Convolutional Neural Network - CNN

Outline

1. Why convolutional neural network
2. Comparison of ANN with CNN in a simple example
3. The structure of convolutional neural network
    - What is filter or convolution
    - What is stride
    - What is padding
    - What is pooling (Max and Average pooling)
4. Looking at different CNN structure (i.e. AlexNet, GoogleNet, ResNet, …)
5. How to code convolutional neural network in python
6. Hands on project
    - What is transfer learning
    - Apply convolutional neural network for image classification

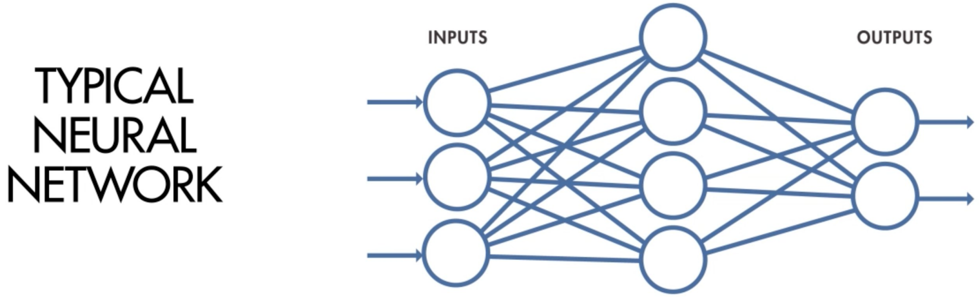
1. Why convolutional neural network

- Convolutional neural networks (CNNs) inspired by the study of the brain's visual cortex.
- Huble and Wiesel (1958) showed that many neurons in the visual cortex have a small local receptive field . More specifically they showed that some neurons react only to images of horizontal lines while others react only to lines with different orientations.
- They also showed some neurons might have larger receptive fields and they react to more complex patterns that are combinations of the lower-level patterns.
- CNNs has been used in image recognition since the 1980s.
- Image search service, self-driving cars, automatic video classification systems.

1. Why convolutional neural network

- Why not simply use a deep neural network with fully connected layers for image recognition task?`
  - This might work for small images
  - It breaks down for large images because of the huge number of required parameters
- As an example:
  - Suppose we have a 100 by 100 – pixel image that has 10,000 pixels.
  - The input layer should have 10,000 neurons
  - If we only have one hidden layer with a 1000 neurons in it, then for the fully connected network with just one layer we have at least 10,000,000 connections. CNNs solve this problem using partially connected layers and weight sharing.

2. Comparison of ANN with CNN in a simple example



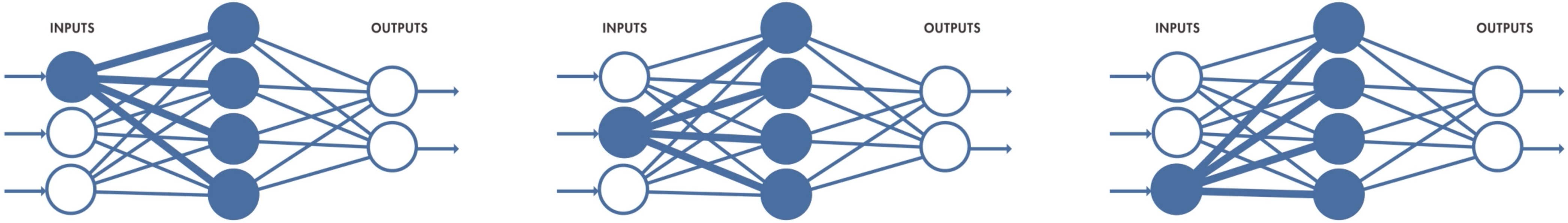


Image provided from MathWork

2. Comparison of ANN with CNN in a simple example

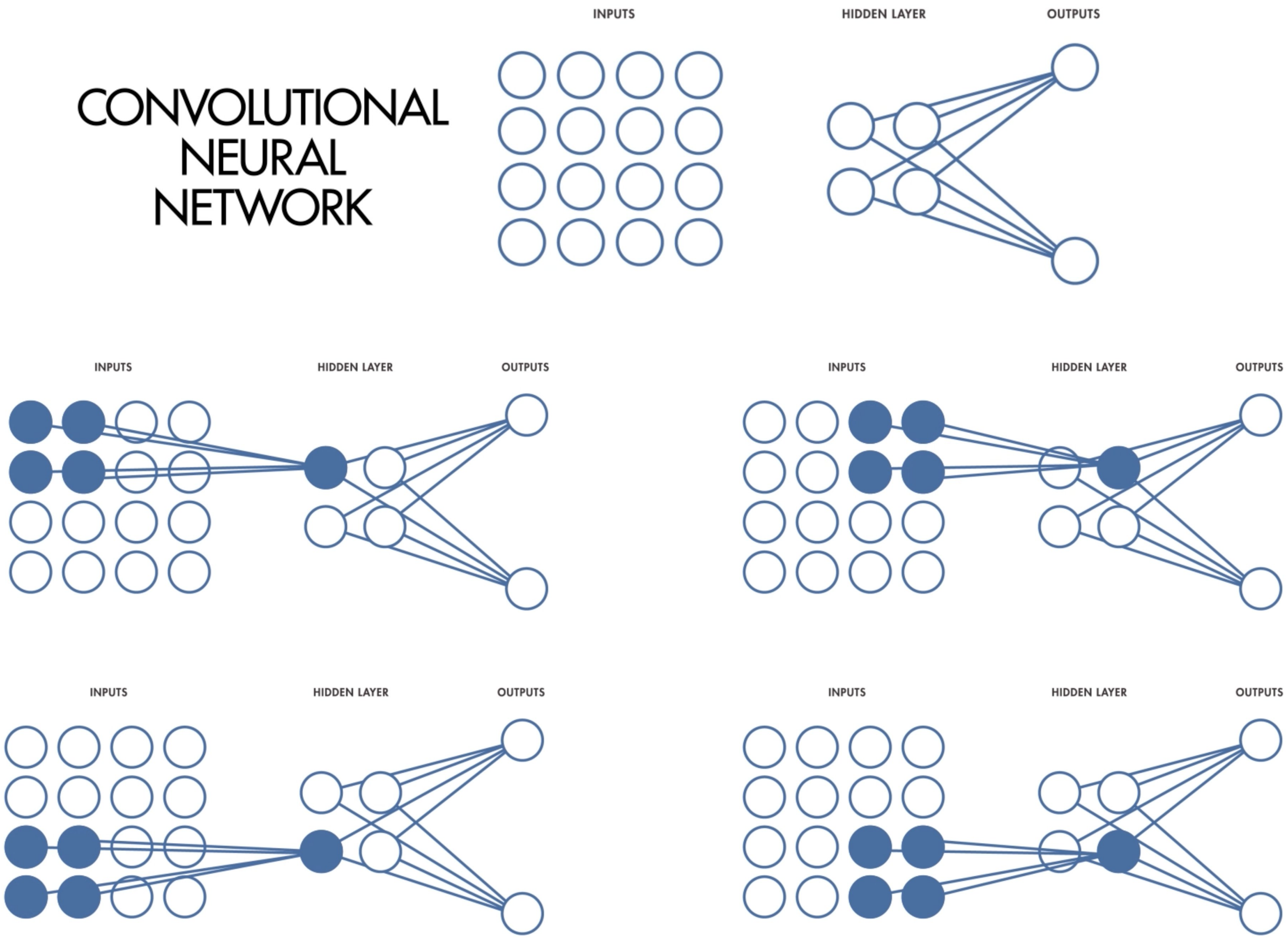
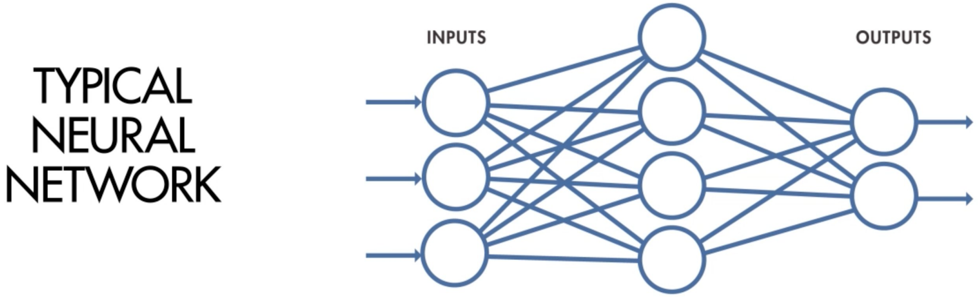


Image provided from MathWork

3. The structure of convolutional neural network

- ➢ Shared weights and biases - ANN



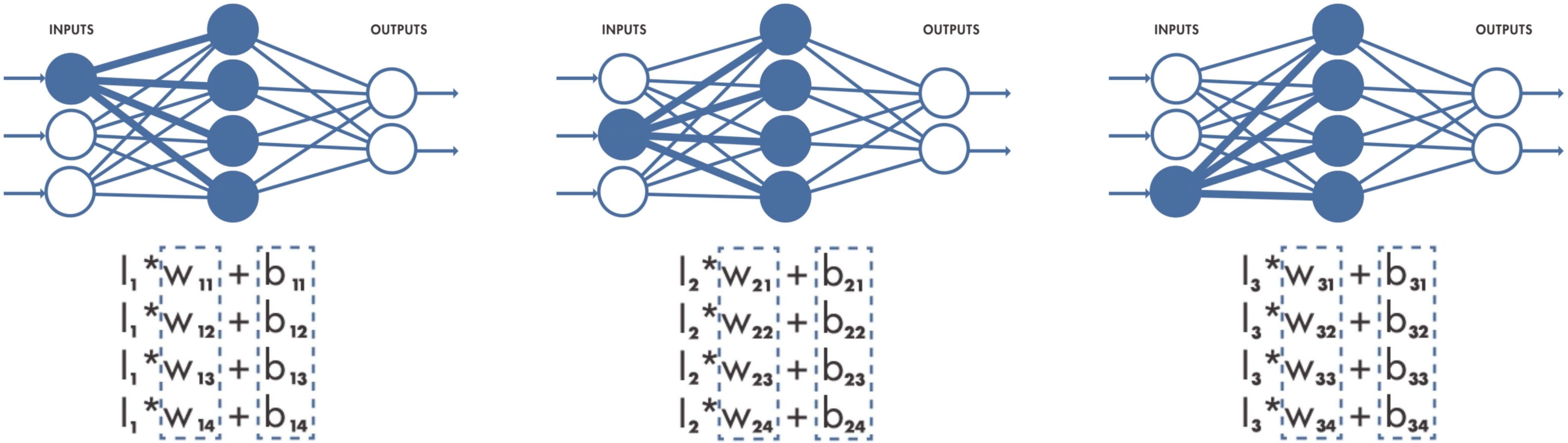
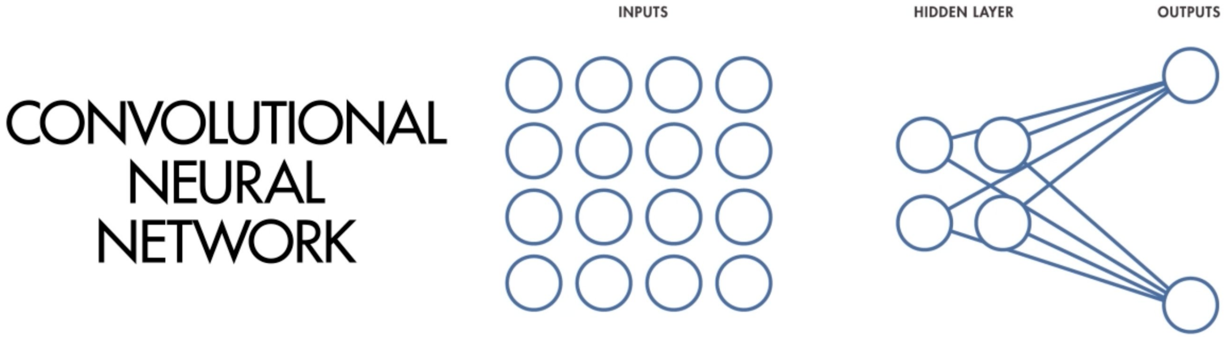
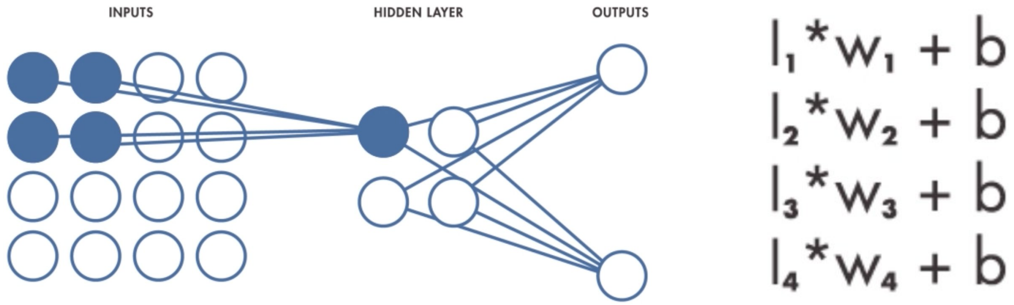


Image provided from MathWork

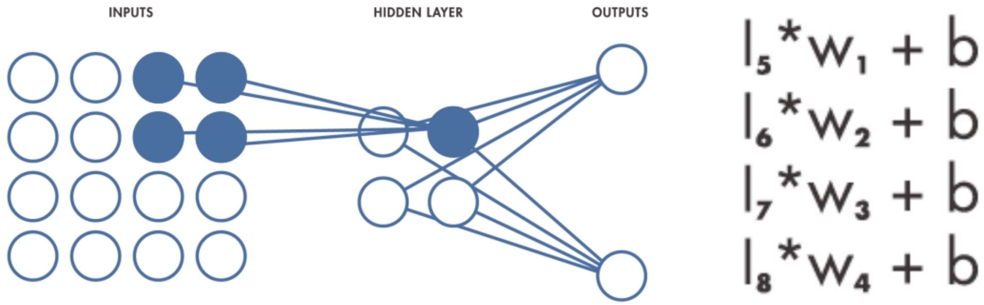
3. The structure of convolutional neural network

- Shared weights and biases - CNNs

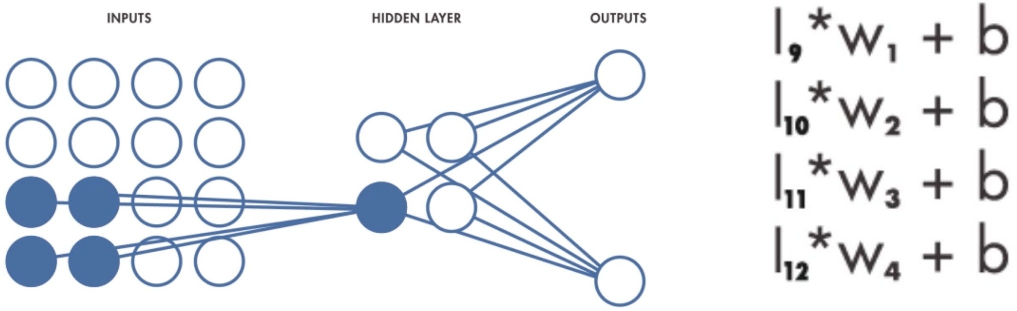




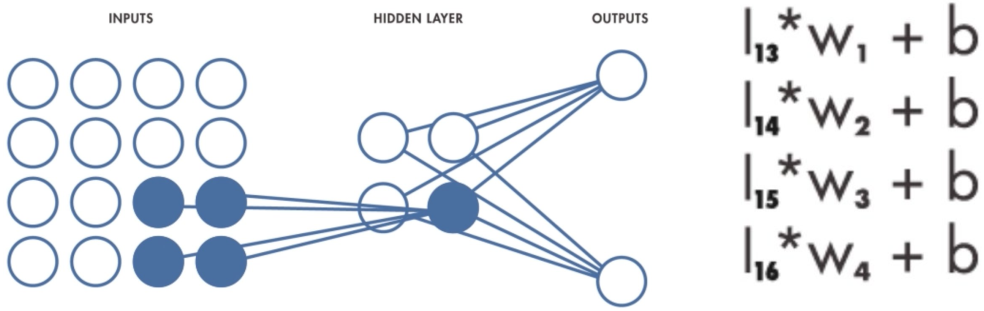
$$I_1*w_1+b$$
$$I_2*w_2+b$$
$$I_3*w_3+b$$
$$I_4*w_4+b$$



$$I_5*w_1+b$$
$$I_6*w_2+b$$
$$I_7*w_3+b$$
$$I_8*w_4+b$$



$$I_9*w_1+b$$
$$I_{10}*w_2+b$$
$$I_{11}*w_3+b$$
$$I_{12}*w_4+b$$



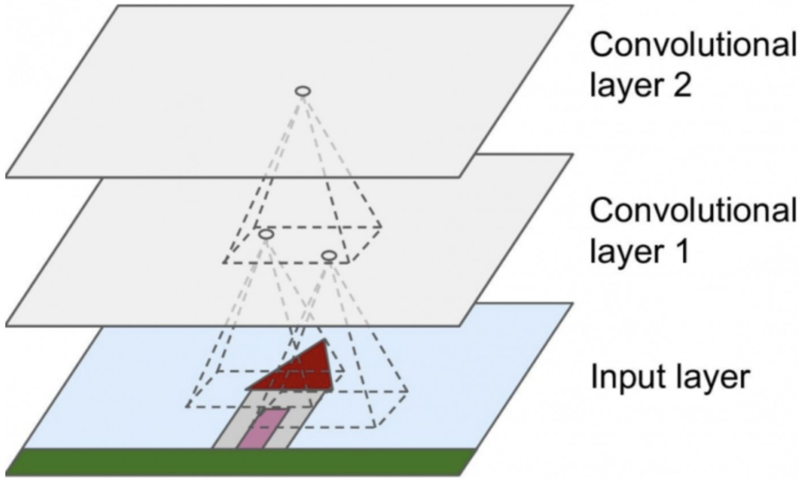
$$I_{13}*w_1+b$$
$$I_{14}*w_2+b$$
$$I_{15}*w_3+b$$
$$I_{16}*w_4+b$$

Image provided from MathWork

3. The structure of convolutional neural network

➢ Convolutional Layer

- Neurons in the convolutional layer are not connected to every single pixel in the input image but only to pixels in their receptive fields.
- This architecture allows the network to concentrate on small low-level features in the first hidden layer, then assemble them into larger higher-level features in the next hidden layer and so on.



3. The structure of convolutional neural network

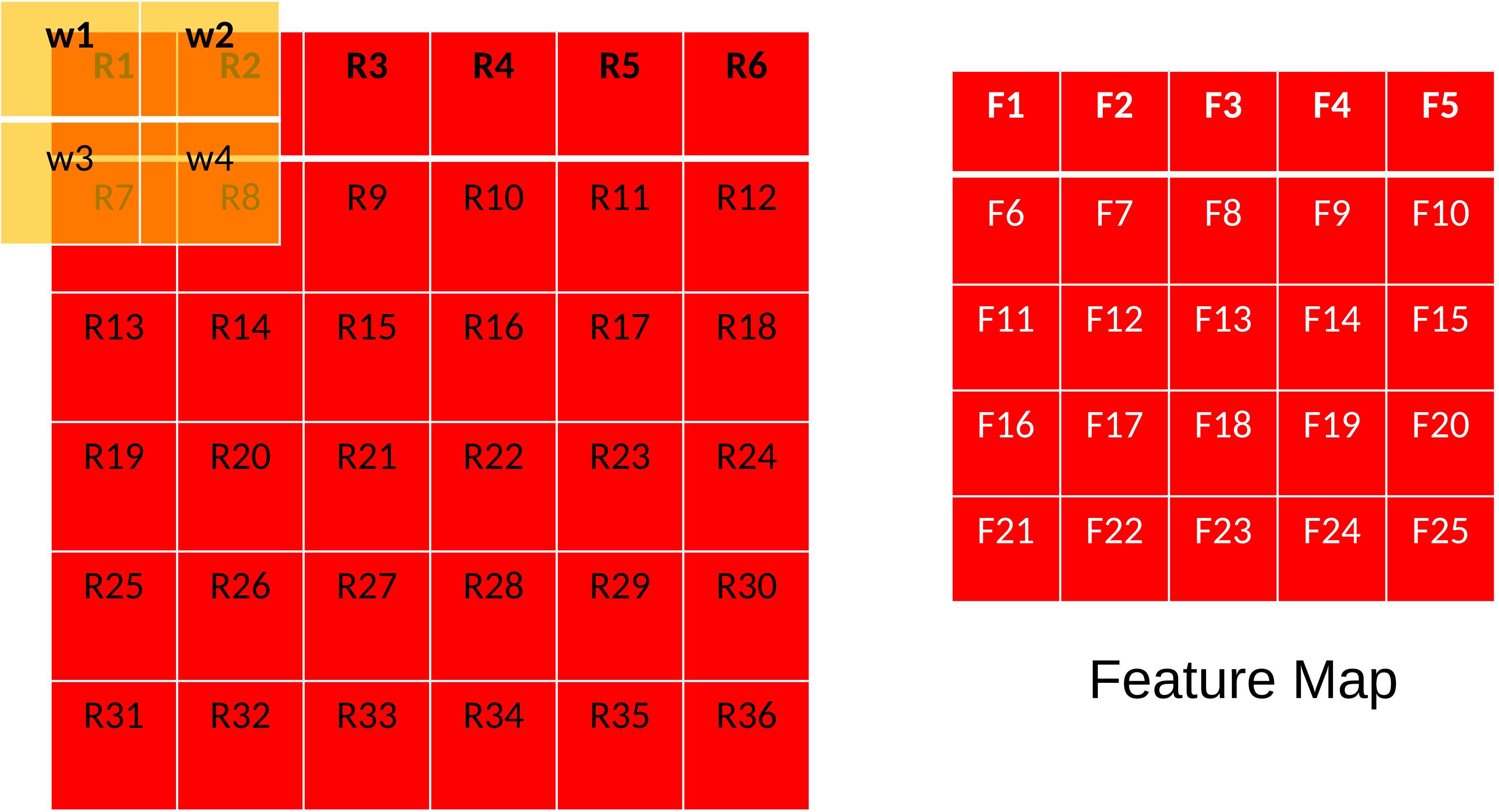
R1	**R2**	**R3**	**R4**	**R5**	**R6**
R7	R8	R9	R10	R11	R12
R13	R14	R15	R16	R17	R18
R19	R20	R21	R22	R23	R24
R25	R26	R27	R28	R29	R30
R31	R32	R33	R34	R35	R36

Input Image

w1	**w2**
w3	w4

Kernel

3. The structure of convolutional neural network



R1	R2	R3	R4	R5	R6
R7	R8	R9	R10	R11	R12
R13	R14	R15	R16	R17	R18
R19	R20	R21	R22	R23	R24
R25	R26	R27	R28	R29	R30
R31	R32	R33	R34	R35	R36

Input Image

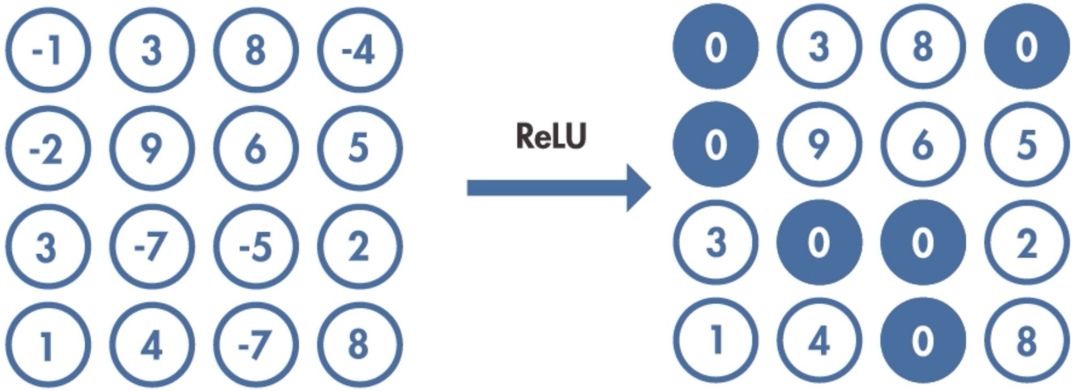
F1	F2	F3	F4	F5
F6	F7	F8	F9	F10
F11	F12	F13	F14	F15
F16	F17	F18	F19	F20
F21	F22	F23	F24	F25

Feature Map

$$F_1 = R_1 * W_1 + R_2 * W_2 + R_3 * W_3 + R_4 * W_4$$

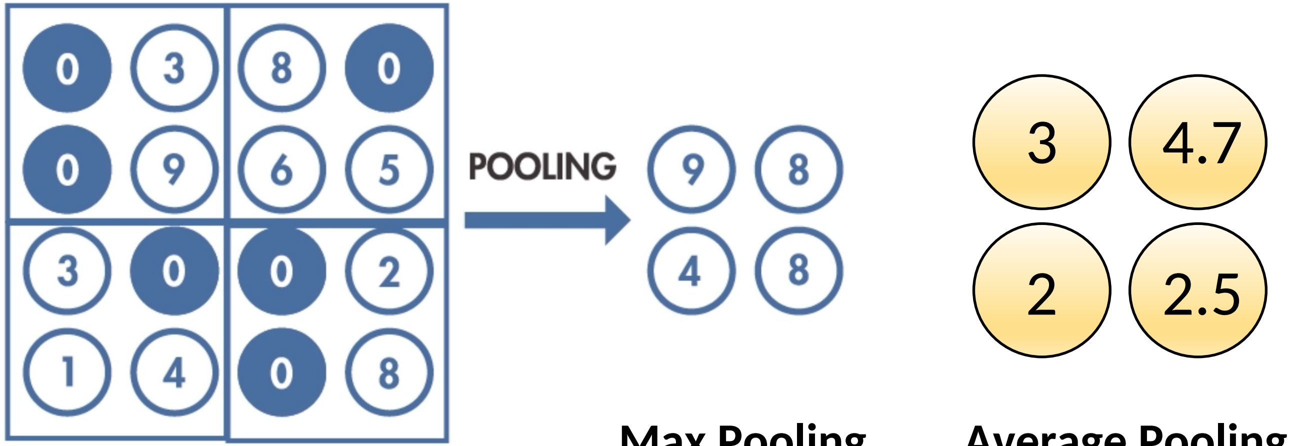
3. The structure of convolutional neural network

- Activation and pooling



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

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



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

9	8
4	8

Max Pooling

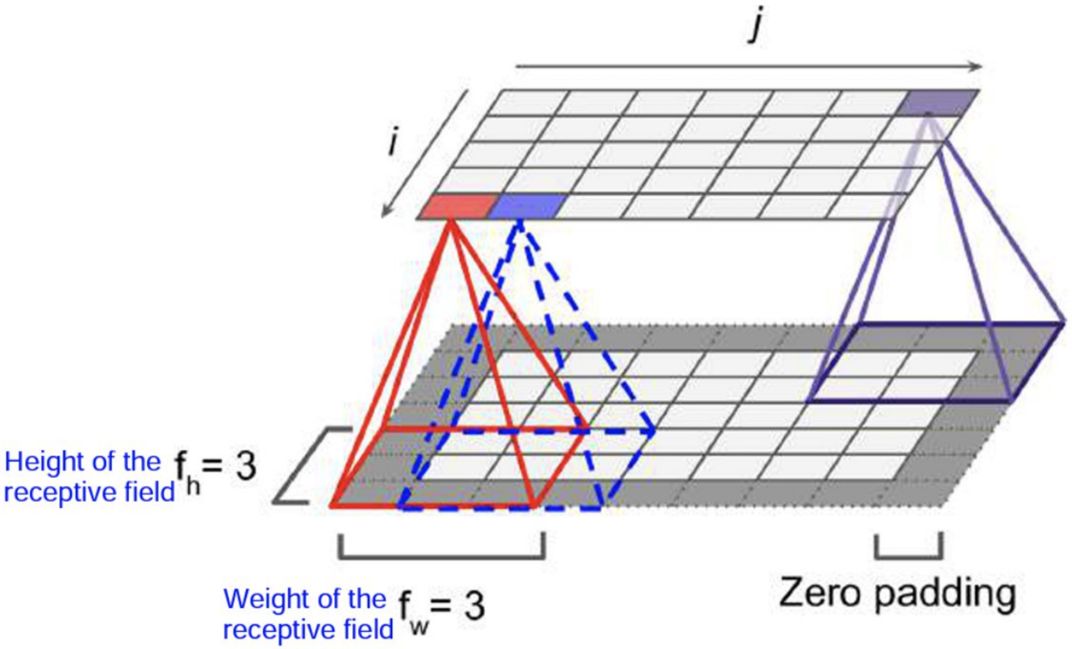
3	4.7
2	2.5

Average Pooling

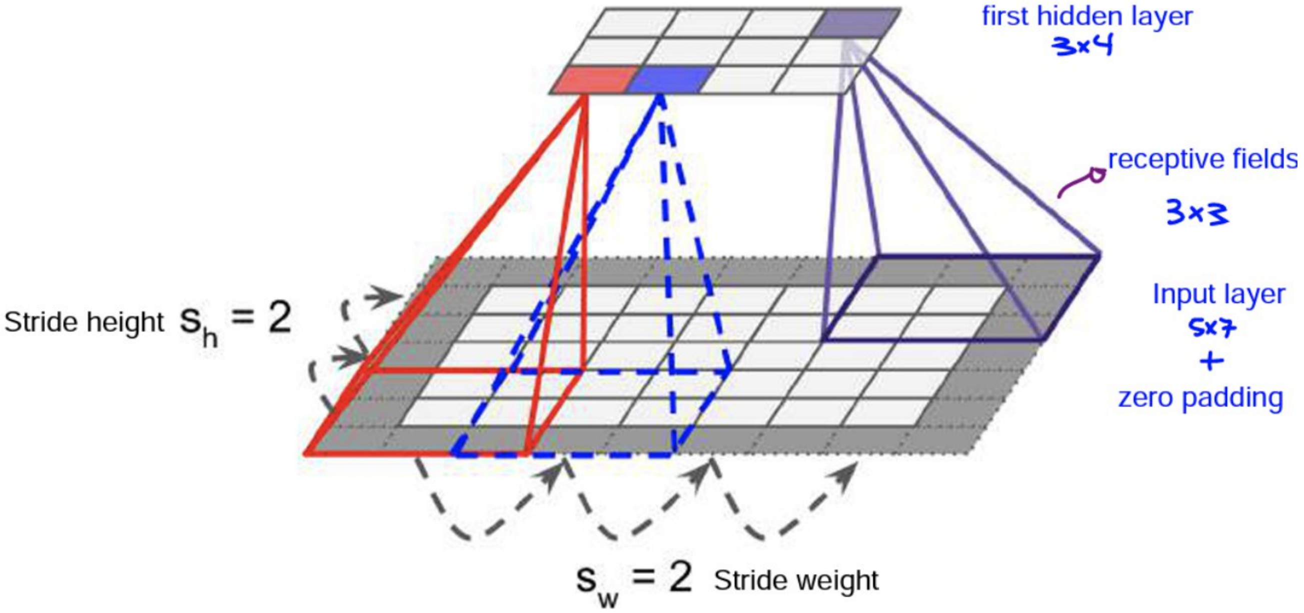
Image provided from MathWorks

3. The structure of convolutional neural network

- Stride and Padding


j
i
Height of the receptive field $f_h = 3$
Weight of the receptive field $f_w = 3$
Zero padding

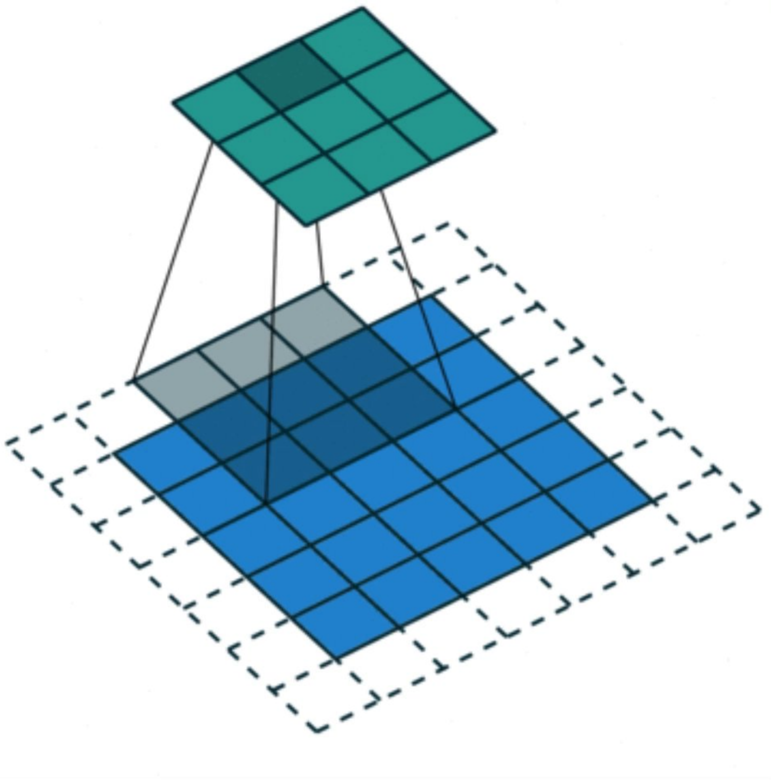
Connections between layers and zero padding



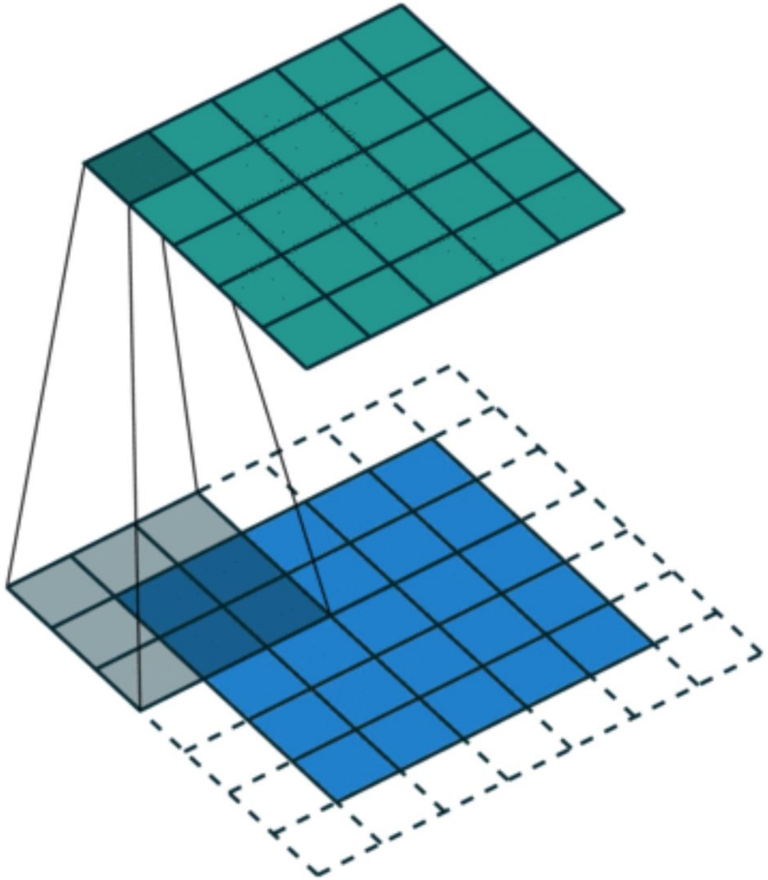
Reducing dimensionality using a stride of 2

3. The structure of convolutional neural network

- Stride and Padding



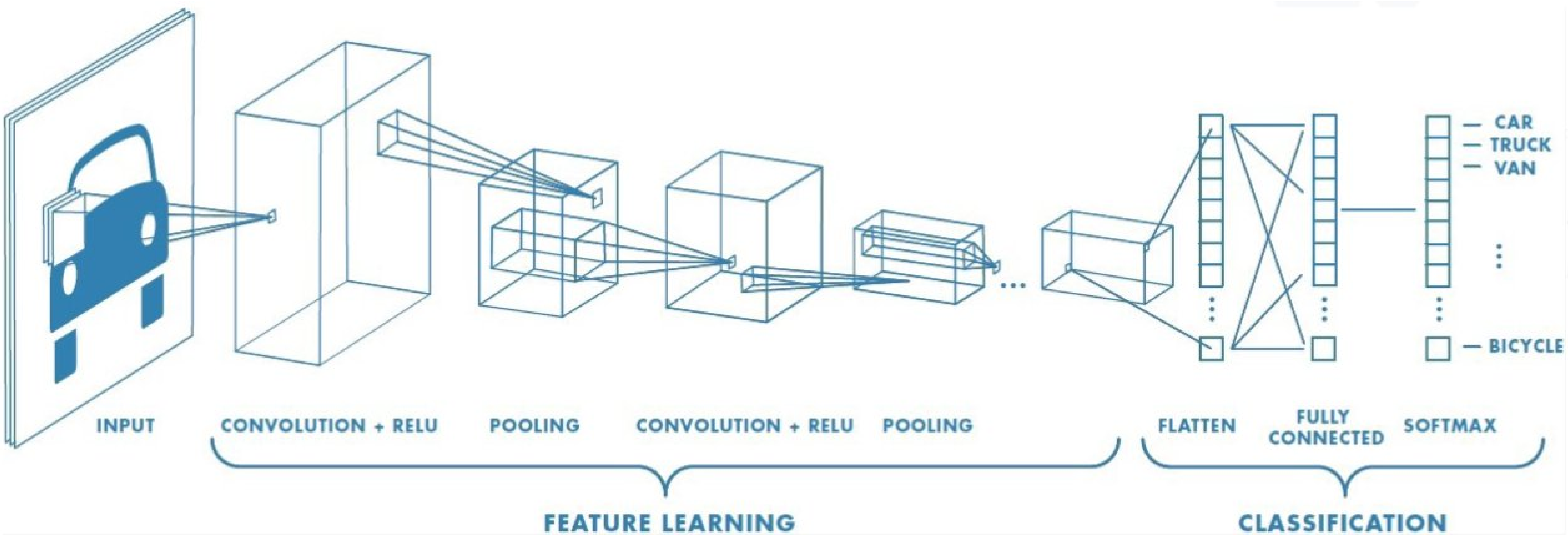
Convolution Operation with Stride Length = 2



SAME padding: 5×5×1 image is padded with 0s to create a 6×6×1 image

Image provided from MathWorks

3. The structure of convolutional neural network



Schematic representation of a convolutional neural network with two hidden layers

Image provided from MathWork

4. Looking at different CNN structure

AlexNet – won the 2012 ImageNet challenge

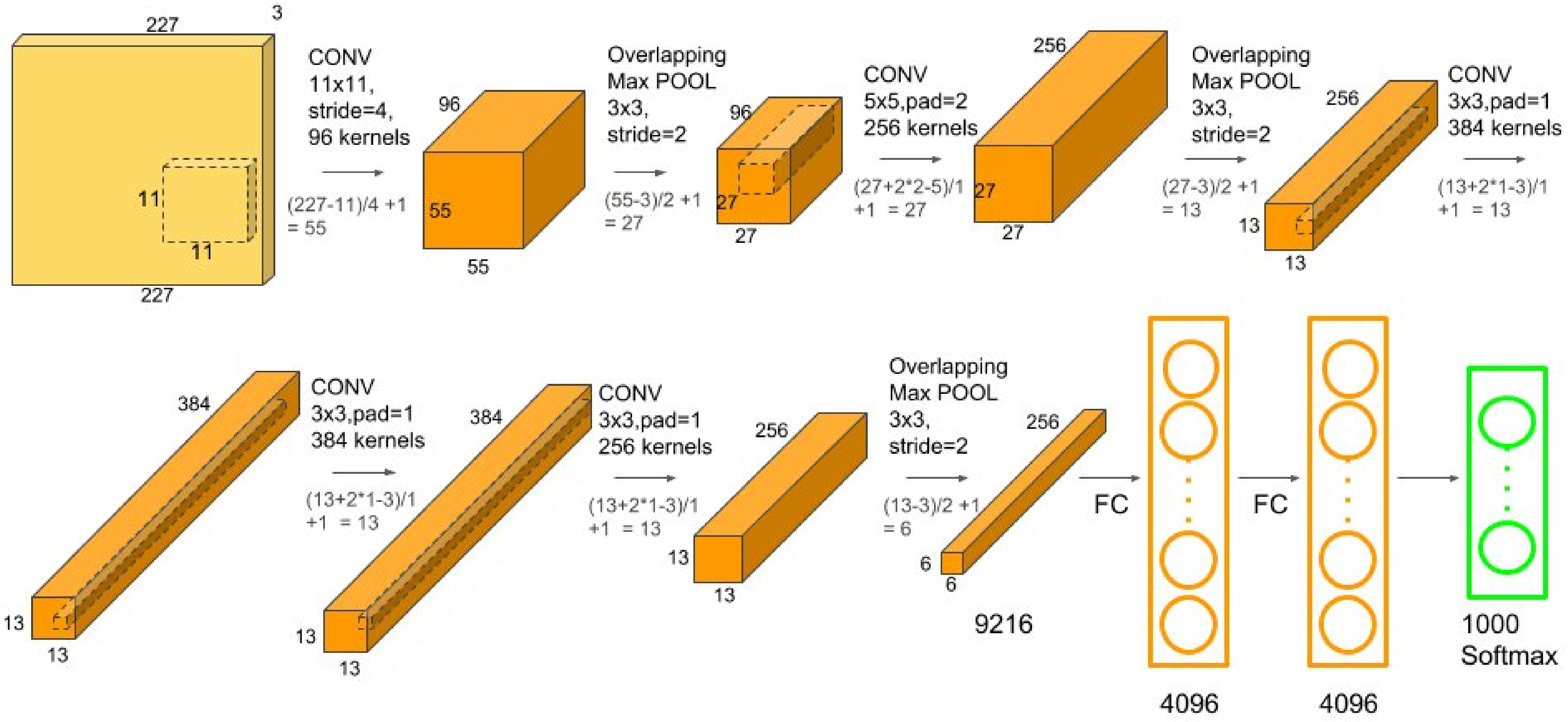


Image provided from LearnOpenCV

4. Looking at different CNN structure

ResNet

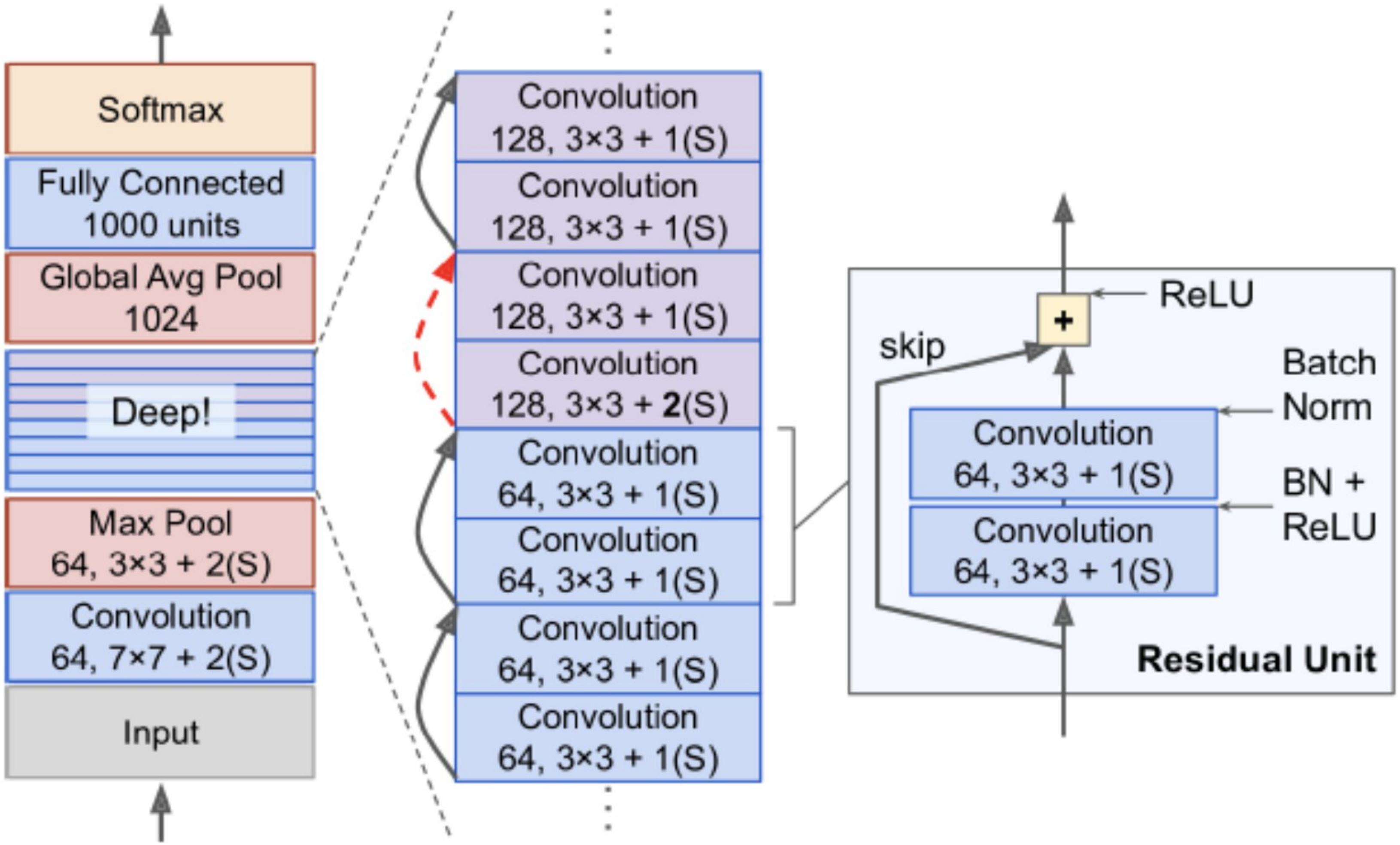


Image provided from https://jason-adam.github.io/resne

5. How to code convolutional neural network in python

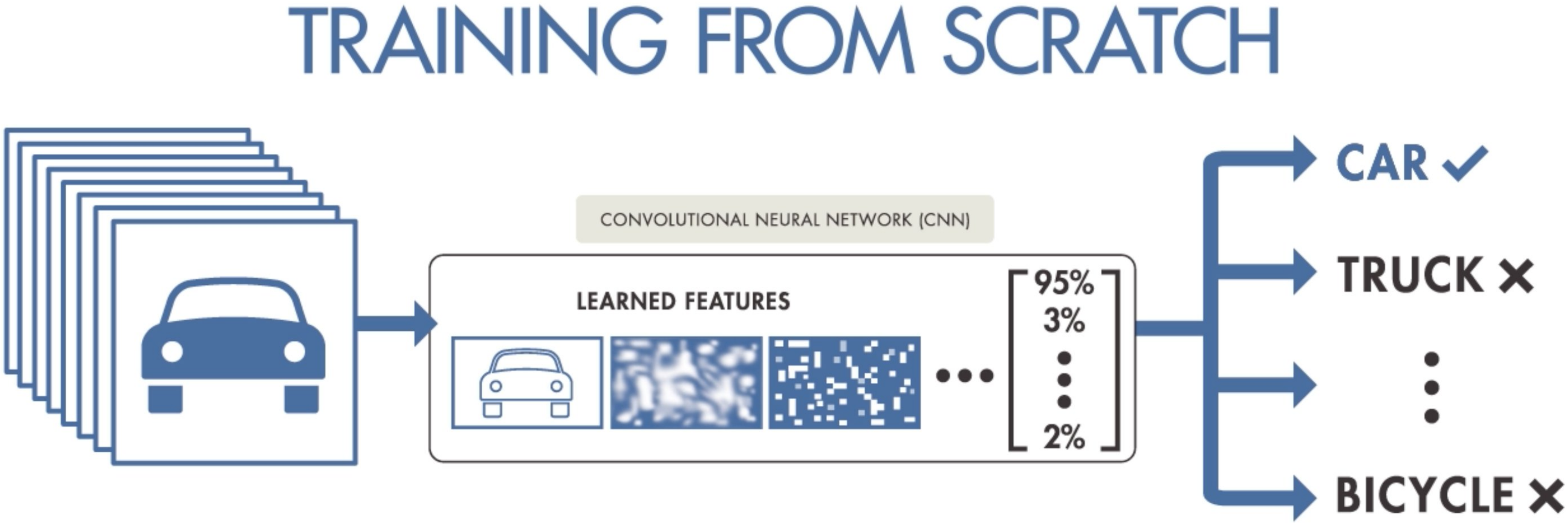


Image provided from MathWork

5. How to code convolutional neural network in python

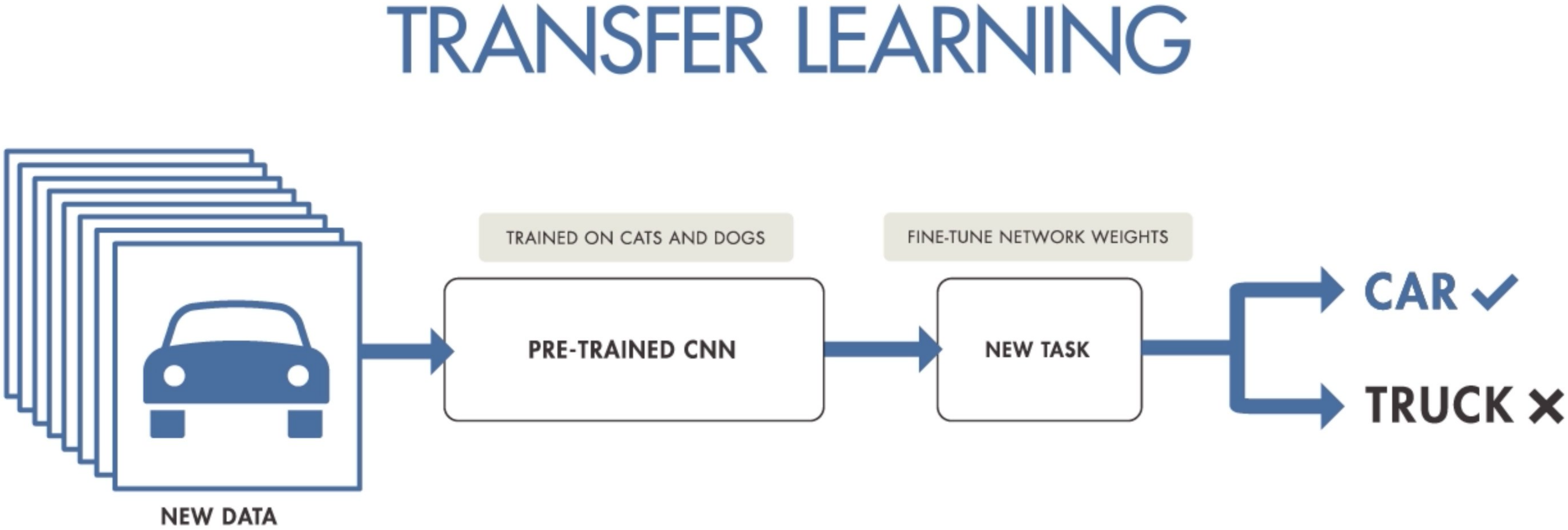


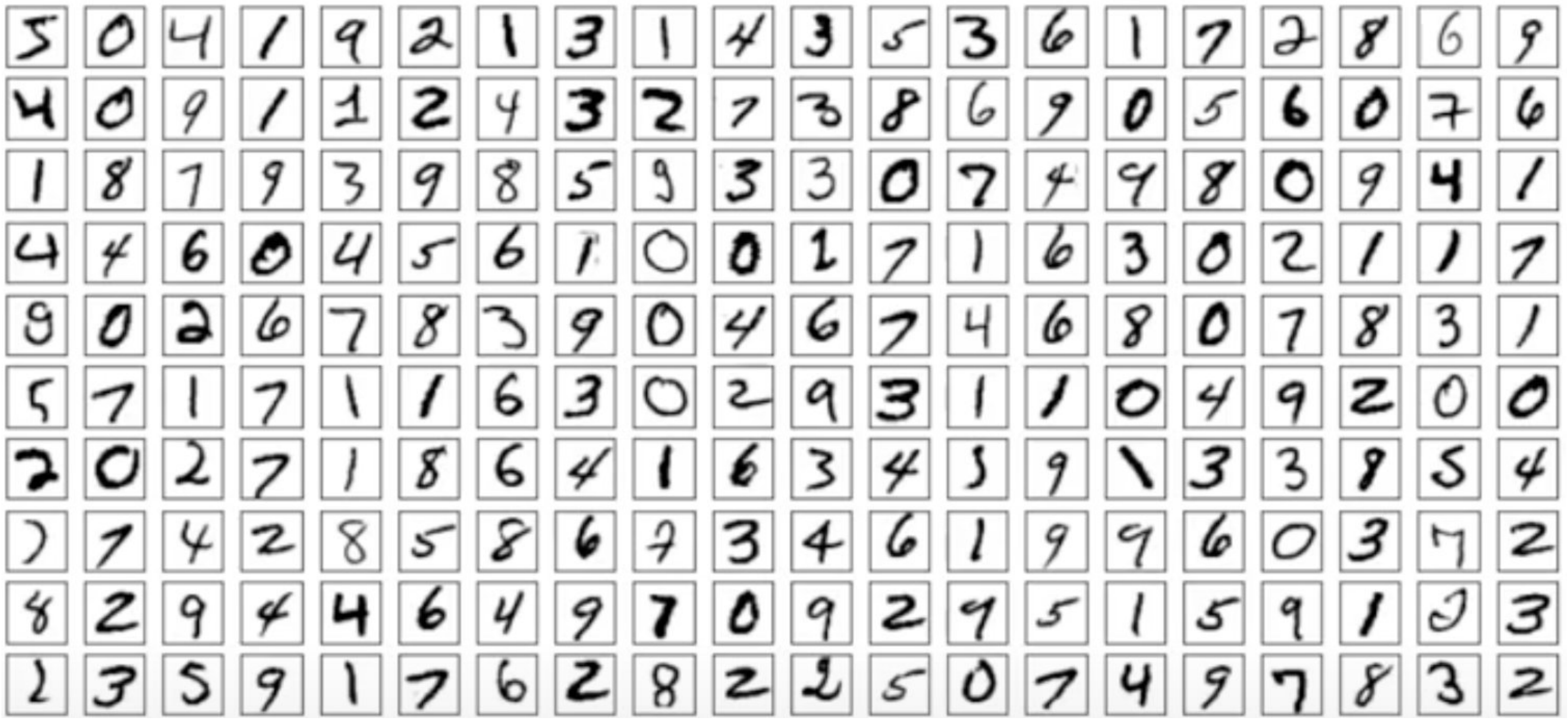
Image provided from MathWork

6. Hands on project

Convolutional Network Demo from 1993

6. Hands on project

We are going to use the handwritten digit dataset and using convolutional neural network for digit classi



Open your Jupyter Notebook for the rest …