RVL 2022
Summer Training
Weak 5
Machine Learning

KUO,LI-CHIA

2022/08/10（Wed）13:00~16:00 at Room 125

Outline

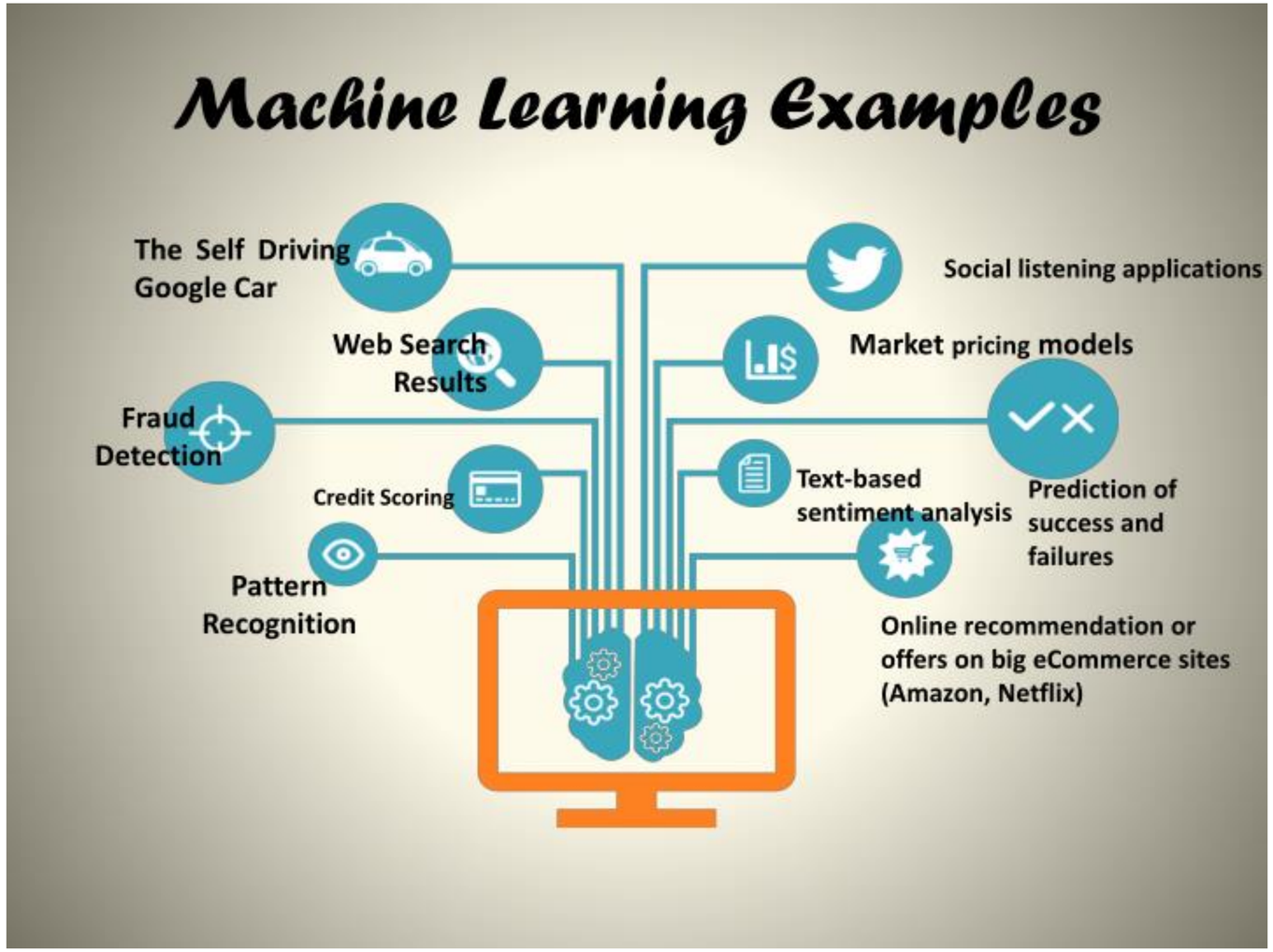
- What Is Machine Learning?
- NumPy
- Matplotlib
- Machine learning framework
- Machine Learning Example
- References

Outline

- What Is Machine Learning?
- NumPy
- Matplotlib
- Machine learning framework
- Machine Learning Example
- References

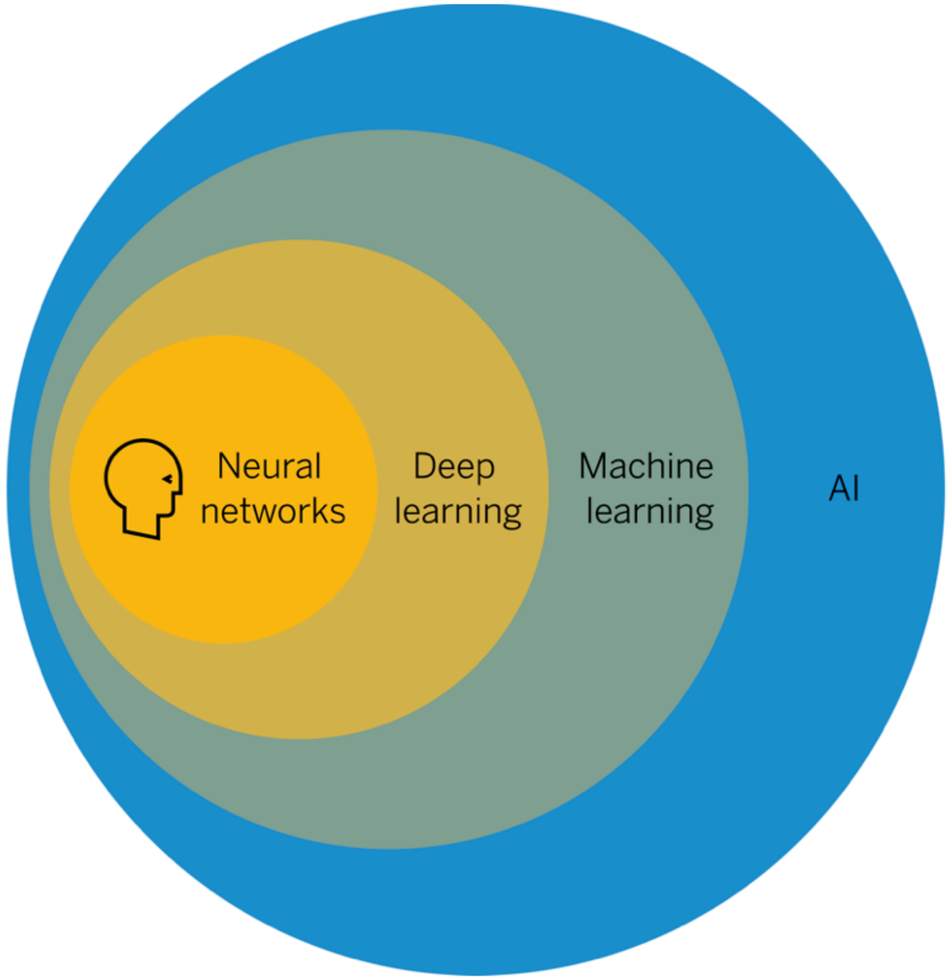
What Is Machine Learning?

Where can use machine learning?



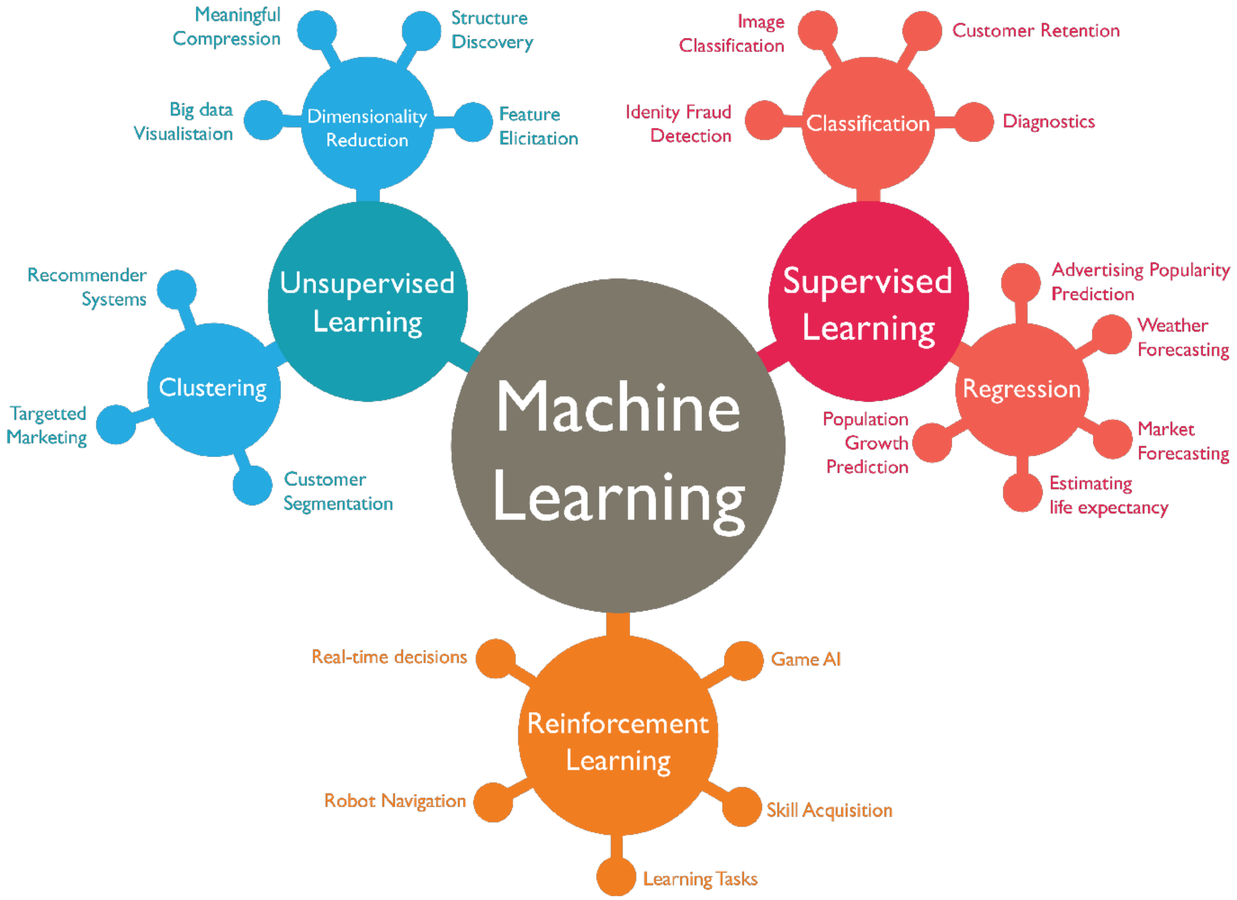
What Is Machine Learning?

AI and Machine Learning



What Is Machine Learning?

Machine Learning Type



Outline

- What Is Machine Learning?
- NumPy
- Matplotlib
- Machine learning framework
- Machine Learning Example
- References

NumPy

Feature

POWERFUL N-DIMENSIONAL ARRAYS

Fast and versatile, the NumPy vectorization, indexing, and broadcasting concepts are the de-facto standards of array computing today.

NUMERICAL COMPUTING TOOLS

NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more.

INTEROPERABLE

NumPy supports a wide range of hardware and computing platforms, and plays well with distributed, GPU, and sparse array libraries.

PERFORMANT

The core of NumPy is well-optimized C code. Enjoy the flexibility of Python with the speed of compiled code.

EASY TO USE

NumPy's high level syntax makes it accessible and productive for programmers from any background or experience level.

OPEN SOURCE

Distributed under a liberal BSD license, NumPy is developed and maintained publicly on GitHub by a vibrant, responsive, and diverse community.

NumPy

Getting Started

- Installation of NumPy:

```
pip install numpy
```

- Import NumPy:

```
import numpy as np
```

NumPy

Array

- Create a NumPy array:

```
arr = np.array([1, 2, 3 , 4, 5])
```

- Array Indexing:

```
print(arr[2]+arr[3])       #7
```

- Slicing arrays:

```
[start:end],[start:end:step]

print(arr[1:5:2])          #[2,4]
```

NumPy

Shape and Reshape

- Shape of an Array:

```
arr = np.array([1, 2, 3, 4], ndmin=5)        # [[[[[1  2  3  4]]]]]
print(arr.shape)          # (1, 1, 1, 1, 4)
```

- Reshape Array:

```
arr = np.array([1,  2,  3,  4,  5,  6,  7,  8,  9,  10,  11,  12])
newarr = arr.reshape(2,  3,  2)      #(2,  3,  2)
[[[ 1  2] [ 3  4]  [ 5  6]]  [[ 7  8] [ 9  10]  [11  12]]]
```

NumPy

Iterating

- Iterating Arrays:

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
for x in arr:
    print(x)     #[1  2  3]  [4  5  6]
```

NumPy

Joining and Splitting

- Joining Arrays:

```
arr = np.concatenate((arr1,  arr2))
```

- Splitting Arrays:

```
newarr = np.array_split(arr,  3)
```

NumPy

Searching and Sorting

- Searching Arrays:

```
x = np.where(arr ==  4)

x = np.where(arr%2  == 1)
```

- Sorting Arrays:

```
np.sort(arr)
```

NumPy

ufuncs

- + : np.add(x, y)
- - : np.subtract(arr1, arr2)
- * : np.multiply(arr1, arr2)
- / : np.divide(arr1, arr2)
- ** : np.power(arr1, arr2)
- % : np.remainder(arr1, arr2)
- divmod(): np.divmod(arr1, arr2)
- abs(): np.absolute(arr)

NumPy

NumPy ufuncs

```
import numpy as np

def myadd(x, y):
  return x+y

myadd = np.frompyfunc(myadd, 2, 1)

print(myadd([1, 2, 3, 4], [5, 6, 7, 8]))
```

NumPy

Difference Between Copy and View

```
arr = np.array([1, 2,  3,  4,  5])

x = arr

arr[0] = 42


print(arr)  # [42   2   3  4  5]

print(x)    # [42   2   3  4  5]
```

NumPy

Difference Between Copy and View

```
arr = np.array([1, 2, 3, 4, 5])
x = arr.copy()
arr[0] = 42

print(arr)  # [42  2  3  4  5]
print(x)    # [1  2  3  4  5]
```

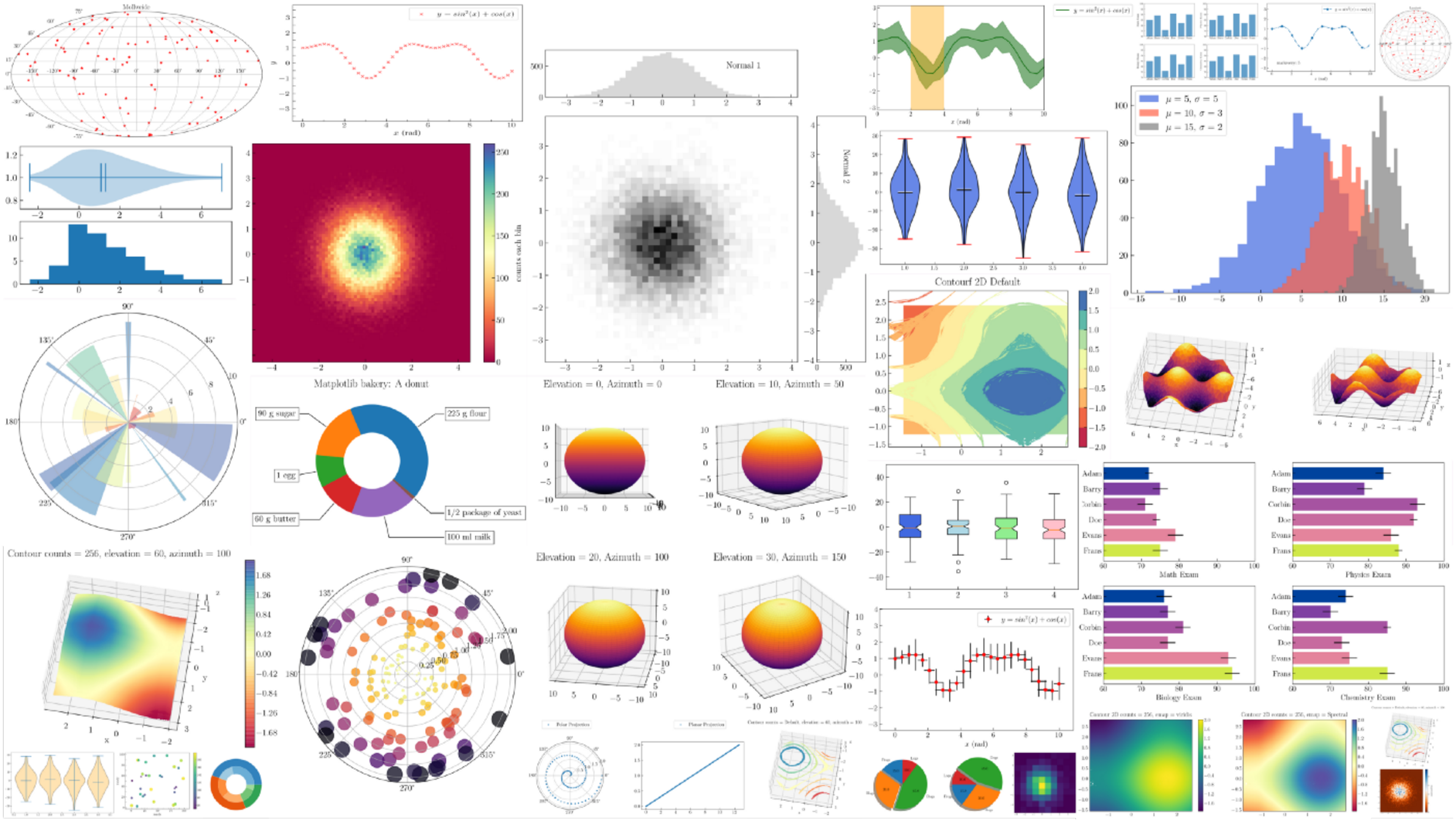
```
arr = np.array([1, 2, 3, 4, 5])
x = arr.view()
arr[0] = 42

print(arr)  # [42  2  3  4  5]
print(x)    # [42  2  3  4  5]
```

Outline

- What Is Machine Learning?
- NumPy
- Matplotlib
- Machine learning framework
- Machine Learning Example
- References

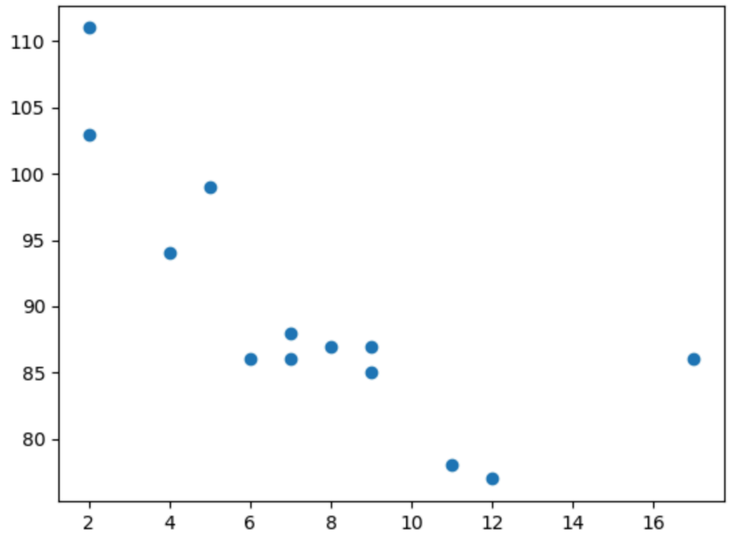
Matplotlib

What Is Matplotlib?

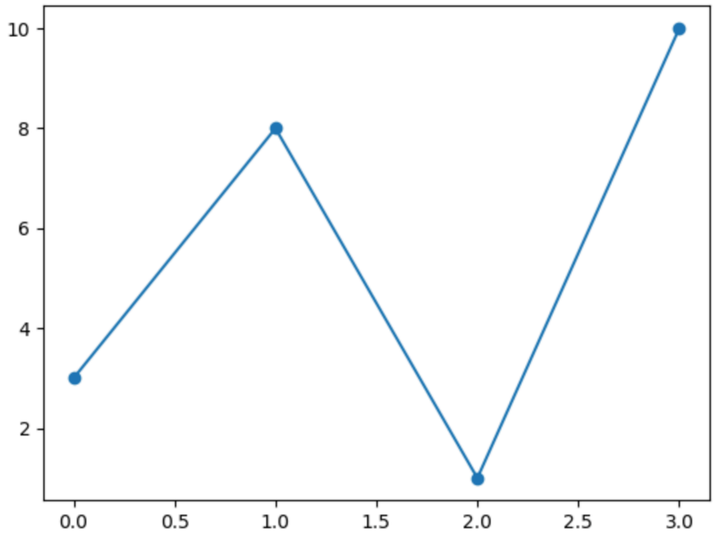


Matplotlib

Function



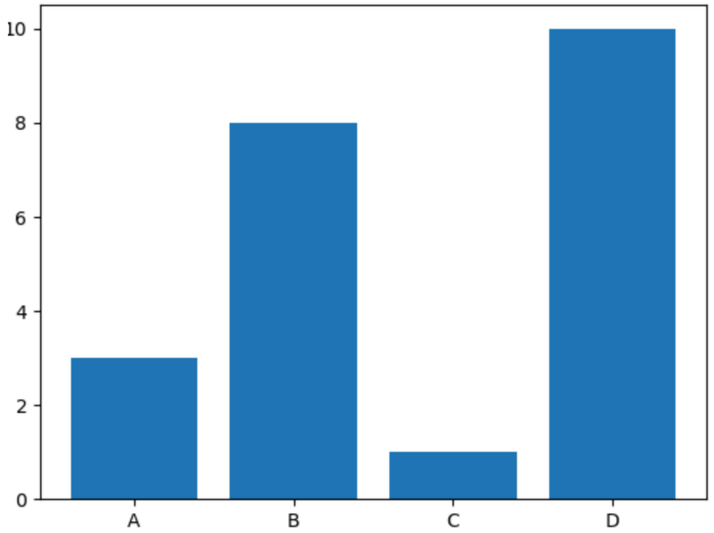

```
import matplotlib.pyplot as plt
import numpy as np

x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

plt.scatter(x, y)
plt.show()
```




```
import matplotlib.pyplot as plt
import numpy as np

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, marker = 'o')
plt.show()
```

Matplotlib

Function




```
import matplotlib.pyplot as plt
import numpy as np

x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.bar(x,y)
plt.show()
```

```
import matplotlib.pyplot as plt
import numpy as np

y = np.array([35, 25, 25, 15])

plt.pie(y)
plt.show()
```

Outline

- What Is Machine Learning?
- NumPy
- Matplotlib
- Machine learning framework
- Machine Learning Example
- References

Machine learning framework

Best Machine Learning Frameworks

- TensorFlow
- Shogun
- Sci-Kit Learn
- PyTorch
- CNTK

Machine learning framework

conda environment create

- create conda environment:

  conda create --name ENV_NAME python=3.7

  conda env create -f ENV_NAME.yml

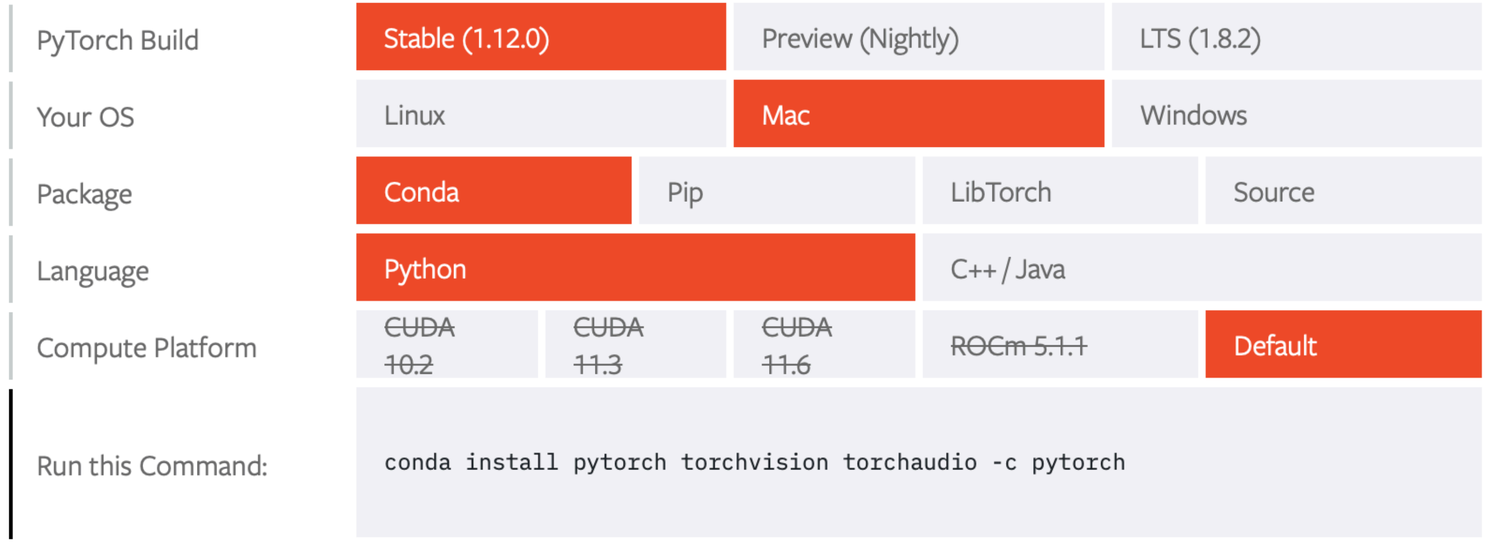
- Activate environment:

  Conda activate ENV_NAME

Machine learning framework

Install PyTorch

- Official website: https://pytorch.org/get-started/locally/



Machine learning framework

tensor

- Specialized data structure that are very similar to arrays and matrices.
- Use tensors to encode the inputs, outputs and model's parameters.
- Tensors are similar to NumPy's ndarrays, except that tensors can run on GPUs or other hardware accelerators.
- If you're familiar with ndarrays, you'll be right at home with the Tensor API.

Machine learning framework

Initializing a Tensor

- Directly from data:

```
data = [[1, 2],[3, 4]]
x_data = torch.tensor(data)
```

- From a NumPy array:

```
np_array = np.array(data)
x_np = torch.from_numpy(np_array)
```

Machine learning framework

Initializing a Tensor

- From another tensor:

```
x_ones = torch.ones_like(x_data)
x_rand = torch.rand_like(x_data, dtype=torch.float)
```

Machine learning framework

With random or constant values

- random:

```
shape = (2,3,)
rand_tensor = torch.rand(shape)
```

- constant:

```
ones_tensor = torch.ones(shape)
zeros_tensor = torch.zeros(shape)
```

Machine learning framework

Attributes of a Tensor

```
tensor = torch.rand(3, 4)


print(f"Shape of tensor: {tensor.shape}")
print(f"Datatype of tensor: {tensor.dtype}")
print(f"Device tensor is stored on: {tensor.device}")
```

Machine learning framework

Operations on Tensors

- move tensor to the GPU if available:

```
if torch.cuda.is_available():
    tensor = tensor.to(“cuda")
```

- indexing and slicing:

```
print(f"First row: {tensor[0]}")
print(f"First column: {tensor[:, 0]}")
print(f"Last column: {tensor[..., -1]}")
tensor[:,1] = 0
```

Machine learning framework

Operations on Tensors

- Joining tensors:

```
t1 = torch.cat([tensor, tensor, tensor], dim=1)
```

- Arithmetic operations:

```
torch.matmul(tensor, tensor.T, out=y3)

tensor.sum()

tensor.add_(5)
```

Machine learning framework

machine learning workflows

- Most machine learning workflows involve working with
    - working with data
    - creating models
    - optimizing model parameters
    - saving the trained models

Machine learning framework

Dataset & DataLoader

- torch.utils.data.Dataset:
    - stores the samples and their corresponding labels

- torch.utils.data.DataLoader:
    - wraps an iterable around the Dataset

Machine learning framework

Dataset & DataLoader: Creating a Custom Dataset for your files

```python
import os
import pandas as pd
from torchvision.io import read_image

class CustomImageDataset(Dataset):
    def __init__(self, annotations_file, img_dir, transform=None, target_transform=None):
        self.img_labels = pd.read_csv(annotations_file)
        self.img_dir = img_dir
        self.transform = transform
        self.target_transform = target_transform

    def __len__(self):
        return len(self.img_labels)

    def __getitem__(self, idx):
        img_path = os.path.join(self.img_dir, self.img_labels.iloc[idx, 0])
        image = read_image(img_path)
        label = self.img_labels.iloc[idx, 1]
        if self.transform:
            image = self.transform(image)
        if self.target_transform:
            label = self.target_transform(label)
        return image, label
```

- __init__:
  - run once when instantiating the Dataset object.
- __len__:
  - returns the number of samples in our dataset.
- __getitem__:
  - loads and returns a sample from the dataset at the given index idx.

Machine learning framework

Dataset & DataLoader: DataLoader

```
from torch.utils.data import DataLoader

train_dataloader = DataLoader(training_data, batch_size=64, shuffle=True)
```

Machine learning framework

TRANSFORMS

- transform: modify the features
- target_transform: modify the labels

```
ds = datasets.FashionMNIST(
    root="data",
    train=True,
    download=True,
    transform=ToTensor(),
    target_transform=Lambda(lambda y: torch.zeros(10, dtype=torch.float).scatter_(0,
torch.tensor(y), value=1))
)
```

- torchvision.transforms API:

  https://pytorch.org/vision/stable/transforms.html

Machine learning framework

BUILD THE NEURAL NETWORK

```
class NeuralNetwork(nn.Module):
    def __init__(self):
        super(NeuralNetwork, self).__init__()
        self.flatten = nn.Flatten()
        self.linear_relu_stack = nn.Sequential(
            nn.Linear(28*28, 512),
            nn.ReLU(),
            nn.Linear(512, 512),
            nn.ReLU(),
            nn.Linear(512, 10),
        )

    def forward(self, x):
        x = self.flatten(x)
        logits = self.linear_relu_stack(x)
        return logits
```

- Get Device for Training:

```
device = "cuda" if torch.cuda.is_available() else "cpu"
```

- create NeuralNetwork, move to the device:

```
model = NeuralNetwork().to(device)
```

- use the model:

```
X = torch.rand(1, 28, 28, device=device)
logits = model(X)
pred_probab = nn.Softmax(dim=1)(logits)
```

Machine learning framework

OPTIMIZING MODEL PARAMETERS

```python
def train_loop(dataloader, model, loss_fn, optimizer):
    size = len(dataloader.dataset)
    for batch, (X, y) in enumerate(dataloader):
        # Compute prediction and loss
        pred = model(X)
        loss = loss_fn(pred, y)

        # Backpropagation
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        if batch % 100 == 0:
            loss, current = loss.item(), batch * len(X)
            print(f"loss: {loss:>7f}  [{current:>5d}/{size:>5d}]")
```

```python
def test_loop(dataloader, model, loss_fn):
    size = len(dataloader.dataset)
    num_batches = len(dataloader)
    test_loss, correct = 0, 0

    with torch.no_grad():
        for X, y in dataloader:
            pred = model(X)
            test_loss += loss_fn(pred, y).item()
            correct += (pred.argmax(1) == y).type(torch.float).sum().item()

    test_loss /= num_batches
    correct /= size
    print(f"Test Error: \n Accuracy: {(100*correct):>0.1f}%, Avg loss: {test_loss:>8f} \n")
```

```python
loss_fn = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate)

epochs = 10
for t in range(epochs):
    print(f"Epoch {t+1}\n-------------------------------")
    train_loop(train_dataloader, model, loss_fn, optimizer)
    test_loop(test_dataloader, model, loss_fn)
```

Machine learning framework

SAVE AND LOAD THE MODEL

- Saving Model Weights:

```
torch.save(model.state_dict(), ‘model_weights.pth')
```

- Loading Model Weights:

```
model.load_state_dict(torch.load('model_weights.pth'))
model.eval()
```

Machine learning framework

Example

- https://colab.research.google.com/github/pytorch/tutorials/blob/gh-pages/_downloads/c30c1dcf2bc20119bcda7e734ce0eb42/quickstart_tutorial.ipynb

Outline

- What Is Machine Learning?
- NumPy
- Matplotlib
- Machine learning framework
- Machine Learning Example
- References

Machine Learning Example

class exercise 1

- quickstart_tutorial_original.py
- quickstart_tutorial.py

- Compare the differences.
- Let quickstart_tutorial.py can run.

Machine Learning Example

class exercise 2

- https://tbrain.trendmicro.com.tw/Competitions/Details/20

Outline

- What Is Machine Learning?
- NumPy
- Matplotlib
- Machine learning framework
- Machine Learning Example
- References

References

- NumPy
    - https://www.w3schools.com/python/numpy/default.asp
- Matplotlib
    - https://www.w3schools.com/python/matplotlib_intro.asp
- PyTorch
    - https://pytorch.org/get-started/locally/
    - https://pytorch.org/docs/1.12/
    - https://pytorch.org/tutorials/beginner/basics/intro.html