HW9 – Short Answers

Arnav Jain, Solomon Starkes, and Sidy Thiam

1. Parameter Count of the CNN

Below is a layer-by-layer calculation of the total number of trainable parameters in our final CNN architecture. As required, batch normalization parameters are excluded from the count.

Convolutional Layers

Conv1: $3 \to 32$, kernel 3×3, stride 1, padding 1

$$\text{Params} = 32 \times (3 \cdot 3 \cdot 3 + 1) = 32 \times 28 = 896$$

Conv2: $32 \to 64$, kernel 3×3, stride 1, padding 1

$$\text{Params} = 64 \times (32 \cdot 3 \cdot 3 + 1) = 64 \times 289 = 18{,}496$$

Conv3: $64 \to 128$, kernel 3×3, stride 1, padding 1

$$\text{Params} = 128 \times (64 \cdot 3 \cdot 3 + 1) = 128 \times 577 = 73{,}856$$

Conv4: $128 \to 256$, kernel 3×3, stride 1, padding 1

$$\text{Params} = 256 \times (128 \cdot 3 \cdot 3 + 1) = 256 \times 1{,}153 = 295{,}168$$

Total Convolutional Parameters:

$$896 + 18{,}496 + 73{,}856 + 295{,}168 = 388{,}416$$

Fully Connected Layers

After four pooling operations and an adaptive max-pool to 4×4, the flattened feature size is:

$$256 \times 4 \times 4 = 4096.$$

FC1: $4096 \to 256$

$$\text{Weights} = 4096 \times 256 = 1{,}048{,}576, \qquad \text{Biases} = 256$$

$$\text{Total FC1 Params} = 1{,}048{,}576 + 256 = 1{,}048{,}832$$

FC2: $256 \to 11$ classes

$$\text{Weights} = 256 \times 11 = 2{,}816, \qquad \text{Biases} = 11$$

$$\text{Total FC2 Params} = 2{,}816 + 11 = 2{,}827$$

Total Fully Connected Parameters:

$$1{,}048{,}832 + 2{,}827 = 1{,}051{,}659$$

Grand Total

$$\textbf{Total Parameters} = 388{,}416 + 1{,}051{,}659 = 1{,}440{,}075.$$

Thus, the final network contains **1,440,075 trainable parameters**.

2. Improving CIFAR-10 Accuracy: What Was the Issue?

The original `CNN_small` model contained only a single convolutional layer followed by a large fully connected layer. This architecture was too shallow to learn meaningful hierarchical features from CIFAR-10, which caused the accuracy to plateau at a relatively low value.

To improve performance, we made the following changes to CNN_small:

1. **Added an extra convolutional layer:** A second convolution allowed the model to extract richer spatial features before flattening.
2. **Added max pooling:** Pooling reduced spatial dimensions and introduced translation invariance, while also reducing the size of the fully connected layer and making optimization easier.
3. **Included batch normalization and ReLU activations:** Batch normalization stabilized training and helped gradients flow better, improving convergence.
4. **Tuned basic hyperparameters:** Adjusting learning rate and training for more epochs improved optimization and final accuracy.

These modifications enabled the warm-up model to comfortably exceed the required 70% test accuracy on CIFAR-10.