Neural Network

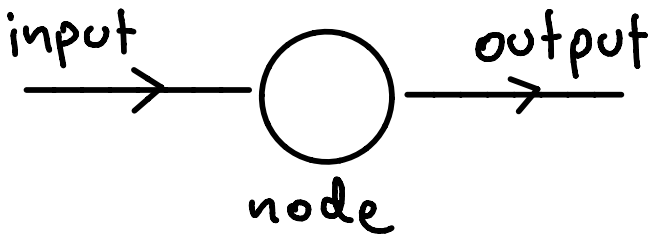
Petchara Pattarakijwanich

Introduction to Data Science, 25 November 2022

Goal of this week

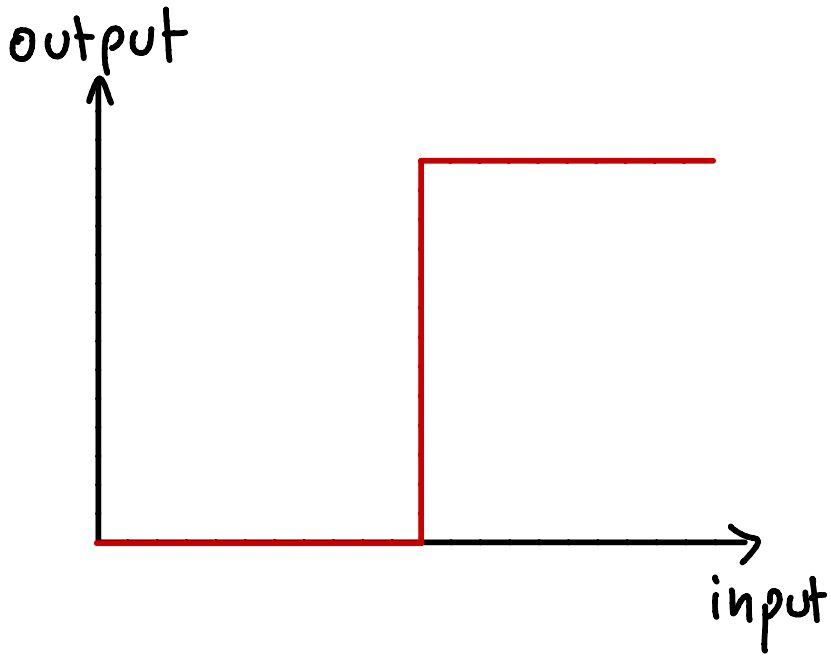
- Introduction to Neural Network
- Feed-Forward Neural Network (FNN)
  - Simple Example
  - Activation Function
  - Classification vs Regression
- Convolutional Neural Network (CNN)
  - Filters
  - Convolution and Pooling
- Back Propagation and Fitting
- How to Prevent Overfitting

Intro to NN



(≈ เซลล์ประสาท)

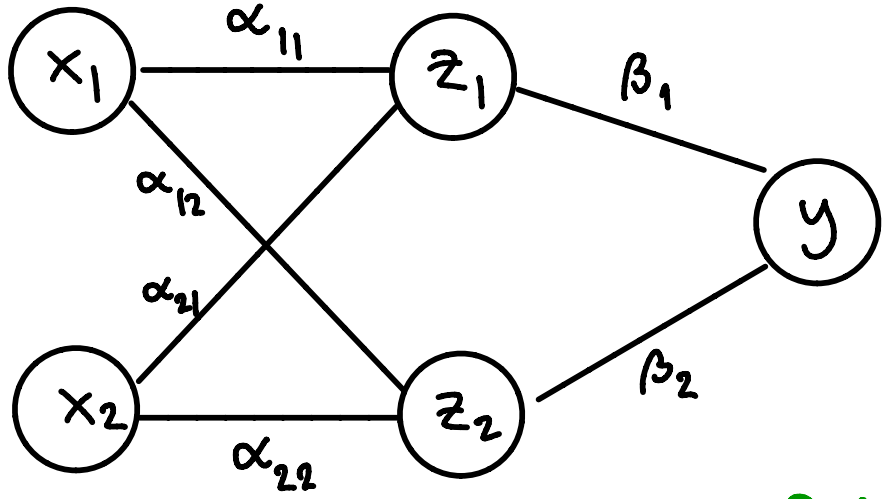
Activation function



output ขึ้นกับ input

Intro to NN

Ex. Shown input : $\{x_1, x_2\}$ output : y



$$z_1 = \sigma(\alpha_{11}x_1 + \alpha_{21}x_2 + \alpha_1)$$

$$z_2 = \sigma(\alpha_{12}x_1 + \alpha_{22}x_2 + \alpha_2)$$

$$y = \sigma(\beta_1 z_1 + \beta_2 z_2 + \beta_0)$$

$\alpha_1, \alpha_2, \beta_0$ = bias

$$z_1 = \sigma(\alpha_{11}x_1 + \alpha_{21}x_2 + \alpha_1)$$

$$z_2 = \sigma(\alpha_{12}x_1 + \alpha_{22}x_2 + \alpha_2)$$

Linear combination

$$y = \sigma(\beta_1 z_1 + \beta_2 z_2 + \beta_0)$$

Linear combination

σ = activation function
= non-linear mapping

σ - Sigmoid
- hyperbolic tangent
- ReLU
- Leaky ReLU

Intro to NN

เป้าหมายของ NN

หาค่า weight α, β ที่ทำให้โมเดล
output ใกล้จริง ($y_{model} \approx y_{data}$)

$\Rightarrow$ คือ train Neural Network

(α, β เรียกว่า learnable parameters)

ถ้ามี hidden node เพียงพอ
NN ประมาณฟังก์ชันต่อเนื่อง
ใดๆ ได้

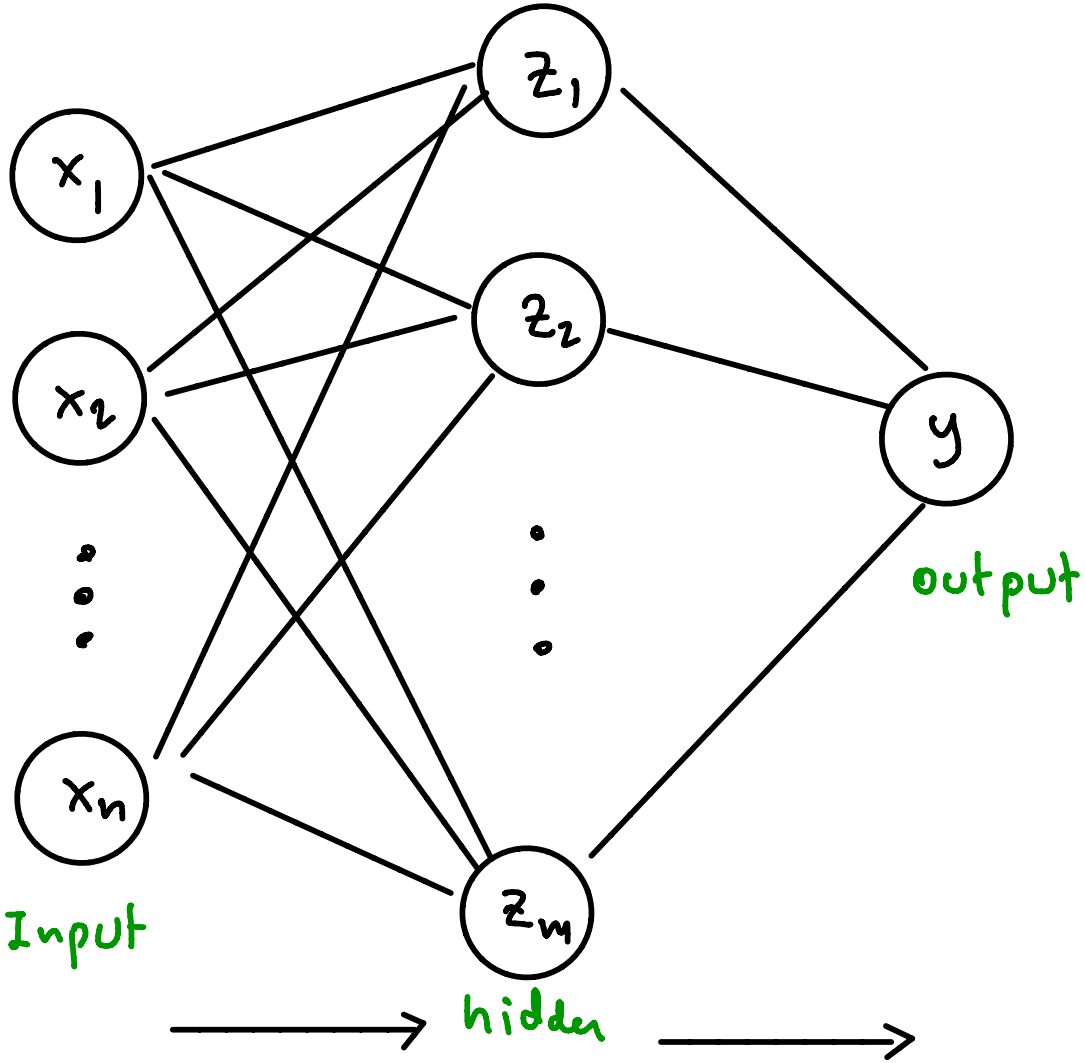
"Universal Approximation Theorem"

Hyperparameters (optimize ไม่ได้)

- จำนวน node / จำนวน hidden layer (depth)
- ใช้ activation function อะไร
- เชื่อม NN ยังไงๆ ↗ fully-connected
  ↘ substructure

Feed-Forward NN

Input : $\{x_1, \dots x_n\}$ output : $\{y\}$



เป้าหมาย : หา weight α/β ที่ดีที่สุด

(จำนวน α, β มีกี่เยอะ)

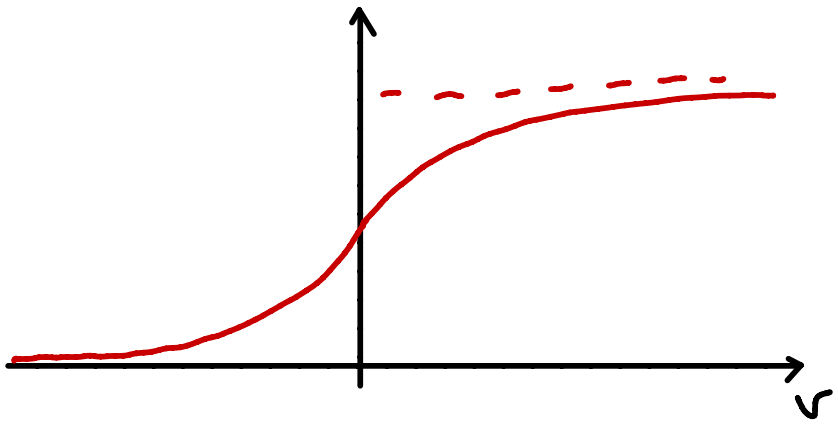
$$y = f(x_1, x_2, \dots x_n)$$

ฟังก์ชัน

$$z_i = \sigma(\vec{\alpha} \cdot \vec{x} + \alpha_0) \qquad y = \sigma(\vec{\beta} \cdot \vec{z} + \beta_0)$$

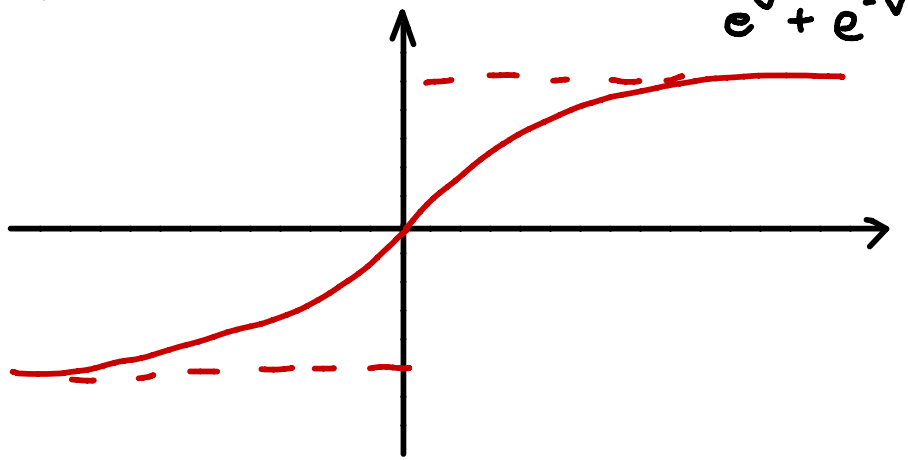
Activation Functions

Sigmoid $\sigma(v) = \dfrac{1}{1+e^{-v}}$



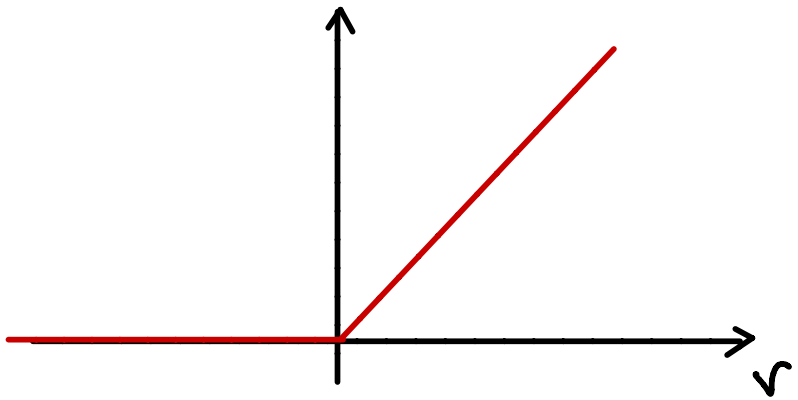
$\sigma(v) = \tanh(v)$

Hyperbolic tangent $= \dfrac{e^{v}-e^{-v}}{e^{v}+e^{-v}}$

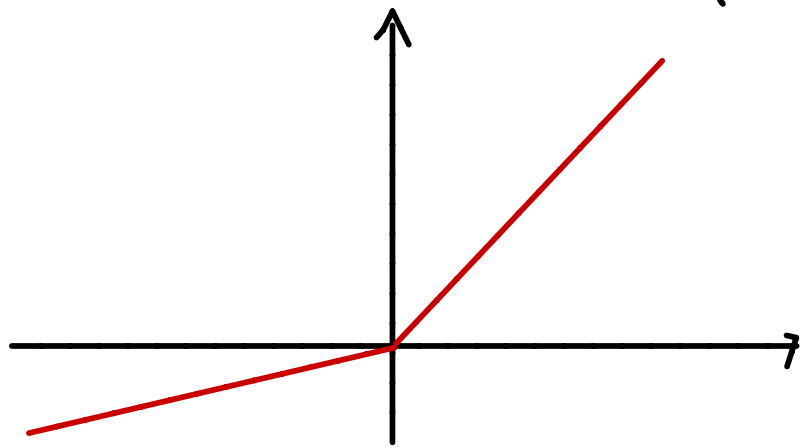


ReLU $\sigma(v) = \max(v,0) = \begin{cases} v \\ 0 \end{cases}$

(Rectified Linear Unit)

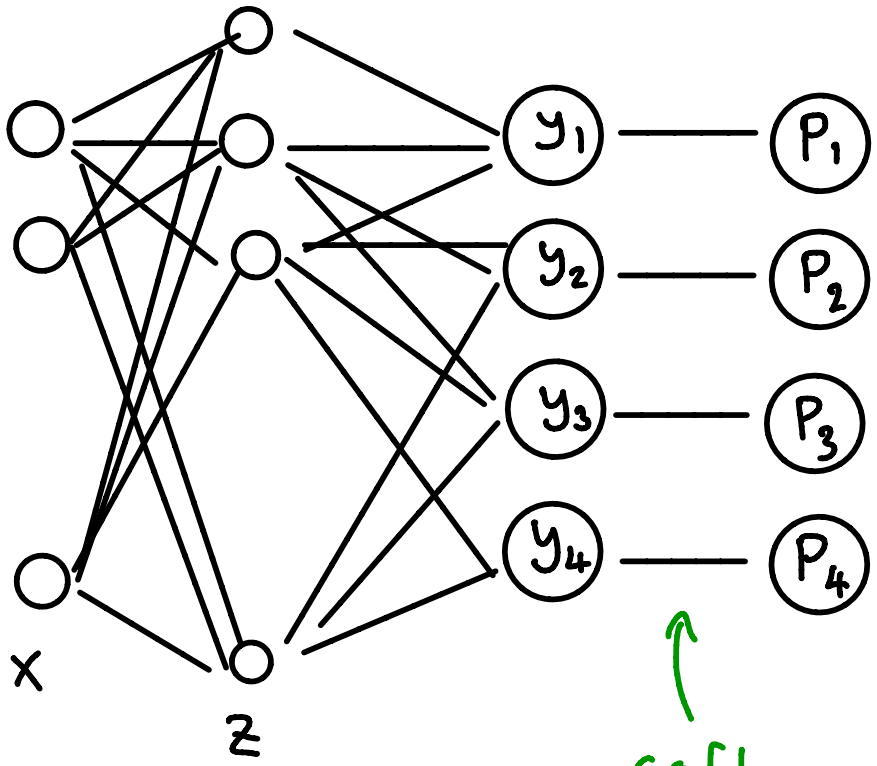


Leaky ReLU $\sigma(v) = \begin{cases} v \\ av \end{cases}$ $(a \approx 0.01)$



Classification vs Regression

ต้องการ output มี 4 ตัว



softmax

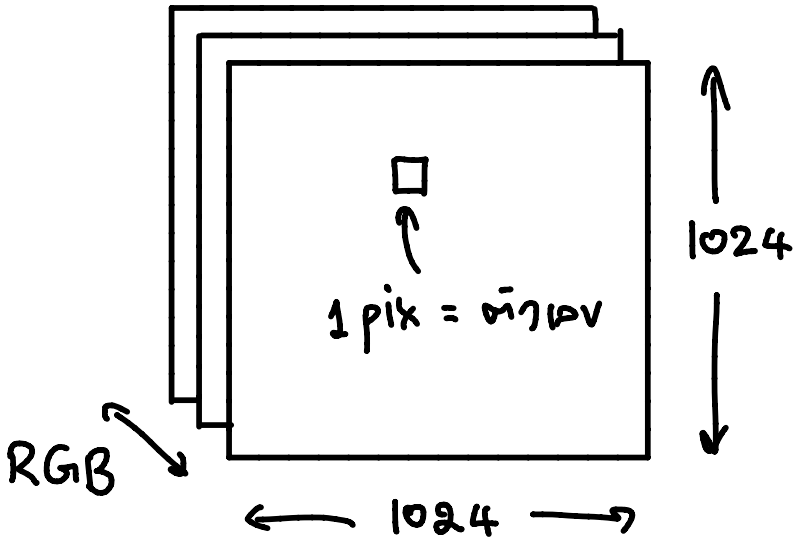
- y เข้า → P ออก
- $\sum P_i = 1$

($P_i \approx$ probability ของตัวที่ i)

softmax function

$$P_i = \frac{e^{y_i}}{e^{y_1} + e^{y_2} + e^{y_3} + e^{y_4}}$$

Convolutional NN



$\Rightarrow$ 1 รูป = $1024 \times 1024 \times 3$

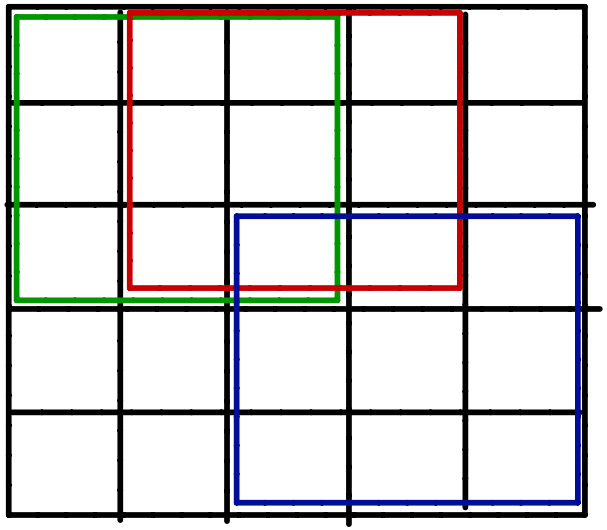
$\approx 3 \times 10^6$ ตัวเลข

- FNN สามารถ classify รูปภาพได้ เพราะตัวแปรเยอะเกิน
- มี length scale ที่ระดับของภาพ
- pixel ที่อยู่ใกล้กัน มีความสัมพันธ์กัน

$\Rightarrow$ ใช้ FNN ไม่ได้

CNN

Filter and Convolution



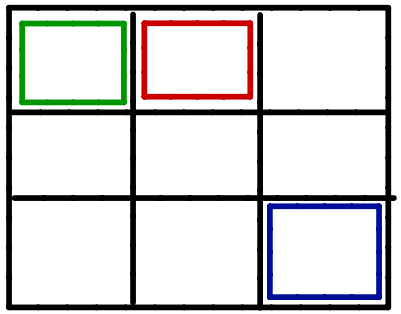
รูป 5×5 pix^2

0	1	0
1	1	1
0	1	0

Filter 3×3

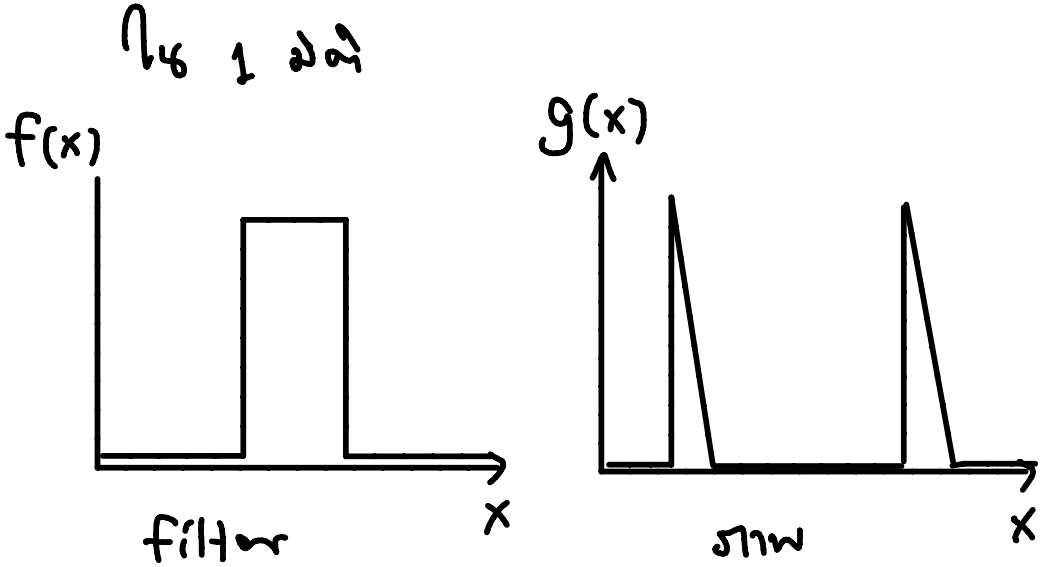
convolution

[รูป × filter → ขอบ
⇒ ใช้ filter หลายๆ]



รูป ⊛ filter

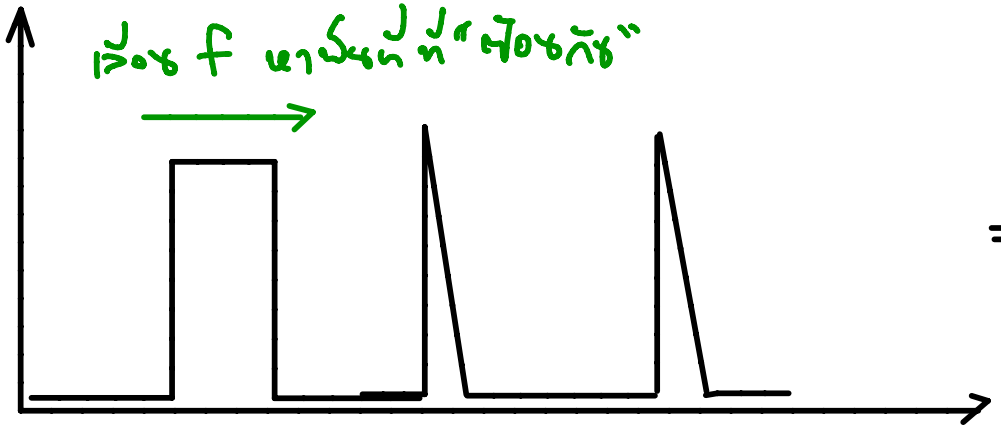
Filter and Convolution



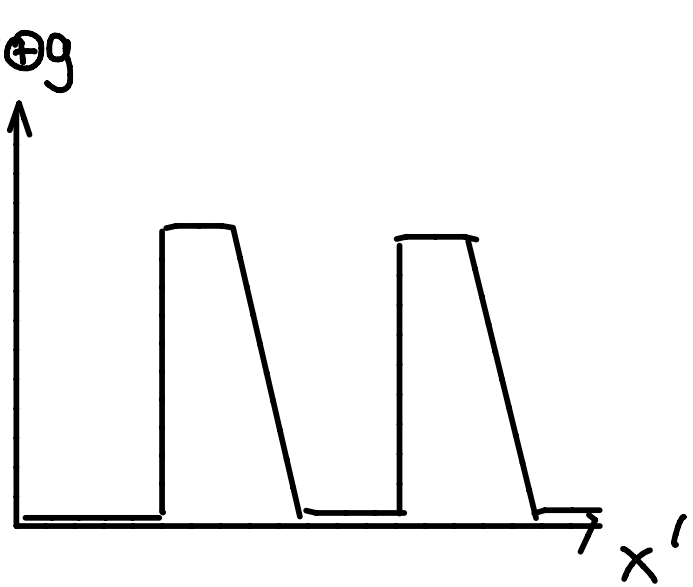
convolution

$$f \oplus g(x') = \int f(x - x')\, g(x)\, dx$$

เลื่อน f ไปตาม x'



$\Rightarrow$



Filter and Convolution

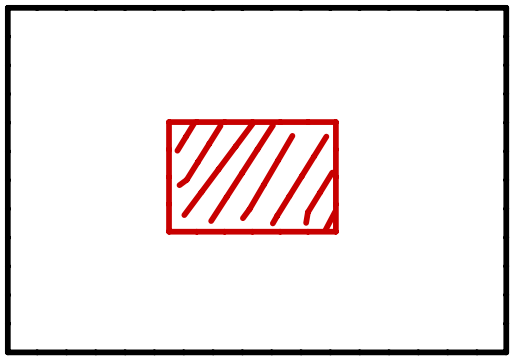
Ex.

1	1	1
0	0	0
-1	-1	-1

Horizontal Edge Detection

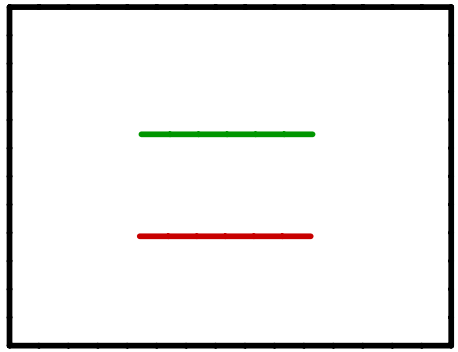
Filter ใช้ทำอะไรบ้าง

- เบลอภาพ
- หาจุด
- หาขอบ

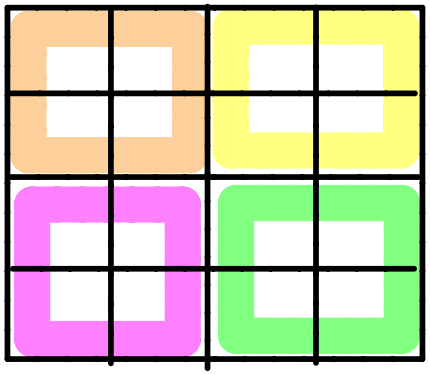


รูป

⇒ convolute

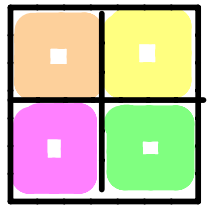


Pooling



รูปต้นฉบับ 4×4

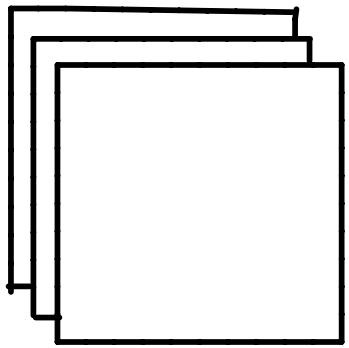
2×2 pooling



ผลลัพธ์ 2×2

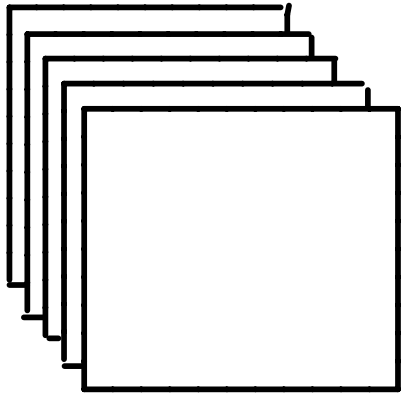
- Average Pooling (ค่าเฉลี่ย)
- Max Pooling (ค่ามากสุด) (นิยมใช้)

Convolutional NN



Input Layer

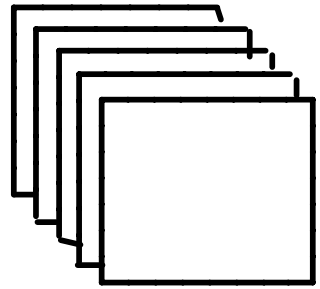
ภาพ = 1024×1024×3



Convolutional Layer

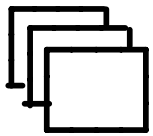
ใช้ 5 filter

≈ 1024×1024×5



2×2 pooling

≈ 512×512×5



filter

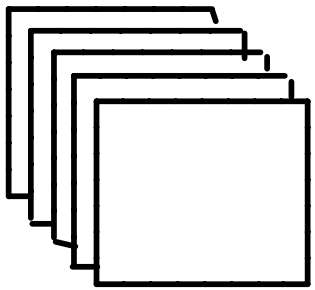
3×3×3 (5 ตัว)

hyperparamter

[ต่างๆ filter ก็มี Learnable parameters]

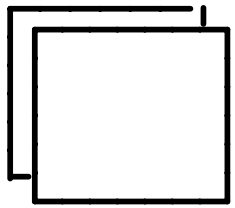
[optimize filter ที่เหมาะสม]

Convolutional NN



2×2 pooling

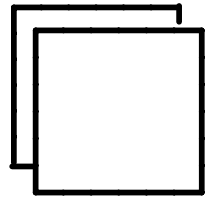
≈ 512×512×5



convolutional layer

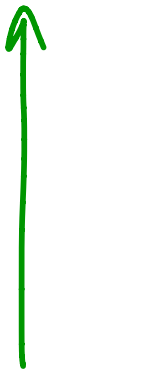
ใช้ 2 filter

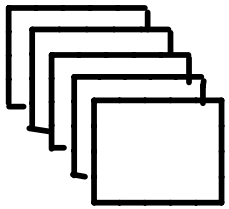
≈ 512×512×2



2×2 pooling

≈ 256×256×2





filter

3×3×5 (2 ชุด)

Convolutional NN

(scale เล็ก)

(scale ใหญ่)

Input Full Image ($\approx 1024 \times 1024$) $\xrightarrow{C+P}$ ภาพเล็กลง ($\approx 512 \times 512$) $\xrightarrow{C+P}$ ภาพเล็กลง ($\approx 256 \times 256$) $\xrightarrow{C+P}$ ... $\Rightarrow$ ภาพเล็กลง ($\approx 8 \times 8$)

$\Downarrow$

Flatten & FNN

$\Downarrow$

Output

output = f(input image)

↑ ตัวแปร

เป้าหมายของ CNN

- filter แต่ละตัวคือ learnable parameters
- ปรับ weight ของ filter เพื่อให้ได้ผลลัพธ์ดี

Hyperparameter

- convolution + pooling กี่ชั้น
- filter แต่ละชั้นกี่ pixel กี่ layer
- activation function อันไหน, ...