Homework 3
PyTorch Training

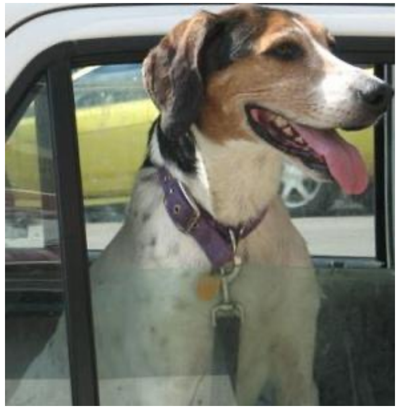
11/24

Introduction

- Goal: Build your own model, prepare the dataset and train the model on the classification dataset
- Please finish the “?” parts and the areas surrounding by comment blocks in the given jupyter-notebook file
- There are three major parts of this homework
    - Preparing data
    - Building a model
    - Training the model

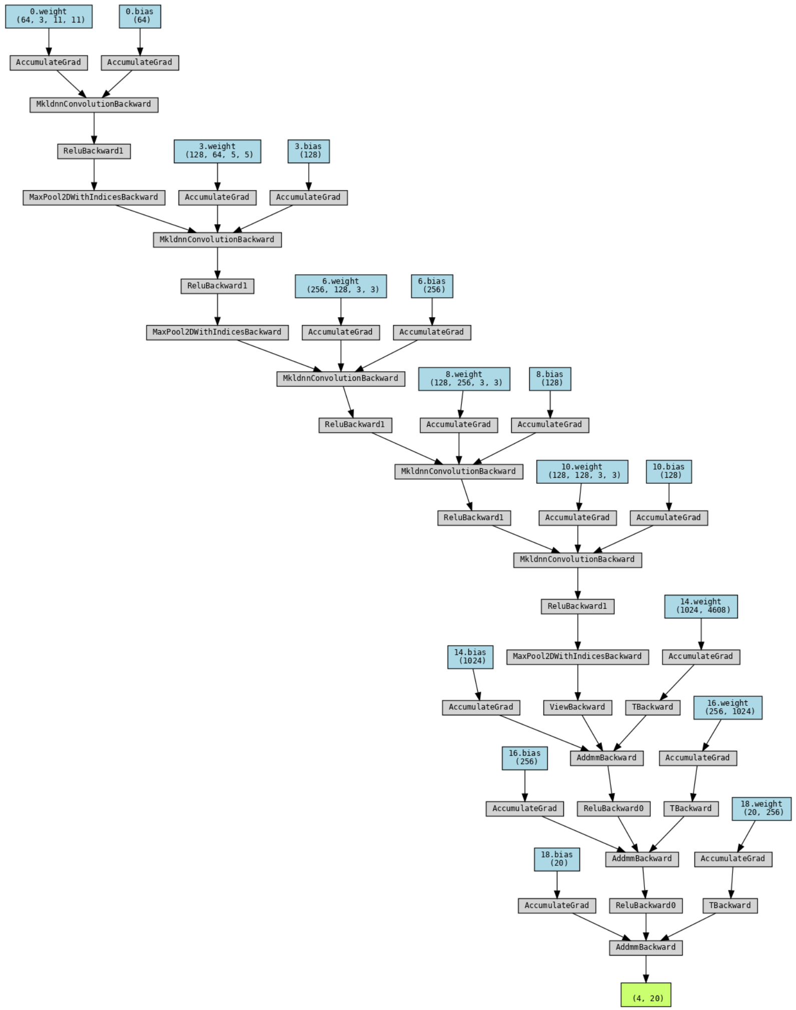
Preparing Data – 30%

- Transformation
  - Nothing to do here but you can add or modify the combinations as you want
- Custom dataset definition – 10%
  - We are using a subset of Stanford Dogs that has 20 classes
- Create datasets – 10%
- Create dataloaders – 10%



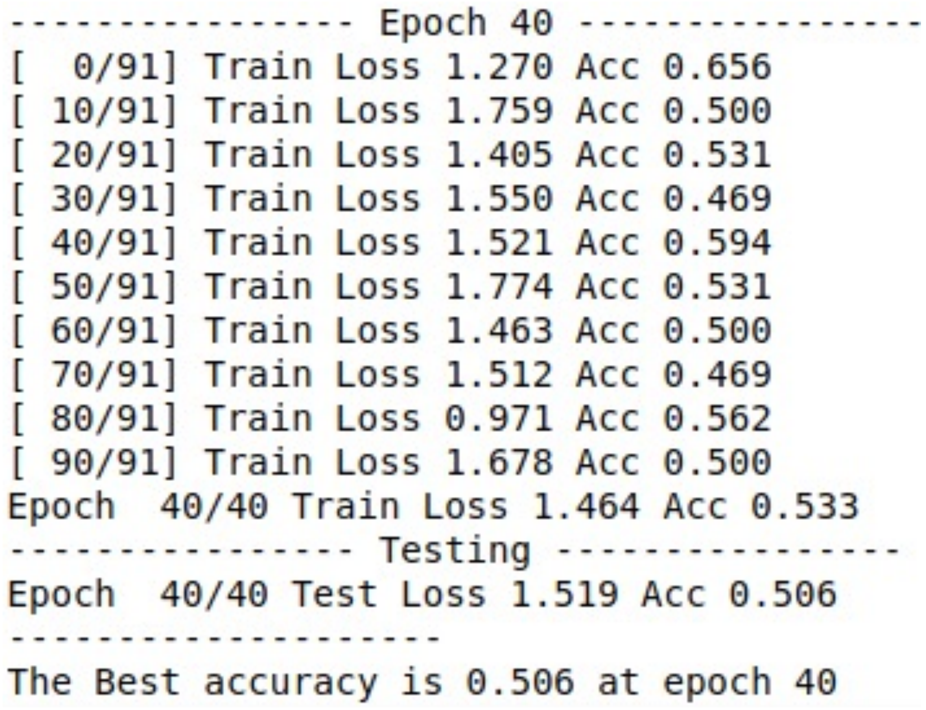
Building Models – 35%

- Flatten function – 5%
- Build a model using
  - nn.Module – 10%
  - nn.Sequential – 10%
  - You can compare the generated figures of the models with the provided images
- Design your own model – 10%
  - You cannot use models from torchvision.models
  - You cannot use pretrained weights
  - Explain your design



Training Models – 35%

- Loss function & Optimizer – 5%
- Training – 20%
  - Complete the code – 10%
  - Best testing accuracy (within the last 5 epochs)
    - Reaches the baseline of 50% – 5%
    - Bonus – 5%
- Saving model - 5%
- Loss and accuracy curves - 5%



```
---------------- Epoch 40 ----------------
[  0/91] Train Loss 1.270 Acc 0.656
[ 10/91] Train Loss 1.759 Acc 0.500
[ 20/91] Train Loss 1.405 Acc 0.531
[ 30/91] Train Loss 1.550 Acc 0.469
[ 40/91] Train Loss 1.521 Acc 0.594
[ 50/91] Train Loss 1.774 Acc 0.531
[ 60/91] Train Loss 1.463 Acc 0.500
[ 70/91] Train Loss 1.512 Acc 0.469
[ 80/91] Train Loss 0.971 Acc 0.562
[ 90/91] Train Loss 1.678 Acc 0.500
Epoch  40/40 Train Loss 1.464 Acc 0.533
---------------- Testing ----------------
Epoch  40/40 Test Loss 1.519 Acc 0.506
--------------------
The Best accuracy is 0.506 at epoch 40
```

Submission

- Deadline: 12/8 23:59
- Submit only the .ipynb file to moodle
  - You have to show all the execution results on the notebook
  - Do not upload the trained model
  - Filename: StudentID_name_hw2.ipynb
    - Ex: P76543210_王大明_hw2.ipynb
    - 10 points deduction for wrong names
- If you have problems, you can send mails to ne6094041@gs.ncku.edu.tw or come to 65601, CSIE
  - TA hour: Mon. and Thurs. 14:00 ~ 16:30

Common Problems

- Python environments
  - torch==1.7.1
  - torchvision==0.8.2
  - scikit-learn
  - numpy
  - jupyter
  - torchviz
  - Matplotlib
- You can simply install them by the requirements.txt using "$ pip install -r requirements.txt"

Common Problems

- You may need to install ”graphviz” to utilize ”torchviz” package
  - If “pip install graphviz” does not work, try
    - “$ sudo apt-get install graphviz” for linux-like OS
    - “$ brew install graphviz” for macOS
    - Download and install from https://graphviz.org/download/ for Windows

Common Problems

- Cuda OOM (Out of memory)
  - Reasons
    - Too much data in your GPU memory
    - The linear layers are too large (a nn.Linear(m, n) needs $\mathrm{O(mn)}$ memory)
  - Solutions
    - Decrease your batch size until the data fits your GPU
    - Free tensors or variables you don’t need any more
    - Decrease the size of linear layers

Ref
https://pytorch.org/docs/stable/notes/faq.html

Common Problems

- Runtime Error: size mismatch, m1: [a x b], m2: [c x d] or similar problems
  - Reasons
    - Size of dimensions mismatch while doing the matrix multiplication, ex: [76800 x 256], m2: [784 x 128]
    - Channel sizes mismatch
    - This may occur in many places of your model
  - Solutions
    - Check carefully that what the shape of output from the previous layer is and what the shape of the input of the current layer needs

Ref
https://stackoverflow.com/questions/49606482/how-to-resolve-runtime-error-due-to-size-mismatch-in-pytorch