Lab Seminar: 2022. 07. 05.

Dive Into Convolutional Neural Network

(for newer to CNN)

IDEALAB

Improving lives through learning

JongHyeon Kim

School of Computer Science/Department of AI Convergence Engineering

Gyeongsang National University (GNU)

경상국립대학교 Gyeongsang National University

Contents

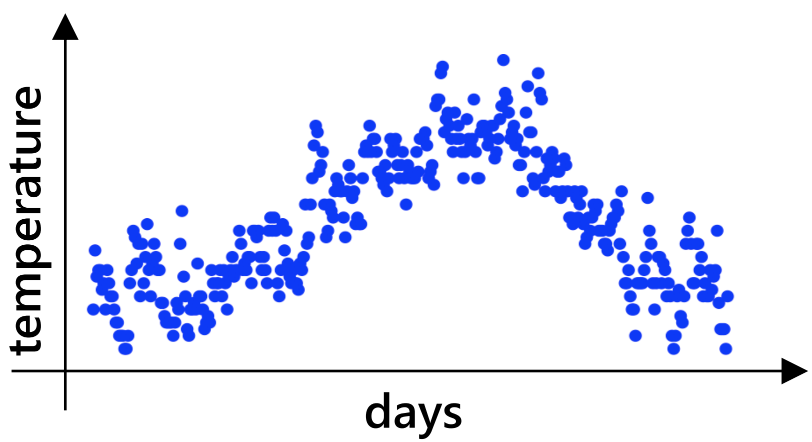
Convolution

- **Definition**
  - An operation on two functions of a real-valued argument
    - Expression: $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$

- **Example**
  - Suppose we are tracking the location of a plane with a laser sensor
    - The sensor provides a single output $x(t)$: position of the plane at time t
    - Both x and t: real-valued

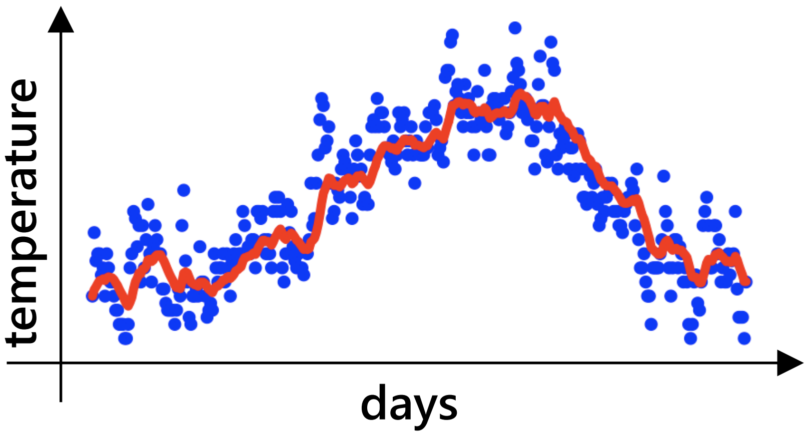
Convolution

- **Exponentially Weighted Moving Average (EWMA)**
  - A method of making the effect of old data exponentially decay when calculating the moving average of data
  - Expression: $V_t = \beta \times V_{t-1} + (1-\beta) \times \theta_t$, $V_{t-1} = \beta \times V_{t-2} + (1-\beta) \times \theta_{t-1}$
    - β: hyperparameter (0~1), V: trends, θ: new data, t: time
    - $V_t = \beta \times (\beta \times V_{t-2} + (1-\beta) \times \theta_{t-1}) + (1-\beta) \times \theta_t$ > exponential
    - $V \approx \frac{1}{1-\beta}$

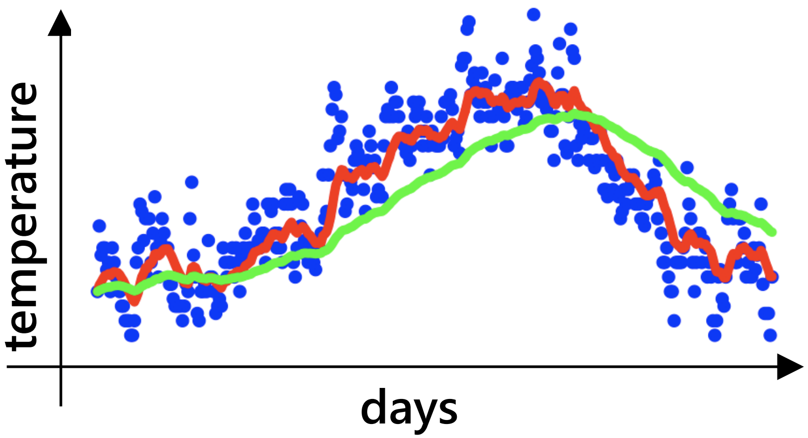
- **Example (Temperature of London)**
  - 

Convolution

- **Example**
  - $\beta = 0.9$ (red line, for 10 days)

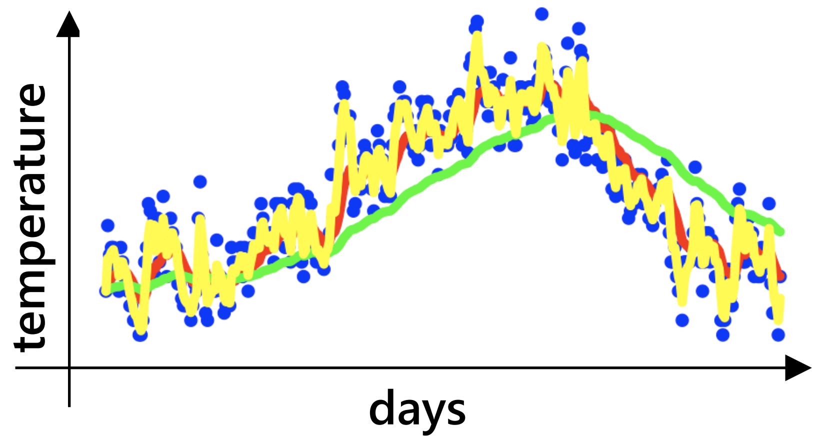


  - $\beta = 0.98$ (green line, for 50 days)



Convolution

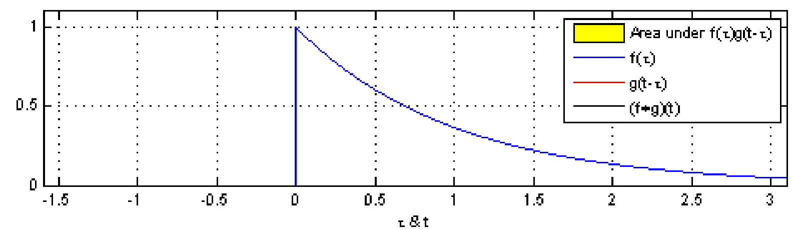
- **Example**
  - $\beta = 0.5$ (yellow line, for 2 days)



- **Parameter β**
  - if close to 1, more weight to former data
  - if close to 0, more weight to now data

Convolution

- **Example (Convolution)**
  - Suppose the sensor is noisy; objective > to get less noisy estimate of x
    - We average together several measurements
    - More recent measurements are more relevant (have more weight)
  - Define a weighting function $w(a)$
    - To give more weight to recent measurements
    - a: the age of a measurement
    - w is learnable (parameter)



  - Apply weighted average operation at every moment
    - A new function: s providing a smoothed estimate of x
      - $s(t) = \sum_{-\infty}^{\infty} x(a)w(t-a) \triangleq (x * w)(t)$

Convolution

- **Arguments**

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$

  - First argument (function x): *input*
  - Second argument (function w): *kernel* or *filter*
  - Output (function s): *feature map* or *activation map*

- **2D Convolution**
  - Suppose we use 2D image I as input, 2D kernel K
    - $S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$; $(I * K)(i,j) = (K * I)(i,j)$

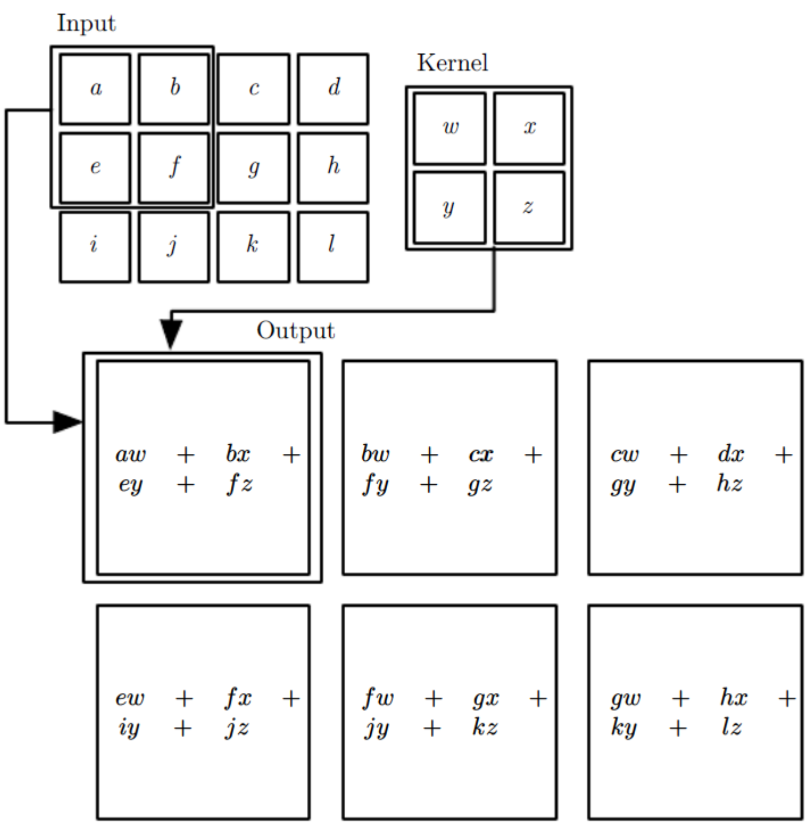
- **Convolution means "*Convolution with Kernel Flipping*"**

Cross Correlation

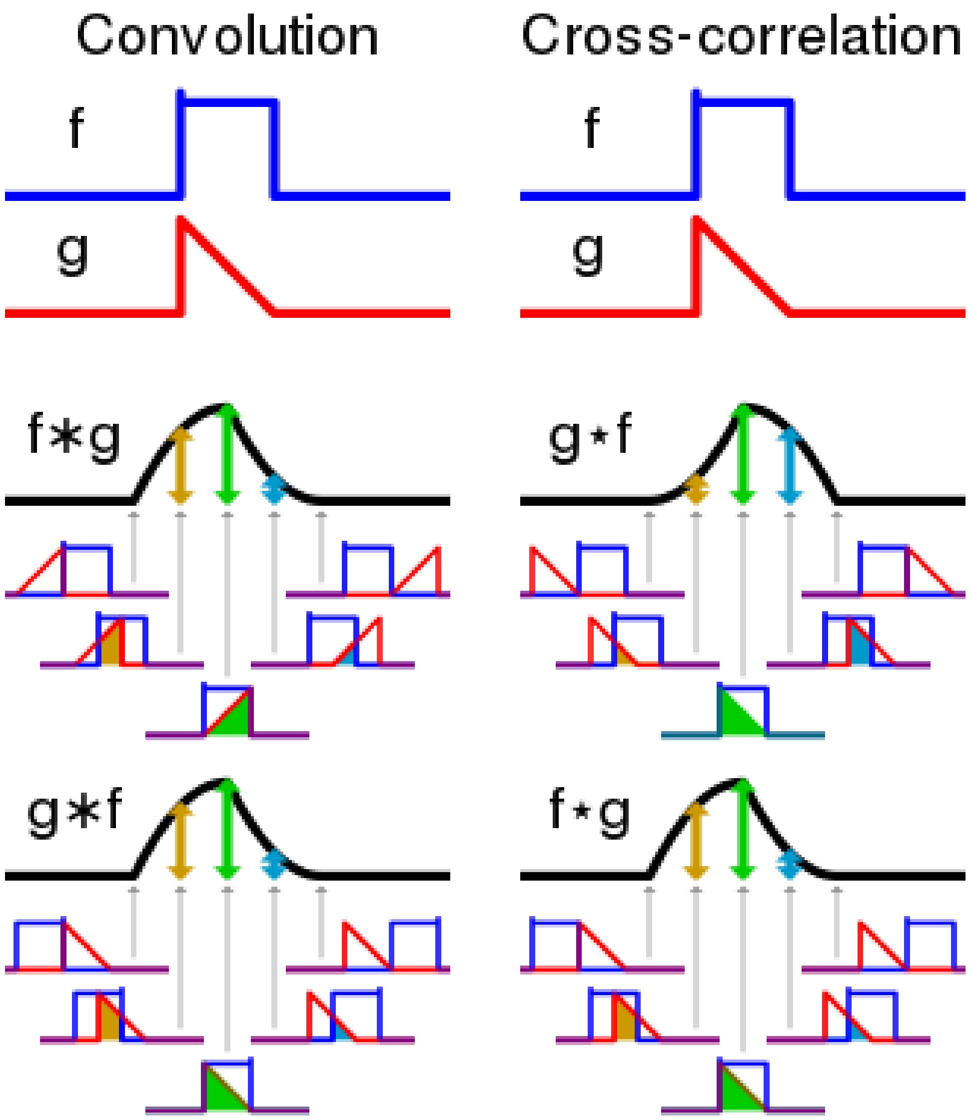
Definition

- The same as convolution, but *without flipping the kernel*
- Expression: $S(i,j) = (K * I)(i,j) = \sum_m \sum_n K(m,n) I(i + m, j + n)$

2D Cross Correlation (element-wise; dot product)



Convolution vs. Cross Correlation



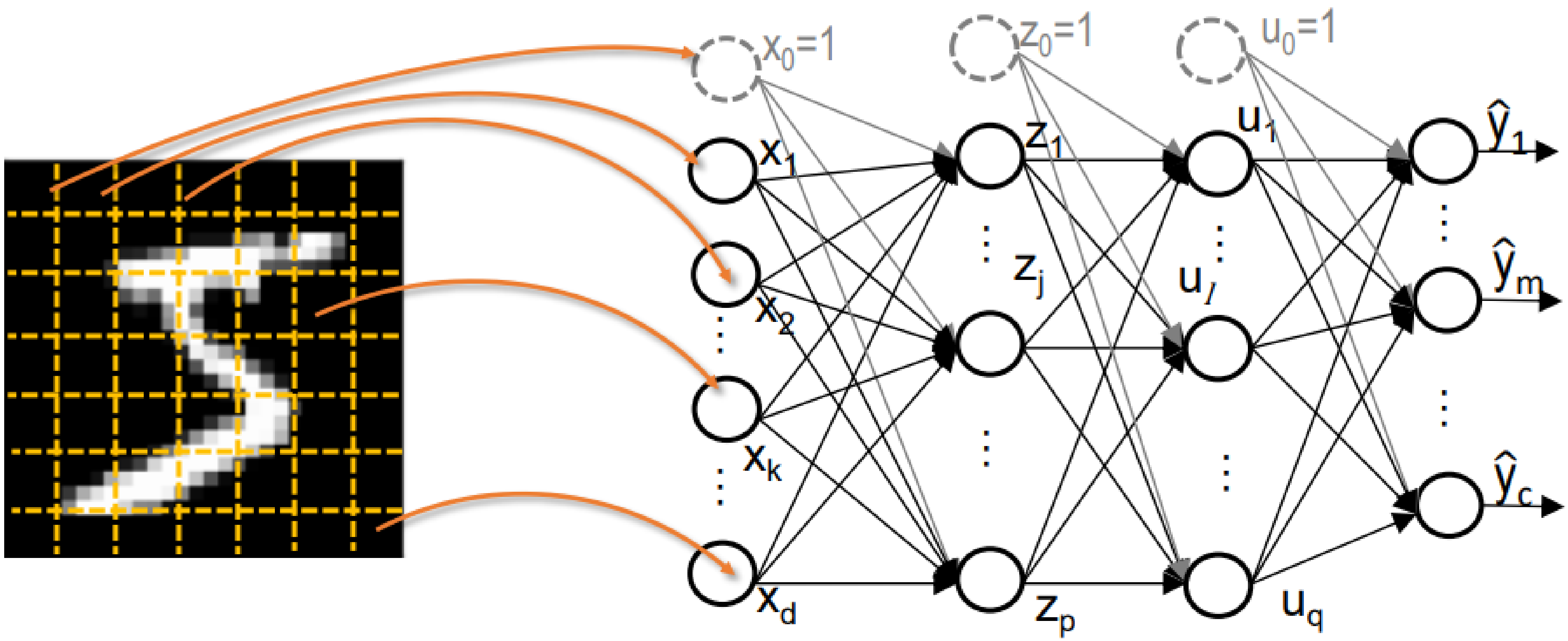
Convolution vs. Cross Correlation

- **Convolution means..**
  - To find the output of a system of impulse response for an input
    > used to calculate the output of a system

- **Cross correlation means..**
  - A process to find the degree of similarity between two signals

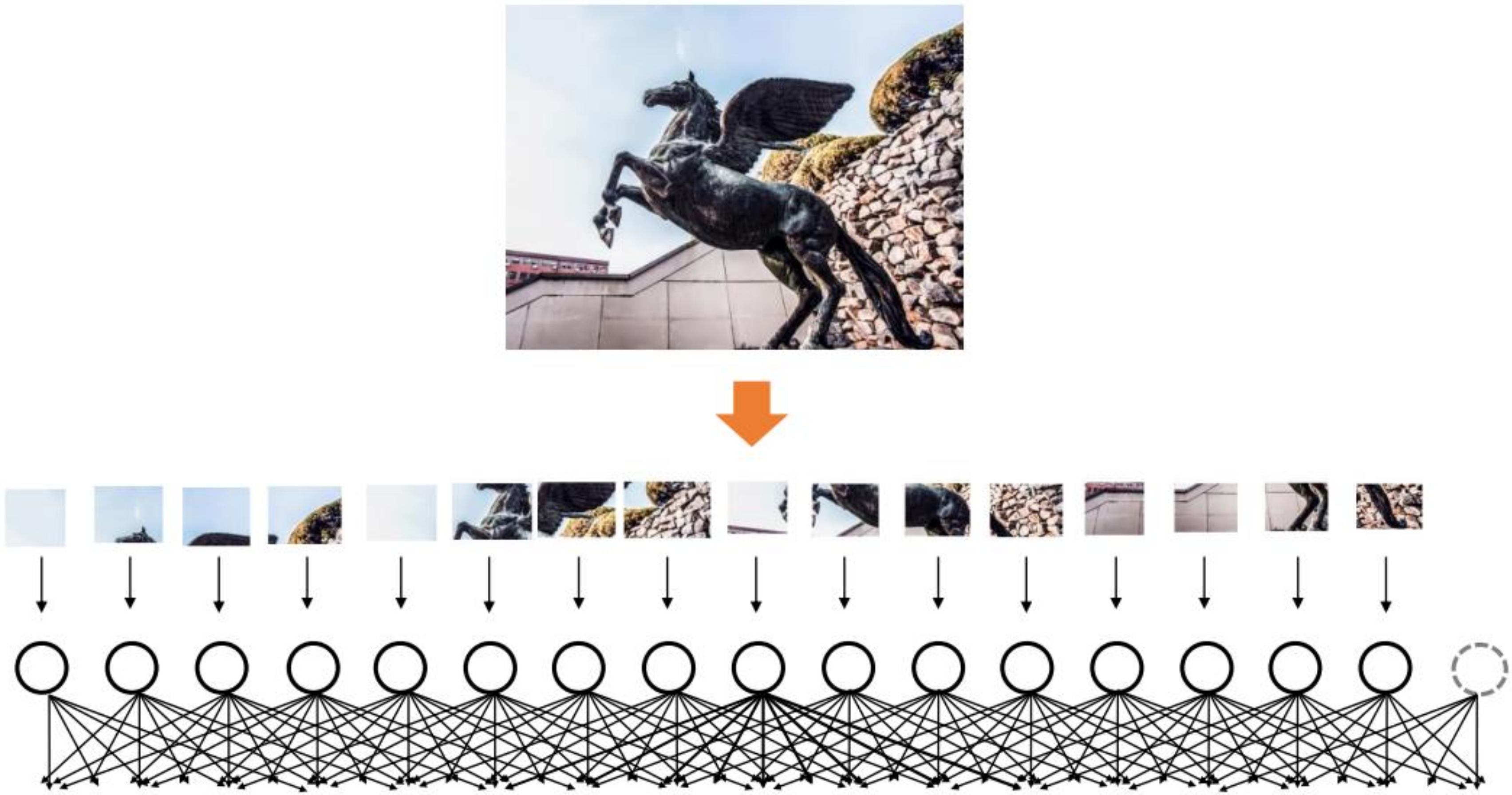
Fully-Connected Neural Network

- **Problems**
  - Large computation
  - Overfitting risks
  - Data loss problem (Flatten)



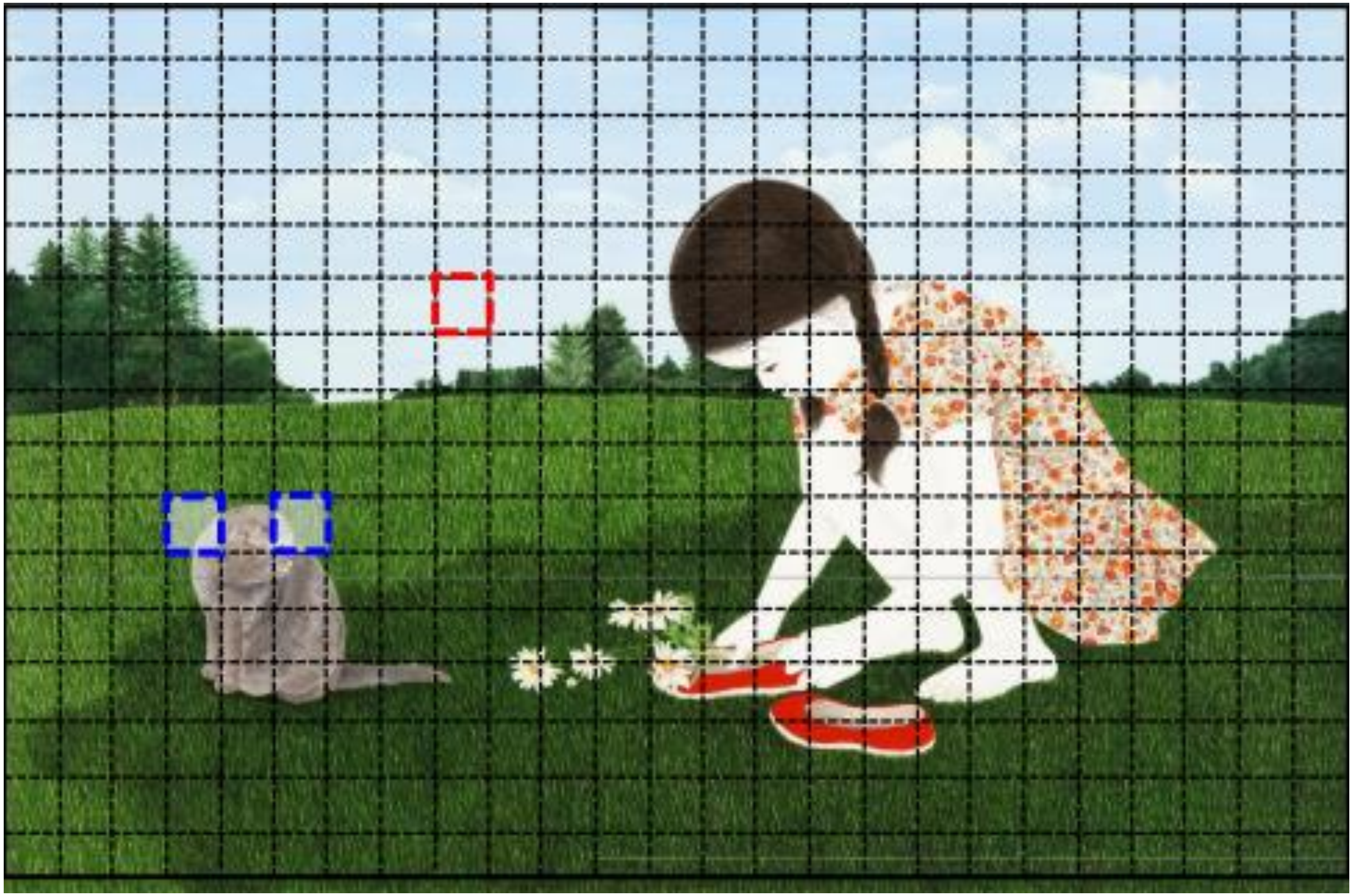
Fully-Connected Neural Network

- **Problems**
  - Data loss problem (Flatten)



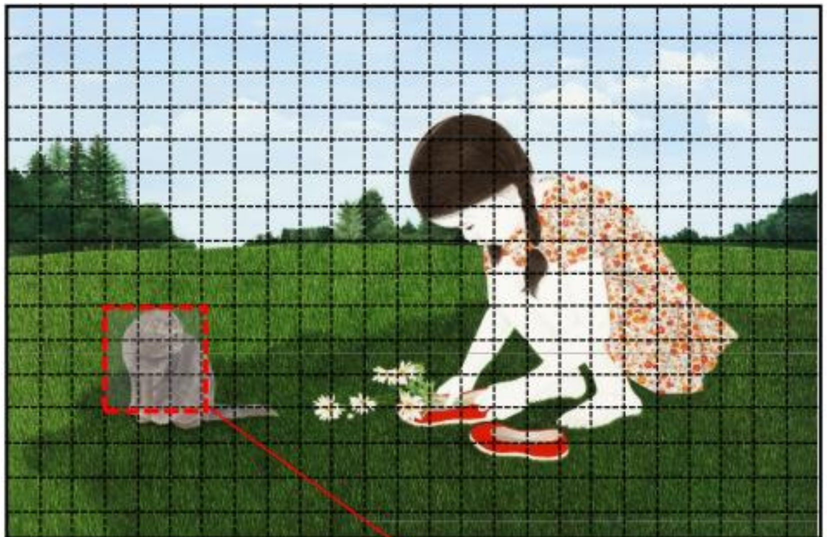
Convolutional Neural Network

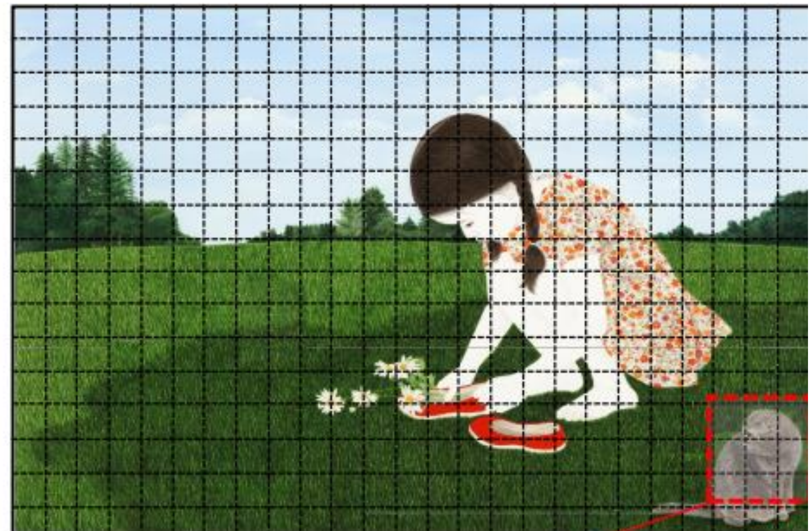
- **Motivated from Partially Connection**
  - Due to *locality*



Convolutional Neural Network

- **Translation equivariance**





Should have the same calculation results, like

I see hair

I see an eye

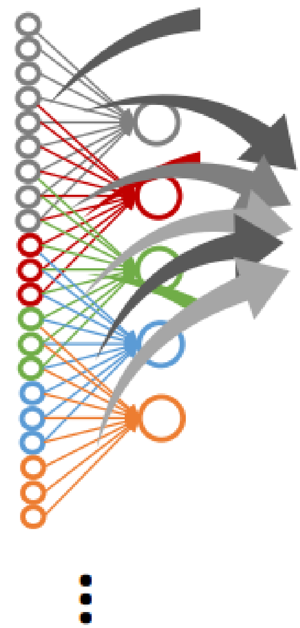
I see something…. cute…

How can we Consider this respect into MLP classification?

Convolutional Neural Network

- **Weight sharing**

-6	5	4	3	-4	7	3	7	0	-1
8	2	-1	-7	6	9	9	10	3	6
9	-10	-7	6	4	-2	-8	6	-1	-8
8	6	-9	8	9	6	-5	4	8	9
4	-4	6	9	-3	2	-8	-8	-7	7
-7	7	6	4	10	7	0	-3	-8	7
-2	-1	-3	-10	-4	-2	-10	7	-8	3
6	3	-5	-2	-5	-5	3	-7	3	-8
0	-8	5	9	10	4	-9	0	0	-5
4	3	-2	4	7	5	2	8	-3	8



1	**4**	**-1**
-3	**2**	**5**
-2	**3**	**-4**

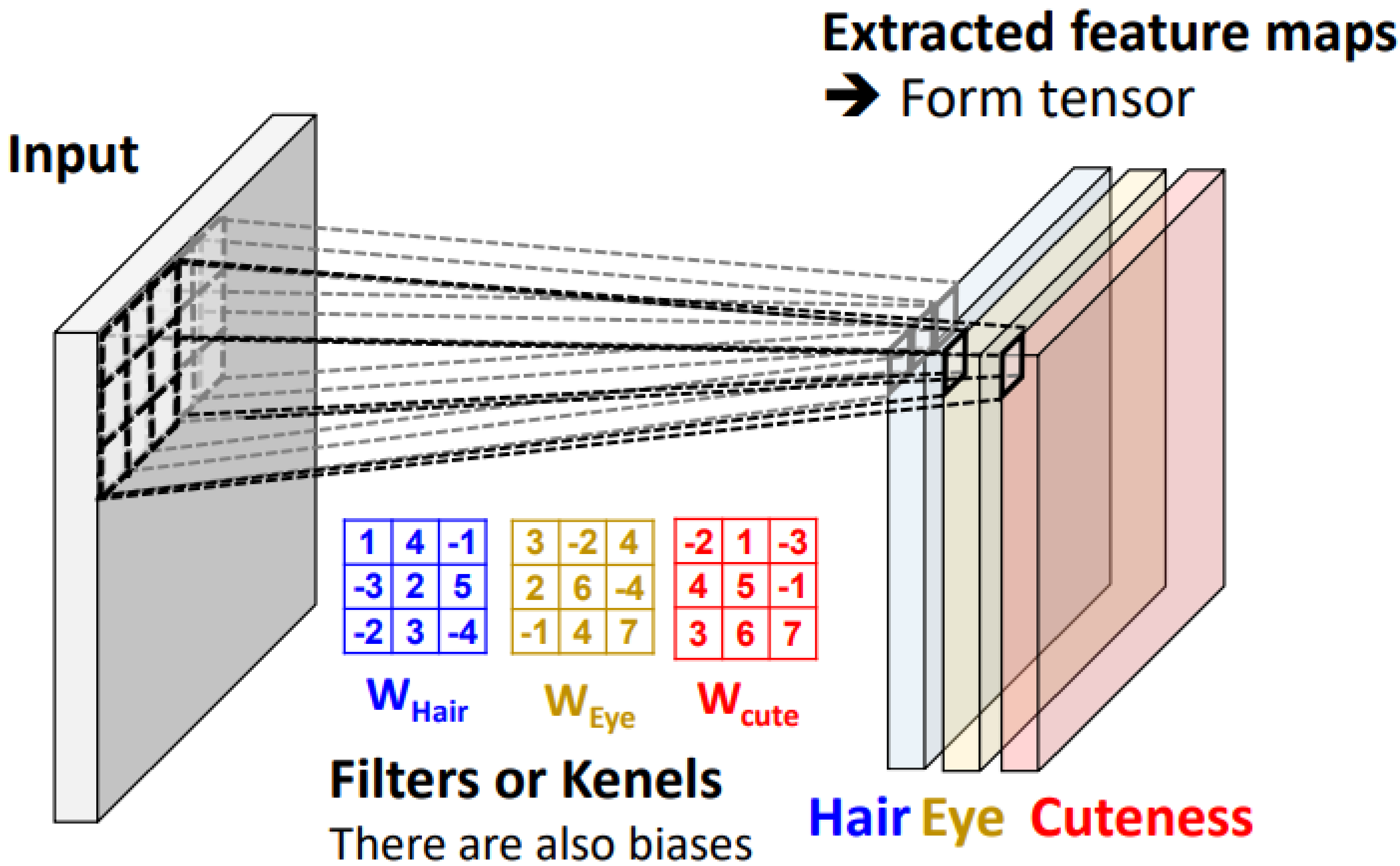
Feature

"Extracting hair"

➔ **Parameter Sharing or Weight Sharing**

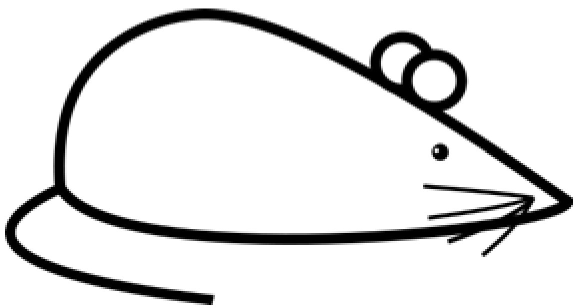
Convolutional Neural Network

- **Mechanism**

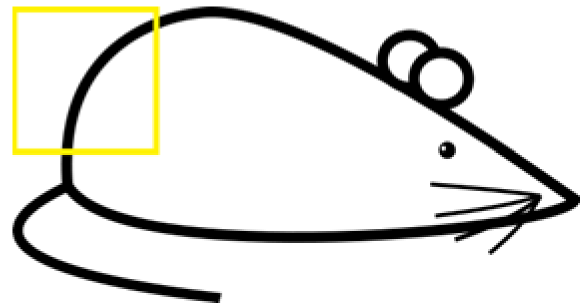


Details of CNN

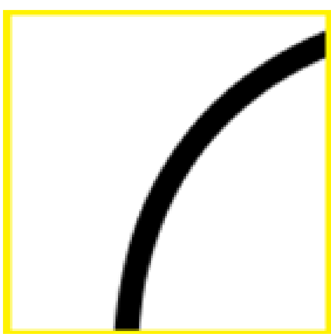
- **Filter (=Kernel)**



Original image



Visualization of the filter on the image



Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

$*$

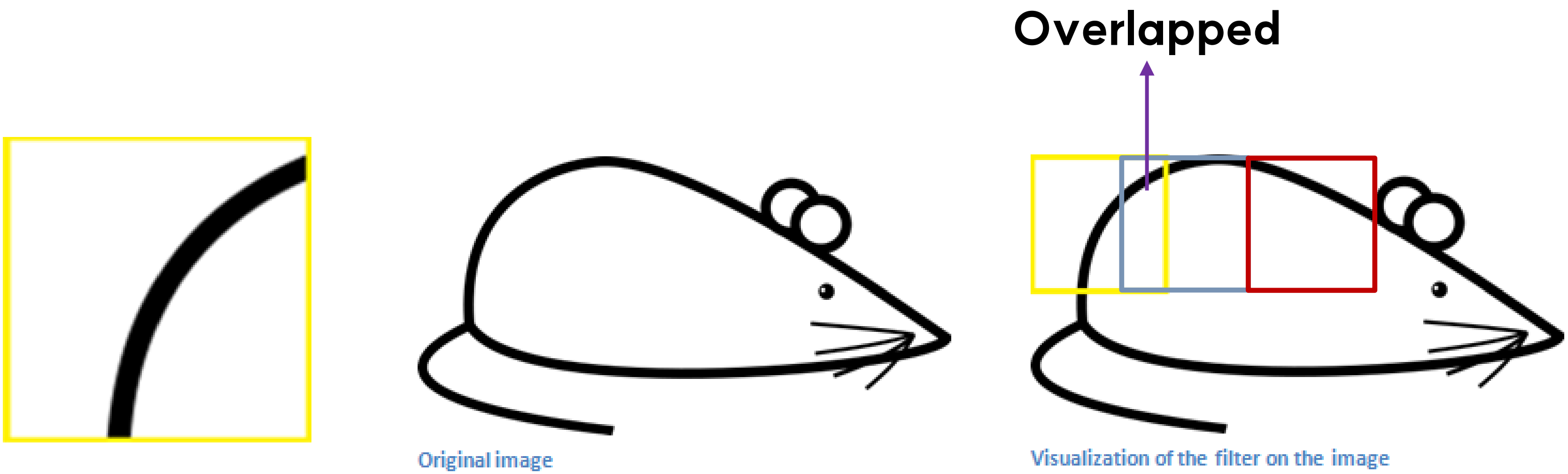
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

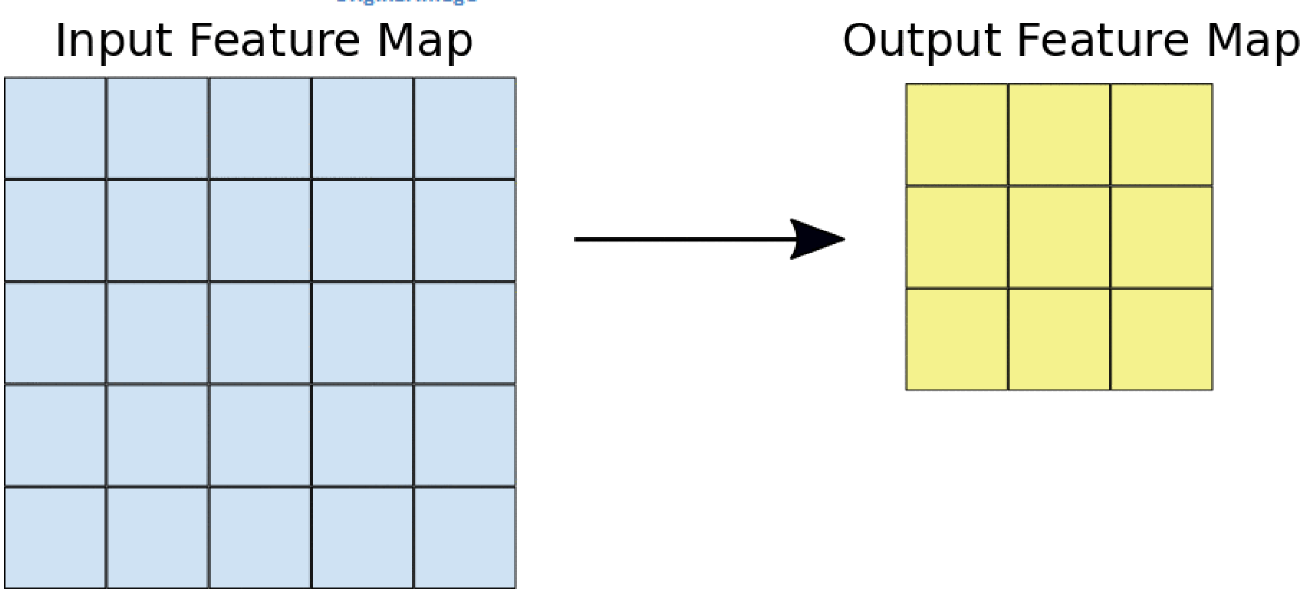
Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

Details of CNN

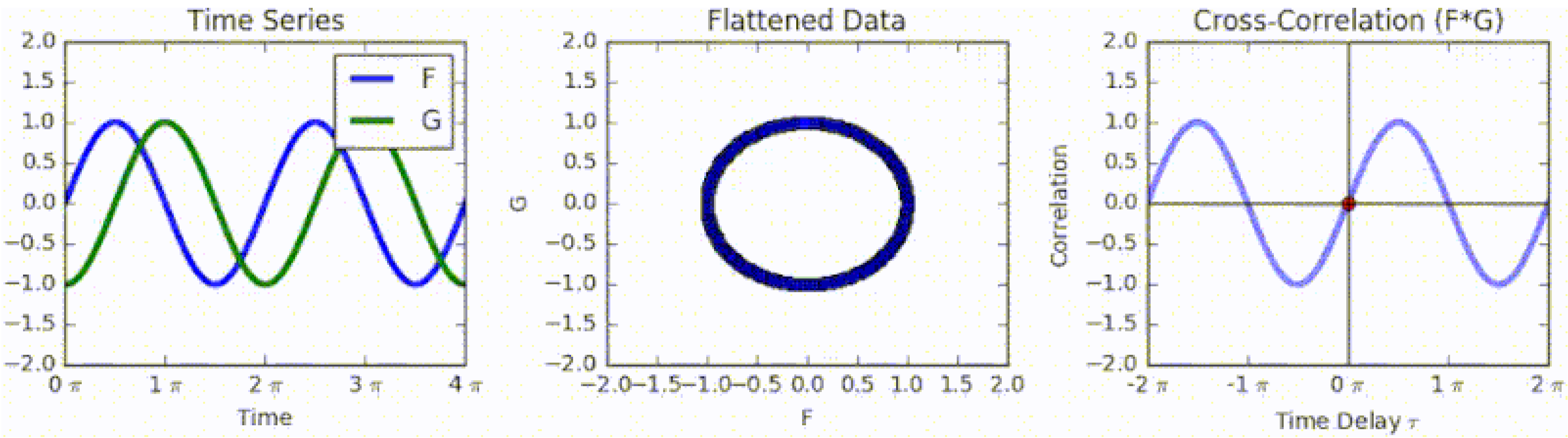
- **Stride**





Details of CNN

- **Stride**
  - Recap: cross correlation expr (1D)
    - $(f \star g)(x) = \int_{-\infty}^{\infty} f(x)g(x+t)dx$
    - t = stride



Details of CNN

- **Padding**
  - To prevent data loss when convolution
  - To preserve edge data information
  - Usually pads to 0

$$\begin{bmatrix} 0 & 1 & 7 & 5 \\ 5 & 5 & 6 & 6 \\ 5 & 3 & 3 & 0 \\ 1 & 1 & 1 & 2 \end{bmatrix} \circledast \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 41 & 33 \\ 25 & 23 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 7 & 5 & 0 \\ 0 & 5 & 5 & 6 & 6 & 0 \\ 0 & 5 & 3 & 3 & 0 & 0 \\ 0 & 1 & 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \circledast \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 26 & 42 & 55 & 35 \\ 34 & 41 & 33 & 28 \\ 18 & 25 & 23 & 14 \\ 3 & 9 & 8 & 8 \end{bmatrix}$$

Details of CNN

▪ Pooling

- To extract summary statistics
- Accelerate the speed, reduce the noise effect, overfitting
- Max pooling, Average pooling, Weighted average pooling, …
- in CNN, usually uses Max pooling
- Pooling does not increase parameters (just simple calculation)

0	5	4	3	0	0
8	2	0	0	6	9
9	0	0	6	4	0
8	6	0	8	9	6
4	0	6	9	0	2
0	7	6	4	10	7

s=2

Max pooling

8	4	9
9	8	9
7	9	10

Details of CNN

- **Output tensor size of convolution layer**
  - Notation
    - O: size(width) of output image
    - I: size(width) of input image
    - K: size(width) of kernels
    - N: number of kernels
    - S: stride of the convolution operation
    - P: padding size
  - O(width of output image)
    - $O = \frac{I-K+2P}{S} + 1$
  - Channels of output image == number of filters (N)

Details of CNN

- **Output tensor size of max pooling layer**
  - Notation
    - O: size(width) of output image
    - I: size(width) of input image
    - S: stride of the convolution operation
    - P_s: pooling size
  - O(width of output image)
    - $O = \frac{I - P_s}{S} + 1$
  - Channels of output image == number of input images

Details of CNN

- **Number of parameters on convolution layer**
  - Notation
    - W_c: number of weights of the convolution layer
    - B_c: number of biases of the convolution layer
    - P_c: number of parameters of the convolution layer
    - K: size(width) of kernels used in the convolution layer
    - N: number of kernels
    - C: number of channels of the input image
  - $W_c = K^2 \times C \times N$, $B_c = N$, $P_c = W_c + B_c$
  - Depth of filters == input image channels

GNU

경상국립대학교

Gyeongsang National University

Improving lives through learning

IDEALAB