AI on Chip
Lab1 AI Model

AOC 2024 Lab1

TAs: course.aislab@gmail.com

Outline

- Google Colab
- What is Pytorch?
- Neural Networks(NN)
- Convolutional Neural Networks(CNN)
- Dataset
- Advanced Topics
- Assignment

Outline

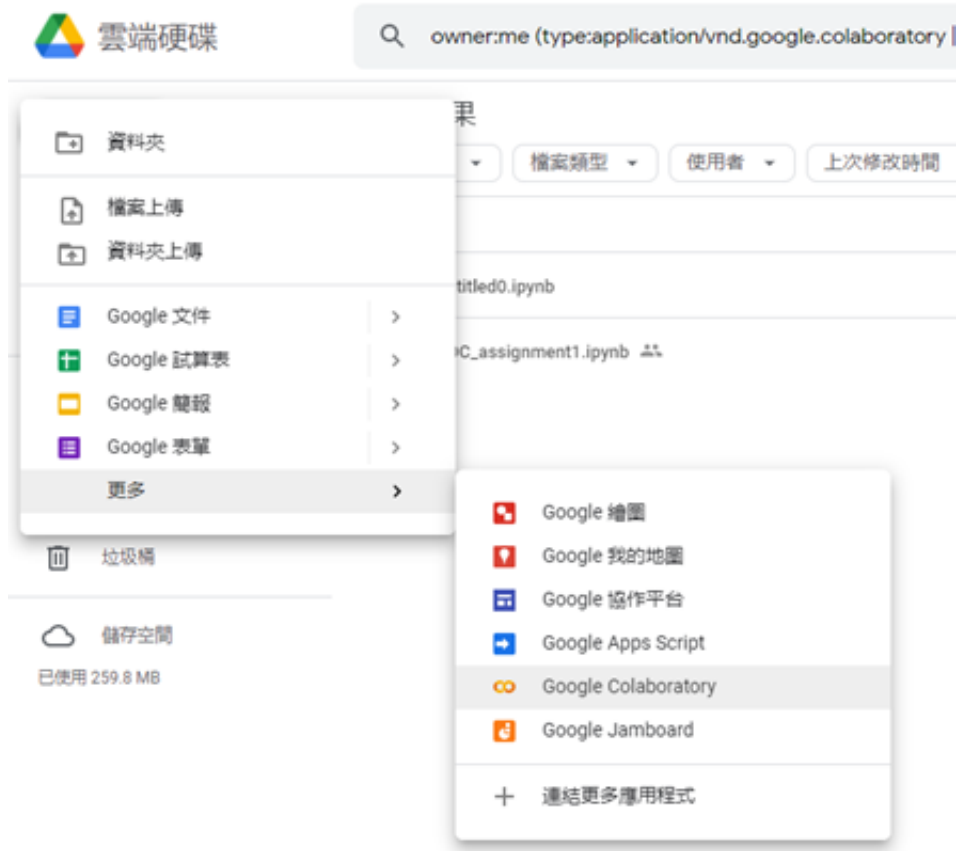
- Google Colab
- What is Pytorch?
- Neural Networks(NN)
- Convolutional Neural Networks(CNN)
- Dataset
- Advanced Topics
- Assignment

Google Colab

- Google offer Jupyter Notebook develop environment
- Free 12GB GPU(Tesla T4)
- 50GB storage
- 12 hours continuous using limited.
- Idling over 90 minutes will be kick out by host. Need to reconnect.
- Google Colab run in Ubuntu Linux

Google Colab

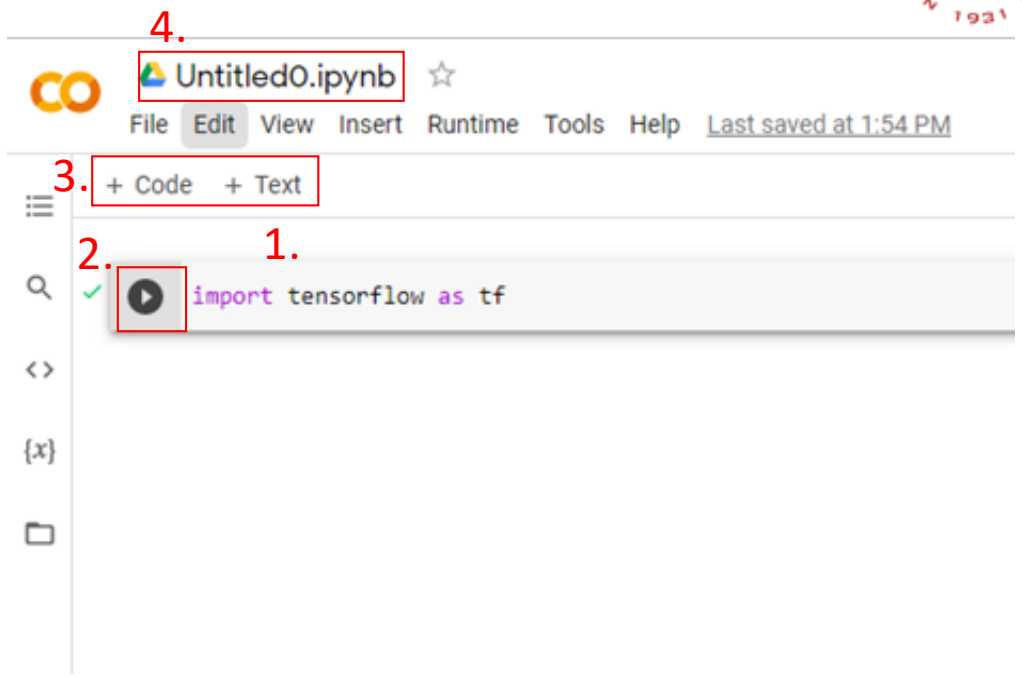
- Goto Google Drive and new Google Colaboratory file



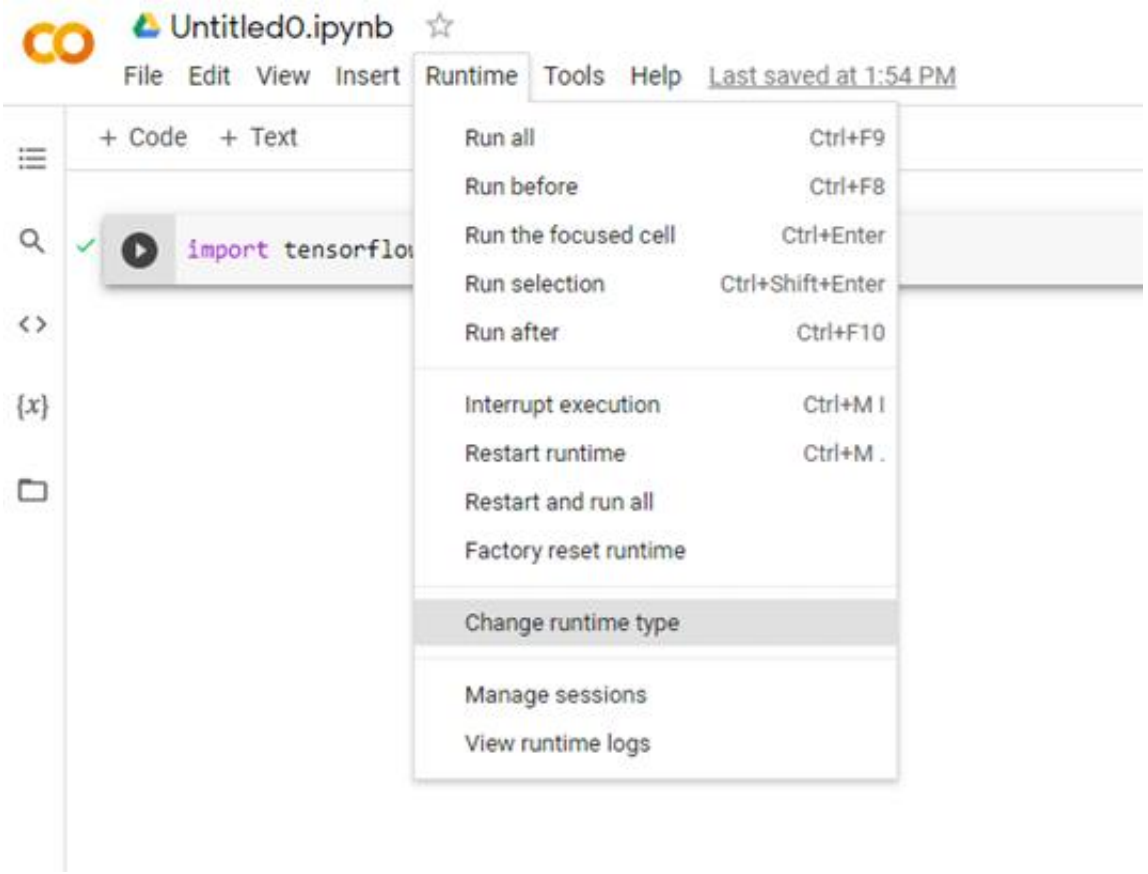
First open Colab file

- It will show Jupyter Notebook like page
- After creating a file

1. Code section
2. Run cell (run your code)
3. Append Code section or Text section
4. Rename
5. Resource usage
6. Code section operation



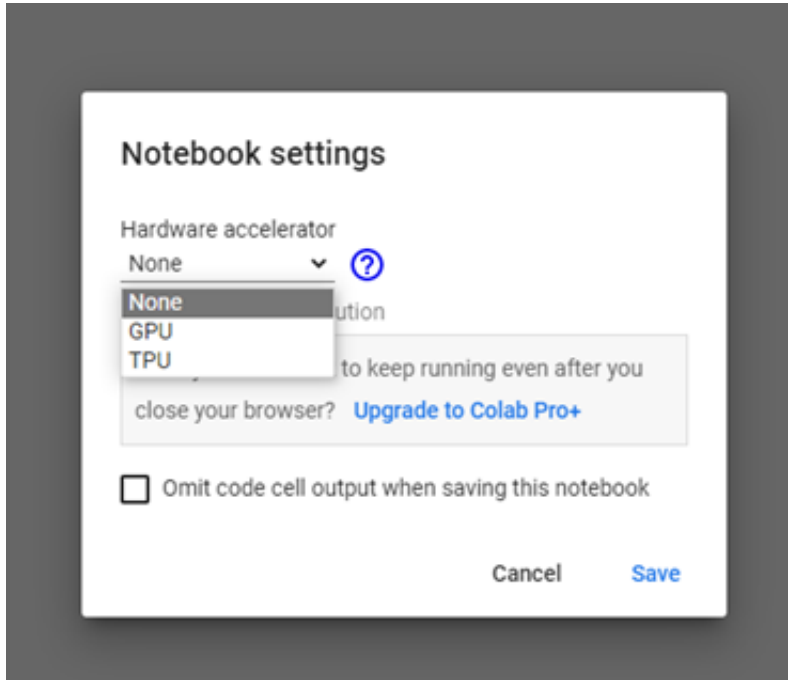
Choose GPU as runtime



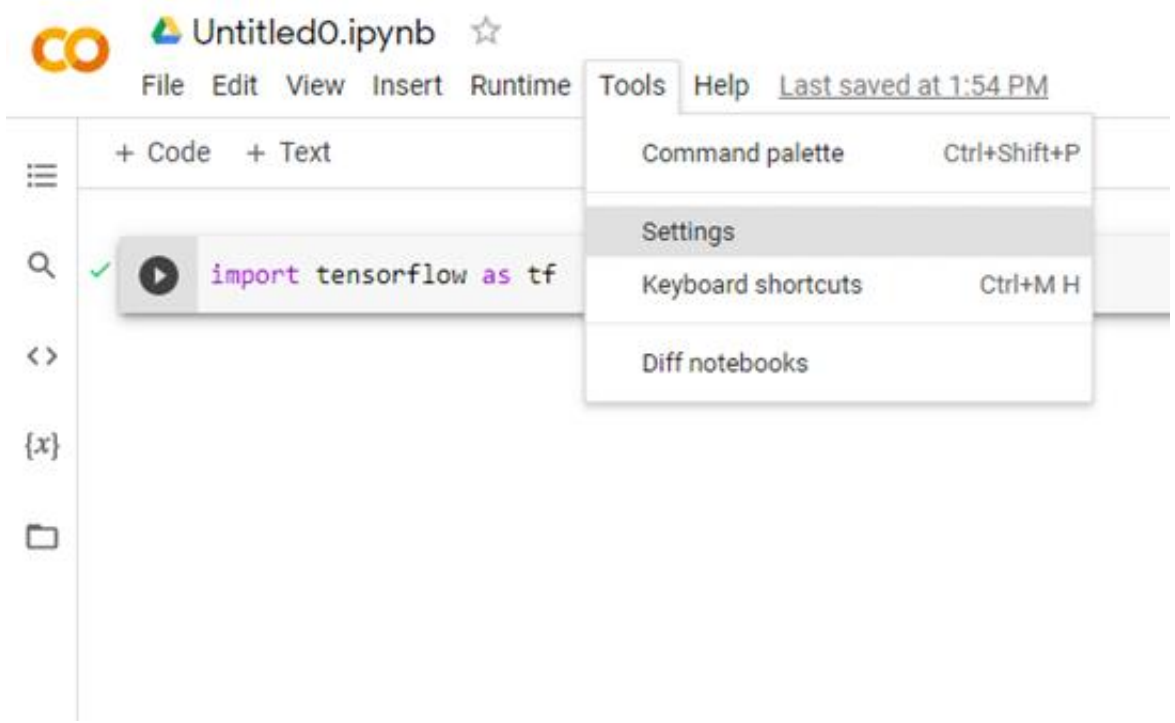
Default runtime: CPU

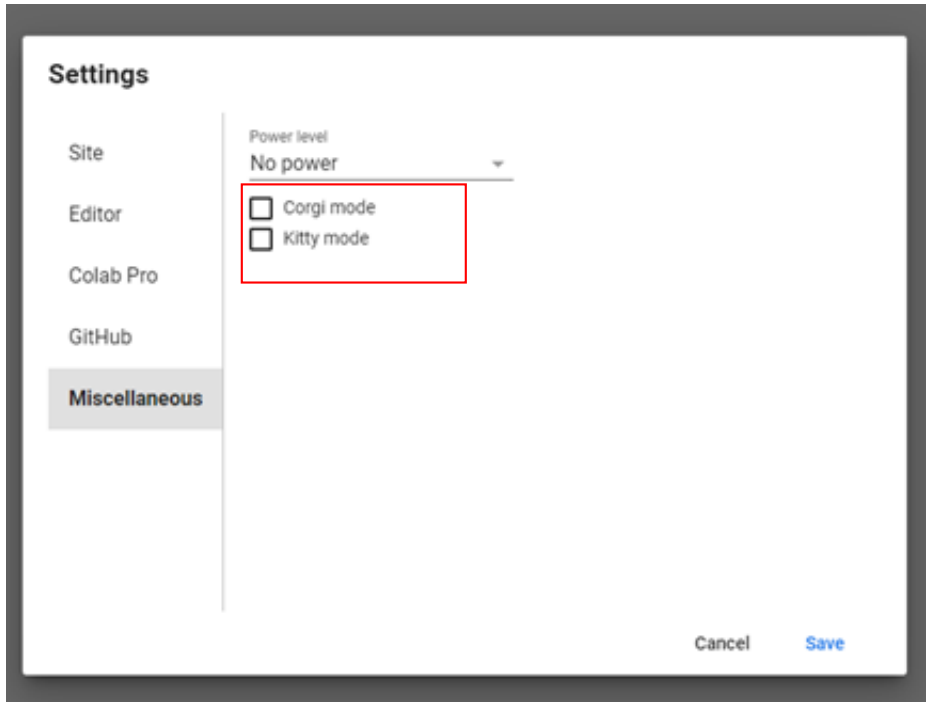
You can change it as needed

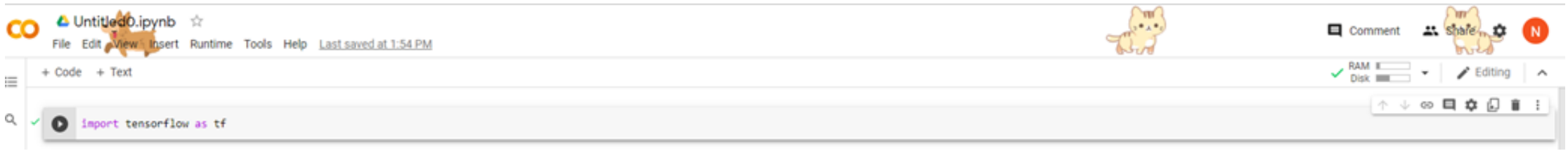
Runtime -> Change runtime type -> GPU



Choose a pet


Untitled0.ipynb
File Edit View Insert Runtime Tools Help Last saved at 1:54 PM
+ Code + Text
import tensorflow as tf
Command palette Ctrl+Shift+P
Settings
Keyboard shortcuts Ctrl+M H
Diff notebooks

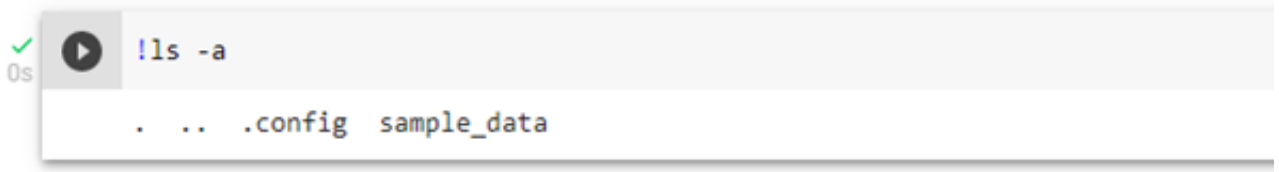

Settings
Site
Editor
Colab Pro
GitHub
Miscellaneous
Power level
No power
Corgi mode
Kitty mode
Cancel
Save


Untitled0.ipynb
File Edit View Insert Runtime Tools Help Last saved at 1:54 PM
Comment
Share
+ Code + Text
RAM
Disk
Editing
import tensorflow as tf

If you want to use shell instruction in Colab

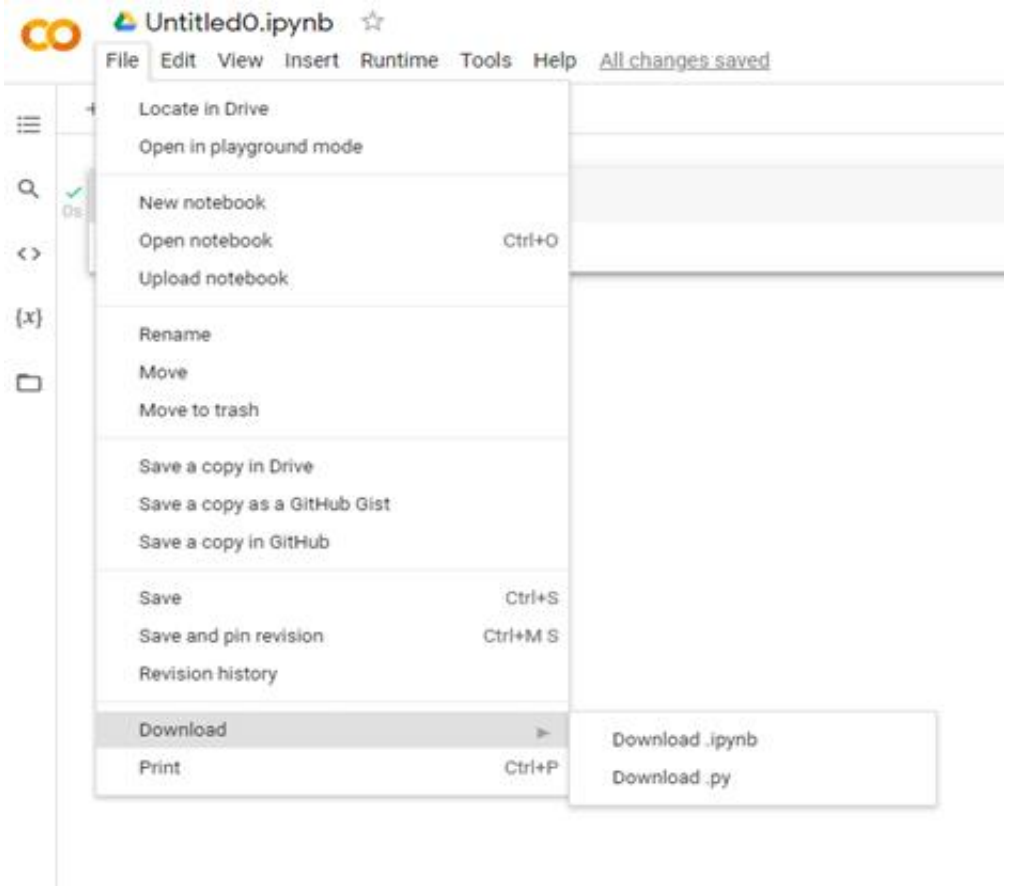
Add ! For most shell instruction

Ex: !ls -a, for list all directories



Google Colab - Save & Export

Save: *.ipynb



Export: export as HTML, LaTex, Markdown, PDF, Python…

Outline

- Google Colab
- What is Pytorch?
- Neural Networks(NN)
- Convolutional Neural Networks(CNN)
- Dataset
- Advanced Topics
- Assignment

What is Pytorch

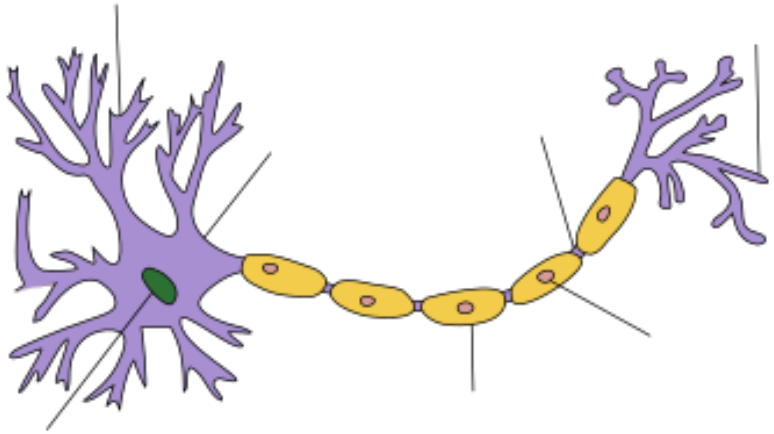
- https://pytorch.org/
- Developed by Facebook AI
- The most popular machine learning framework amongst ML developers.
- Other framework: Tensorflow, Caffe …

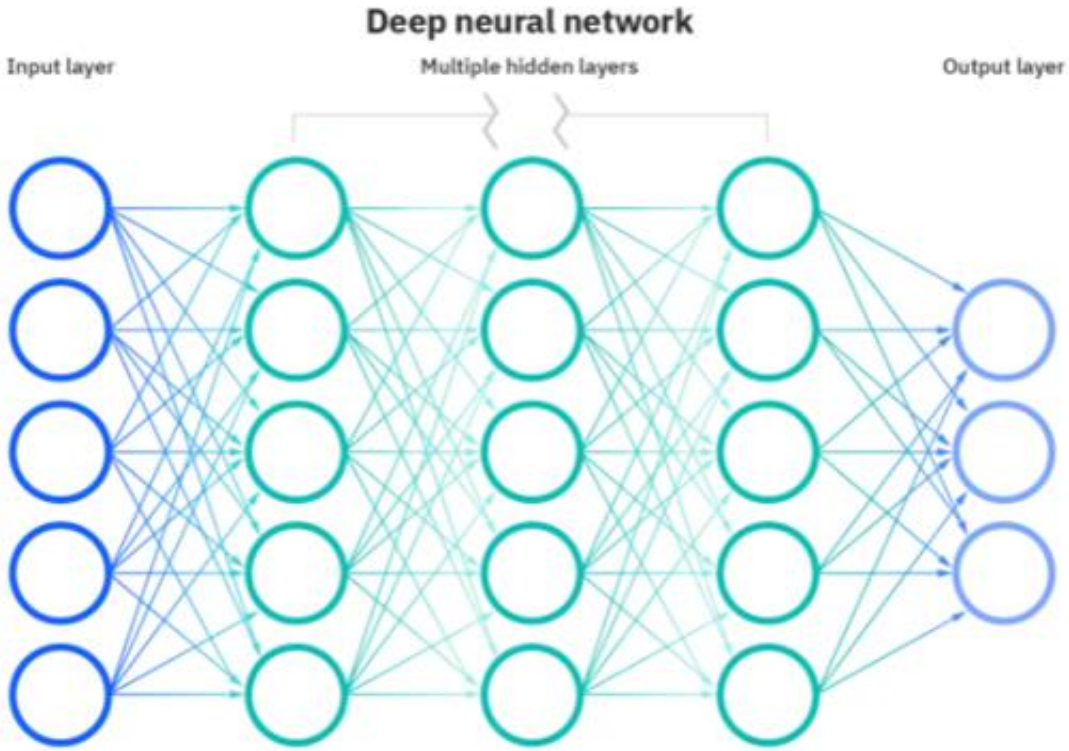
Outline

- Google Colab
- What is Pytorch?
- Neural Networks(NN)
- Convolutional Neural Networks(CNN)
- Dataset
- Advanced Topics
- Assignment

Neural Networks(NN)

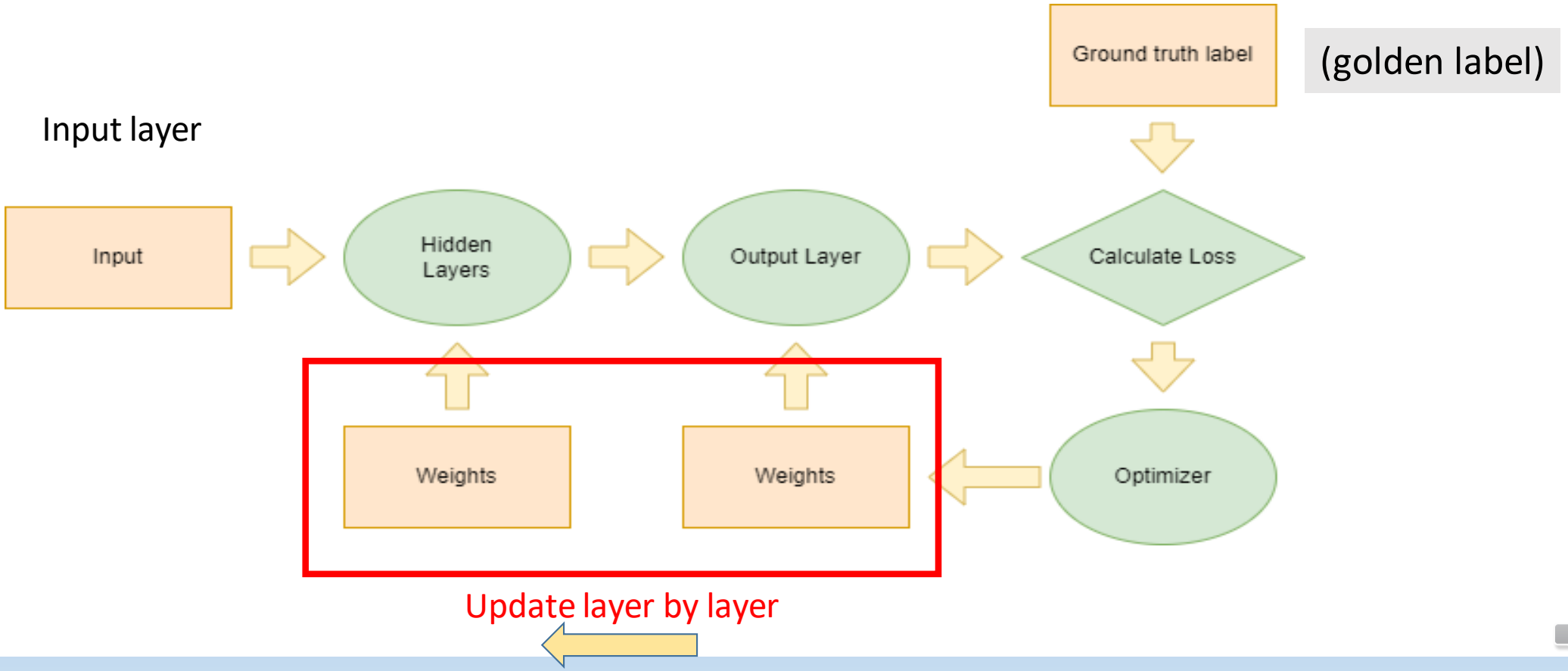
- Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.





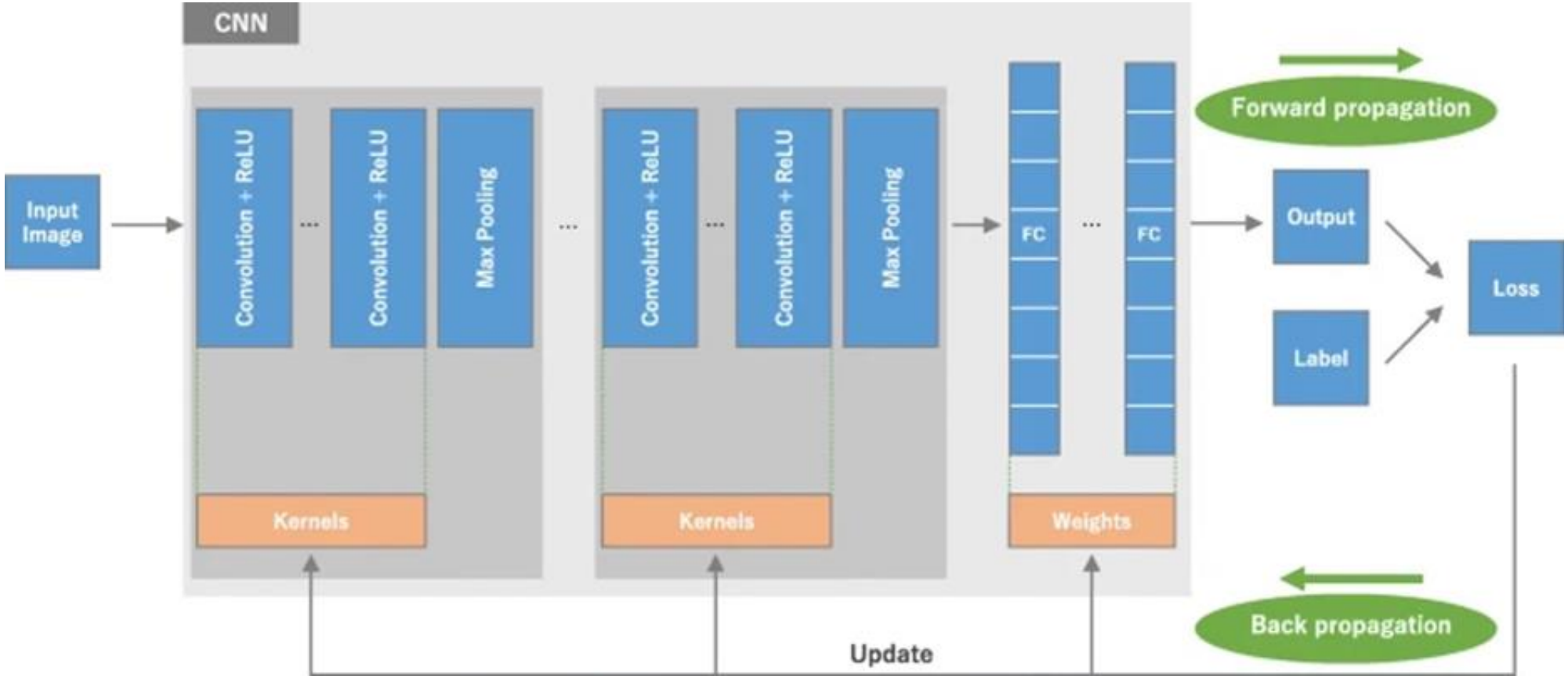
How Neural Networks(NN) work?

- Neural Networks in general are composed of a collection of neurons that are organized in layers, each with their own learnable weights and biases.



What is CNN (Convolutional Neural Network)?

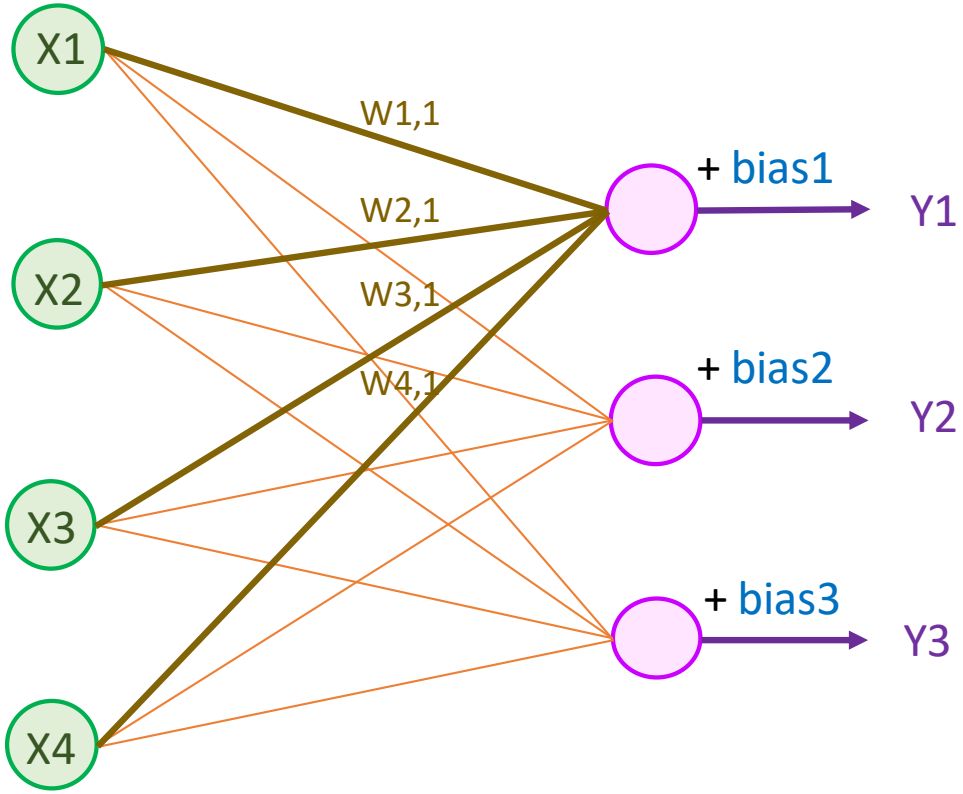
- A CNN is a neural network: An algorithm used to recognize patterns in data.



Convolution, Activation, Pool, Fully connection could repeat many times

Fully Connection

- A fully connected neural network consists of a series of fully connected layers that connect every neuron in one layer to every neuron in the other layer.



$$Y1 = X1*W1,1 + X2*W2,1 + X3*W3,1 + X4*W4,1 + bias1$$

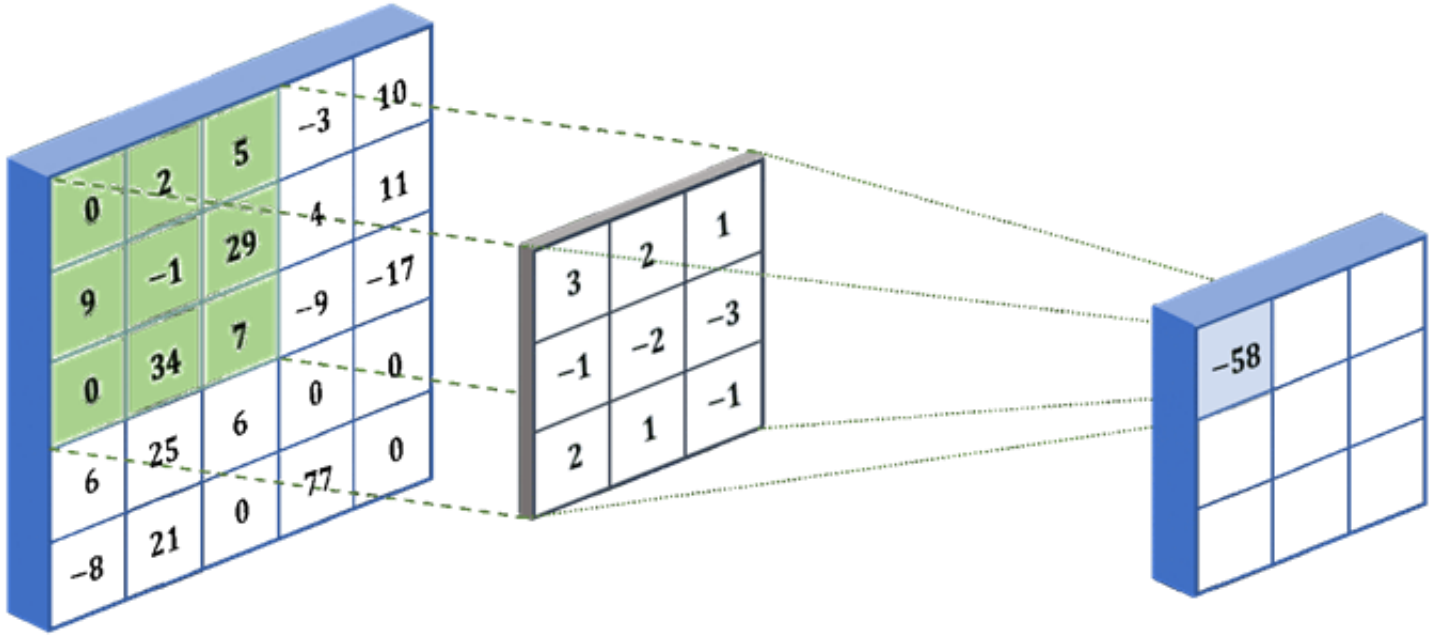
$$\begin{bmatrix} W1,1 & \cdots & W4,1 \\ \vdots & \ddots & \vdots \\ W1,3 & \cdots & W4,3 \end{bmatrix} \begin{bmatrix} X1 \\ \dots \\ X4 \end{bmatrix} + \begin{bmatrix} \text{bias1} \\ \dots \\ \text{bias3} \end{bmatrix} = \begin{bmatrix} Y1 \\ \dots \\ Y3 \end{bmatrix}$$

Outline

- Google Colab
- What is Pytorch?
- Neural Networks(NN)
- Convolutional Neural Networks(CNN)
- Dataset
- Advanced Topics
- Assignment

Convolution layer

- Basic operation of convolution

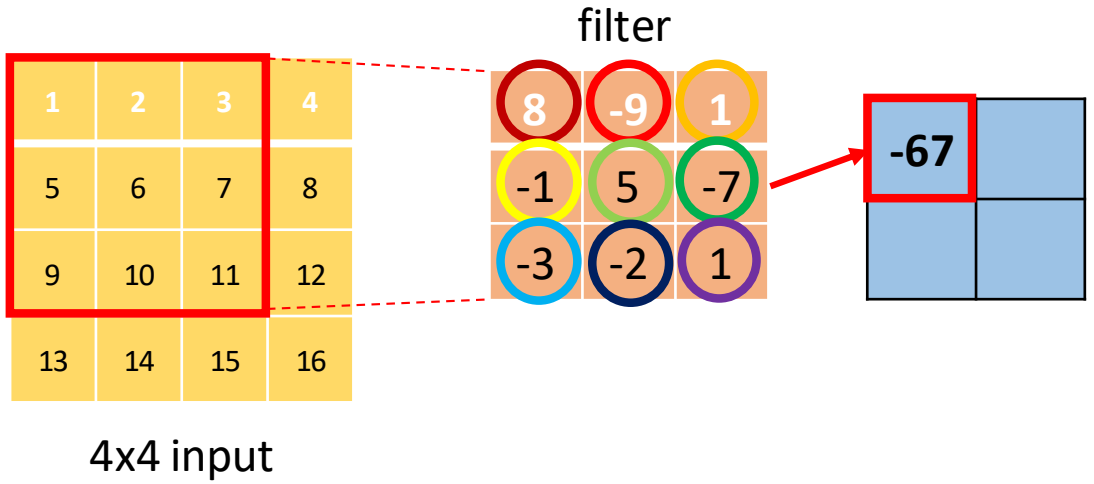


Input

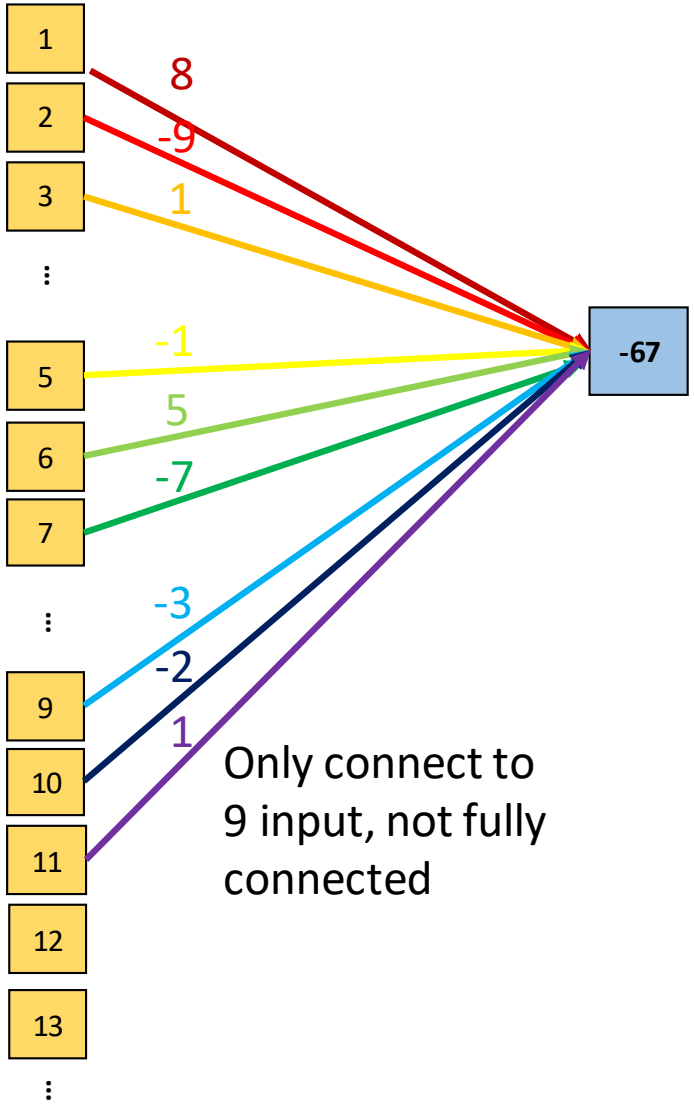
filter
(has 1 kernel)

Output

What is CNN(Convolutional Neural Network)? - Not fully connected



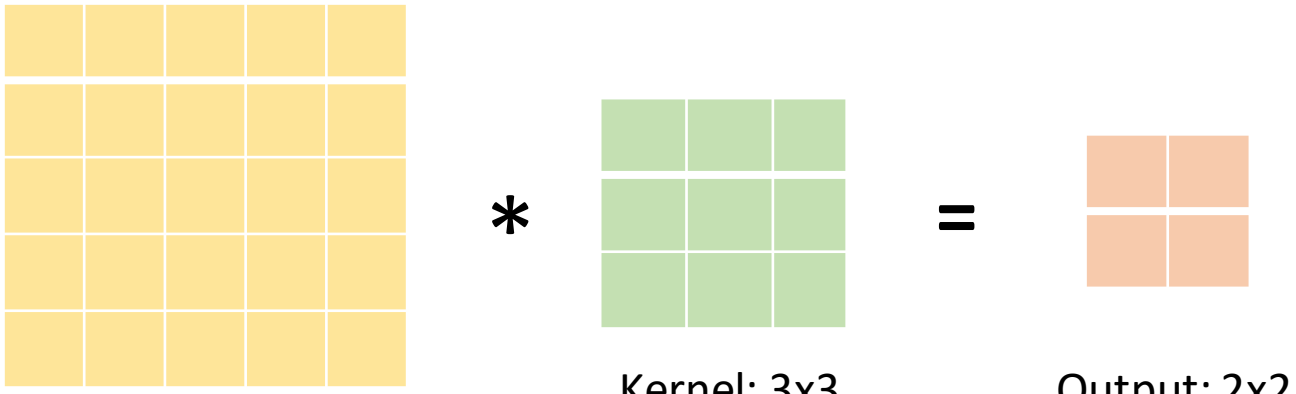
8x1 + (-9)x2 + 1x3 + (-1)x5 + 5x6 + (-7)x7 + (-3)x9 + (-2)x10 +1x11 = -67



What is CNN(Convolutional Neural Network)? - Stride

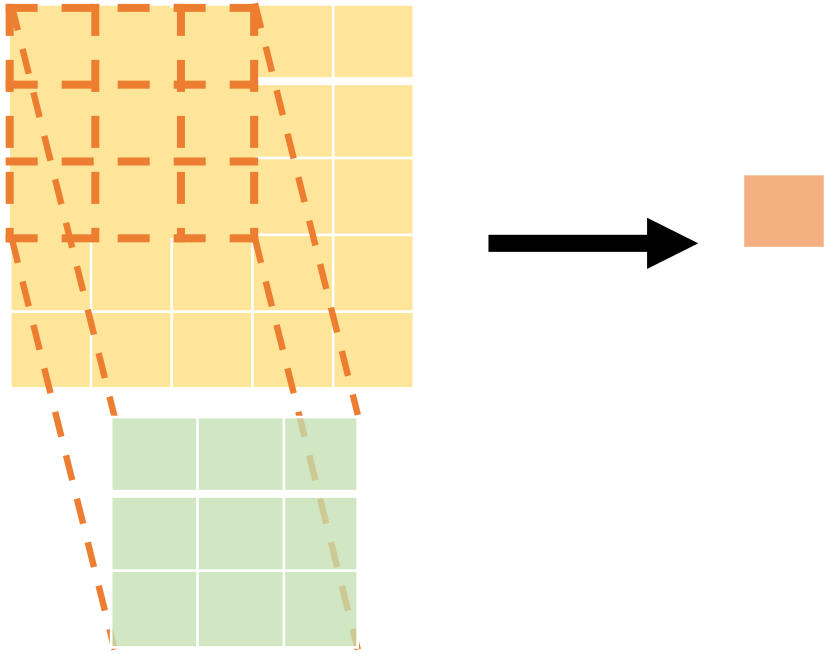
Here, set stride to 2. It means kernel slides on the input with step of 2.

We will explain how it work in following pages.



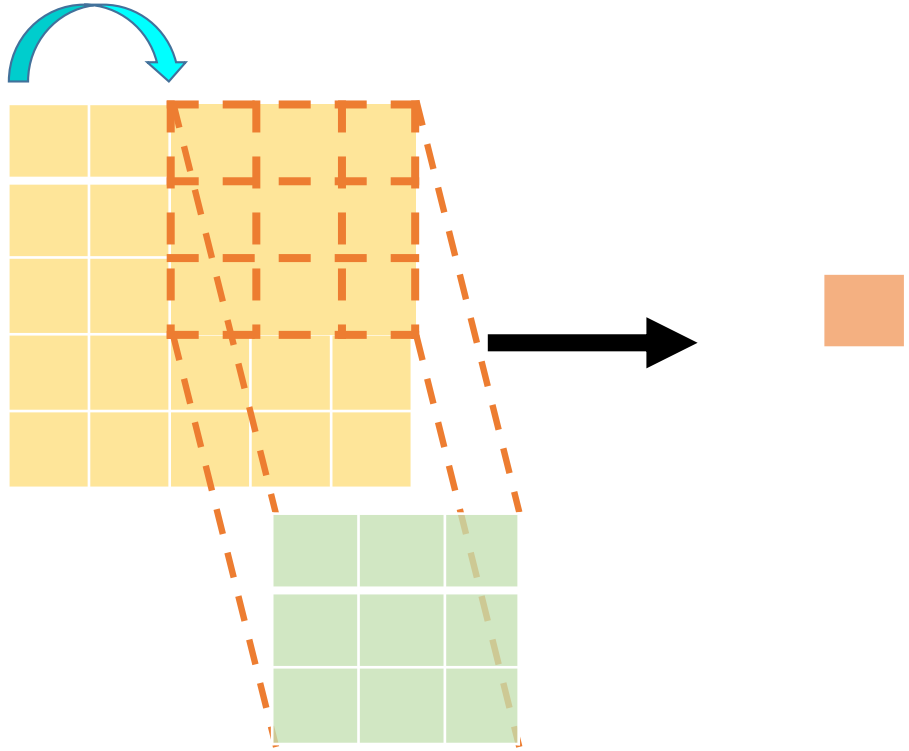
What is CNN(Convolutional Neural Network)? - Stride

Step 1.



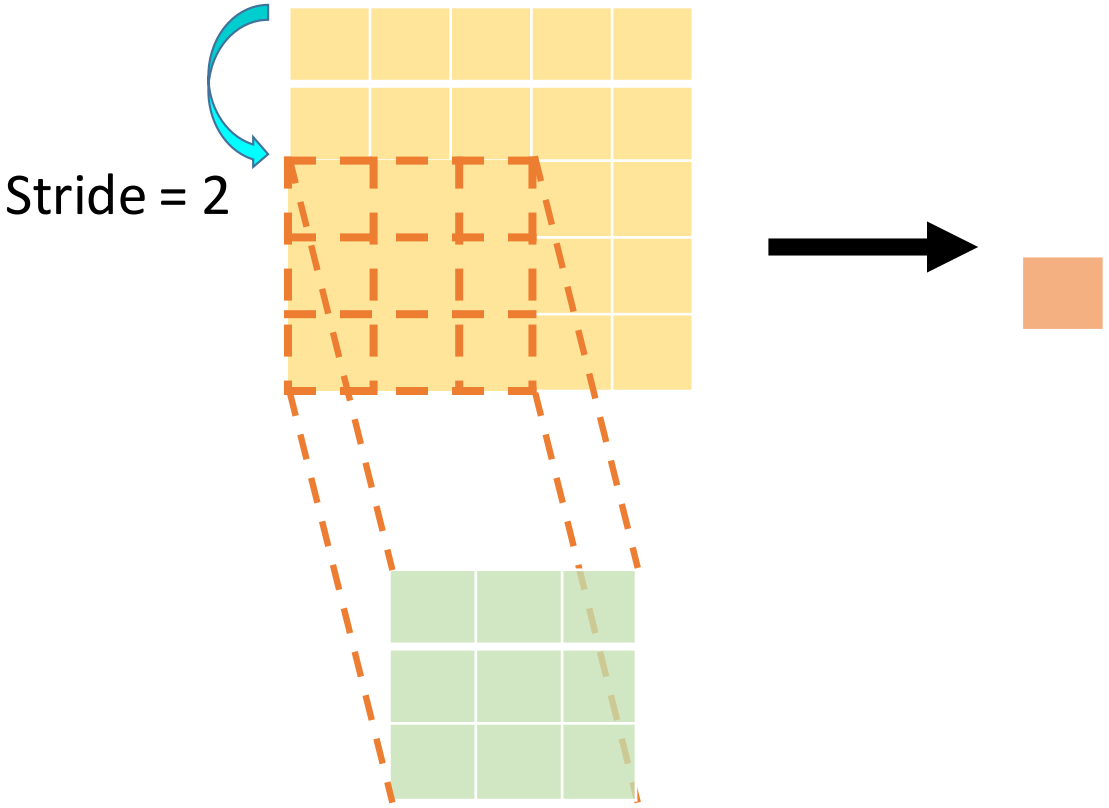
Stride = 2

Step 2.

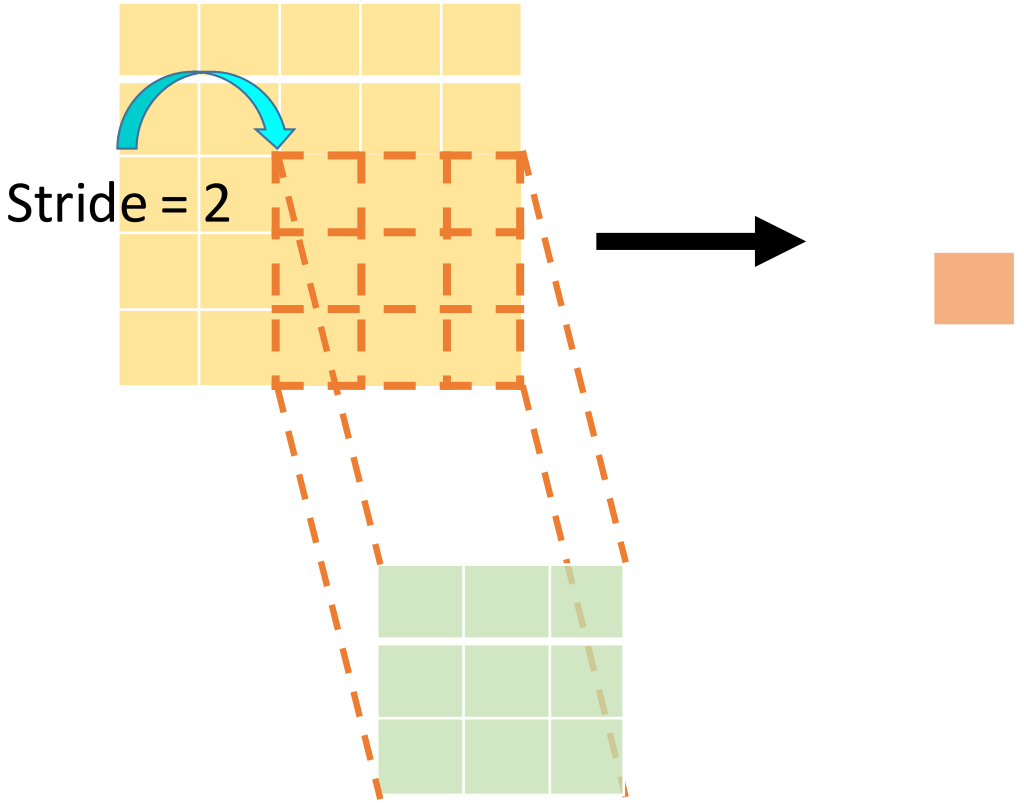


What is CNN(Convolutional Neural Network)?
- Stride

Step 3.

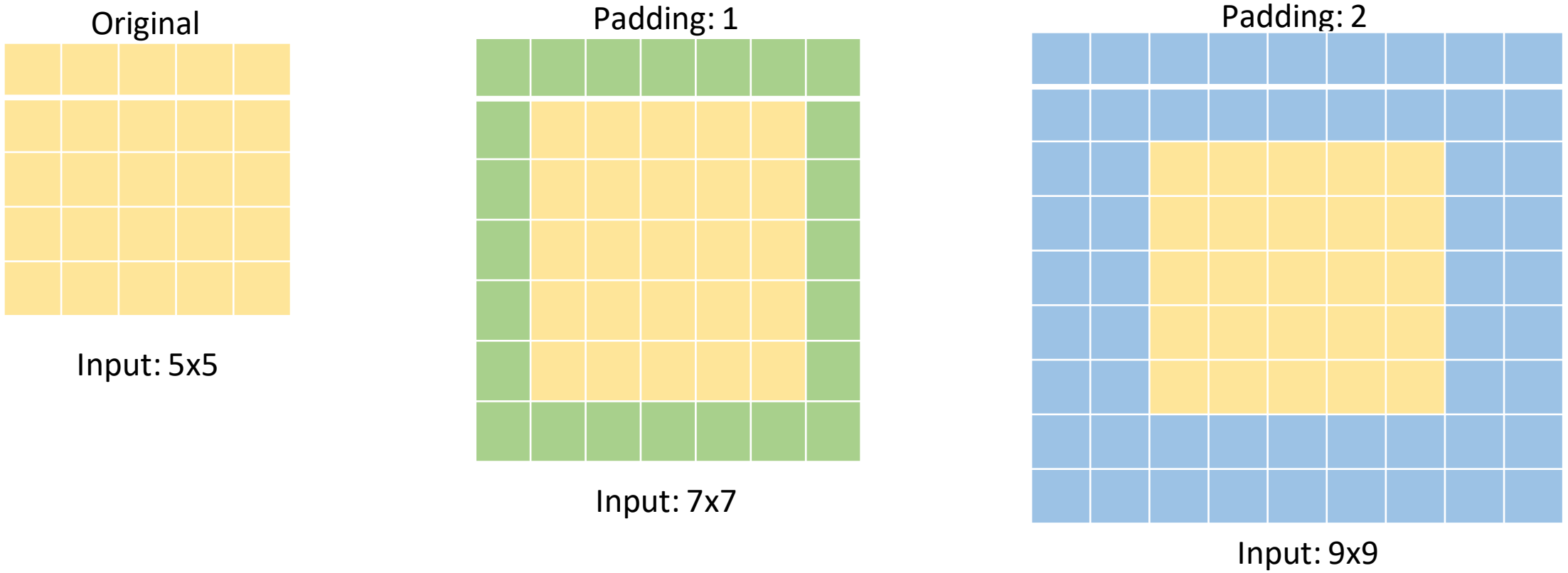


Step 4.



What is CNN(Convolutional Neural Network)? - Padding

Also, we could set padding. It means padding extra pixel surrounding to input.

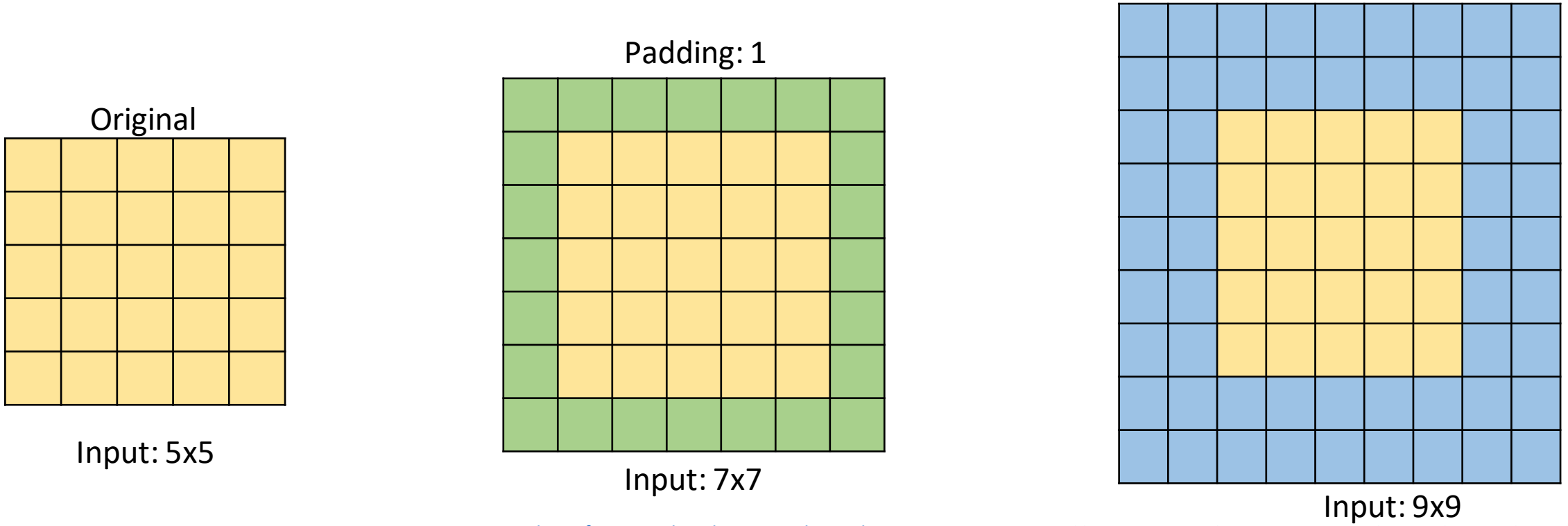


What is CNN(Convolutional Neural Network)? - Padding mode

Besides, we could set padding mode, and there are four padding modes we can use in pytorch :

1. constant padding :

Constant padding refers to filling the edges of an array with constant values (e.g., 0).



torch.nn.functional.pad — PyTorch 2.2 documentation

https://blog.csdn.net/weixin_42211626/article/details/122542323

What is CNN(Convolutional Neural Network)? - Padding mode

Besides, we could set padding mode, and there are four padding modes we can use in pytorch :

2. reflection padding :

Reflection padding involves symmetrically padding the edges of an array with respect to a particular row or column in the array.

Original

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

Input: 5x5

Padding: 1

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

Input: 7x7

torch.nn.functional.pad — PyTorch 2.2 documentation

https://blog.csdn.net/weixin_42211626/article/details/122542323

What is CNN(Convolutional Neural Network)? - Padding mode

Besides, we could set padding mode, and there are four padding modes we can use in pytorch :

3. replication padding :

Replication padding entails copying the edges of an array and filling them around the array.

Original

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

Input: 5x5

Padding: 1

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

Input: 7x7

torch.nn.functional.pad — PyTorch 2.2 documentation

https://blog.csdn.net/weixin_42211626/article/details/122542323

What is CNN(Convolutional Neural Network)? - Padding mode

Besides, we could set padding mode, and there are four padding modes we can use in pytorch :

4. circular padding :

Circular padding involves infinitely extending from top to bottom.

Original

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

Input: 6x6

Circular Padding

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

Input: 8x8

Padding: 1

torch.nn.functional.pad — PyTorch 2.2 documentation

https://blog.csdn.net/weixin_42211626/article/details/122542323

Convolution layer – effect of filter

Height = 3
Width = 3
Channel = 1
3*3*1
Count = 5

filter

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$
Ridge detection

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$
Ridge detection

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$
Sharpen

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$
Box blur

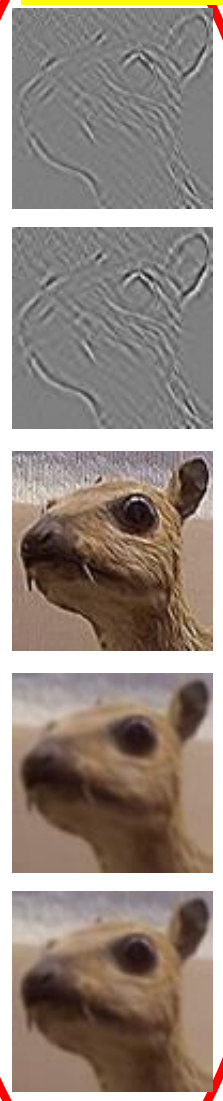
$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$
Gaussian blur

Input

Height = 28
Width = 28
Channel = 1
28*28*1
Count = 1

* =

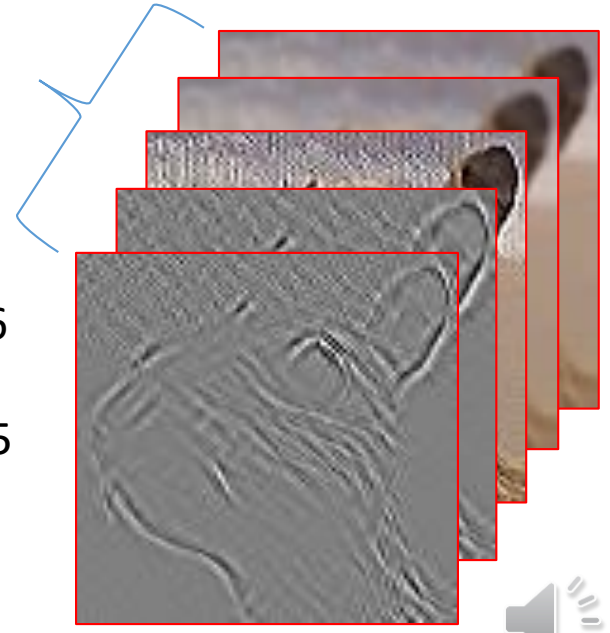
Output



- Use filter to extract object's **feature**.
- We can convolve the input with multiple filters to get a multi-channel output

(Output Channel = Filter Counts)

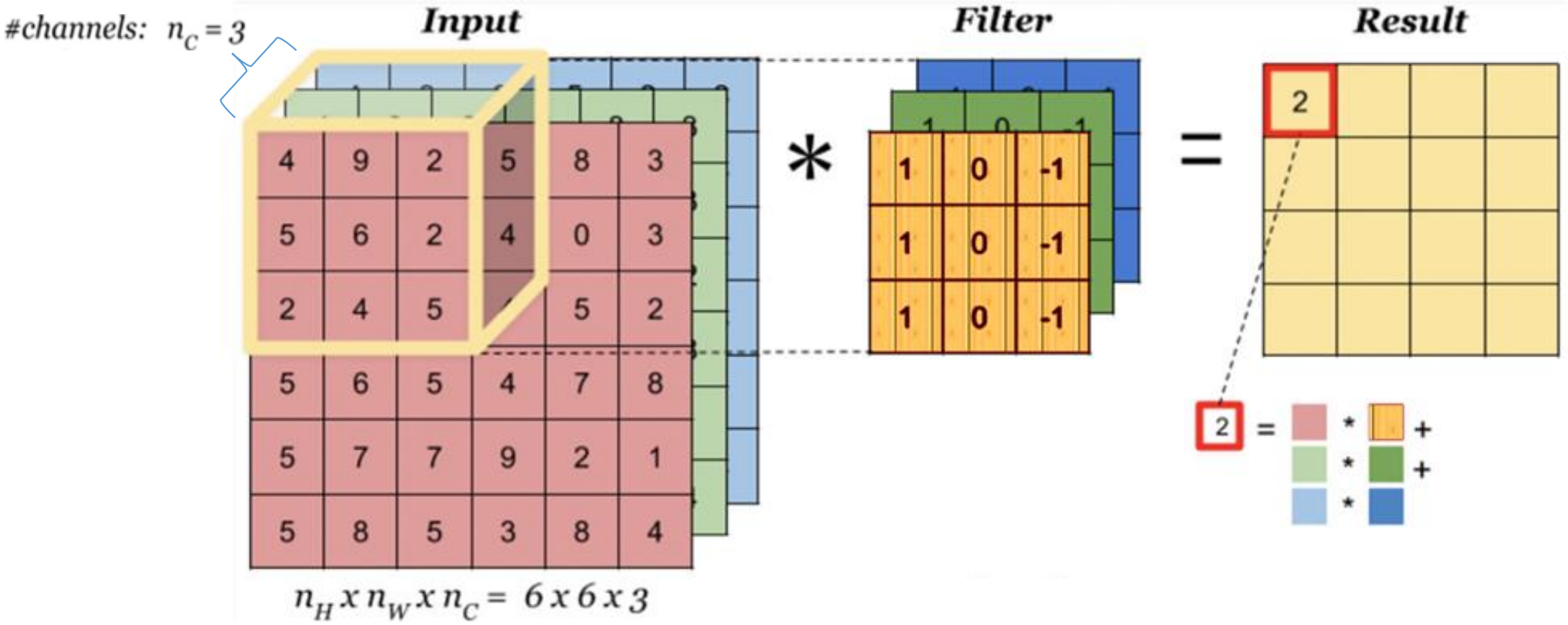
number of output channel : 5



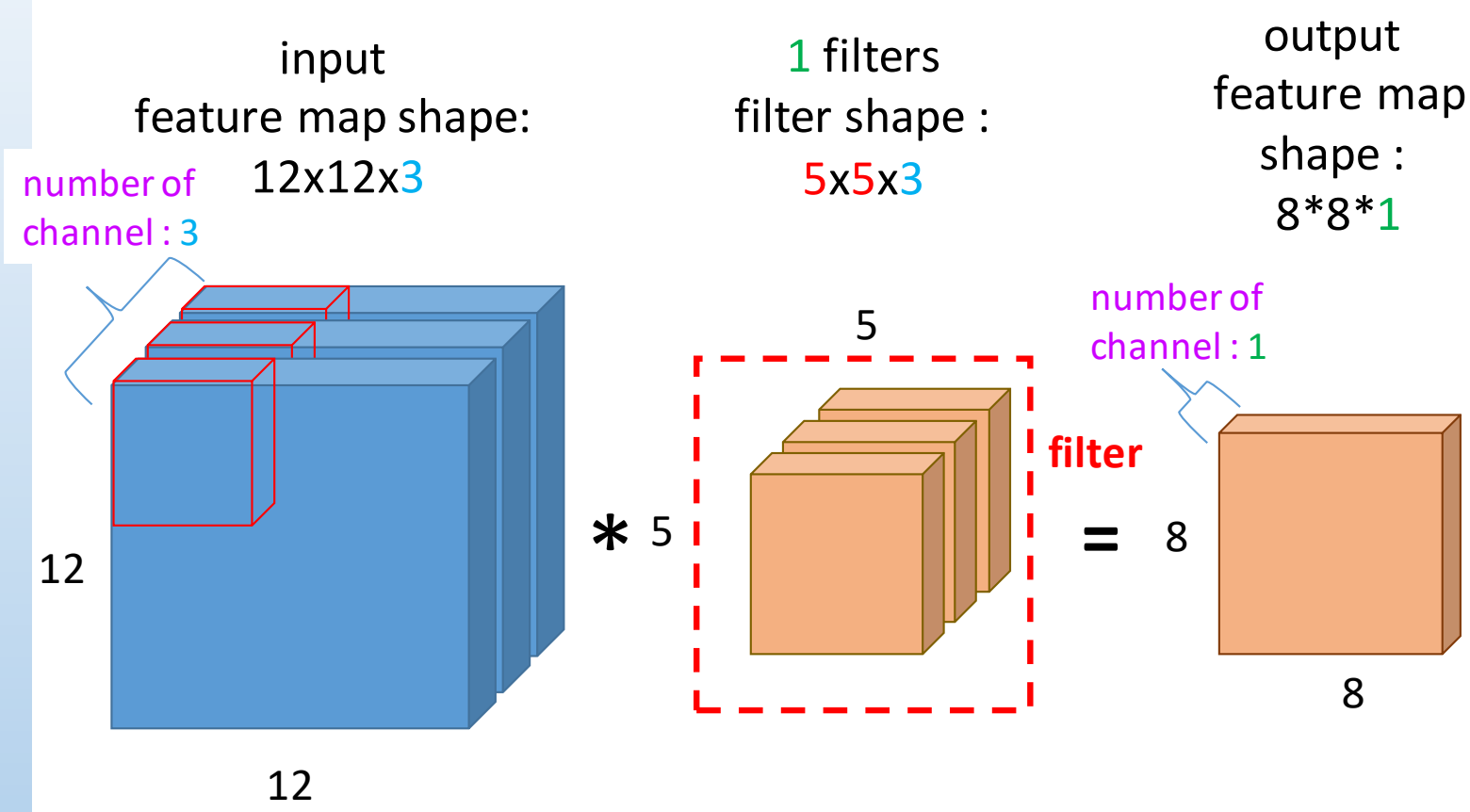
Height = 26
Width = 26
Channel = 5
26*26*5
Count = 1

Convolution layer – If input has multiple channel

- If input has multiple channel, filter's channel needs to be the same as the input.
- As in the previous ppt, 1 filter generates 1 channel output

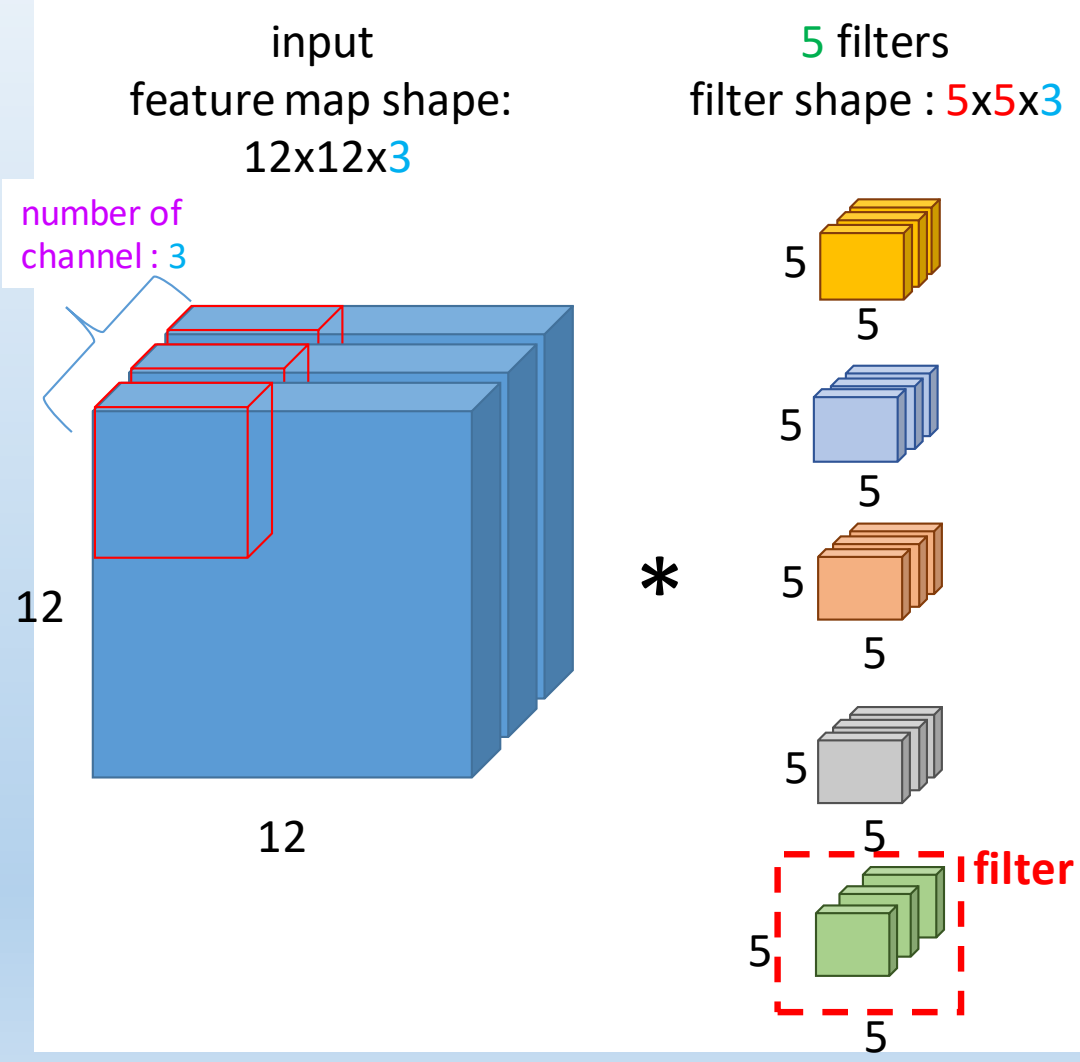


What is CNN(Convolutional Neural Network)? - Filter, Channel



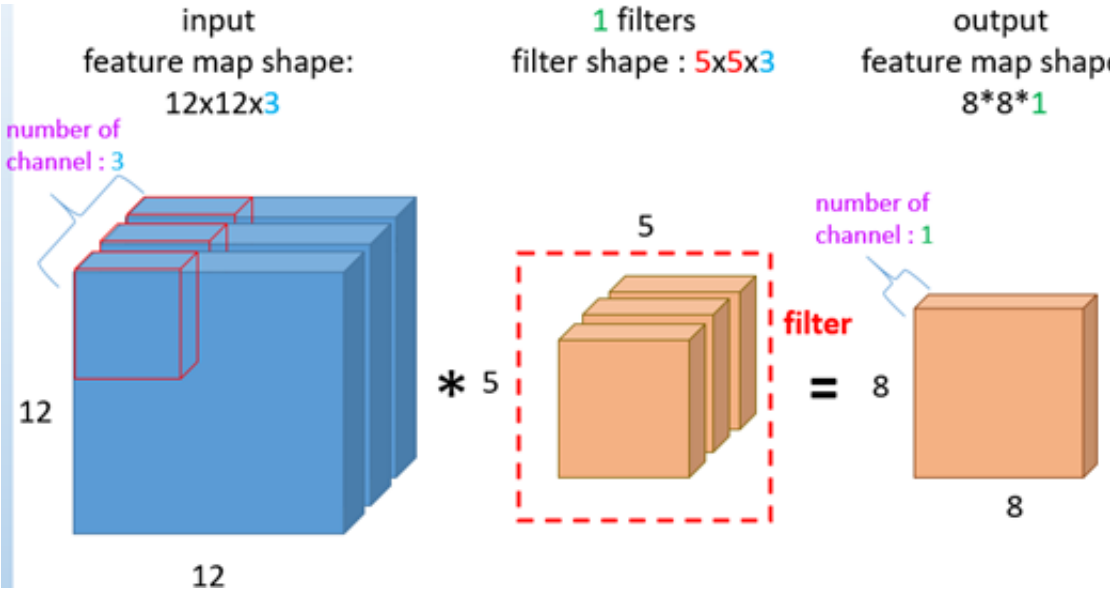
- The example is 3 channel input and 1 filter, so the filter has 3 channel, and output has 1 channel.
- **The number of channel of filter** must be same as **the number of channel of input**
- **The number of channel of output** is same as **the number of filter**

What is CNN(Convolutional Neural Network)? - Filter, Channel



- The example is 3 channel input and 5 filter, so each filter has 3 channel, and output has 5 channel.
- **The number of channel of filter** must be same as **the number of channel of input**
- **The number of channel of output** is same as **the number of filter**

What is CNN(Convolutional Neural Network)?
- Output feature map size



FORMULA:

$$\text{Height}_{out} = \text{floor}\left(\frac{\text{Height}_{in} + 2 \times \text{padding} - \text{kernel\_size}}{\text{stride}} + 1\right)$$

$$\text{Width}_{out} = \text{floor}\left(\frac{\text{Width}_{in} + 2 \times \text{padding} - \text{kernel\_size}}{\text{stride}} + 1\right)$$

In this example, we have

Height_{in}=12

Width_{out} =12

padding = 0

kernel_size = 5

stride = 1

Applying formula above, we get

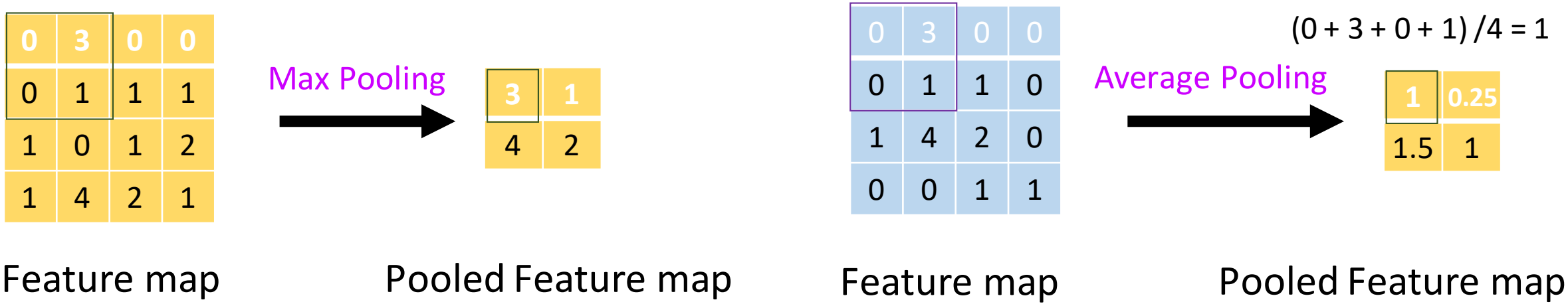
$$\text{Height}_{out} = 8 = \text{floor}\left(\frac{12 + 2 \times 0 - 5}{1} + 1\right)$$

$$\text{Width}_{out} = 8 = \text{floor}\left(\frac{12 + 2 \times 0 - 5}{1} + 1\right)$$

Much more detail [Conv2d — PyTorch 1.10 documentation]

Pooling layer

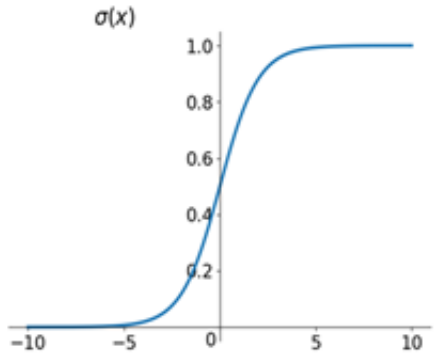
- Remain feature information and reduce parameters



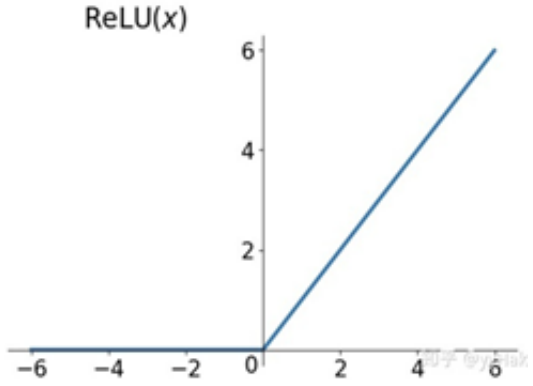
Effect of pooling :
https://youtu.be/fApFKmXcp2Y

Activation function

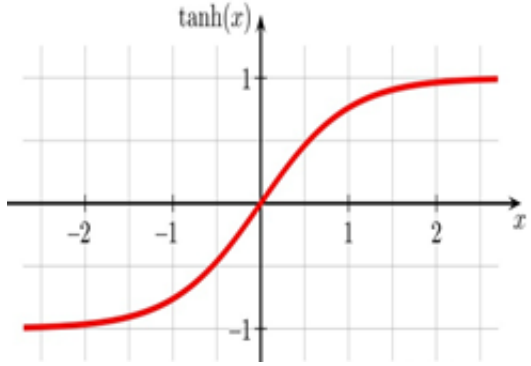
Sigmoid, $\sigma(x) = \frac{1}{1+e^{-x}}$



$\text{ReLU}(x) = max(0, x)$



$\tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$



Softmax $\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$ for $j = 1, \ldots, K$.

It's a mathematical function used in neural networks to introduce **non-linearity** into the model. This helps neural networks learn and represent complex patterns and relationships in data.

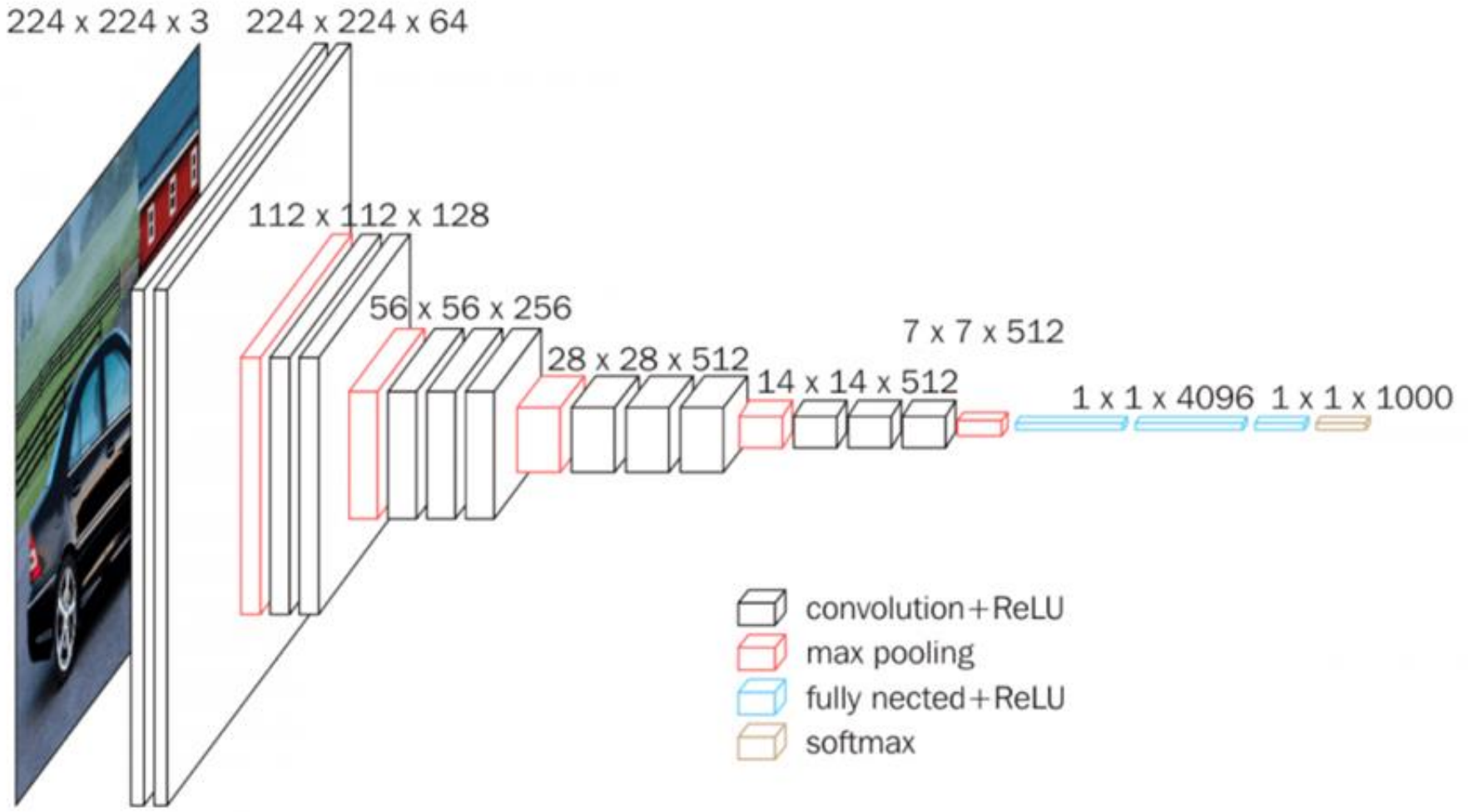
Common CNN network – LeNet5

- LeNet5

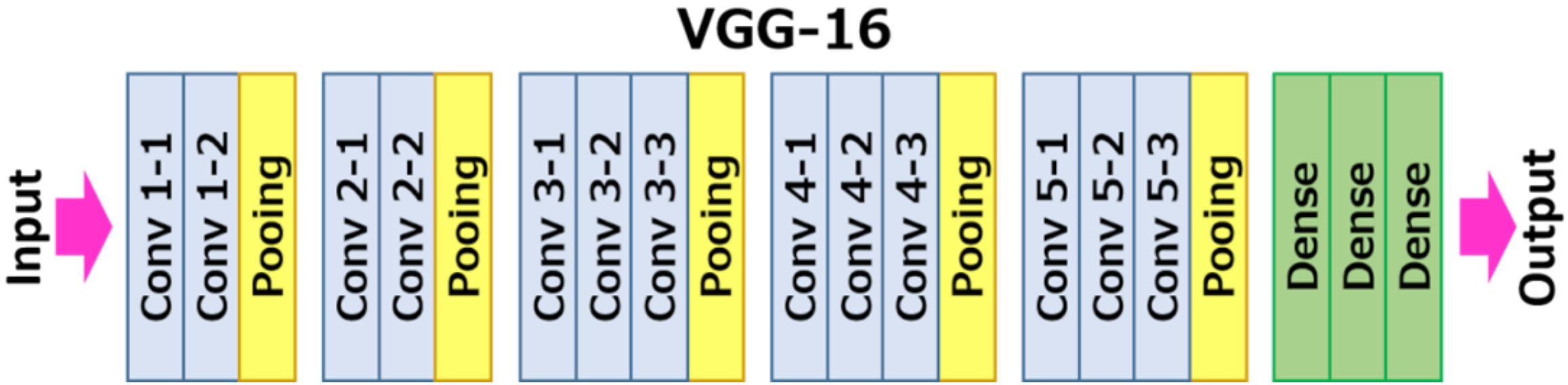
Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax

Common CNN network – VGG16

- VGG16



Common CNN network – VGG16



- Dense layer means fully connected layer

Common CNN network

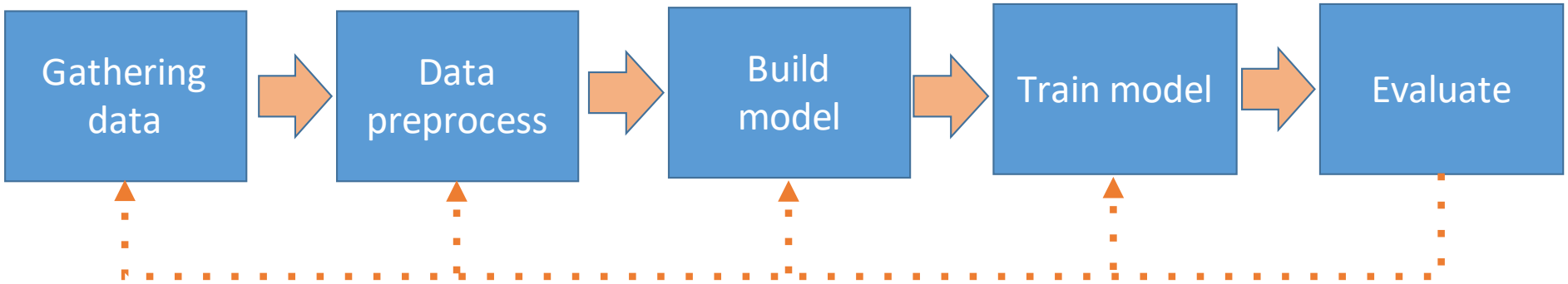
- AlexNet
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenetclassification with deep convolutional neural networks. Advances in neural information processing systems, 25.
- ResNet
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition(pp. 770-778).
- DenseNet•Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition(pp. 4700-4708).
- MobileNet
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXivpreprint arXiv:1704.04861.
- ShuffleNet
- Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE conference on computer vision and pattern recognition(pp. 6848-6856).

From lecture pdf

Import predefined model in pytorch

- You could also import predefined model from pytorch.
  - VGG16
  - ResNet50
  - MobileNet
  - EfficientNet
  - DenseNet121
  - …..

Classic flow for training a model



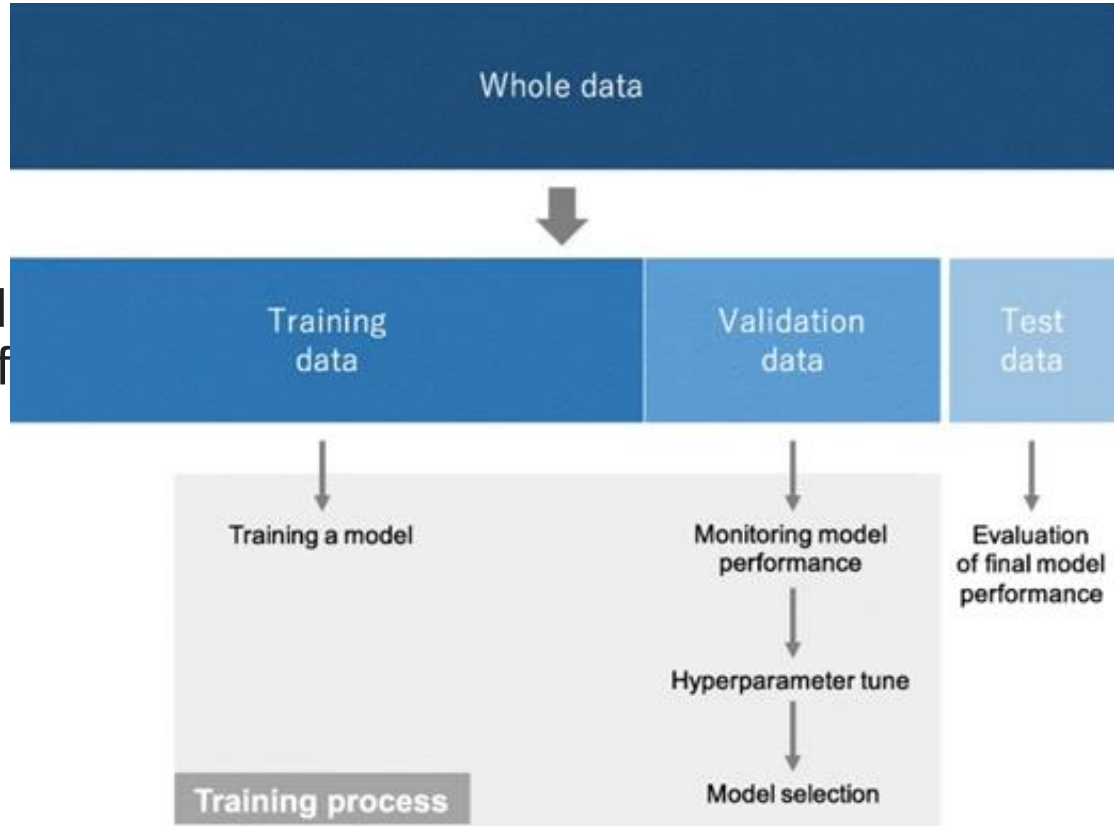
- **Gathering data** → From network or gather by yourself
- **Data preprocess** → Raw data will probably lead to bad classification performances
- **Build model** → Design a model for predicting data
- **Train model** → Learn good values for all the parameters from labeled training data
- **Evaluate** → Evaluate model could give a suitable response from its experience

Outline

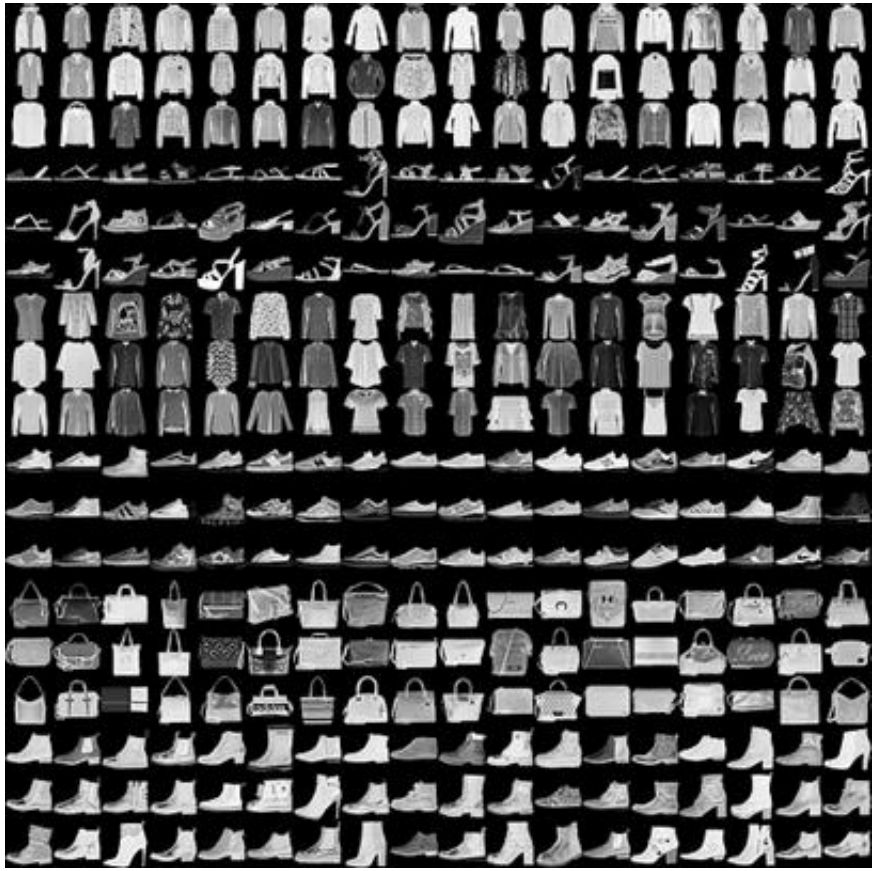
- Google Colab
- What is Pytorch?
- Neural Networks(NN)
- Convolutional Neural Networks(CNN)
- Dataset
- Advanced Topics
- Assignment

Dataset

- training set
  - A dataset of examples is used during the learning process and is used to fit the parameters(e.g., weights).
- validation set
  - A validation data set is a dataset of examples used to tune the hyperparameter (i.e. the architecture) of a model.
  - Validation set is not necessary.
- testing set
  - A testing set is used to evaluate the final ability of the model.
  - It should not be used as a basis for parameter adjustment, selection of features.



Dataset : Fashion-MNIST



Label	Description
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

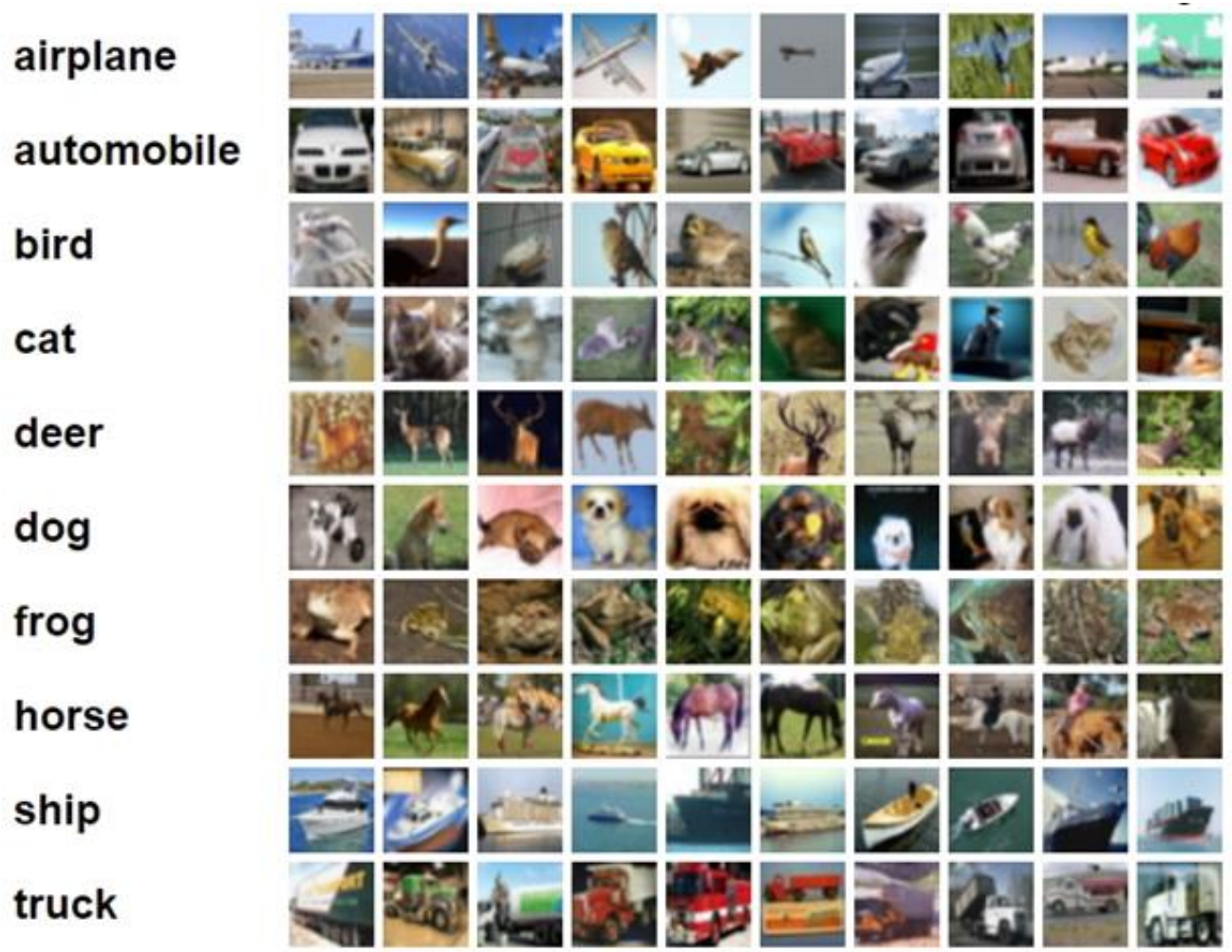
Why Fashion-MNIST ?

- MNIST is too easy and overused
- MNIST can not represent modern Computer Vision tasks

Fashion-MNIST

- 60,000 training data
- 10,000 testing data
- 10 categories (0~9).
- Each gray-scale image is 28x28.

Dataset : CIFAR-10



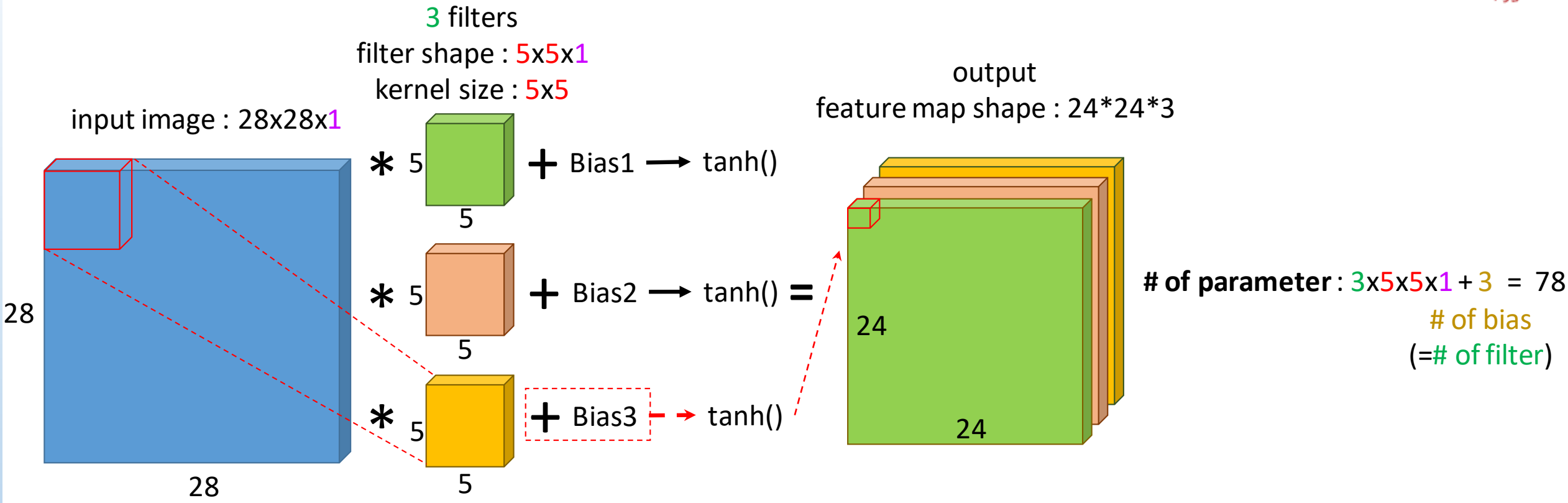
CIFAR-10

- 32x32 color
- 50,000 training images
- 10,000 test images
- labeled over 10 categories.

Outline

- Google Colab
- What is Pytorch?
- Neural Networks(NN)
- Convolutional Neural Networks(CNN)
- Dataset
- Advanced Topics
- Assignment

Layer 1 : Conv2D



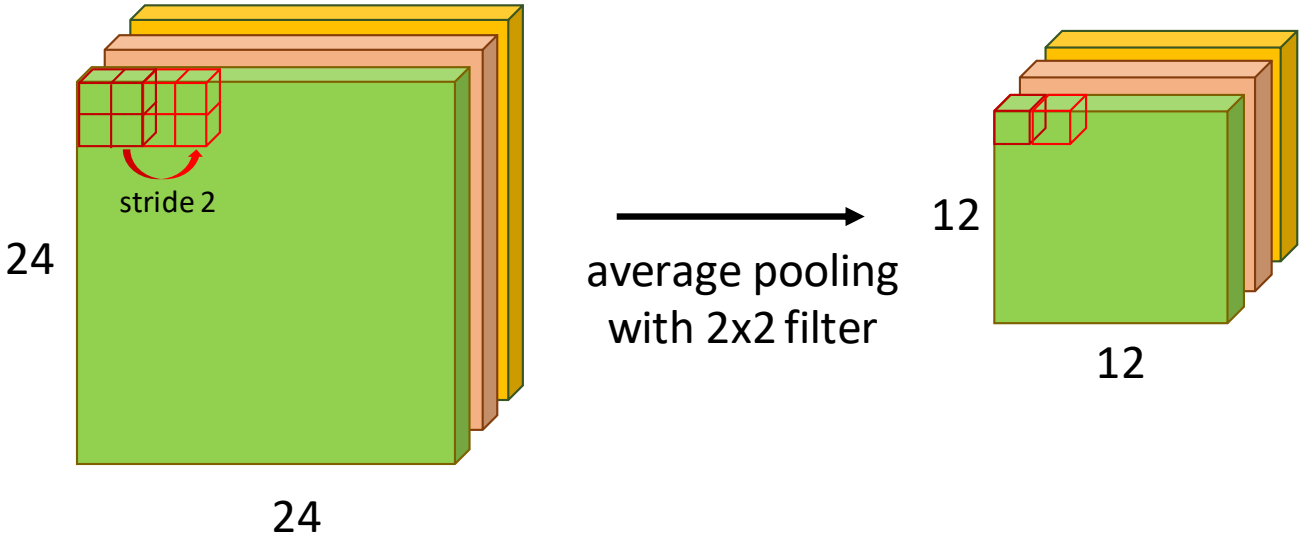

```
self.conv1 = nn.Conv2d(in_channels=1,out_channels=3,kernel_size=5)
```

- When using this layer as the first layer in a model, provide the keyword argument **input_shape.**

Layer 2 : Average Pooling

input
feature map shape : 24x24x3

output
feature map shape : 12x12x3

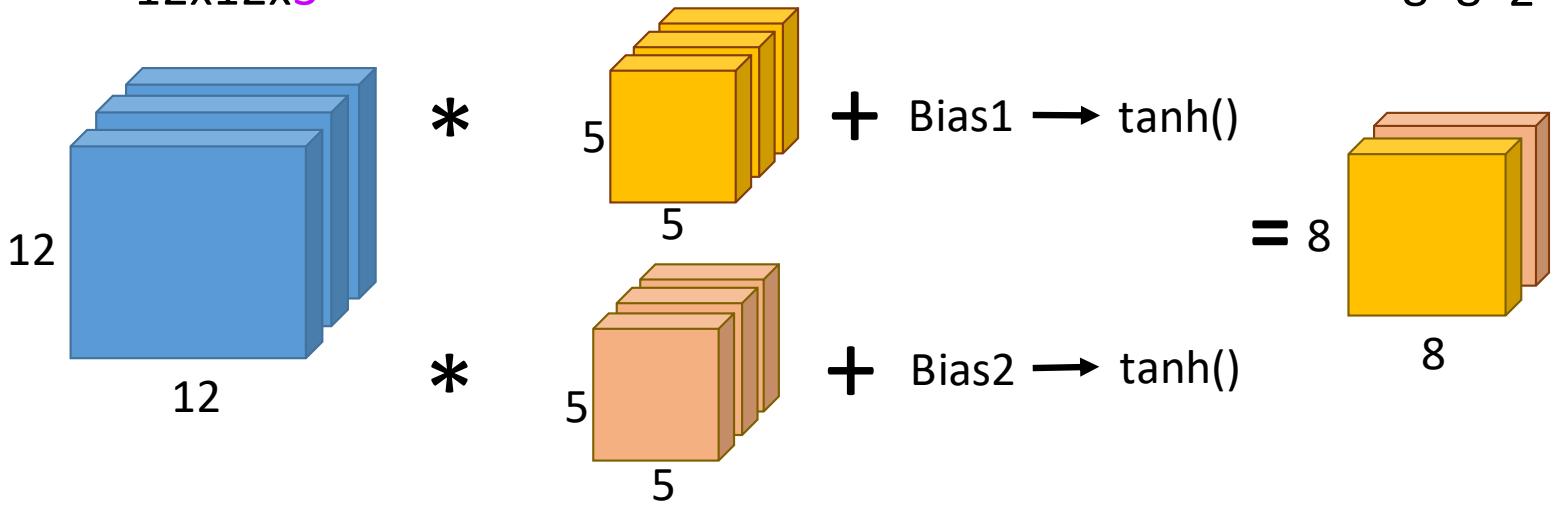



```
self.pool1 = nn.AvgPool2d(kernel_size =2,stride=2)
```

Layer 3 : Conv2D

input
feature map shape:
12x12x3

2 filters
filter shape : 5x5x3
kernel size : 5x5



output
feature map shape :
8*8*2

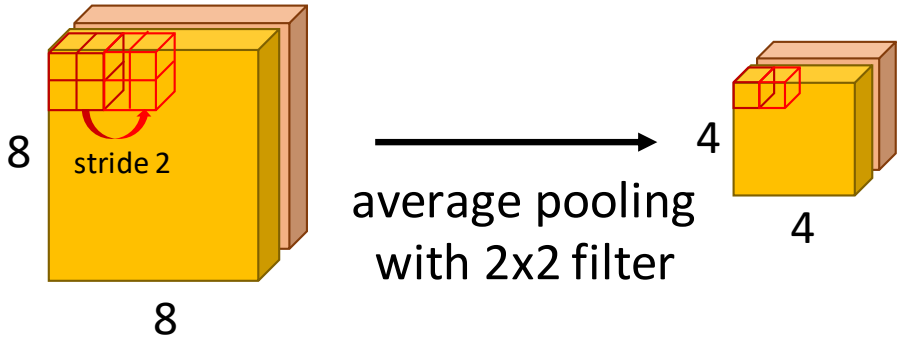
# of parameter : 2x5x5x3 + 2 = 152
# of bias
(=# of filter)

```
self.conv2 = nn.Conv2d(in_channels=3,out_channels=2,kernel_size=5)
```

Layer 4 : Average Pooling

input
feature map shape : 8x8x2

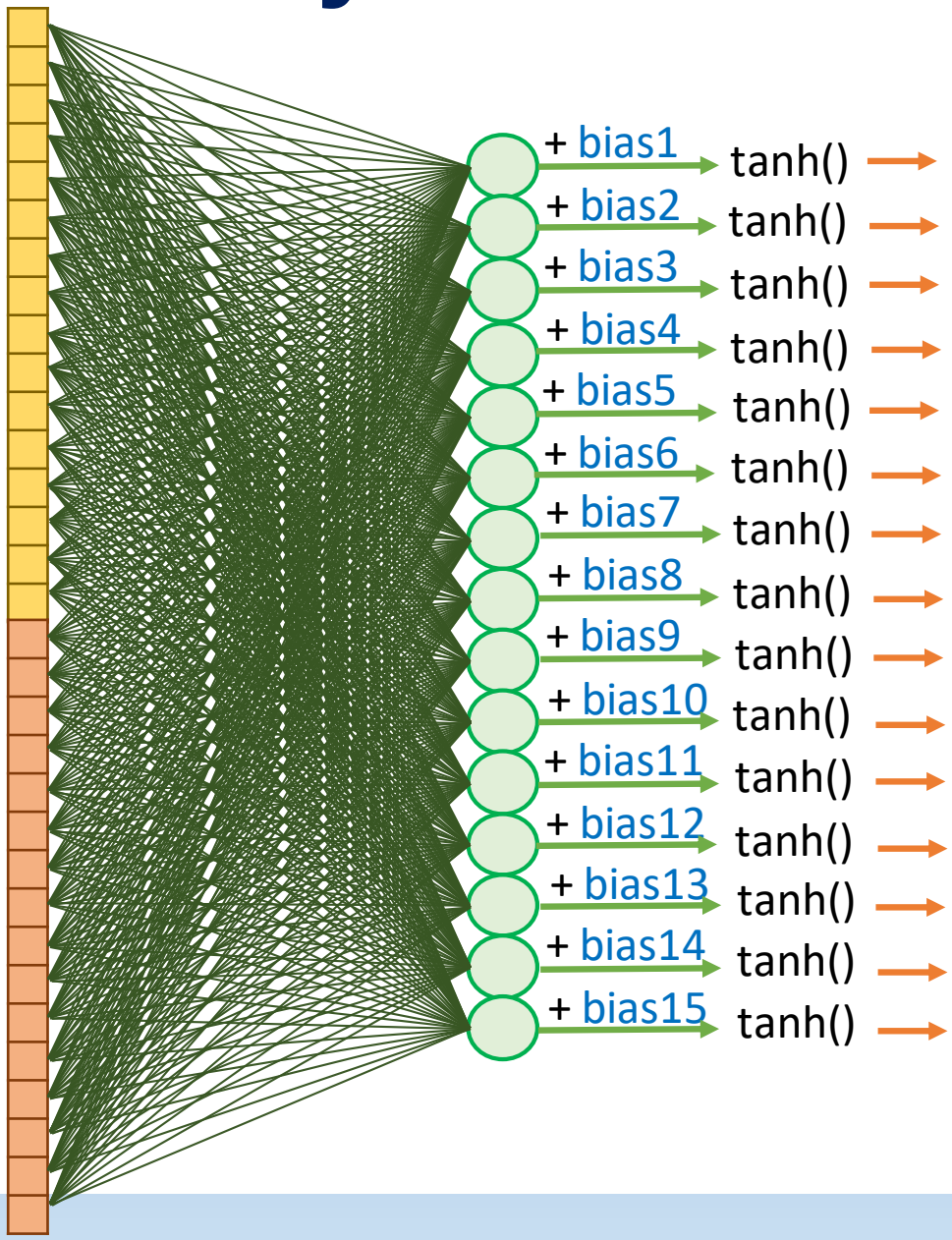
output
feature map shape : 4x4x2




```
self.pool2 = nn.AvgPool2d(kernel_size =2,stride=2)
```

Layer 6 : Fully Connected Layer

input
feature map
shape: 32

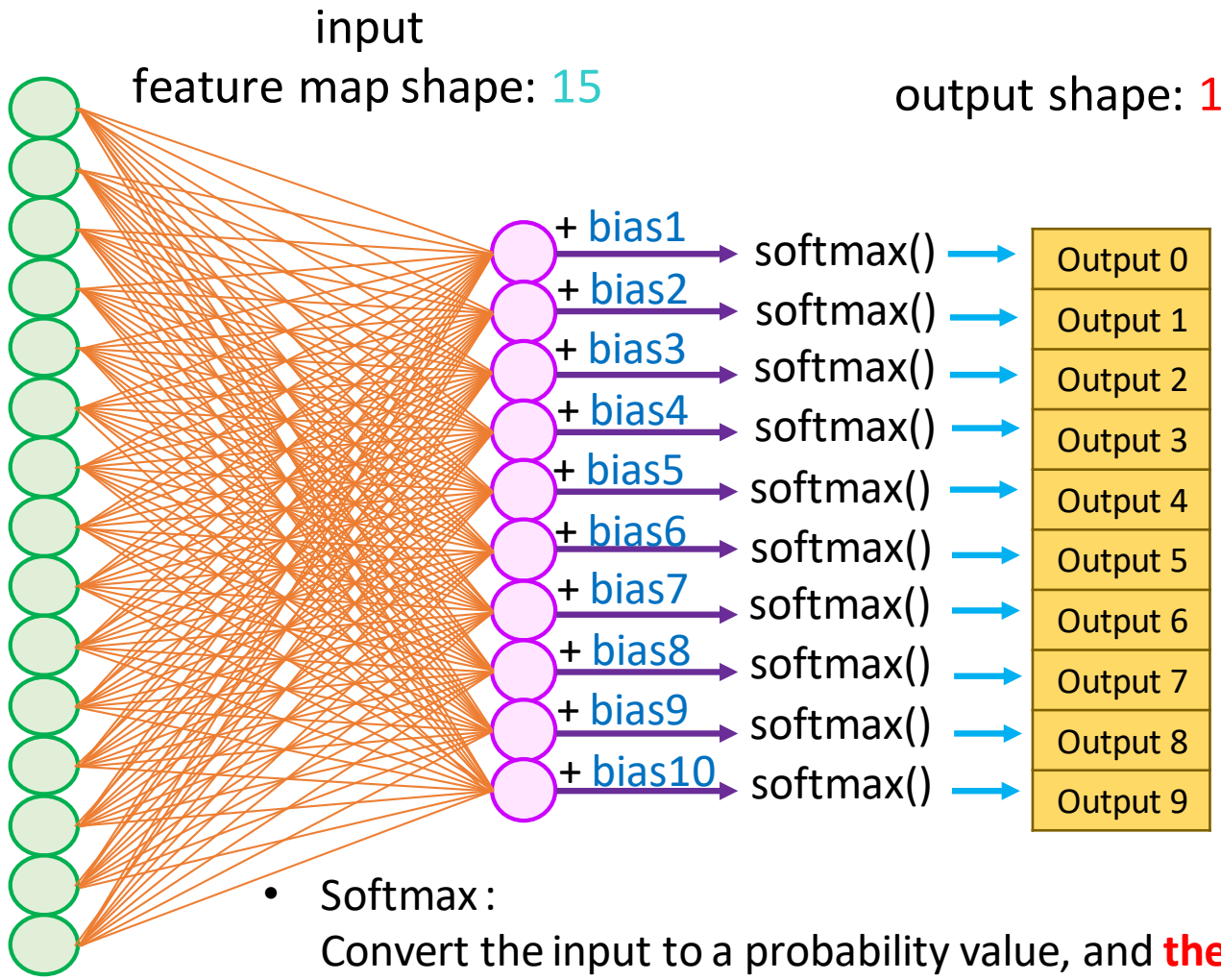


output
feature map shape:
15

# of parameter : 32 x 15 + 15 = 495
# of bias

- dense layer means fully connected layer

Layer 7 : Fully Connected Layer



# of parameter : 15 x 10 + 10 = 160
# of bias

- Softmax :
  Convert the input to a probability value, and **the sum of the probability of all output classes is equal to 1**
- **Output 0 + Output 1 + … Output 9 = 1**
- **Each output means the probability (confidense score) of the corresponding class**

Total Parameters

```
from torchsummary import summary
```

```
summary(net)
```

```
==========================================================================
Layer (type:depth-idx)                             Param #
==========================================================================
├─Conv2d: 1-1                                      78
├─AvgPool2d: 1-2                                   --
├─Conv2d: 1-3                                      152
├─AvgPool2d: 1-4                                   --
├─Linear: 1-5                                      495
├─Linear: 1-6                                      160
==========================================================================
Total params: 885
Trainable params: 885
Non-trainable params: 0
==========================================================================
```

number of filter

filter size number of bias (number of filter)

3x5x5x1 + 3

2x5x5x3 + 2

32x15 + 15

15x10 + 10

number of bias

total parameters :

78 + 152 + 495 + 160 = 885

Total MACs

THOP: PyTorch-OpCounter

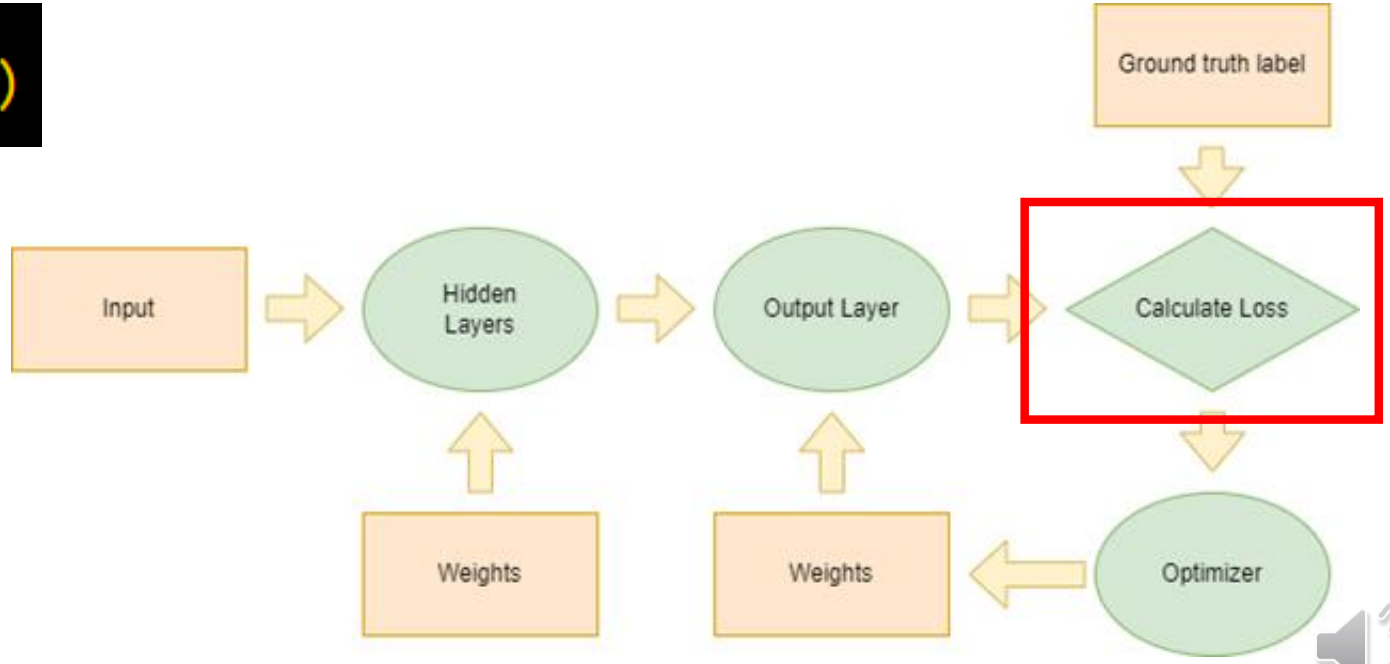
```
from thop import profile
input1 = torch.randn(1,3,32,32).cuda()
MACs, params = profile(net, inputs=(input1, ))
print('MACs = ' + str(MACs/1000**3) + 'G')
print('Params = ' + str(params/1000**2) + 'M')
```

```
MACs = 0.027222016G
Params = 0.199242M
```

Loss function

- To evaluate if model is good/bad.
- Loss means residual between ground truth value and predict value. Thus, we want to minimize residual.
- Choose cross entropy to model our classification problem

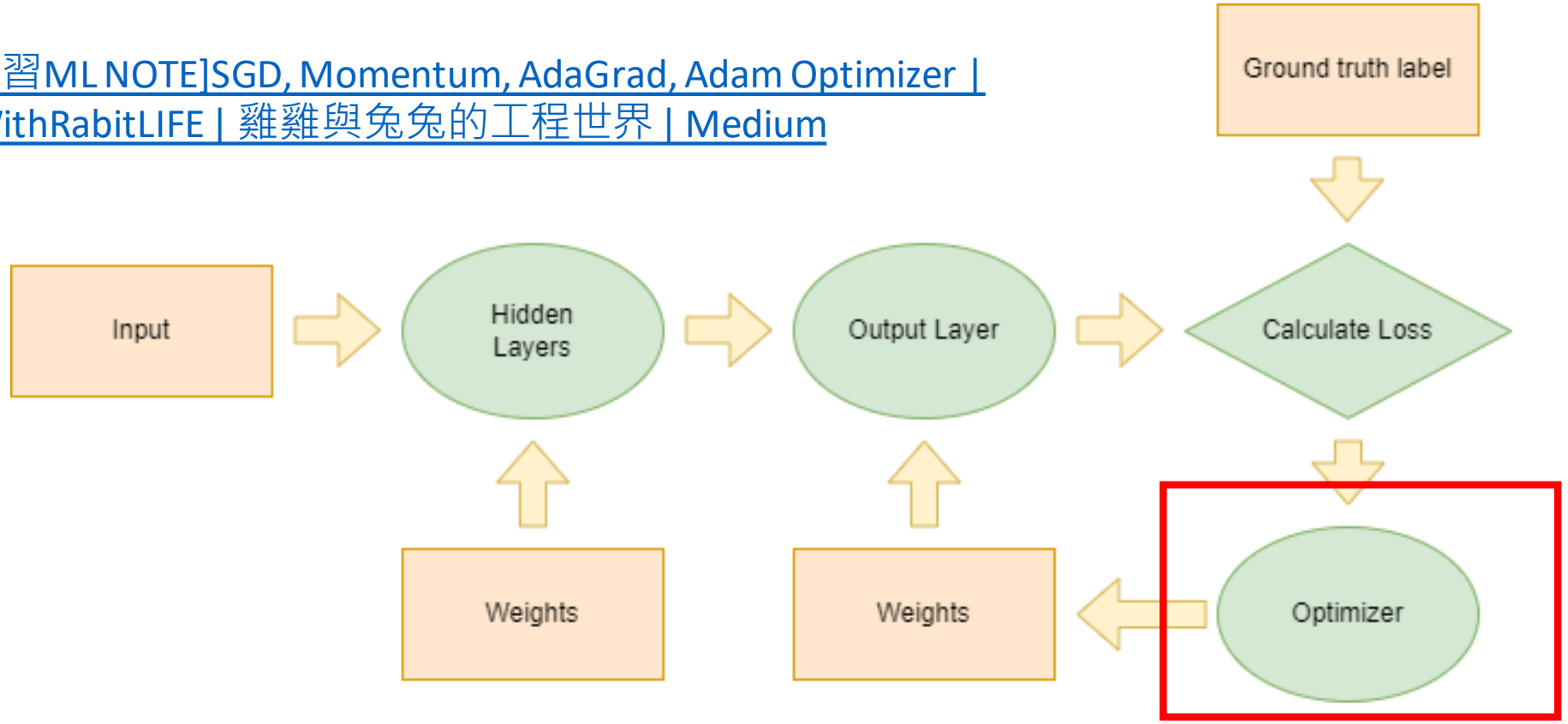
```
criterion = nn.CrossEntropyLoss()
```



Optimizer

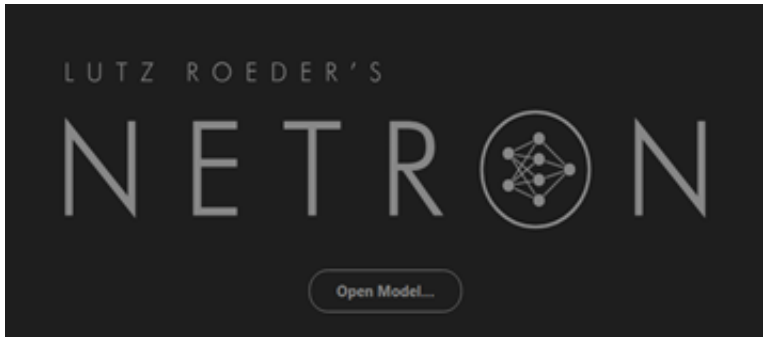
- **Optimizers** are algorithms or methods used to minimize an error function(*loss function*)or to maximize the efficiency of production.

[機器學習ML NOTE]SGD, Momentum, AdaGrad, Adam Optimizer | by GGWithRabitLIFE | 雞雞與兔兔的工程世界 | Medium



Netron - Introduction

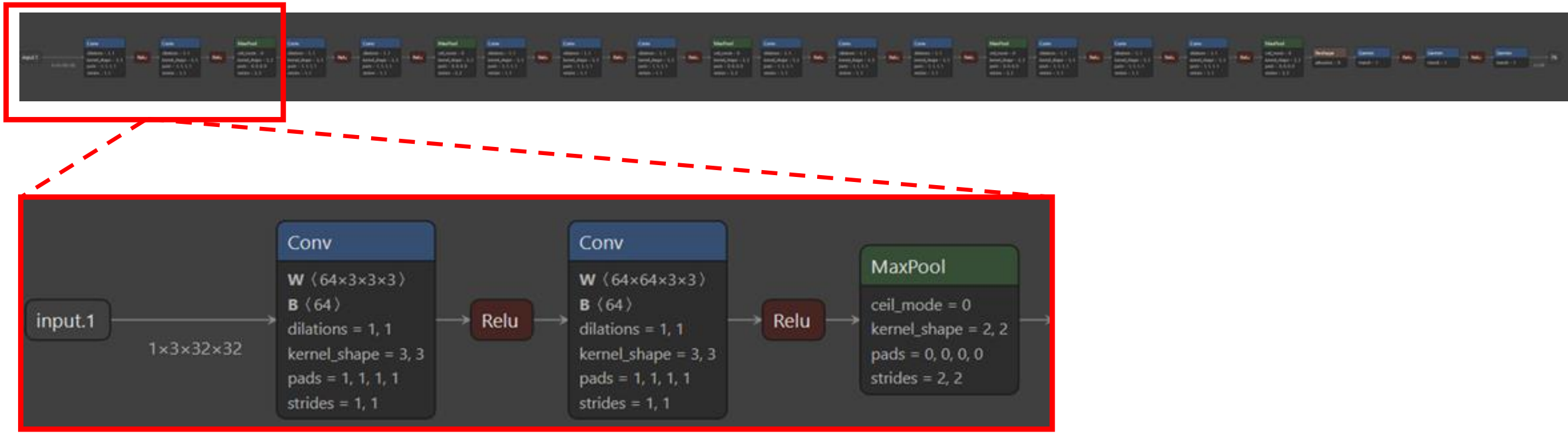
- Netron is a tool for visualizing deep learning models.
- It helps you understand and explore the structure of neural networks, convolutional neural networks (CNNs), and other models.
- You can drag and drop trained model files (such as ONNX, Keras, Core ML, TensorFlow Lite, etc.) into Netron, and it will parse the model and display it in a visual format, helping you quickly understand the model architecture.



lutzroeder/netron: Visualizer for neural network, deep learning and machine learning models (github.com)

Netron – Model visualization

- Using NETRON to visualize the each layer of the model



Outline

- Google Colab
- What is Pytorch?
- Neural Networks(NN)
- Convolutional Neural Networks(CNN)
- Dataset
- Advanced Topics
- Assignment

Assignment

Parameter / MAC	Cifar Point	Fashion point
Rank 1% - 25%	5 / 5	2.5 / 2.5
Rank 25% - 50%	4 / 4	2 / 2
Rank 50% - 75%	3/3	1.5 / 1.5
Rank 75%-100%	2/2	1 / 1

- Requirement :
- Need to design 2 CNN models for 2 different datasets listed below.
- Each models achieves the specified **accuracy** respectively.

  (You can get the basic grade 40 if you meet the requirement of each dataset, otherwise get 0 point)
- According to the number of **parameters** and **MACs**, you will be rated 40-55, the **fewer parameter** and **MAC** are the better.
- We hope to maintain a certain level of precision while reduce hardware cost.

1. **Fashion MNIST dataset, an alternative to MNIST (classification)**
   - **Accuracy >= 85% (for test data)**
2. **CIFAR10 small images classification dataset (classification)**
   - **Accuracy >= 75% (for test data)**

Assignment

You can modify the two parts in the codes in the following :

Model Design

```python
import torch
import torch.nn as nn

class DNN(nn.Module):
    def __init__(self):
        super(DNN, self).__init__()

        ###### TODO : You can modify model architechture ######
        self.features = nn.Sequential(
            nn.Conv2d(3, 64, kernel_size=3, padding=1),
            nn.BatchNorm2d(64),
            nn.ReLU(inplace=True),

            nn.Conv2d(64, 64, kernel_size=3, padding=1),
            nn.BatchNorm2d(64),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=2, stride=2),
```

Configuration

```python
### TODO : You can modify the configuration for model training ###

# For the classification task, we use cross-entropy as the measurement of performance.
criterion = nn.CrossEntropyLoss()

# Initialize optimizer, you may fine-tune some hyperparameters such as learning rate on your own.
optimizer = optim.SGD(net.parameters(), lr=0.001, momentum=0.9)

# The number of batch size.
batch_size = 512

# If no improvement in 'patience' epochs, early stop.
patience = 10

# The number of training epochs
n_epoch = 5

_exp_name = "sample"
```

Assignment

You should save the model in .onnx format :

DNN.onnx file

Downloading DNN.onnx file, and using Netron to visualize the model

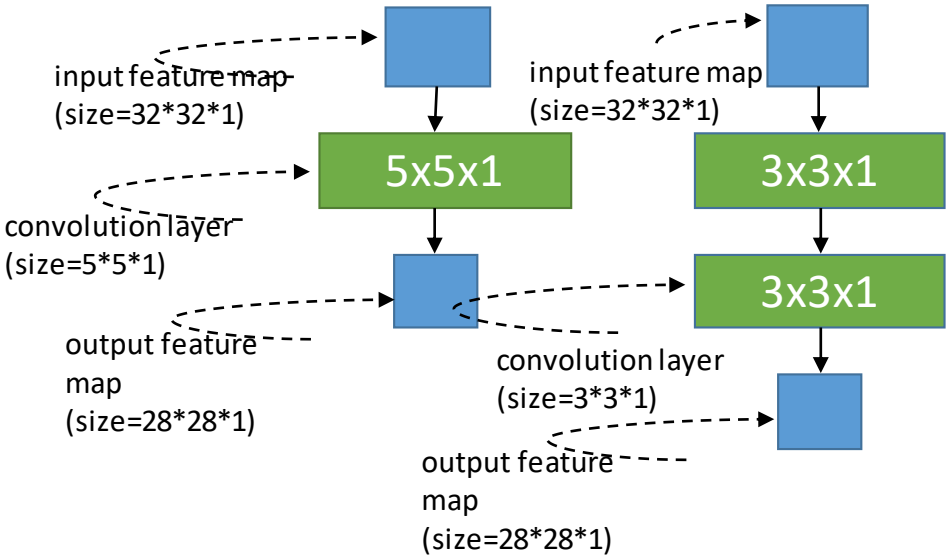
```
# Some standard imports
import io
import torch.utils.model_zoo as model_zoo
import torch.onnx


########## YOU DESIGN #########


########## YOU DESIGN #########
```

Question List

- 假設有一個size為 32*32*1 input feature map ，若要用convolution來產生一個 28*28*1 output feature map ，請比較使用以下兩種 Kernal size 計算時，所需的 parameter數量 和 MAC數量，須包含計算過程才予以計分。(16 points)

  (p.s no padding & stride=1)

  - [5x5x1 CONV] (p.s 一個convolution layer)
  - [3x3x1 CONV] + [3x3x1 CONV] (p.s 兩個convolution layer串聯)



- 藉由 Netron 將你所設計的兩個模型視覺化並截圖 (5 points)
- 畫出 train/val loss curve ，並判斷是否overfitting? (10 points)
- 你覺得這個lab有什麼可以改進的地方以及你的心得? (4 points 認真表達心得一律滿分)
- 實作上你做了甚麼調整 (learning rate, image augmentation 等等) 維持精準度,減少MAC, pararmeters, FLOP 和避免模型 overfitting? 加不同的data augmentation 會在testing data 精準度上面有甚麼影響? 請具體以文字和數據描述。(10 points)

Assignment Format

- Upload the assignment to Moodle
- File format: (total 3 files)
  - 1. StudentID_Name_mnist.ipynb
  - 2. StudentID_Name_cifar10.ipynb
  - 3. StudentID_Name_Lab1.pdf
  - ex: N123456789_蔡小明_cifar10.ipynb

- Deadline: 3/14(Thu.)23:59