AIL721: Deep Learning

Instructor: James Arambam

ScAI | Yardi School of Artificial Intelligence
Indian Institute of Technology Delhi

Class Announcements

ScAI

❑ Final Groups

Group No.	Student 1	Email 1	Student 2	Email 2	Student 3	Email 3
1	Krishna Sharma	mt1221263@maths.iitd.ac.in	Sarthak Maheshwari	mt1221258@iitd.ac.in	Shikhar Gupta	mt1221925@maths.iitd.ac.in
2	Shreyash Padeer	aib242800@iitd.ac.in	Sanskar Gupta	aib242293@iitd.ac.in	Parth Prajapati	aib242288@iitd.ac.in
3	Kashish Srivastava	aib242289@iitd.ac.in	Sudipto Ghosh	aiz248311@iitd.ac.in	Samidha Verma	aiz238705@iitd.ac.in
4	Vikramjeet Maitra	try227501@tripp.iitd.ac.in				
5	Samyak Jain	mt1221658@iitd.ac.in	Pranav Kumar	ee3221191@iitd.ac.in	Sarthak Gupta	ee1221678@iitd.ac.in
6	Arijit Das	aiz248317@iitd.ac.in	Shubhojit Naskar	aib242285@iitd.ac.in	Shreyash Shimpi	aib242291@iitd.ac.in
7	Priyal Jain	mt6210949@iitd.ac.in	Aditya Thomas	mt6210944@iitd.ac.in		
8	Suryansh	ee3221431@iitd.ac.in	Kabir Nagpal	ee3221743@iitd.ac.in	Shahid Khan	ee1221163@iitd.ac.in
9	Ankit Mazumder	aiy227513@iitd.ac.in	Shashank K.Vempati	aiy227509@iitd.ac.in		
10	Aditya	ce1210494@iitd.ac.in				
11	Poonam Rajput	aiz248308@iitd.ac.in	Rohan Roy	mt1221294@iitd.ac.in		
12	Harshit Joshi	mas237141@iitd.ac.in	Kartikeya Rajput	mt1210922@iitd.ac.in		
13	Raghvendra kumar	aiz248680@iitd.ac.in				
14	Moksh Malhotra	siy237588@iitd.ac.in				
15	Vaibhav Seth	mt1210236@iitd.ac.in	Dhruv Kushwaha	mt1210235@iitd.ac.in	Tvisha Mallik	cs1210085@iitd.ac.in
16	Anamitra Singha	siy237536@iitd.ac.in				
17	Aryan Sudan	ph1221425@iitd.ac.in				
18	Vedant Sharma	csy247553@iitd.ac.in	Saurabh Barthwal	csy247562@iitd.ac.in	Kushagra Karar	csy247554@iitd.ac.in
19	Gaurav Meena	aib242286@iitd.ac.in	Utkarsh Giri	aib242287@iitd.ac.in	Aadhaar	eez248407@iitd.ac.in
20	Khushall Gourav	eea232002@iitd.ac.in	Shuvam Routray	eea232001@iitd.ac.in	Virendra Patel	eey237518@iitd.ac.in
21	Pranav Khetarpal	ce1210480@iitd.ac.in				
22	Ravi Parihar	ee1210156@iitd.ac.in	Shivyankar Singh Rathore	aiy247546@iitd.ac.in	Mudit Sharma	aiz248313@iitd.ac.in

Class Announcements

- Random Groups

1. BHAVIK SANKHLA, 2020MT60873
2. PRADYOT DUBEY, 2021CH10374
3. NEITHALA S., 2022HCS7017
4. DEBANJAN HALDER, 2023PHS7161
5. HIMANSHU CHANDRASHEKHAR CHALPE, 2024AIB2294
6. Krishna Kunal, 2024SMZ8081

Story so far

- We learned about the **supervised learning** pipeline of neural networks.
- **Fully connected** network architecture.

Computer Vision

ScAI

- Major application area of **machine learning**.
  - Revolutionized by **Deep Learning**.
- **Applications** of ML in computer vision
  - Image Classification.
  - Object Detection.
  - Image Segmentation.
  - Caption Generation
  - Image Synthesis.
  - Depth Prediction.
  - Scene Reconstruction.
  - Super-Resolution.

Structure of Data

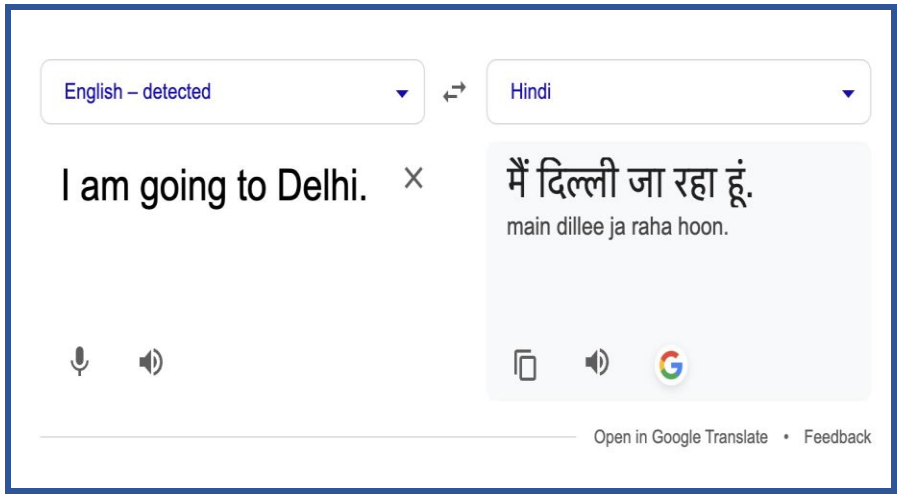
- In simple ML models, we typically assume **observed data** values are **unstructured**.

$$X = (x_1, \ldots x_D)$$

- We **do not assume** anything about how each **data element** might relate to each other.

- However, in **NLP** tasks, each **word forms a sequence**.
  - We can expect some **dependencies** among words.

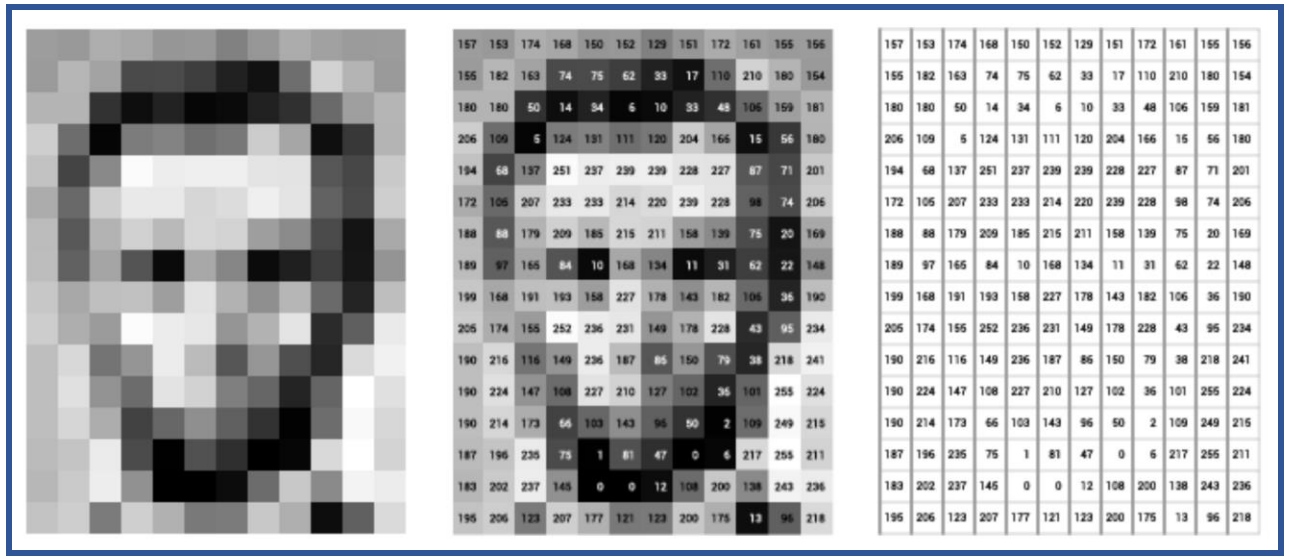
- Similarly, **pixels of image data** also have a **spatial relationship**.



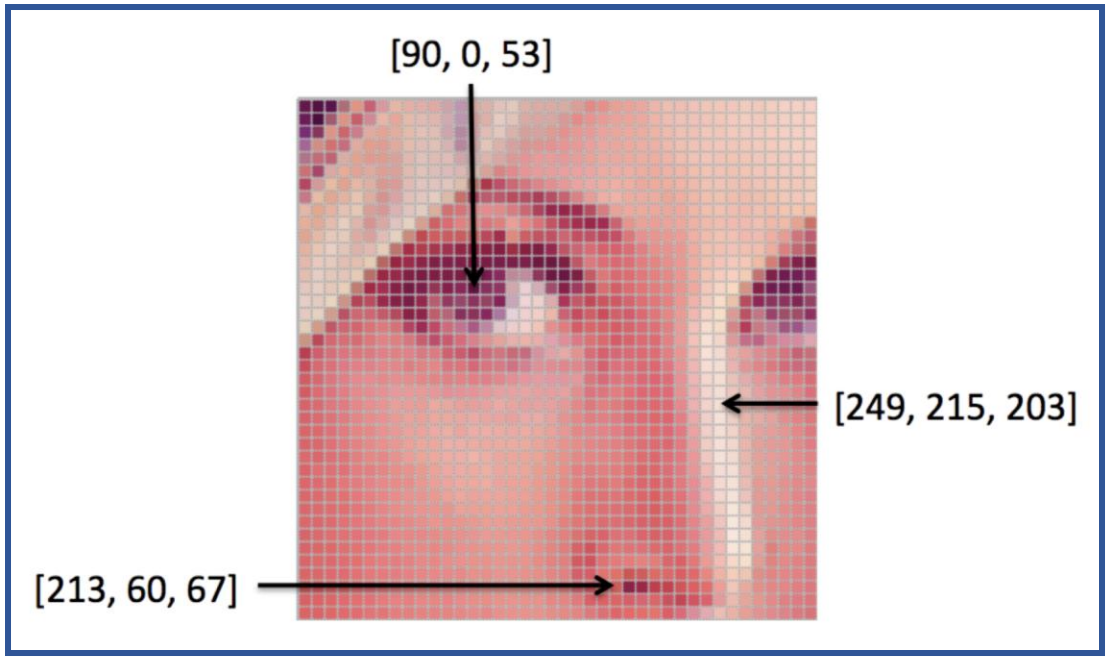
[Source: Bishop 2023 DL Book]

Structure of Image Data

- An image can be represented as a **matrix of pixel values**.
- Each pixel has an **intensity** between [0, 255].



Grey-scale image

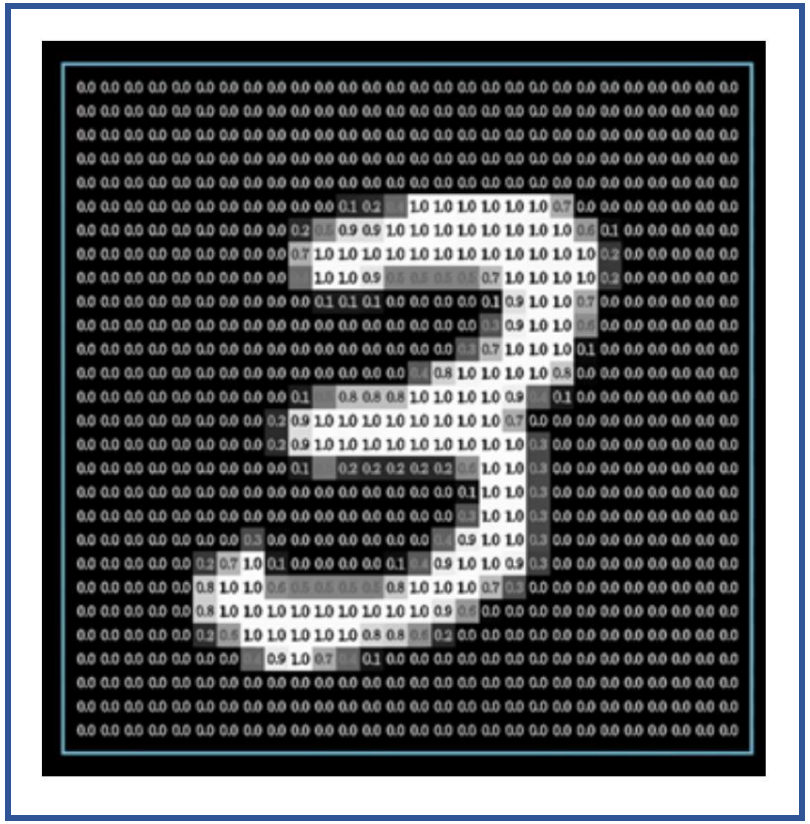


RGB image (three channels)

[Image from https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html]

Structure of Image Data

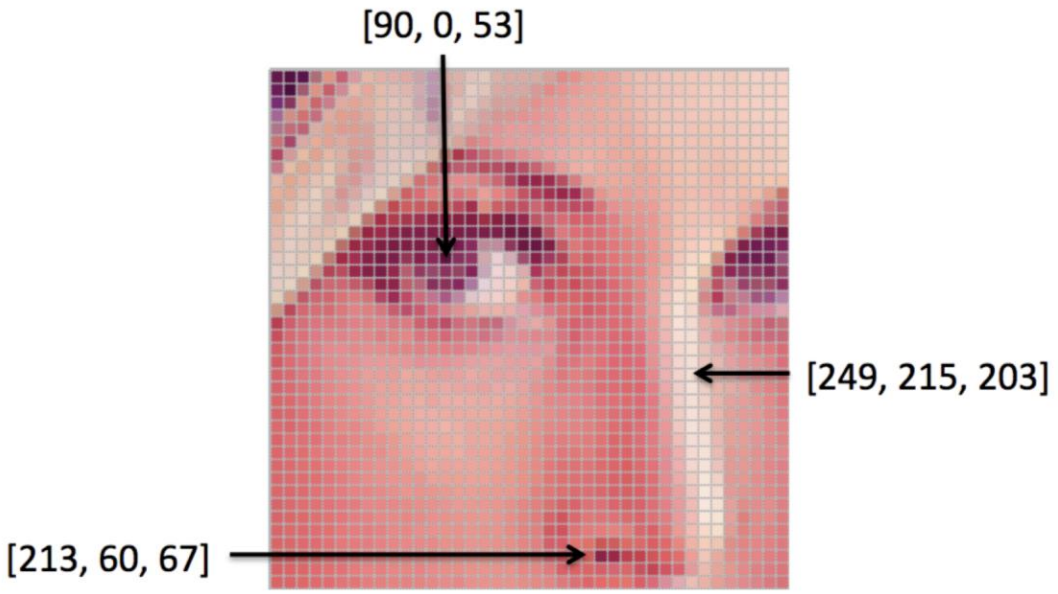
- Normalized pixel values between (0, 1), divide by 255.



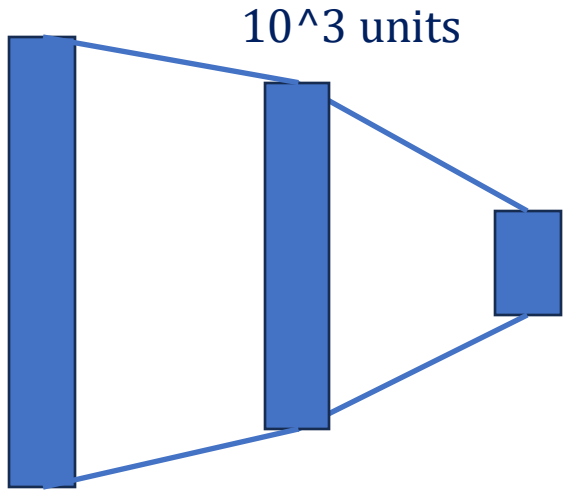
[Source: Kaggle]

Notebook

Structure of Image Data



10^3 x 10^3 Pixels, with RGB channels



Fully Connected Layer

- How many weights are in the first layer?

  3x10^9

- Also, need to learn invariances & equivariances.

  Need for Huge Datasets.

How to solve?

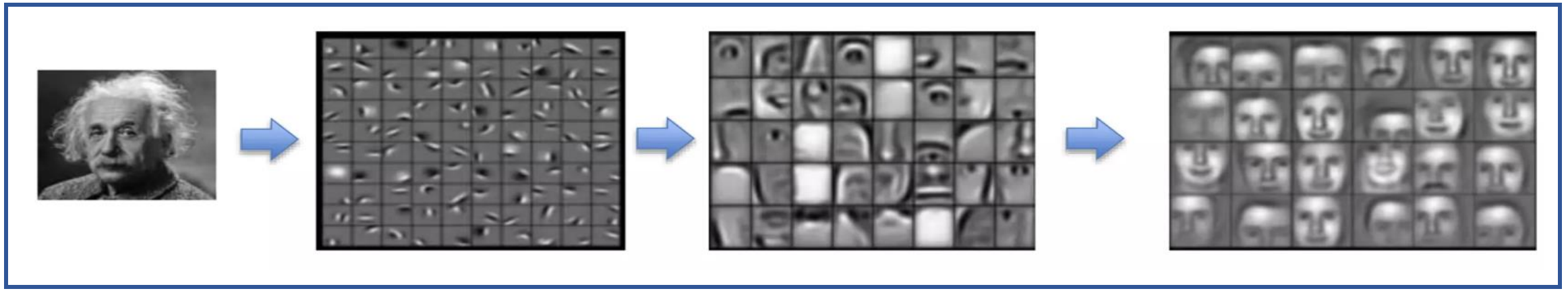
Inductive Bias!

What does inductive mean?

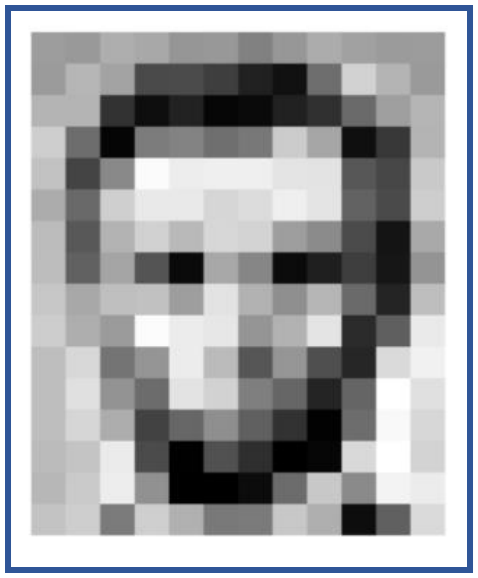
Inductive Vs Deductive?

Structure of Image Data

- We need an architecture that exploits the following key concepts:



Hierarchy



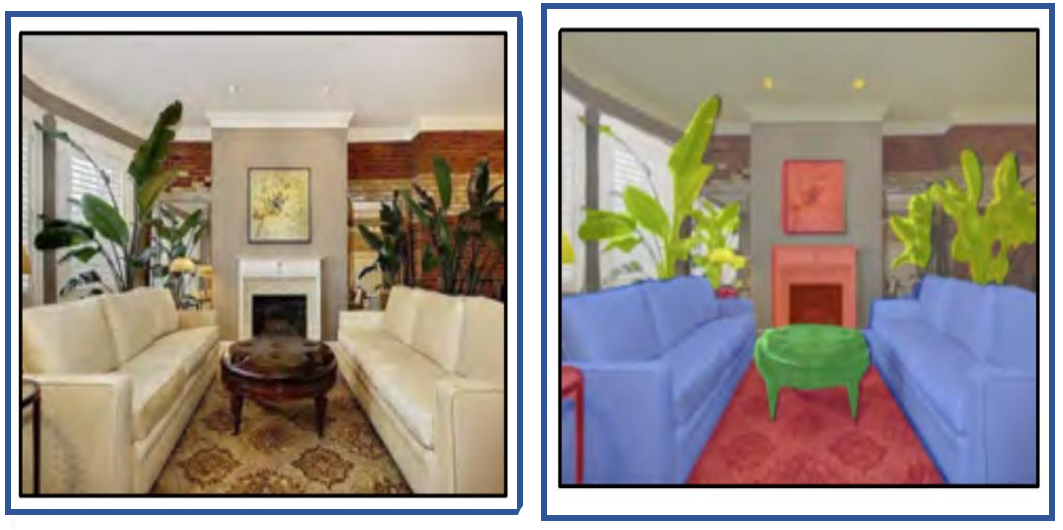
Locality

Structure of Image Data

Invariant to Translation

What's in the images?

Are the images same?



Equivariant to Translation

[Image from Prince 2023 UDL Book]

Structure of Image Data

- We need an architecture that exploits the following key concepts:
  - Hierarchy
  - Locality
  - Invariant to translation.
  - Equivariant to translation.

Convolutional Neural Networks

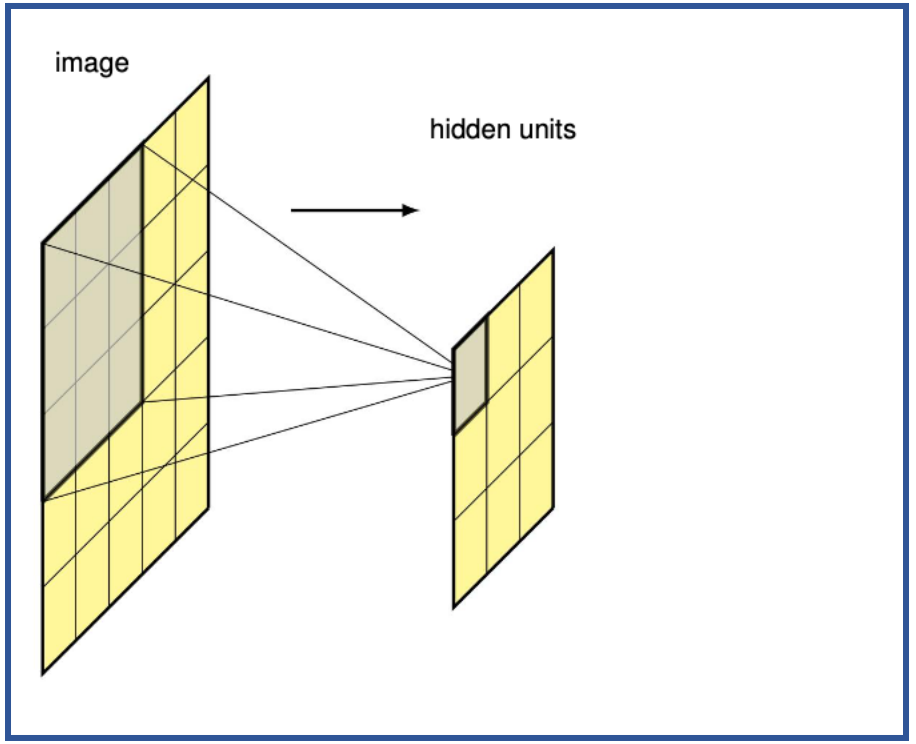
Convolutional Neural Network

❑ Feature Detector

- Notion of Locality.
- Weights to learn low-level features.

$$z = \mathrm{ReLU}(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0)$$

Receptive Fields



What does this capture?

[Source: Bishop 2023 DL Book]

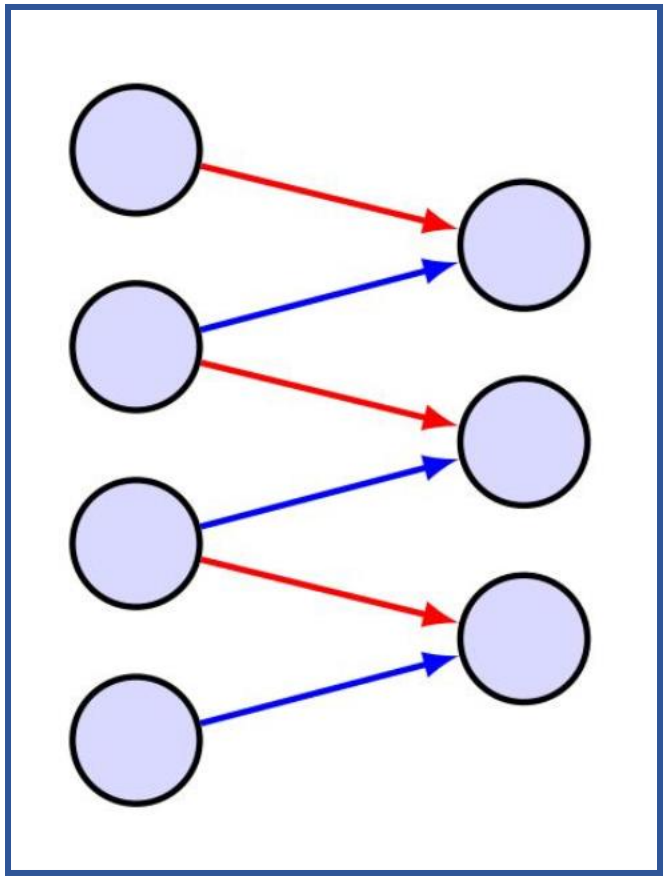
Convolutional Neural Network

❑ Translation Equivariance

- For example, **eye pixels** in one location should be the **same** in other locations.
- **Generalize** concepts learned in one patch of an image to other parts.

How do we do that?

- Use the **same weight** parameters across different parts of image.
- Connections are sparse, weights are shared by all hidden units.



Convolution

[Image from Bishop 2023 DL Book]

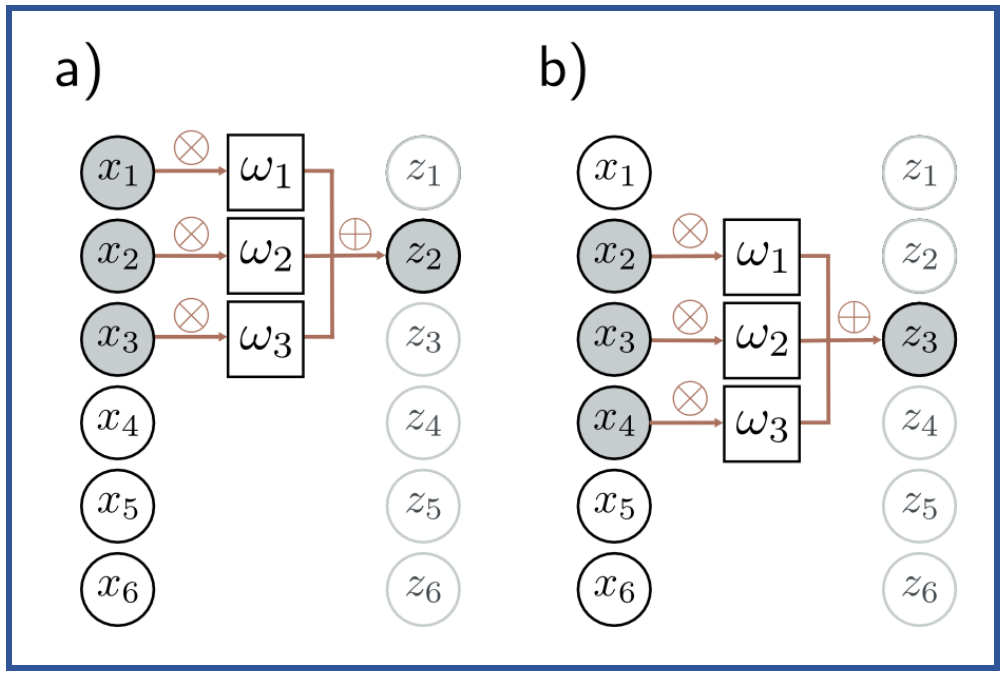
Convolutional Neural Network

❑ Convolutional operation for 1D input:

$$z_i = \omega_1 x_{i-1} + \omega_2 x_i + \omega_3 x_{i+1}$$

$$\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^T$$

Convolution Kernel or Filter



[Source: Prince 2023 UDL Book]

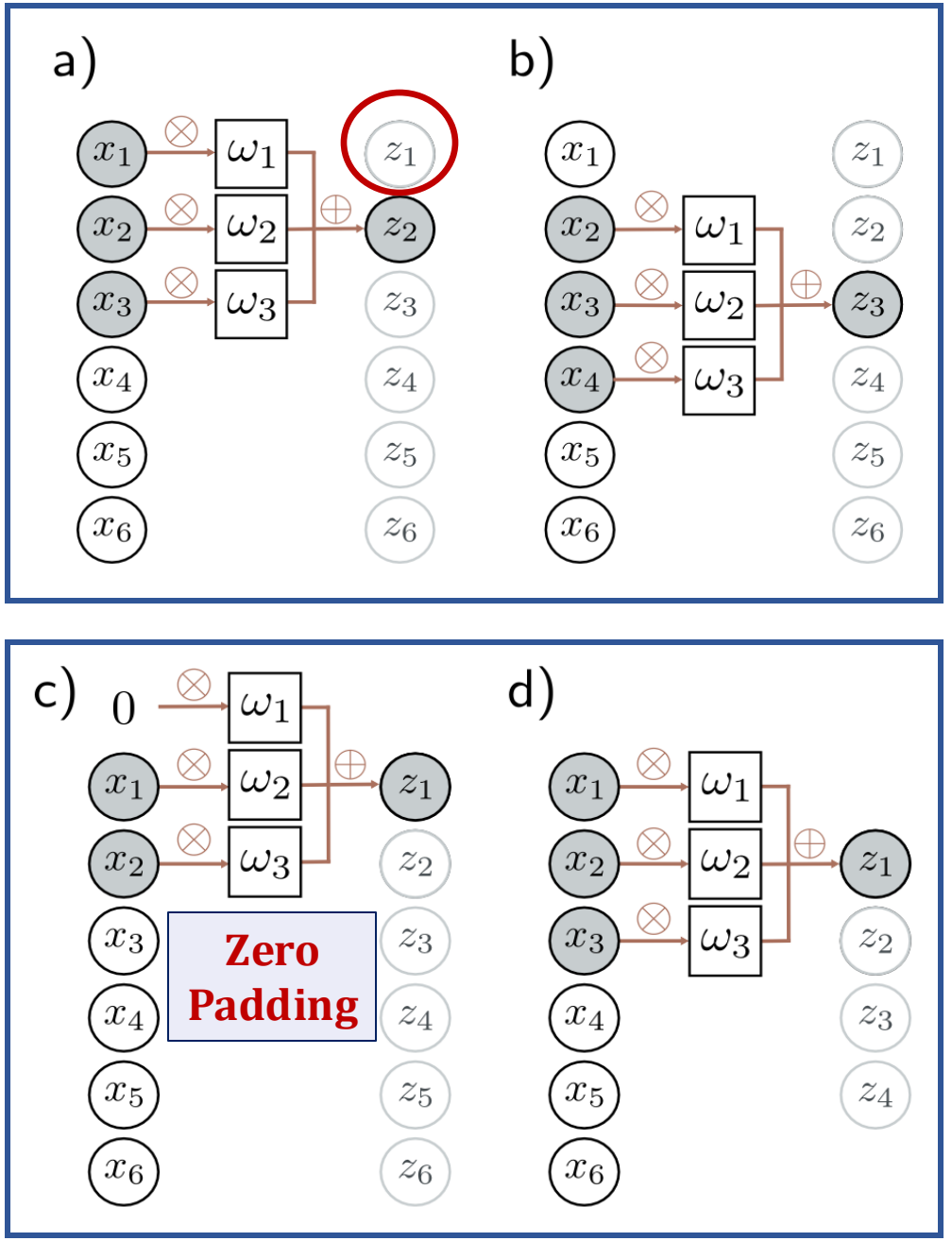
Convolutional Neural Network

❑ Convolutional operation for 1D input:

$$z_i = \omega_1 x_{i-1} + \omega_2 x_i + \omega_3 x_{i+1}$$

$$\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^T$$

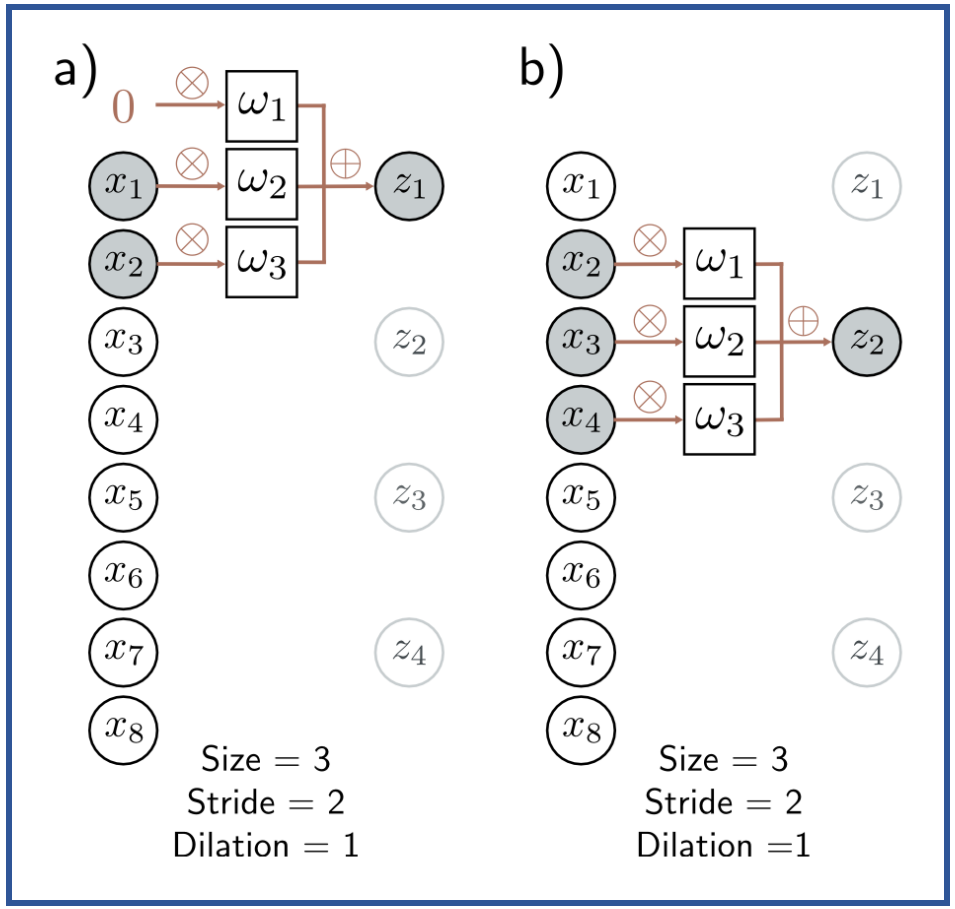
Convolution Kernel or Filter

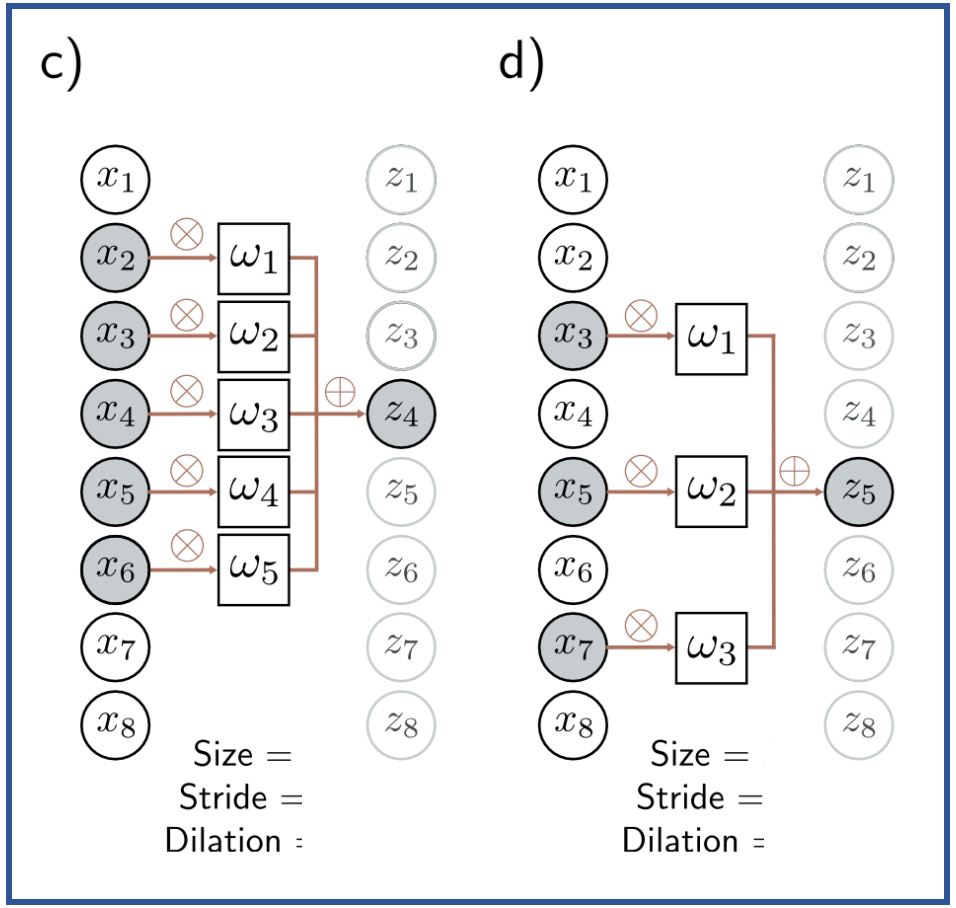


[Image from Prince 2023 UDL Book]

Convolutional Neural Network

❑ Kernel size, Stride and Dilation





[Image from Prince 2023 UDL Book]

Convolutional Neural Network

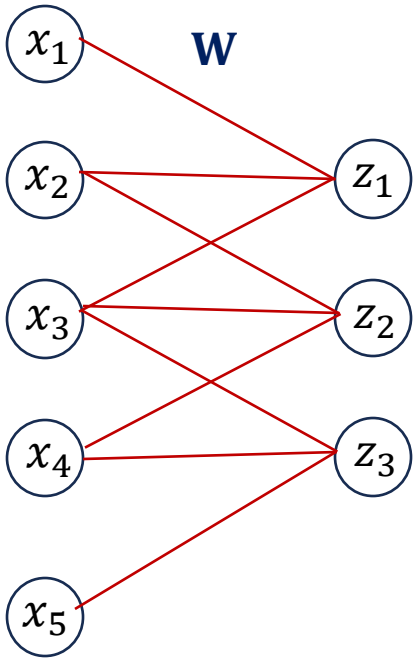
❑ Convolutional Layer

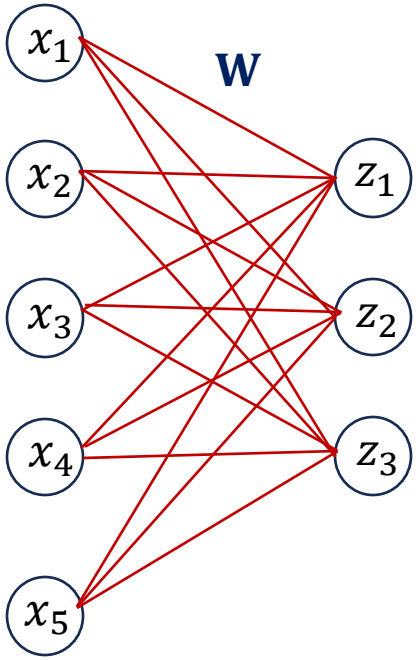
$$
\begin{aligned}
h_i &= \mathrm{a}\left[\beta + \omega_1 x_{i-1} + \omega_2 x_i + \omega_3 x_{i+1}\right] \\
&= \mathrm{a}\left[\beta + \sum_{j=1}^{3} \omega_j x_{i+j-2}\right],
\end{aligned}
$$

[Image from Prince 2023 UDL Book]

Convolutional Neural Network

❑ Fully connected layer vs Convolutional layer

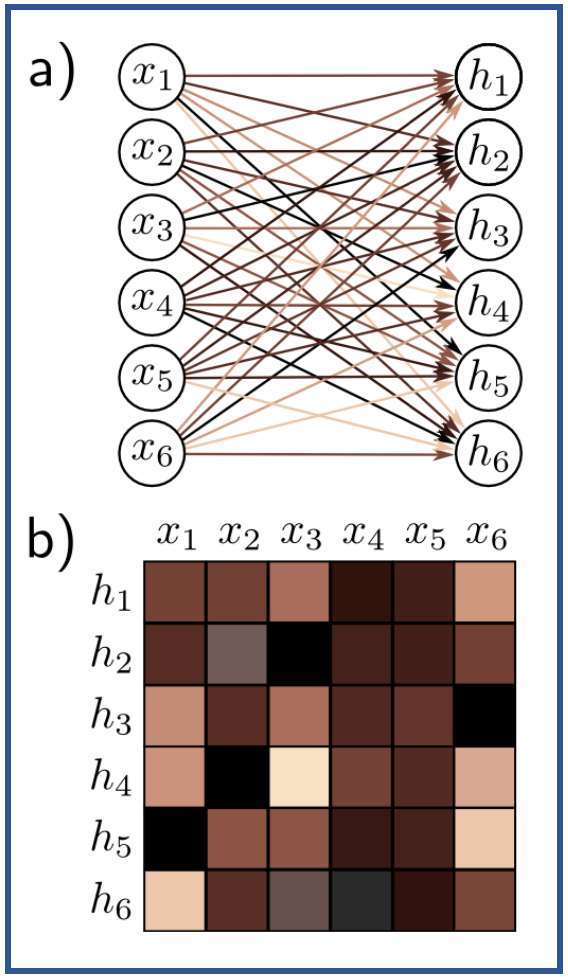


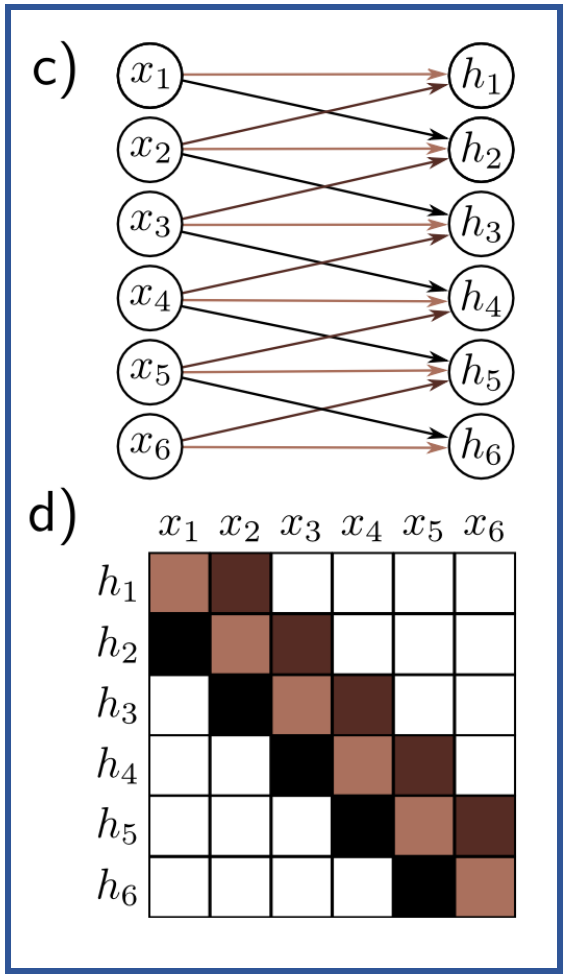


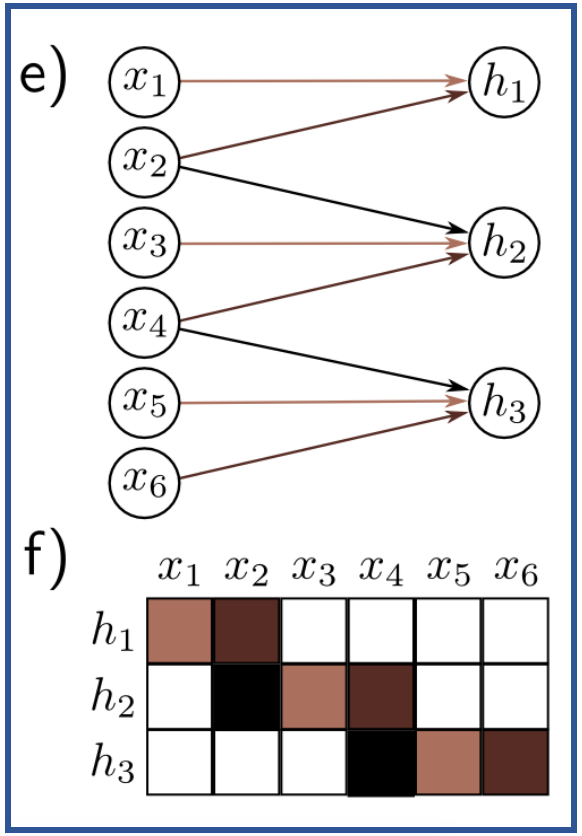
[Image from Prince 2023 UDL Book]

Convolutional Neural Network

❑ Fully connected layer vs Convolutional layer







Kernel size and stride?

[Image from Prince 2023 UDL Book]

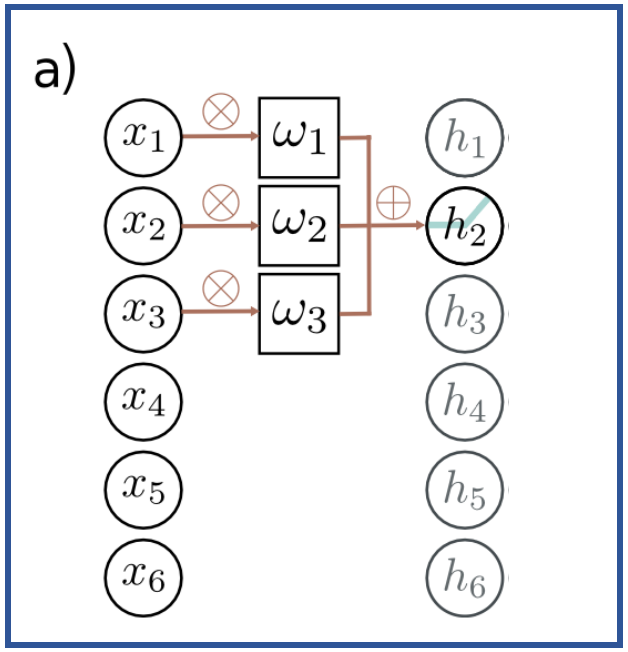
Convolutional Neural Network

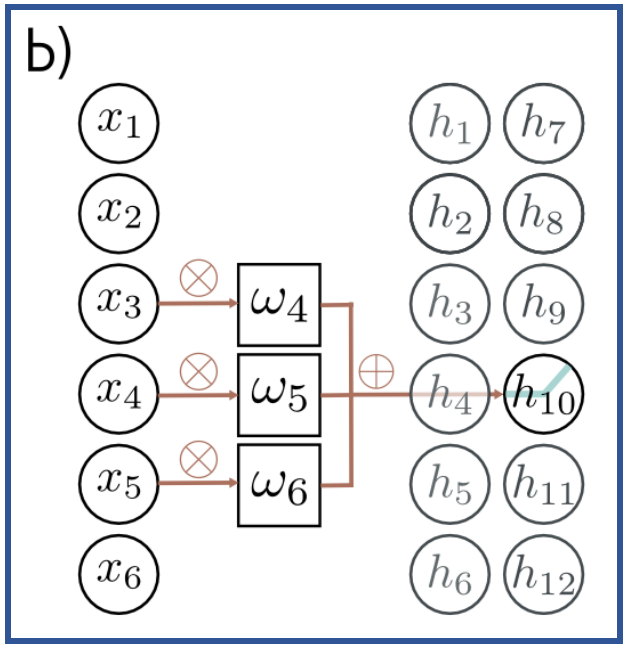
❑ Channels:

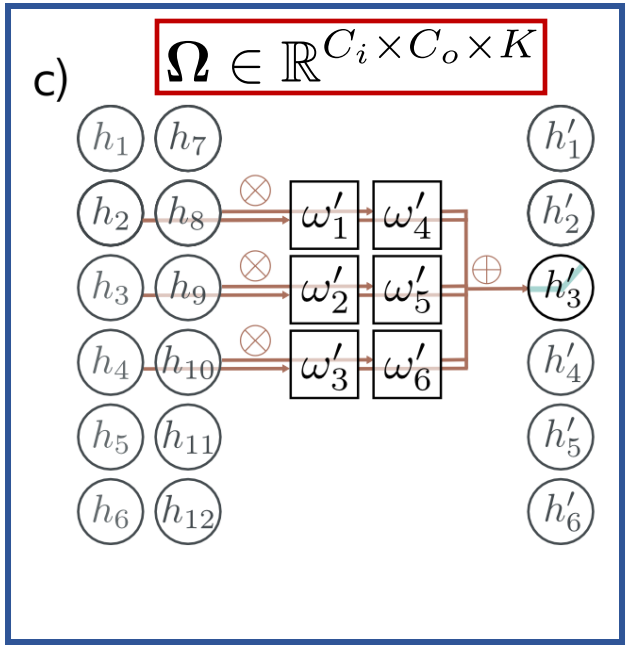
- Using **single** convolution is not effective in practice.
- Convolution involves **averaging** of nearby inputs.
- **ReLU** activation **clips** results that are **less than zero**.

C_i Channels Kernel size K per channels

C_o Channels in next layer.

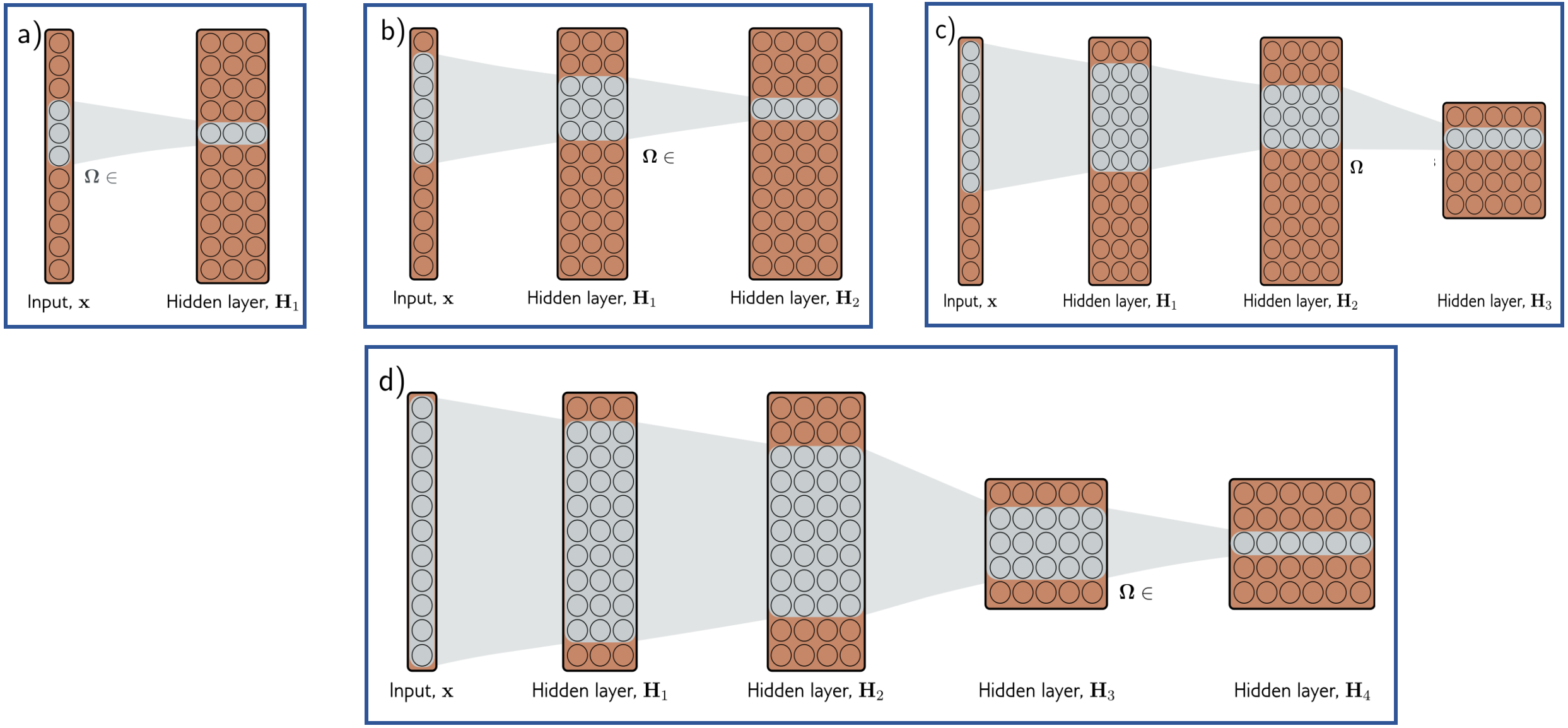






[Image from Prince 2023 UDL Book]

Convolutional Neural Network



[Image from Prince 2023 UDL Book]

Convolutional Neural Network

Input : 3 Channels

Kernel Size : 3

1st Hidden Layer : 4 Channels

Kernel Size : 5

2nd Hidden Layer : 10 Channels

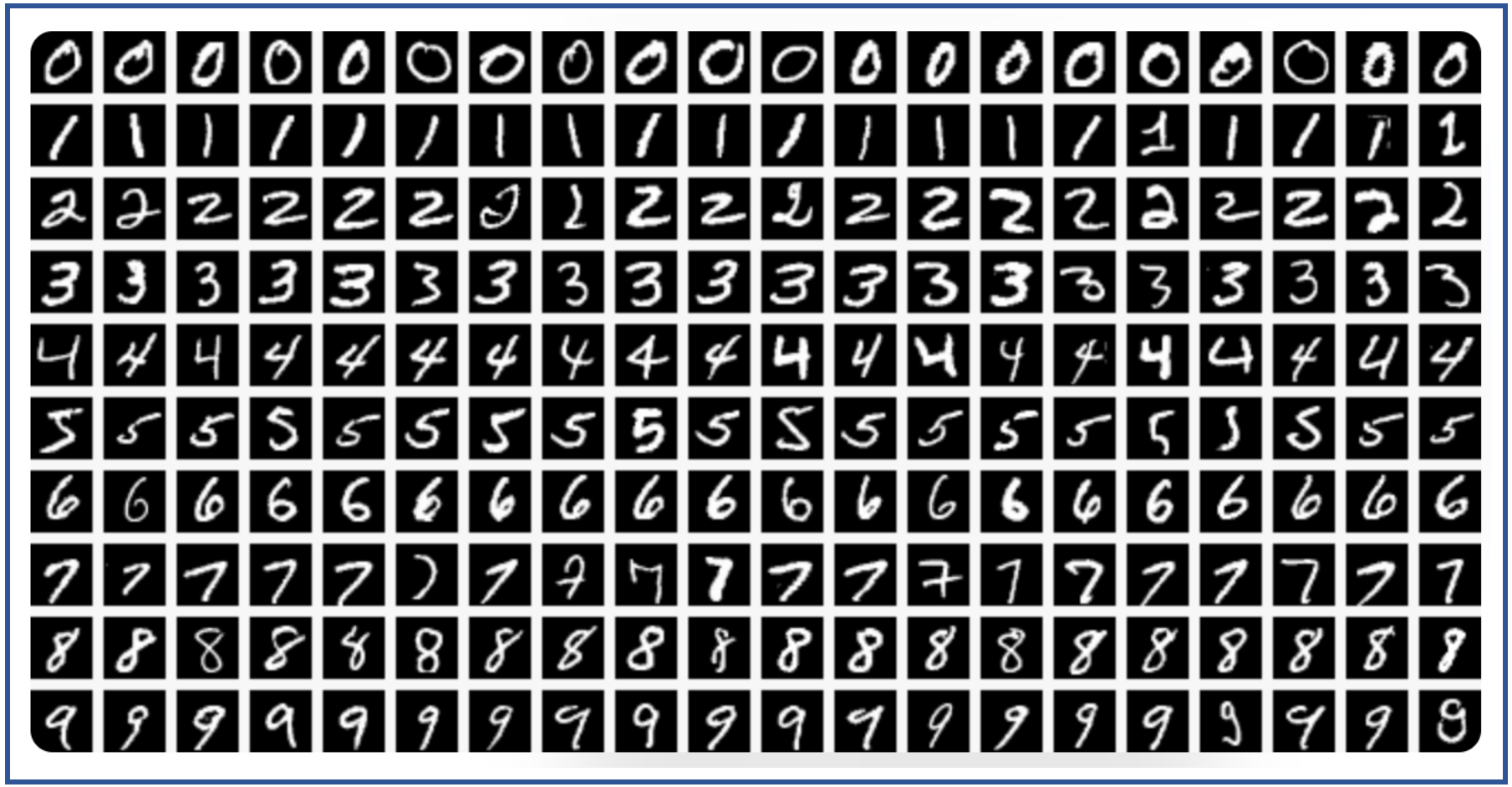
How many weights and biases for the two hidden layers?

[Image from Prince 2023 UDL Book]

Convolutional Neural Network

ScAI

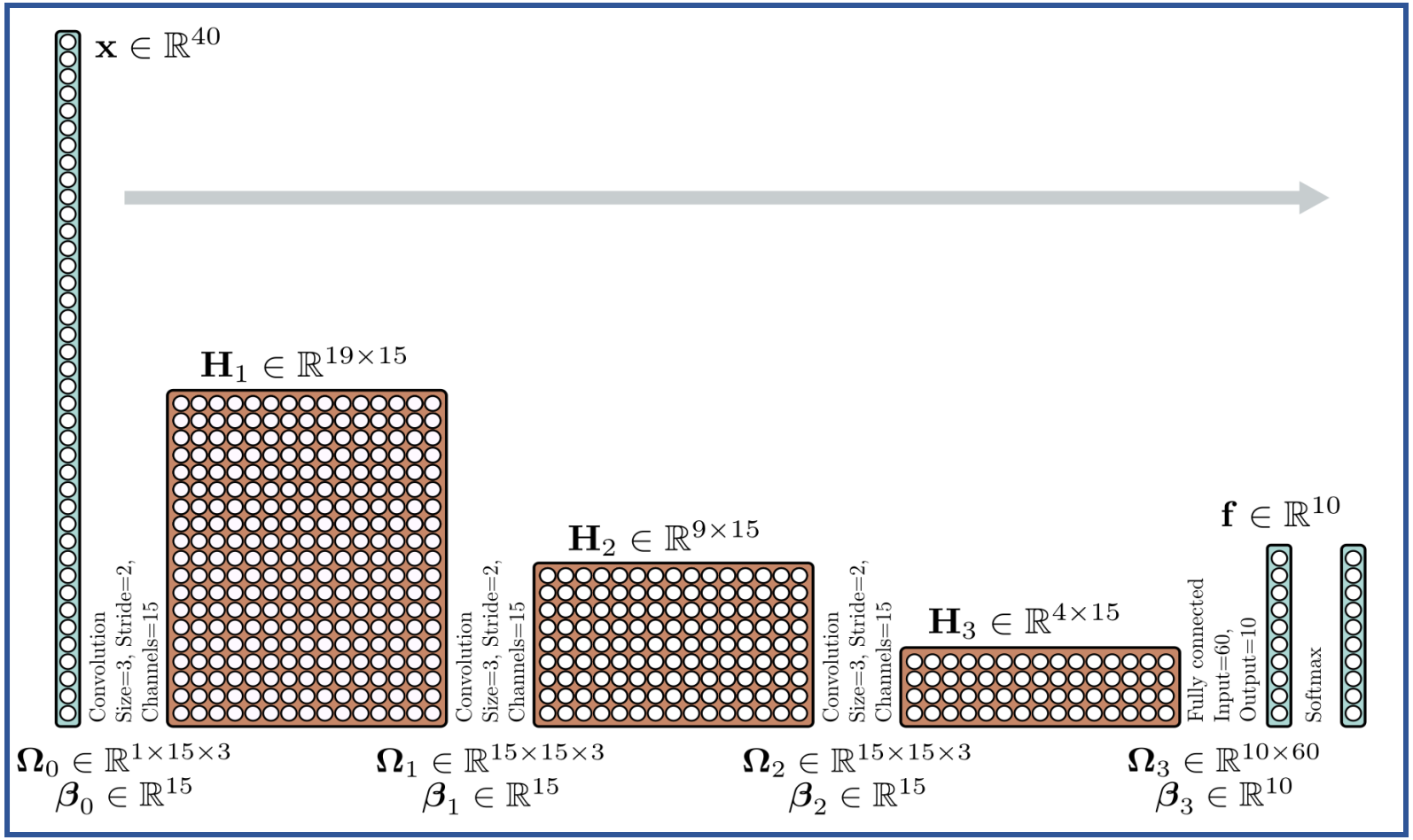
❑ MNIST-1D classification problem:



[Image from Wikipedia]

Convolutional Neural Network

❑ MNIST-1D classification problem:

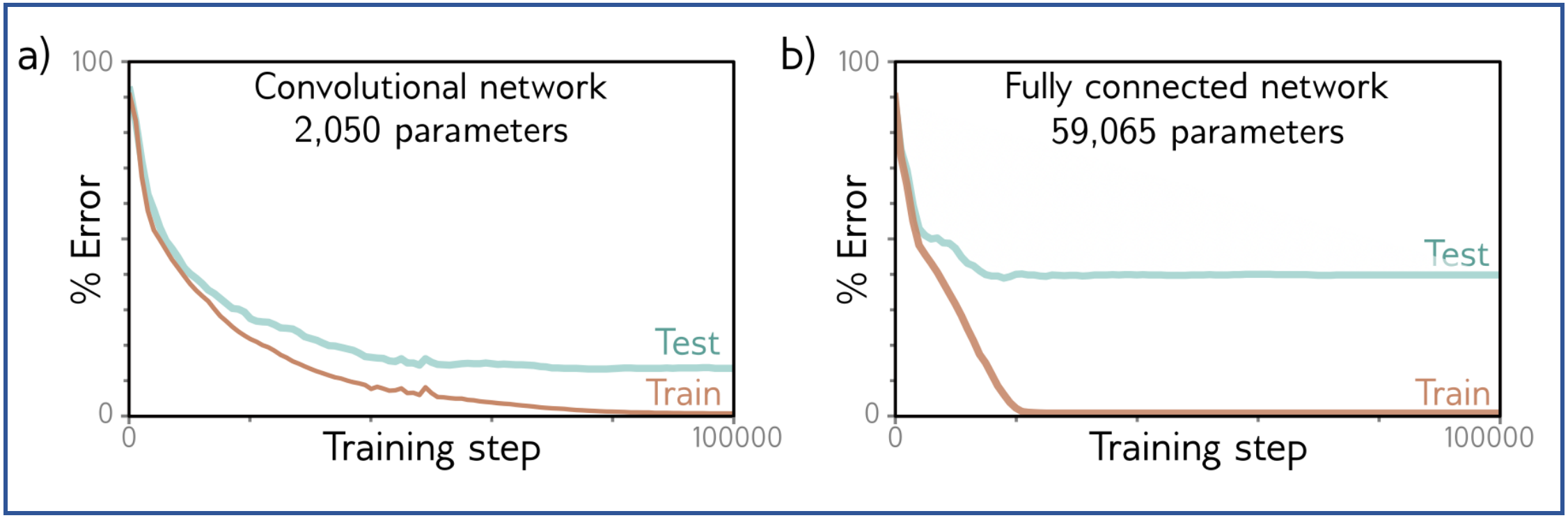


Convolutional Network

[Image from Prince 2023 UDL Book]

Convolutional Neural Network

❑ MNIST-1D classification problem:



Convolutional Network vs Fully connected Network

[Image from Prince 2023 UDL Book]

Convolutional Neural Network

❑ Convolutional layer for 2D image data

$$\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^T$$

1D Convolution Kernel or Filter

$$\begin{aligned} h_i &= \mathrm{a}\left[\beta + \omega_1 x_{i-1} + \omega_2 x_i + \omega_3 x_{i+1}\right] \\ &= \mathrm{a}\left[\beta + \sum_{j=1}^{3} \omega_j x_{i+j-2}\right], \end{aligned}$$

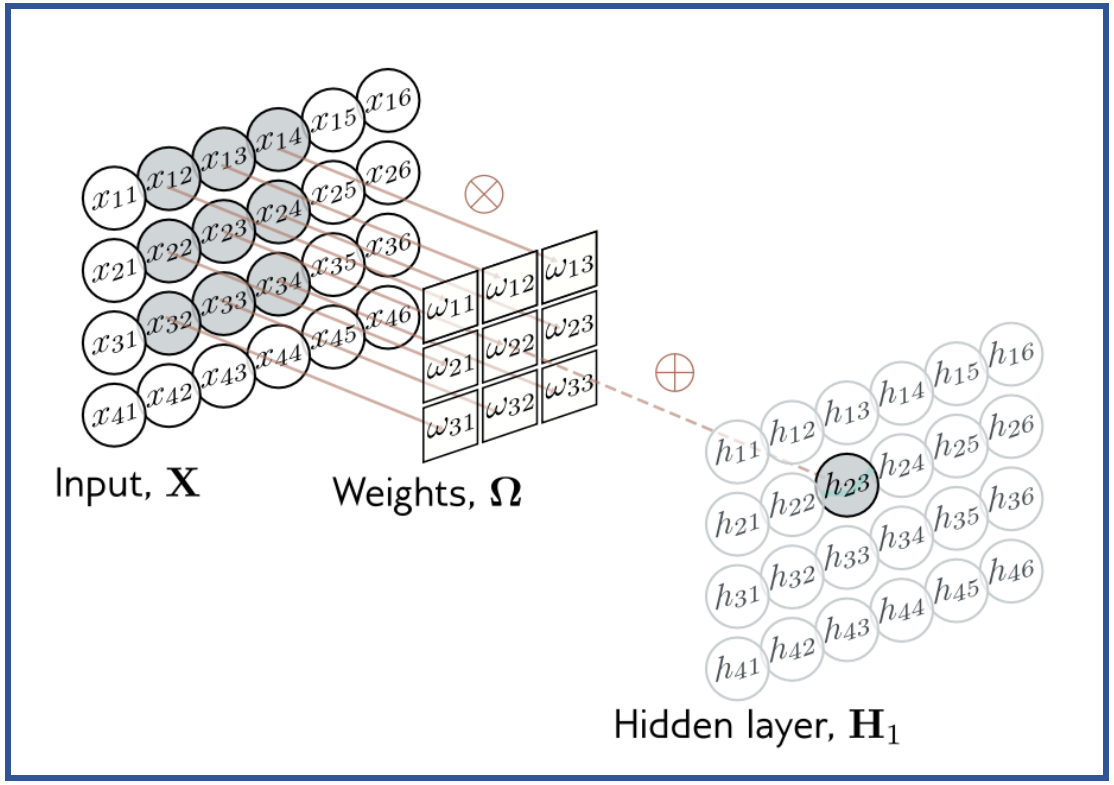
2D Convolution Kernel or Filter

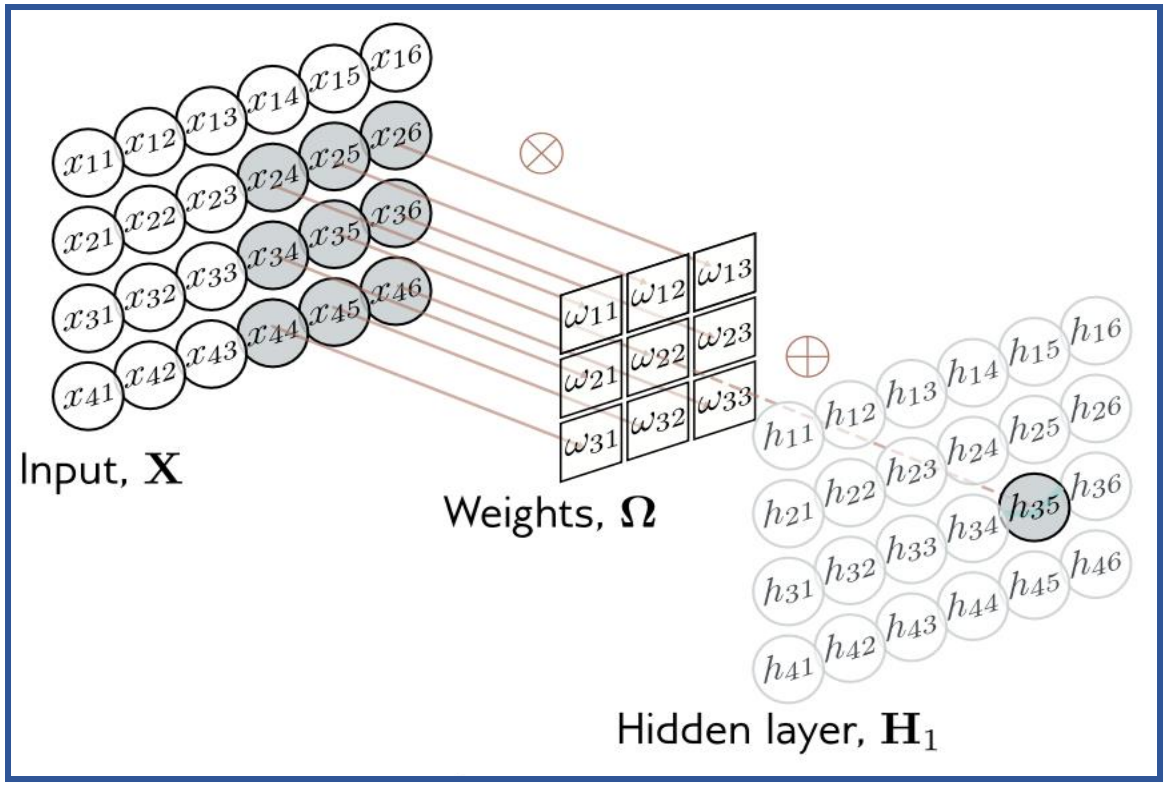
$$h_{ij} = \mathrm{a}\left[\beta + \sum_{m=1}^{3}\sum_{n=1}^{3} \omega_{mn} x_{i+m-2,j+n-2}\right]$$

[Image from Prince 2023 UDL Book]

Convolutional Neural Network

❑ Convolutional layer for 2D image data

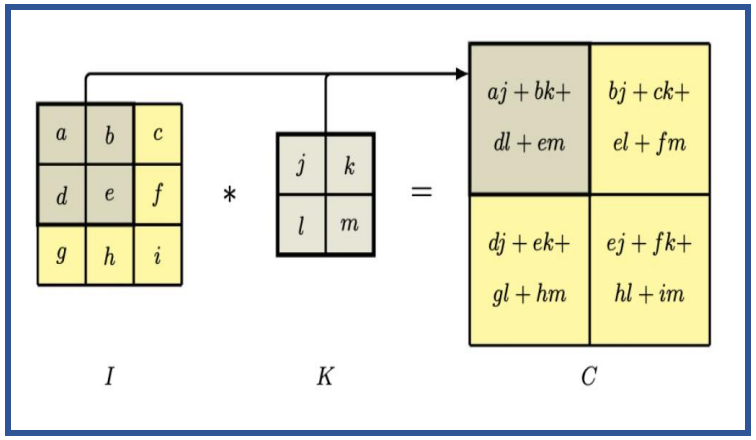


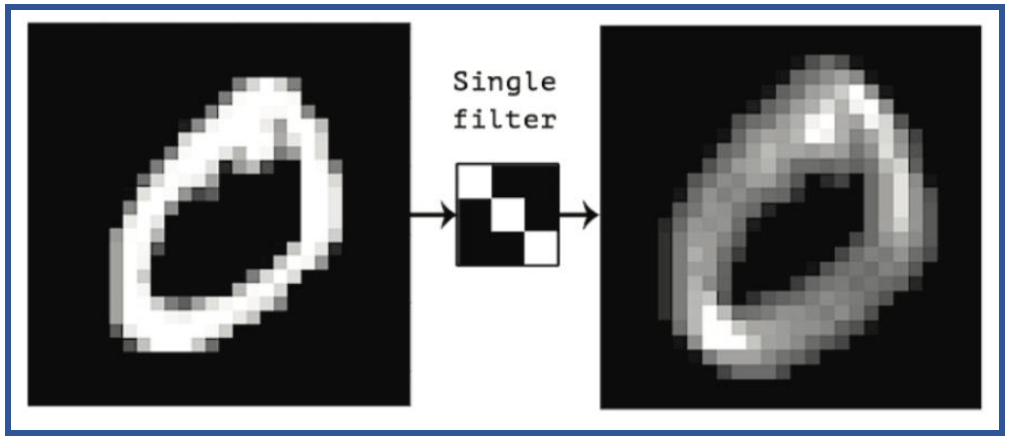


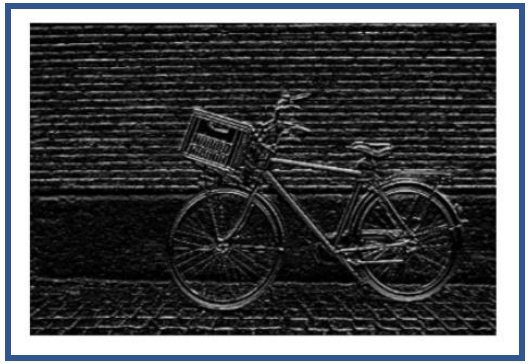
[Image from Prince 2023 UDL Book]

Convolutional Neural Network

❑ Convolutional layer for 2D image data



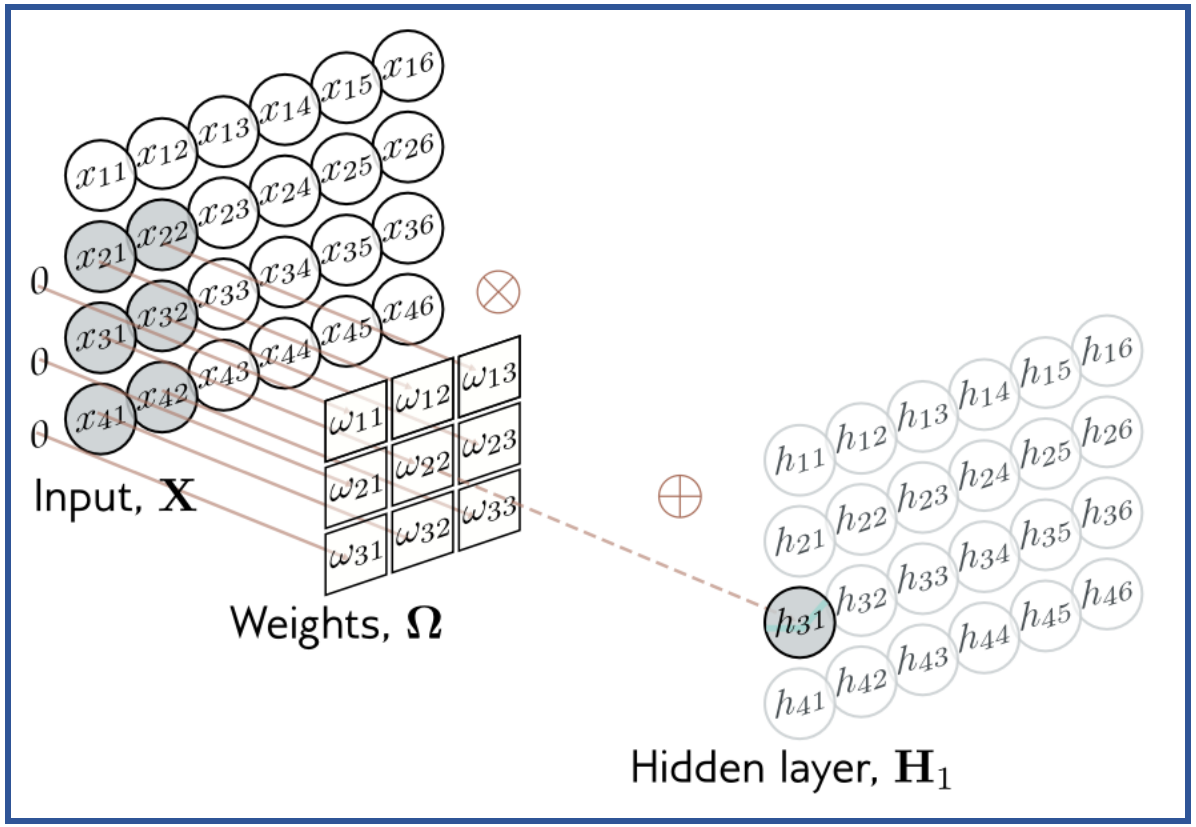


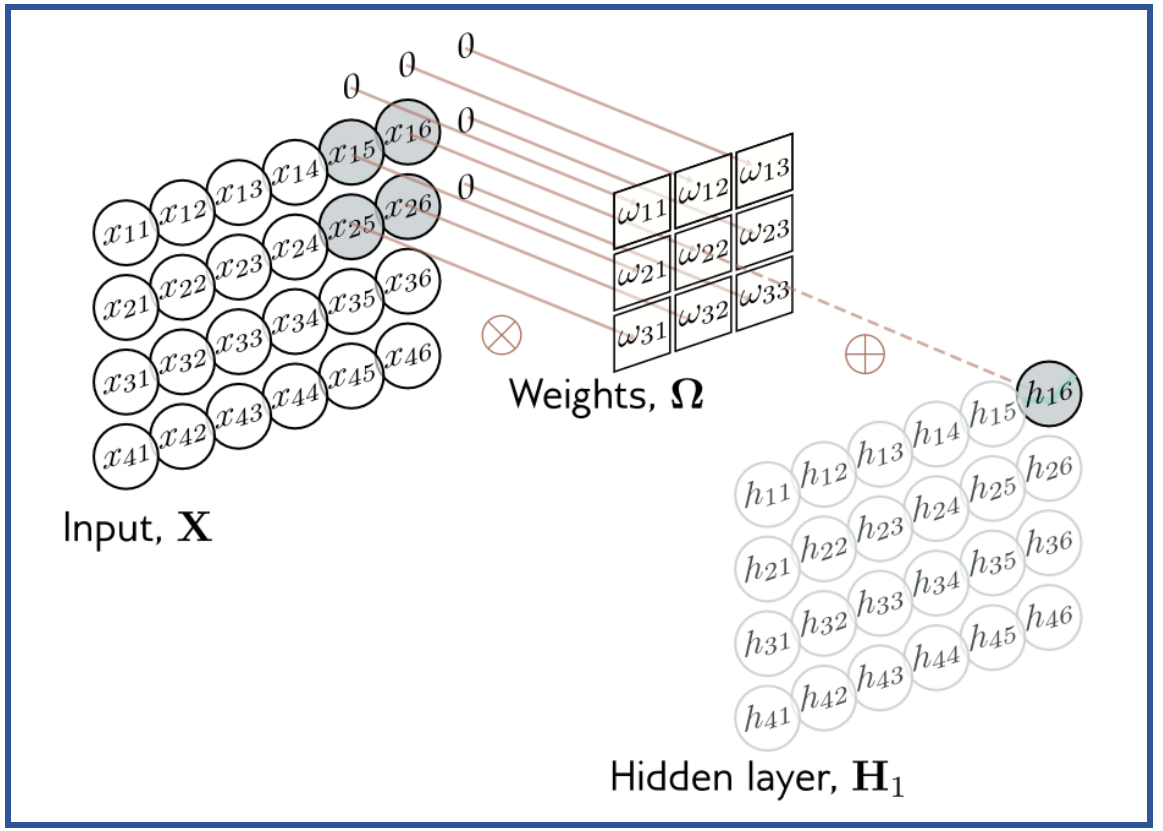


[Source: Bishop DL Book & K.Murphy ML Book]

Convolutional Neural Network

❑ Convolutional layer for 2D image data

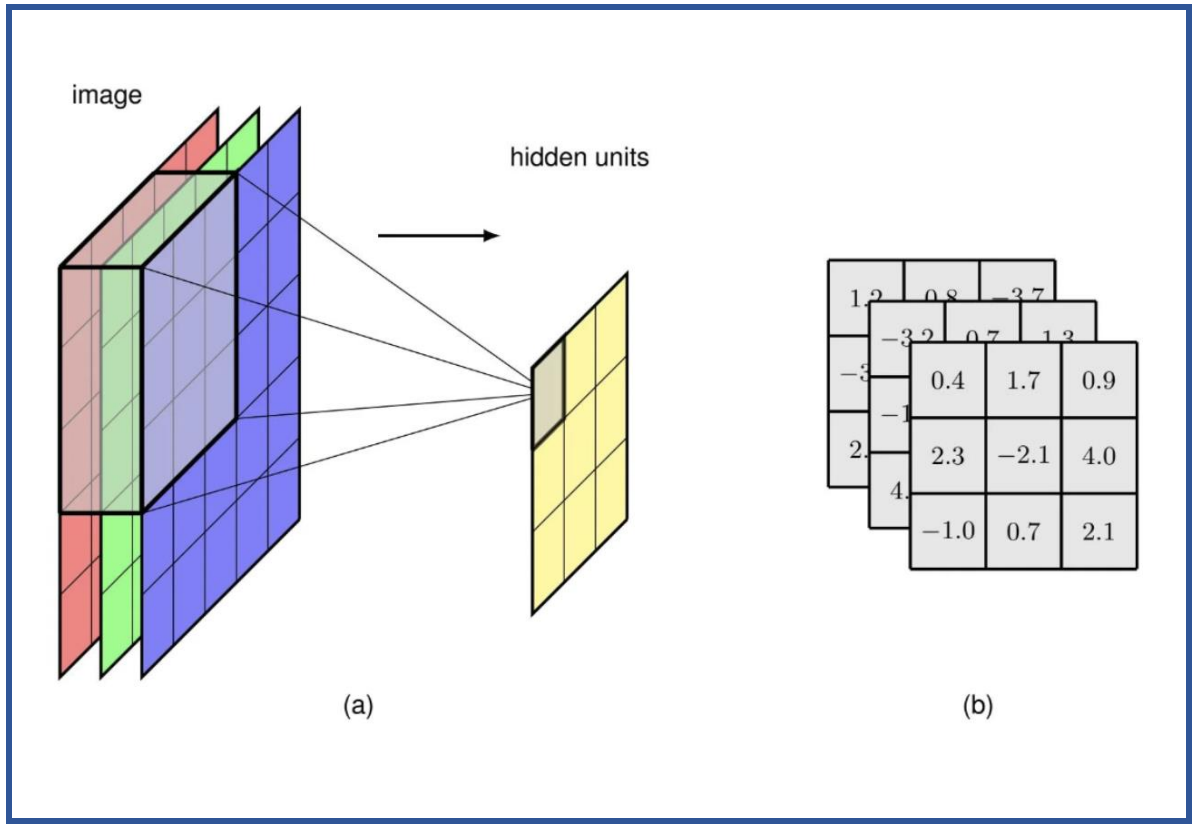


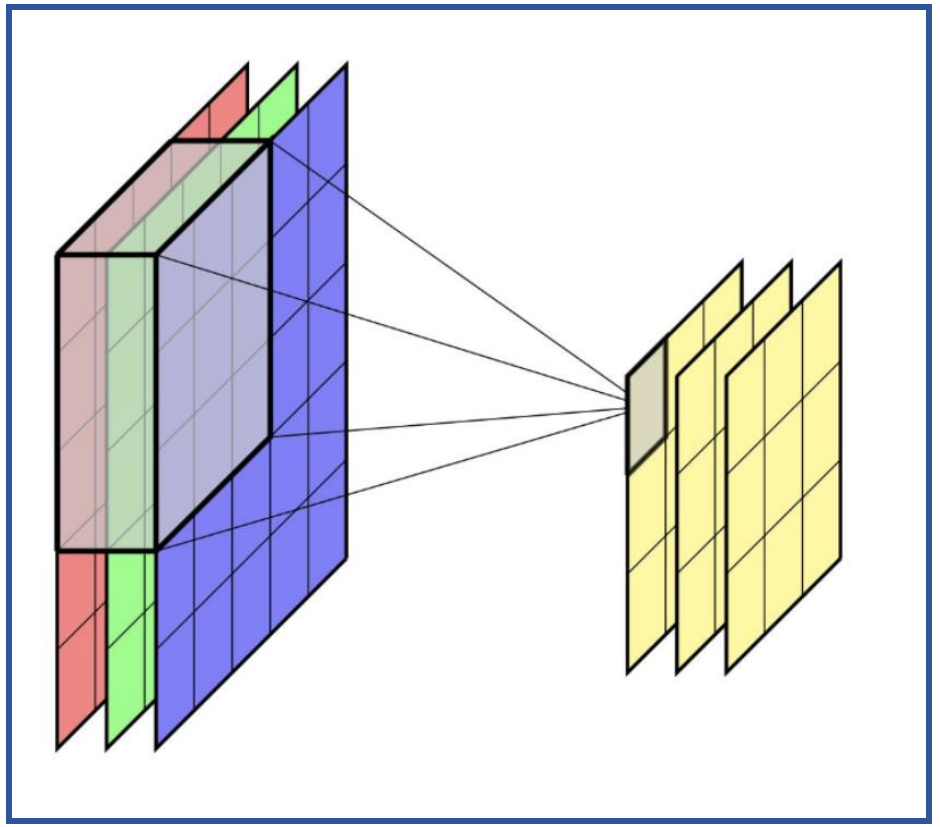


[Image from Prince 2023 UDL Book]

Convolutional Neural Network

❑ Multi-dimensional Convolutions

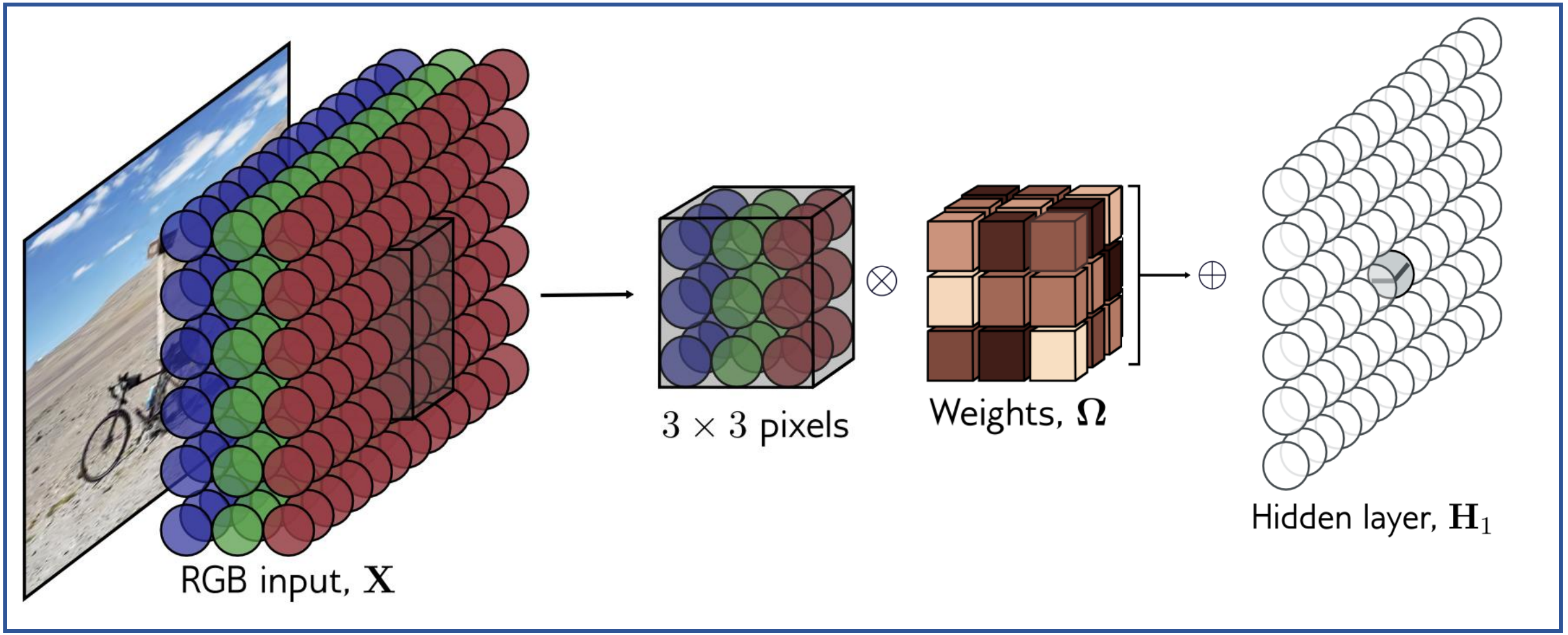




[Source: Bishop Book]

Convolutional Neural Network

❑ Convolutional layer for 2D image data

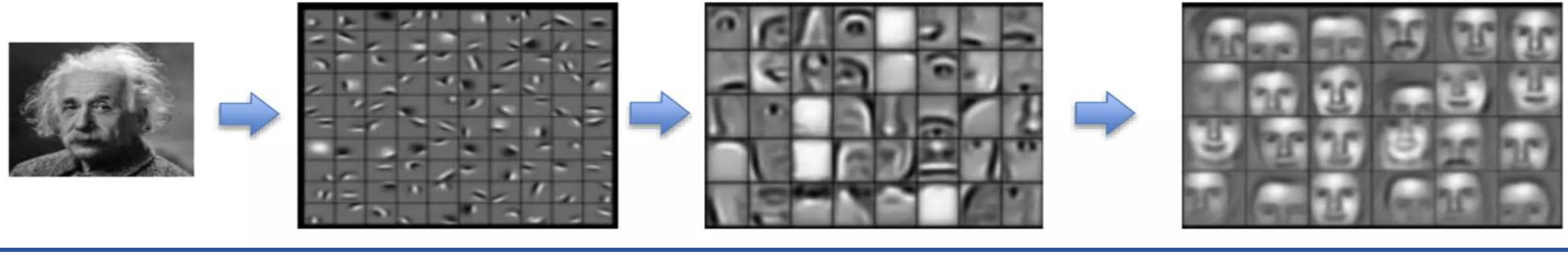


[Image from Prince 2023 UDL Book]

Convolutional Neural Network

❑ Translation Invariance

Hierarchical Structure

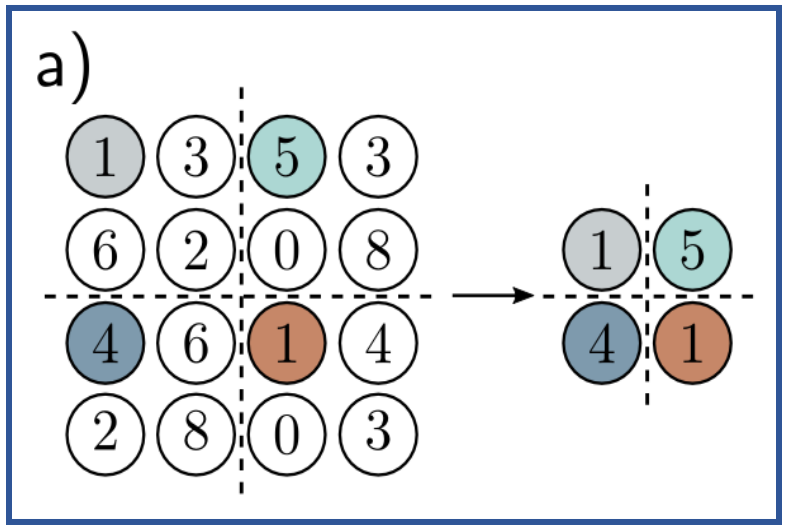


- **Complex features** are **learned** from **simple features** in previous layers.
- **Spatial relationships** between those simple features are **crucial**.
  - The relative position between the eyes, nose, and mouth helps determine the presence of a face.
- We want to be **invariant** to **small changes** in the relative position of features.

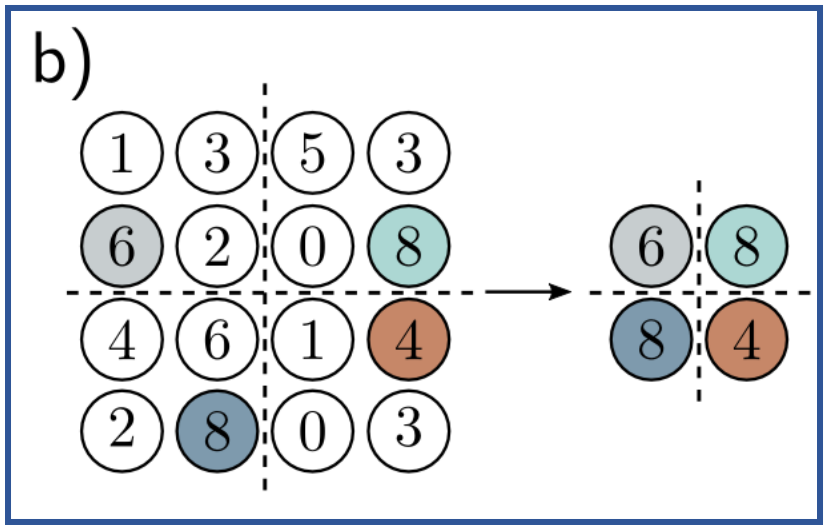
How do we do this?

Convolutional Neural Network

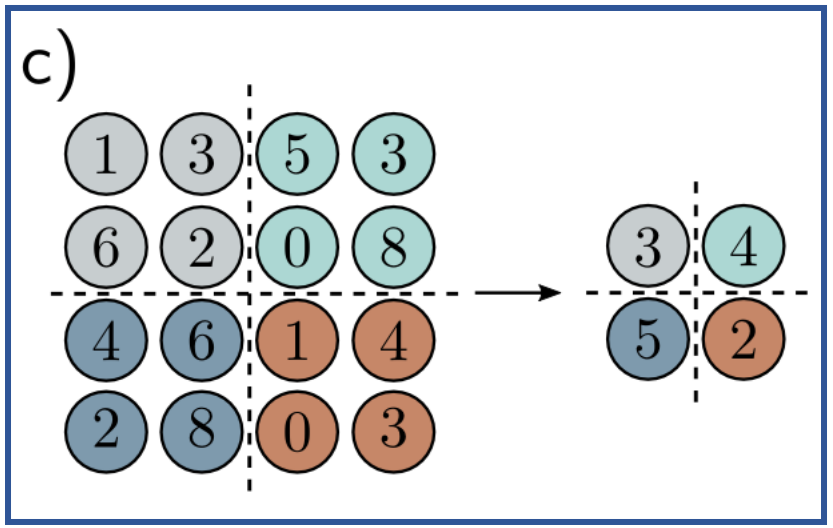
❑ Pooling



Sub-Sampling



Max Pooling



Mean Pooling

How did it solve the invariance issue?

[Image from Prince 2023 UDL Book]

End