2248-CSE 6363-002 - MACHINE LEARNING – HW2 REPORT

Team Members

Nikhil Reddy Duggireddy-1001968033  
Laya Kalali-1002128876  
Sasikala Paturu-1002123123  
Lokeshwar Kodipunjula-1002175121

IMPLEMENTING NEURAL NETWORKS AND CNNS USING CIFAR-10 DATASETS:

Part 1-Fully Connected Network from Scratch:

1.1 Loading the CIFAR-10 Dataset  
Dataset Loaded: The CIFAR-10 dataset was loaded with PyTorch, normalized using a mean and standard deviation of 0.5, and split into 90% training and 10% validation sets.  
Data Preprocessing: Images were flattened into 3072 features (32x32x3), and labels were one-hot encoded.

1.2 Implementing the Neural Network  
Architecture: A fully connected neural network was designed with one hidden layer of 128 neurons and an output layer of 10 classes using a softmax activation.  
Training: The model was trained using cross-entropy loss and gradient descent, running for 50 iterations with a batch size of 64 and a learning rate of 0.01.  
Results: After 50 iterations, the model achieved 66.15% accuracy on the training set and 51.14% on the validation set.

1.3 Predictions  
The trained model was used to make predictions, some of which were displayed, highlighting its ability to correctly classify test images (e.g., accurately predicting a "frog" image).

Part 2: Convolutional Neural Network using PyTorch:

**2.1 Loading the CIFAR-10 Dataset**  
The CIFAR-10 dataset, previously used in Part 1, was reloaded, and PyTorch data loaders were set up for training, validation, and testing purposes.

**2.2 Implementing the CNN**  
Architecture: The CNN consisted of two convolutional layers, each followed by ReLU activations, batch normalization, and pooling layers, with fully connected layers added at the end.  
Training: The model was trained for 10 epochs using the SGD optimizer, with a learning rate of 0.01 and momentum applied.  
Results: After completing the 10 epochs, the model reached a training accuracy of 97.68% and a validation accuracy of 74.76%.

**2.3 Predictions**  
The model made predictions on test images, correctly classifying several images, such as identifying a "cat," "ship," and "frog" accurately.

Part 3: Methods to Improve Performance:

**Optimizer Change**: The Adam optimizer was tested with various learning rates (0.001, 0.01, 0.0001). The model trained using Adam with a learning rate of 0.001 reached a final validation accuracy of 74.76%.

**Data Augmentation**: To enhance the model's generalization, data augmentation techniques like random horizontal and vertical flips, random rotations, and color jitter were applied.

Conclusion:

In this assignment, we implemented a fully connected neural network (FCN) and a convolutional neural network (CNN) using PyTorch to classify CIFAR-10 images. The FCN, with one hidden layer and ReLU activation, achieved a training accuracy of 66.15% and validation accuracy of 51.14%, but its inability to capture spatial information limited its performance. In contrast, the CNN, with two convolutional layers and max-pooling, performed significantly better, reaching 97.68% training accuracy and 74.76% validation accuracy after 10 epochs. We improved performance further by using the Adam optimizer, testing different learning rates, and applying data augmentation techniques like random flips and rotations. Ultimately, the CNN met the assignment's goal of 70% validation accuracy, demonstrating the power of convolutional layers for image classification.