Dong-A Univ. (ISPL)

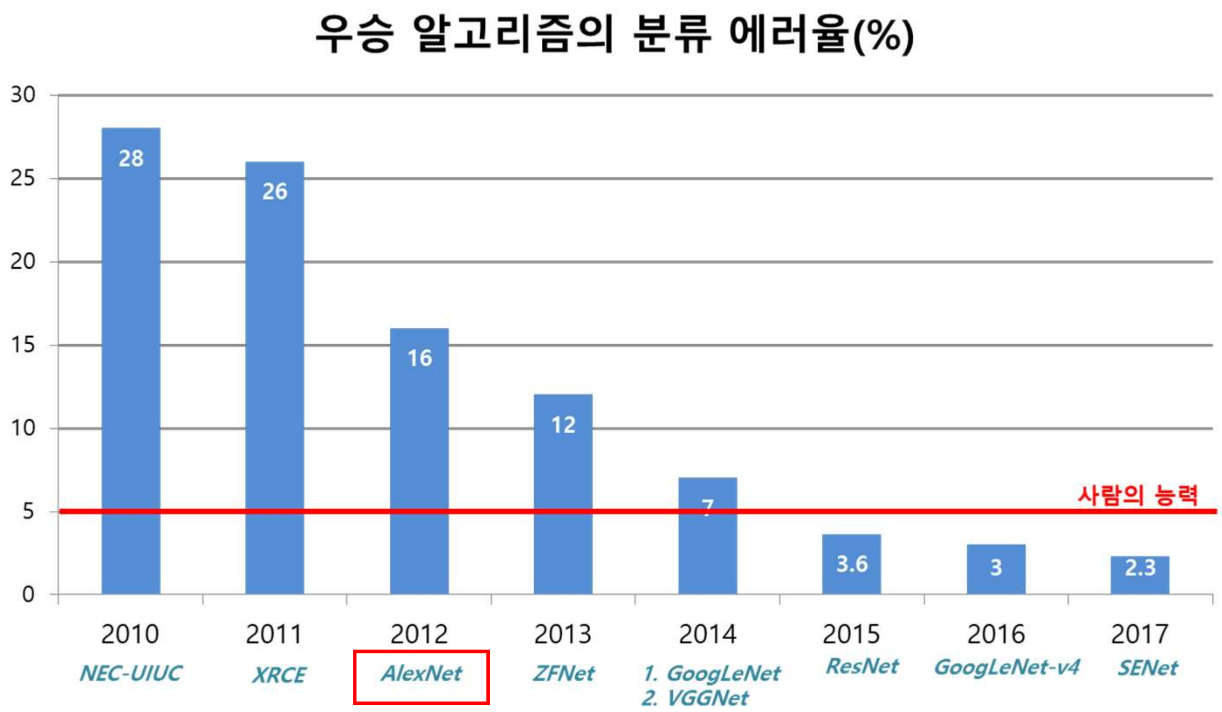
동아대학교
DONG-A UNIVERSITY

Popular CNN Models & CNN Backpropagation

컴퓨터AI공학부 AI학과
2024년 1학기 인공지능

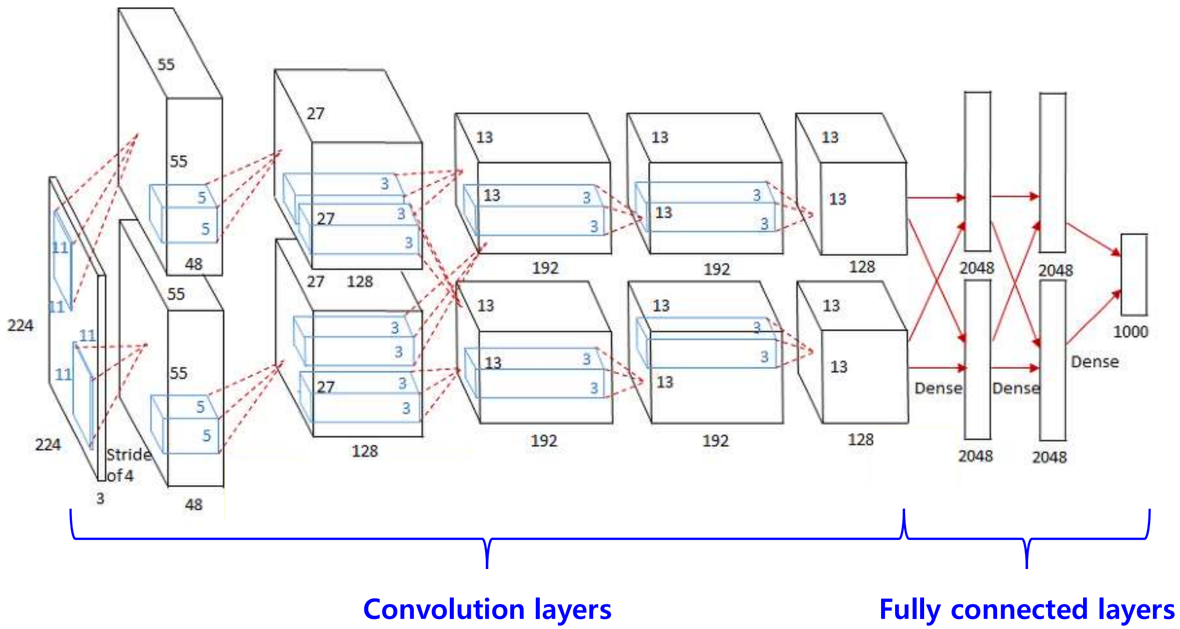
주요 CNN 구조 소개

- **ImageNet Large Scale Visual Recognition Challenge (ILSVRC)**
  - 대용량의 이미지셋 (1000개의 클래스) 에 대한 이미지 분류 알고리즘 성능 평가 대회



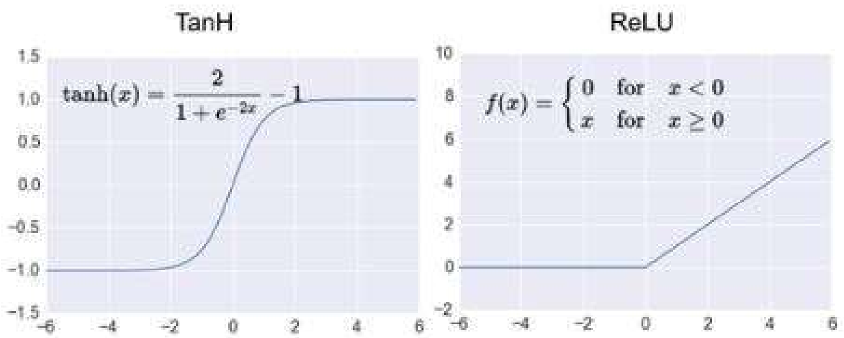
AlexNet(2012)

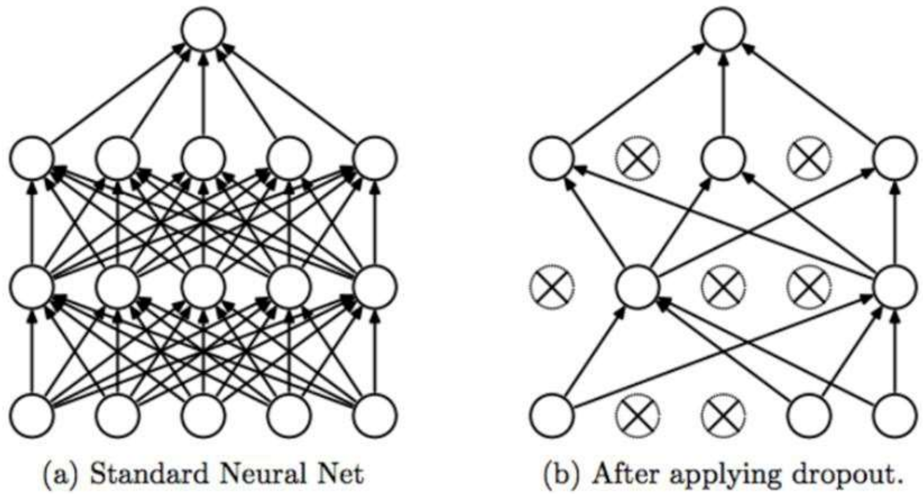
- 2개의 GPU로 병렬 연산을 수행하기 위해 병렬적인 구조로 설계
- 총 8개의 layer로 구성(5개의 컨볼루션 레이어, 3개의 Fully connected layer)



AlexNet(2012)

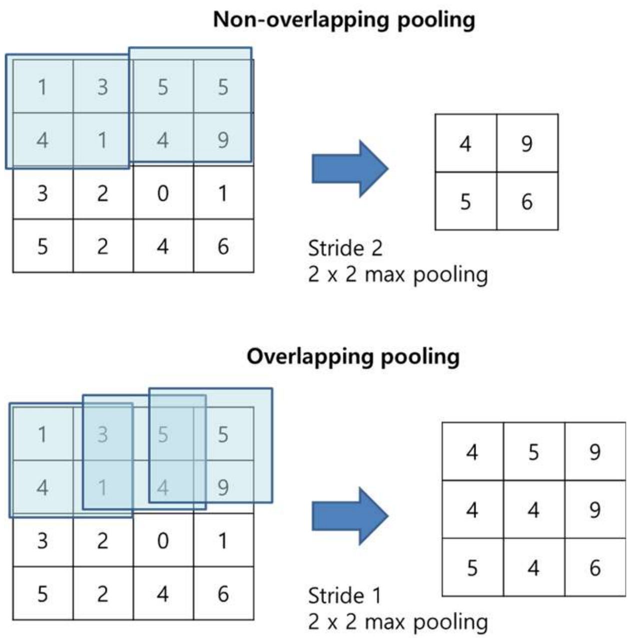
- **Activation function으로 ReLU함수를 사용(TanH함수를 사용할 때마다 6배 빠름)**
- **Over-fitting을 막기위해 Dropout을 사용**

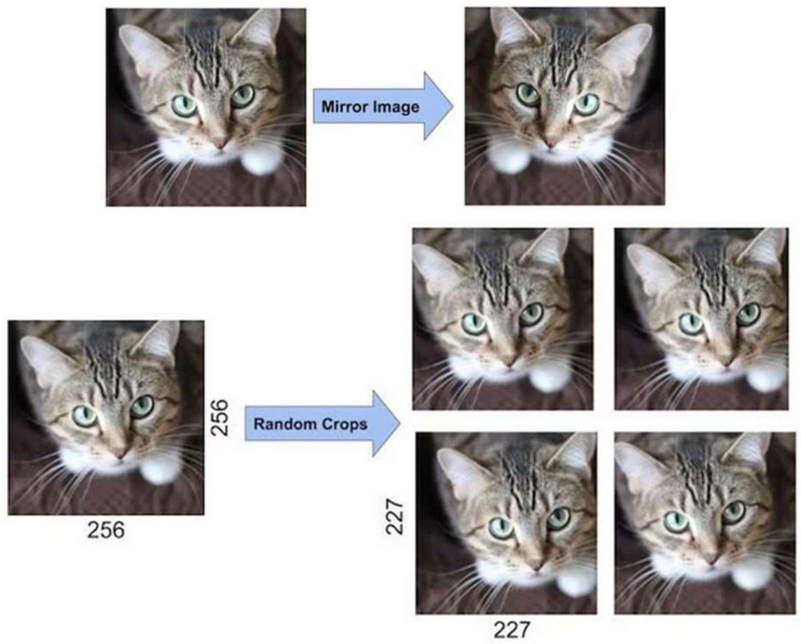




AlexNet(2012)

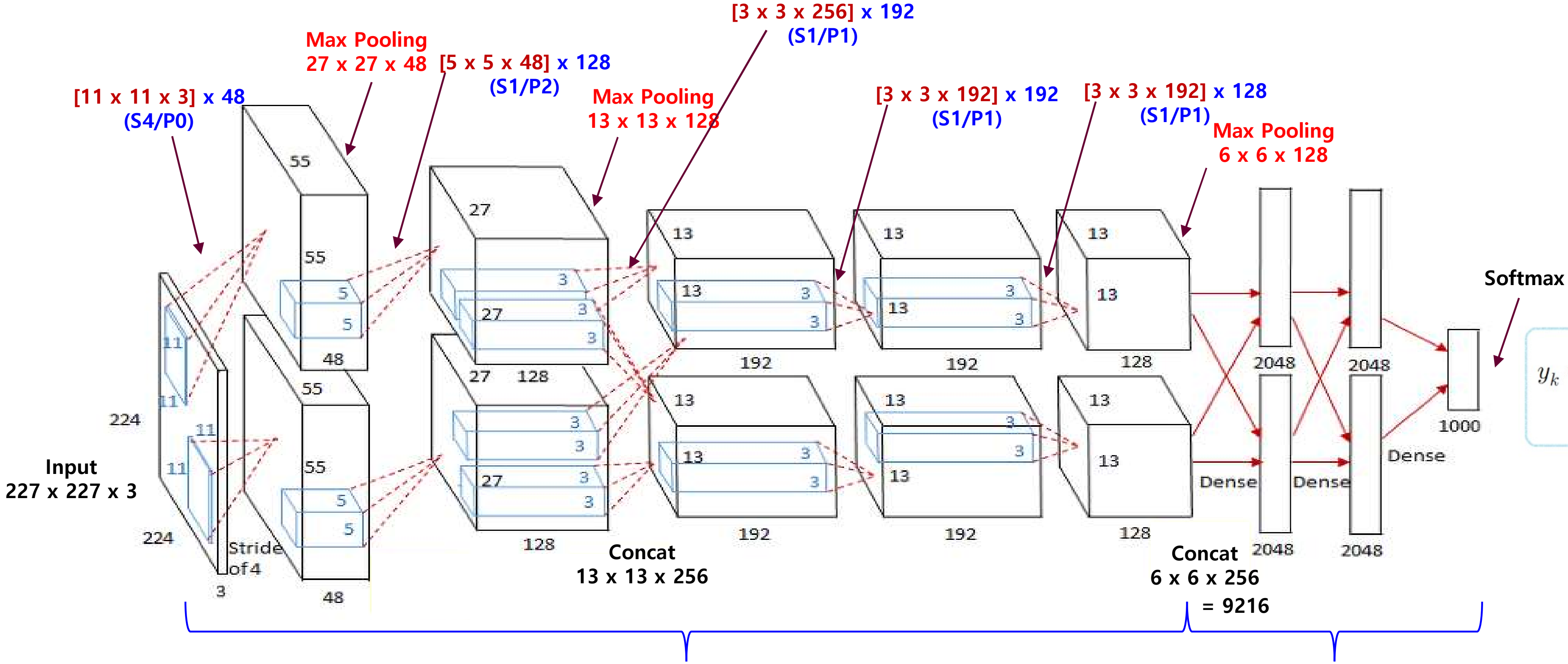
- **3x3 Overlapping pooling 사용 (Stride = 2)**
- **Data augmentation으로 데이터 양을 증가 ➔ overfitting 문제 줄임**





AlexNet(2012)

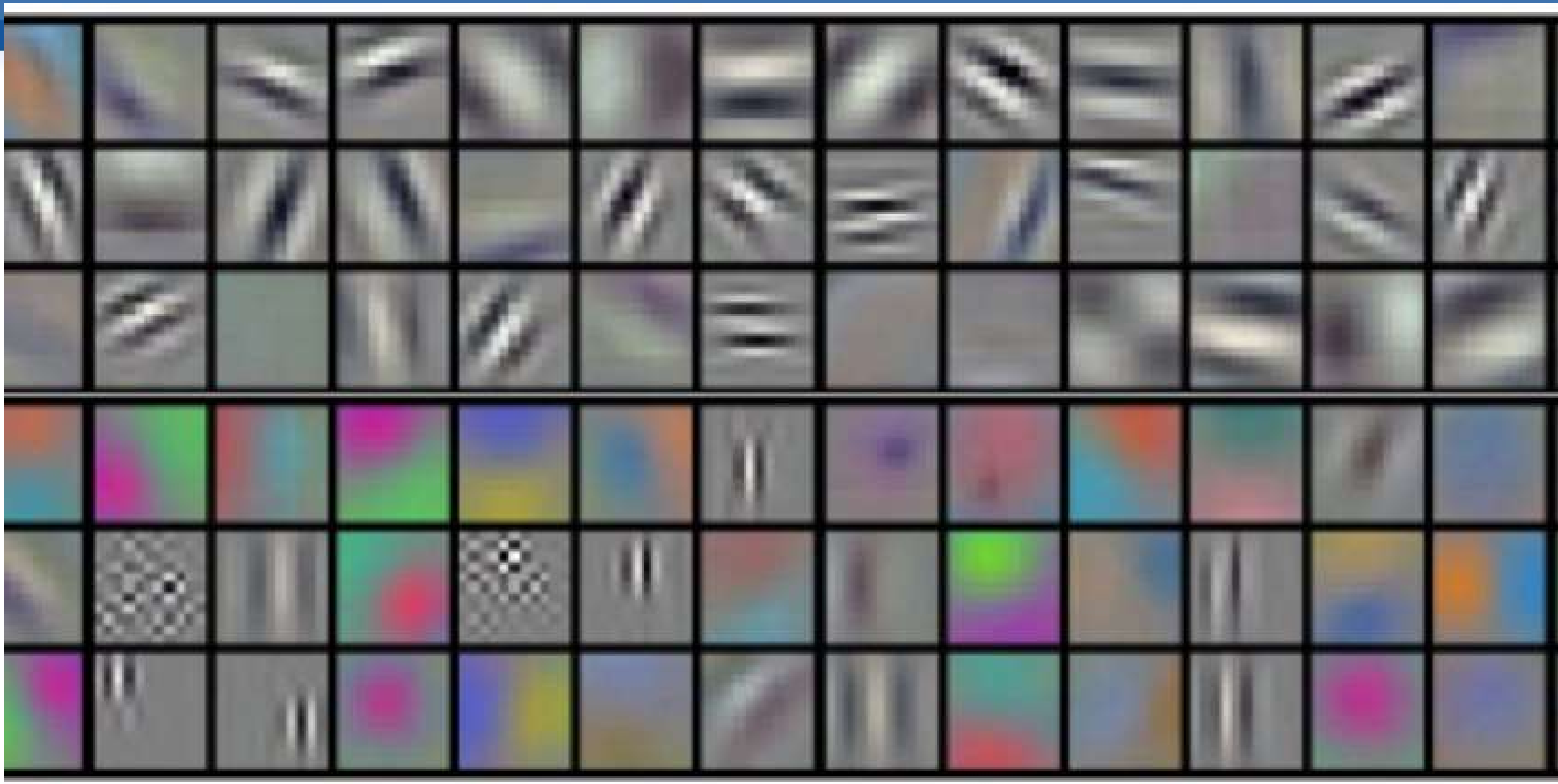
- **AlexNet (3x3 Max Pooling@S2, ReLU)**

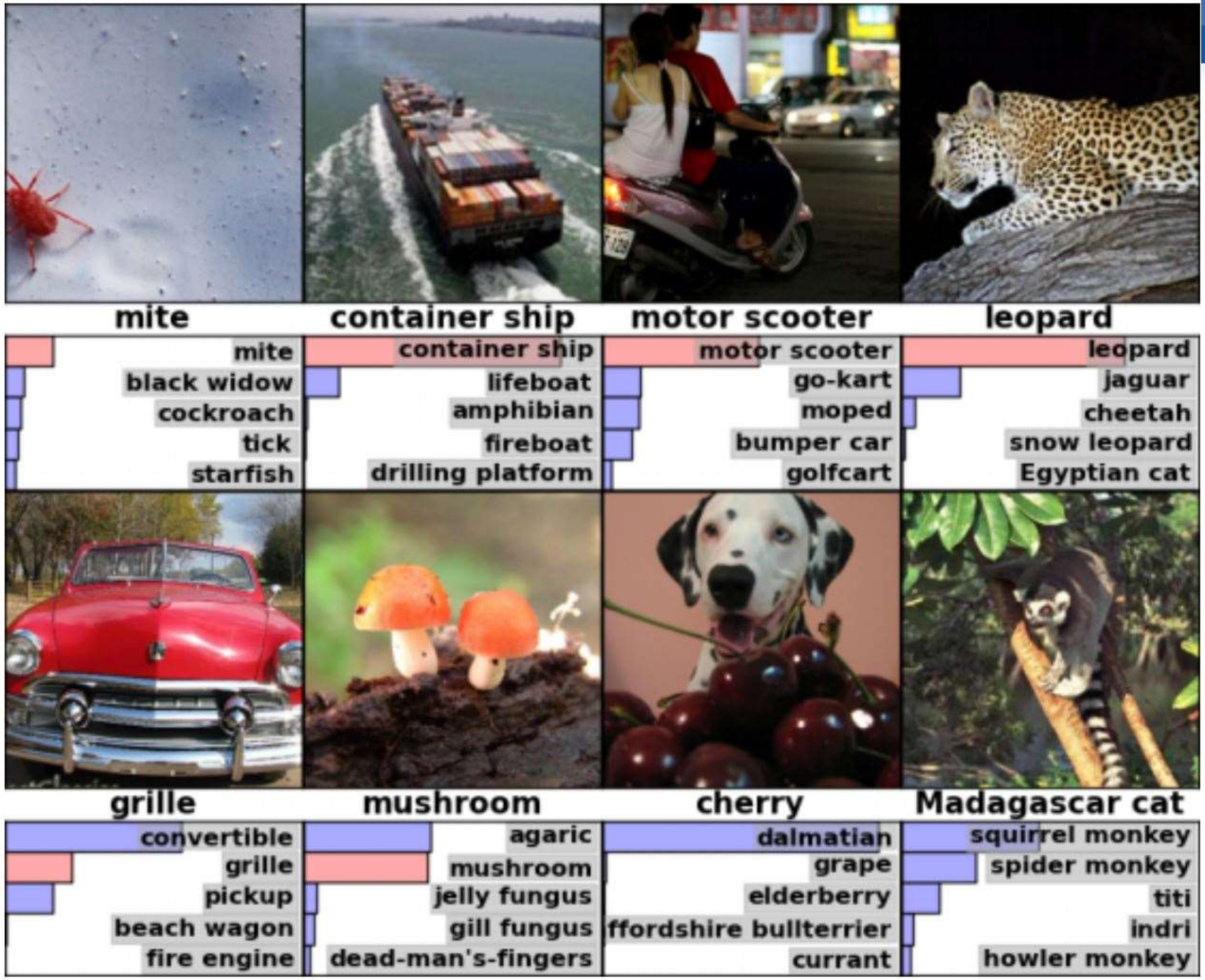


$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^{n} \exp(a_i)}$$

Ref) https://bskyvision.com/425

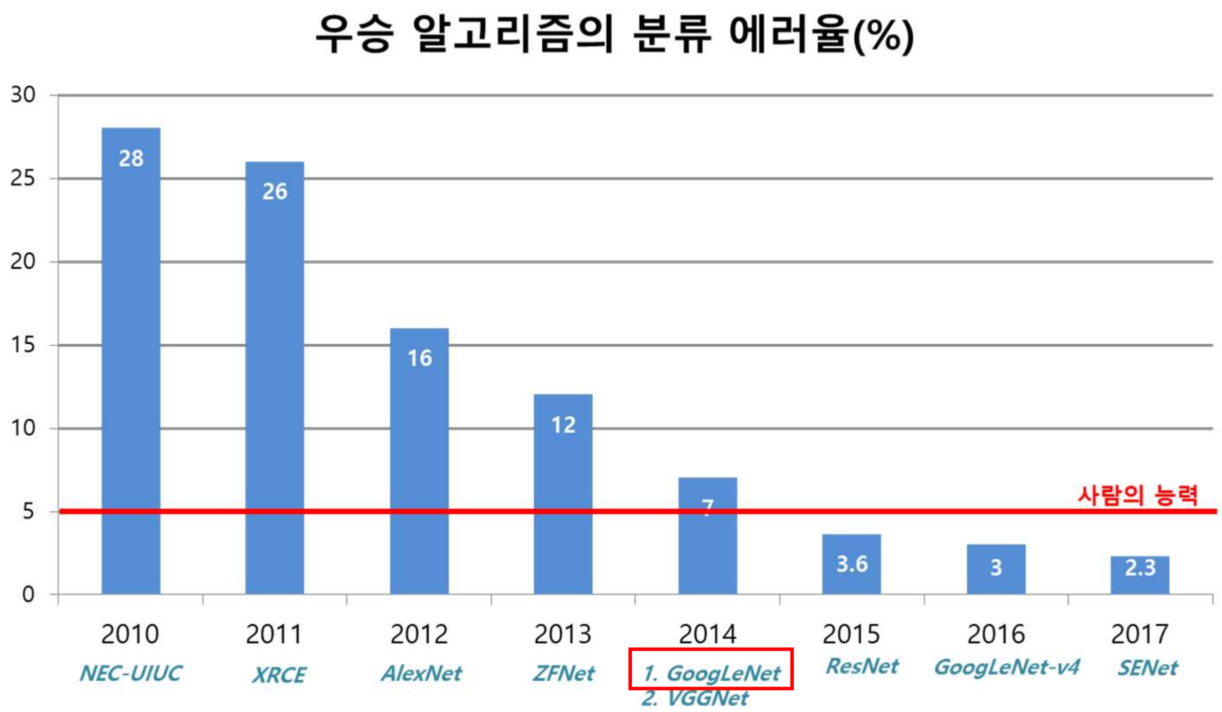
동아대학교 DONG-A UNIVERSITY




mite
container ship
motor scooter
leopard
mite
black widow
cockroach
tick
starfish
container ship
lifeboat
amphibian
fireboat
drilling platform
motor scooter
go-kart
moped
bumper car
golfcart
leopard
jaguar
cheetah
snow leopard
Egyptian cat
grille
mushroom
cherry
Madagascar cat
convertible
grille
pickup
beach wagon
fire engine
agaric
mushroom
jelly fungus
gill fungus
dead-man's-fingers
dalmatian
grape
elderberry
ffordshire bullterrier
currant
squirrel monkey
spider monkey
titi
indri
howler monkey

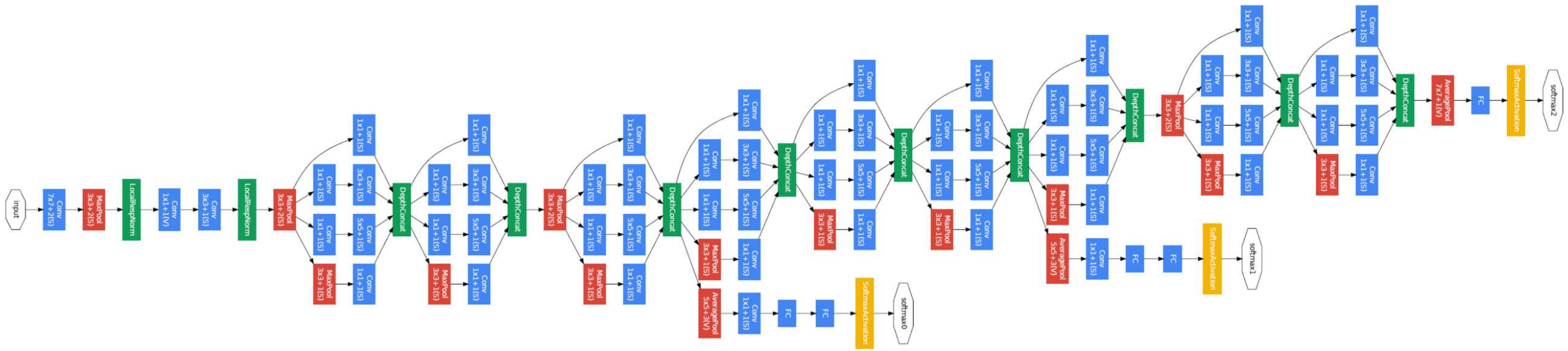
주요 CNN 구조 소개

- **ImageNet Large Scale Visual Recognition Challenge (ILSVRC)**
  - 대용량의 이미지셋 (1000개의 클래스) 에 대한 이미지 분류 알고리즘 성능 평가 대회



GooGleNet(2014)

- **VGGNet을 이기고 우승을 차지한 알고리즘 (Inception)**
- **총 22개 layer로 구성**



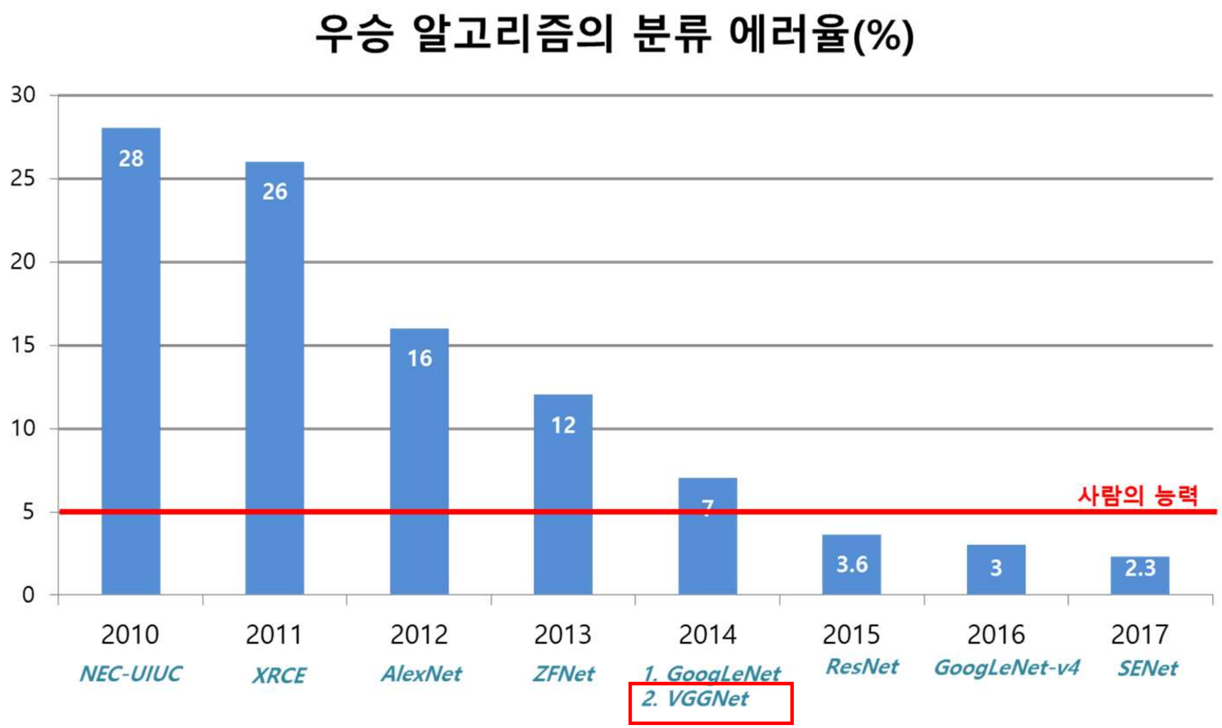
<GoogLeNet 구조>

Ref) https://bskyvision.com/539

동아대학교 DONG-A UNIVERSITY

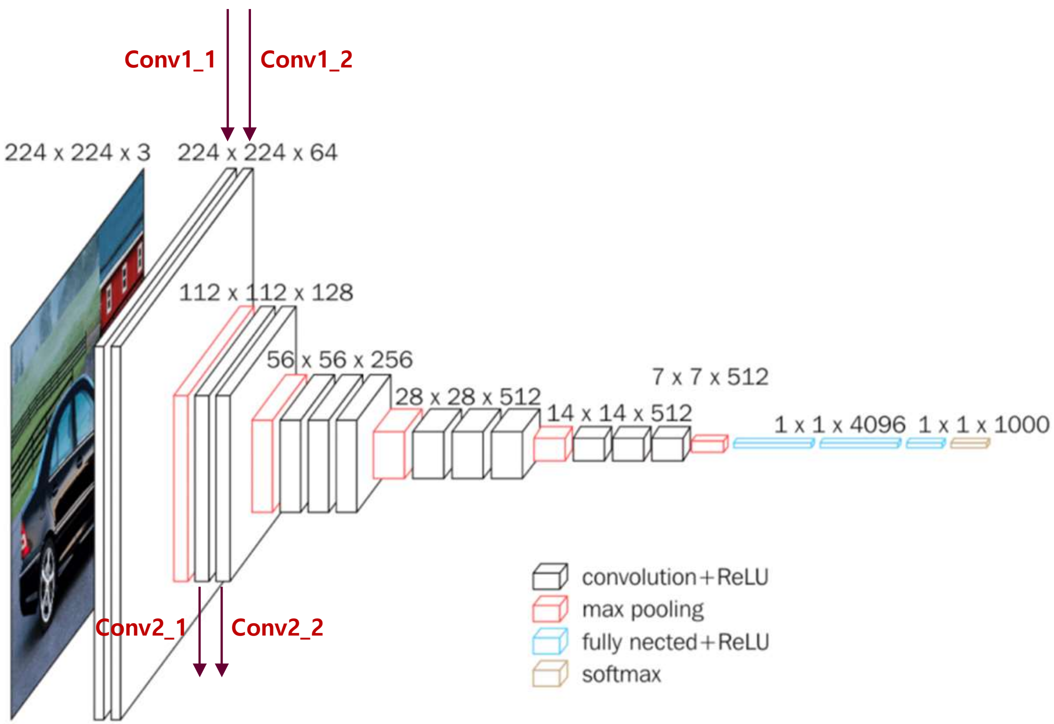
주요 CNN 구조 소개

- **ImageNet Large Scale Visual Recognition Challenge (ILSVRC)**
  - 대용량의 이미지셋 (1000개의 클래스) 에 대한 이미지 분류 알고리즘 성능 평가 대회



VGGNet(VGG-16 / VGG-19)

- **Weight parameter의 개수와 성능에 대한 trade-off를 탐색**
- **Network의 깊이가 깊어짐에 따라 높은 성능을 보임을 증명 (이후부터 네트워크 레이어를 증가시키는 추세가 활발히 이루어짐)**
- **필터크기는 3x3으로 고정 & ReLU 사용**



Ref) https://bskyvision.com/504

<VGG-16 구조>

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

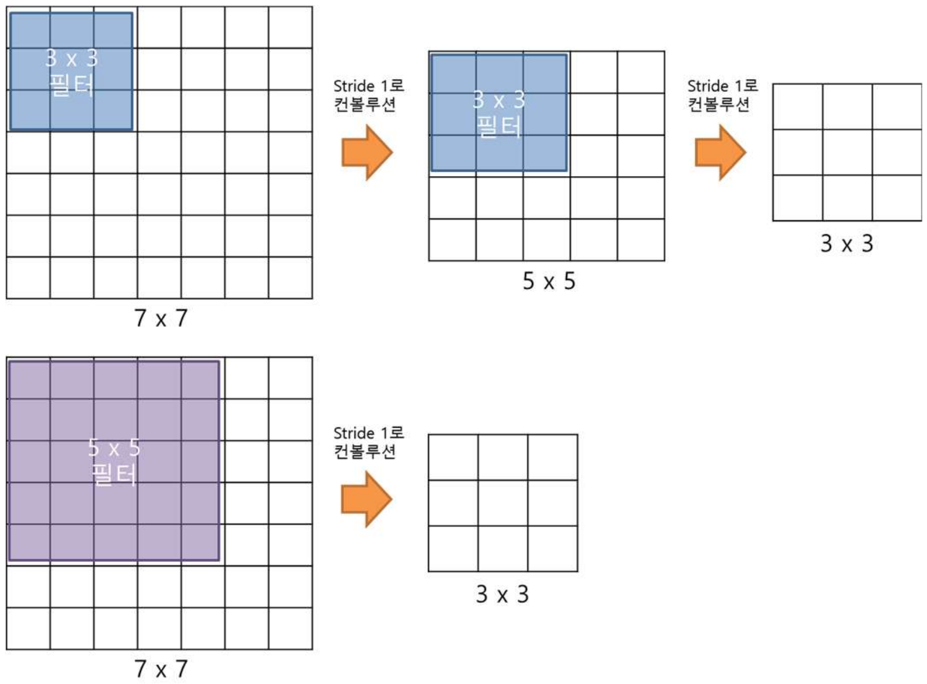
<VGGNet 실험 설정 예시>

VGGNet(VGG-16, 2014)

- 기존 높은 필터사용을 없애고 3x3필터로 통일

➔ Parameter 수를 줄임(3x3x2=18개, 5x5=25개) -> Light Memory

→ Fast Training

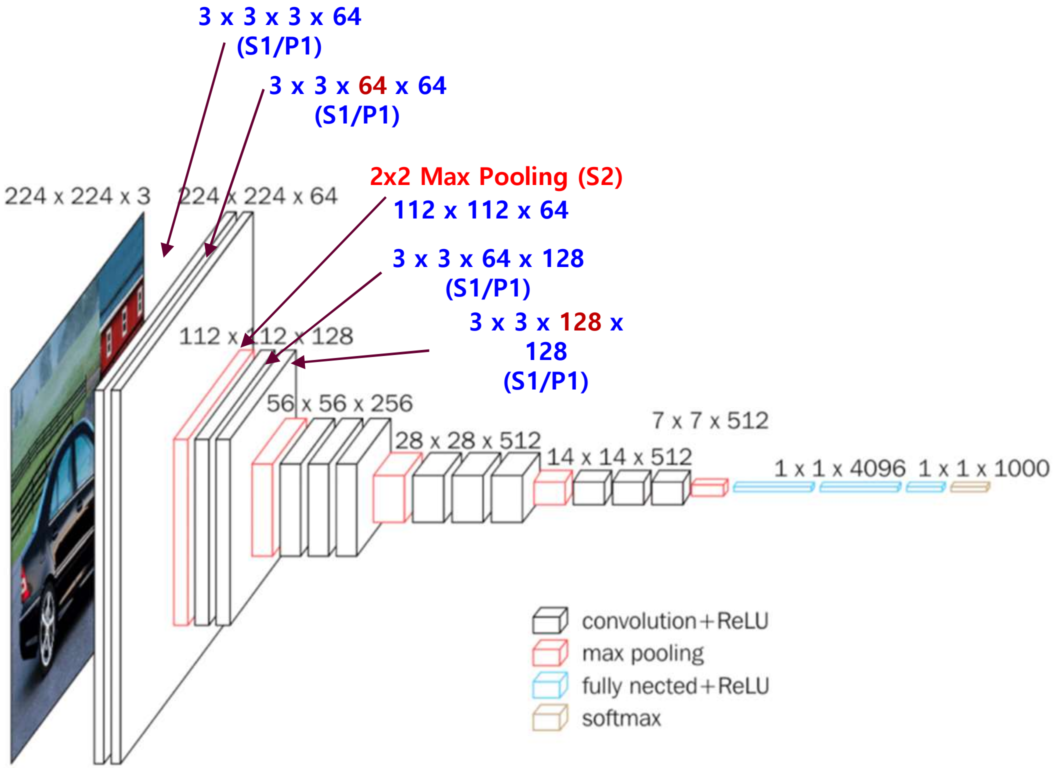


Ref) https://bskyvision.com/504

동아대학교 DONG-A UNIVERSITY

VGGNet(VGG-16 / VGG-19)

- **Weight parameter의 개수와 성능에 대한 trade-off를 탐색**
- **Network의 깊이가 깊어짐에 따라 높은 성능을 보임을 증명 (이후부터 네트워크 레이어를 증가시키는 추세가 활발히 이루어짐)**
- **필터크기는 3x3으로 고정 & ReLU 사용**



Ref) https://bskyvision.com/504

<VGG-16 구조>

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

<VGGNet 실험 설정 예시>

REVIEW (PART1 MLP)

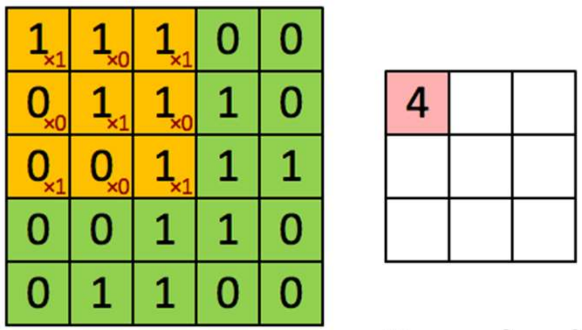
- MLP(FC Layer) Forward Propagation: Perceptron, Activation Functions, L1/L2 Loss Functions
- MLP(FC Layer) Backward Propagation: Gradient Descent(GD) Method
- Various Techniques & Implementations
  - Overfitting Problem
  - Vanishing Gradient
  - Data Argumentation
  - Optimization
  - Drop-out
  - Hyper-parament Control (Ex., Adaptive Learning Rate, mini-batch, epoch, etc...)
  - Ablation Works

REVIEW (PART2 CNN)

- CNN Forward Propagation: Convolution, Max/Avg Pooling, Padding, Stride, Kernel(Filter)
- CNN Backward Propagation
- CNN Network Design Schemes: SKIP Connection, Dense Connection, Channel Attention, Bottleneck Layer
- Popular CNN Networks: LeNet5, AlexNet, VGG, ResNet
- CNN Implementation: LeNet5, VGG

CNN 1D Backpropagation

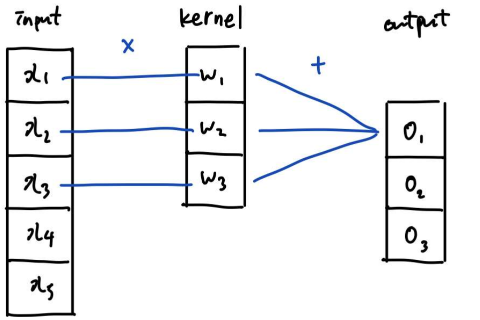
CNN 2D Convolution Operation (Forward Pass@S1/P0)



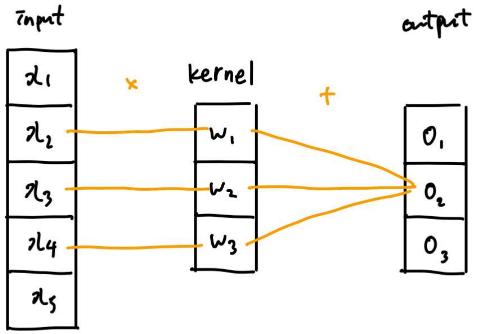
Image

Convolved Feature

CNN 1D Convolution Operation (Forward Pass@S1/P0)

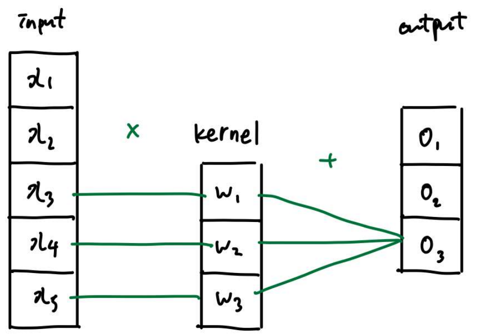


$o_1 = \boxed{w_1 x_1} + w_2 x_2 + w_3 x_3$



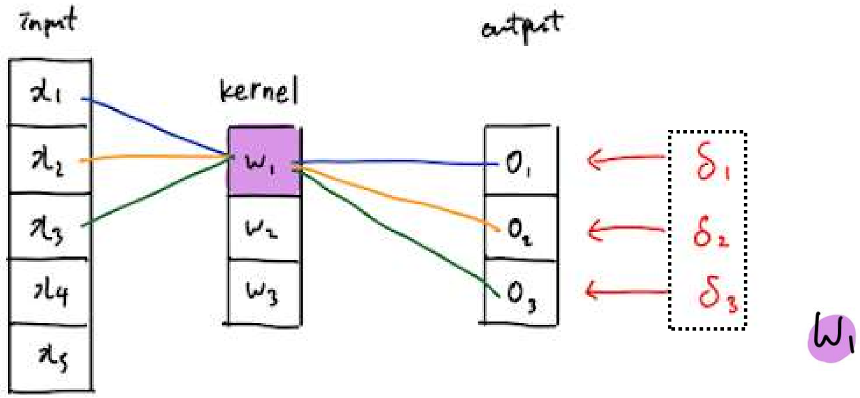
$o_2 = \boxed{w_1 x_2} + w_2 x_3 + w_3 x_4$

$$\therefore \quad o_i = \sum_j w_j x_{i+j-1}$$

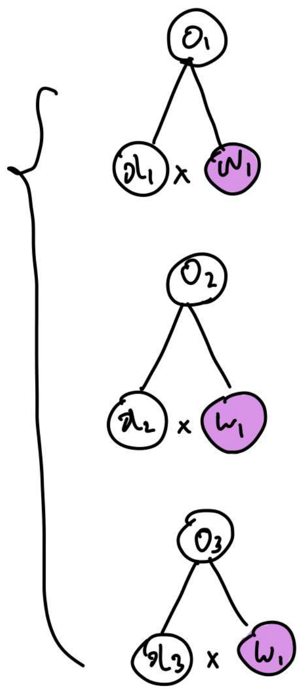


$o_3 = \boxed{w_1 x_3} + w_2 x_4 + w_3 x_5$

CNN 1D Convolution Operation (Backpropagation)



Global Gradient



$O_1 = w_1 x_1 + w_2 x_2 + w_3 x_3$

$$\frac{\partial O_1}{\partial W_1} = x_1$$

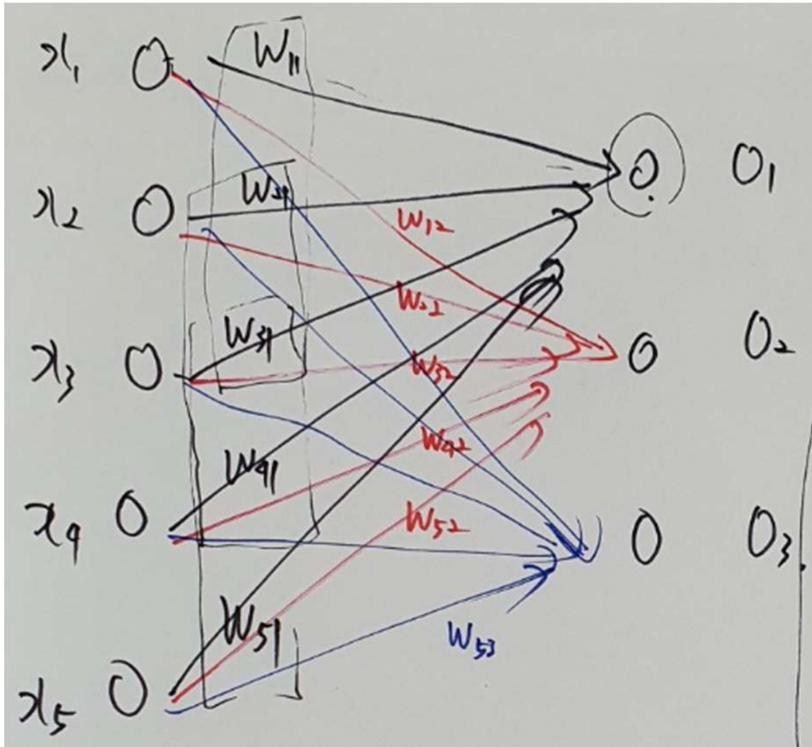
$$\frac{\partial O_2}{\partial W_1} = x_2$$

$$\frac{\partial O_3}{\partial W_1} = x_3$$

Local Gradient

$$\frac{\partial L}{\partial W_1} = \delta_1 \cdot \frac{\partial O_1}{\partial W_1} + \delta_2 \cdot \frac{\partial O_2}{\partial W_1} + \delta_3 \cdot \frac{\partial O_3}{\partial W_1}$$

$$= \delta_1 x_1 + \delta_2 x_2 + \delta_3 x_3$$



$\therefore O_1 = w_{11}x_1 + w_{21}x_2 + \cdot + w_{51}x_5$

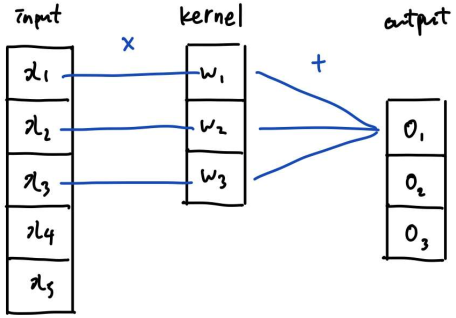
$O_2 = W_{12}\cdot x_1 + W_{22}x_2 + \cdot + W_{52}x_5$

$O_3 = W_{13}x_1 + W_{23}x_2 + \cdot + W_{53}x_3$

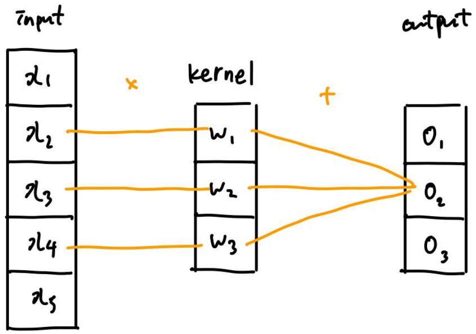
$\Downarrow$

서로 다른 weights.

CNN 1D Convolution Operation (Forward Pass@S1/P0)

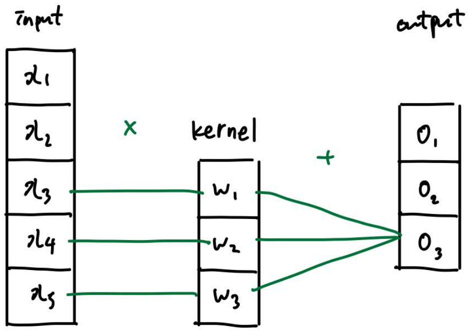


$$O_1 = w_1 x_1 + \boxed{w_2 x_2} + w_3 x_3$$



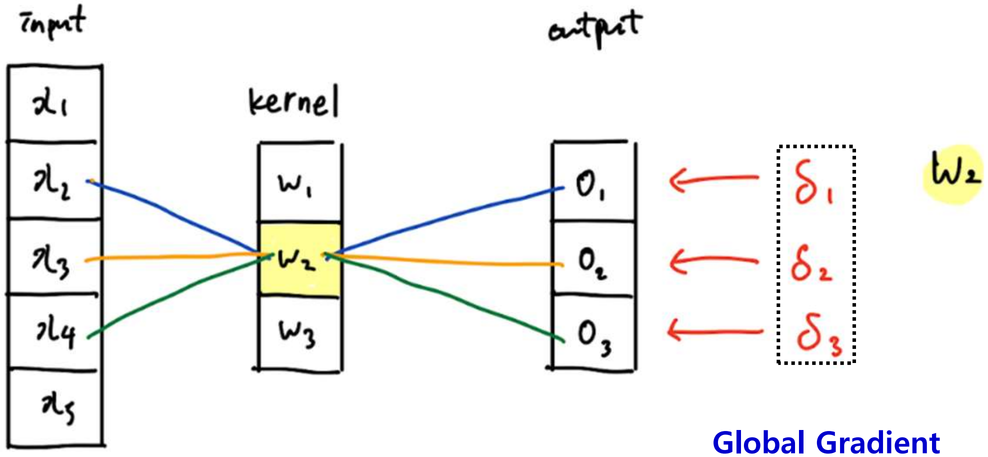
$$O_2 = w_1 x_2 + \boxed{w_2 x_3} + w_3 x_4$$

$$\therefore O_i = \sum_j w_j x_{i+j-1}$$

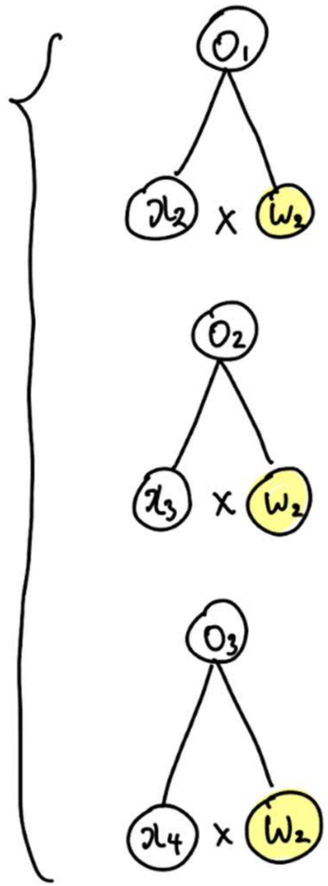


$$O_3 = w_1 x_3 + \boxed{w_2 x_4} + w_3 x_5$$

CNN 1D Convolution Operation (Backpropagation)



Global Gradient



$$\frac{\partial O_1}{\partial W_2} = x_2$$

$$\frac{\partial O_2}{\partial W_2} = x_3$$

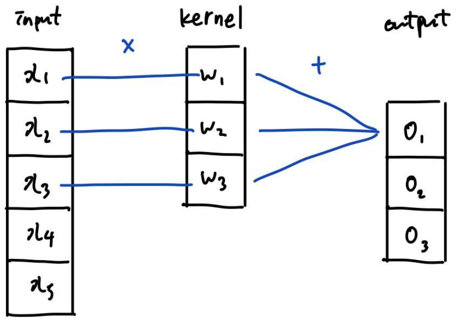
$$\frac{\partial O_3}{\partial W_2} = x_4$$

Local Gradient

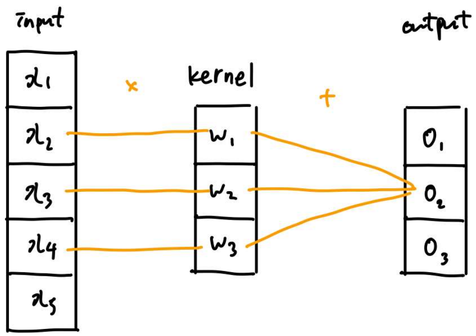
$$\frac{\partial L}{\partial W_2} = \delta_1 \cdot \frac{\partial O_1}{\partial W_2} + \delta_2 \cdot \frac{\partial O_2}{\partial W_2} + \delta_3 \cdot \frac{\partial O_3}{\partial W_2}$$

$$= \delta_1 x_2 + \delta_2 x_3 + \delta_3 x_4$$

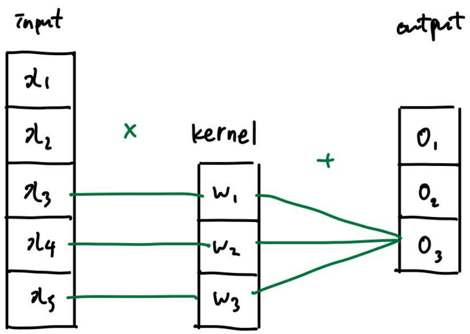
CNN 1D Convolution Operation (Forward Pass@S1/P0)



$$O_1 = w_1x_1 + w_2x_2 + \boxed{w_3x_3}$$



$$O_2 = w_1x_2 + w_2x_3 + \boxed{w_3x_4}$$

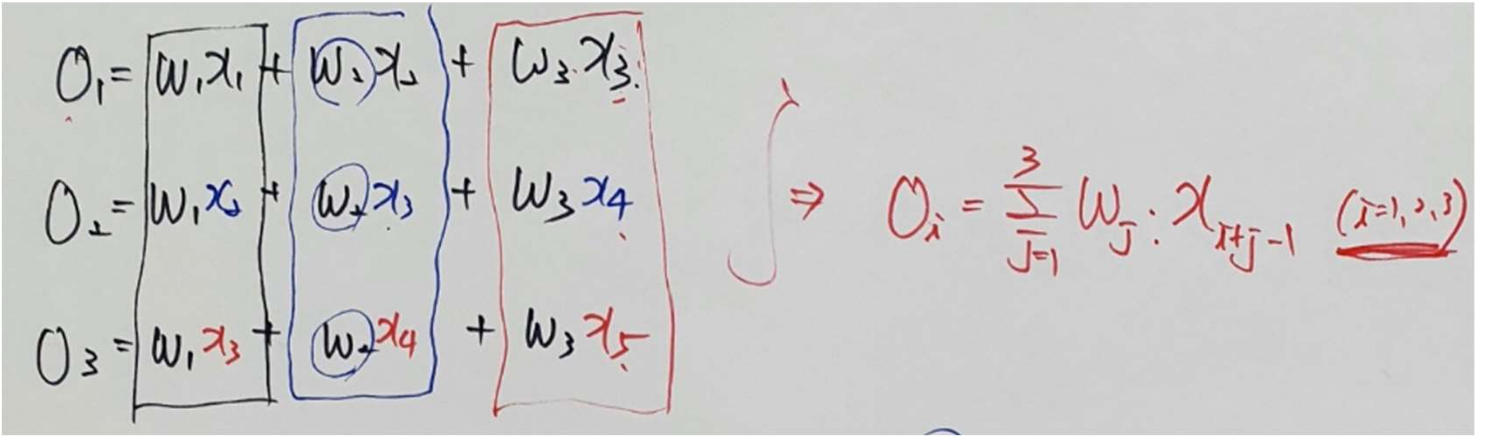


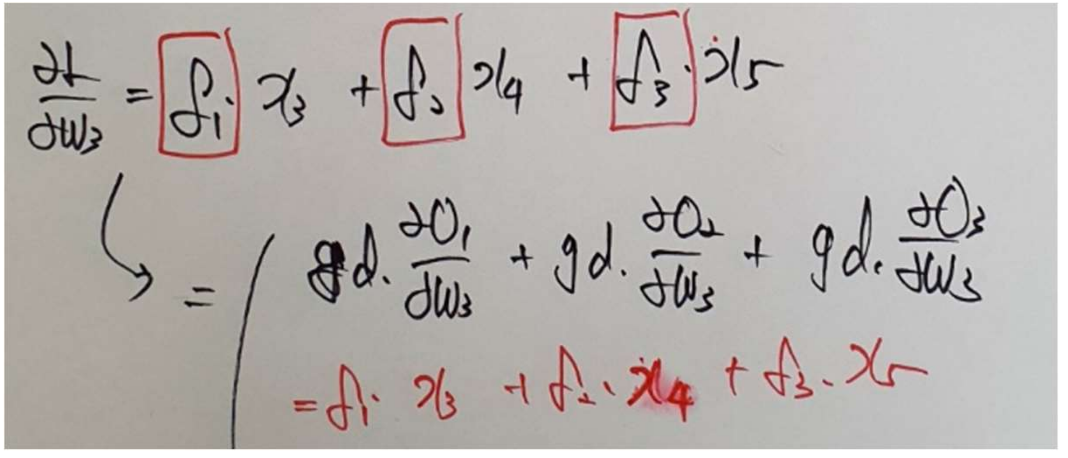
$$O_3 = w_1x_3 + w_2x_4 + \boxed{w_3x_5}$$

$$\therefore\ O_i = \sum_j w_j x_{i+j-1}$$

CNN 1D Convolution Operation (Backpropagation)

Gradient w_3?





CNN 1D Convolution Operation (Backpropagation)

$\boxed{2}$ i) $\dfrac{\partial L}{\partial w_1} = \underbrace{g.d.}_{\delta_1}\,\underbrace{\dfrac{\partial O_1}{\partial w_1}}_{x_1} + \underbrace{g.d.}_{\delta_2}\,\underbrace{\dfrac{\partial O_2}{\partial w_1}}_{x_2} + \underbrace{g.d.}_{\delta_3}\,\underbrace{\dfrac{\partial O_3}{\partial w_1}}_{x_3}$

$= \boxed{\delta_1} x_1 + \boxed{\delta_2} x_2 + \boxed{\delta_3} x_3$

ii) $\dfrac{\partial L}{\partial w_2} = g.d.\dfrac{\partial O_1}{\partial w_2} + g.d.\dfrac{\partial O_2}{\partial w_2} + g.d\,\dfrac{\partial O_3}{\partial w_3}$

$= \boxed{\delta_1} \cdot x_2 + \boxed{\delta_2} \cdot x_3 + \boxed{\delta_3} \cdot x_4$

iii) $\dfrac{\partial L}{\partial w_3} = \boxed{\delta_1} \cdot x_3 + \boxed{\delta_2} \cdot x_4 + \boxed{\delta_3} \cdot x_5$

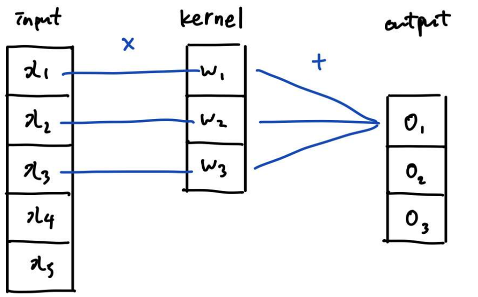
$= g.d.\dfrac{\partial O_1}{\partial w_3} + g.d.\dfrac{\partial O_2}{\partial w_3} + g.d.\dfrac{\partial O_3}{\partial w_3}$

$= \delta_1 \cdot x_3 + \delta_2 \cdot x_4 + \delta_3 \cdot x_5$

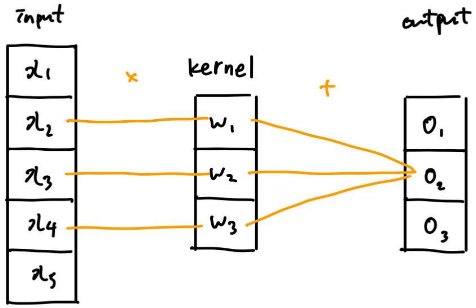
$$\therefore \frac{\partial L}{\partial w_i} = \sum_j \delta_j x_{i+j-1}$$

CNN 1D Convolution Operation (Backpropagation)

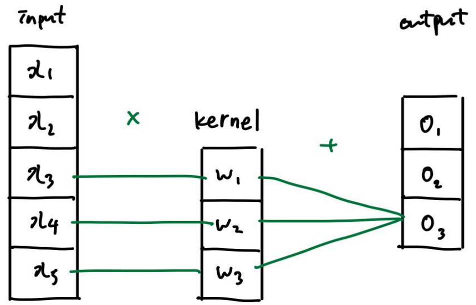
Gradient x?



$O_1 = w_1x_1 + w_2x_2 + w_3x_3$



$O_2 = w_1x_2 + w_2x_3 + w_3x_4$



$O_3 = w_1x_3 + w_2x_4 + w_3x_5$

$\boxed{3}$

i) $\frac{\partial L}{\partial x_1} = g.d \cdot \frac{\partial O_1}{\partial x_1} = f_1 \cdot W_1$

ii) $\frac{\partial L}{\partial x_2} = g.d \frac{\partial O_1}{\partial x_2} + g.d \cdot \frac{\partial O_2}{\partial x_2} = f_1 \cdot W_2 + f_2 \cdot W_1$

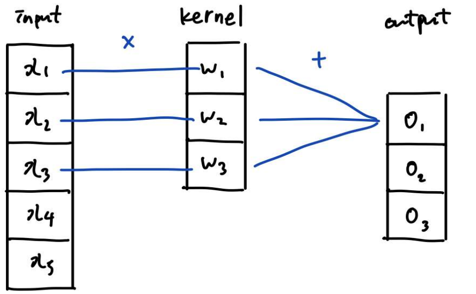
iii) $\frac{\partial L}{\partial x_3} = g.d. \frac{\partial O_1}{\partial x_3} + g.d. \frac{\partial O_2}{\partial x_3} + g.d. \frac{\partial O_3}{\partial x_3} = f_1W_3 + f_2W_2 + f_3W_1$

iv) $\frac{\partial L}{\partial x_4} = g.d \frac{\partial O_2}{\partial x_4} + g.d. \frac{\partial O_3}{\partial x_4} = f_2W_3 + f_3W_2$

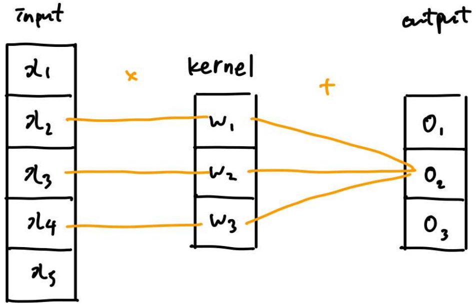
v) $\frac{\partial L}{\partial x_5} = g.d \frac{\partial O_3}{\partial x_5} = f_3 \cdot W_3$

CNN 1D Convolution Operation (Backpropagation)

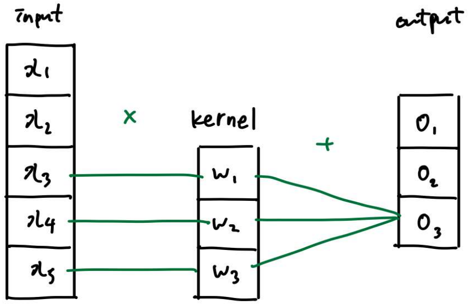
Gradient x?



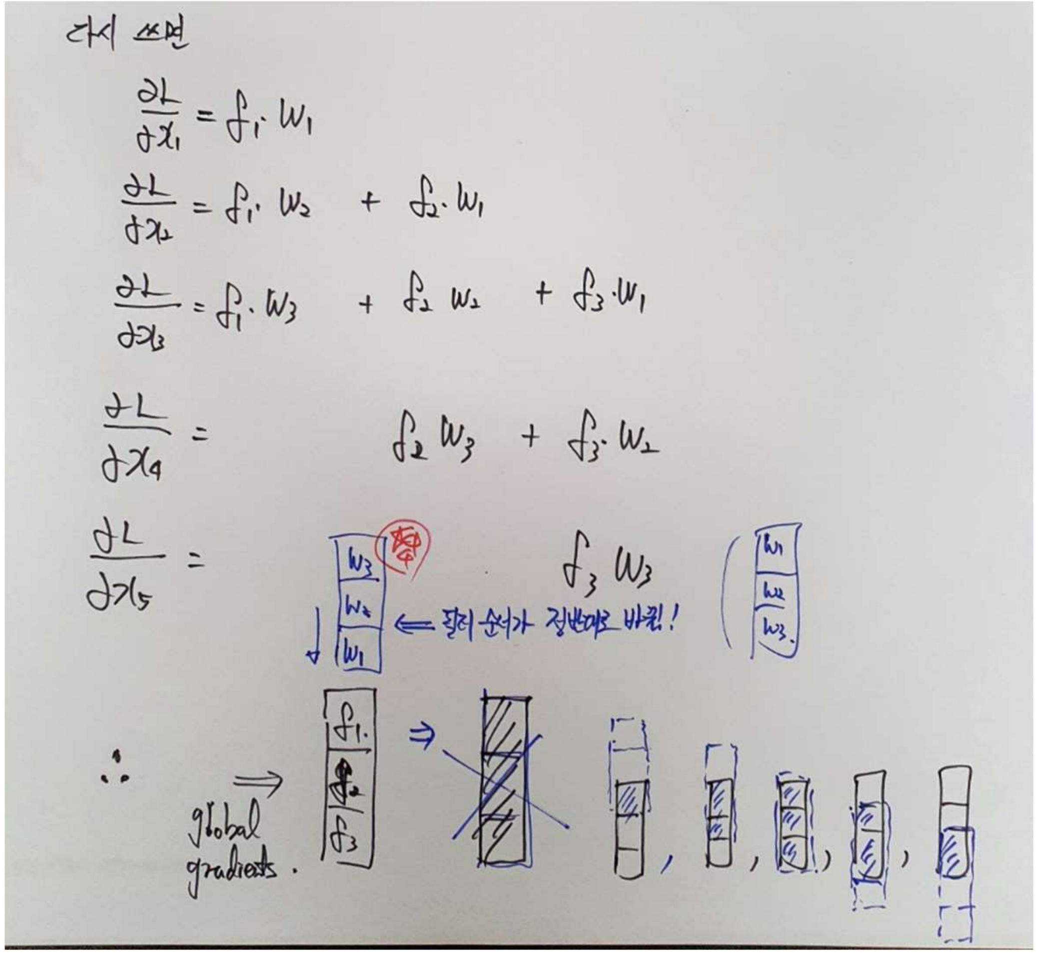
$O_1 = w_1 x_1 + w_2 x_2 + w_3 x_3$



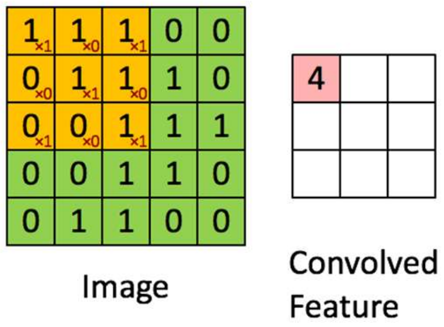
$O_2 = w_1 x_2 + w_2 x_3 + w_3 x_4$

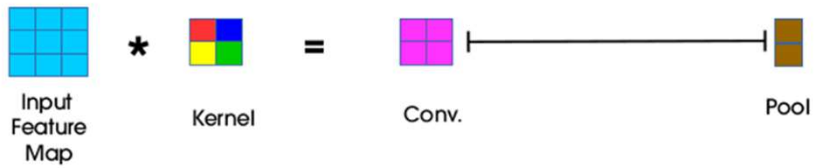


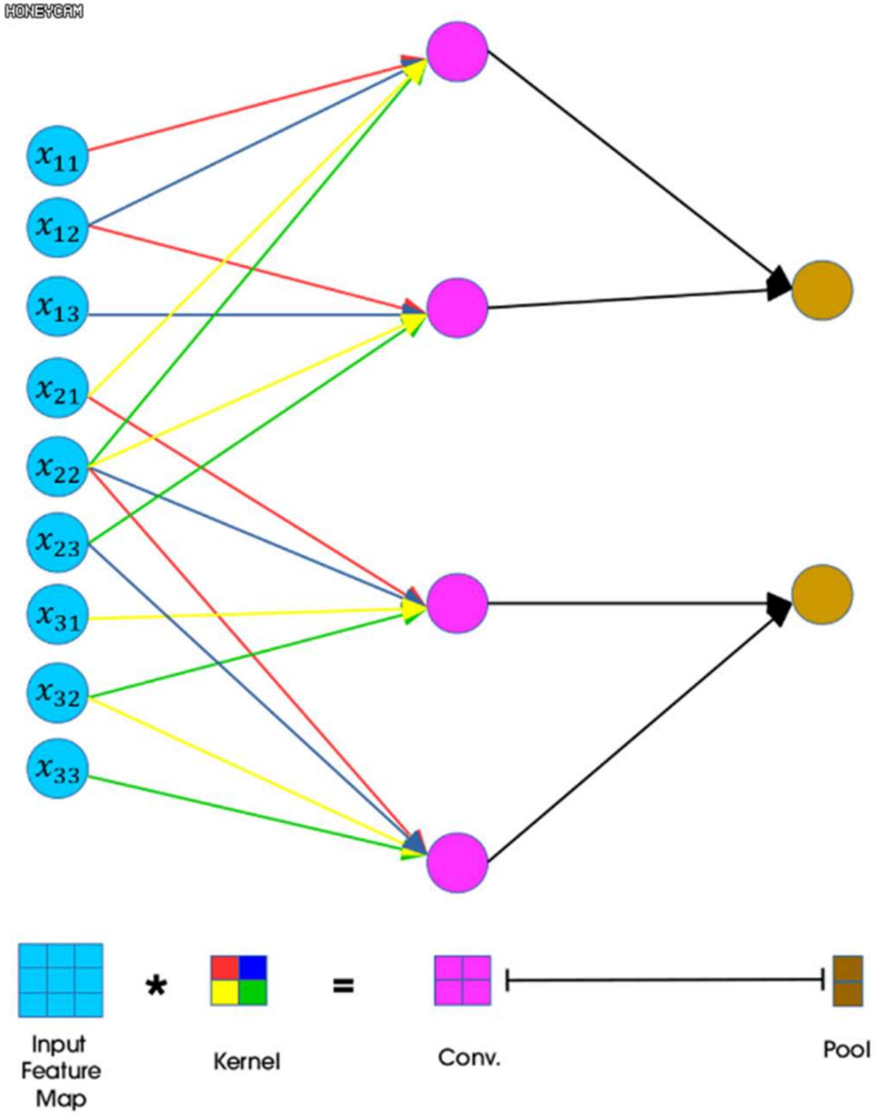
$O_3 = w_1 x_3 + w_2 x_4 + w_3 x_5$

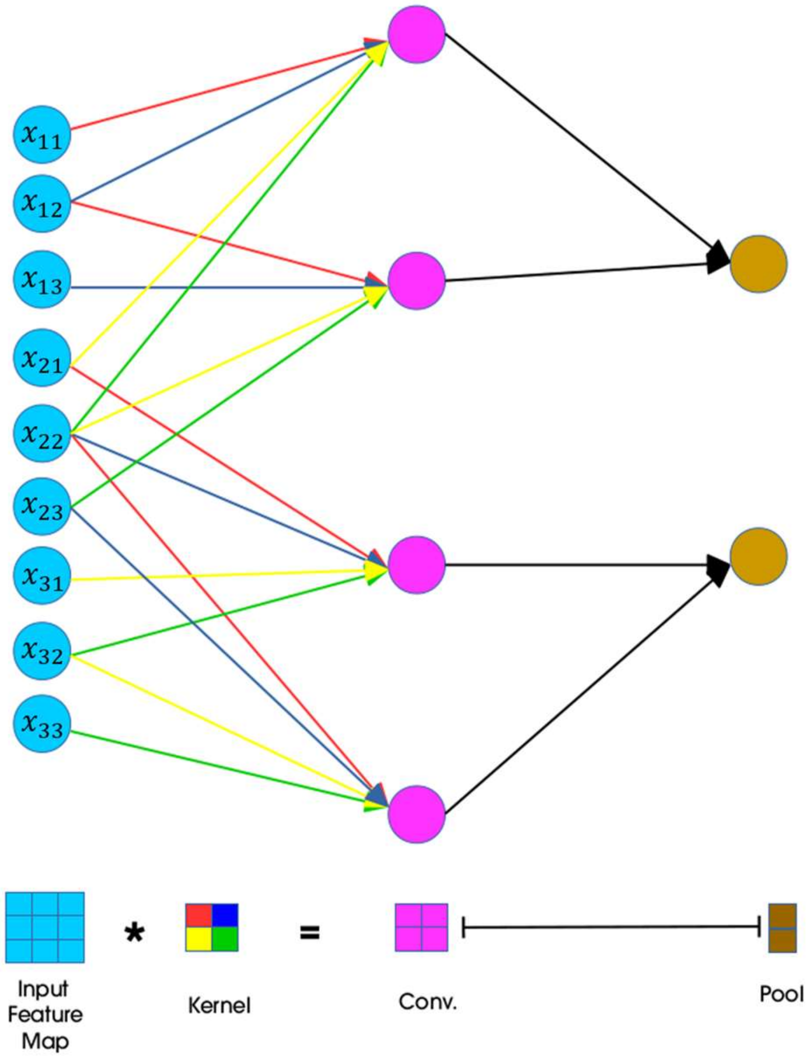


CNN 2D Backpropagation



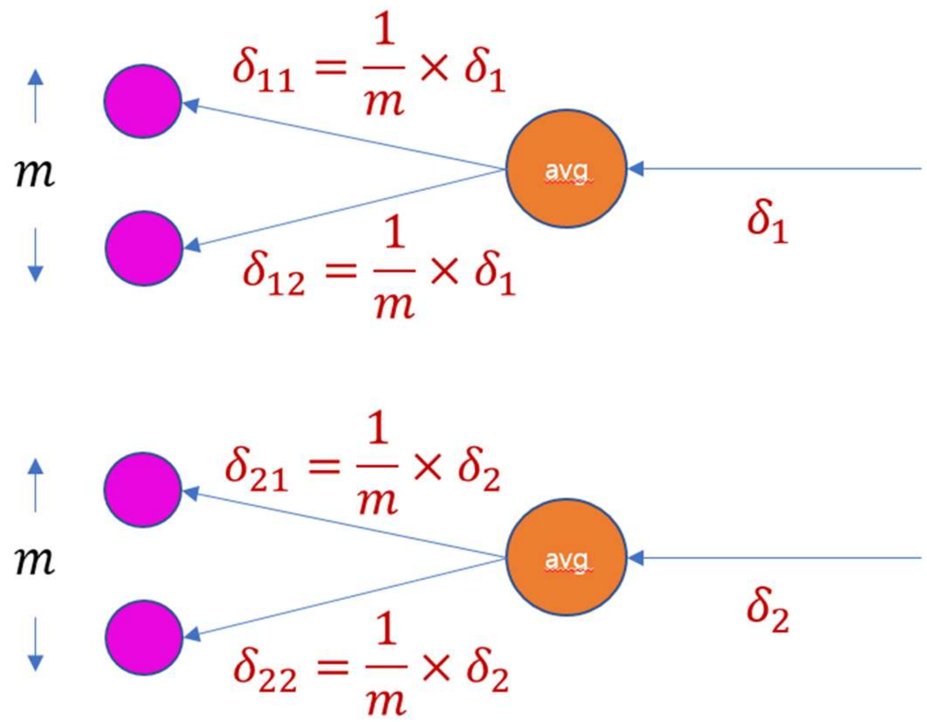



x_{11}
x_{12}
x_{13}
x_{21}
x_{22}
x_{23}
x_{31}
x_{32}
x_{33}
*
=
Input Feature Map
Kernel
Conv.
Pool

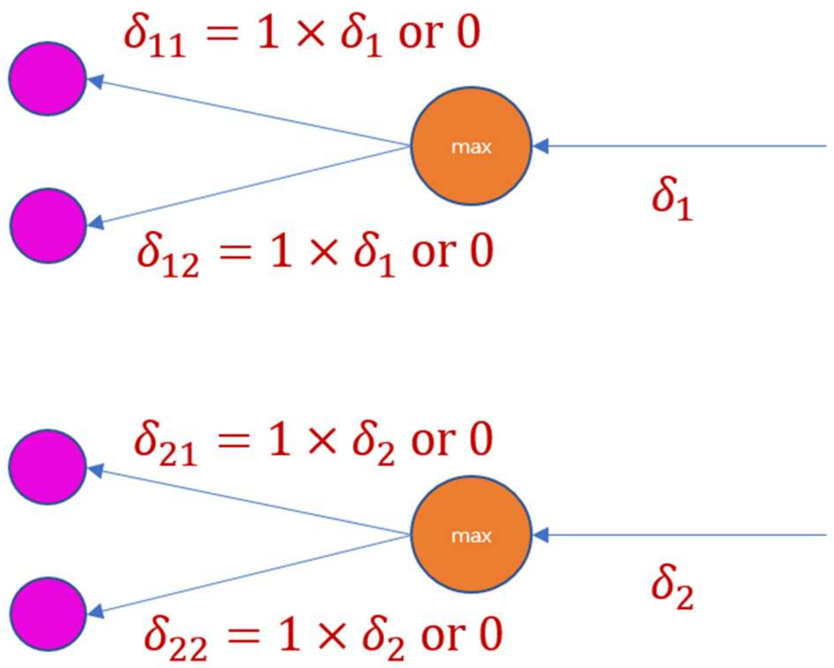

x_{11}
x_{12}
x_{13}
x_{21}
x_{22}
x_{23}
x_{31}
x_{32}
x_{33}
*
=
Input Feature Map
Kernel
Conv.
Pool

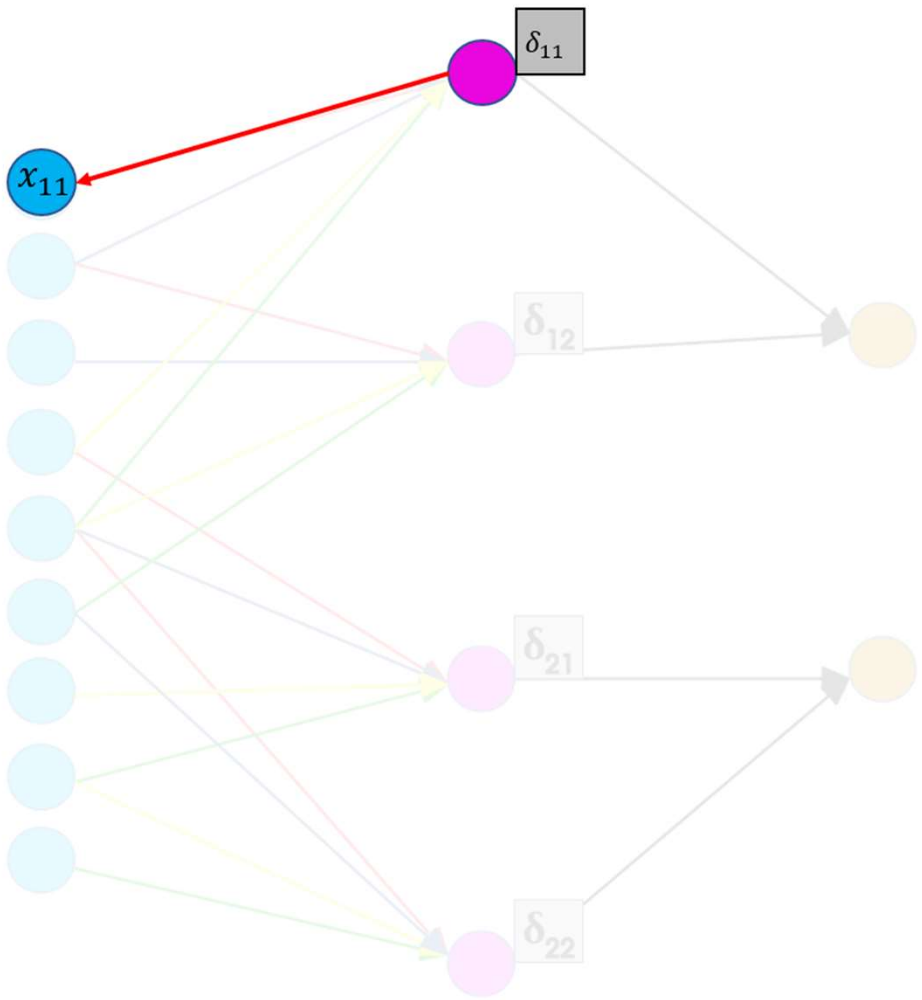
참고

Average Pooling

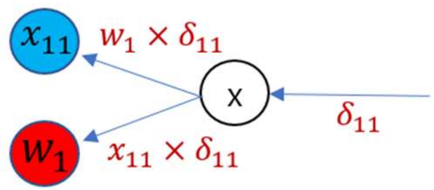


Max Pooling

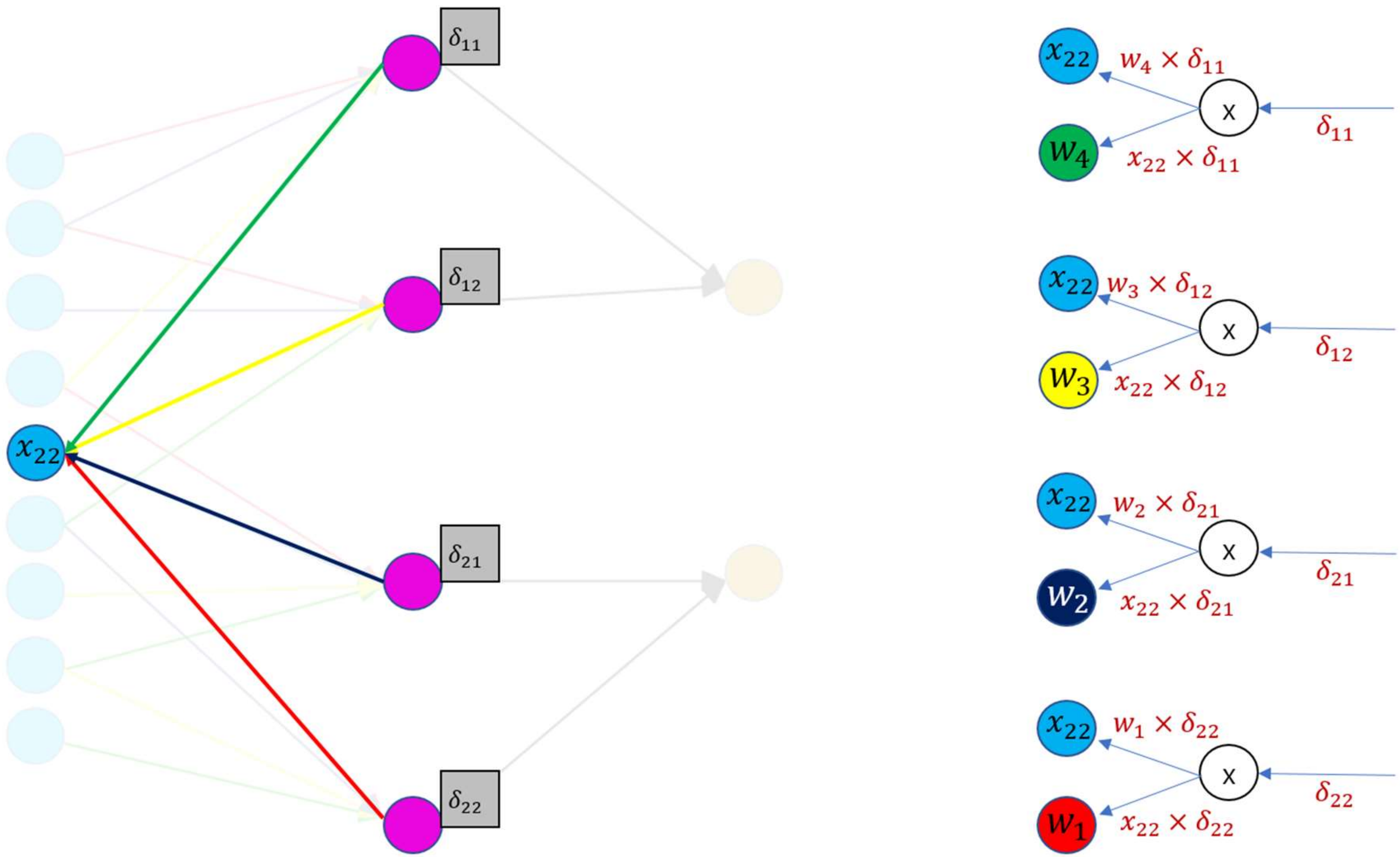


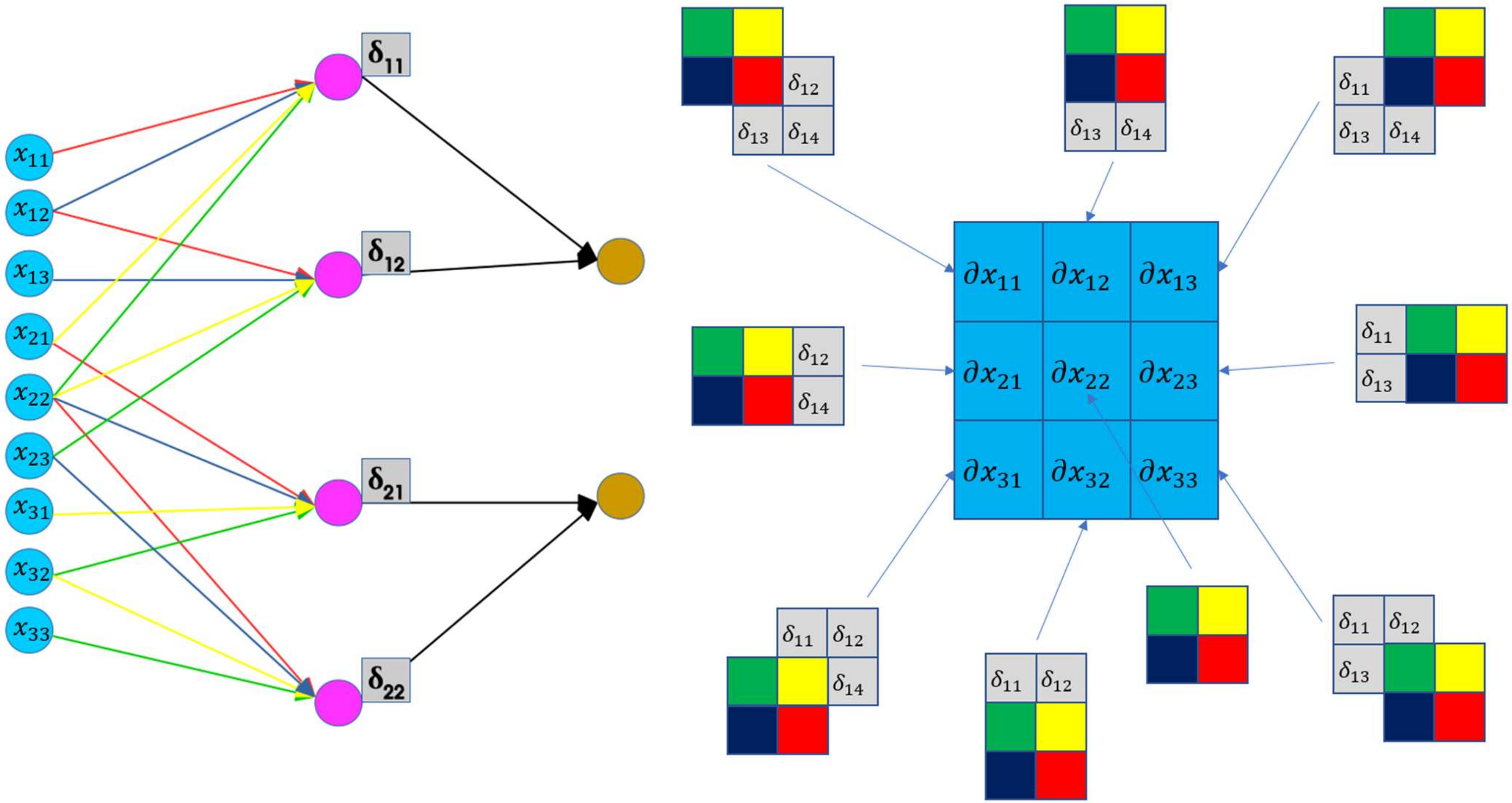


δ_{11}
x_{11}
δ_{12}
δ_{21}
δ_{22}



x_{11}
$w_1 \times \delta_{11}$
x
δ_{11}
w_1
$x_{11} \times \delta_{11}$


δ_{11}
δ_{12}
δ_{21}
δ_{22}
x_{22}
x_{22}
$w_4 \times \delta_{11}$
x
δ_{11}
w_4
$x_{22} \times \delta_{11}$
x_{22}
$w_3 \times \delta_{12}$
x
δ_{12}
w_3
$x_{22} \times \delta_{12}$
x_{22}
$w_2 \times \delta_{21}$
x
δ_{21}
w_2
$x_{22} \times \delta_{21}$
x_{22}
$w_1 \times \delta_{22}$
x
δ_{22}
w_1
$x_{22} \times \delta_{22}$



x_{11}
x_{12}
x_{13}
x_{21}
x_{22}
x_{23}
x_{31}
x_{32}
x_{33}
δ_{11}
δ_{12}
δ_{21}
δ_{22}
∂x_{11} ∂x_{12} ∂x_{13}
∂x_{21} ∂x_{22} ∂x_{23}
∂x_{31} ∂x_{32} ∂x_{33}
δ_{12} δ_{13} δ_{14}
δ_{13} δ_{14}
δ_{11} δ_{13} δ_{14}
δ_{12} δ_{14}
δ_{11} δ_{13}
δ_{11} δ_{12} δ_{14}
δ_{11} δ_{12}
δ_{11} δ_{12} δ_{13}