

CIFAR-10 Classification using Custom Convolutional Neural Network

This project focuses on building a convolutional neural network (CNN) and a Multi-Layer perceptron to classify a subset of the CIFAR-10 dataset, specifically cat, deer, and dog classes. The CNN model is designed to handle both training and testing, with a custom convolutional layer (ManualConv2d). TensorFlow's standard Conv2D layers were only used for training to enhance speed and efficiency, while the custom convolution layer was used for testing.

Model Architecture

The CNN consists of:

- Three convolutional layers: the first convolution uses TensorFlow's Conv2D during training and switches to the custom ManualConv2d for testing.
- Batch normalization and ReLU activations following the convolution layers.
- Max-pooling layers for down-sampling.
- A fully connected layer to output class logits for classification.

MLP Model consists of:

- The MLP consists of fully connected layers
- The input images are flattened before being passed through several dense layers with ReLU activations.
- The final layer outputs logits for classification into the cat, deer, and dog classes.

Both models were trained for 10 epochs.

Results

CNN Model

Training Accuracy:

- Epoch 1: 65.77%
- Epoch 2: 71.27%
- Epoch 3: 75.98%
- Epoch 4: 76.77%
- Epoch 5: 79.69%
- Epoch 6: 81.01%
- Epoch 7: 82.76%
- Epoch 8: 86.32%
- Epoch 9: 87.71%
- Epoch 10: 90.30%

Testing Accuracy:

- Final Testing Accuracy: 77.30%

MLP Model**Training Accuracy:**

- Epoch 1: 53.73%
- Epoch 2: 54.49%
- Epoch 3: 52.69%
- Epoch 4: 57.67%
- Epoch 5: 58.80%
- Epoch 6: 57.11%
- Epoch 7: 58.03%
- Epoch 8: 60.23%
- Epoch 9: 60.37%
- Epoch 10: 60.87%

Testing Accuracy:

- Final Testing Accuracy: 60.03%

Files

At present, there are no known bugs or issues with the model. The training process runs smoothly, and the custom convolution layer functions correctly during testing.

Following files are included in this submission:

- assignment.py: Main script to train and test the model.
- cnn.py: Contains the CNN model definition.
- manual_convolution.py: Implementation of the custom ManualConv2d layer used during testing.
- mlp.py: An alternative model file.
- base_model.py: Contains common functionality, including the loss and accuracy methods.
- preprocess.py: Handles the data preprocessing.
- predictions_cnn.npy: Contains the model's predictions for the test dataset.
- README.md: This file.

Conclusion

Both models performed reasonably well, with the CNN surpassing the required 75% testing accuracy and the MLP reaching the 60% threshold. The CNN demonstrated strong generalization, with the custom convolution layer integrated for testing. The MLP model also showed steady improvement, achieving 60.87% training accuracy and 60.03% testing accuracy.