MANARAT INTERNATIONAL UNIVERSITY

Department of Computer Science and Engineering

Project Report on:

Computer Vision & Robotics (CSE-433)

Neural Network & Fuzzy System (CSE-437)

Contest Name:

CIFAR-10 Object Recognition in Images

Name of the Team: Rokon-Rahman

Contestants Name & ID: Abdur Rahman (1641CSE00556)
Nafis Fuad Shovon (1641CSE00558)
MD Abir Hossain (1641CSE00544)

Click ***here*** for git source

Introduction:

The CIFAR-10 dataset is a collection of images that are commonly used to train computer vision algorithms. It is one of the most widely used datasets for deep learning research. The CIFAR-10 dataset contains 60,000 32x32 color images in 10 different classes which are airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6,000 images of each class.

This lab project offers a Kaggle competition on this dataset for the MIU students to implement deep learning algorithms which have been taught on both courses in a cool real-world project. The competition challenges you to predict the labels of total 300,000 images on this dataset using only 60,000 labeled images for training.

Data preprocessing:

As our contest problem we have 50000 train sample, 300000 test samples, total 10 number of classes, our image size was 32*32, and our image channel was 3(RGB), so we get our image shape 32*32*3. We use 0.001 learning rate for train our image, our image batch size was 500 (as we have 8GB RAM). To train our images we use around 60 to 350 epochs.

For data preprocessing, we have used normalization and augmentation. In this process, we have loaded train image dataset and also import given train data labels. Then we have done one-hot encoding image label.

Normalization:

It is a process that changes the range of pixel intensity values. To normalize images in this process, we divide image pixel by 255, then subtract by 0.5 and multiply by 2. By normalization we get the image range into -1 to +1.

We use data augmentation for increase our training data set: we zoom in our images in range=0.2, rotation our images in range=15, width shift range=0.1, height shift range=0.1

Network Architecture:

To complete this project, we have used Baseline CNN Model.

In Baseline CNN Model we use ReLU.

“Baseline CNN Model” and “Simple CNN 90%+ Model”.

In Simple CNN 90%+ Model we use Leaky ReLU.

Training Procedure:

First, we have used ReLU in our Baseline Model. We have used convoc2D (32) where pooling size is 2,2 and dropout is 0.20. Secondly, we have used (64) where pooling size and dropout are same. We have also used L2 regularizers (.01) and then used softmax activation function. Here we train our data set around 3 times. And get average result. It was round 80% accuracy.

After using this model, we use Simple CNN 90%+ Model for better performance. In this model, we have used leaky ReLU with alpha where alpha =0.01. we also use convoc2D three times (32,64,128) where pooling size is 2*2 and dropout for 32 convoc2D is 0.05, for 64 convoc2D is 0.1, for 128 convoc2D is 0.15. Here, we also use softmax activation function. Here we train our data set around 4 times. And we get around 88% accuracy. That was better than previous result.

Result:

When we use the baseline CNN model, we got our First Submission Score which was 0.79510   by running around 50 epochs. By upgrading epochs by 100 at the beginning we faced some problem. Because training files stopped working because of the upgrade. That took 72/100 epochs. So, we updated the practice by making it 50 to 30. Then we got our second submission score which was 0.80030.

Conclusion:

During the working time we face difference problem of the project. As we had only 50000 train images, with the help of augmentation we increase our data set. Convocation layer is another problem, we can’t increase our convocation layer because our pc becomes slower with the layer increase. Filter size is another problem as like as convocation layers. High number of parameters is a common problem. To reduce the parameters number, we have to use drop out. And the train images batch size really depends on user pc’s RAM. We use 500 batch size as we have 8GB of RAM.

Our PC configuration:

OS: Windows 10 pro

Processor: Intel core i3

RAM: 8GB

GPU: Intel HD (with 4GB of shared graphics)