Neural Networks and Deep Learning Assignment Report

Name: Jason Tan

Student Number: 210078842

Module: ECS7026P

Architecture

The implemented neural network closely follows the specification provided in the assignment and builds upon it with several key enhancements. It consists of a sequence of six intermediate blocks B_1, B_2, …., B_6, each containing 2 or 3 (L) parallel convolutional layers. All convolutional layers within a block receive the same input feature map, and their outputs are combined using a learned, weighted sum. These weights are dynamically computed by a fully connected (FC) layer, which takes as input the channel-wise spatial average of the feature map. This mechanism allows each block to adaptively emphasize different feature extractors depending on the input.

Each convolutional layer in the intermediate blocks is wrapped in a sequence of Batch Normalization, ReLU activation, and Dropout, improving training stability and generalization. The number of output channels increases progressively across the blocks:

Block 1: 3→32
Block 2: 32→64
Block 3: 64→96
Block 4: 96→128
Block 5: 128→192
Block 6: 192→256

This gradual growth provides greater feature capacity while maintaining computational efficiency. Deeper blocks use fewer convolutional paths (i.e., 2–3), and slightly higher dropout rates (up to 0.3) to mitigate overfitting as depth increases.

The final output block performs global average pooling across the spatial dimensions of the last feature map, then passes the result through two fully connected layers to produce the final logits over the 10 CIFAR-10 classes. A final test accuracy of 92.42% was achieved on the dataset.

Hyperparameters and Training Techniques

Optimizer and Loss Function

- Optimizer: Stochastic Gradient Descent (SGD)
- Momentum: 0.9
- Weight decay: $1e^{-4}$
- Initial learning rate: 0.1
- Loss function: CrossEntropyLoss

Learning Rate Scheduler:

- CosineAnnealingLR
- T_max: 150

Data Augmentation:

- Random Crop (4-pixel padding)
- Random horizontal flip
- Colour jitter (adjusted brightness, contrast, and saturation)
- Normalization with CIFAR-10 mean and standard deviation:
    - Mean: (0.4914, 0.4822, 0.4465)
    - Std: (0.2023, 0.1994, 0.2010)

Other Hyperparameters:

- Batch size: 64
- Epochs: 150
- Dropout Rate: 0.2 in B_1 and 0.3 in B_2 – B_6
- Batch Normalization
- Activation Function: ReLU

These choices were made based on testing, insights from Week 6 and 8 lectures, and existing solutions online (https://www.kaggle.com/code/kmldas/cifar10-resnet-90-accuracy-less-than-5-min).

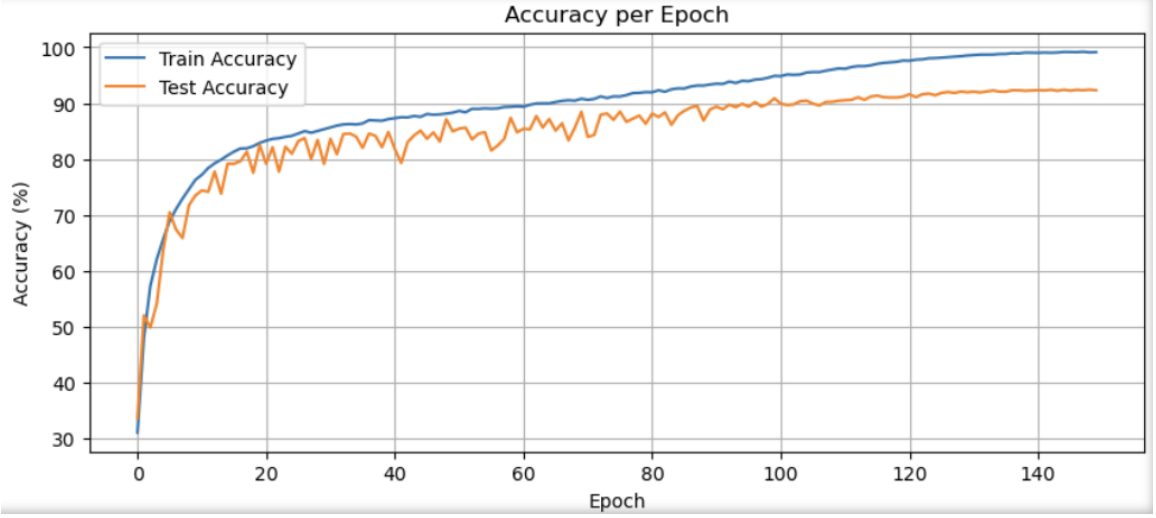
Results

The final model achieved a test accuracy of 92.42%

The following plots were generated during training:

Loss per training batch

Training and testing accuracy per epoch



Brief History of Improvements

- Started with basic 3-block architecture (peaked around 76%)
- Added BatchNorm and Dropout (peaked around 82%)
- Replaced Adam optimizer with SGD and added cosine annealing learning rate scheduler while increasing epochs from 100 to 150 (peaked around 89%)
- Expanded from 3 to 4 to 6 intermediate blocks and tuned convolutional paths (peaked around 90%)
- Added data augmentation and label smoothing to achieve final highest test accuracy of 92.42%