tilde{W}_i = \alpha B_i = \frac{\sum |W_i|}{n} * \mathrm{sign}(W_i) \qquad \frac{\partial \mathrm{sign}(r)}{\partial r} = r1_{|r|\le 1} = 1_{|r|\le 1} = \begin{cases} 1, & |r| \le 1 \\ 0, & otherwise \end{cases}$$

$$\frac{\partial C}{\partial W_i} = \frac{\partial C}{\partial \tilde{W}_i}\frac{\partial \tilde{W}_i}{\partial W_i} = \frac{\partial C}{\partial \tilde{W}_i}\frac{\partial}{\partial W_i}[\frac{\sum |W_i|}{n}\mathrm{sign}(W_i)]$$

$$= \frac{\partial C}{\partial \tilde{W}_i}[\frac{\partial \frac{\sum |W_i|}{n}}{\partial W_i}\mathrm{sign}(W_i) + \frac{\sum |W_i|}{n}\frac{\partial \,\mathrm{sign}(W_i)}{\partial W_i}]$$

$$= \frac{\partial C}{\partial \tilde{W}_i}[\frac{1}{n}\frac{\partial (c + |\mathrm{W}_i|)}{\partial W_i}\mathrm{sign}(W_i) + \alpha W_i 1_{|\mathrm{W}_i|\le 1}]$$

$$= \frac{\partial C}{\partial \tilde{W}_i}\{\frac{1}{n} * \begin{cases} 1*1 & w \ge 0 \\ -1*-1 & w < 0 \end{cases} + \alpha W_i 1_{|\mathrm{W}_i|\le 1}\}$$

$$= \frac{\partial C}{\partial \tilde{W}_i}[\frac{1}{n} + \alpha \frac{\partial \,\mathrm{sign}(W_i)}{\partial W_i}]$$

Binary Network and XNOR Network

$$\frac{\partial C}{\partial w_{ij}} = \frac{\partial C}{\partial \tilde{w}_{ij}} \frac{\partial \tilde{w}_{ij}}{\partial w_{ij}} = \frac{\partial C}{\partial \tilde{w}_{ij}} \frac{\partial}{\partial w_{ij}} \{ \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} | w_{ij} |}{n} \mathrm{sign}(w_{ij}) \}$$

$$= \frac{\partial C}{\partial \tilde{w}_{ij}} [\frac{1}{m * n} + \alpha \frac{\partial sign(w_{ij})}{\partial w_{ij}}]$$

Update: $$W^{new} = W^{old} - lr * \frac{\partial C}{\partial W^{old}}$$

Binary Network and XNOR Network

Batch_Normalization(BN):

Reference: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

BN is norm the output, for each channel:

if the output size:[weight, height, channel]

For each[weight, height] we use once BN.

Number of BN == number of output channel

$$u = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$input:[x_1, x_2, x_3, \cdots x_m]$$

$$\mathrm{var} = \sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(\mathrm{x}_i - \mathrm{u})^2$$

$$\varepsilon = 10^{-6}$$

$$\hat{x}_i = \frac{x_i - u}{\sqrt{\mathrm{var} + \varepsilon}}$$

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

Binary Network and XNOR Network

BN-BP: cost function: l

$$\frac{\partial l}{\partial \gamma}=\sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\hat{x}_i \qquad \frac{\partial l}{\partial \beta}=\sum_{i=1}^{m}\frac{\partial l}{\partial y_i}$$

$$\gamma=\gamma-lr*\frac{\partial l}{\partial \gamma}$$

$$\beta=\beta-lr*\frac{\partial l}{\partial \beta}$$

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

$$\frac{\partial l}{\partial var}=\sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}_i}\frac{\partial \hat{x}_i}{\partial var}=\sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\gamma(x_i-u)(-\frac{1}{2})(var+\varepsilon)^{-\frac{3}{2}}$$

$$\frac{\partial l}{\partial u}=(\sum_{i=1}^{m}\frac{\partial l}{\partial \hat{x}_i}\frac{-1}{\sqrt{var+\varepsilon}})+\frac{\partial l}{\partial var}\frac{\sum_{i=1}^{m}-2(x_i-u)}{m}$$

$$\frac{\partial l}{\partial x_i}=\frac{\partial l}{\partial \hat{x}_i}\frac{1}{\sqrt{var+\varepsilon}}+\frac{\partial l}{\partial var}\frac{2(x_i-u)}{m}+\frac{\partial l}{\partial u}\frac{1}{m}$$

Binary Network and XNOR Network

A typical block in cnn

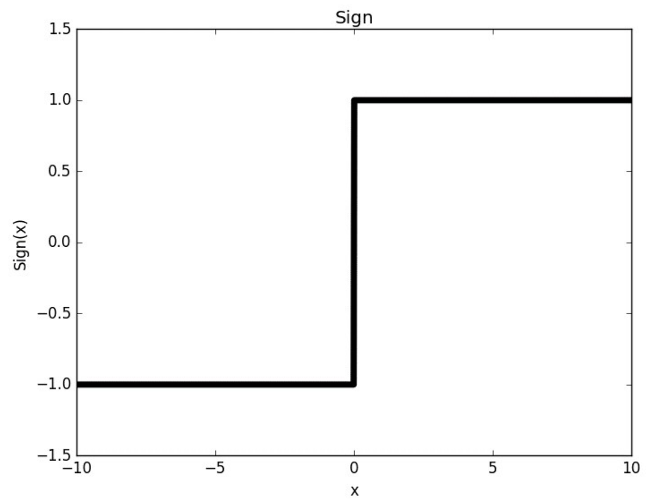
- Convolutionl
- Add biases
- Batch Normalization
- Active function
- Pooling

A block in XNOR-Net

- Batch Normalization
- Active function
- Convolutionl
- Pooling

Binary Network and XNOR Network

Why XNOR-Net has different block?



Convolutio
nl

Add biases

Batch Normalization

Active function

Pooling

1	-1	1	1
-1	1	1	1
-1	-1	-1	1
1	-1	1	-1

1	1
1	1

Information loss!

Binary Network and XNOR Network

Convolutionl

Add biases

Batch Normalization

Pooling

Active function

-0.2	0.1	0.5	-0.3
-0.5	-0.4	-0.1	-0.8
0.7	-0.8	0.4	0.9
-0.5	0.1	0.2	0.7

0.1	0.5
0.7	0.9

1	1
1	1

Information loss!

Binary Network and XNOR Network

A block in XNOR-Net

Batch Normalization

0.3	-0.6	0.7	0.3
-0.5	0.4	0.1	0.2
-0.9	-0.8	-0.8	0.3
0.5	-0.7	0.4	-0.7

Active function

1	-1	1	1
-1	1	1	1
-1	-1	-1	1
1	-1	1	-1

Convolutionl

-0.2	0.1	0.5	**-0.3**
-0.5	-0.4	-0.1	-0.8
0.7	-0.8	0.4	0.9
-0.5	0.1	0.2	0.7

Pooling

0.1	0.5
0.7	0.9

Next Week

- One shot with Mul CNN
- Scene Parsing