

PyTorch for Beginners

UW
DATA SCIENCE
CLUB.



Presented by Soyeon Jang

Resources Location

<https://bit.ly/3IMVEIw>



Workshop Outline

1. What is PyTorch?
2. Tensor Basics
3. Datasets
4. Transforms
5. DataLoaders
6. PyTorch Neural Network

✨ Goal ✨

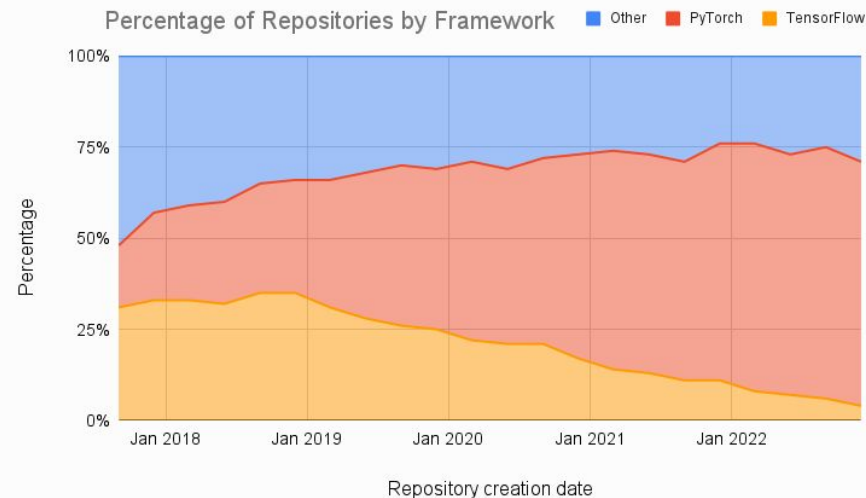
Be able to load datasets, perform data augmentation, train a model using PyTorch Neural Network module

What is PyTorch?

- Open-source library
- End-to-end Machine Learning framework
- Originally developed by Meta AI
- Part of the Linux Foundation umbrella



There are other frameworks like **Tensorflow, Apache MXNet, JAX**, etc



When to use **PyTorch**?

- If you feel more comfortable
- Favoured for research

Tensors

0D Tensor
Scalar

24

1D Tensor
Vector

$\begin{bmatrix} 2 & -8 & 7 \end{bmatrix}$

row

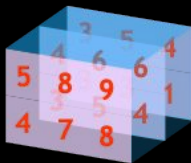
or
column $\begin{bmatrix} 2 \\ -8 \\ 7 \end{bmatrix}$

2D Tensor
Matrix

$\begin{bmatrix} 6 & 4 & 24 \\ 1 & -9 & 8 \end{bmatrix}$

row(s) × column(s)

3D Tensor



etc!

- Specialized data structure
- Similar to arrays and matrices
- Used to encode inputs, outputs and the model's parameters

Tensor vs NumPy Multidimensional arrays

- Similar except that tensors can run on GPUs or other hardware accelerators
- By default, tensors are created on the CPU and need to explicitly move to GPU

tensor.ipynb

<https://bit.ly/3IMVEIw>



Image

—

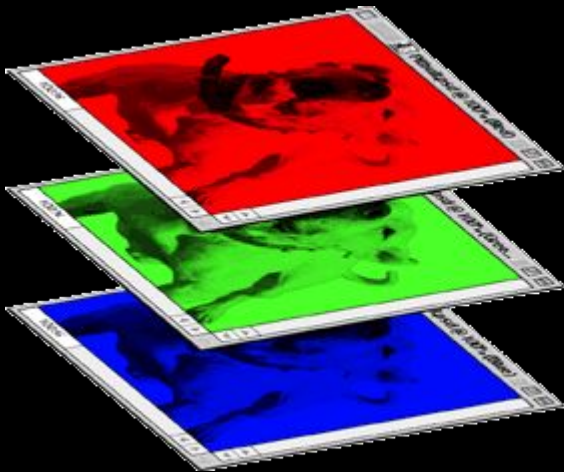
- **RGB image**
 - Red, green, blue are the primary colours of light
 - 3 channels
- **Gray image**
 - Only one channel

Pixels in the range of $[0, 255]$

0 = black

255 = white

$[C, H, W]$



Datasets and DataLoaders

Two data primitives:

- torch.utils.data.**DataLoader**
- torch.utils.data.**Dataset**

Dataset:

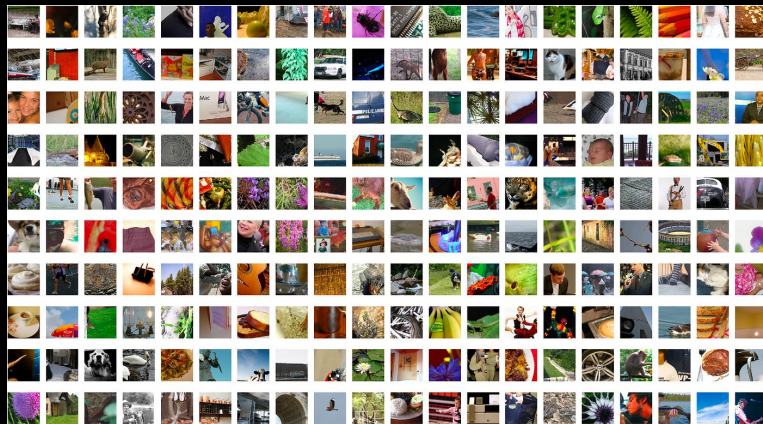
- Stores data and corresponding labels

DataLoader:

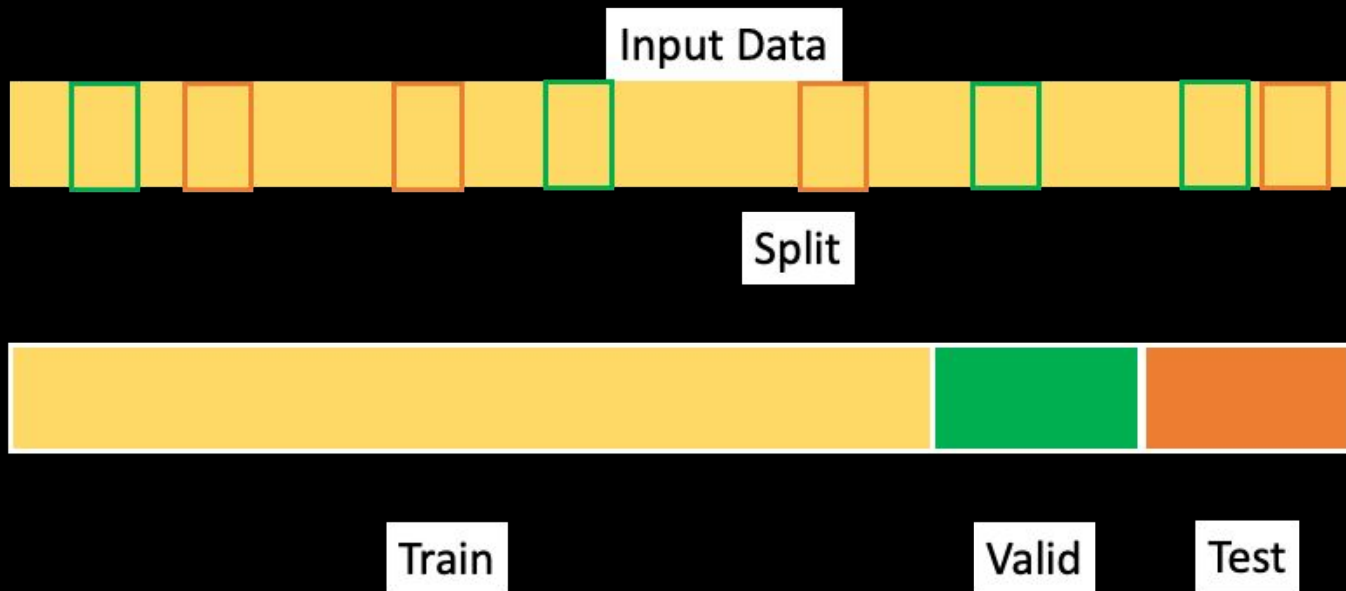
- Wraps an iterable around the Dataset to enable easy access to the data

Custom Dataset

- `__init__`
- `__len__`
- `__getitem__`



Train, Validation, and Test set



- 60-80% **Training** data
- 10-20% **validation** data
- 10-20% **test** data

Transforms

- Data augmentation for training

Techniques include:

- Randomly flip, crop, rotate
- Randomly change RGB colour channels, contrast and brightness
- Randomly change the sharpness or blur image



Why do we want to augment data?

- To expand the dataset
- Make it more robust to different variations in the data -> improve generalization, prevent models from overfitting

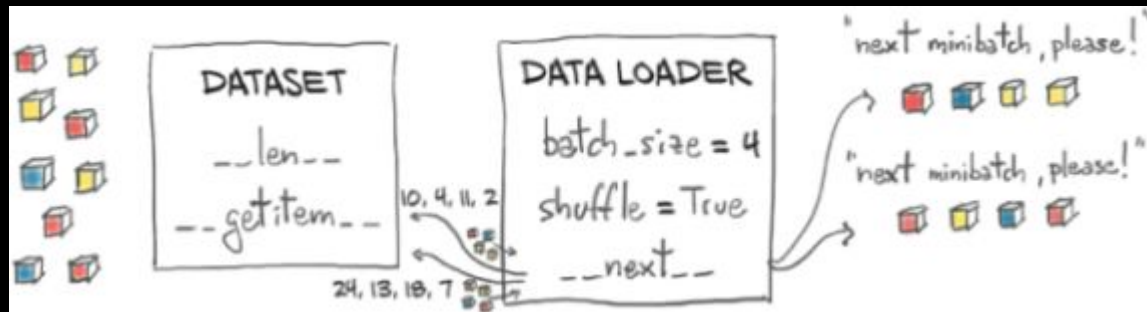
Need to be **cautious!**

- If not used carefully, it can introduce noise into the training data

Eg. applying multiple transformations on the same images

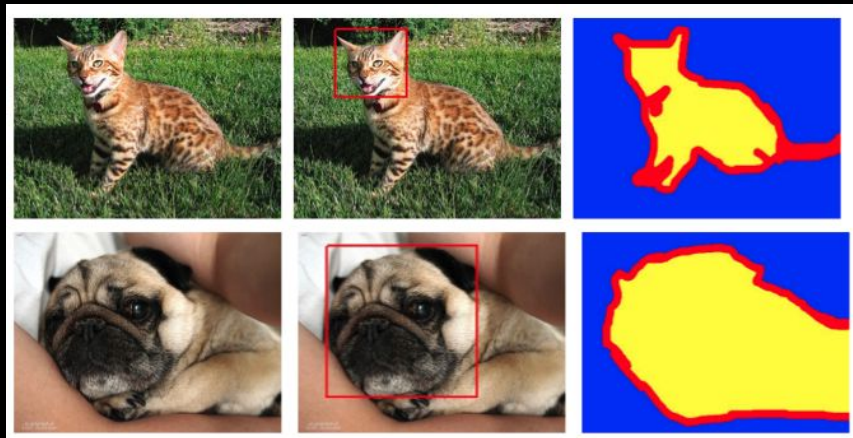
DataLoaders

- Iterable that combines a **dataset** and a **sampler**
- Shuffle = True -> if you have a lot of the same labels sequentially
 - Good to have batches to have examples of a mix of different labels
 - Data reshuffle at every epoch



Datasets

The Oxford-IIIT Pet Dataset



- Species and breed name
- A tight bounding box (ROI) around the head of the animal
- A pixel level foreground-background segmentation (Trimap)

dataset_dataloader.ipynb

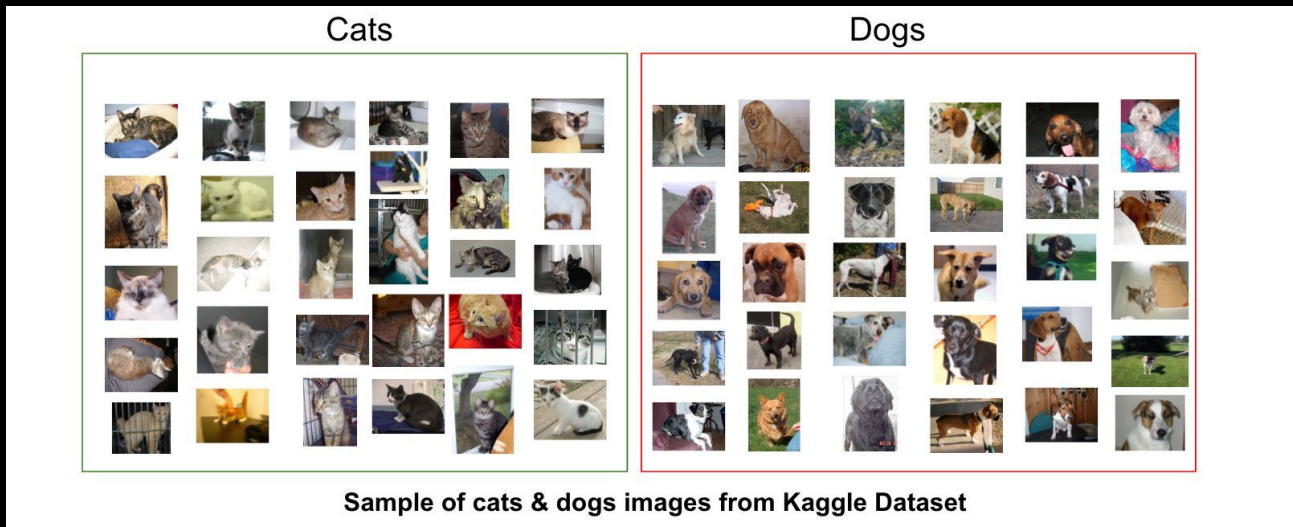
<https://bit.ly/3IMVEIw>



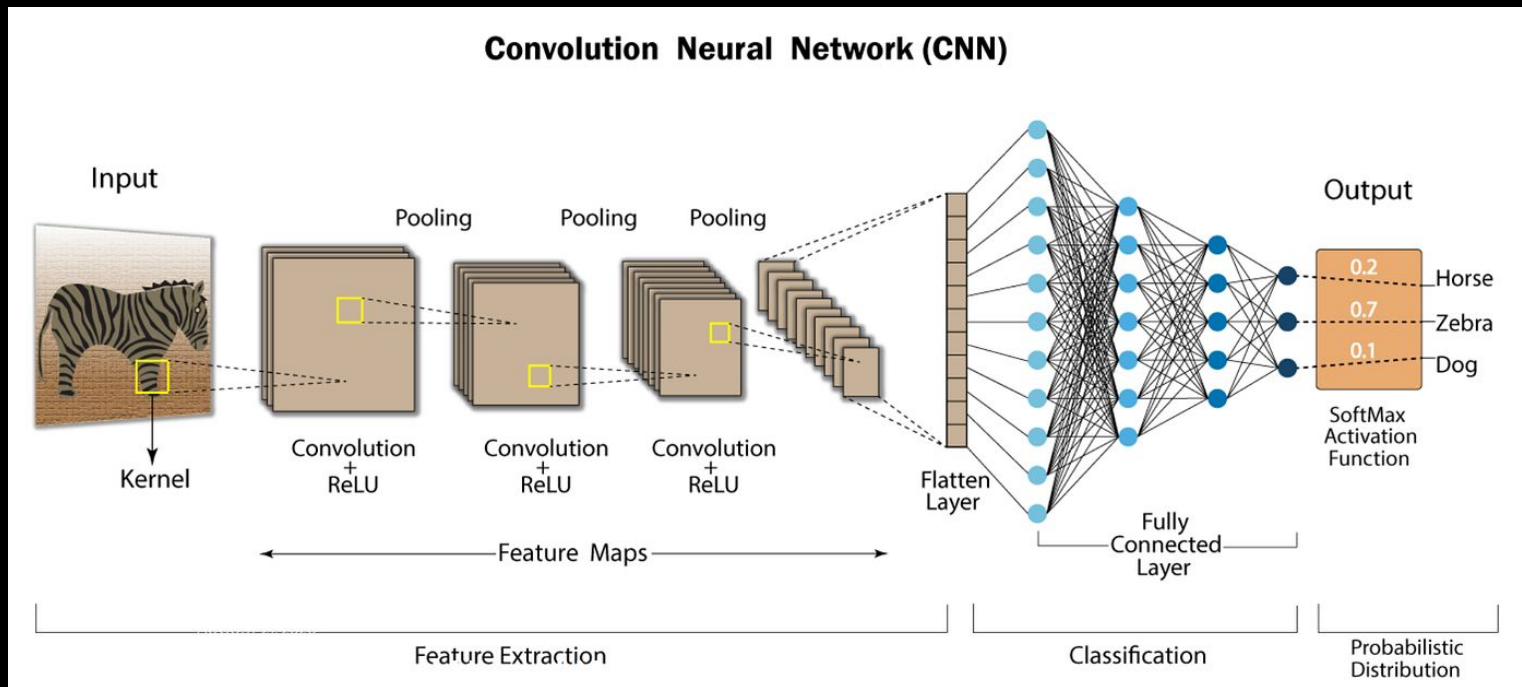
Classification Problem

“Sorting things into different groups based on their similarities”

Link to [Intro to Classification Workshop](#)

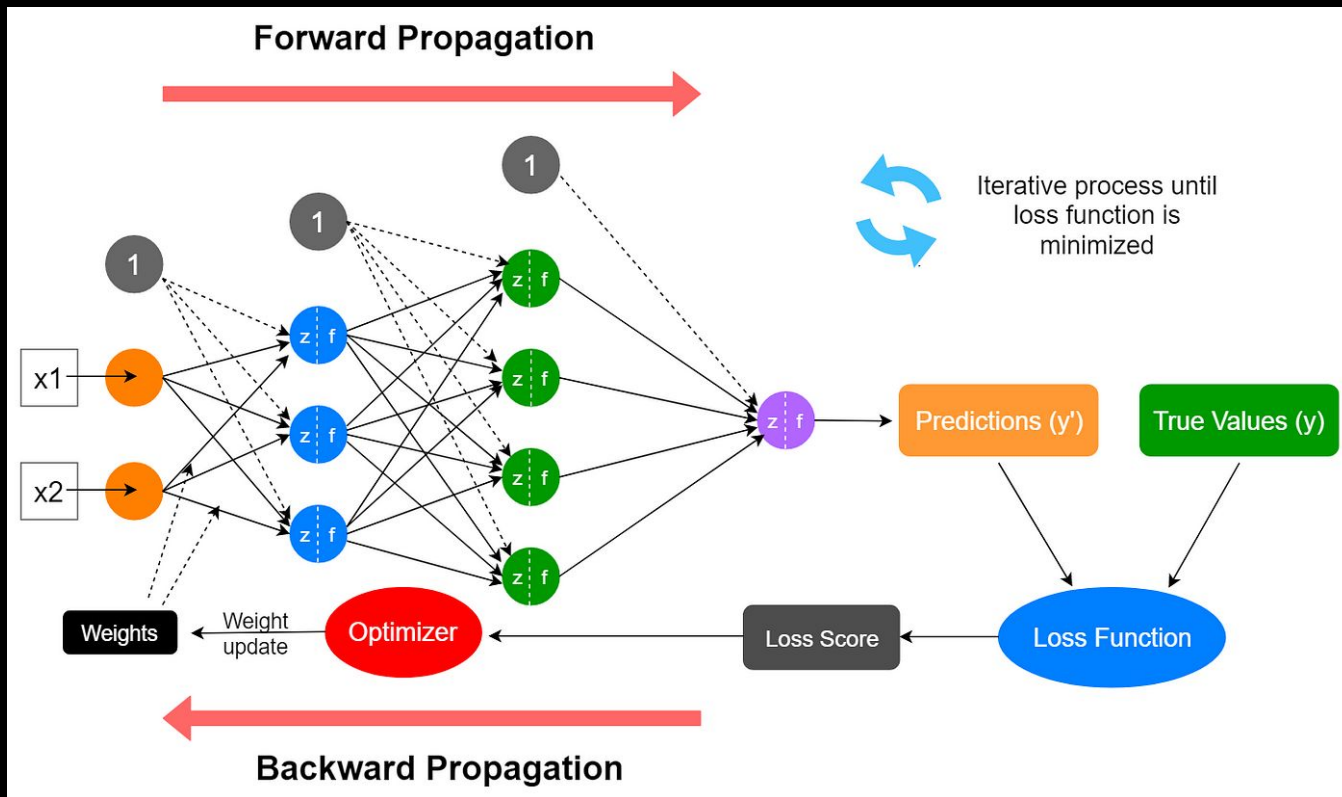


How?



Link to [Intro to Neural Networks & Computer Vision Workshop](#)

Forward Pass and Backpropagation



Loss function:

- Aim to minimize during training

Optimizer

- Adjust the parameters of the model to minimize the error between the predicted output and the actual output

PyTorch Neural Network

—

```
class NeuralNetwork(nn.Module):
    def __init__(self):
        super().__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
```

train_neural_network.ipynb

<https://bit.ly/3IMVEIw>

