



Kissipo Learning for Deep Learning

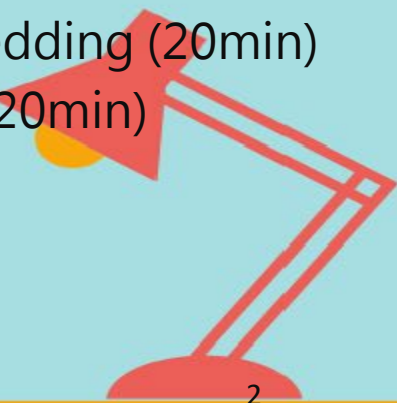
Topic 15: CNN with PyTorch (20min)

Hsueh-Ting Chu

KLDL-W6-T15

Topics

- Topic 01: Introduction to Deep Learning (20min)
- Topic 02: KISSIPO Learning for Deep Learning (20min)
- Topic 03: Python quick tutorial (20min)
- Topic 04: Numpy quick tutorial (15min)
- Topic 05: Pandas quick tutorial (15min)
- Topic 06: Scikit-learn quick tutorial (15min)
- Topic 07: OpenCV quick tutorial (15min)
- Topic 08: Image Processing basics (20min)
- Topic 09: Machine Learning basics (20min)
- Topic 10: Deep Learning basics (20min)
- Topic 11: TensorFlow overview (20min)
- Topic 12: CNN with TensorFlow (20min)
- Topic 13: RNN with TensorFlow (20min)
- Topic 14: PyTorch overview (20min)
- **Topic 15: CNN with PyTorch (20min)**
- Topic 16: RNN with Pytorch (20min)
- Topic 17: Introduction to AOI (20min)
- Topic 18: AOI simple Pipeline (A) (20min)
- Topic 19: AOI simple Pipeline (B) (20min)
- Topic 20: Introduction to Object detection (20min)
- Topic 21: YoloV5 Quick Tutorial (20min)
- Topic 22: Using YoloV5 for RSD (20min)
- Topic 23: Introduction to NLP (20min)
- Topic 24: Introduction to Word Embedding (20min)
- Topic 25: Name prediction project (20min)



Week 6 Topics

- Topic 14: PyTorch overview (20min)
- Topic 15: CNN with PyTorch (20min)
- Topic 16: RNN with Pytorch (20min)



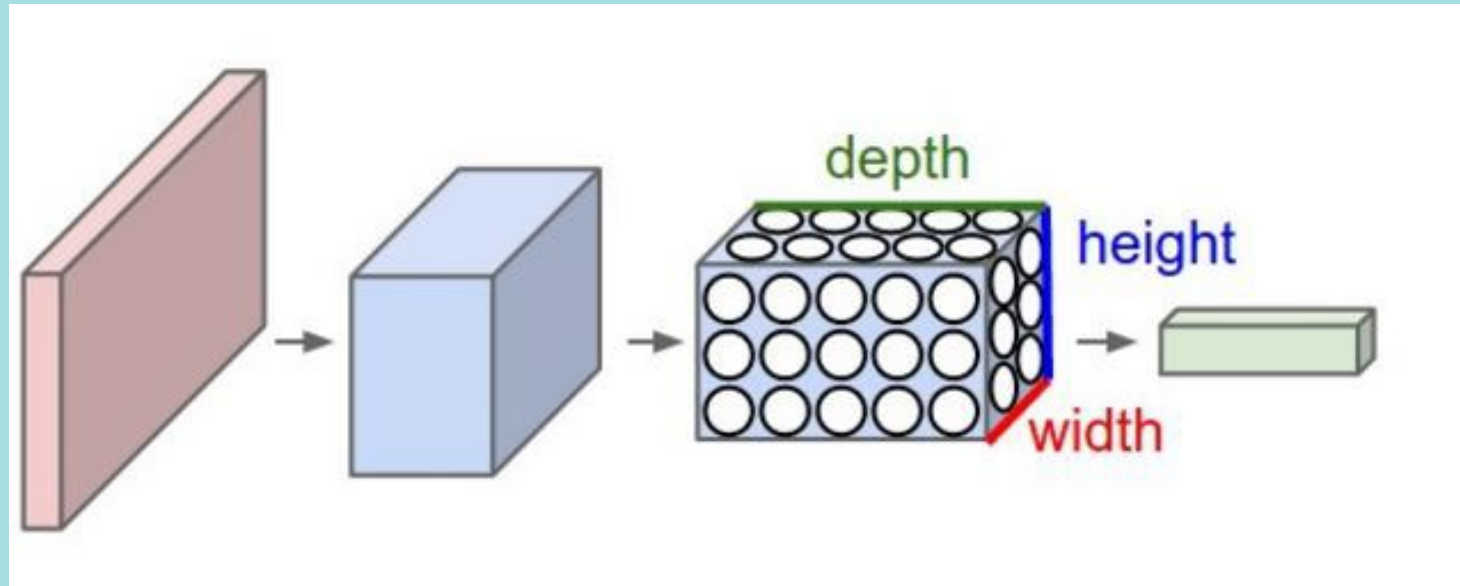
Topic 15 Content

- Topic 15: PyTorch CNN (20min)
 - Convolutional Neural Network using PyTorch
 - Training of PyTorch models
 - Dataset and DataLoader
 - Datasets.ImageFolder()
 - Transforms

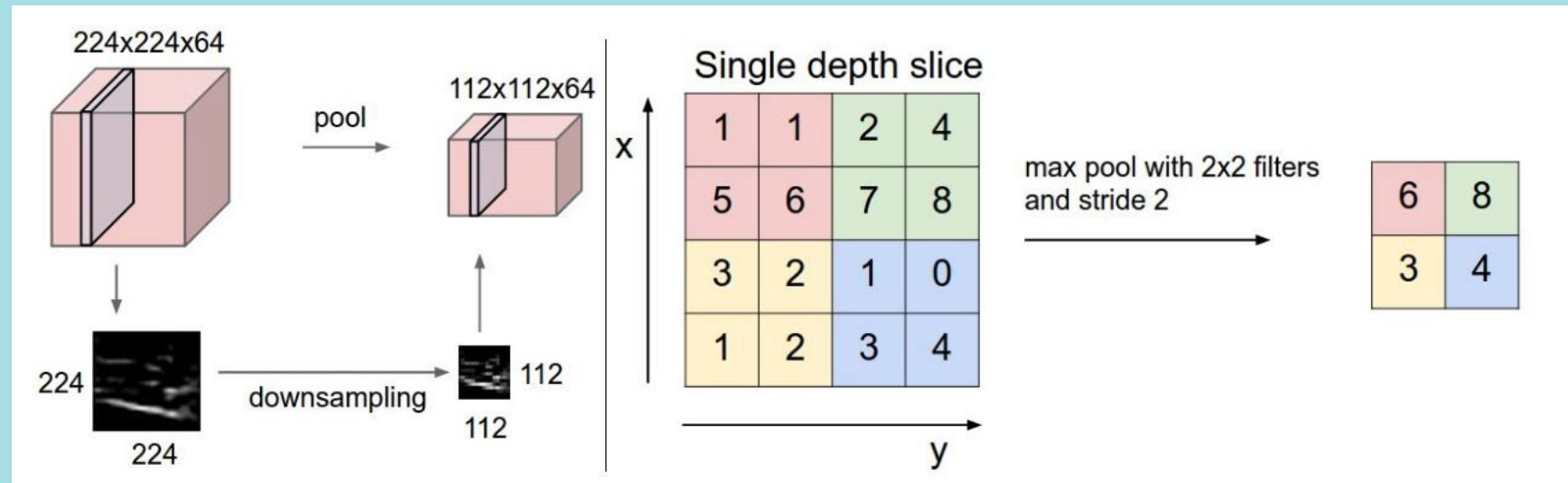


Convolution & Pooling layers

Convolution layer



Pooling layer



Convolutional neural network(CNN)

```
class Net(nn.Module):  
  
    def __init__(self):  
        super(Net, self).__init__()  
        # 1 input image channel, 6 output channels, 5x5 kernel  
        self.conv1 = nn.Conv2d(1, 6, 5)  
        self.conv2 = nn.Conv2d(6, 16, 5)  
        # an affine operation: y = Wx + b  
        self.fc1 = nn.Linear(16 * 5 * 5, 120)  
        self.fc2 = nn.Linear(120, 84)  
        self.fc3 = nn.Linear(84, 10)
```

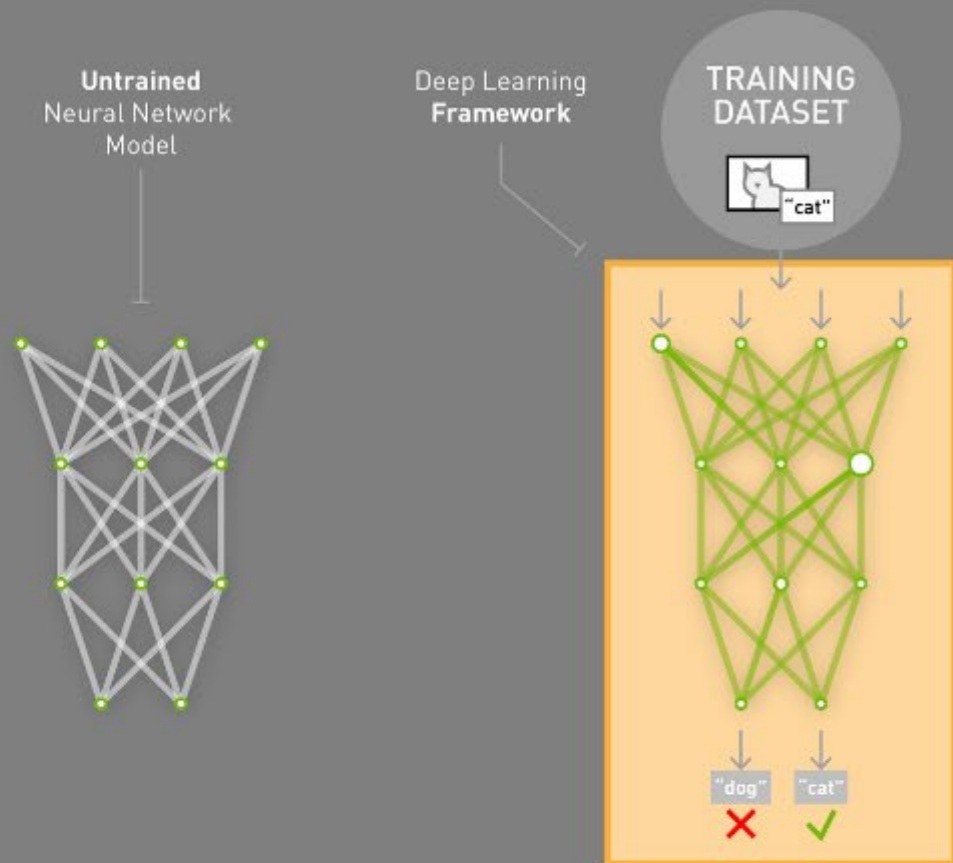
```
    def forward(self, x):  
        # Max pooling over a (2, 2) window  
        x = F.max_pool2d(F.relu(self.conv1(x)), (2, 2))  
        # If the size is a square, you can specify with  
        x = F.max_pool2d(F.relu(self.conv2(x)), 2)  
        x = torch.flatten(x, 1) # flatten all dimensions except batch  
        x = F.relu(self.fc1(x))  
        x = F.relu(self.fc2(x))  
        x = self.fc3(x)  
        return x
```



DEEP LEARNING

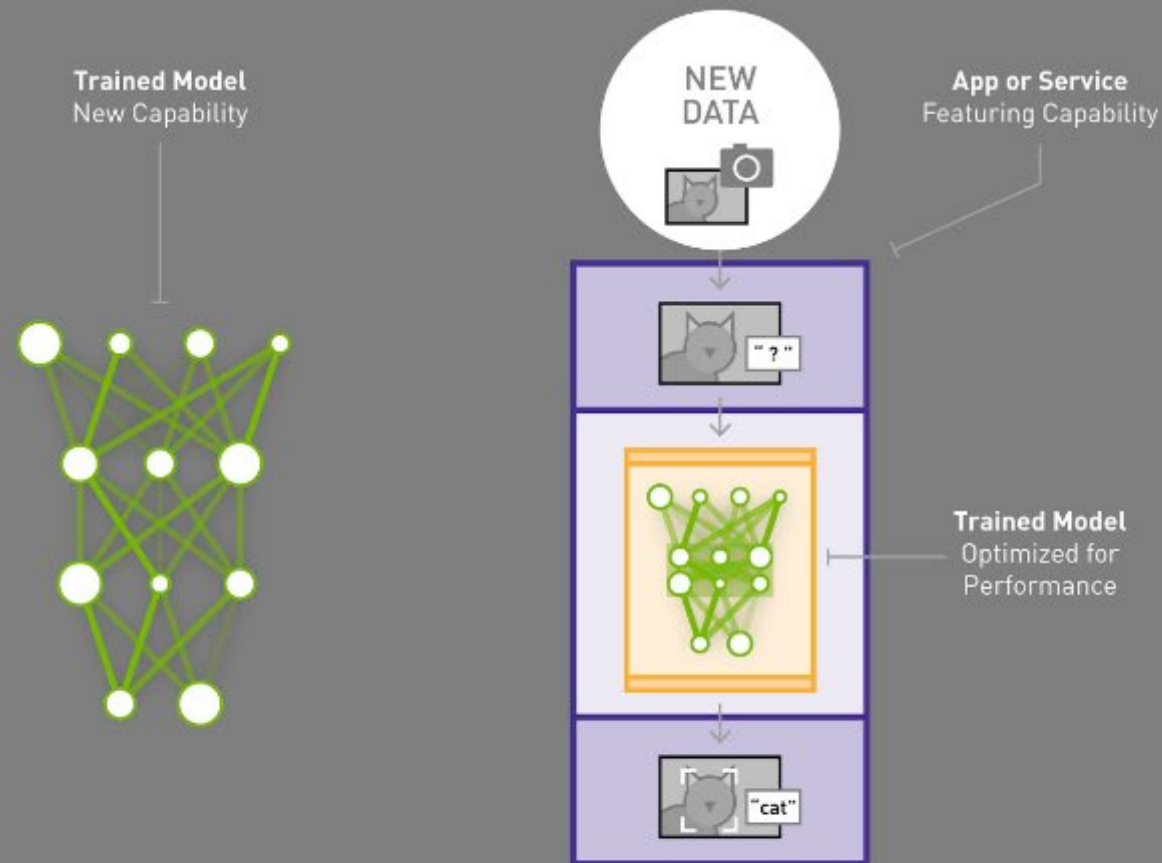
TRAINING

Learning a new capability
from existing data

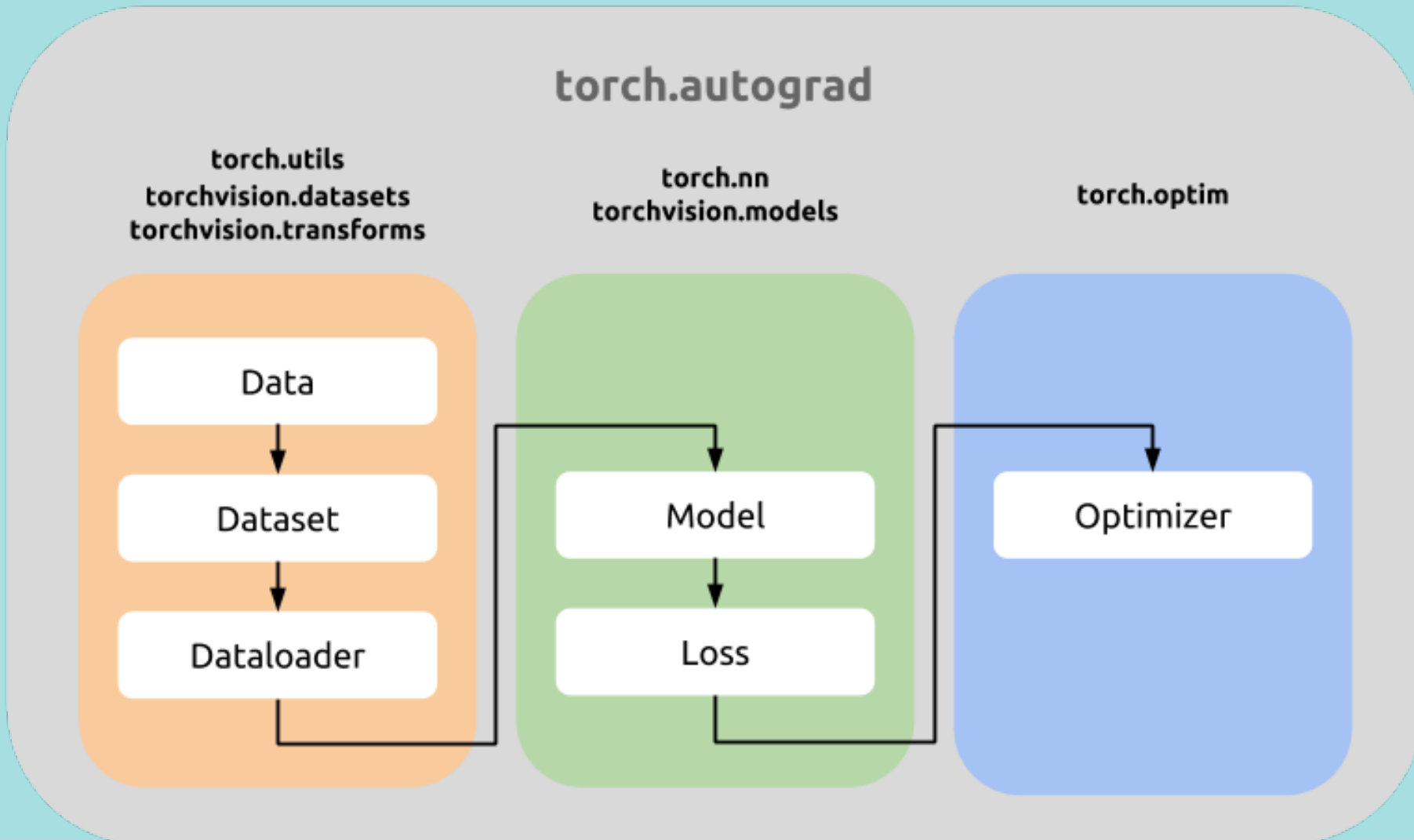


INFERENCE

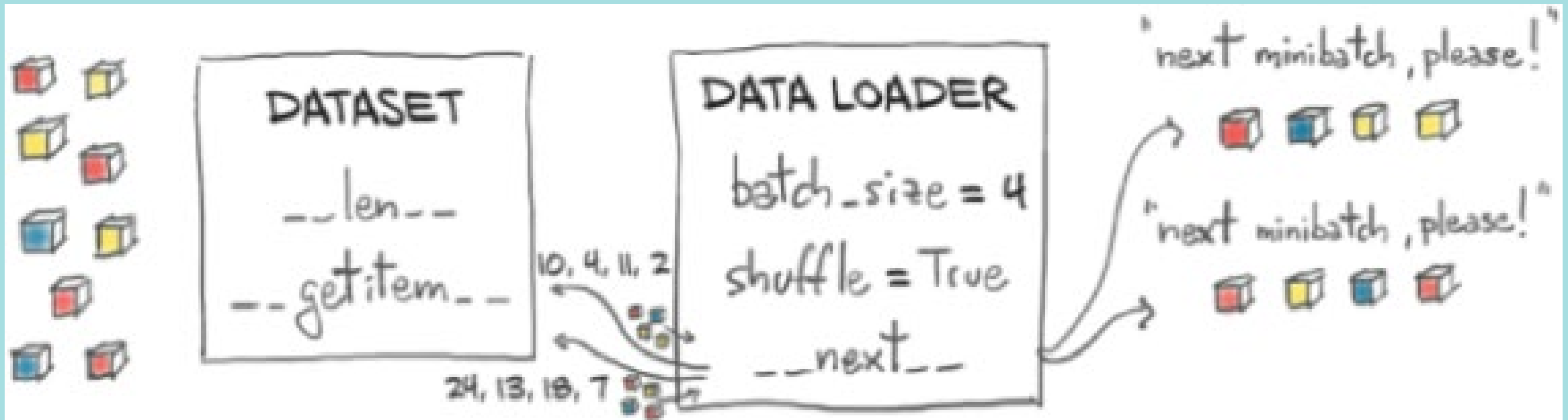
Applying this capability
to new data



Training of PyTorch models



Dataset and DataLoader



Datasets.ImageFolder()

```
CLASS torchvision.datasets.ImageFolder(root: str, transform:  
    ~typing.Optional[~typing.Callable] = None, target_transform:  
    ~typing.Optional[~typing.Callable] = None, loader: ~typing.Callable[[str],  
    ~typing.Any] = <function default_loader>, is_valid_file:  
    ~typing.Optional[~typing.Callable[[str], bool]] = None) \[SOURCE\]
```

A generic data loader where the images are arranged in this way by default:

```
root/dog/xxx.png  
root/dog/xyx.png  
root/dog/.../xxz.png  
  
root/cat/123.png  
root/cat/nsdf3.png  
root/cat/.../asd932_.png
```

This class inherits from [DatasetFolder](#) so the same methods can be overridden to customize the dataset.

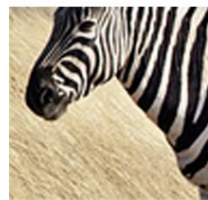
Parameters:

- **root** (*string*) – Root directory path.
- **transform** (*callable, optional*) – A function/transform that takes in an PIL image and returns a transformed version. E.g, `transforms.RandomCrop`
- **target_transform** (*callable, optional*) – A function/transform that takes in the target and transforms it.
- **loader** (*callable, optional*) – A function to load an image given its path.
- **is_valid_file** – A function that takes path of an Image file and check if the file is a valid file (used to check of corrupt files)



Transforms

RANDOM RESIZECROP



RANDOM HORIZONTAL FLIP



COLOR JITTER



pythoslabs.com



Thanks!

Q&A

