[CIFAR10-project]

STUDENT NUMBER:3006868

Introduction

Giving one or more class labels to an input image is a fundamental problem in computer vision known as image classification. Due to their capacity to recognize spatial elements in images, convolutional neural networks (CNN) have been extensively used for image classification tasks. In this project, the goal was to create a CNN model that could categorize photos from the CIFAR10 dataset into ten different groups: truck, ship, airplane, car, bird, cat, deer, and frog. In the Cifar-10 dataset, there are 60,000 32*32 color images with 5000 images per class. 10,000 test photos and 50,000 training images are available.

Proposed solution

I implemented my CNN image classification model using PyTorch. There are three convolutional layers in the CNN. Except for the final fully connected layer, the ReLU activation function is applied after each convolutional and fully connected layer. A softmax activation function receives the output of the final fully connected layer and produces a probability distribution over the ten classes.

This architecture was my choice because it has a proven track record of success on Cifar-10 classification assignments and is also a pretty straightforward architecture to put into practice. The max-pooling layers aid in minimizing the spatial dimensions of the input, which aids in lowering the number of model parameters and assisting in avoiding overfitting. The ReLU activation function was selected because research has shown that it can help deep neural networks perform better. As a standard loss function for classification problems, I also utilized cross-entropy loss**.**

RESULTS

With a "val acc" of 2.0956456810235977 2 and a "val loss" of 0.3040597438812256, the model was operating satisfactorily**.**

Discussion

In this project, I first loaded the CIFAR-10 dataset using the PyTorch library, normalized the pixel values, then divided the dataset into training and validation datasets. In order to efficiently load the data in batches during training and validation, I created a dataloader object for each dataset. Next, I determined some of the characteristics of the data set and later visualized them using the make grid helper function from torch vision. After that, I moved on to the base model class and GPU training, and the final step was training the model and evaluation.

Conclusion

In conclusion, I have developed a convolutional network using PyTorch to correctly classify images from the CIFAR-10 dataset, and my implementation can be used as a starting point for more complex image classification.