AI and Deep Learning

4-2 Convolution Neural Network I

Zhonglei Wang

WISE, SOE, CHOW

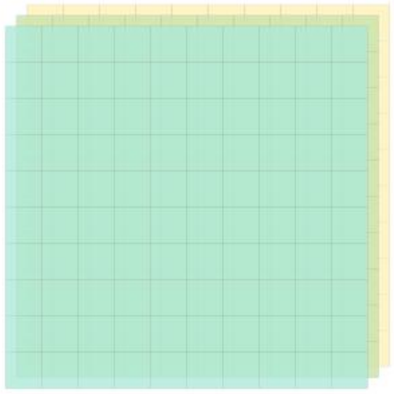
XMU

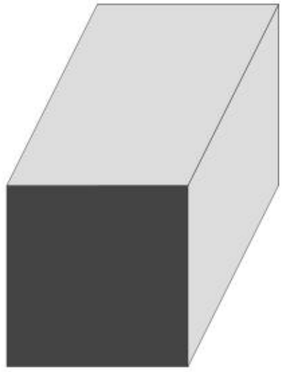
(Version v1)

Contents

Background

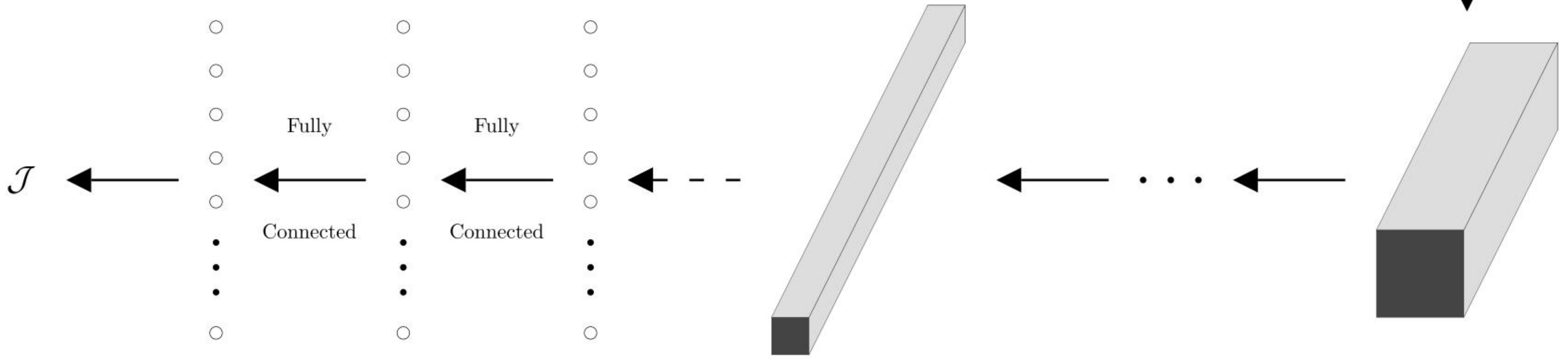
1. Given an image, why not use a fully connected neural network?
   - Number of parameters is overwhelmingly large, easily leading to overfitting
     - To overcome this problem, we may require more training data, more memory and faster computation algorithms
     - Almost impossible in practice
   - Overlooks the spatial structure
   - Not location (translation) invariant
2. Consider a new network, convolution neural network, for image processing





Input $(\mathbb{R}^{d_C \times d_H \times d_W})$

To be discussed



$\mathcal{J}$

$\boldsymbol{A}^{[L-1]} \in \mathbb{R}^{d^{[L-1]} \times 1 \times 1}$

$\boldsymbol{A}^{[L-2]} \in \mathbb{R}^{d_C^{[L-2]} \times d_W^{[L-2]} \times d_H^{[L-2]}}$

$\boldsymbol{A}^{[L]} \in \mathbb{R}^{d^{[L]} \times 1 \times 1}$

Summary

1. Generally, $d_C^{[l]}$ increases as l increases, but $d_W^{[l]}$ and $d_H^{[l]}$ decrease
2. The vectorization step does not involve any new model parameter
3. Typically, there are three "layers" for CNN
   - Convolution layers (CONV)
   - Pooling layers (POOL)
   - Fully connected layers (FC)

Notation

1. l: Layer index
2. $d_C^{[l]}$: Number of channels of the lth layer
3. $d_H^{[l]}$: Height of the "image" of the lth layer
4. $d_W^{[l]}$: Width of the "image" of the lth layer
5. $f^{[l]}$: Kernel size associated with the lth layer ($f_1^{[l]} = f_2^{[l]}$ by default)
6. $p^{[l]}$: Padding associated with the lth layer ($p_1^{[l]} = p_2^{[l]}$ by default)
7. $s^{[l]}$: Stride associated with the lth layer ($s_1^{[l]} = s_2^{[l]}$ by default)
8. No dilation

Forward propagation for a CONV layer

1. $\boldsymbol{Z}^{[l]} \in \mathbb{R}^{n \times d_C^{[l]} \times d_H^{[l]} \times d_W^{[l]}}$: Output of linear transformation

$$\boldsymbol{Z}^{[l]} = \boldsymbol{A}^{[l-1]} \text{“} * \text{”} \boldsymbol{W}^{[l]} \text{“} + \text{”} \boldsymbol{b}^{[l]}$$

- $\boldsymbol{A}^{[l-1]} \in \mathbb{R}^{n \times d_C^{[l-1]} \times d_H^{[l-1]} \times d_W^{[l-1]}}$: Input for the lth layer
- $\boldsymbol{W}^{[l]} \in \mathbb{R}^{d_C^{[l]} \times d_C^{[l-1]} \times f^{[l]} \times f^{[l]}}$: Kernels
- $\boldsymbol{b}^{[l]} \in \mathbb{R}^{d_C^{[l]} \times 1 \times 1}$: Bias term
- “ $*$ ”: Convolution for each sample and each channel
- “ $+$ ”: Common bias for each channel

2. $\boldsymbol{A}^{[l]} \in \mathbb{R}^{n \times d_C^{[l]} \times d_H^{[l]} \times d_W^{[l]}}$: Output of activation

$$\boldsymbol{A}^{[l]} = \sigma^{[l]}(\boldsymbol{Z}^{[l]})$$

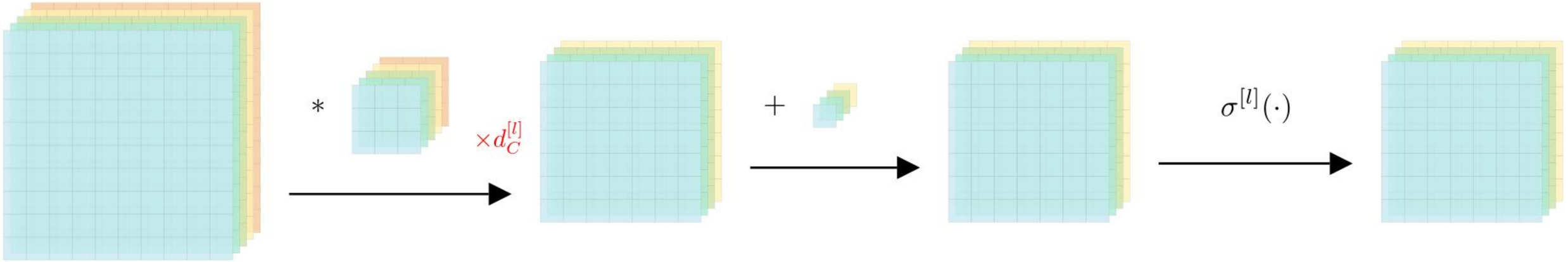
- $\sigma^{[l]}(\cdot)$: Activation function for the lth layer

Forward propagation for a CONV layer

1. For simplicity, we show forward- and back-propagation for $n = 1$
   - $\boldsymbol{A}^{[l-1]} \in \mathbb{R}^{d_C^{[l-1]} \times d_H^{[l-1]} \times d_W^{[l-1]}}$
   - $\boldsymbol{Z}^{[l]} \in \mathbb{R}^{d_C^{[l]} \times d_H^{[l]} \times d_W^{[l]}}$
2. Forward propagation:

$$\boldsymbol{Z}^{[l]} = \boldsymbol{A}^{[l-1]} \text{“} * \text{”} \boldsymbol{W}^{[l]} \text{“} + \text{”} \boldsymbol{b}^{[l]},$$
$$\boldsymbol{A}^{[l]} = \sigma^{[l]}(\boldsymbol{Z}^{[l]})$$

Visulization



$\boldsymbol{A}^{[l-1]} \in \mathbb{R}^{d_C^{[l-1]} \times d_H^{[l-1]} \times d_W^{[l-1]}}$

$\boldsymbol{Z}_0^{[l]} = \mathbb{R}^{d_C^{[l]} \times d_H^{[l]} \times d_W^{[l]}}$

$\boldsymbol{Z}^{[l]} \in \mathbb{R}^{d_C^{[l]} \times d_H^{[l]} \times d_W^{[l]}}$

$\boldsymbol{W}^{[l]} \in \mathbb{R}^{d_C^{[l]} \times d_C^{[l-1]} \times f^{[l]} \times f^{[l]}}$

$\boldsymbol{b}^{[l]} \in \mathbb{R}^{d_C^{[l]} \times 1 \times 1}$

$\boldsymbol{A}^{[l]} \in \mathbb{R}^{d_C^{[l]} \times d_H^{[l]} \times d_W^{[l]}}$

Pooling

1. Pooling is typically used after convolution layers to achieve
   - Reduce height and width for each channel (downsampling), and computation efficiency is improved
   - Achieves robustness by neglecting useless or repeated information
   - Increase the receptive field
   - Achieve robustness against small translations, rotations, and other distortions in the input
   - Prevent overfitting, especially when combined with dropout

Pooling

1. Two types of pooling:
   - Average pooling
   - Max pooling
2. Notice:
   - Pooling is conducted for each channel
   - Pooling does not involve new model parameters

Average pooling

1. Input size: 6×6
2. Kernel size: 2×2
3. Stride: 2

Average pooling

Input

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

Kernel

0.25	0.25
0.25	0.25

Result

4.25	6.0	6.25
3.0	6.0	2.25
3.25	3.75	3.0

Max pooling

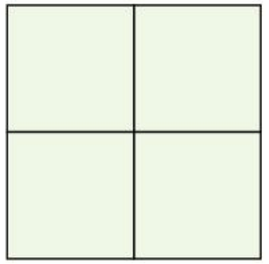
1. Input size: 6×6
2. Kernel size: 2×2
3. Stride: 2

Max pooling

Input

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

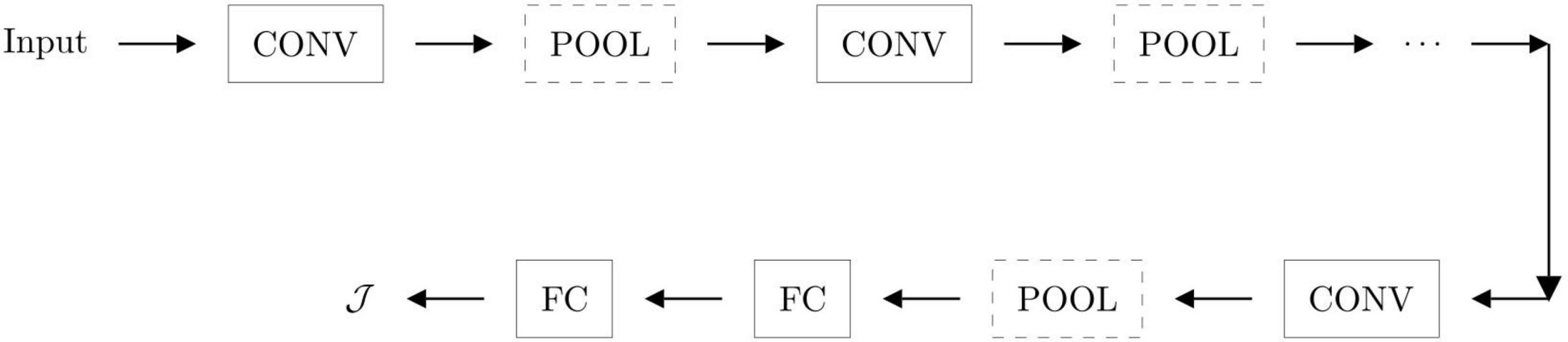
Kernel



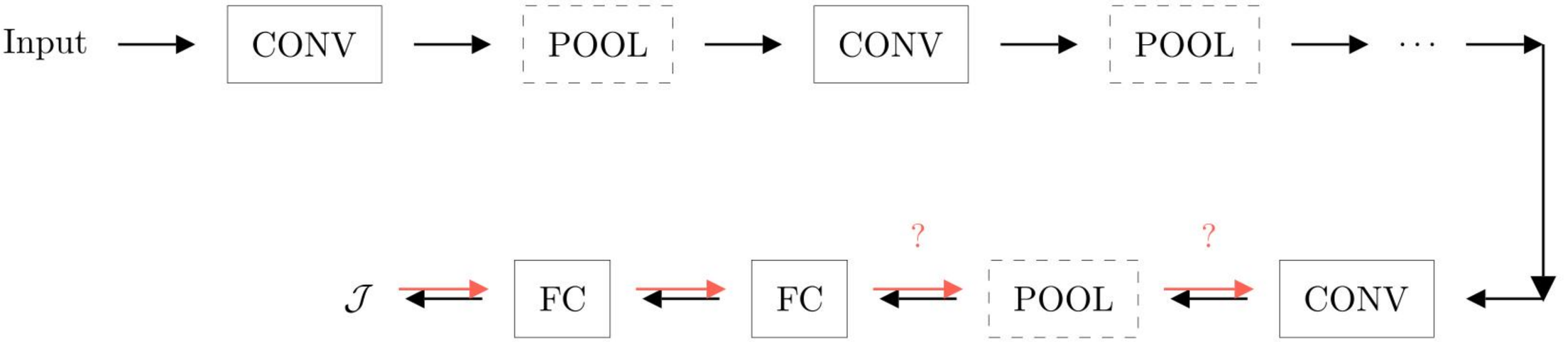
Result

7	9	9
5	9	7
9	7	7

Forward propagation



Backpropagation

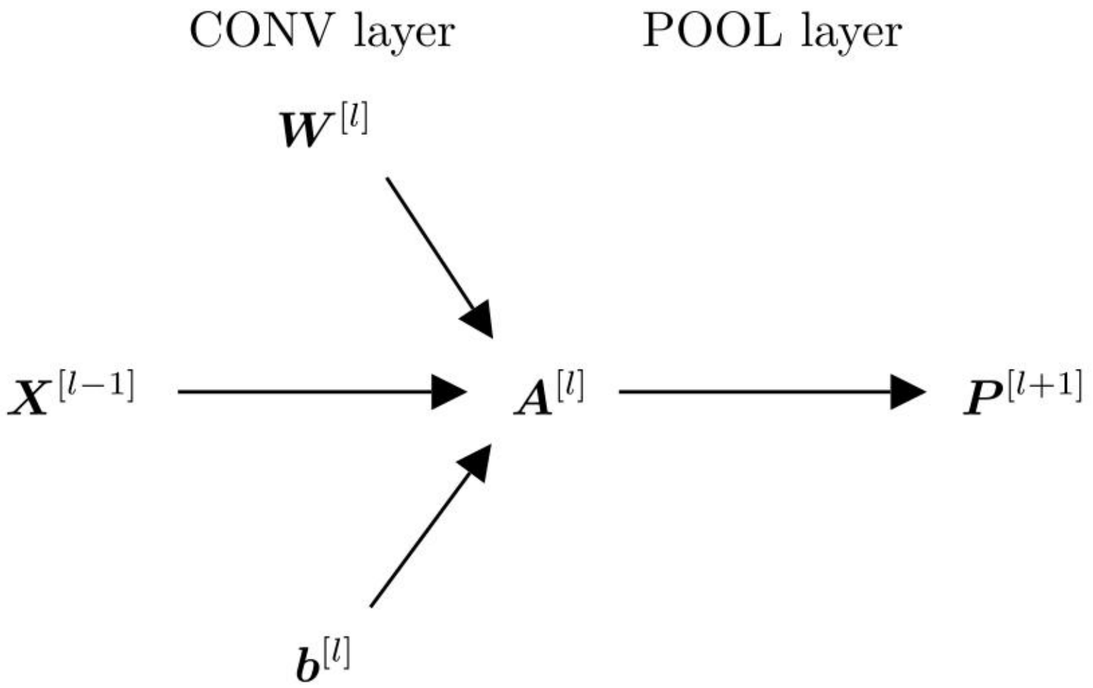


1. We have learnt backpropagation for fully connected layers
2. Thus, it remains to show the backpropagation for
   - The pooling layer (POOL)
   - The convolution layer (CONV)
3. In the following, we assume a POOL is conducted right after a CONV

Backpropagation

1. Consider the following case:
   - The lth layer is a CONV layer
   - The $(l+1)$th layer is a POOL layer
2. Denote
   - $\boldsymbol{W}^{[l]} \in \mathbb{R}^{d_C^{[l]} \times d_C^{[l-1]} \times f^{[l]} \times f^{[l]}}$: Kernel for the lth layer
   - $\boldsymbol{b}^{[l]} \in \mathbb{R}^{d_C^{[l]} \times 1 \times 1}$: Bias for the lth layer, taking $\boldsymbol{P}^{[l-1]}$ as input
   - $\boldsymbol{A}^{[l]} \in \mathbb{R}^{d_C^{[l]} \times d_H^{[l]} \times d_W^{[l]}}$: Output of the lth layer (CONV)
   - $\boldsymbol{P}^{[l+1]} \in \mathbb{R}^{d_C^{[l]} \times d_H^{[l+1]} \times d_W^{[l+1]}}$: Output of the $(l+1)$th layer (POOL)

Structure



1. Assume that $\mathrm{d}\boldsymbol{P}^{[l+1]}$ is available
   - Backpropagation for vectorization is trivial

Backpropagation for average pooling

$\boldsymbol{A}^{[l]}$

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

Kernel $\boldsymbol{M}$

0.25	0.25
0.25	0.25

$\boldsymbol{P}^{[l+1]}$

4.25	6.0	6.25
3.0	6.0	2.25
3.25	3.75	3.0

$$\boldsymbol{P}_{11}^{[l+1]} = 0.25 \times (\boldsymbol{A}_{11}^{[l]} + \boldsymbol{A}_{12}^{[l]} + \boldsymbol{A}_{21}^{[l]} + \boldsymbol{A}_{22}^{[l]})$$

$$\mathrm{d}\boldsymbol{A}_{11}^{[l]} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{A}_{11}^{[l]}} = \sum_{i=1}^{3}\sum_{j=1}^{3} \frac{\partial \mathcal{J}}{\partial \boldsymbol{P}_{ij}^{[l+1]}} \times \frac{\partial \boldsymbol{P}_{ij}^{[l+1]}}{\partial \boldsymbol{A}_{11}^{[l]}} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{P}_{11}^{[l+1]}} \times \frac{\partial \boldsymbol{P}_{11}^{[l+1]}}{\partial \boldsymbol{A}_{11}^{[l]}} = \mathrm{d}\boldsymbol{P}_{11}^{[l+1]} \times 0.25$$

$$\mathrm{d}\boldsymbol{A}_{12}^{[l]} = \mathrm{d}\boldsymbol{A}_{21}^{[l]} = \mathrm{d}\boldsymbol{A}_{22}^{[l]} = 0.25 \times \mathrm{d}\boldsymbol{P}_{11}^{[l+1]}$$

Backpropagation for average pooling

1. For average pooling, we have

$$\mathrm{d}\boldsymbol{A}^{[l]} = \mathrm{d}\boldsymbol{P}^{[l+1]} \otimes \boldsymbol{M}$$

- $\boldsymbol{A} \otimes \boldsymbol{B}$: Kronecker production of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$
- Valid for a special case:
  - Kernel size "equals to" stride
  - Size of the original image can be "divided by" the kernel size

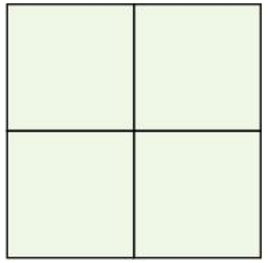
2. Chain rule can be used to derive the result for general cases (check by yourself)

Backpropagation for max pooling

$\boldsymbol{A}^{[l]}$

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

Kernel



$\boldsymbol{P}^{[l+1]}$

7	9	9
5	9	7
9	7	7

$$\boldsymbol{P}_{11}^{[l+1]} = \max\{\boldsymbol{A}_{11}^{[l]} + \boldsymbol{A}_{12}^{[l]} + \boldsymbol{A}_{21}^{[l]} + \boldsymbol{A}_{22}^{[l]}\} = \boldsymbol{A}_{22}^{[l]}$$

$$\mathrm{d}\boldsymbol{A}_{11}^{[l]} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{A}_{11}^{[l]}} = \sum_{i=1}^{3}\sum_{j=1}^{3} \frac{\partial \mathcal{J}}{\partial \boldsymbol{P}_{ij}^{[l+1]}} \times \frac{\partial \boldsymbol{P}_{ij}^{[l+1]}}{\partial \boldsymbol{A}_{11}^{[l]}} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{P}_{11}^{[l+1]}} \times \frac{\partial \boldsymbol{P}_{11}^{[l+1]}}{\partial \boldsymbol{A}_{11}^{[l]}} = \mathrm{d}\boldsymbol{P}_{11}^{[l+1]} \times 0 = 0$$

$$\mathrm{d}\boldsymbol{A}_{12}^{[l]} = \mathrm{d}\boldsymbol{A}_{21}^{[l]} = 0; \quad \mathrm{d}\boldsymbol{A}_{22}^{[l]} = \mathrm{d}\boldsymbol{P}_{11}^{[l+1]}$$

Backpropagation for max pooling

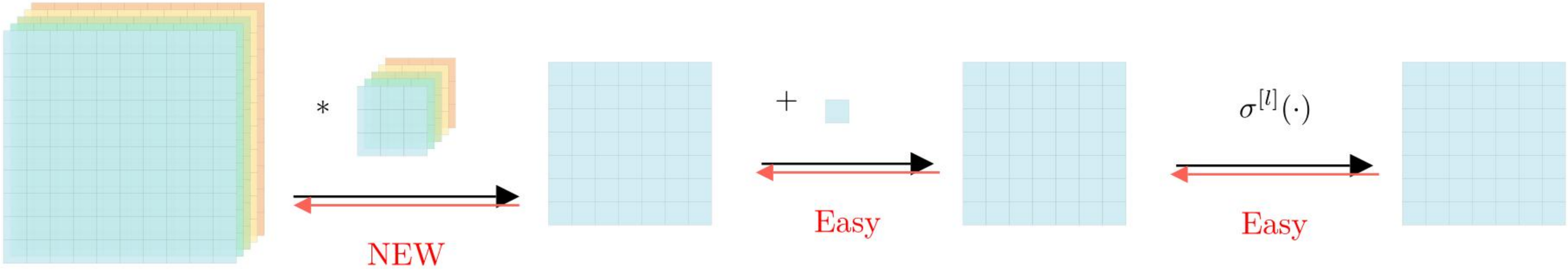
1. For max pooling, $\mathrm{d}\boldsymbol{A}^{[l]}$ is obtained by
   - Stacking small matrices associated with each elements in $\boldsymbol{P}^{[l+1]}$ times a 2×2 mask matrix
   - Valid for a special case:
     - ▷ Kernel size "equals to" stride
     - ▷ Size of the original image can be "divided by" the kernel size
2. Chain rule can be used to derive the result for general cases (check by yourself)

Remarks

1. We have finished the backpropagation from $\mathrm{d}\boldsymbol{P}^{[l+1]}$ to $\mathrm{d}\boldsymbol{A}^{[l]}$
   - For average pooling, under specific conditions, gradients are evenly distributed
   - For max pooling, under specific conditions, gradient is propagated back only at the position of the forward maximum
   - Pooling layer does not involve any model parameters
2. Assume the availability of $\mathrm{d}\boldsymbol{A}^{[l]}$ for the following analysis

Backpropagation for CONV

1. For simplicity, let $d_C^{[l]} = 1$.



$\boldsymbol{X}^{[l-1]} \in \mathbb{R}^{d_C^{[l-1]} \times d_H^{[l-1]} \times d_W^{[l-1]}}$

$\boldsymbol{Z}_0^{[l]} \in \mathbb{R}^{1 \times d_H^{[l]} \times d_W^{[l]}}$

$\boldsymbol{Z}_1^{[l]} \in \mathbb{R}^{1 \times d_H^{[l]} \times d_W^{[l]}}$

$\boldsymbol{W}^{[l]} \in \mathbb{R}^{1 \times d_C^{[l-1]} \times f^{[l]} \times f^{[l]}}$

$\boldsymbol{b}^{[l]} \in \mathbb{R}^{1 \times 1 \times 1}$

$\boldsymbol{A}^{[l]} \in \mathbb{R}^{1 \times d_H^{[l]} \times d_W^{[l]}}$

$\mathrm{d}\boldsymbol{W}^{[l]} = ?$

$\mathrm{d}\boldsymbol{b}^{[l]} = \sum_{i=1}^{d_H^{[l]}} \sum_{j=1}^{d_W^{[l]}} \mathrm{d}Z_{1,ij}^{[l]}$

$\mathrm{d}\boldsymbol{X}^{[l-1]} = ?$

$\mathrm{d}\boldsymbol{Z}_0^{[l]} = \mathrm{d}\boldsymbol{Z}_1^{[l]}$

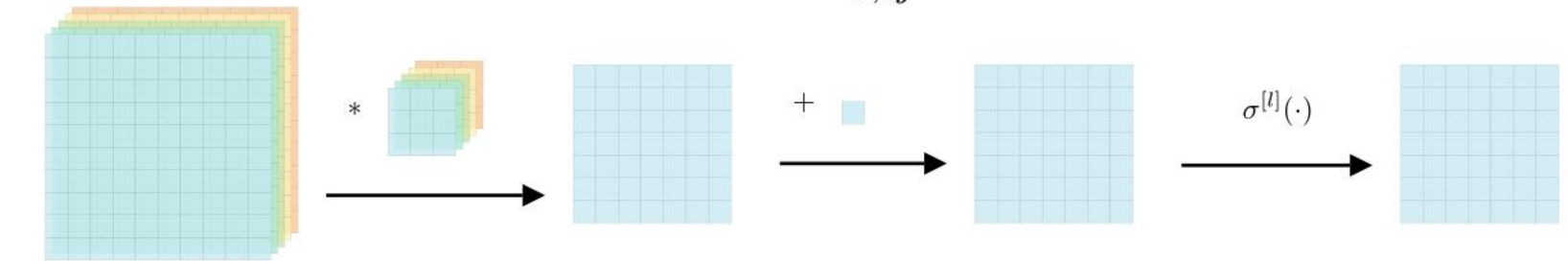
$\mathrm{d}\boldsymbol{Z}_1^{[l]} = \sigma^{[l]'}(\boldsymbol{Z}^{[l]}) \circ \mathrm{d}\boldsymbol{A}^{[l]}$

Backpropagation for CONV

1. To find derivative with respect to something, we need to find where its information is contained
2. Initialize $\mathrm{d}\boldsymbol{X}^{[l-1]}$ as a zero matrix of the same dimension as the input image $\boldsymbol{X}^{[l-1]}$
3. Informally, obtain

$$\mathrm{d}\boldsymbol{X}^{[l-1]}_{\text{slice},ij} \mathrel{+}= \mathrm{d}\boldsymbol{Z}^{[l]}_{0,ij} \times \boldsymbol{W}^{[l]}$$

- $\boldsymbol{X}^{[l-1]}_{\text{slice},ij}$: The part of $\boldsymbol{X}^{[l-1]}$ used to obtain $\boldsymbol{Z}^{[l]}_{0,ij}$

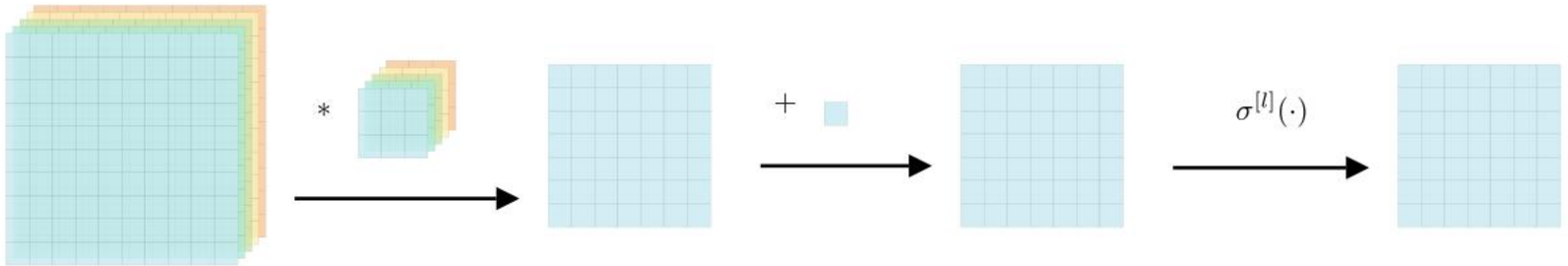


Backpropagation for CONV

1. It is obvious that every element in $\boldsymbol{Z}_0^{[l]}$ contains information of $\boldsymbol{W}^{[l]}$
2. Thus, we have

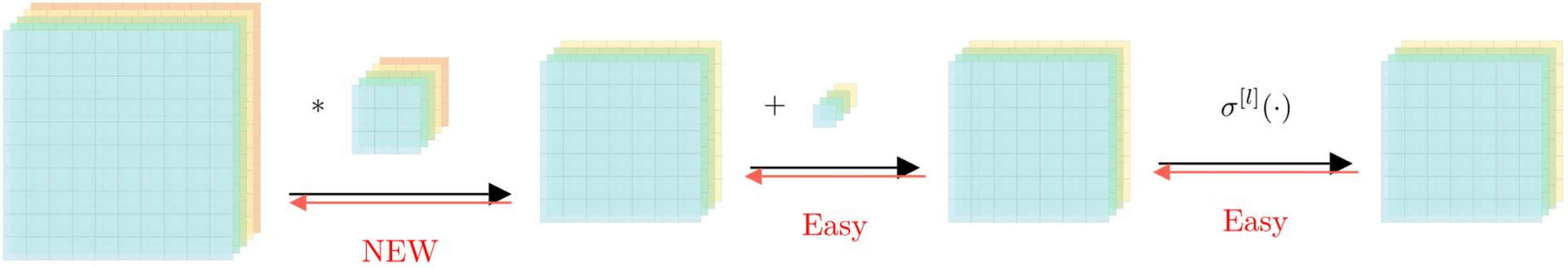
$$\mathrm{d}\boldsymbol{W}^{[l]} = \sum_{i=1}^{d_H^{[l]}} \sum_{j=1}^{d_W^{[l]}} \mathrm{d}\boldsymbol{Z}_{0,ij}^{[l]} \times \boldsymbol{B}_{ij}^{[l-1]}$$

- $\boldsymbol{B}_{ij}^{[l-1]}$: The part of $\boldsymbol{X}^{[l-1]}$ used to obtain $Z_{0,ij}^{[l]}$



Backpropagation for CONV

1. Consider the general case where there exist $d_C^{[l]}$ channels in the lth layer



$$\boldsymbol{X}^{[l-1]} \in \mathbb{R}^{d_C^{[l-1]} \times d_H^{[l-1]} \times d_W^{[l-1]}} \qquad \boldsymbol{Z}_0^{[l]} \in \mathbb{R}^{d_C^{[l]} \times d_H^{[l]} \times d_W^{[l]}} \qquad \boldsymbol{Z}_1^{[l]} \in \mathbb{R}^{d_C^{[l]} \times d_H^{[l]} \times d_W^{[l]}}$$

$$\boldsymbol{W}^{[l]} \in \mathbb{R}^{d_C^{[l]} \times d_C^{[l-1]} \times f^{[l]} \times f^{[l]}} \qquad \boldsymbol{b}^{[l]} \in \mathbb{R}^{d_C^{[l]} \times 1 \times 1} \qquad \boldsymbol{A}^{[l]} \in \mathbb{R}^{d_C^{[l]} \times d_H^{[l]} \times d_W^{[l]}}$$

$$\mathrm{d}\boldsymbol{W}^{[l]} = ? \qquad \mathrm{d}\boldsymbol{b}_c^{[l]} = \sum_{i=1}^{d_H^{[l]}} \sum_{j=1}^{d_W^{[l]}} \mathrm{d}Z_{1,cij}^{[l]} \quad (c = 1, \ldots, d_C^{[l]})$$

$$\mathrm{d}\boldsymbol{X}^{[l-1]} = ? \qquad \mathrm{d}\boldsymbol{Z}_0^{[l]} = \mathrm{d}\boldsymbol{Z}_1^{[l]} \qquad \mathrm{d}\boldsymbol{Z}_1^{[l]} = \sigma^{[l]'}(\boldsymbol{Z}^{[l]}) \circ \mathrm{d}\boldsymbol{A}^{[l]}$$

Backpropagation for CONV

1. Think about where the information is contained when calculating derivatives
2. The information about $\boldsymbol{X}^{[l-1]}$ is contained in every channel of $\boldsymbol{Z}_0^{[l]}$
3. Initialize $\mathrm{d}\boldsymbol{X}^{[l-1]}$ as a zero matrix of the same dimension as $\boldsymbol{X}^{[l-1]}$
4. Informally, obtain

$$\mathrm{d}\boldsymbol{X}_{\text{slice},ij}^{[l-1]} += \sum_{c=1}^{d_C^{[l]}} \mathrm{d}\boldsymbol{Z}_{0,ijc}^{[l]} \times \boldsymbol{W}_c^{[l]}$$

- $\boldsymbol{X}_{\text{slice},ij}^{[l-1]}$: The part of $\boldsymbol{X}^{[l-1]}$ used to obtain $\boldsymbol{Z}_{0,ijc}^{[l]}$ for $c = 1, \ldots, d_C^{[l]}$
- $\boldsymbol{W}_c^{[l]}$: The cth kernel in the lth layer

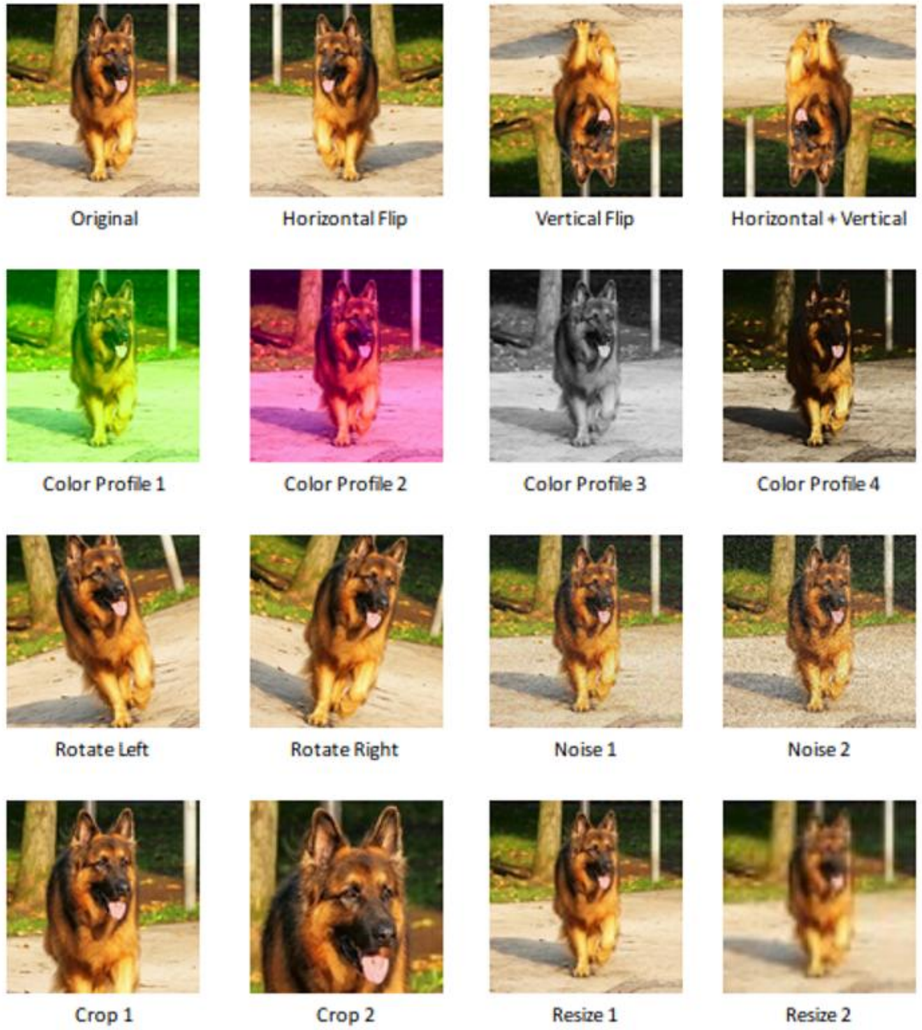
Backpropagation for CONV

1. It is obvious that every element in $\boldsymbol{Z}_{0,c}^{[l]}$ contains information of $\boldsymbol{W}_c^{[l]}$ for $c = 1, \ldots, d_C^{[l]}$
2. However, $\boldsymbol{W}_c^{[l]}$ is not used for calculation of other channels
3. Thus, we have

$$\mathrm{d}\boldsymbol{W}_c^{[l]} = \sum_{i=1}^{d_H^{[l]}} \sum_{j=1}^{d_W^{[l]}} \mathrm{d}\boldsymbol{Z}_{0,cij}^{[l]} \times \boldsymbol{B}_{ij}^{[l-1]} \quad (c = 1, \ldots, d_C^{[l]})$$

- $\boldsymbol{B}_{ij}^{[l-1]}$: The part of $\boldsymbol{X}^{[l-1]}$ used to obtain $\boldsymbol{Z}_{0,cij}^{[l]}$

Data augmentation



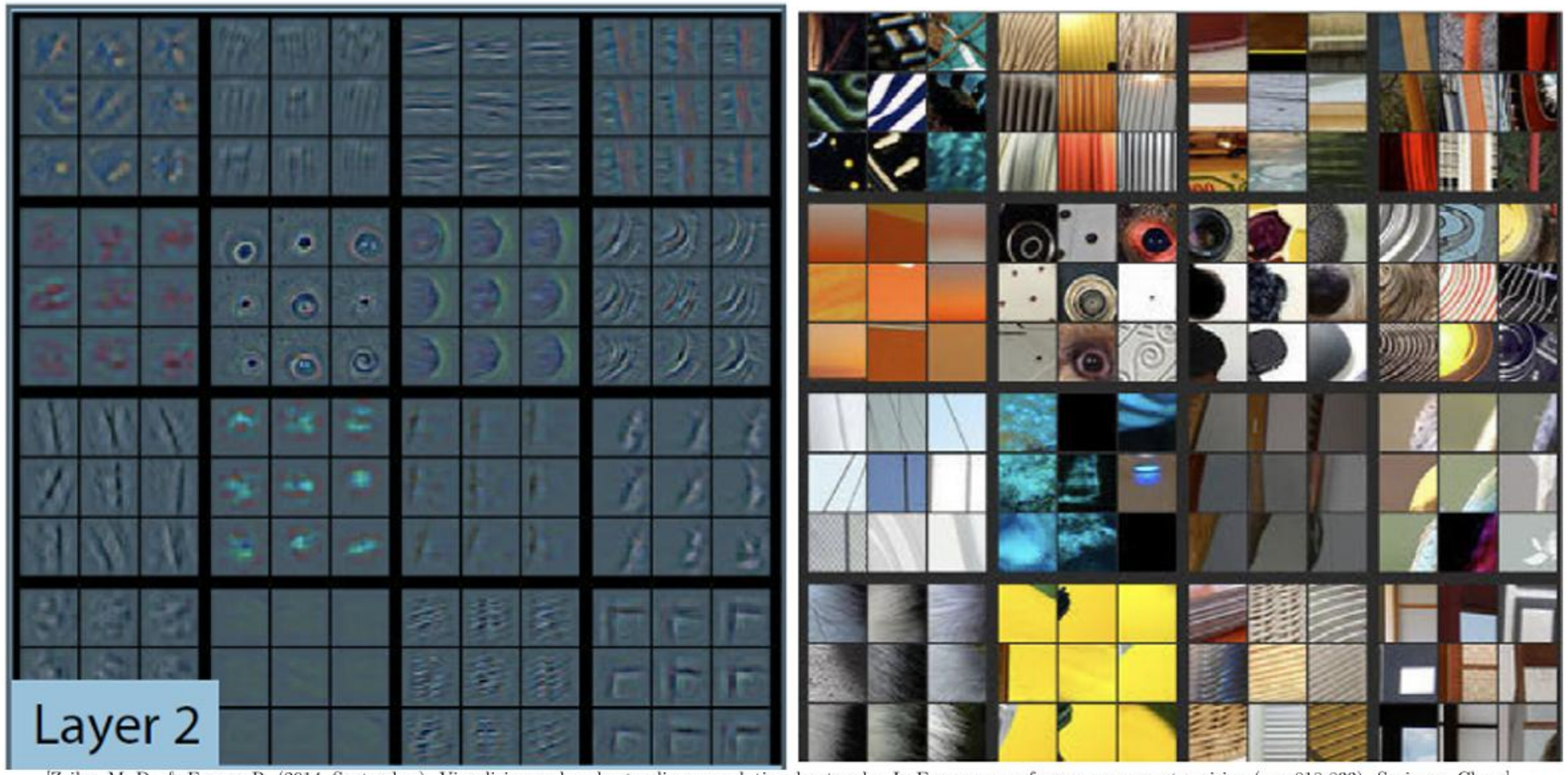
[https://www.volansys.com/blog/data-augmentation-in-ml/]

Understanding convolution networks

Layer 1

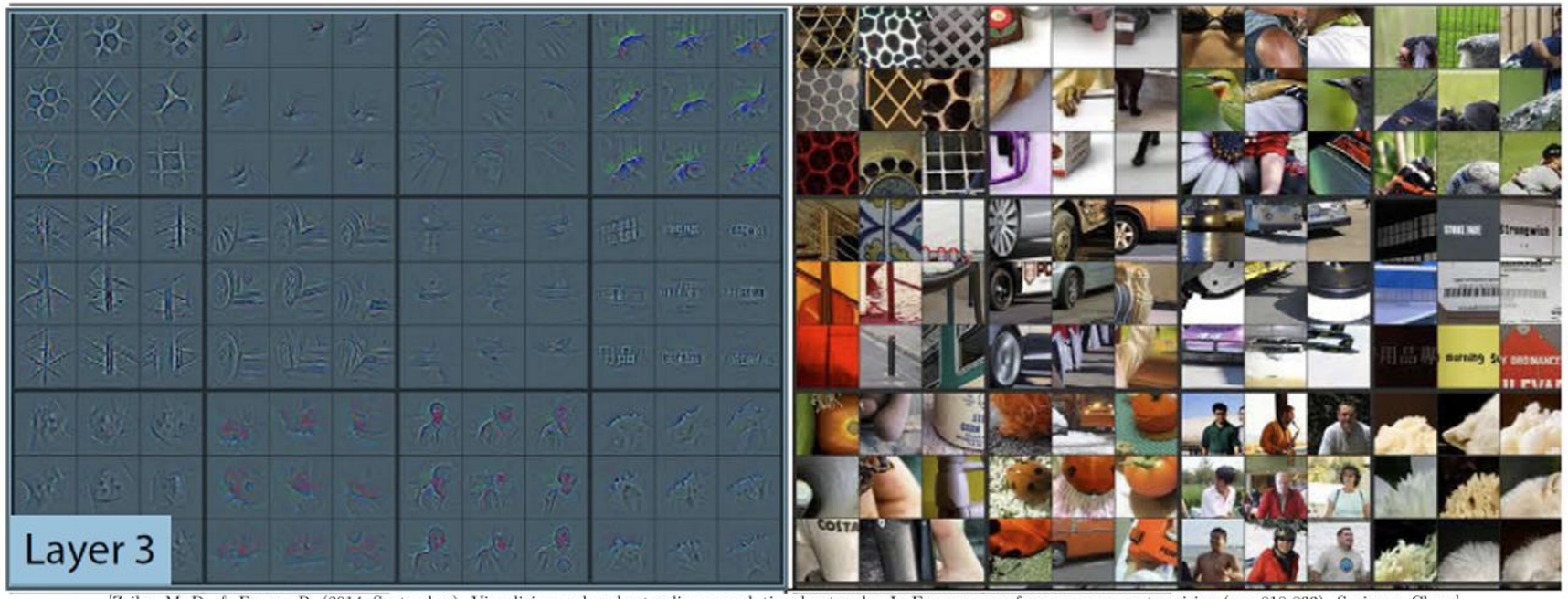
[Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham]

Understanding convolution networks



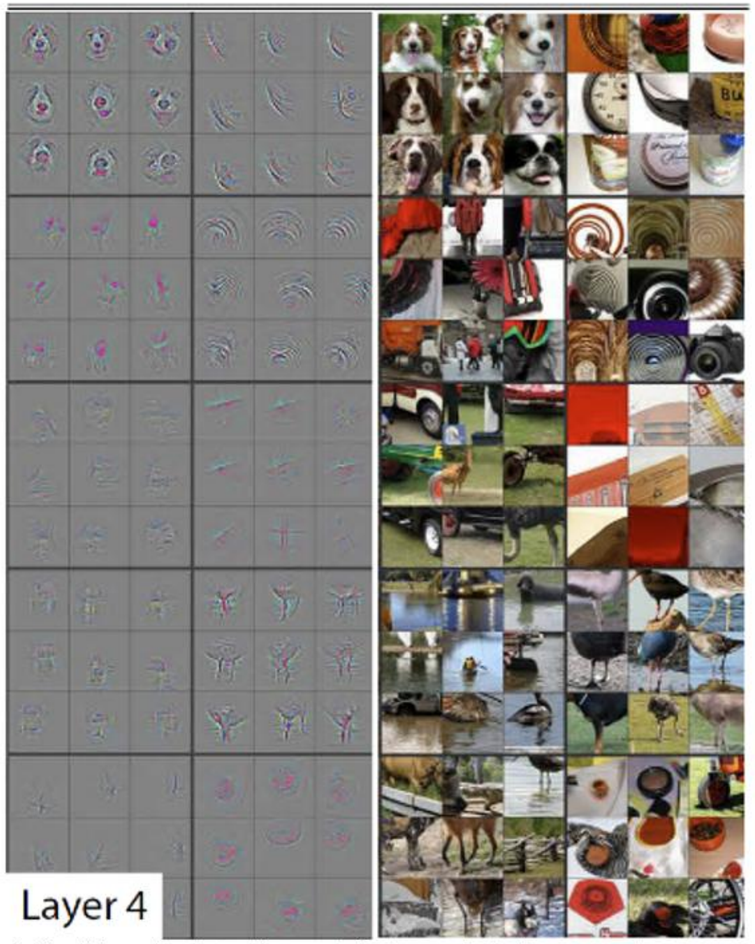
[Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham]

Understanding convolution networks



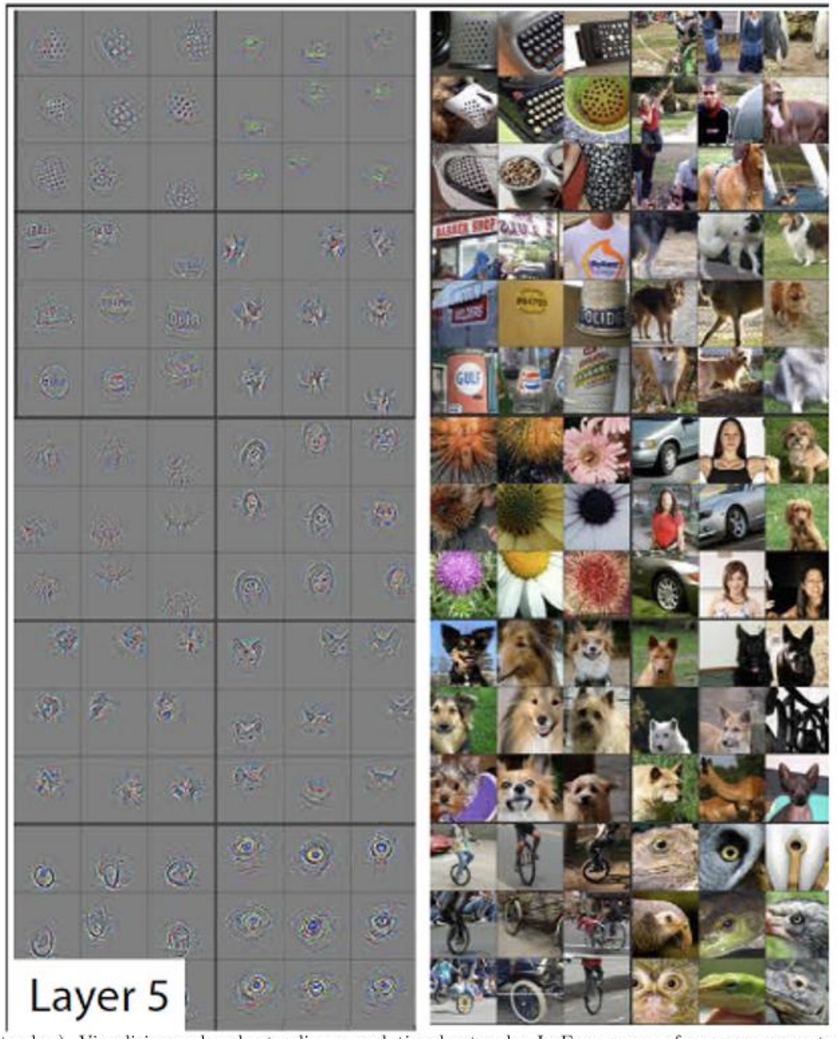
[Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham]

Understanding convolution networks



[Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham]

Understanding convolution networks



[Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham]

Summary

1. Motivation: Spatial structure, locality, and parameter sharing
2. Convolution and pooling layer: Extraction of local information
3. CNN exhibits a hierarchical organization to exctract information from simple to complex