Deep Learning Assignment II

資訊碩一 P76121233 藍翊庭,

GitHub 連結: https://github.com/Sherry2580/Deep-Learning-Assignment-II.git

1. **Dataset:**

Mini-ImageNet is a subset of the ImageNet dataset which contains 50 classes.

- Training Set: 63325
- Validation Set: 450
- Testing Set: 450

Data Preprocessing :

- **Image Resize**: Since images vary in size, we resize all images to 256x256 pixels.
- **Data Augmentation:** Enhance the dataset using techniques like random horizontal flipping and rotation to improve the model's generalization ability.
- **Image Channel:** we convert all images to RGB format (3 channels).

➢ Channel Processing

- **Single Channel:** Duplicate the channel twice to create three identical channels, ensuring compatibility with the RGB format.
- **Double Channel:** Add an empty channel filled with zeros and append it to the existing two channels to form a three-channel image.
- **More than Three Channels:** Simplify by only utilizing the first three channels, disregarding any extra channels.

➢ Channel Mapping

- **Single Channel to Three Channels**: To effectively learn the transformations required to convert grayscale images to a pseudo-RGB format, we train a convolutional layer to map single-channel images to three channels.
- **Double Channel to Three Channels**: Similarly, we train another convolutional layer to map two-channel images to three-channel images, ensuring consistent channel numbers for all input images.

2. **Task 1: Designing a Convolution Module for Variable Input Channels**

Goal: Designing a convolutional neural network (CNN) that is capable of handling an arbitrary number of input channels. The primary task is to implement a special convolution module that is invariant to spatial size and can dynamically adjust to different input channels.

2.1. Hyperparameters

- Epoch: 25
- Learning Rate: 0.001
- Batch Size: 64
- Input Size: (3, 256, 256)
- Optimizer: Adam
- Loss: CrossEntropyLoss

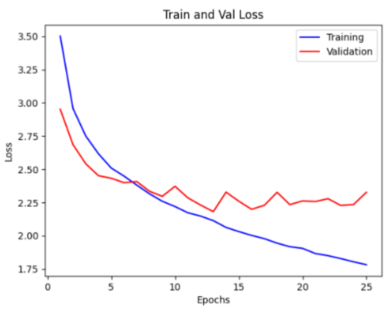
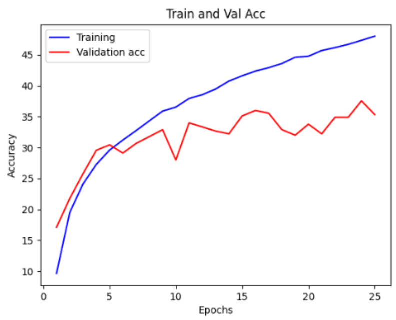
2.2. ImprovedCNN

- Model Architecture:

```
Device: cuda
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 32, 256, 256]             896
         MaxPool2d-2         [-1, 32, 128, 128]               0
            Conv2d-3         [-1, 64, 128, 128]          18,496
         MaxPool2d-4           [-1, 64, 64, 64]               0
            Conv2d-5          [-1, 128, 64, 64]          73,856
         MaxPool2d-6          [-1, 128, 32, 32]               0
            Linear-7                  [-1, 512]      67,109,376
           Dropout-8                  [-1, 512]               0
            Linear-9                   [-1, 50]          25,650
================================================================
Total params: 67,228,274
Trainable params: 67,228,274
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.75
Forward/backward pass size (MB): 35.01
Params size (MB): 256.46
Estimated Total Size (MB): 292.21
```

- **Train & Validation Acc/Loss**

  Overall, the model is able to learn from the data, improving accuracy and reducing loss. However, there is still a slight overfitting issue that needs to be addressed. And during training, we save the best model based on the validation loss.

Loss	Accuracy (%)
	

- **Test Accuracy**

	RGB	RG	GB	R	G	B
Accuracy(%)	35.11	19.11	13.78	25.78	25.33	22.0

3. Task 2: Designing a Two-Layer Network for Image Classification

Goal: The objective is to design a (2-4)-layer convolutional neural network (CNN), Transformer, or recurrent neural network (RNN) that achieves 90% of the performance of ResNet34 on the ImageNet-mini dataset.

3.1. Hyperparameters

- Epoch: 25
- Learning Rate: 0.001
- Batch Size: 64
- Input Size: (3, 256, 256)
- Optimizer: Adam
- Loss: CrossEntropyLoss

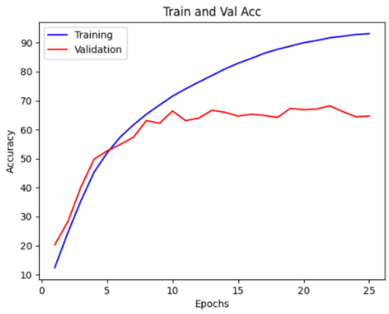
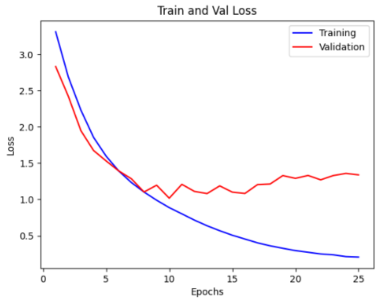
3.2. ResNet34

- **Model Architecture:**

```
================================================================
Total params: 21,310,322
Trainable params: 21,310,322
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.75
Forward/backward pass size (MB): 125.75
Params size (MB): 81.29
Estimated Total Size (MB): 207.80
----------------------------------------------------------------
```

- **Train & Validation Acc/Loss:**

  We save the best model based on the validation loss.

Accuracy	Loss
	

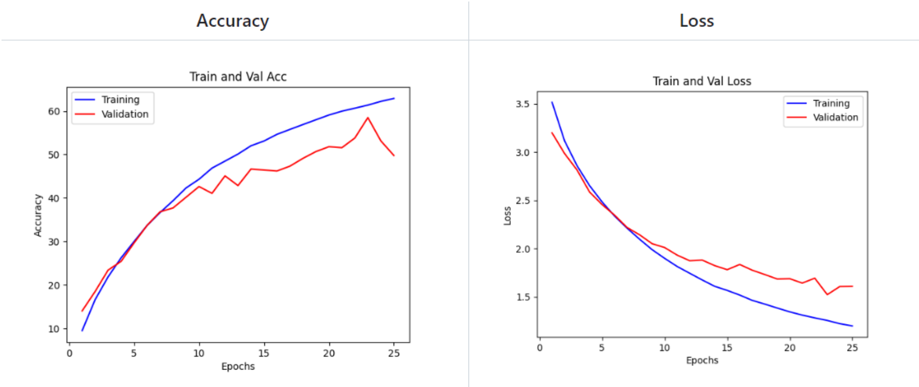
3.3. ComplexCNN

- **Model Architecture**

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 256, 256]           1,792
         MaxPool2d-2         [-1, 64, 128, 128]               0
            Conv2d-3        [-1, 128, 128, 128]          73,856
         MaxPool2d-4          [-1, 128, 64, 64]               0
            Conv2d-5            [-1, 1, 64, 64]              98
           Sigmoid-6            [-1, 1, 64, 64]               0
  SpatialAttention-7            [-1, 1, 64, 64]               0
            Conv2d-8          [-1, 128, 64, 64]         147,584
            Conv2d-9          [-1, 128, 64, 64]         147,584
           Conv2d-10          [-1, 128, 64, 64]         147,584
           Conv2d-11          [-1, 128, 64, 64]         147,584
           Conv2d-12          [-1, 128, 64, 64]         147,584
ResidualDenseBlock-13          [-1, 128, 64, 64]               0
          Dropout-14          [-1, 128, 64, 64]               0
AdaptiveAvgPool2d-15            [-1, 128, 1, 1]               0
           Linear-16                   [-1, 512]          66,048
           Linear-17                    [-1, 50]          25,650
================================================================
Total params: 905,364
Trainable params: 905,364
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.75
Forward/backward pass size (MB): 88.10
Params size (MB): 3.45
Estimated Total Size (MB): 92.30
----------------------------------------------------------------
```

- **Validation Accuracy & Loss**

  We save the best model based on the validation loss.



3.4. Comparison

Model	Accuracy (%)
ComplexCNN	58.667
ResNet34	67.778

The performance of the designed two-layer network (ComplexCNN) was evaluated against ResNet34. Despite showing promising results, the ComplexCNN marginally fell short of achieving 90% of ResNet34's performance on ImageNet-mini. The experimental results indicate that with further refinements, the designed model has the potential to reach and even surpass the target benchmark.

Conclusion : The designed ComplexCNN model was able to handle variable input channels effectively and achieved competitive performance. However, there is still room for improvement to match the performance of ResNet34. Future work can involve further tuning of hyperparameters, adding more advanced regularization techniques, and experimenting with different network architectures.