Project Report

Name– Shobhit Rawat
Student ID - 230262571
Module Code – ECS659P/ECS7026P

INTRODUCTION

This report outlines the process of developing a neural network and the results achieved in classifying images from the CIFAR-10 dataset, which consists of 60,000 color images, each 32x32 in size, distributed across ten categories. The objective was to construct a foundational architecture and subsequently enhance the classification accuracy through iterative refinements.

NEURAL NETWORK ARCHITECTURE

In our upgraded neural network architecture, we've made several significant enhancements compared to a simpler, more basic reference model. The reference architecture typically features two blocks, each equipped with two convolutional layers, without the incorporation of advanced techniques such as batch normalization or specialized activation functions. In contrast, our model expands this to **four blocks**, each containing **three convolutional layers**, to capture more complex features and increase the network's depth and complexity.

Our modifications include the addition of batch normalization and **ReLU** activation following each convolutional layer. These additions help stabilize the learning process and speed up convergence by normalizing layer inputs and introducing non-linearity, respectively, without affecting the receptive fields of the convolution operation. Furthermore, we implement an **AdaptiveMaxPool2d** and a dropout layer with a rate of **0.2** post-convolutional stages. This approach not only reduces spatial data but also helps in mitigating overfitting by randomly dropping units during training, enhancing the model’s robustness.

A distinctive feature of our architecture is the introduction of a feature re-weighting mechanism within the **IntermediateBlock**. Here, outputs from a fully connected layer are passed through a **SoftMax** function to weigh the outputs of each convolutional layer before they are summed. This dynamic weighting allows the network to prioritize more informative features actively. Additionally, our output block integrates a **GlobalAvgPool2d**, which condenses each feature map to a single value, summarizing the spatial information before it reaches the fully connected layer for final classification.

These structural enhancements not only improve the network's ability to generalize from training to unseen data but also make it more adaptable and robust for various image recognition tasks, setting it apart significantly from the simpler designs typically seen in basic models.

HYPERPARAMETERS AND TECHNIQUES USED

In the training and evaluation of our enhanced neural network model, several key hyperparameters and techniques are employed to optimize performance and ensure robust learning:

1. Hyperparameters:

- **Learning Rate**: Set at **0.01,** the learning rate dictates the step size at each iteration while moving toward a minimum of a loss function. This rate is crucial for balancing the speed and stability of the training process.
- **Weight Decay**: Configured to **10e-6** weight decay is used as a regularization technique to prevent overfitting by penalizing large weights.
- **Number of Epochs**: The model is trained for **60** epochs, allowing sufficient iterations over the entire dataset to minimize the loss function effectively.
- **Batch Size**: Set at **128**, this size determines the number of training samples processed before the model's internal parameters are updated.

2. Techniques:

- **Optimizer**: The **AdamW** optimizer is utilized, which is an extension of the Adam optimization algorithm. It includes a correction factor to the update rule, helping to improve convergence and make the training more stable, especially in complex networks and datasets.
- **Loss Function**: **Cross-Entropy Loss** is used, which is standard for classification tasks. This function measures the performance of the classification model whose output is a probability value between 0 and 1.

RESULTS AND ACCURACY

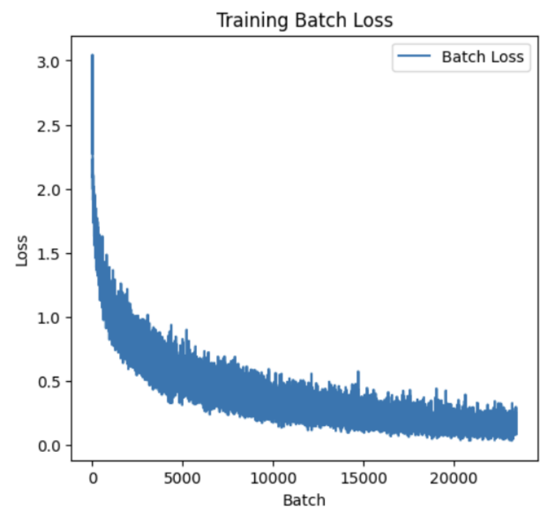
The highest accuracy recorded on the testing dataset throughout the training epochs is **87.04%.**

This figure represents the peak generalization performance of the neural network, highlighting the efficacy of the architecture and the training regimen employed.

```
Epoch 58, Train Accuracy: 94.774%, Test Accuracy: 87.04%
```

VISUALIZATION OF RESULTS.

- **Plot of the loss for each training batch.**



The plot traces the trajectory of training loss across each batch, which shows a pronounced initial decrease and then a more gradual decline, stabilizing as the epochs progress.

- **Plot of the training accuracy and testing accuracy for each training epoch.**

The second plot contrasts the training and testing accuracy at each epoch, illustrating a consistent improvement in the model's ability to classify both seen and unseen data. The training accuracy is predictably higher; However, the testing accuracy also shows a commendable increase, which underscores the model's generalization capabilities.