AI workshop

2023.06.29

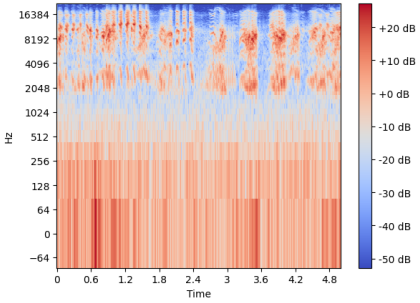
張庭維

Outline

- Introduction
- Basic Skill
- Build a Neural Network
- Today’s Project
- Discussion and Summary

Introduction

- A primary lecture for beginners of Artificial Intelligence and Python programming
  - Google Colab is recommended in Today's workshop
  - Today's project: Animal sound classification



Dog

All the material can be found here

https://github.com/Ting-Wei-Chang626/Primary_AI.git

www.aiengineer.tw

Basic Python Skill

Introduction of OOP in python

- Class (類別)
- Object (物件)
- Attribute (屬性)
- Constructor (建構式)
- Method (函式)

```
class  sample:
    # default constructor
    def __init__(self):
        # initializing variable instance
        self.number=1001
        # a method
    def print_method(self):
        print("number variable : ",self.number)
obj=sample()
obj.print_method()
```

Class (類別) and Object (物件)

- **A class is a user-defined blueprint or prototype from which objects are created.**

Building Class

Building 1

- SRC
- 1,600 million
- 6 story

Building 2

- RC
- 1,200 million
- 5 story

Constructor (建構式)、Attribute (屬性)

- **The constructor is a method that is called when an object is created.**

```
class Net:
    meow = 0
    # default constructor
    def __init__(self,):
        self.l1 = nn.Linear(128, 8)
        self.id = 0
```

```
my_net = Net()
my_net.id = 10
```

```
class Net:
    # default constructor
    def __init__(self, pretrain=False):
        self.l1 = nn.Linear(128, 8)
        self.id = 0
        self.load_init_weight = pretrain
```

```
my_net = Net(pretrain=True)
my_net.id = 10
print (my_net.load_init_weight)

>> True
```

Refer to demo_attribute.ipynb

Method

1. **Instance method**
2. **Class method**
3. **Static method**
4. **Abstract method**

Refer to demo_method.ipynb

Inheritance

- Inheritance allows us to define a class that inherits all the methods and properties from another class.

```
# define a superclass
class super_class:
    # attributes and method definition

# inheritance
class sub_class(super_class):
    # attributes and method of super_class
    # attributes and method of sub_class
```

Always used to create your own dataset and model

```
class Net(nn.Module):
    def __init__(self,):
        ........
```

Refer to demo_inheritance.ipynb

The philosophy of programming

- **Thinking hard**
- **Clear data structure**
- **Debugging logically**
- **Google is the best friend**
- **Never ever write something you can’t understand**

www.aiengineer.tw

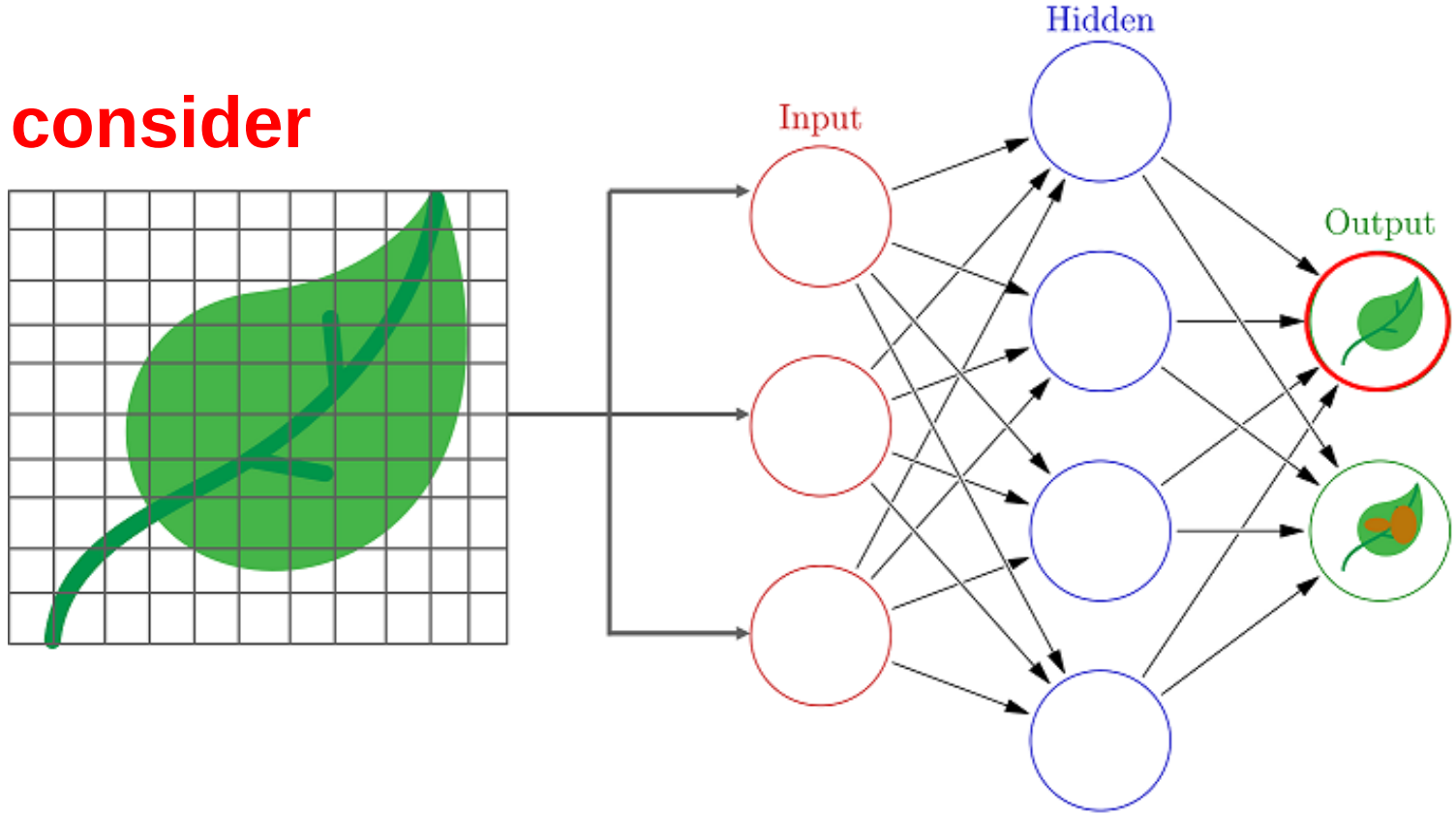
Basic Neural Network

Neural Network

Hint:

For research you should consider

1. Clear io
2. Computation limitation
3. Tiny dataset test
4. SOTA Model test

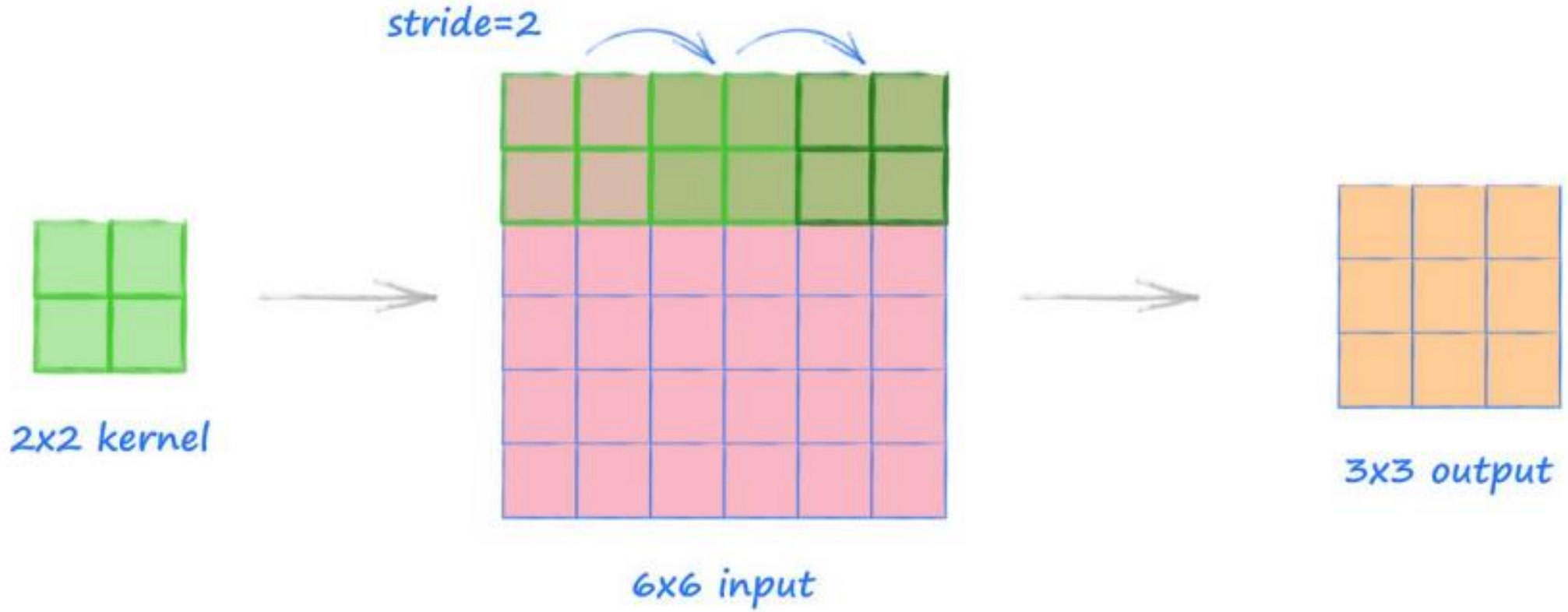


Layers

- Conv2d
  (https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html)
- BatchNorm2d
  (https://pytorch.org/docs/stable/generated/torch.nn.BatchNorm2d.html)
- Dropout
  (https://pytorch.org/docs/stable/generated/torch.nn.Dropout.html)
- Linear
  (https://pytorch.org/docs/stable/generated/torch.nn.Linear.html)

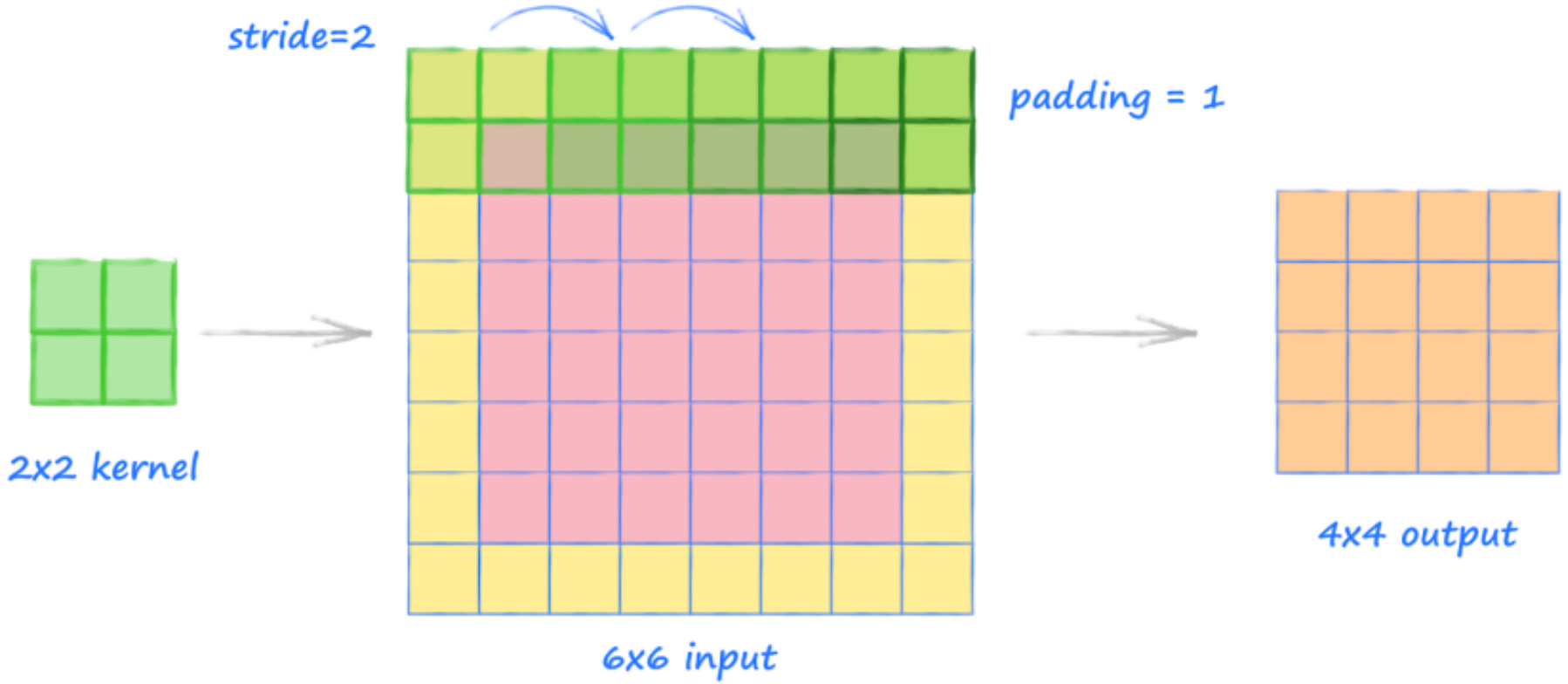
Conv2d

nn.Conv2d(in_channels, out_channels, kernel_size=2, stride=2)



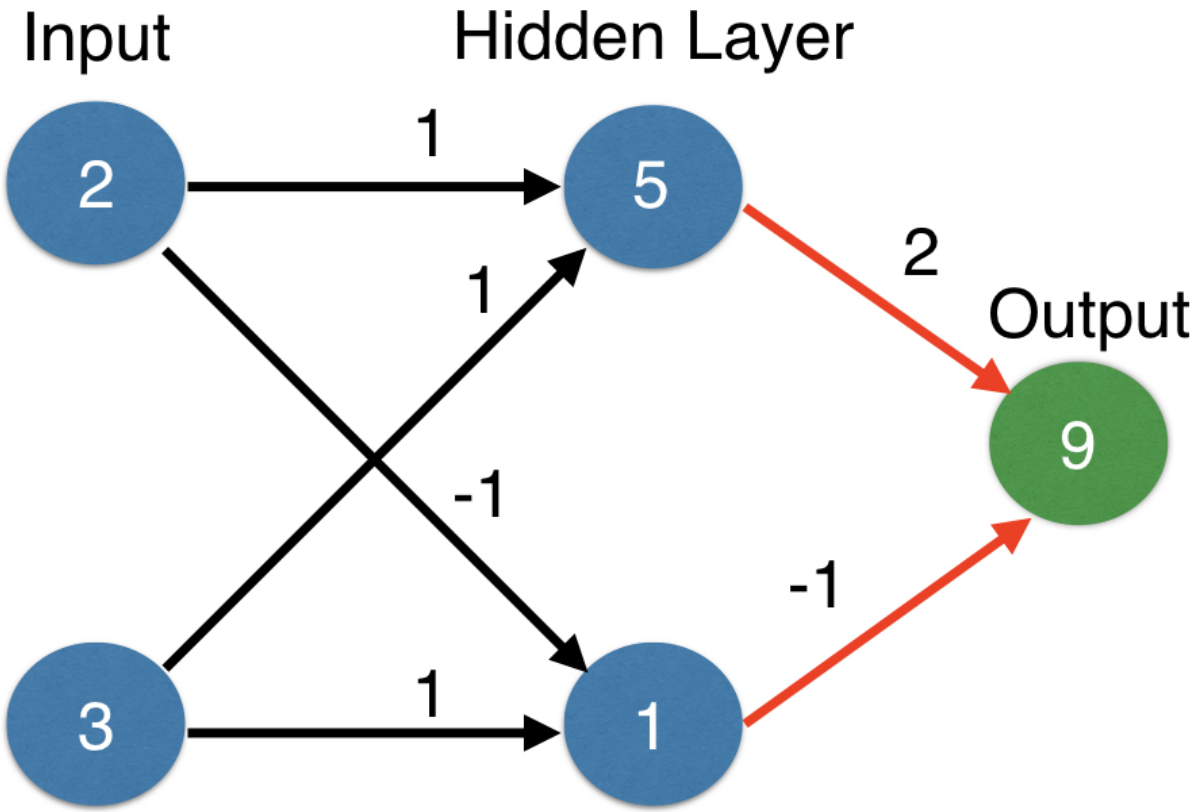
Conv2d

nn.Conv2d(in_channels, out_channels, kernel_size=2, stride=2, padding=1)

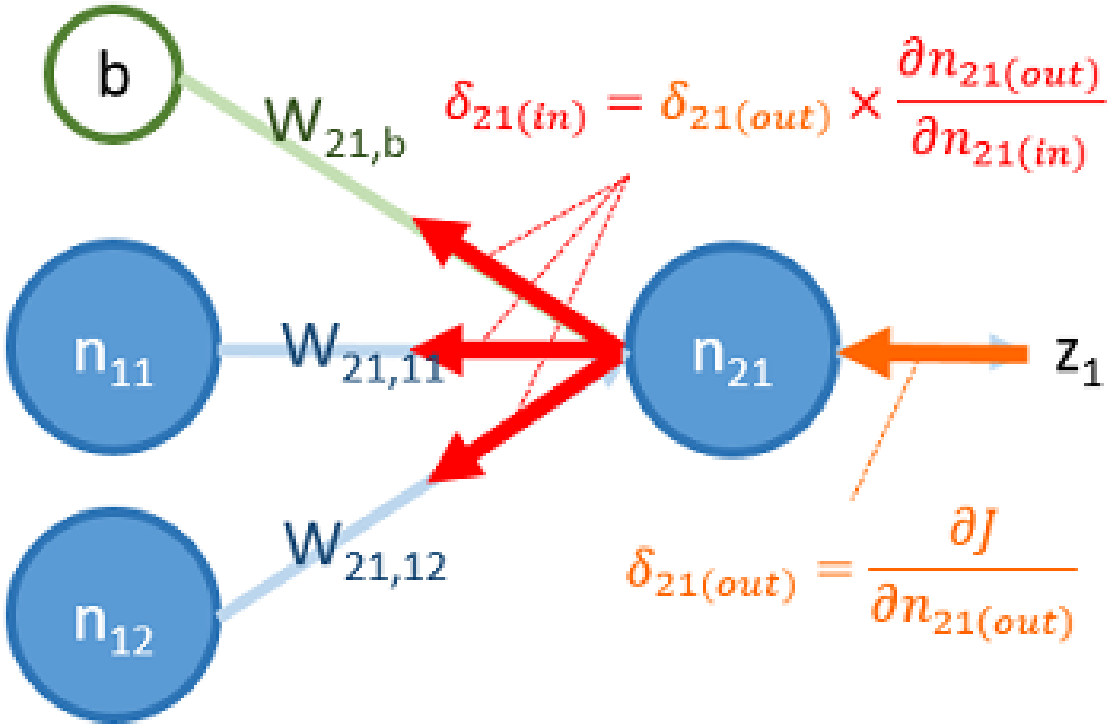


Forward Propagation / Backward Propagation

feedforward

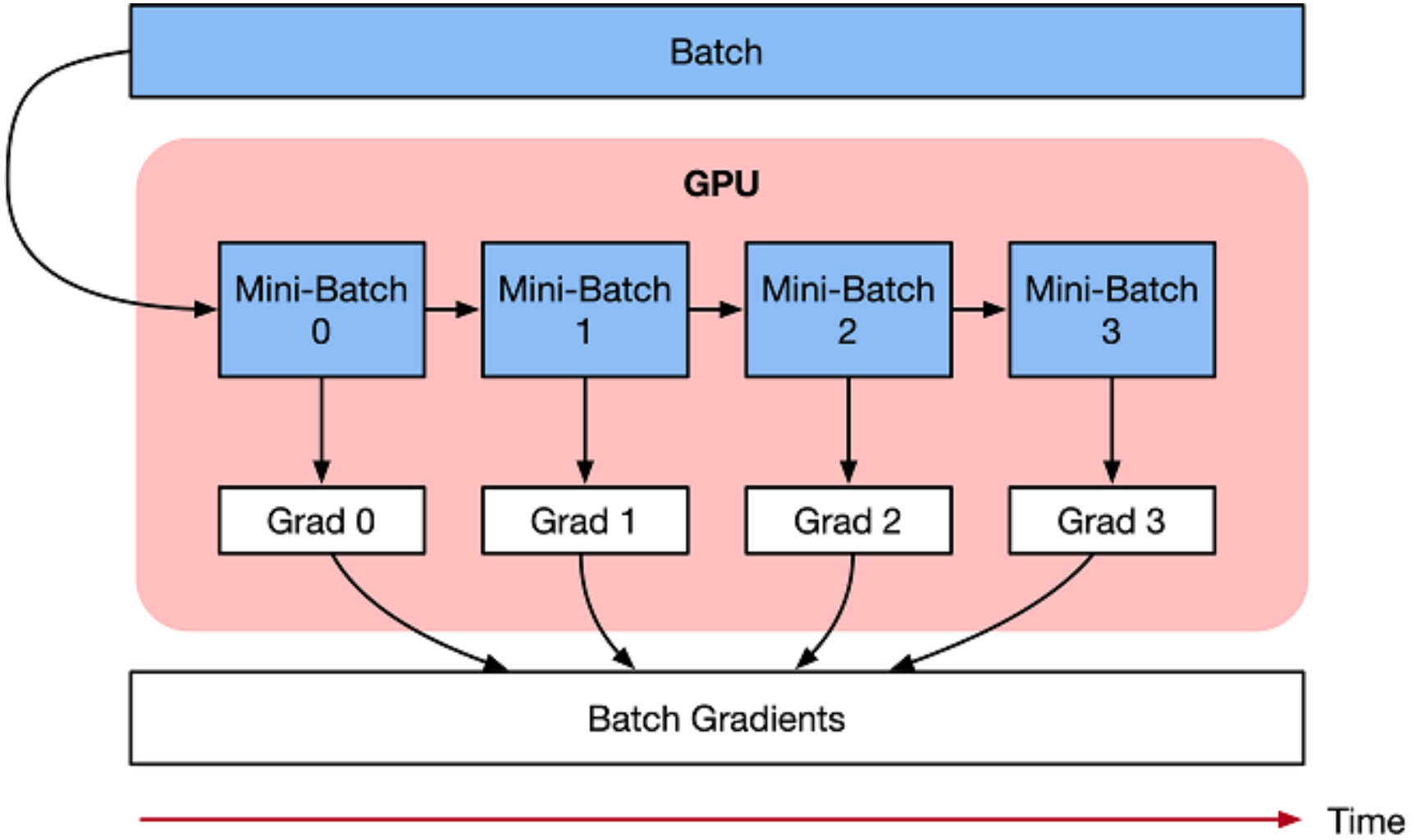


backforward



Refer to demo_forward_propagation.ipynb

Gradient accumulation



Source: https://www.mindspore.cn/tutorials/experts/en/r1.8/others/gradient_accumulation.html

Step-by-Step Conolution Neural Network

1. Load and preprocessing the data (Dataset, Dataloader)
2. Define a Convolutional Neural Network (torch.nn.Module)
3. Define a loss function (loss, Optimizer)
4. Train the network on the training data (Forward / Backward propagation)
5. Test the network on the test data

Refer to demo_mnist.ipynb

Project -Animal Sound Classify

- **Target：Build an image classifier to classify animal sound**

We'll learn

1. basic audio processing skill
2. create the custom dataset
3. how to build a NN model
4. perform training & testing on Colab

Source: Audio Classification using Librosa and Pytorch | by Hasith Sura | Medium

www.aiengineer.tw

Data processing

Dataset

- The **ESC-50 dataset** is a labeled collection of **2000** environmental audio recordings.
- Animal sounds occupy 1/5 (400/2,000)

Animal	Natural soundscapes & water sounds	Human, non-speech sounds	Interior/domestic sounds	Exterior/urban noises

Dog	Rooster	Pig	Cow	Frog
Cat	Hen	Insects	Sheep	Crow

Basic audio processing

- **Visualize the audio**

  step1: mount the google drive

  step2.1: librosa.load(path)

  step2.2: check the wave plot

  step3: convert the audio to fixed length

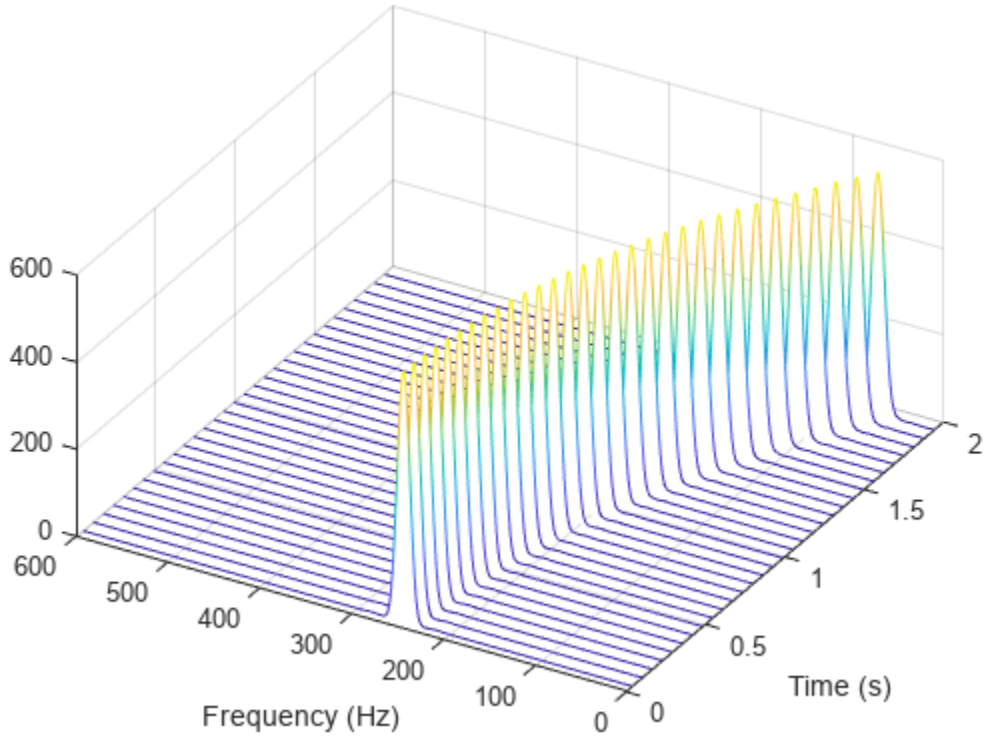
  step4: get the mel-scaled spectrogram

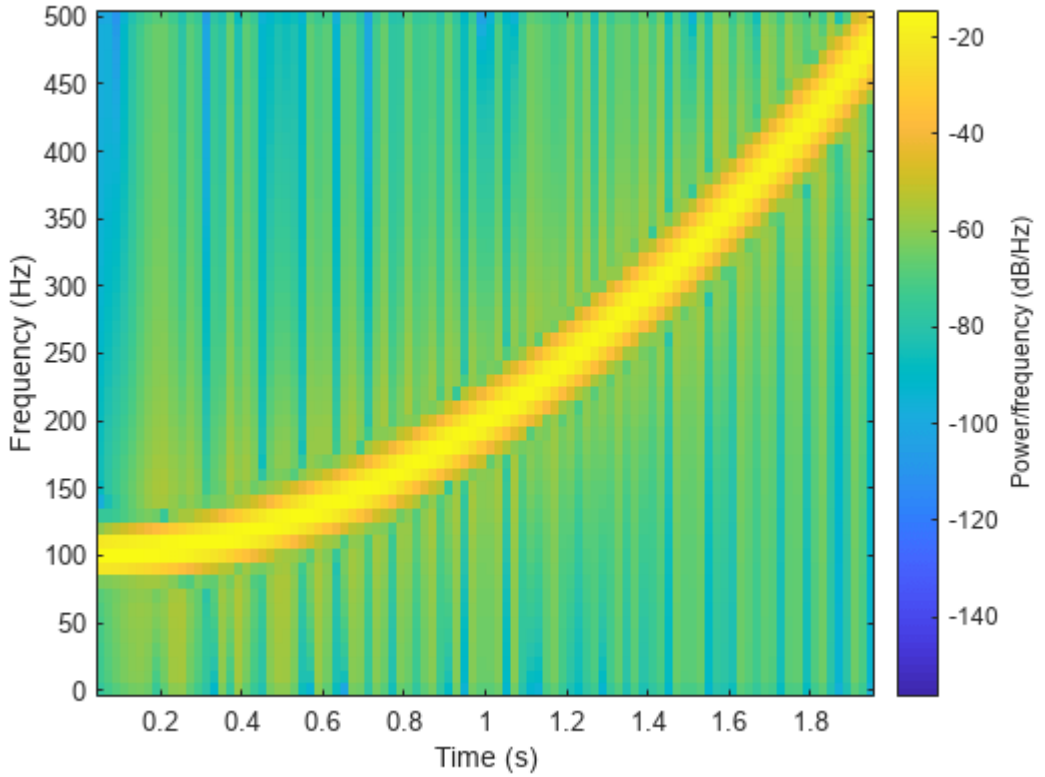
  step5 transform spectrogram to 0~255

Refer to demo_audio_processing.ipynb

Spectrogram

- A *spectrogram* is a visual representation of the spectrum of frequencies of a signal as it varies with time.





Source: https://pnsn.org/spectrograms/what-is-a-spectrogram

www.aiengineer.tw

Build a Neural Network and Practice Transfer Learning

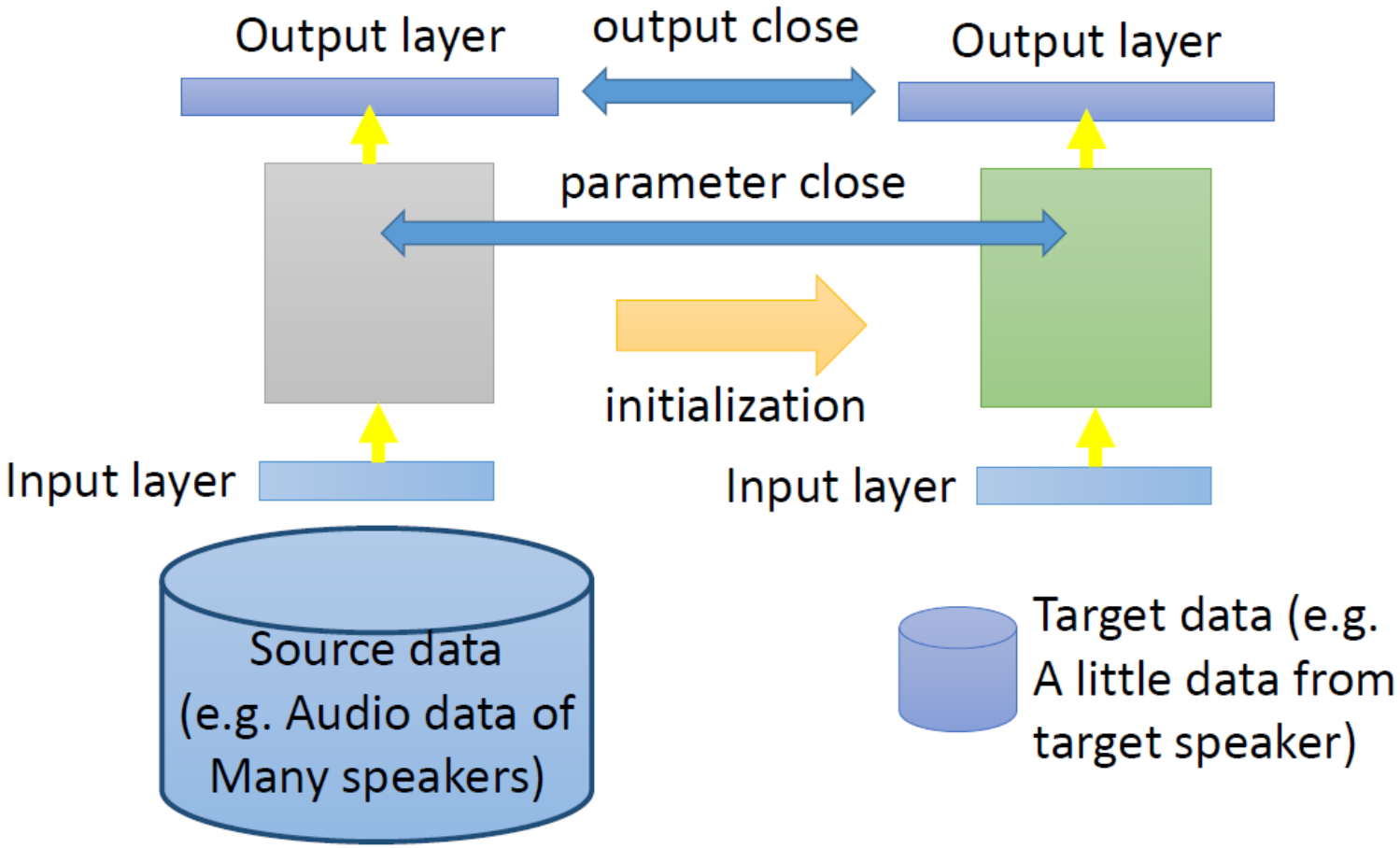
Build audio classification model

1. Load data
2. Custom dataloader
3. Declare model
4. Loss function
5. Design training process
6. Save model weight

Refer to demo_project.ipynb

Transfer learning

- **Large source data and fewer target data**



Summary

We have learned

1. basic audio processing skill
2. create the custom dataset
3. build a NN model
4. perform training & testing on Colab

What's the next?

It's your turn!