2022 Enterprise Boost Program

AI/ML Basic 05

08. 이미지를 위한 신경망

CNN으로 패션상품을 분류해보자!

전현상

Solutions Architect

2022.06.28

AI/ML Basic Tracks

AI/ML Basic 01
Week 4
- 나의 첫 머신 러닝
- 데이터 다루기

AI/ML Basic 02
Week 5
- 회귀 알고리즘과 모델 규제
- 다양한 분류 알고리즘

AI/ML Basic 03
Week 6
- 트리 알고리즘
- 비지도 학습

AI/ML Basic 04
Week 7
- 딥러닝을 시작합니다.

AI/ML Basic 05
Week 8
- 이미지를 위한 신경망

AI/ML Basic 06
Week 9
- 텍스트를 위한 신경망

AI/ML Basic 07
Week 10
- 파트너 실전 PoC 사례 공유

Table of contents

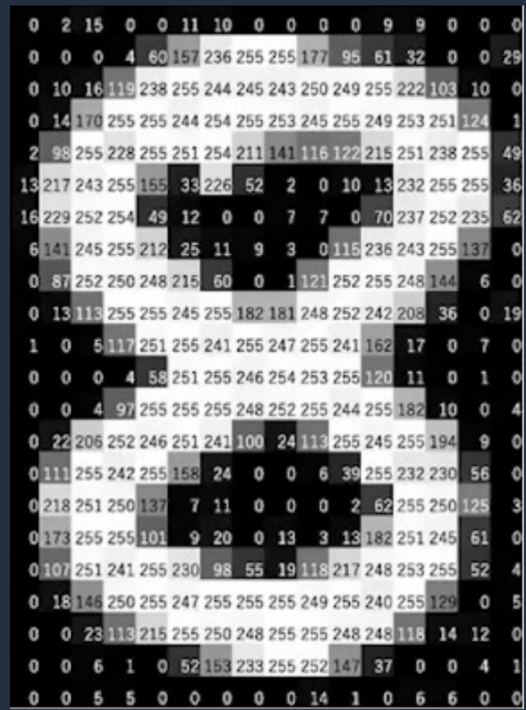
Convolutional Neural Networks

How images are stored in a Computer?

Grayscale image

height

width



0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

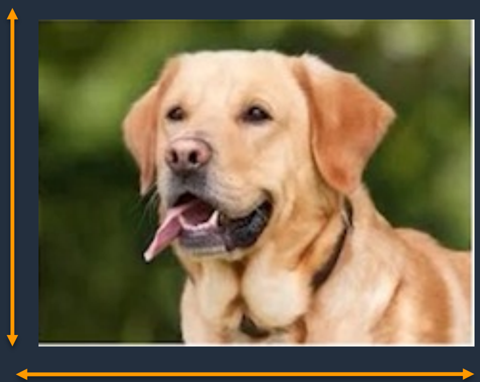
(height x width) pixels

Figures copyright , MNIST Databse

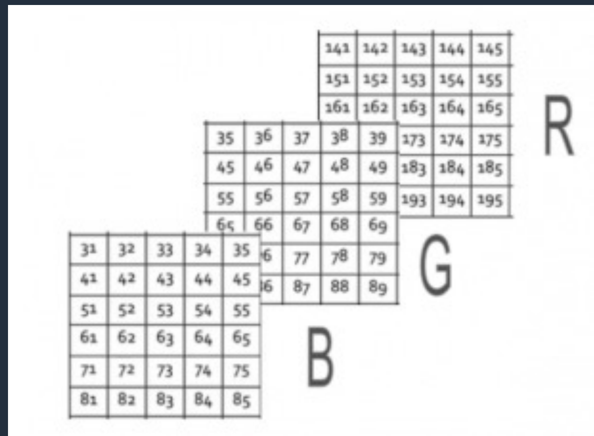
How images are stored in a Computer?

Colored image

height



width

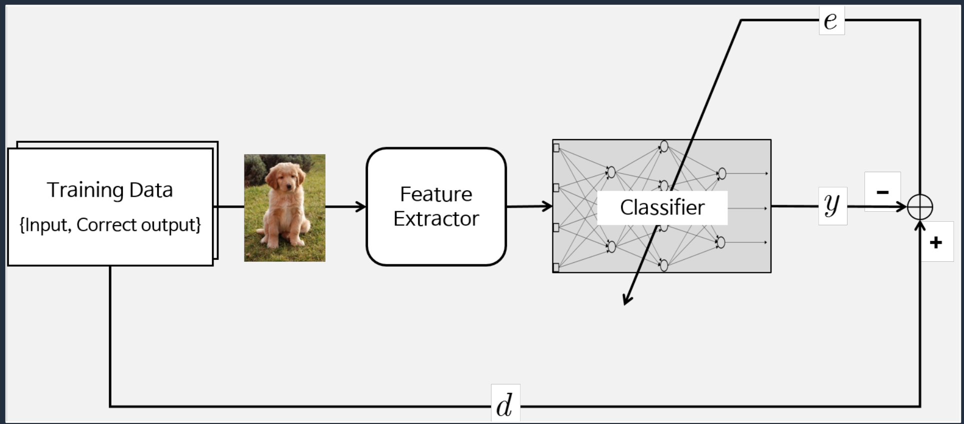


(height x width) x 3 pixels

Figures copyright , Himanshi, https://www.analyticsvidhya.com/blog/2021/03/grayscale-and-rgb-format-for-storing-images/

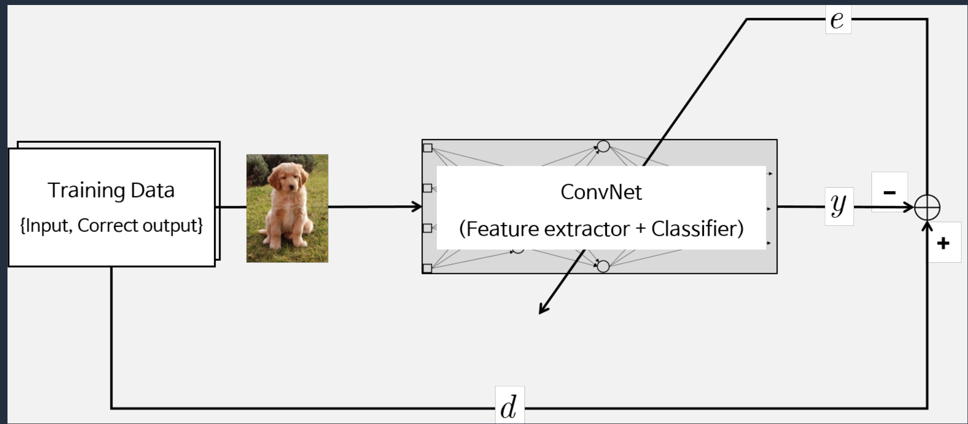
Image recognition

Perceptron neural network



Feature extraction + Classification network

Convolutional neural network

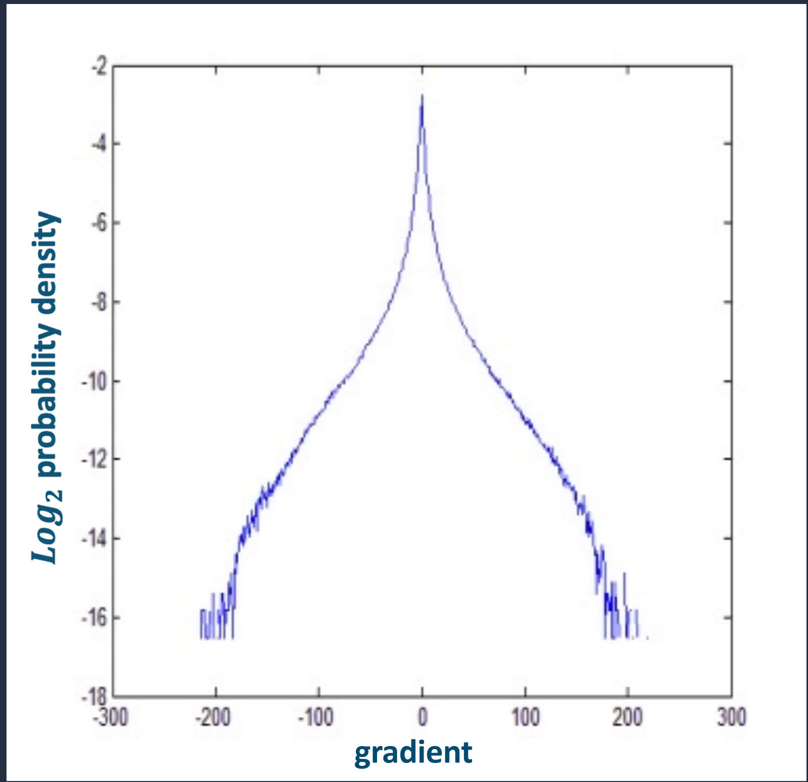


Feature extraction network + Classification network

Figures copyright , 김성필, 머신러닝에서 컨벌루션 신경망까지, 딥러닝 첫걸음

Statistics of image gradients

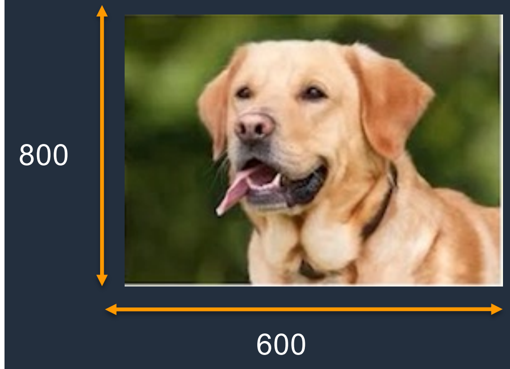
■ ■ ■

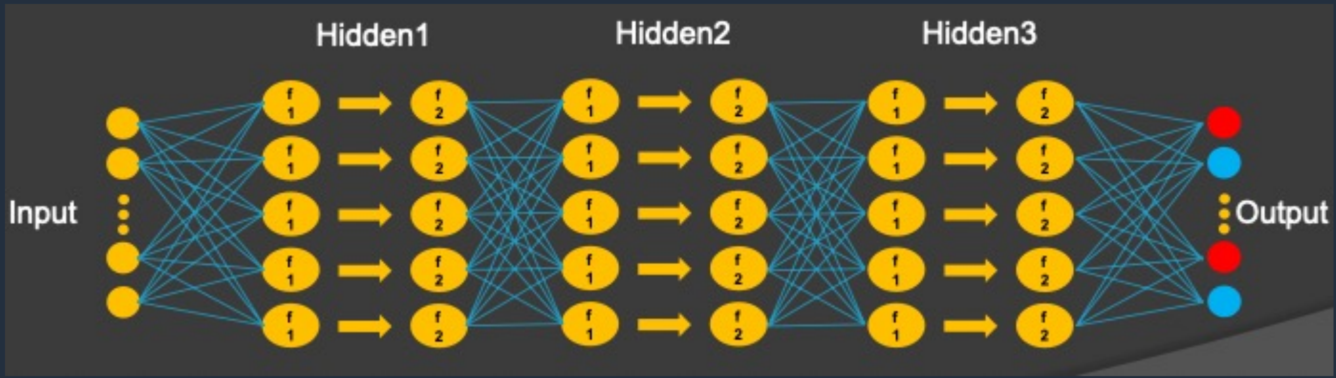


Logarithmic density of gradients from 10 natural images.

DNN Image recognition

Colored image





- 2차원 input data
- Pixel array : (800, 600, 3)
- Bytes : 800 x 600 x 3 = 1,440,000 bytes

- Dense Layer에서는 모든 입력이 Hidden Layer에 연결(fully conecteted layer)
- Hidden Layer가 늘어날수록 back propagation에서 weight, bias update 계산량이 기하급수로 늘어남
- Image 의 평탄한 영역은 학습에 불필요한 영역, 특징이 잘 드러나지 않음
- 특징 추출기는 사람이 직접 설계

- 학습에서 모든 data(pixels)를 각각 node에 연결하는 것보다 효율적인 방법을 찾아보자.

Figures copyright , Deep Neural Net by Daebang Kim.

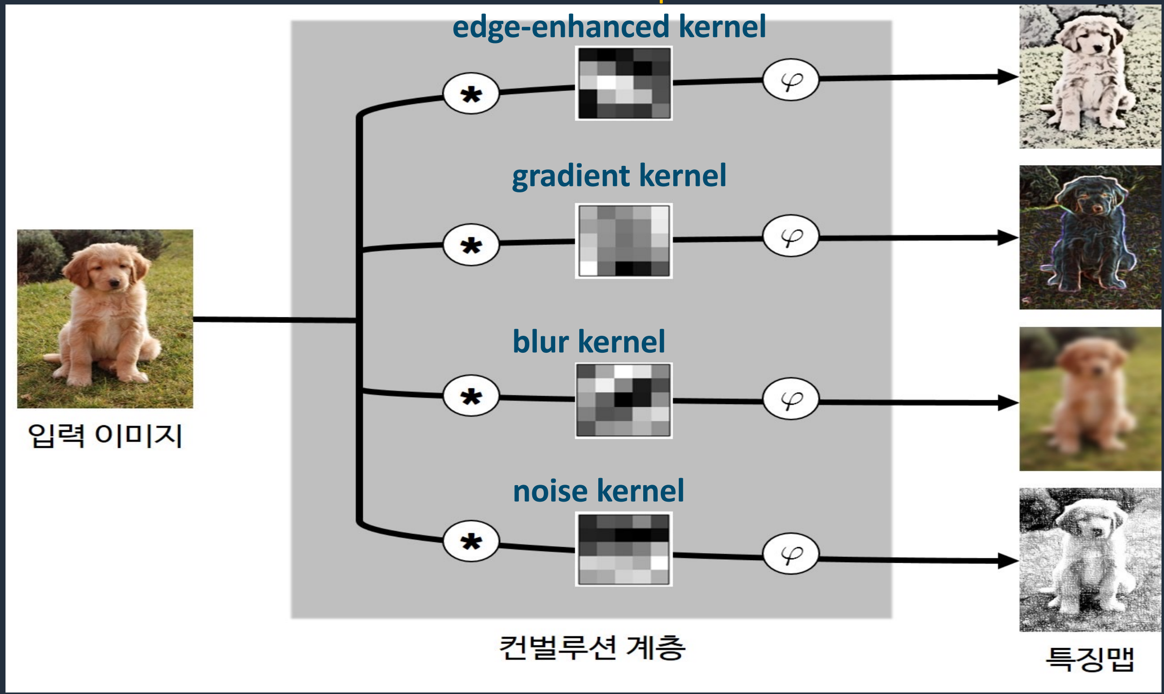
Figures copyright , Himanshi, https://www.analyticsvidhya.com/blog/2021/03/grayscale-and-rgb-format-for-storing-images/

CNN Image recognition

Convolution Layer

- 특징점(feature point) 추출filer(kernel)을 사용하여 입력데이터를 처리
- kernel의 개수가 특징맵(feature map)개수

Kernel example



Figures copyright , 김성필, 머신러닝에서 컨벌루션 신경망까지, 딥러닝 첫걸음

CNN operation

Convolution?

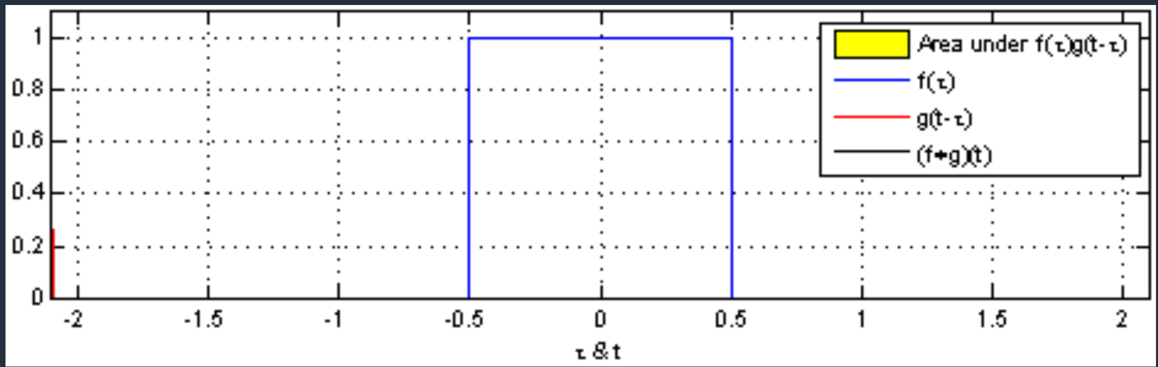
두 신호함수의 적분(중첩)

Convolution

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

Cross-Correlation

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t+\tau)d\tau$$



Figures copyright , Convolution, Wikipedia.

2D convolution for image filtering

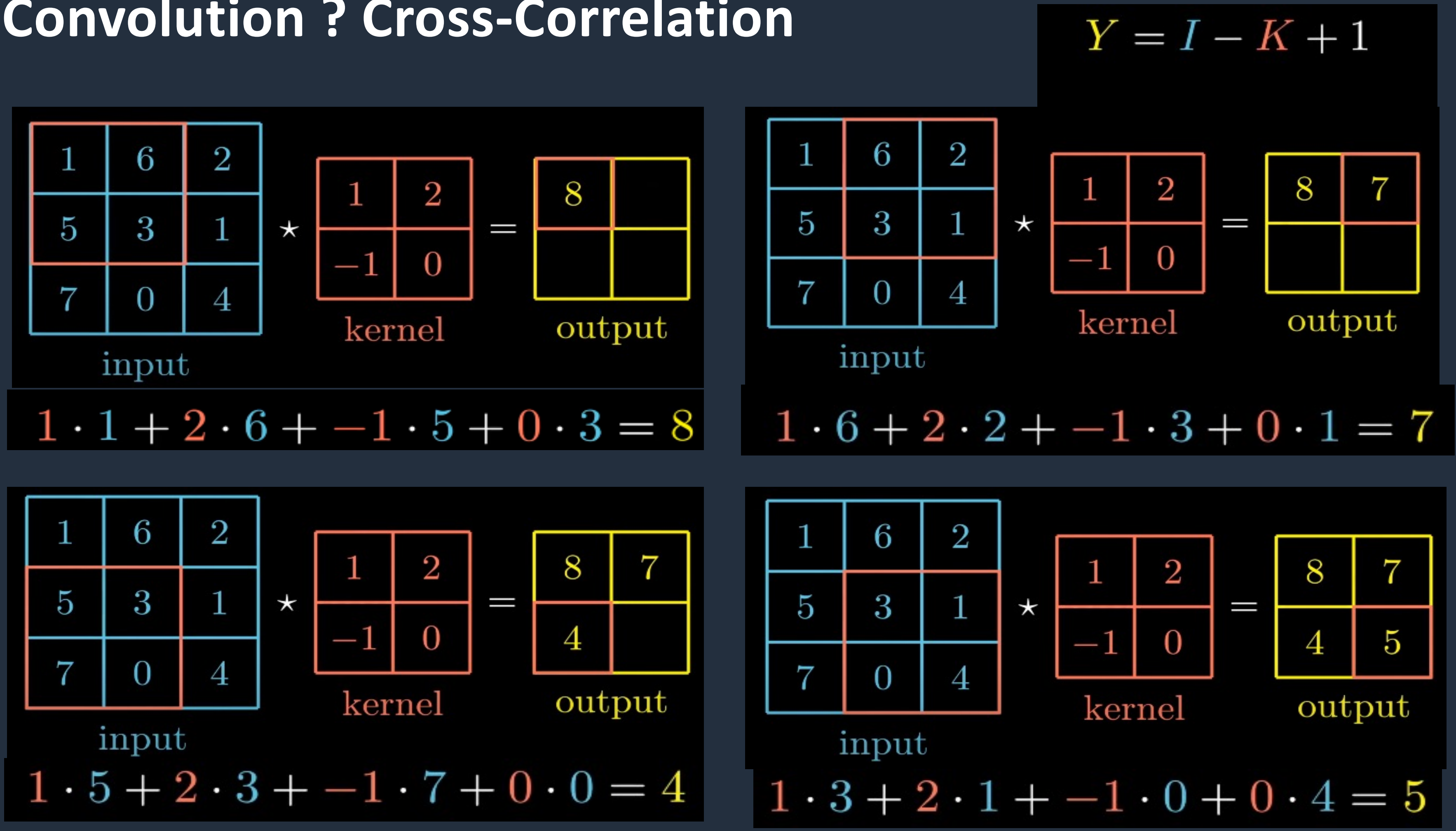
Convolution

$$f(x, y) * k(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\tau_1, \tau_2) k(x - \tau_1, y - \tau_2)\, d\tau_1\, d\tau_2$$

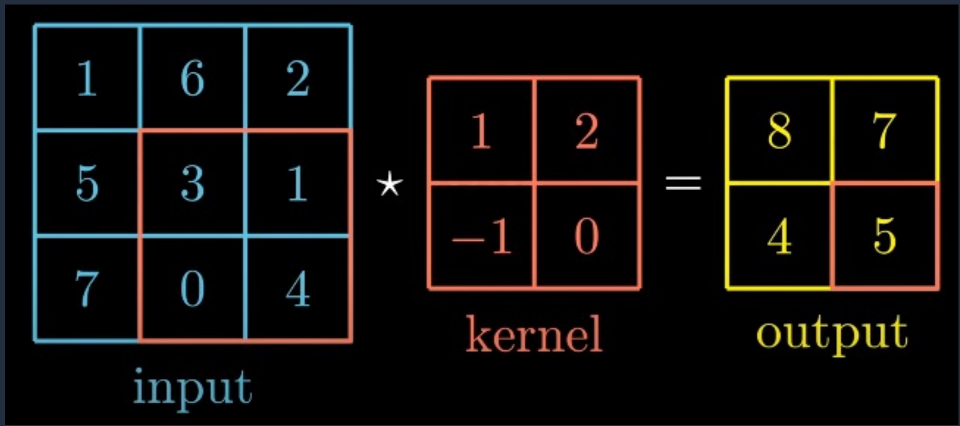
$$g = f * k \qquad g[i, j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} k[u, v] f[i - u, j - v]$$

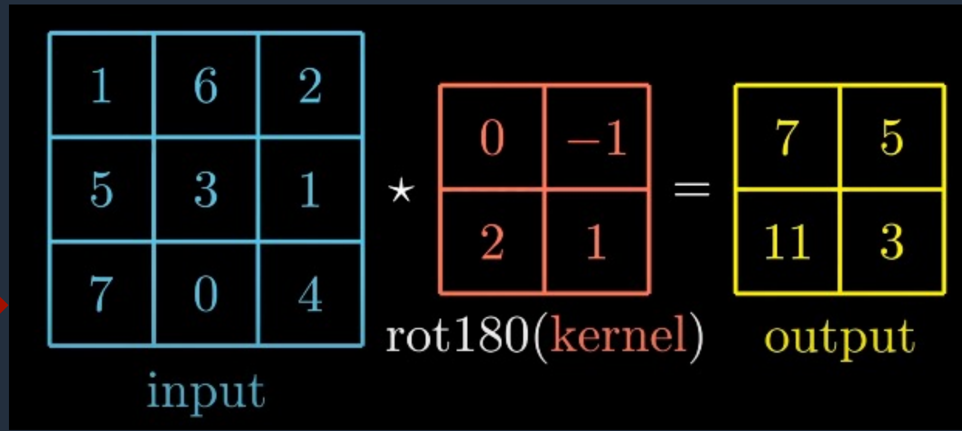
Figures copyright , Convolutional Neural Network Visualization by Otavio Good, https://www.youtube.com/watch?v=f0t-OCG79-U

2D Convolution ? Cross-Correlation



Convolution & Cross Correlation





Cross-Correlation

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t+\tau)d\tau$$

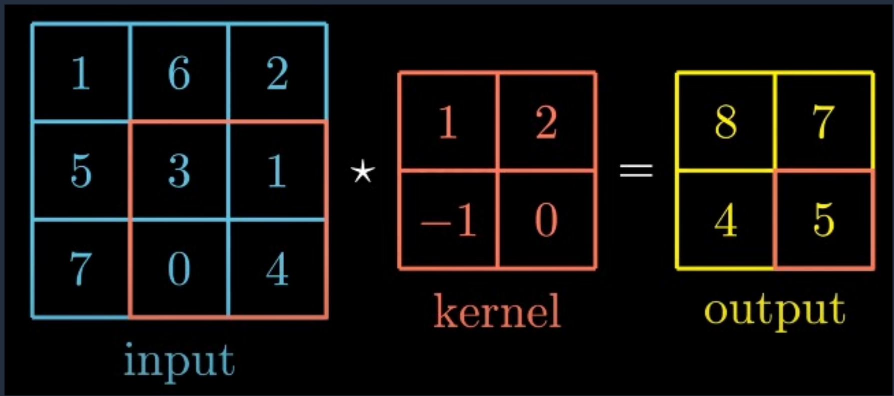
$$I * K = I \star rot180(K)$$

Convolution

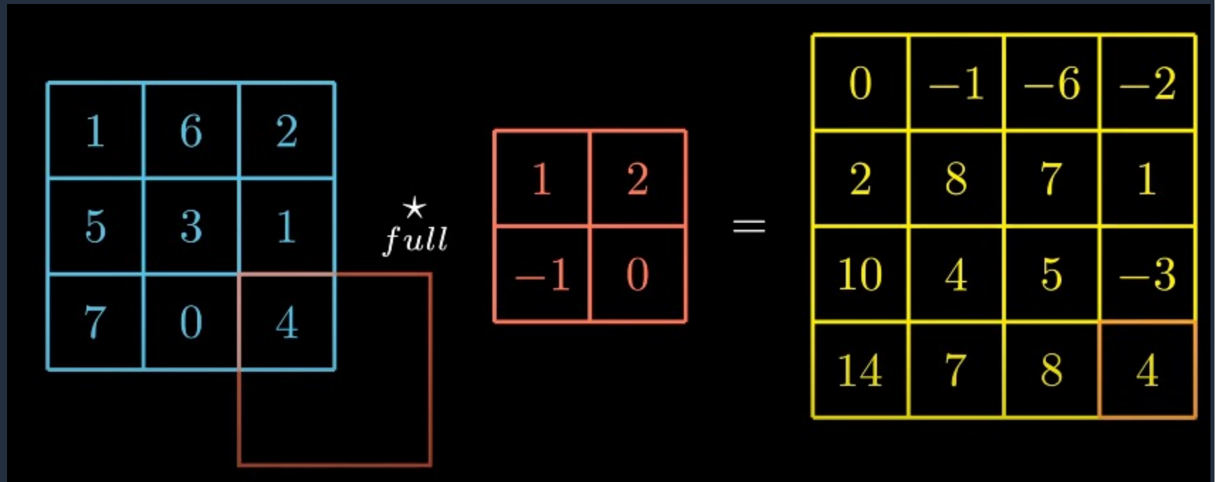
$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

Padding

- valid



- full cross correlation(same)



Keras functional parameter : padding (default = 'valid')

```
model.add(keras.layers.Conv2D(32, kernel_size=3, activation='relu',
                              padding='same', input_shape=(28,28,1)))
```

- valid : kernel에 매칭되지 않는 부분 skip, (0,0)시작
- same : kernel에 매칭 되지 않는 부분 0 padding

Convolution Layer

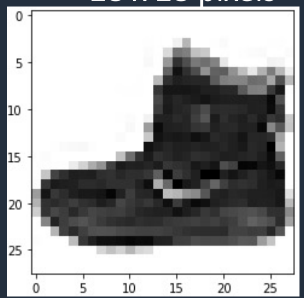
CNN for Image

Tensorflow keras CNN example

```
model.add(keras.layers.Conv2D(32, kernel_size=3, activation='relu', padding='same', input_shape=(28,28,1)))
```

Input image

- 28 x 28 pixels

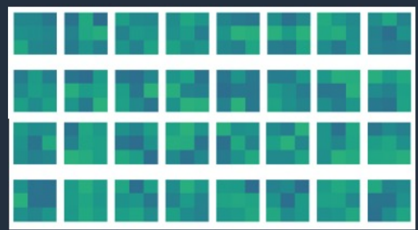


28

28

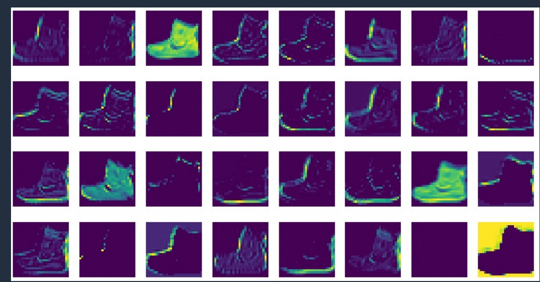
Convolutional Layer

- Weight kernel : 32개
- 3x3 pixels



Output

- Feature map: 32개
- 28x28 pixels

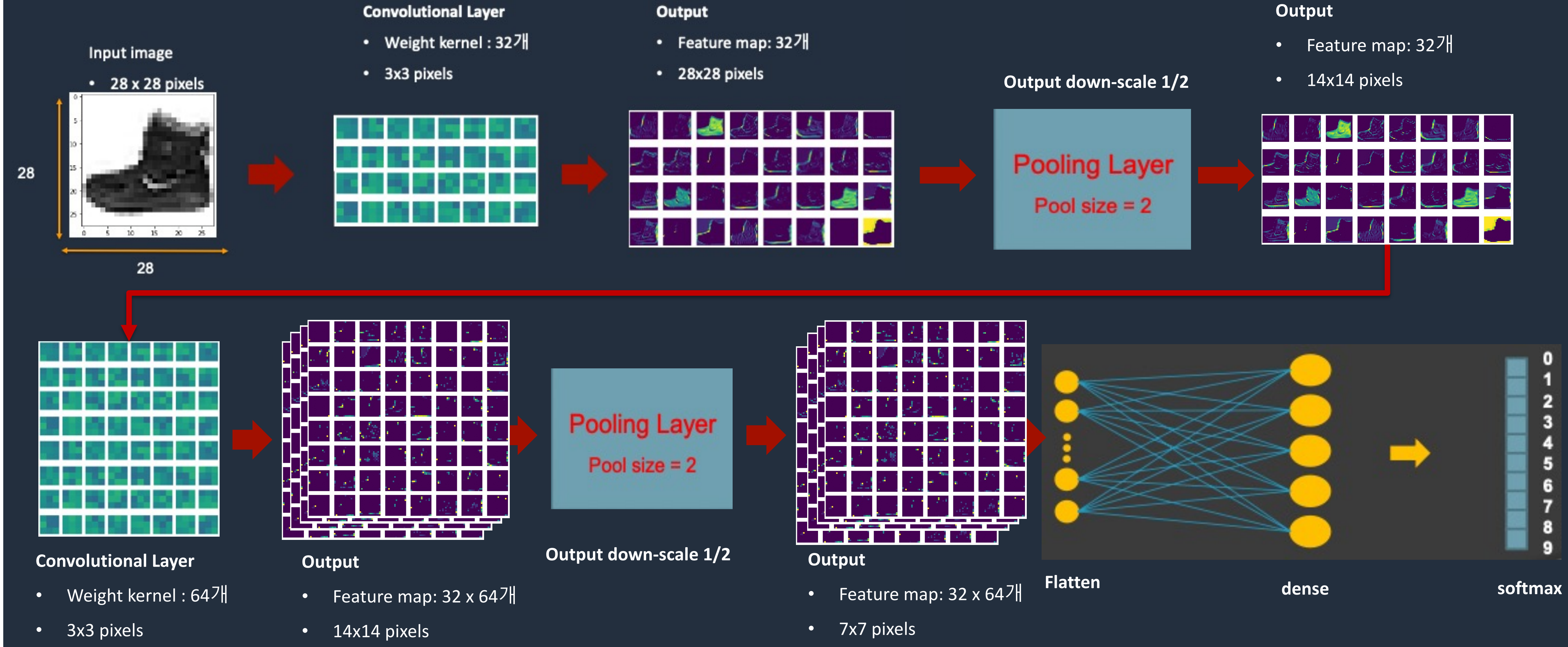


Weight Parameters :

- Input_channel * w_kernel * h_kernel* + biases
- 1 x 3 x 3 x 32 +32 = 320

CNN for Image

Tensorflow keras CNN example

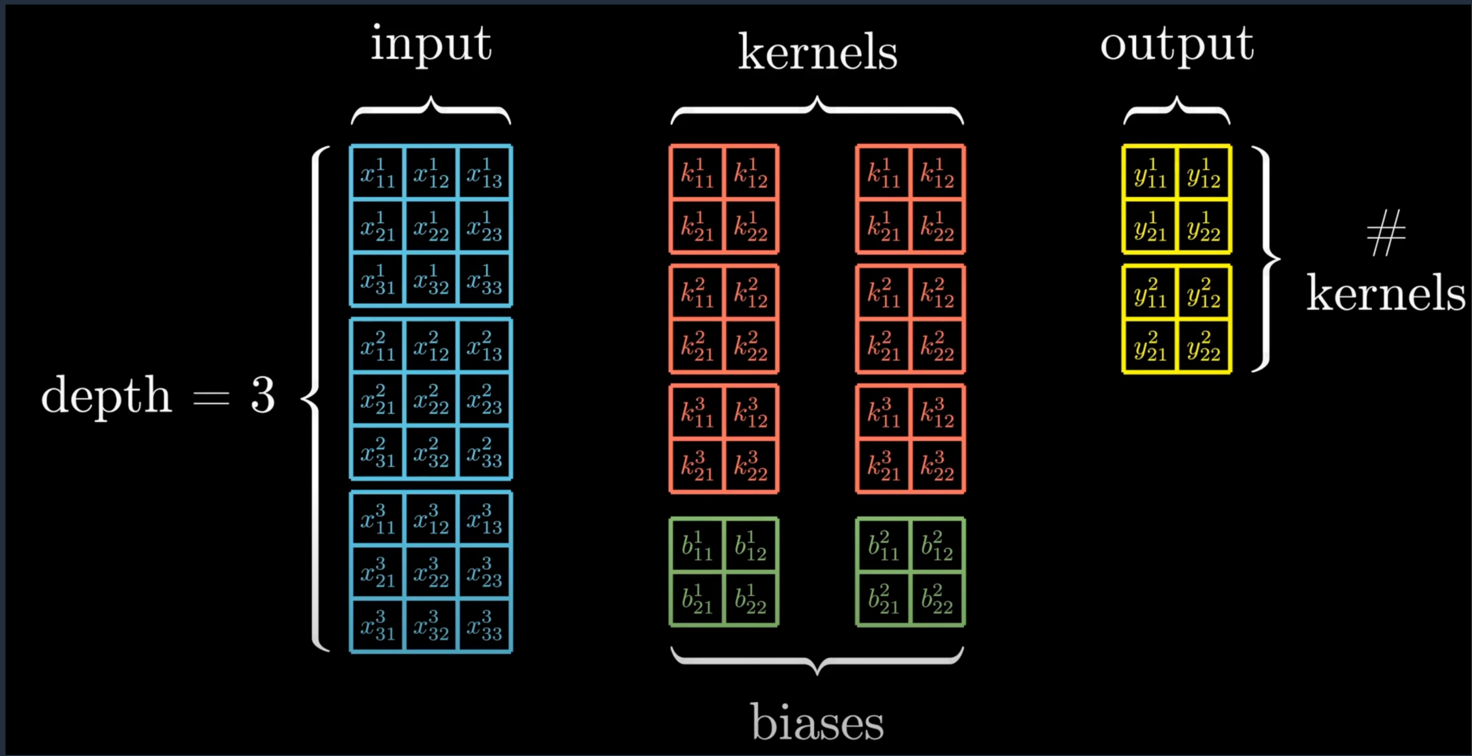


Figures copyright ,fashion mnist, https://github.com/zalandoresearch/fashion-mnist
Figures copyright ,박해선, 한빛미디어, 혼자공부하는 머신러닝 딥러닝

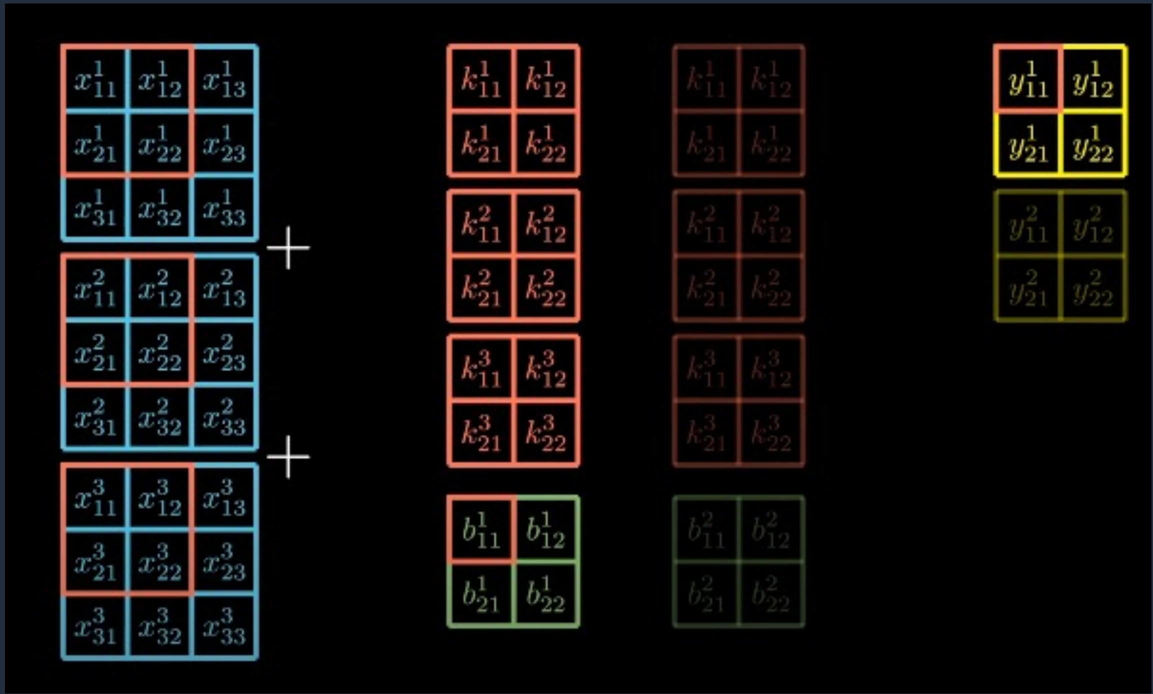
Visualization of Multi-Layer Perceptron & CNN

Figures copyright , Denis Dmitriev, Neural Network 3D Simulation, https://www.youtube.com/watch?v=3JQ3hYko51Y&t=98s

Forward propagation



Forward propagation



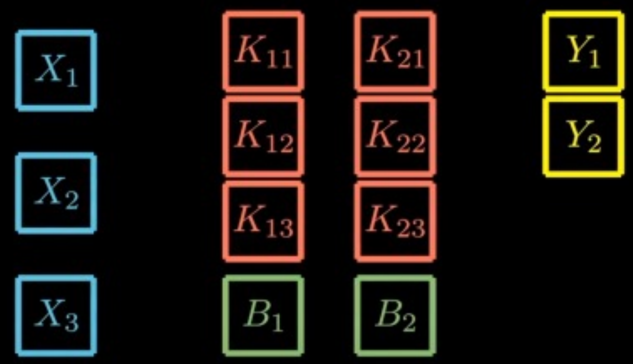
Forward propagation

$$\boxed{Y_1} = \boxed{B_1} + \boxed{X_1} \star \boxed{K_{11}} + \boxed{X_2} \star \boxed{K_{12}} + \boxed{X_3} \star \boxed{K_{13}}$$

$$\boxed{Y_2} = \boxed{B_2} + \boxed{X_1} \star \boxed{K_{21}} + \boxed{X_2} \star \boxed{K_{22}} + \boxed{X_3} \star \boxed{K_{23}}$$

$$\vdots$$

$$\boxed{Y_d} = \boxed{B_d} + \boxed{X_1} \star \boxed{K_{d1}} + \boxed{X_2} \star \boxed{K_{d2}} + \boxed{X_3} \star \boxed{K_{d3}}$$



Figures copyright, The Independent Code, Convolutional Neural Network from Scratch | Mathematics & Python Code

Forward propagation

$$Y_i = B_i + \sum_{j=1}^{n} X_j \star K_{ij}, \quad i = 1 \ldots d$$

$$\boxed{Y_1} = \boxed{B_1} + \boxed{X_1} \star \boxed{K_{11}} + \cdots + \boxed{X_n} \star \boxed{K_{1n}}$$

$$\boxed{Y_2} = \boxed{B_2} + \boxed{X_1} \star \boxed{K_{21}} + \cdots + \boxed{X_n} \star \boxed{K_{2n}}$$

$$\vdots$$

$$\boxed{Y_d} = \boxed{B_d} + \boxed{X_1} \star \boxed{K_{d1}} + \cdots + \boxed{X_n} \star \boxed{K_{dn}}$$

Figures copyright, The Independent Code, Convolutional Neural Network from Scratch | Mathematics & Python Code

Forward propagation

$$Y_i = B_i + \sum_{j=1}^{n} X_j \star K_{ij}, \quad i = 1 \dots d$$

$$\begin{aligned} Y_1 &= B_1 + X_1 \star K_{11} + \cdots + X_n \star K_{1n} \\ Y_2 &= B_2 + X_1 \star K_{21} + \cdots + X_n \star K_{2n} \\ &\vdots \\ Y_d &= B_d + X_1 \star K_{d1} + \cdots + X_n \star K_{dn} \end{aligned}$$

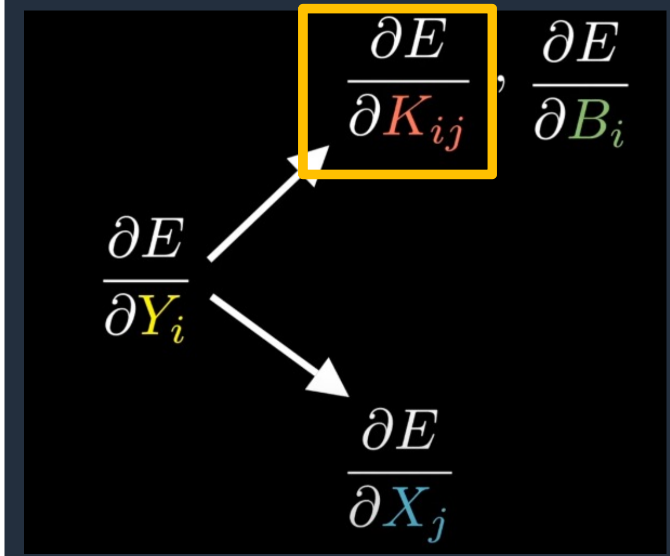
$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_d \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_d \end{bmatrix} + \begin{bmatrix} K_{11} & K_{12} & \dots & K1n \\ K_{21} & K_{22} & \dots & K2n \\ \vdots & \vdots & \vdots & \vdots \\ K_{d1} & K_{d2} & \dots & Kdn \end{bmatrix} \cdot|\star \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}$$

$$Y = B + K \cdot|\star X$$

Convolution Layer

Backward Operation

Backward Kernel



$$Y_i = B_i + \sum_{j=1}^{n} X_j \star K_{ij}$$

Forward propagation

$$\boxed{Y_i = B_i + X_1 \star K_{i1}} + \cdots + X_n \star K_{in}$$

Simplified example

$$\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} + \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} \star \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

$$\begin{cases} y_{11} = b_{11} + k_{11}x_{11} + k_{12}x_{12} + k_{21}x_{21} + k_{22}x_{22} \\ y_{12} = b_{12} + k_{11}x_{12} + k_{12}x_{13} + k_{21}x_{22} + k_{22}x_{23} \\ y_{21} = b_{21} + k_{11}x_{21} + k_{12}x_{22} + k_{21}x_{31} + k_{22}x_{32} \\ y_{22} = b_{22} + k_{11}x_{22} + k_{12}x_{23} + k_{21}x_{32} + k_{22}x_{33} \end{cases}$$

Backward Kernel

$$\begin{cases} y_{11} = b_{11} + \boxed{k_{11}x_{11}} + k_{12}x_{12} + k_{21}x_{21} + k_{22}x_{22} \\ y_{12} = b_{12} + \boxed{k_{11}x_{12}} + k_{12}x_{13} + k_{21}x_{22} + k_{22}x_{23} \\ y_{21} = b_{21} + \boxed{k_{11}x_{21}} + k_{12}x_{22} + k_{21}x_{31} + k_{22}x_{32} \\ y_{22} = b_{22} + \boxed{k_{11}x_{22}} + k_{12}x_{23} + k_{21}x_{32} + k_{22}x_{33} \end{cases}$$

$$\frac{\partial E}{\partial Y} = \begin{bmatrix} \frac{\partial E}{\partial y_{11}} & \frac{\partial E}{\partial y_{12}} \\ \frac{\partial E}{\partial y_{21}} & \frac{\partial E}{\partial y_{22}} \end{bmatrix} \longrightarrow \begin{bmatrix} \frac{\partial E}{\partial k_{11}} & \frac{\partial E}{\partial k_{12}} \\ \frac{\partial E}{\partial k_{21}} & \frac{\partial E}{\partial k_{22}} \end{bmatrix}$$

$$\frac{\partial E}{\partial k_{11}} = \frac{\partial E}{\partial y_{11}}\frac{\partial y_{11}}{\partial k_{11}} + \frac{\partial E}{\partial y_{12}}\frac{\partial y_{12}}{\partial k_{11}} + \frac{\partial E}{\partial y_{21}}\frac{\partial y_{21}}{\partial k_{11}} + \frac{\partial E}{\partial y_{22}}\frac{\partial y_{22}}{\partial k_{11}}$$

$$\frac{\partial E}{\partial k_{11}} = \frac{\partial E}{\partial y_{11}}\underbrace{\frac{\partial y_{11}}{\partial k_{11}}}_{x_{11}} + \frac{\partial E}{\partial y_{12}}\underbrace{\frac{\partial y_{12}}{\partial k_{11}}}_{x_{12}} + \frac{\partial E}{\partial y_{21}}\underbrace{\frac{\partial y_{21}}{\partial k_{11}}}_{x_{21}} + \frac{\partial E}{\partial y_{22}}\underbrace{\frac{\partial y_{22}}{\partial k_{11}}}_{x_{22}}$$

$$\frac{\partial E}{\partial k_{11}} = \frac{\partial E}{\partial y_{11}}x_{11} + \frac{\partial E}{\partial y_{12}}x_{12} + \frac{\partial E}{\partial y_{21}}x_{21} + \frac{\partial E}{\partial y_{22}}x_{22}$$

$$\begin{aligned} \frac{\partial E}{\partial k_{11}} &= \frac{\partial E}{\partial y_{11}}x_{11} + \frac{\partial E}{\partial y_{12}}x_{12} + \frac{\partial E}{\partial y_{21}}x_{21} + \frac{\partial E}{\partial y_{22}}x_{22} \\ \frac{\partial E}{\partial k_{12}} &= \frac{\partial E}{\partial y_{11}}x_{12} + \frac{\partial E}{\partial y_{12}}x_{13} + \frac{\partial E}{\partial y_{21}}x_{22} + \frac{\partial E}{\partial y_{22}}x_{23} \\ \frac{\partial E}{\partial k_{21}} &= \frac{\partial E}{\partial y_{11}}x_{21} + \frac{\partial E}{\partial y_{12}}x_{22} + \frac{\partial E}{\partial y_{21}}x_{31} + \frac{\partial E}{\partial y_{22}}x_{32} \\ \frac{\partial E}{\partial k_{22}} &= \frac{\partial E}{\partial y_{11}}x_{22} + \frac{\partial E}{\partial y_{12}}x_{23} + \frac{\partial E}{\partial y_{21}}x_{32} + \frac{\partial E}{\partial y_{22}}x_{33} \end{aligned}$$

Figures copyright, The Independent Code, Convolutional Neural Network from Scratch | Mathematics & Python Code

Backward Kernel

$$\frac{\partial E}{\partial k_{11}} = \frac{\partial E}{\partial y_{11}} x_{11} + \frac{\partial E}{\partial y_{12}} x_{12} + \frac{\partial E}{\partial y_{21}} x_{21} + \frac{\partial E}{\partial y_{22}} x_{22}$$

$$\frac{\partial E}{\partial k_{12}} = \frac{\partial E}{\partial y_{11}} x_{12} + \frac{\partial E}{\partial y_{12}} x_{13} + \frac{\partial E}{\partial y_{21}} x_{22} + \frac{\partial E}{\partial y_{22}} x_{23}$$

$$\frac{\partial E}{\partial k_{21}} = \frac{\partial E}{\partial y_{11}} x_{21} + \frac{\partial E}{\partial y_{12}} x_{22} + \frac{\partial E}{\partial y_{21}} x_{31} + \frac{\partial E}{\partial y_{22}} x_{32}$$

$$\frac{\partial E}{\partial k_{22}} = \frac{\partial E}{\partial y_{11}} x_{22} + \frac{\partial E}{\partial y_{12}} x_{23} + \frac{\partial E}{\partial y_{21}} x_{32} + \frac{\partial E}{\partial y_{22}} x_{33}$$

$$\begin{bmatrix} \frac{\partial E}{\partial k_{11}} & \frac{\partial E}{\partial k_{12}} \\ \frac{\partial E}{\partial k_{21}} & \frac{\partial E}{\partial k_{22}} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} \star \begin{bmatrix} \frac{\partial E}{\partial y_{11}} & \frac{\partial E}{\partial y_{12}} \\ \frac{\partial E}{\partial y_{21}} & \frac{\partial E}{\partial y_{22}} \end{bmatrix}$$

$$\boxed{Y = B + X \star K \Rightarrow \frac{\partial E}{\partial K} = X \star \frac{\partial E}{\partial Y}}$$

Simplified version

$$\frac{\partial E}{\partial K} = X \star \frac{\partial E}{\partial Y}$$

Actual version

$$Y_i = B_i + \sum_{j=1}^{n} X_j \star K_{ij}, \quad i = 1 \dots d$$

$$\frac{\partial E}{\partial K_{ij}} = X_j \star \frac{\partial E}{\partial Y_i}$$

Figures copyright, The Independent Code, Convolutional Neural Network from Scratch | Mathematics & Python Code

Backward bias

$$\frac{\partial E}{\partial Y_i} \rightarrow \frac{\partial E}{\partial K_{ij}}, \boxed{\frac{\partial E}{\partial B_i}}$$

$$\frac{\partial E}{\partial Y_i} \rightarrow \frac{\partial E}{\partial X_j}$$

$$Y_i = B_i + \sum_{j=1}^{n} X_j \star K_{ij}$$

Forward propagation

$$\boxed{Y_i = B_i + X_1 \star K_{i1}} + \cdots + X_n \star K_{in}$$

Simplified example

$$\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} + \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} \star \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

$$\begin{cases} y_{11} = b_{11} + k_{11}x_{11} + k_{12}x_{12} + k_{21}x_{21} + k_{22}x_{22} \\ y_{12} = b_{12} + k_{11}x_{12} + k_{12}x_{13} + k_{21}x_{22} + k_{22}x_{23} \\ y_{21} = b_{21} + k_{11}x_{21} + k_{12}x_{22} + k_{21}x_{31} + k_{22}x_{32} \\ y_{22} = b_{22} + k_{11}x_{22} + k_{12}x_{23} + k_{21}x_{32} + k_{22}x_{33} \end{cases}$$

Backward bias

$$\begin{cases} y_{11} = \boxed{b_{11}} + k_{11}x_{11} + k_{12}x_{12} + k_{21}x_{21} + k_{22}x_{22} \\ y_{12} = b_{12} + k_{11}x_{12} + k_{12}x_{13} + k_{21}x_{22} + k_{22}x_{23} \\ y_{21} = b_{21} + k_{11}x_{21} + k_{12}x_{22} + k_{21}x_{31} + k_{22}x_{32} \\ y_{22} = b_{22} + k_{11}x_{22} + k_{12}x_{23} + k_{21}x_{32} + k_{22}x_{33} \end{cases}$$

$$\frac{\partial E}{\partial b_{11}} = \frac{\partial E}{\partial y_{11}} \underbrace{\boxed{\frac{\partial y_{11}}{\partial b_{11}}}}_{1} + \frac{\partial E}{\partial y_{12}} \underbrace{\boxed{\frac{\partial y_{12}}{\partial b_{11}}}}_{0} + \frac{\partial E}{\partial y_{21}} \underbrace{\boxed{\frac{\partial y_{21}}{\partial b_{11}}}}_{0} + \frac{\partial E}{\partial y_{22}} \underbrace{\boxed{\frac{\partial y_{22}}{\partial b_{11}}}}_{0}$$

$$\frac{\partial E}{\partial b_{11}} = \frac{\partial E}{\partial y_{11}}$$

$$\frac{\partial E}{\partial b_{12}} = \frac{\partial E}{\partial y_{12}}$$

$$\frac{\partial E}{\partial b_{21}} = \frac{\partial E}{\partial y_{21}}$$

$$\frac{\partial E}{\partial b_{22}} = \frac{\partial E}{\partial y_{22}}$$

$$\frac{\partial E}{\partial B} = \frac{\partial E}{\partial Y}$$

Simplified version

$$Y = B + X \star K \Rightarrow \frac{\partial E}{\partial B} = \frac{\partial E}{\partial Y}$$

$$\frac{\partial E}{\partial Y} = \begin{bmatrix} \frac{\partial E}{\partial y_{11}} & \frac{\partial E}{\partial y_{12}} \\ \frac{\partial E}{\partial y_{21}} & \frac{\partial E}{\partial y_{22}} \end{bmatrix} \longrightarrow \begin{bmatrix} \frac{\partial E}{\partial b_{11}} & \frac{\partial E}{\partial b_{12}} \\ \frac{\partial E}{\partial b_{21}} & \frac{\partial E}{\partial b_{22}} \end{bmatrix}$$

Actual version

$$Y_i = B_i + \sum_{j=1}^{n} X_j \star K_{ij}, \quad i = 1 \dots d$$

$$\frac{\partial E}{\partial B_i} = \frac{\partial E}{\partial Y_i}$$

Figures copyright, The Independent Code, Convolutional Neural Network from Scratch | Mathematics & Python Code

Backward input

$$\begin{tikzcd}
 & \frac{\partial E}{\partial K_{ij}}, \frac{\partial E}{\partial B_i} \\
\frac{\partial E}{\partial Y_i} \arrow[ur] \arrow[dr] & \\
 & \boxed{\frac{\partial E}{\partial X_j}}
\end{tikzcd}$$

$$Y_i = B_i + \sum_{j=1}^{n} X_j \star K_{ij}$$

Forward propagation

$$\boxed{Y_i = B_i + X_1 \star K_{i1}} + \cdots + X_n \star K_{in}$$

Simplified example

$$\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} + \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} \star \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

$$\begin{cases}
y_{11} = b_{11} + k_{11}x_{11} + k_{12}x_{12} + k_{21}x_{21} + k_{22}x_{22} \\
y_{12} = b_{12} + k_{11}x_{12} + k_{12}x_{13} + k_{21}x_{22} + k_{22}x_{23} \\
y_{21} = b_{21} + k_{11}x_{21} + k_{12}x_{22} + k_{21}x_{31} + k_{22}x_{32} \\
y_{22} = b_{22} + k_{11}x_{22} + k_{12}x_{23} + k_{21}x_{32} + k_{22}x_{33}
\end{cases}$$

Backward bias

$$\begin{cases} y_{11} = b_{11} + \boxed{k_{11}x_{11}} + k_{12}x_{12} + k_{21}x_{21} + k_{22}x_{22} \\ y_{12} = b_{12} + k_{11}x_{12} + k_{12}x_{13} + k_{21}x_{22} + k_{22}x_{23} \\ y_{21} = b_{21} + k_{11}x_{21} + k_{12}x_{22} + k_{21}x_{31} + k_{22}x_{32} \\ y_{22} = b_{22} + k_{11}x_{22} + k_{12}x_{23} + k_{21}x_{32} + k_{22}x_{33} \end{cases}$$

$$\frac{\partial E}{\partial x_{11}} = \frac{\partial E}{\partial y_{11}}\underbrace{\boxed{\frac{\partial y_{11}}{\partial x_{11}}}}_{k_{11}} + \frac{\partial E}{\partial y_{12}}\underbrace{\boxed{\frac{\partial y_{12}}{\partial x_{11}}}}_{0} + \frac{\partial E}{\partial y_{21}}\underbrace{\boxed{\frac{\partial y_{21}}{\partial x_{11}}}}_{0} + \frac{\partial E}{\partial y_{22}}\underbrace{\boxed{\frac{\partial y_{22}}{\partial x_{11}}}}_{0}$$

$$\begin{bmatrix} \boxed{\frac{\partial E}{\partial x_{11}}} & \frac{\partial E}{\partial x_{12}} & \frac{\partial E}{\partial x_{13}} \\ \frac{\partial E}{\partial x_{21}} & \frac{\partial E}{\partial x_{22}} & \frac{\partial E}{\partial x_{23}} \\ \frac{\partial E}{\partial x_{31}} & \frac{\partial E}{\partial x_{32}} & \frac{\partial E}{\partial x_{33}} \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial y_{11}} & \frac{\partial E}{\partial y_{12}} \\ \frac{\partial E}{\partial y_{21}} & \frac{\partial E}{\partial y_{22}} \end{bmatrix} \underset{full}{\star} \begin{bmatrix} k_{22} & k_{21} \\ k_{12} & k_{11} \end{bmatrix}$$

$$\boxed{Y = B + X \star K \Rightarrow \frac{\partial E}{\partial X} = \frac{\partial E}{\partial Y} \underset{full}{*} K}$$

$$\frac{\partial E}{\partial Y} = \begin{bmatrix} \frac{\partial E}{\partial y_{11}} & \frac{\partial E}{\partial y_{12}} \\ \frac{\partial E}{\partial y_{21}} & \frac{\partial E}{\partial y_{22}} \end{bmatrix} \longrightarrow \begin{bmatrix} \frac{\partial E}{\partial x_{11}} & \frac{\partial E}{\partial x_{12}} & \frac{\partial E}{\partial x_{13}} \\ \frac{\partial E}{\partial x_{21}} & \frac{\partial E}{\partial x_{22}} & \frac{\partial E}{\partial x_{23}} \\ \frac{\partial E}{\partial x_{31}} & \frac{\partial E}{\partial x_{32}} & \frac{\partial E}{\partial x_{33}} \end{bmatrix}$$

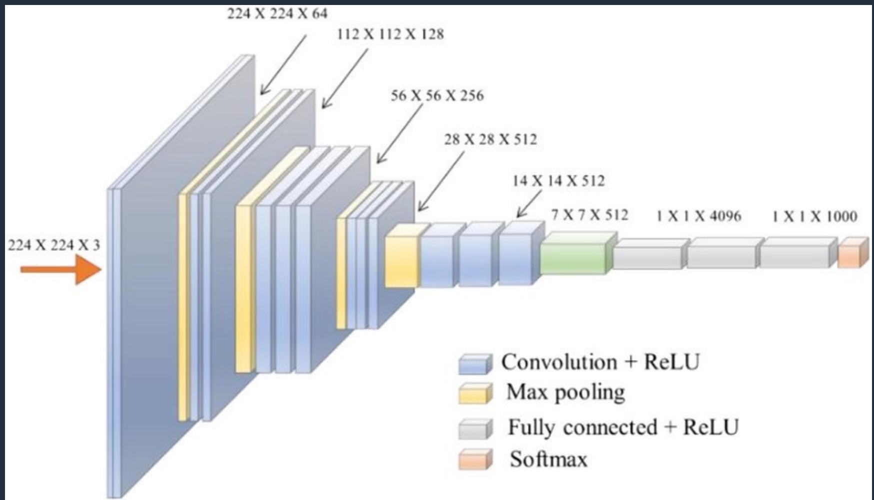
Actual version

$$\boxed{\frac{\partial E}{\partial X_j} = \sum_{i=1}^{n} \frac{\partial E}{\partial Y_i} \underset{full}{*} K_{ij}}$$

CNN Visualization

Deep CNN Layer Visualization

VGG16



Figures copyright, Denis Dmitriev, VGG16 Neural Network Visualization, https://www.youtube.com/watch?v=RNnKtNrsrmg&t=2s

Q&A

감사합니다