Institut Mines-Télécom

APPROXIMATE COMPUTING FOR EMBEDDED MACHINE LEARNING

YANG XUECAN

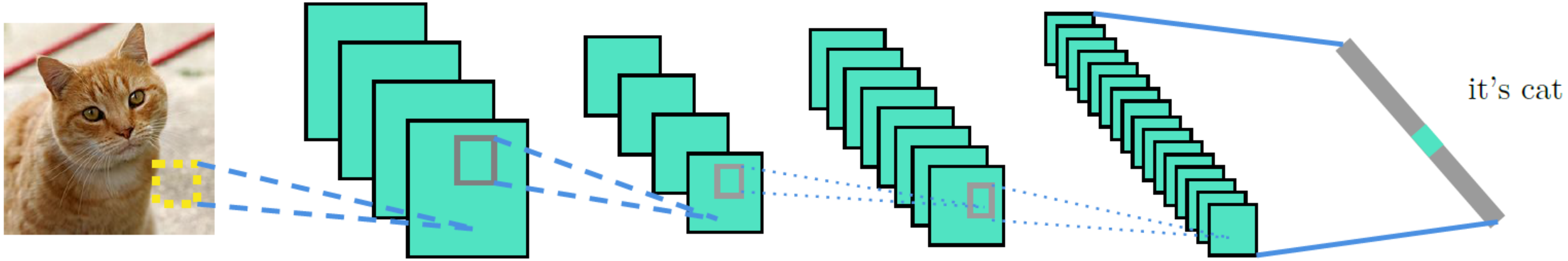
OUTLINE

1. Motivation and Related works
2. Approximate Operation to multiplication
3. Building MinConvNets with approximate operation
4. Conclusion

1. Motivation and Related works

Institut Mines-Télécom

USE CASE OF DEEP CONVOLUTIONAL NEURAL NETWORK

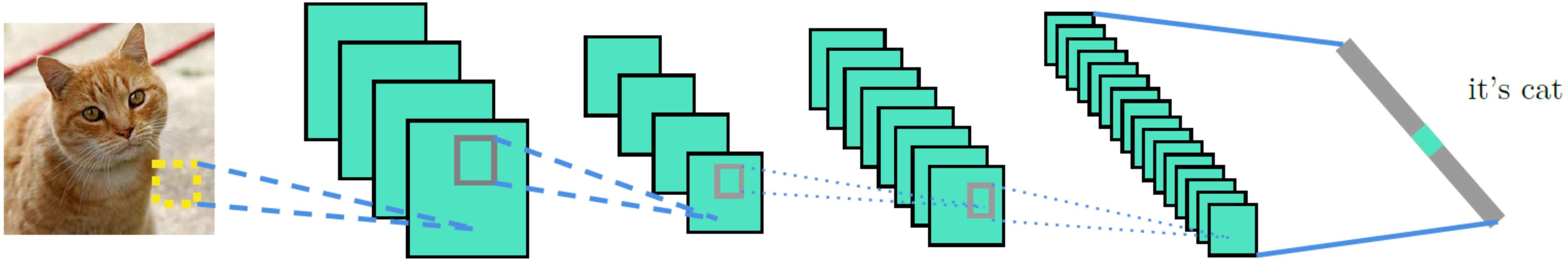


Classification:
Traffic lights is red !

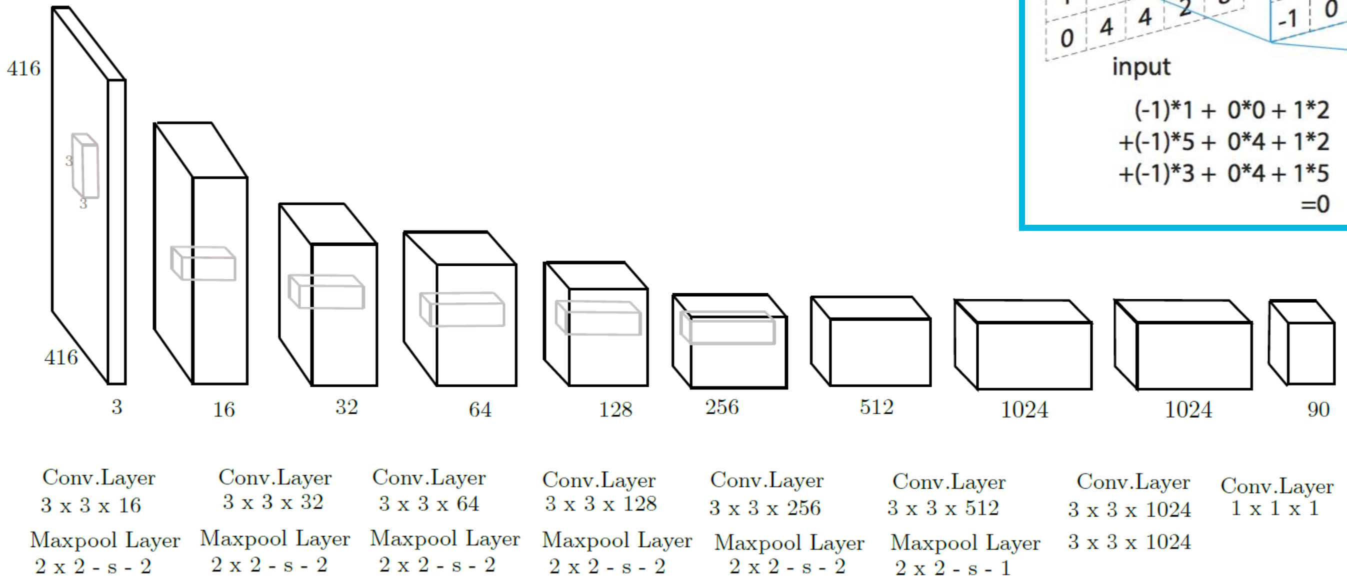
Object detection:
The car is here !

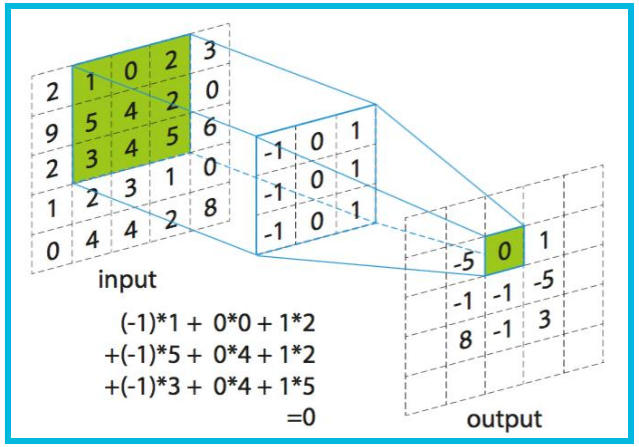
Object tracking:
It has to pay a fine !

USE CASE OF DEEP CONVOLUTIONAL NEURAL NETWORK

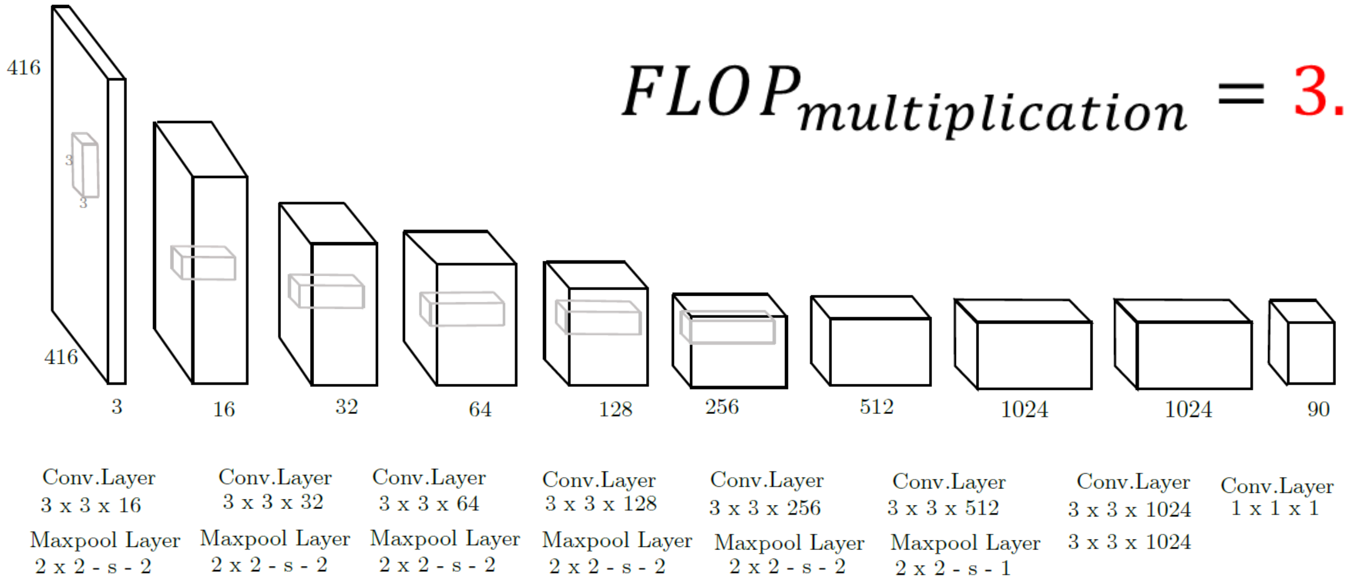


TINY-YOLO [REDMON ET AL.'2016] FOR OBJECT DETECTION





TINY-YOLO [REDMON ET AL.'2016] FOR OBJECT DETECTION



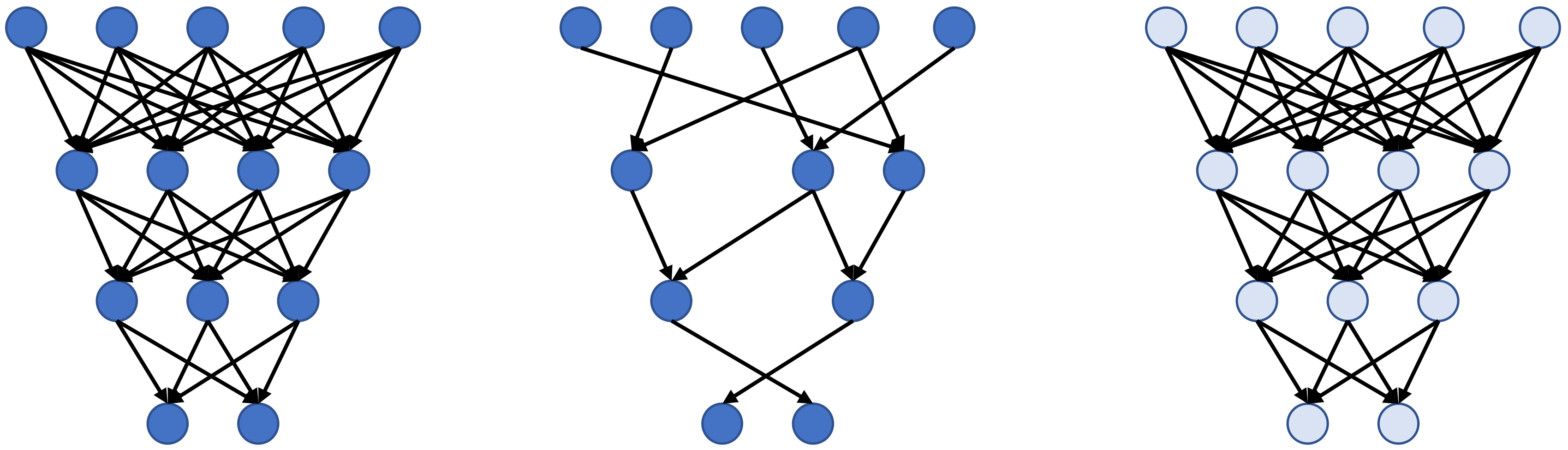
$$FLOP_{multiplication} = 3.48G$$

- **Challenges for embedded systems**
  - Capacity of computing (multiplicator etc.),
  - Memory or bandwidth for loading the data.

MOTIVATION

- How to reduce the computing resources required for convolution which includes a large volume of multiplications?

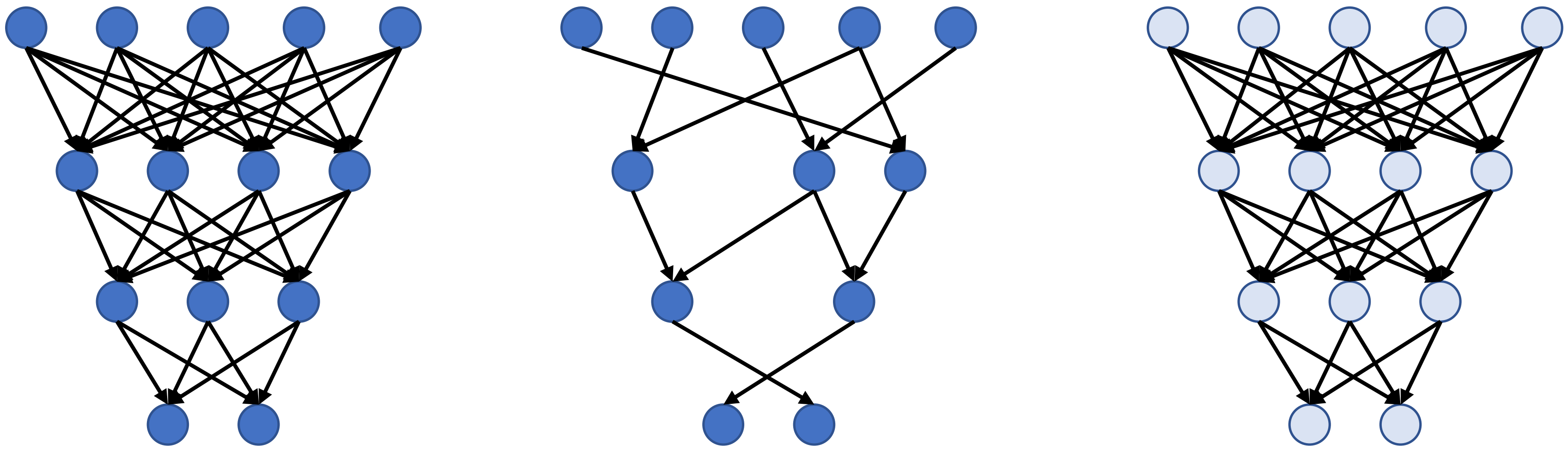
RELATED WORKS TO REDUCE THE COMPUTING RESOURCES



Original Network   Pruning Network   Quantization Network

32 bits floating point   8 bits integer

RELATED WORKS TO REDUCE THE COMPUTING RESOURCES



Original Network

Pruning Network

Quantization Network

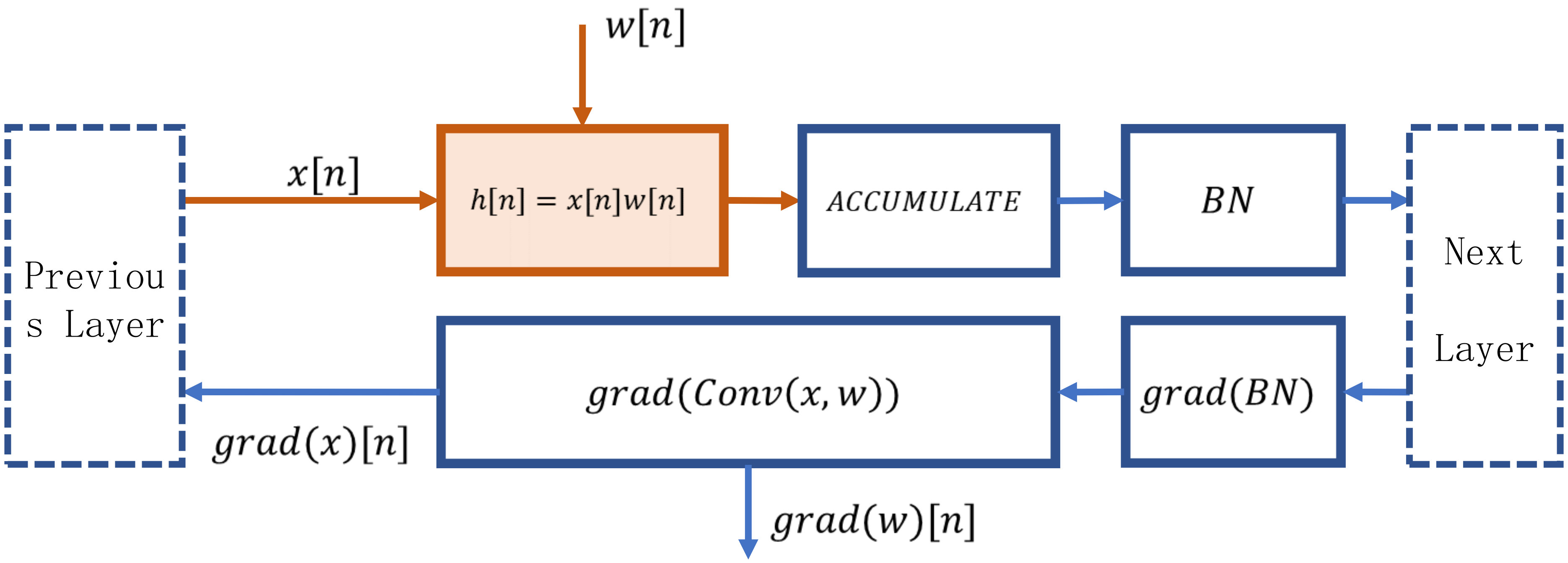
32 bits floating point

8 bits integer

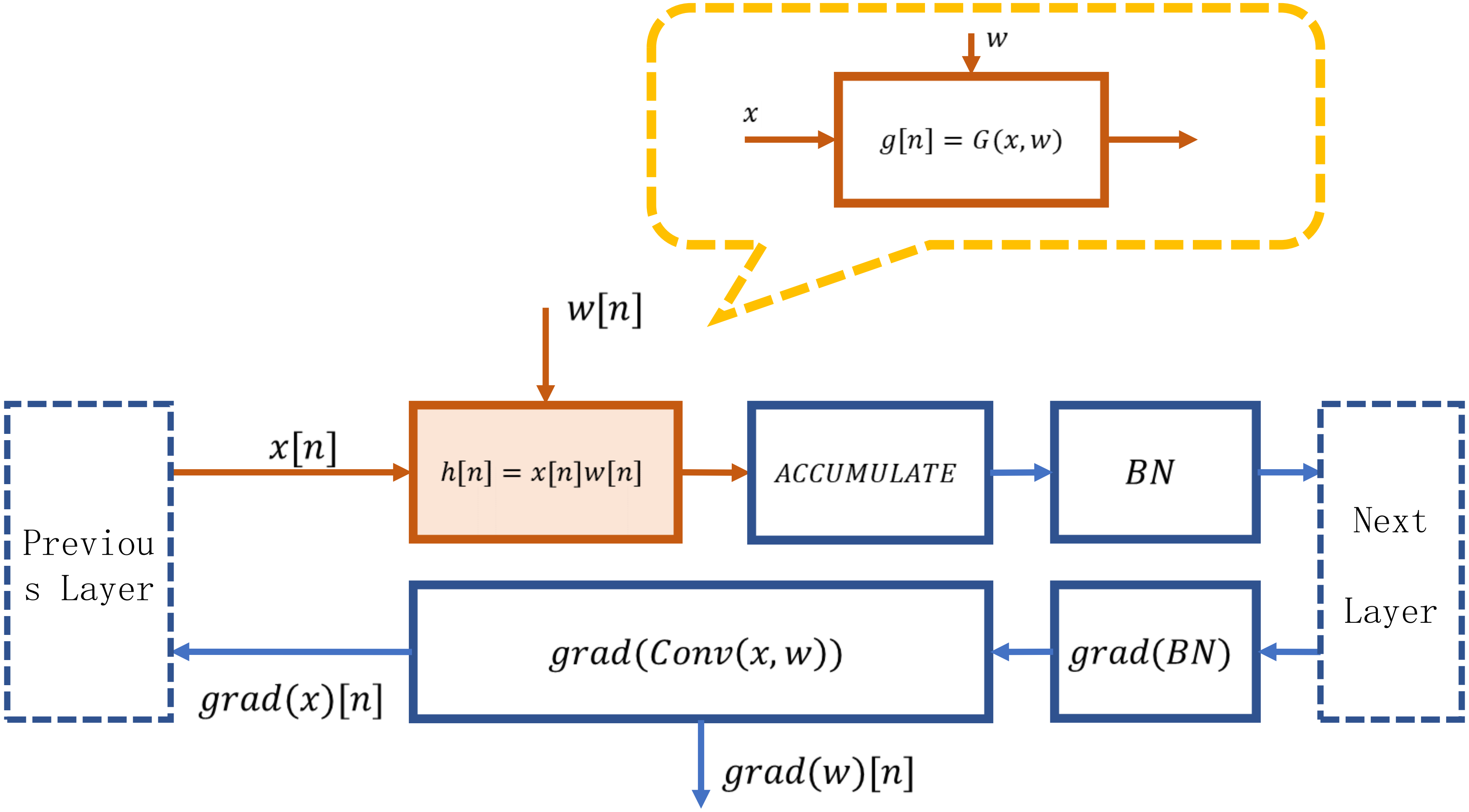
Why always multiplication?

2. Approximate Operation to multiplication

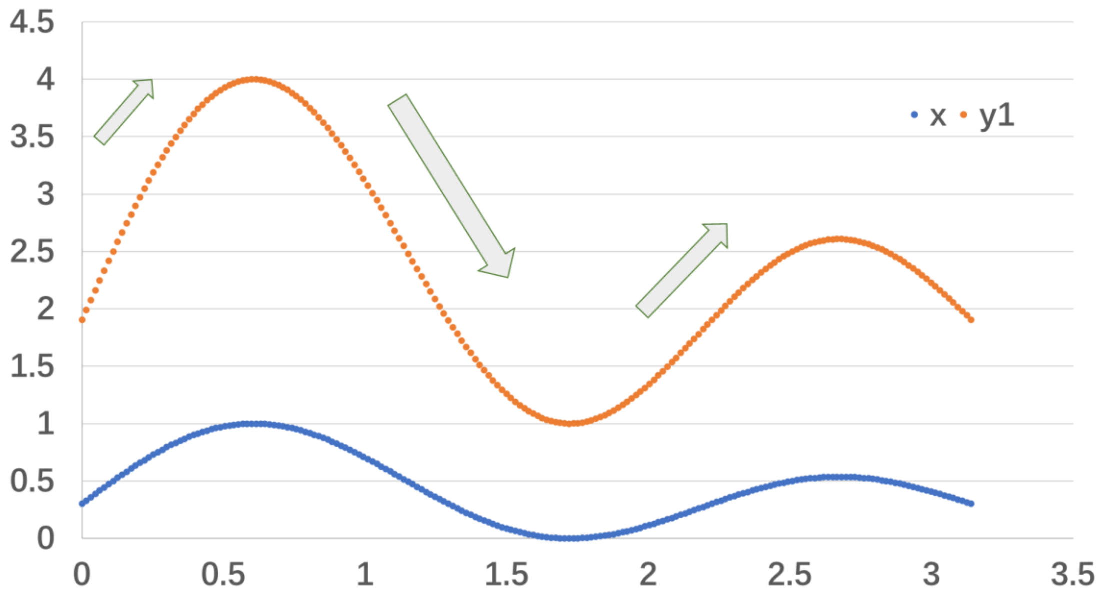
USING APPROXIMATE OPERATION INSTEAD OF MULTIPLICATION?

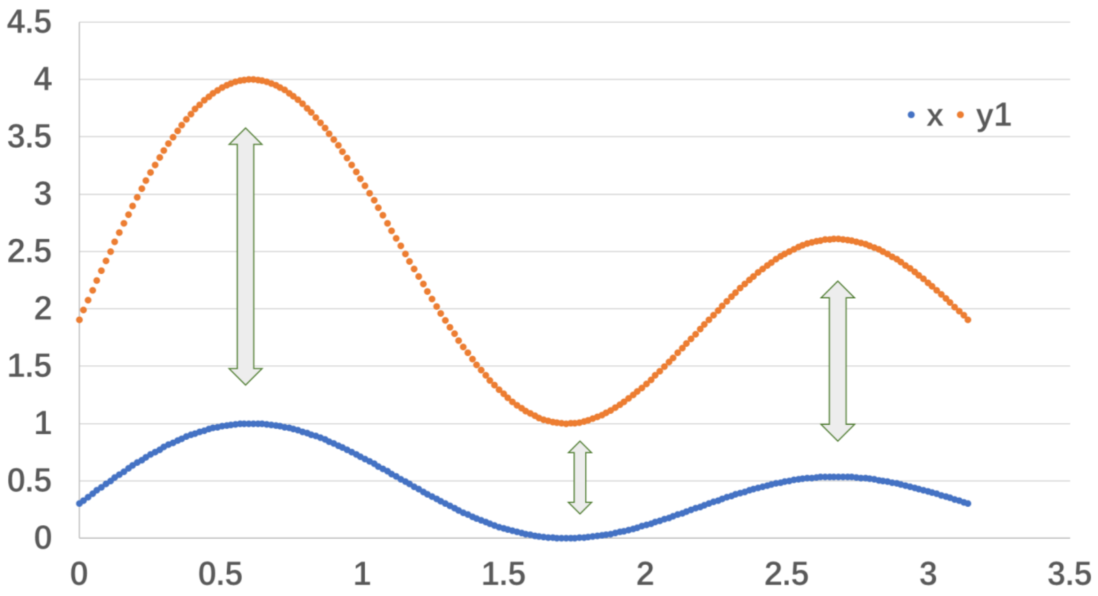


USING APPROXIMATE OPERATION INSTEAD OF MULTIPLICATION?



THE SIMILARITY BETWEEN TWO SIGNALS h AND g

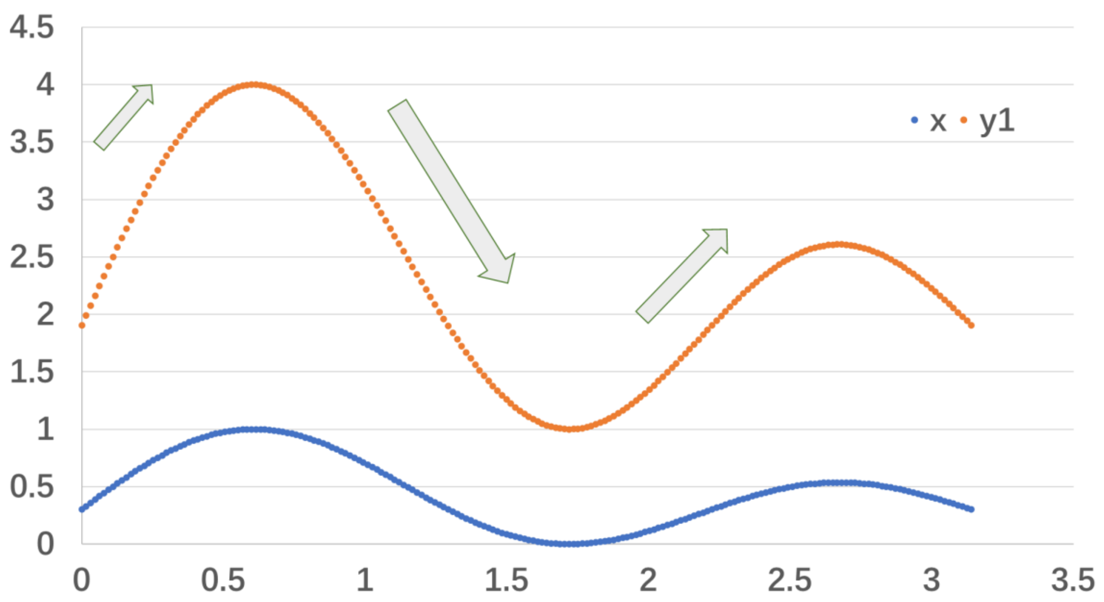




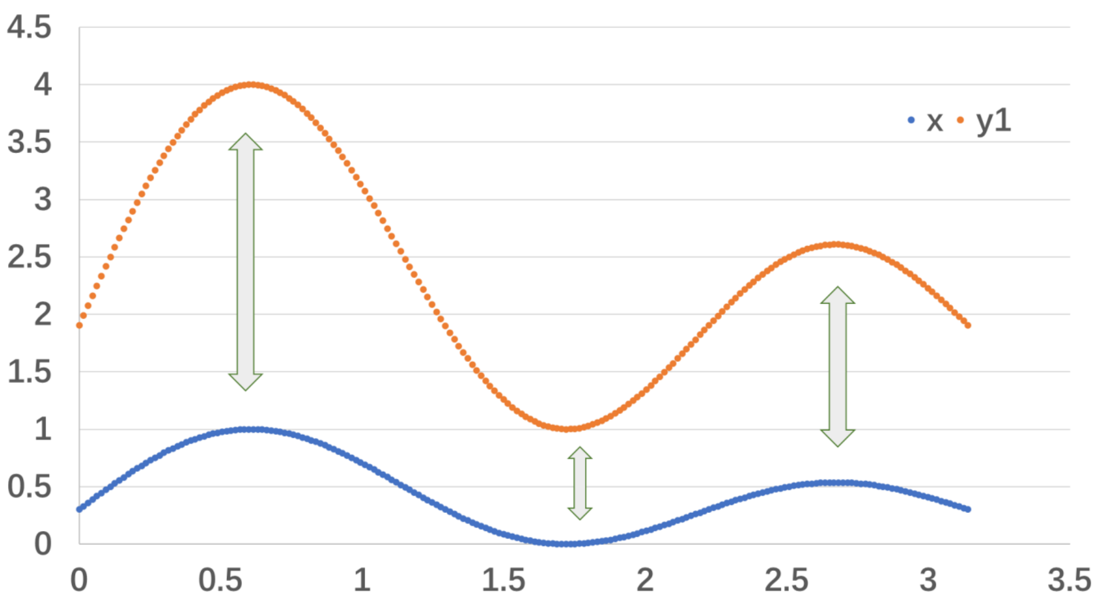
- $\rho(h, g) = \frac{cov(h,g)}{\sqrt{var(h)var(g)}}$
- Similarity of the trends of changes.

- $L(h, g) = \sum \left| \frac{g-h}{h} \right|$
- Distance between signals.

THE SIMILARITY BETWEEN TWO SIGNALS h AND g



- $\rho(h,g) = \frac{cov(h,g)}{\sqrt{var(h)var(g)}}$
- Similarity of the trends of changes.



- $L(h,g) = \sum \left| \frac{g-h}{h} \right|$
- Distance between signals.

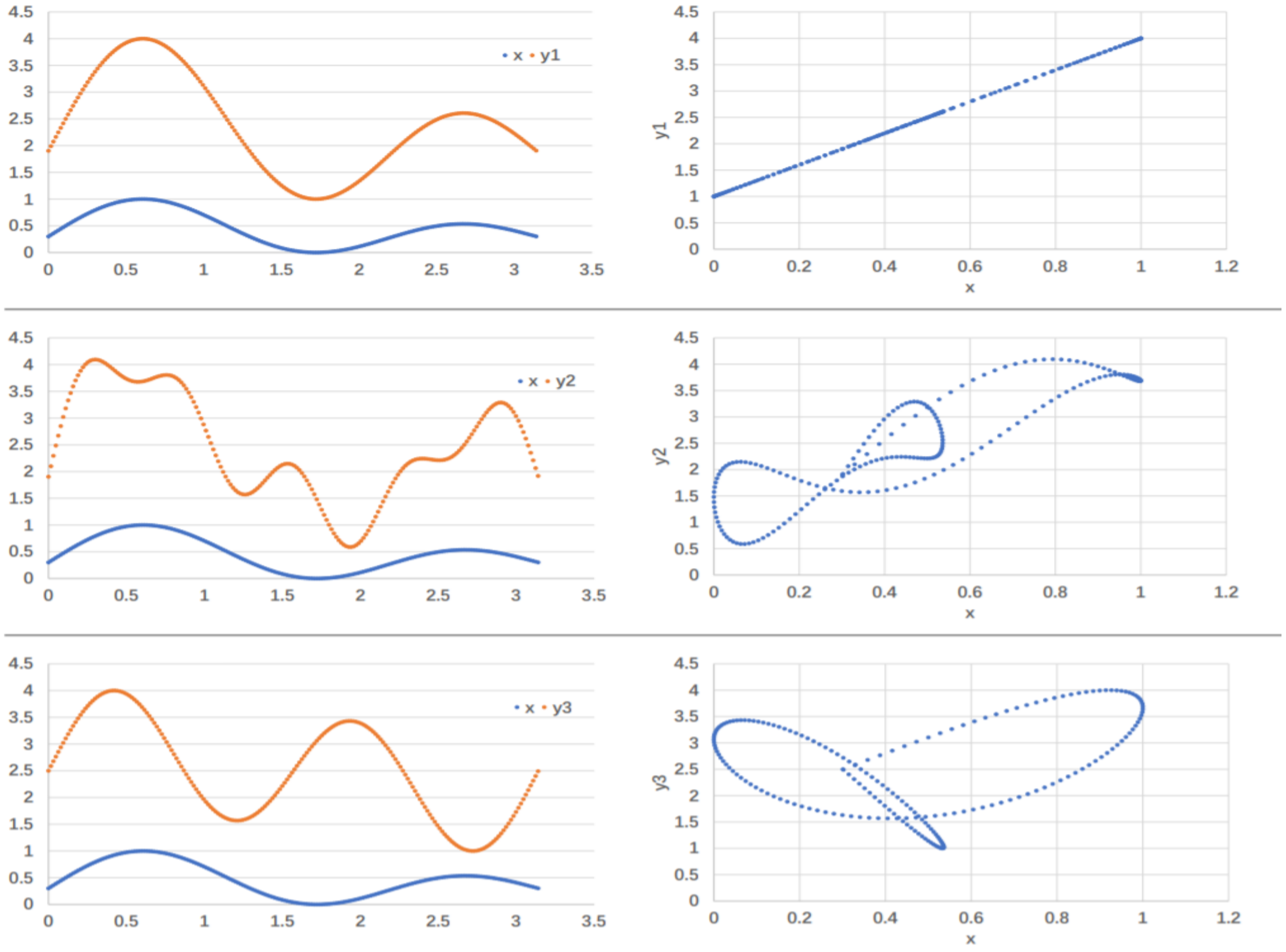
Pearson product-moment correlation coefficient (PPMCC)

$$\rho(h,g) = \frac{cov(h,g)}{\sqrt{var(h)var(g)}}$$

where:

$$\begin{cases} var(h) = \sum_n (h[n] - \mu_h)(h[n] - \mu_h) \\ cov(h,g) = \sum_n (h[n] - \mu_h)(g[n] - \mu_g) \end{cases}$$

Pearson product-moment correlation coefficient (PPMCC)

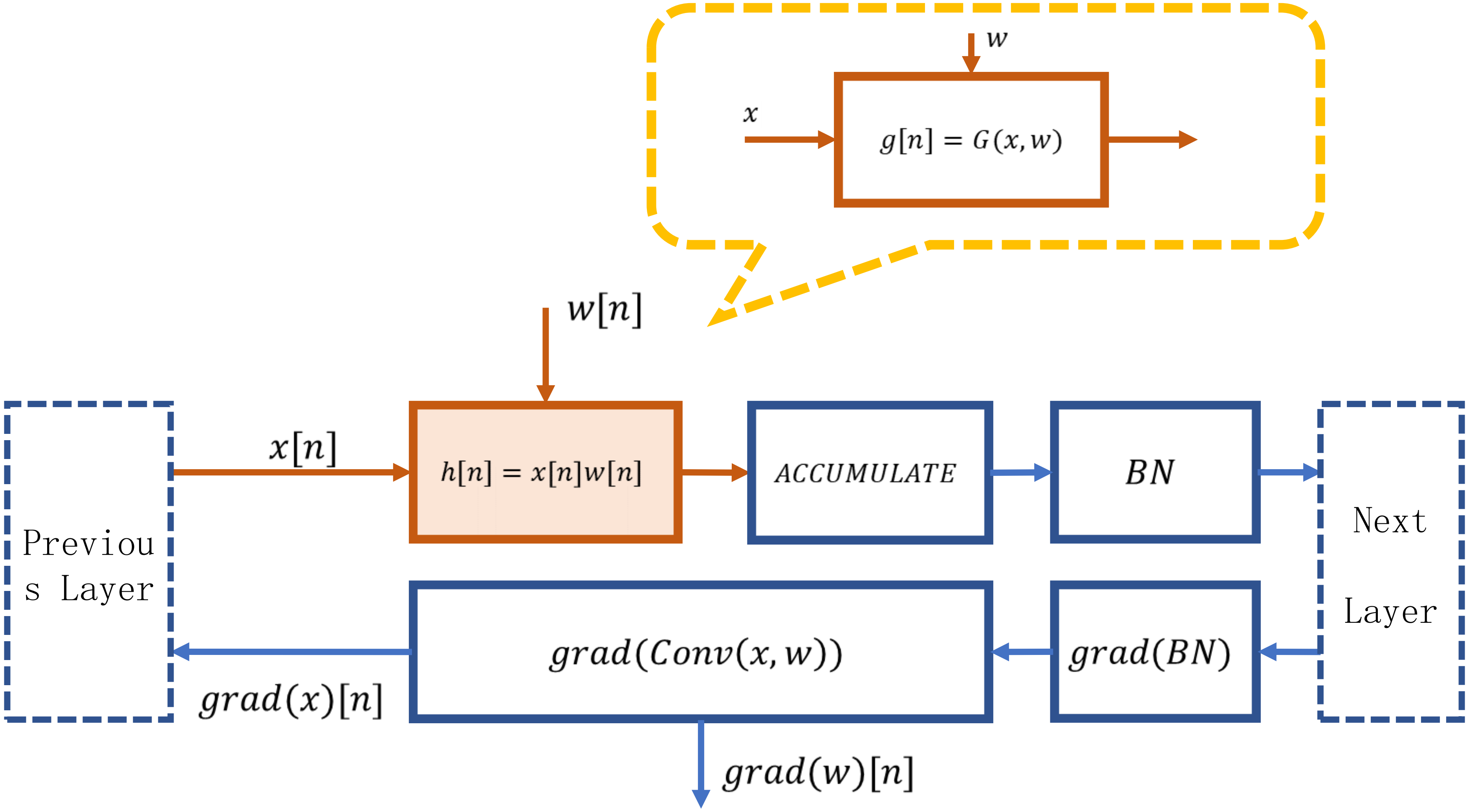


$\rho(x, y_1) = 1$

$\rho(x, y_2) = 0.869$

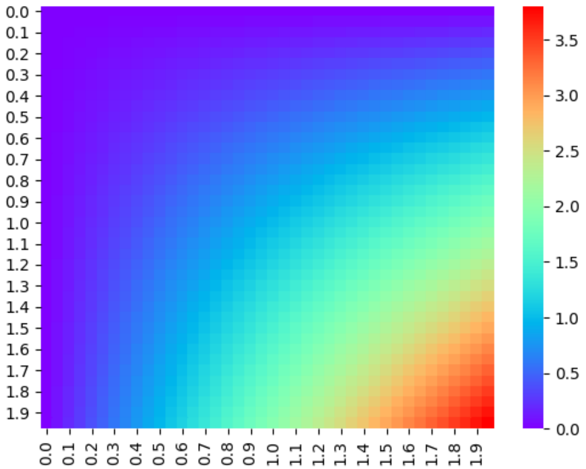
$\rho(x, y_3) = 0.193$

$$\boxed{\rho(h,g) = \frac{cov(h,g)}{\sqrt{var(h)var(g)}}}$$

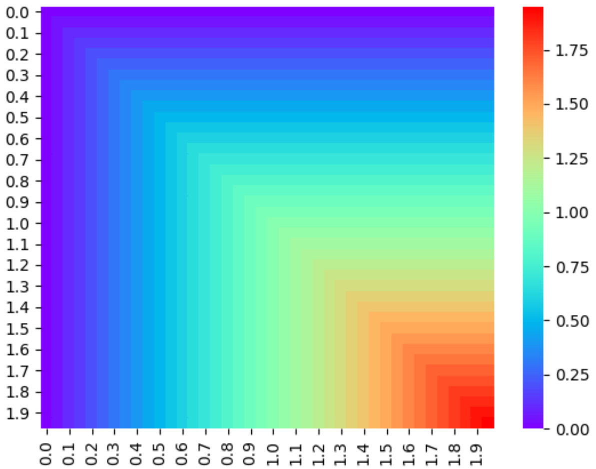

w
x
$g[n] = G(x, w)$
$w[n]$
$x[n]$
$h[n] = x[n]w[n]$
ACCUMULATE
BN
Previou
s Layer
Next
Layer
$grad(Conv(x, w))$
$grad(BN)$
$grad(x)[n]$
$grad(w)[n]$

Correlation with $h = \xi \cdot \eta$	Min-selector $g = min(\xi, \eta)$	Addition $g = \xi + \eta$	Max-selector $g = max(\xi, \eta)$
$\begin{cases} \xi \sim N_f(0,1) \\ \eta \sim N_f(0,1) \end{cases}$	0.908	0.882	0.673
$\begin{cases} \xi \sim N_f(0,1) \\ \eta \sim N_f(0,10) \end{cases}$	0.692	0.683	0.624
$\begin{cases} \xi \sim U(0,1) \\ \eta \sim U(0,1) \end{cases}$	0.962	0.926	0.641
$\begin{cases} \xi \sim U(0,1) \\ \eta \sim U(0,1) \end{cases}$	0.716	0.717	0.655

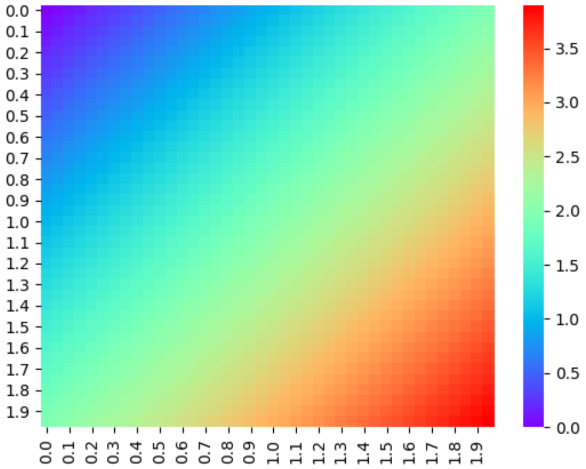
- ξ and η are non-negative value.
- $N_f(\mu, \sigma^2)$: folded normal distribution with expected value μ , variance σ^2.
- $U(a, b)$: a uniform distribution in an interval $[a, b]$



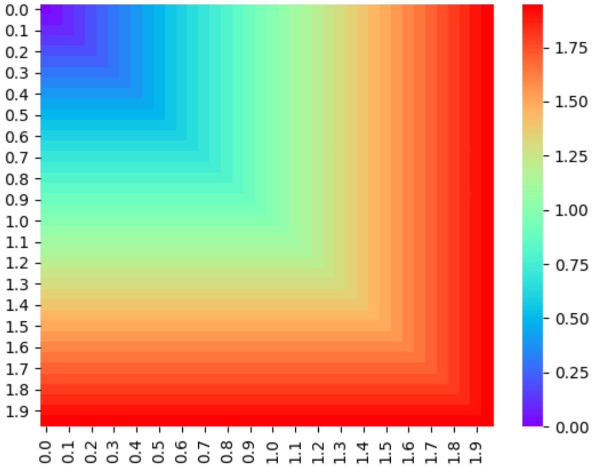
$y_1 = x_1 \cdot x_2$



$y_2 = min(x_1, x_2)$



$y_3 = x_1 + x_2$



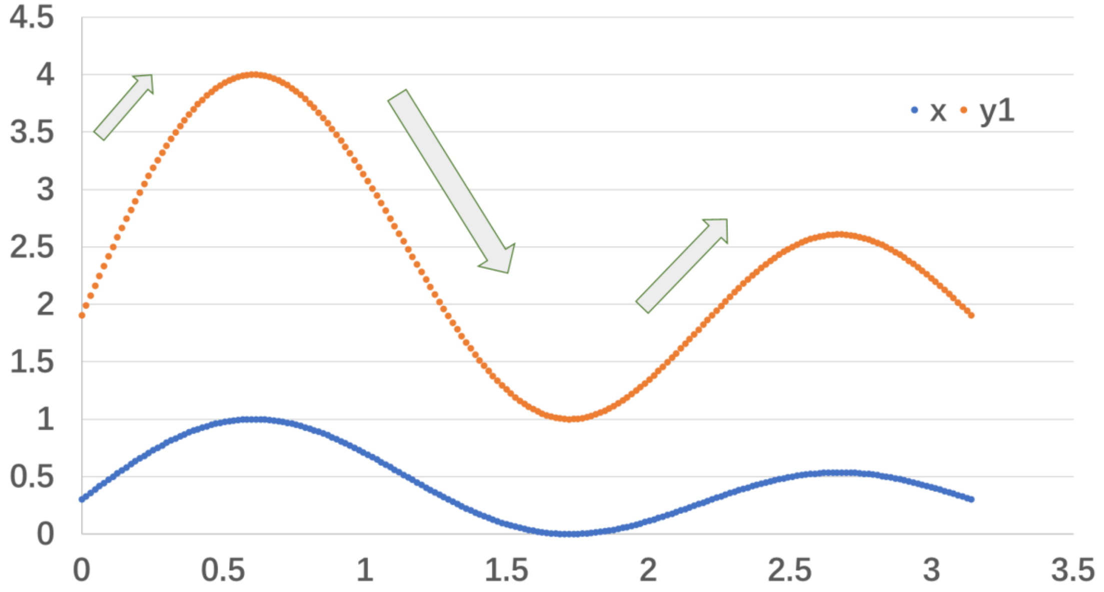
$y_4 = max(x_1, x_2)$

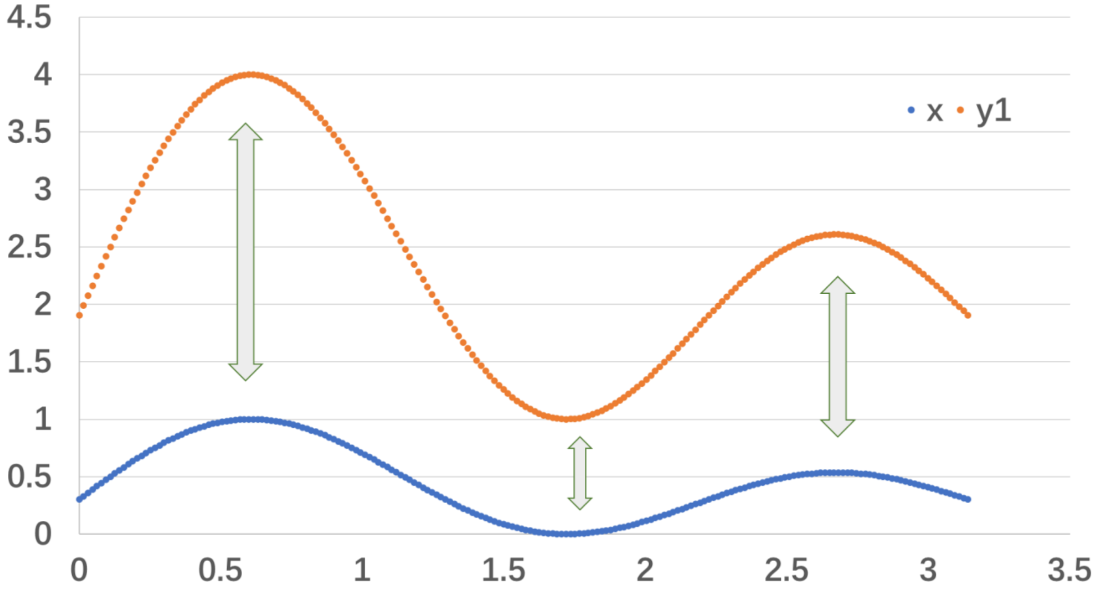
Correlation with $h = \xi \cdot \eta$	Min-selector $g = min(\xi,\eta)$	Addition $g = \xi + \eta$	Max-selector $g = max(\xi,\eta)$
$\begin{cases} \xi \sim N_f(0,1) \\ \eta \sim N_f(0,1) \end{cases}$	0.908	0.882	0.673
$\begin{cases} \xi \sim N_f(0,1) \\ \eta \sim N_f(0,10) \end{cases}$	0.692	0.683	0.624
$\begin{cases} \xi \sim U(0,1) \\ \eta \sim U(0,1) \end{cases}$	0.962	0.926	0.641
$\begin{cases} \xi \sim U(0,1) \\ \eta \sim U(0,1) \end{cases}$	0.716	0.717	0.655

$h = \xi \cdot \eta$ and $g = \min(\xi,\eta)$ have the similar trends of changes, if:

- ξ and η follow similar distribution:
  - They have the same expected values, noted as $\mu_{|\xi|} = \mu_{|\eta|}$
  - They are distributed in similar intervals, noted as $\sigma_\xi \sim \sigma_\eta$

THE SIMILARITY BETWEEN TWO SIGNALS h AND g

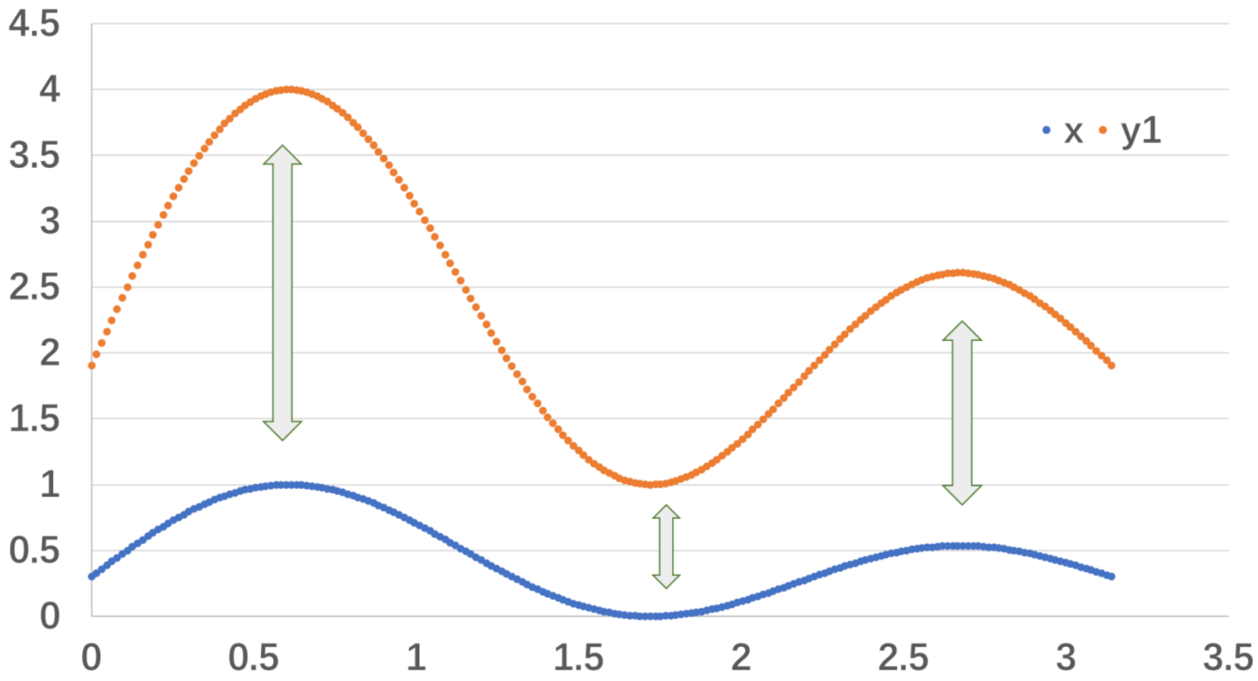




- $\rho(h,g) = \frac{cov(h,g)}{\sqrt{var(h)var(g)}}$
- Similarity of the trends of changes.

- $L(h,g) = \sum \left| \frac{g-h}{h} \right|$
- Distance between signals.

THE DISTANCE BETWEEN TWO SIGNALS h AND g



- $L(h,g) = \sum \left| \frac{g-h}{h} \right|$
- Find the constraints to make L as small as possible.

THE DISTANCE BETWEEN TWO SIGNALS h AND g

Let inputs ξ and η random variables with probability distribution $p_x(\xi)$ and $p_w(\eta)$, and outputs g and h are calculated as:

$$\begin{cases} h = H(\xi,\eta) = \xi \cdot \eta \\ g = G(\xi,\eta) = \min(\xi,\eta) \end{cases}$$

Then the distance between signals is calculated as:

$$L(h,g) = \int_\xi \int_\eta \left| \frac{H(\xi,\eta) - G(\xi,\eta)}{H(\xi,\eta)} \right| \cdot p_x(\xi) p_w(\eta) d\xi d\eta$$

Institut Mines-Télécom

THE DISTANCE BETWEEN TWO SIGNALS h AND g

Let inputs ξ and η random variables with probability distribution $p_x(\xi)$ and $p_w(\eta)$, and outputs g and h are calculated as:

$$\begin{cases} h = H(\xi, \eta) = \xi \cdot \eta \\ g = G(\xi, \eta) = \min(\xi, \eta) \end{cases}$$

Then the distance between signals is calculated as:

$$L(h, g) = \int_\xi \int_\eta \left| \frac{H(\xi, \eta) - G(\xi, \eta)}{H(\xi, \eta)} \right| \cdot p_x(\xi) p_w(\eta) d\xi d\eta$$

THE DISTANCE BETWEEN TWO SIGNALS h AND g

Let inputs ξ and η random variables with probability distribution $p_x(\xi)$ and $p_w(\eta)$, and outputs g and h are calculated as:

$$\begin{cases} h = H(\xi,\eta) = \xi \cdot \eta \\ g = G(\xi,\eta) = \min(\xi,\eta) \end{cases}$$

Then the distance between signals is calculated as:

$$L(h,g) = \int_\xi \int_\eta \left| \frac{H(\xi,\eta) - G(\xi,\eta)}{H(\xi,\eta)} \right| \cdot p_x(\xi) p_w(\eta) d\xi d\eta$$

$$= f_1\left(p_x(\xi), p_w(\eta)\right)$$

THE DISTANCE BETWEEN TWO SIGNALS h AND g

Let inputs ξ and η random variables with probability distribution $p_x(\xi)$ and $p_w(\eta)$, and outputs g and h are calculated as:

$$\begin{cases} h = H(\xi,\eta) = \xi \cdot \eta \\ g = G(\xi,\eta) = \min(\xi,\eta) \end{cases}$$

Then the distance between signals is calculated as:

$$L(h,g) = f_1\left(p_x(\xi), p_w(\eta)\right)$$

THE DISTANCE BETWEEN TWO SIGNALS h AND g

Let inputs ξ and η random variables with probability distribution $p_x(\xi)$ and $p_w(\eta)$, and outputs g and h are calculated as:

$$\begin{cases} h = H(\xi,\eta) = \xi \cdot \eta \\ g = G(\xi,\eta) = \min(\xi,\eta) \end{cases}$$

Then the distance between signals is calculated as:

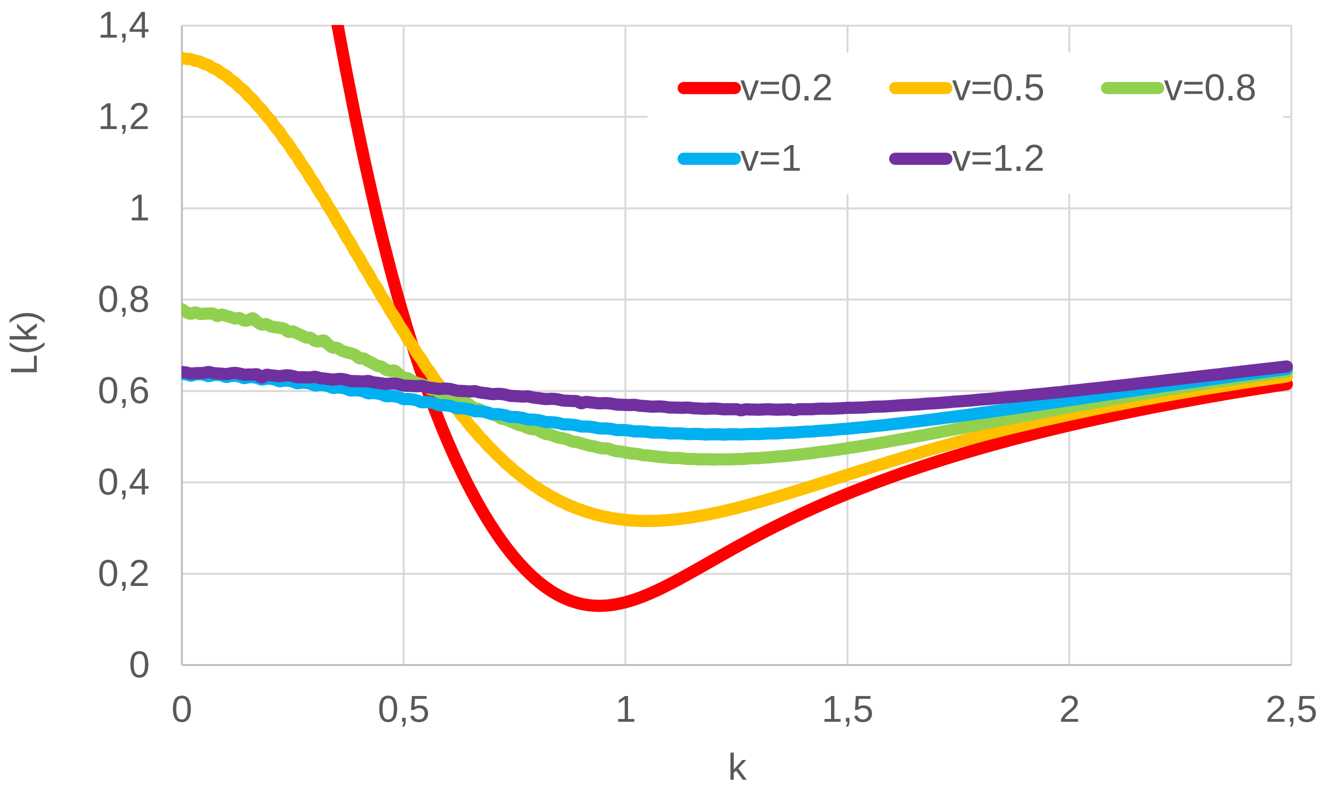
$$L(h,g) = f_1\left(p_x(\xi), p_w(\eta)\right)$$

If ξ and $\eta \sim N_f(k,v)$:

$$L(h,g) = f_2(k,v)$$

where k represents the expected values of ξ and η, and v represents the variance of ξ and η.

THE DISTANCE BETWEEN TWO SIGNALS h AND g



To make $L(k, v)$ as small as possible:

- **C1**: k that minimizes L is around 1, noted as $\mu_{|\xi|} = \mu_{|\eta|} = 1$.
- **C2**: v should be as small as possible.

3. Building MinConvNets with approximate operation

Let matrix multiplication arbitrary:

$$|z| = |x| \cdot |w|$$

be transformed as:

$$\frac{|z|}{\mu_{|x|}\mu_{|w|}} = \frac{|x|}{\mu_{|x|}} \cdot \frac{|w|}{\mu_{|w|}}$$

That meets constraint $\mu_{|\xi|} = \mu_{|\eta|} = 1$, therefore:

$$\frac{|z|}{\mu_{|x|}\mu_{|w|}} \approx \min(\frac{|x|}{\mu_{|x|}}, \frac{|w|}{\mu_{|w|}})$$

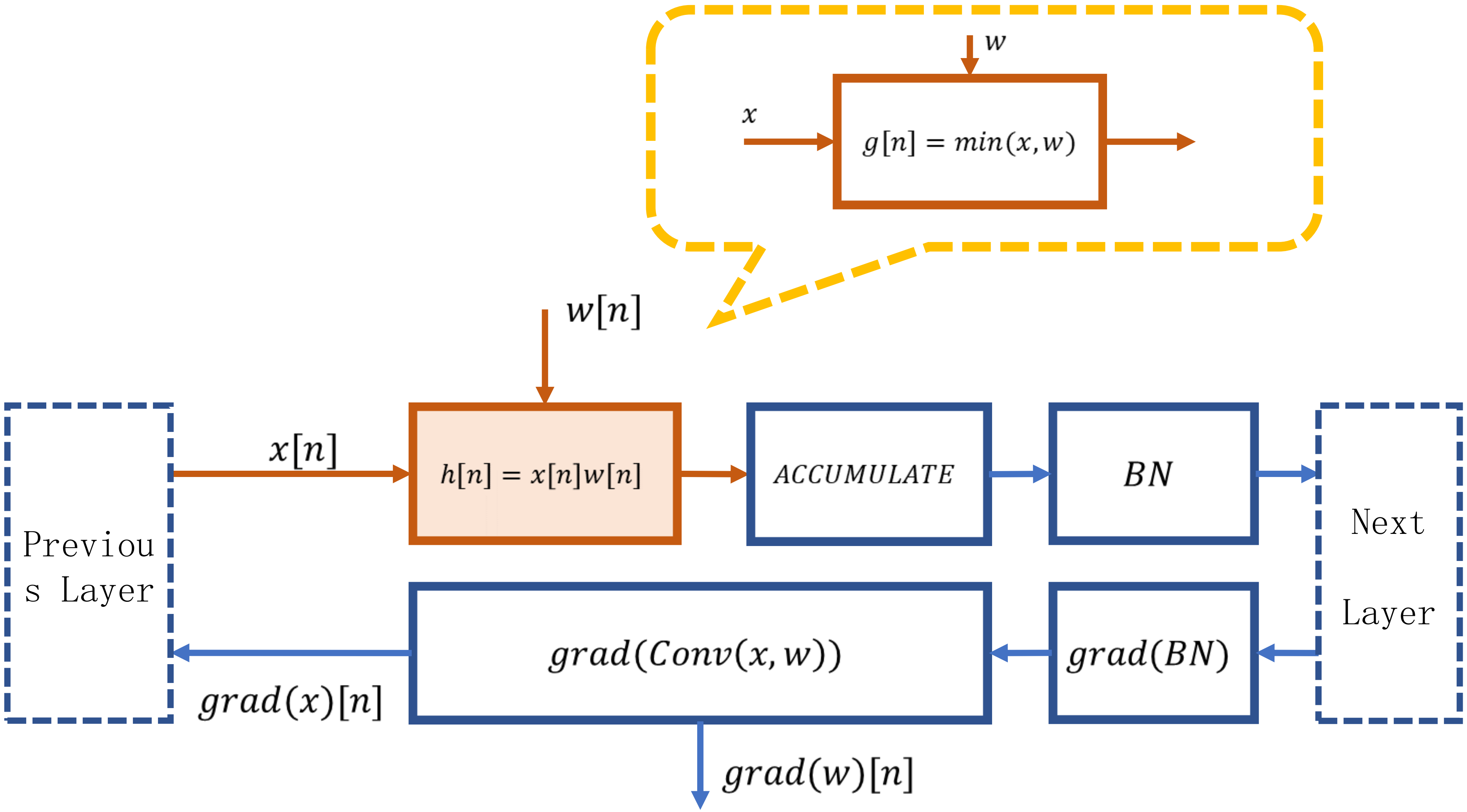
So:

$$|z| = \mu_{|w|} \cdot \min(|x|, \frac{\mu_{|x|}}{\mu_{|w|}} \cdot |w|)$$

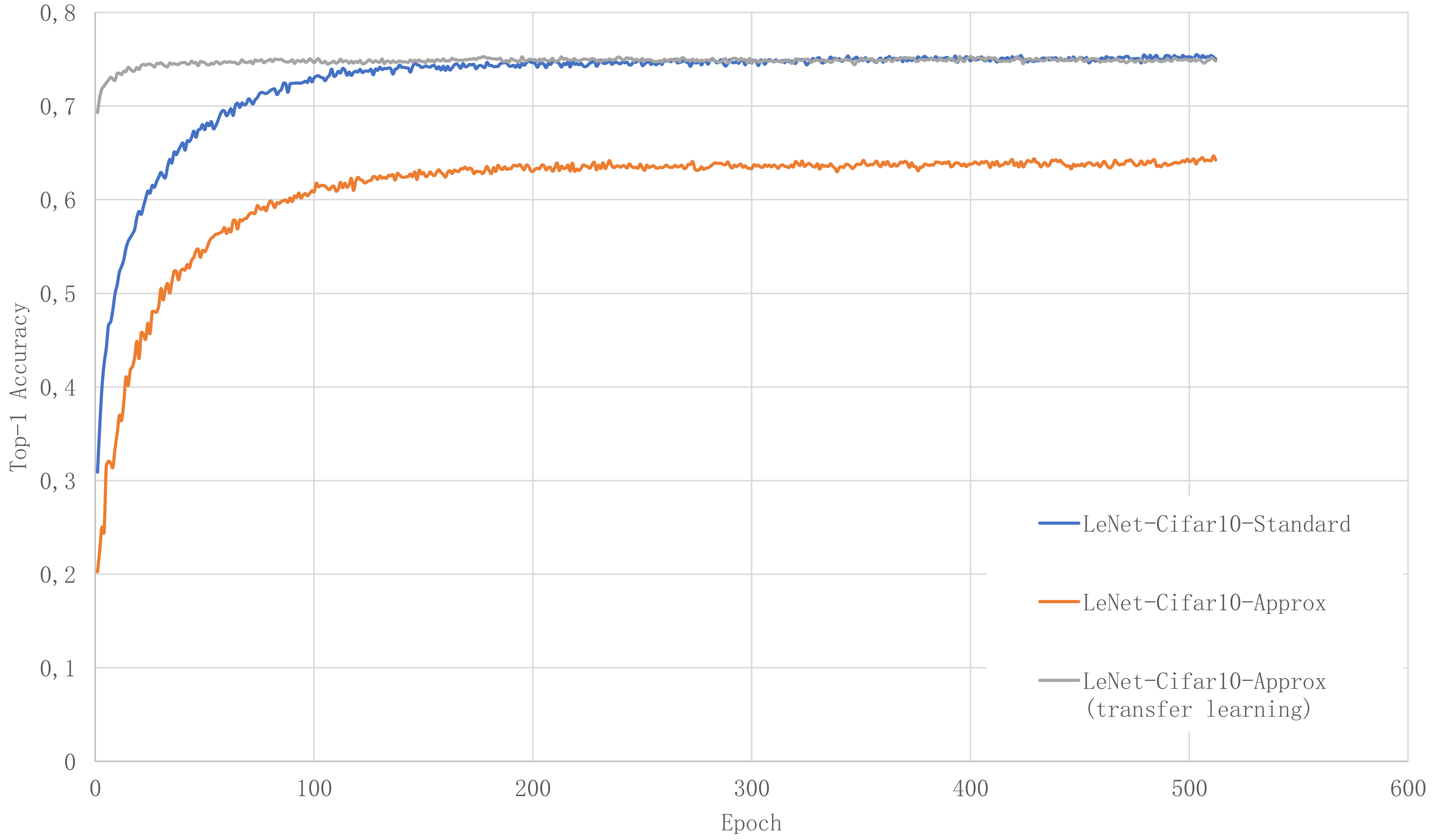
Remove excessively large values:

$$clip(w, \alpha) = \begin{cases} \alpha & if\ w > \alpha \\ -\alpha & if\ w < -\alpha \\ w & otherwise \end{cases}$$

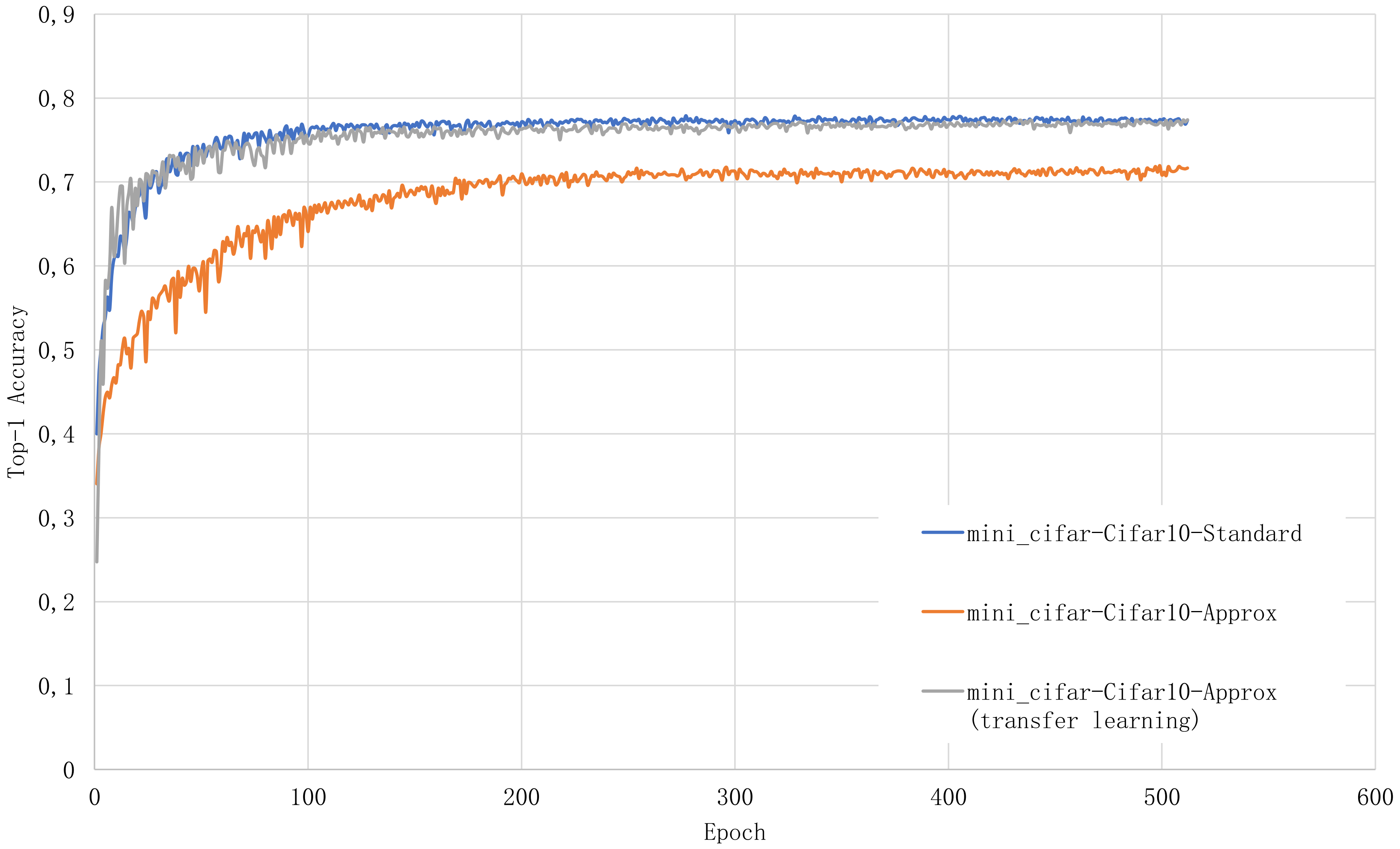
- In these works, $\alpha = 2\mu_{|w|}$ shared by each filter.
- Weights and inputs are both clipped during training.
- Only weights are pre-clipped for inferring.


w
x
$g[n] = min(x,w)$
$w[n]$
$x[n]$
$h[n] = x[n]w[n]$
$ACCUMULATE$
BN
Previou
s Layer
Next
Layer
$grad(Conv(x,w))$
$grad(BN)$
$grad(x)[n]$
$grad(w)[n]$

Top-1 accuracy of LeNet applied to Cifar10



Top-1 accuracy of mini-Cifar applied to Cifar10



4.Conclusion

Institut Mines-Télécom

Architecture		LeNet-MNIST	LeNet-Cifar10	Mini_cifar-Cifar10
Standard Network		99.06%	75.26%	77.30%
Approximate	170 epoch	98.42%		
	512 epoch		64.18%	71.46%
	2048 epoch		65.54%	72.89%
Transfer Learning	512 epoch		74.92%	77.01%
	1024 epoch		75.10%	77.26%

- Approximate Multiplication is proposed.
- MinConvNets are built by using Approximate Multiplication.
- Transfer Learning is used to optimize the training.