Classification of CIFAR10 using CNN

Reetu Hooda
University of Alabama, Huntsville
rh0059@uah.edu

I. Introduction

Convolutional neural network (CNN) have been one of the most popular deep learning methods in solving many computer vision related tasks. It takes locality of input into account and performs very well on both coarse and fine level details. The depth of a CNN plays an important role in the discriminability power the network offers. The deeper the better. This project focuses on solving an image classification problem using CNN. Our experimental results shows that 85.03% image classification accuracy is obtained by the framework proposed.

II. The Dataset

CIFAR-10 dataset consists of 60,000 images of each size 32×32 of 10 classes with 6000 images per class. Out of 60,000 we use 40,000 in training set, 10,000 in validation set and 10,000 in testing set. The test batch includes exactly 1000 randomly selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class. Fig. 2 shows 7 samples from each class.

Fig. 2. CIFAR-10 Dataset

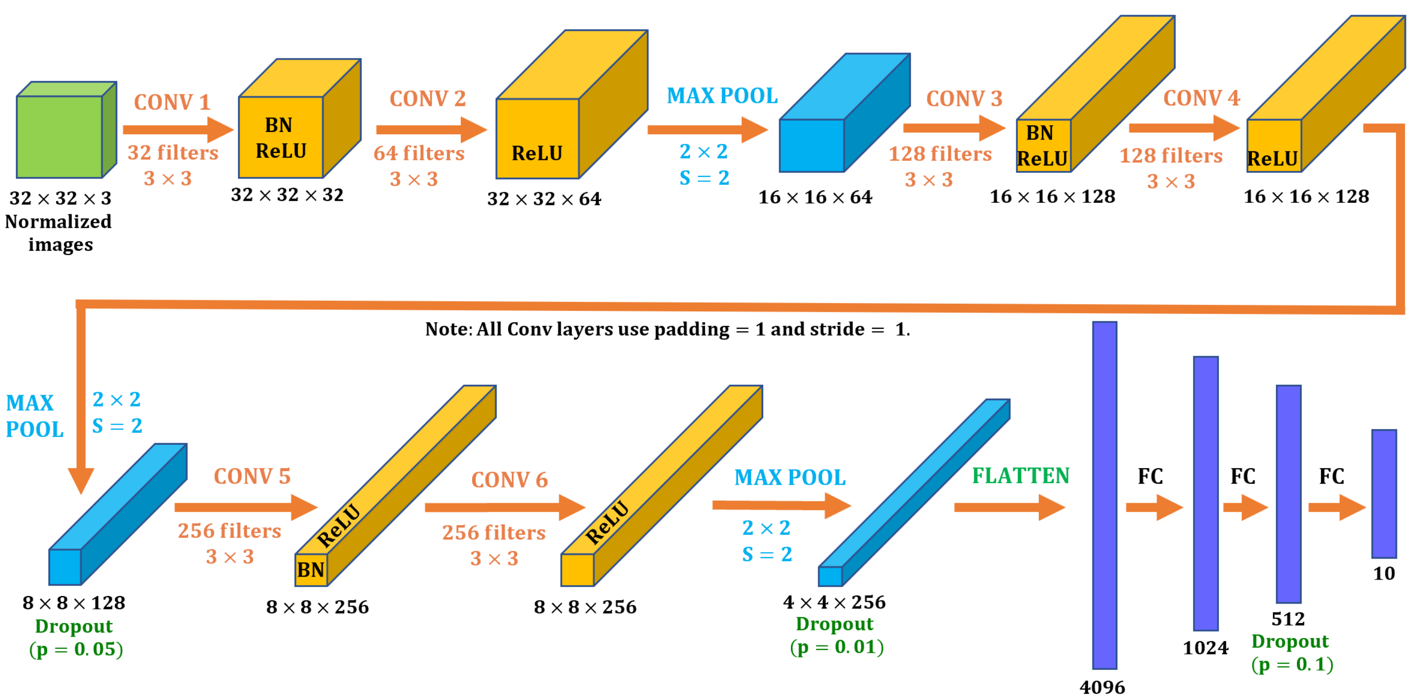

Fig. 1. CNN Network Architecture

III. Network Architecture

The proposed architecture has 6 convolution layers, 3 max-pooling layers and 3 fully connected layers. The configuration of the network is shown in Fig 1. It is well known that the training time and total number of layers have linear relations in CNN architectures. In other words, increasing the number of layers increases the training time. Therefore, we are using GPU source for training the network. All the convolution layers use filters of size 3×3 with a stride and padding of 1. Varying number of channels are used such as $32, 64, 128$ and 256. 2×2 max-pooling layers with a stride of 2 are used. All the layers except last one employs ReLU activation function. A softmax activation function is used for the final layer. The input size of the network was set that of CIFAR-10 RGB images i.e., $32 \times 32 \times 3$. All the images are normalized before feeding to the network using Z-normalization:

$$X = \frac{X - \mu(X)}{\sigma(X)}$$

Data Augmentation : We are using data augmentation to train the proposed large CNN. It will increase the diversity of data available doe training models without actually collecting new data. We have used random cropping and random horizontal flipping for data augmentation. Batch Normalization and Dropout is used to prevent the network from overfitting.

Loss and Optimizer: Since we are using softmax activation for output layer while developing the model, we define cross entropy loss as:

$$E = -\sum_{i}^{nclass} y_i \log(\hat{y}_i)$$

where y_i are the target values encoded as one-hot vectors and $\hat{y}_i$ are the predictions made by our model. Stochastic Gradient Descent (SGD) algorithm is used for training the model with learning rate of 0.001, momentum rate of 0.9 and weight decay of 5×10^{-4}.

IV. Results

The model was trained for 50 epochs. Table I summarizes the accuracies achieved by the model. Fig 3 shows Loss and Accuracy over each epoch, Fig 4 shows the confusion matrix and Fig 5 shows the accuracy of each class.

TABLE I
Accuracy

Set	Training 40,000	Validation 10,000	Testing 10,000
Accuracy	93.7%	86.84%	85.03%

V. Conclusion

In this report we present a CNN architecture for classifying images in the CIFAR10 dataset. The proposed model achieve appreciable training, validation and testing accuracy where overfitting is addressed using dropout and batch normalization.

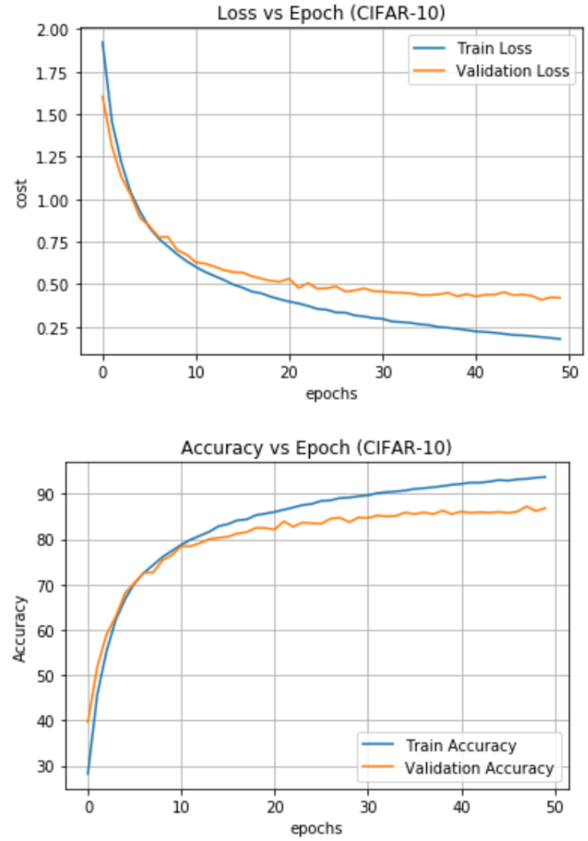

Fig. 3. Loss and Accuracy Vs Epoch

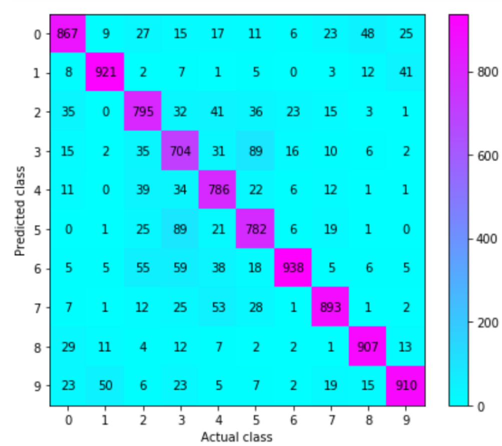

Fig. 4. Confusion Matrix

```
Accuracy of plane : 82 %
Accuracy of   car : 94 %
Accuracy of  bird : 78 %
Accuracy of   cat : 77 %
Accuracy of  deer : 82 %
Accuracy of   dog : 74 %
Accuracy of  frog : 88 %
Accuracy of horse : 86 %
Accuracy of  ship : 92 %
Accuracy of truck : 92 %
```

Fig. 5. Accuracy of each class