Convolutional Neural Network (CNN) using PyTorch – Part 2 of 2

John Wong, 17 Jan 2024

Context

Part 1 of 2 (previous)

- You will have learned:
  - Convolution, Rectified Linear Unit Function, Max Pooling, Flattening, Fully Connected Layer, activation function
  - Image: Colour, grayscale, channels, sizes, tensor forms
  - Pickle file for images and their labels

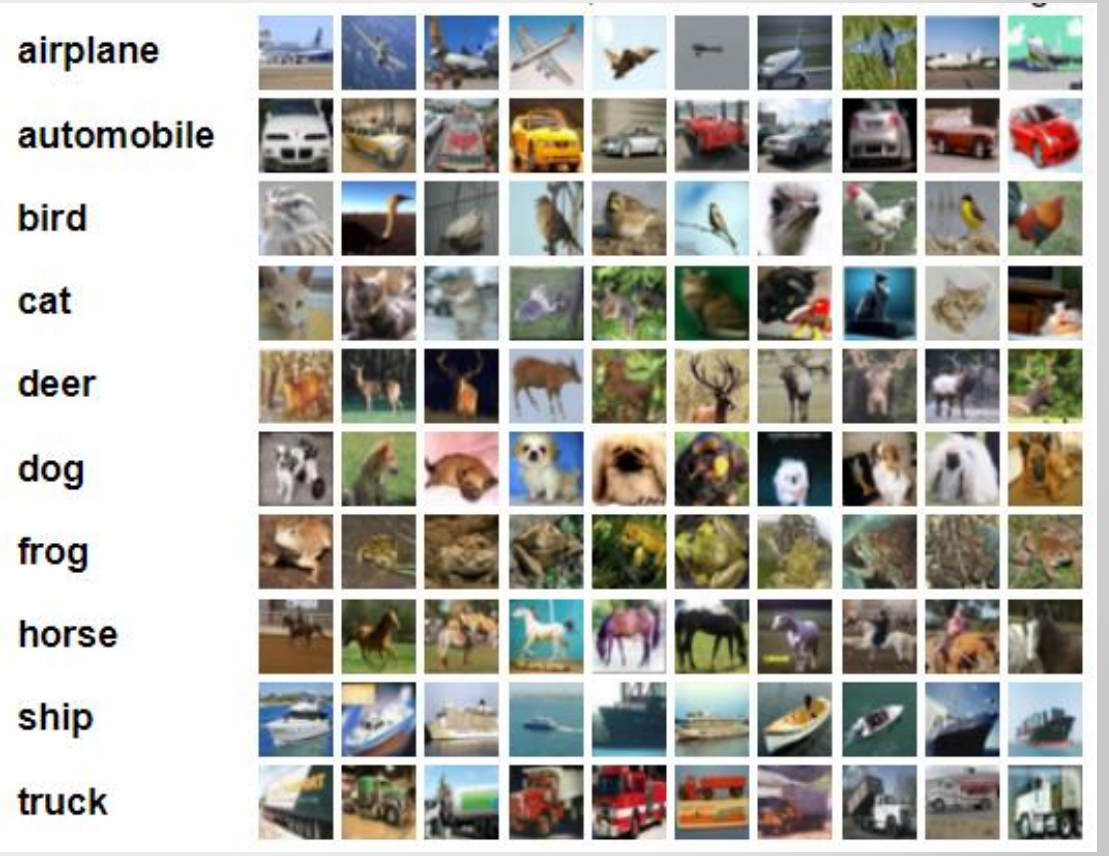
Context

Part 2 of 2 (current)

- What I wanted
  - Dive straight into image classification
- What I found out
  - Significant differences in performance (classification accuracy) of same model with different conditions (to explore as objective)

Dataset

- CIFAR-10
  - 50,000 data for training, 10,000 data for testing
  - 10 classes with 6,000 data per class
  - Size $3 \times 32 \times 32$
  - Zipped pickle files
  - Source 1: https://www.cs.toronto.edu/~kriz/cifar.html
  - Source 2: torchvision.datasets in PyTorch



What others have done

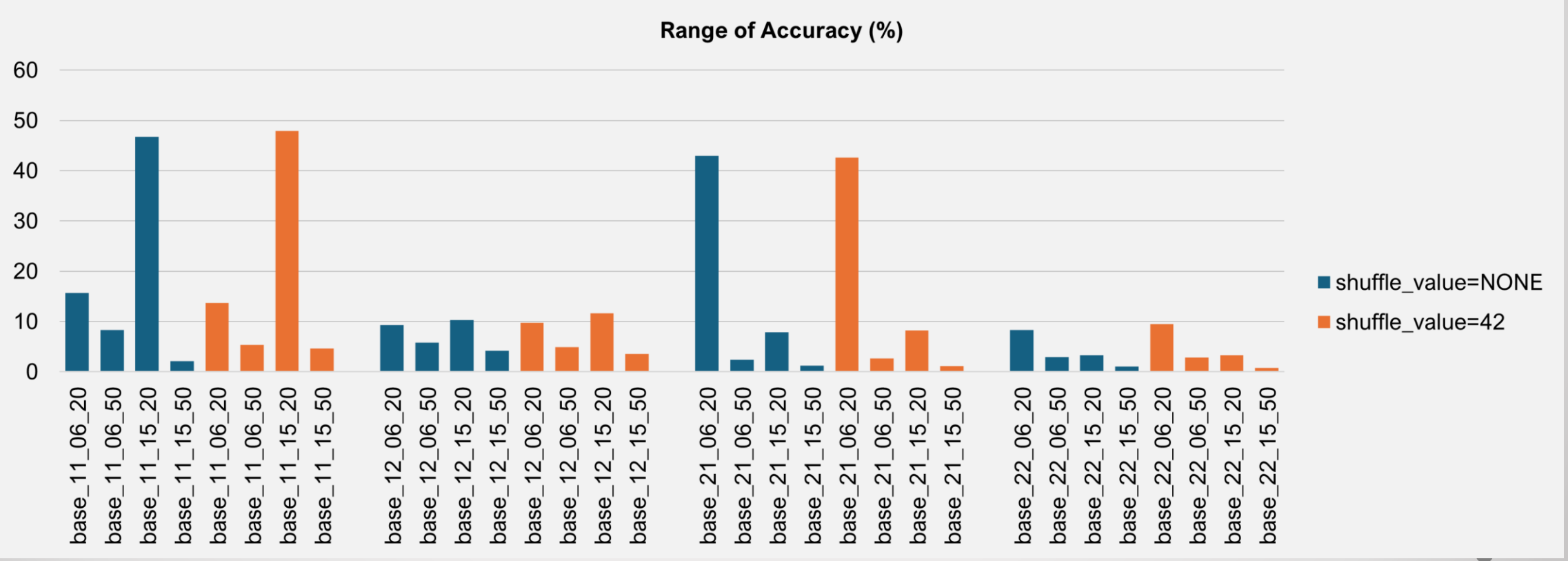
- Less complex
  - Conv2d, ReLU, MaxPool2d
- More complex
  - Added BatchNorm2d and Dropout
- Image augmentation
  - Horizontal flip, vertical flip, rotation

What I have done

- Base model
  - 1 convolution layer + 1 fully connected layer
  - 1 convolution layer + 2 fully connected layer
  - 2 convolutions layer + 1 fully connected layer
  - 2 convolutions layer + 2 fully connected layer
- Output
  - Convolution layer output: Combination of [6,15] for 1st layer and [12,30] for 2nd layer
  - Fully connected layer output: Combination of [20,50] for 1st layer and [40,100] for 2nd layer

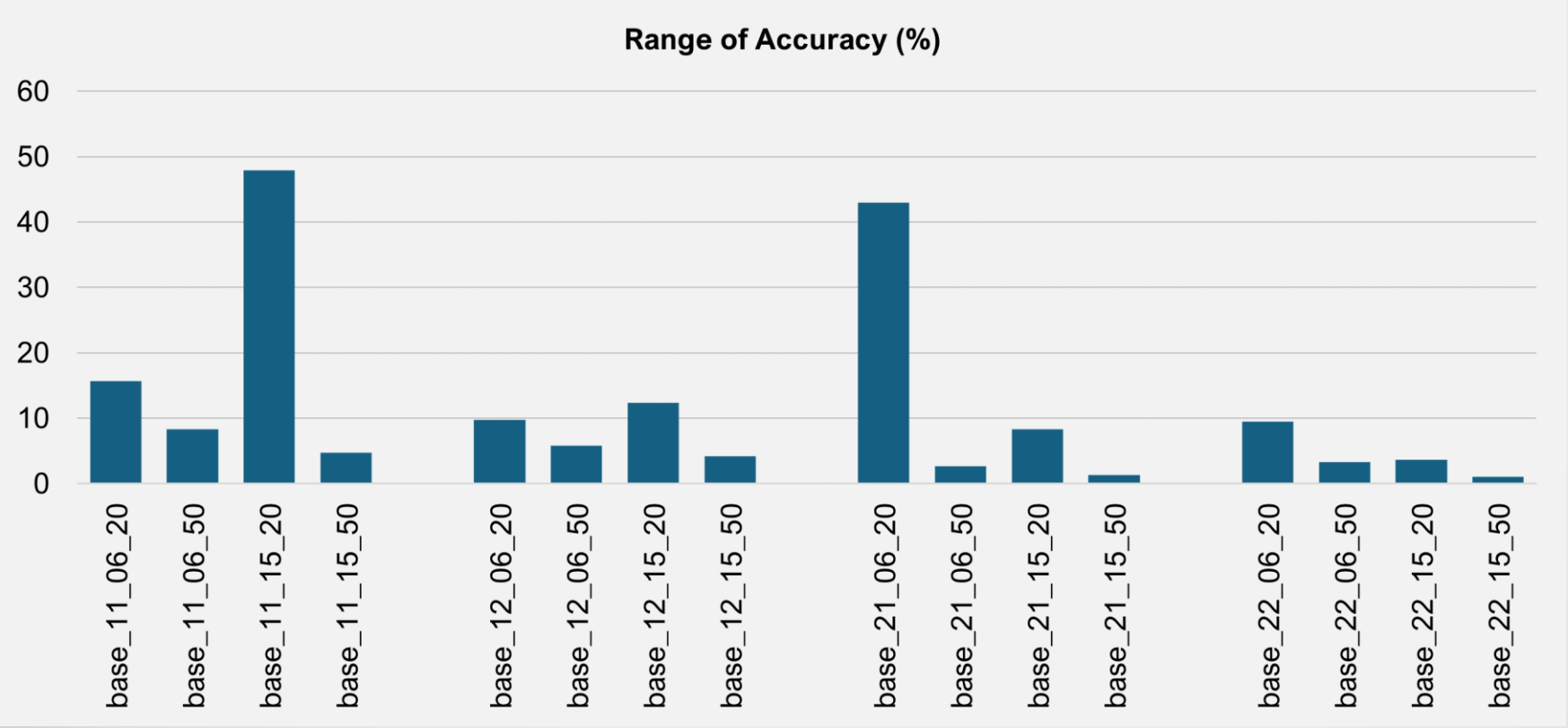
First discovery

- Range (maximum minus minimum) of classification accuracy between different seed values and shuffle values can be large and have high variability



Second discovery

- Range of classification accuracy between different seed values can be large and have high variability



Implications

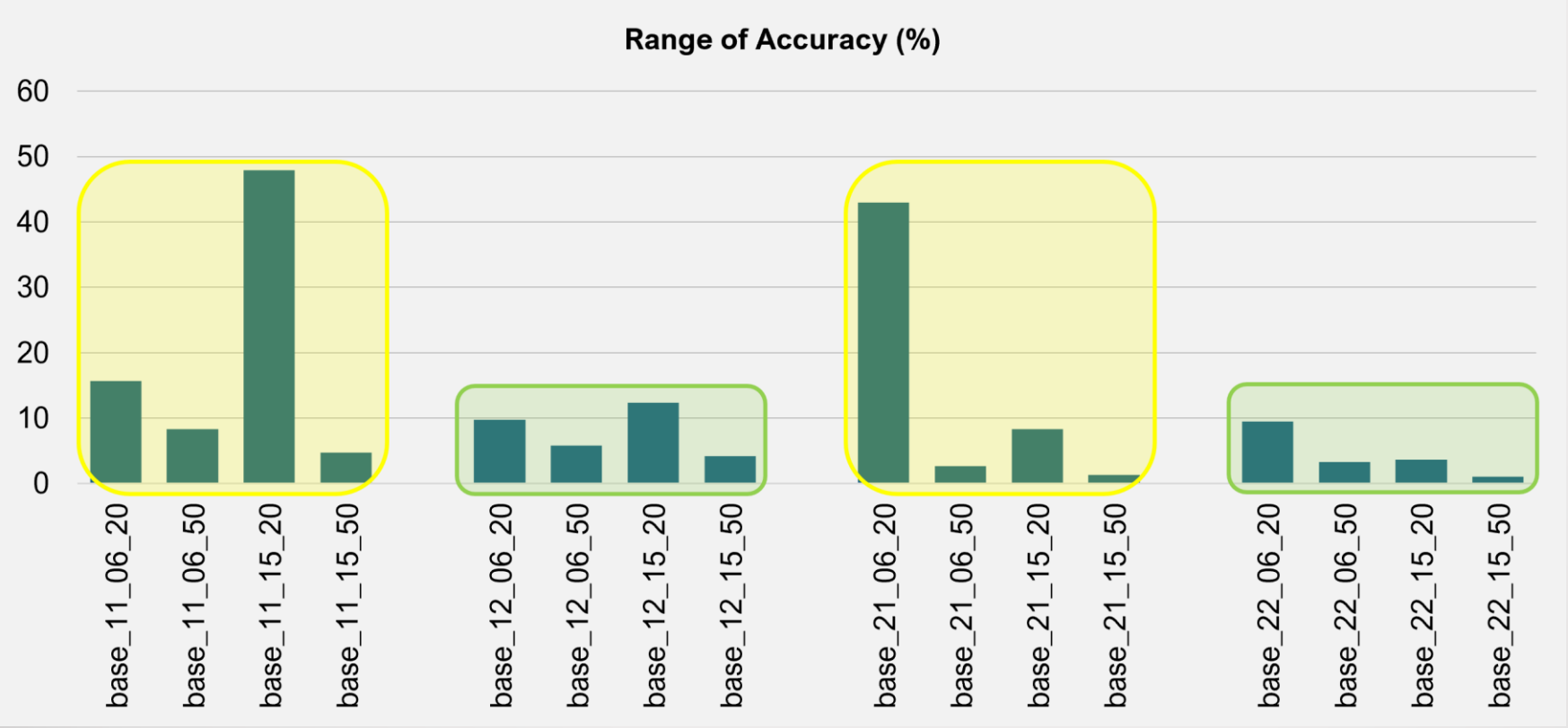
- To have true average performance of CNN model in production, it is important to train with different seed values and shuffle values.

Question

- With so many seed values, shuffle values, and limited time, how do one find the final values to train a CNN model that will give the highest classification accuracy consistently?

Possible answer

- Experiment with more Fully Connected Layer. Range of classification accuracy can be smaller and have lower variability.



The End

Hope you find this useful

Visit my GitHub for the codes

https://github.com/johnwck/my_da_ds_work/tree/master/my_projects_github_pages/pytorch_convolutional_neural_network_part_2