Homework 4: due 2022/05/09 23:59 (100%)

- Tutorial :

1. Training on MNIST: https://www.kaggle.com/code/juiyangchang/cnn-with-pytorch-0-995-accuracy (https://www.kaggle.com/code/juiyangchang/cnn-with-pytorch-0-995-accuracy)
2. Torchvision transforms: https://pytorch.org/vision/stable/transforms.html (https://pytorch.org/vision/stable/transforms.html)
3. Pytorch leraning rate scheduler: https://pytorch.org/docs/stable/optim.html (https://pytorch.org/docs/stable/optim.html)
4. How to get learning rate: https://stackoverflow.com/questions/52660985/pytorch-how-to-get-learning-rate-during-training (https://stackoverflow.com/questions/52660985/pytorch-how-to-get-learning-rate-during-training)

- After you go through the tutorials, you should be able to work on this assignment.

- Please answer the following questions and work directly on this jupyter notebook.

- Make sure the code can be run and show the result and figures properly.

- Please write down your observation with markdown in this notebook briefly.

You will train a multi-class classfication model in this part. The data contains the images with three categories: cats, dogs and pandas. You can find the details of each column at https://www.kaggle.com/datasets/ashishsaxena2209/animal-image-datasetdog-cat-and-panda (https://www.kaggle.com/datasets/ashishsaxena2209/animal-image-datasetdog-cat-and-panda).

In [ ]:

```
# Import necessory modules
%matplotlib inline
import os
import glob
import numpy as np
import random
import torch
import torch.nn as nn
import torch.optim as optim
import matplotlib.pyplot as plt
import torchvision.transforms as tr
import torchvision.models as models
from torch.optim.lr_scheduler import MultiStepLR
import PIL.Image as Image
from torch.utils.data import Dataset, DataLoader
from tqdm.notebook import tqdm
```

In [ ]:

```
# For reproduce the result
torch.manual_seed(0)
random.seed(0)
np.random.seed(0)
```

1. Define the model and dataset (30%)

1.1 Please create a class AnimalDataset for loading the data, and a varibale *transform* for the preprocessing transformation should be created for later usage. (15%)

In [ ]:

```
class AnimalDataset():
    def __init__():


    def __len__():


    def __getitem__():
```

1.2 Please create a class CNN as your network with the architecture below. (15%)

(Note. Please determine the output layer by the task)

Block 1	Block 2	Fully Connected Layer
3x3 conv, 64	3x3 conv, 128	Linear, 512
ReLU	ReLU	Dropout(p=0.5)
2x2 MaxPooling, downsampling factor 2	2x2 MaxPooling, downsampling factor 2	ReLU
		Linear, 256
		Dropout(p=0.5)
		ReLU

In [ ]:

```
class CNN():
    def __init__():


    def forward(self, X):
```

2. Train the model (70%)

2.1 Please load the train/validation/test data from *./animals* respectively and resize the image to 32x32. For the data preprocessing, please apply *RandomHorizontalFlip(p=0.5)* and *RandomRotation((-10,10))* as data augmentations. Also, scale all the value in the range between 0 and 1 and normalize with mean value *(0.485,0.456,0.406)* and standard deviation *(0.229,0.224,0.225)* for RGB channel respectively. At last, create the DataLoaders with batch size 32.(10%)

(Note. The mean and standard deviation is calculated from ImageNet dataset)

In [ ]:

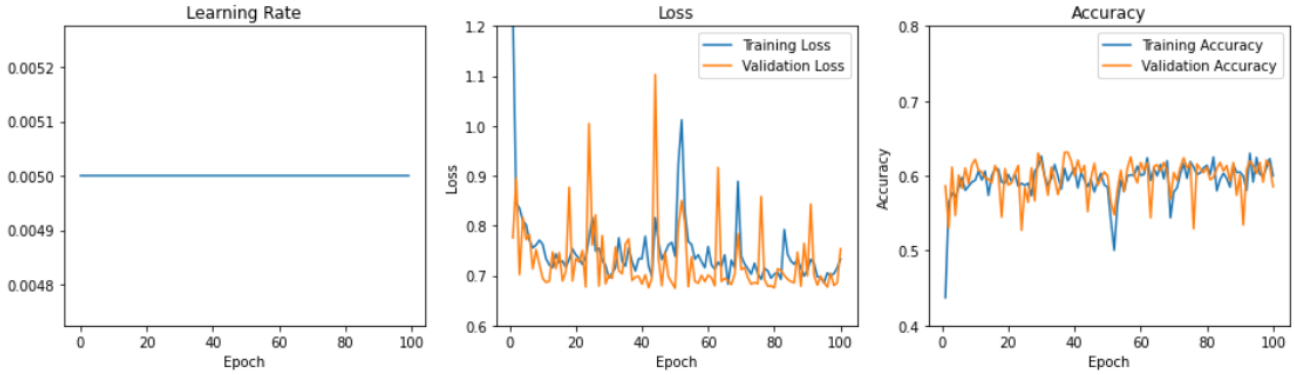
2.2 Train the CNN model with the same hyperpameters below and do the validation every epoch. Choose the appropriate type of loss according to the task. Please record the learning rate, training/validation loss and training/validation accuracy every epoch. Also, save the model weights as *model_without_scheduler.pth*(20%)

	Learning rate	epochs	optimizer	weight decay	$\beta _1$	$\beta _2$
Hyperparameter	5e-3	100	Adam	1e-2	0.9	0.99

In [ ]:

2.3 Please draw the plot the learning rate, training/validation loss and training/validation accuracy and write down the observation. (10%)

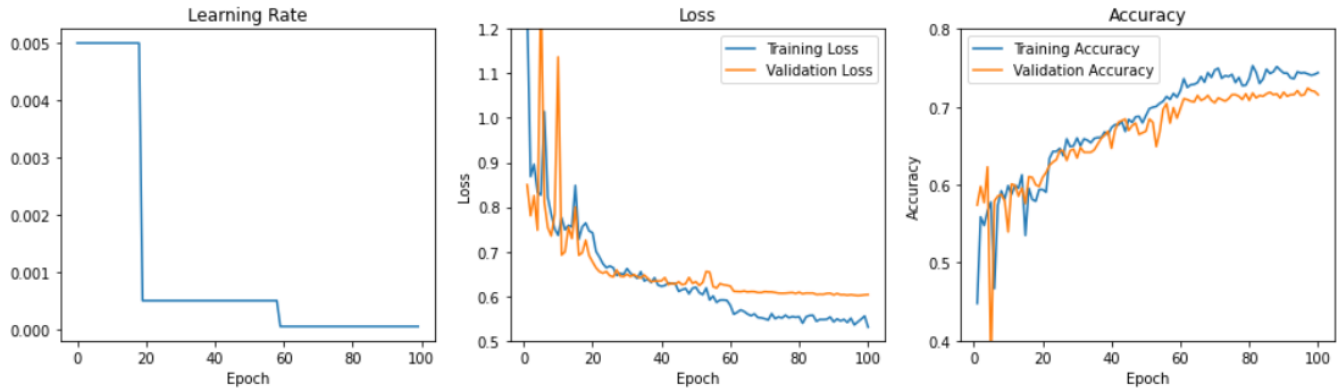
(Example figure)



In [ ]:

2.4 Please retrain model with learning rate decay with decreasing factor 0.1 at 20 epoch and 60 epoch. The other parameters are as same as last question. Also, redraw the learning rate, accuracy and loss curves and save the weights as *model_with_scheduler.pth*. (10%)

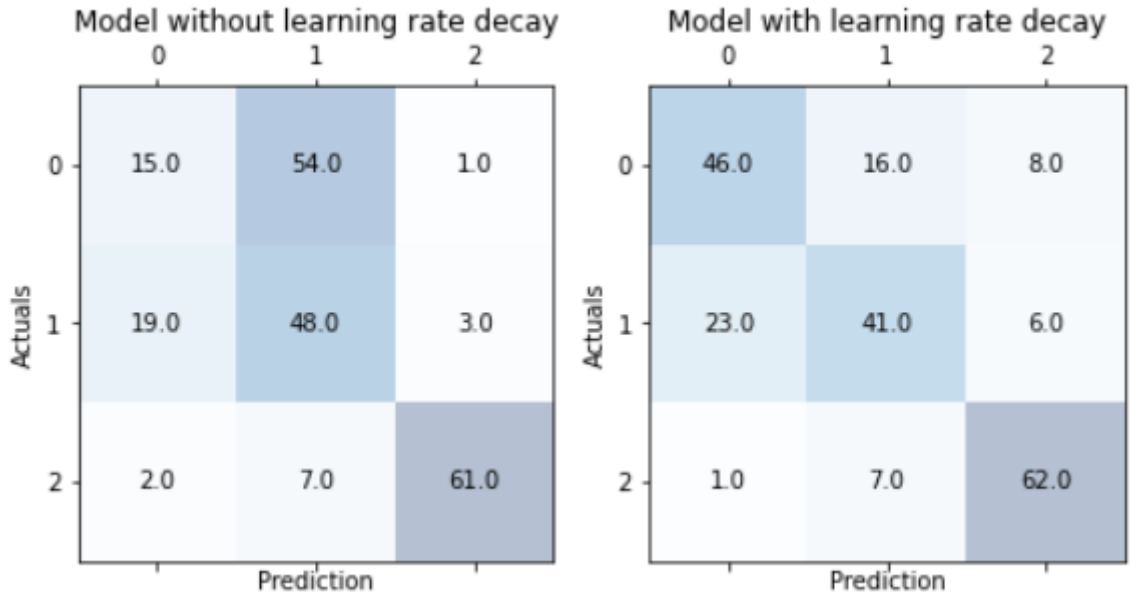
(Example figure)



In [ ]:

2.5 Please calculate the confusion matrix and print the accuracy of two models with the test dataset. (10%)

(Example figure)



In [ ]:

2.6 Please choose the best model to predict the categories of images in the *./animals/predict* folder and show the figure with the prediction as title of each axes. (10%)

(Example figure)

In [ ]: