Image Classification using CNN

CIFAR-100 Dataset

Srishti Sharma
School of Engineering and Applied Sciences
Ahmedabad University
Ahmedabad, India
srishti.s1@ahduni.edu.in

***Abstract*—Machine Learning has evolved to make systems identify the various properties of an object by training the system first with various data of a type similar to that of the object. This identification of similarities and dissimilarities in objects has contributed to the development of image classification using machine learning. Deep learning, a subset of machine learning has lead to the development of Convolutional Neural Networks for predicting objects in an image and classifying the image on the basis of these identified similar patterns. This assignment demonstrates image classification over CIFAR-100 dataset using a CNN that results in an accuracy of around 73% over the test dataset.**

Keywords—Image Classification, CNN, CIFAR-100

I. Introduction

Image Classification is an approach that tries to analyze an image as a whole. The objective behind the image classification task is to assign a label to an image on the basis of the pattern identified from it. For example, an image can be of a flower or can be of an apple or a bird and so on. This task of image classification works only on images that have a single object which can be analyzed. While object detection deals with classification as well as localization as there can be multiple objects involved, image classification deals with a single object images only [1].

The steps involved in the image classification model building using neural networks are as follows: the first step is flattening of the input dimension of the image to 1D that is width_pixel*height_pixel. Next step is normalization of the image pixel values that is dividing the pixel value by 255. This step is followed by one-hot encoding over the categorical column. Next step involves building of the model architecture with dense layers. And the final step involves training the model and then making predictions [5].

Amongst the various types of neural networks such as artificial neural network (ANN), recurrent neural network (RNN), long short term memory (LSTM), convolution neural networks (CNN) are the ones that have become the most popular [2]. The CNNs were first made use of for classifying the hand written digits on MNIST dataset. The CNNs are now found in applications ranging from photo tagging on facebook to self driving autonomous cars [3].

II. Image Classification

Convolution neural networks have convolutional layers, ReLU layers, pooling layers and a fully connected layer. The convolutional layer applies convolution operation on the input and passes information to the following layer. Pooling layers do the task of combining output of a group of neurons into a single neuron in the following layer. Whereas the fully connected layers connect each neuron in the current layer to each neuron in the following layer [5]. A classic CNN architecture would look like:

Input ->Convolution ->ReLU ->Convolution ->ReLU ->Pooling ->
ReLU ->Convolution ->ReLU ->Pooling ->Fully Connected

CNNs work by extracting features from the input image and hence there is no need of manual feature extraction. These features are not taught rather learnt which makes these classification models accurate. The CNN starts with an input image and then applies various filters to the image to form a feature map. The next task is application of an activation function like ReLU so as to increase the non-linearity. Next a max pool layer or an average pool layer is applied to each feature map. The pooled images are then flattened into a single long vector. This vector is then passed as an input to a fully connected layer that provides the exact class label. The model is trained through forward and backward propagation for many epochs till a neural network with trained weights and feature detecting capabilities is obtained [6].

When we make use of CNNs for image classification, there is no need of initially flattening the input images to 1D as the CNNs have the capability of working with the image data in 2 dimensions that is 2D. This property of CNNs helps in retaining the “spatial” image properties [2].

III. Experiment and Results

A. Dataset

The dataset used is named CIFAR-100 [4]. This dataset consists of tiny images of objects such as apple, bed, baby, bird, cat, deer, dog, ship, truck etc. There are 100 classes each containing 600 images present in this dataset. Each image has 3 RGB colour channels and a pixel dimension 32*32 which makes the overall size per input equal to 32*32*32 that is 3072. For each class there are 500 training images and 100 testing images. Hence in all, the dataset has 50K training images and 10K testing images. These images were collected by hey Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. In this dataset, all he 100 classes are divided into 20 groups of superclasses. Hence, each image has a “fine” label associated with the class it belongs to and a “coarse” label associated with the “superclass” it belongs to.

B. Discussion

Initially the dataset is first loaded into two folders train and test. The dataset is then randomized by applying data augmentation and data normalization. These augmentation techniques include cropping, padding and horizontal flipping. The normalization step involves working upon image tensors by subtracting the mean and dividing by standard deviation of the pixel across each channel. And then a batch of 400 images is prepared for processing. I made use of GPU for processing as CPU was time consuming. Next functions for accuracy and loss calculation were defined. Cross-entropy loss is used here.

The CNN used has input, output and boolean value pool. Kernel size used is 3*3. Normalization and ReLU activation function is used. The pooling layer used takes maximum value from a 2*2 box. The model has 12 layers that include 5 convolution layer and 6 residual layers. Hyper parameters used are learning rate, gradient clipping, epochs, and weight decay and optimization functions. The learning rate is kept varying with number of epochs. At beginning it is set at peak and gradually decreased. Learning rate is set to 30% of the number of epochs. By training for 10 epochs 70% accuracy was achieved but training for 10 more epochs accuracy achieved was 72.5%.

C. *Results*

The accuracy of the CNN used for image classification over CIFAR-100 is 73%. Figure 1 shows the graph of accuracy versus number of epochs. Figure 2 shows the training and validation loss graph. Figure 3 shows the learning rate versus batch number.

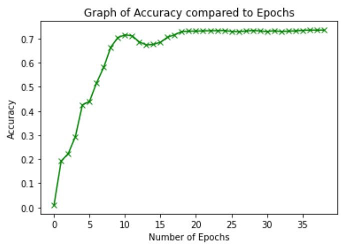

Figure 1: Accuracy Graph vs. Epochs

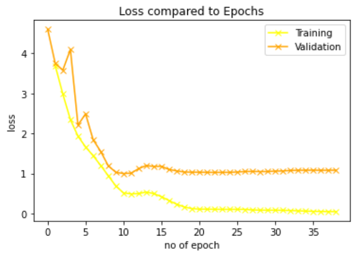

Figure 2: Loss versus Epochs

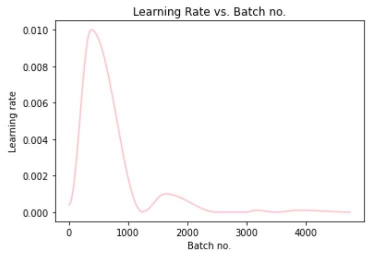

Figure 3: Learning Rate versus Batch Number

Sample test image predictions for three test cases are. Figure 4 has object apple and got classified as an apple. Figure 5 has object aquarium fish and got classified as an aquarium fish and the third image is that of a baby and got classified as a bed which is shown in figure 6.

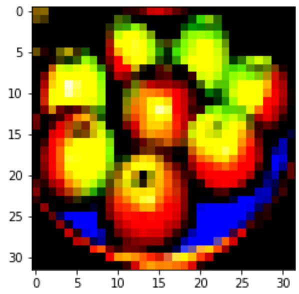
Figure 4: Image of an Apple classified as an apple

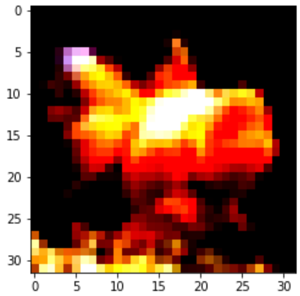
Figure 5: Image of an Aquarium Fish classified as an Aquarium Fish

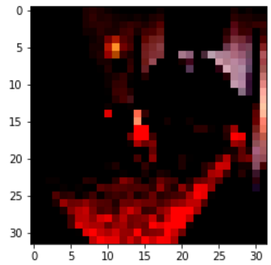
Figure 6: Image of a baby classified as a bed

IV. Conclusion

The CNN developed results in an accuracy of 73% on the CIFAR-100 dataset. This accuracy can be improved by changing the structure of the model or making use of pre-trained model or by changing the activation function used or the learning rate or the optimization function or the number of epochs.

References

[1] Bakhshi, A., Chalup, S. and Noman, N., 2020. Fast evolution of cnn architecture for image classification. In *Deep Neural Evolution* (pp. 209-229). Springer, Singapore.

[2] Basha, S.S., Dubey, S.R., Pulabaigari, V. and Mukherjee, S., 2020. Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, *378*, pp.112-119.

[3] Sharma, N., Jain, V. and Mishra, A., 2018. An analysis of convolutional neural networks for image classification. *Procedia computer science*, *132*, pp.377-384.

[4] https://www.kaggle.com/c/ml2016-7-cifar-100

[5] https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb

[6] https://medium.com/jovianml/image-classification-with-cifar100-deep-learning-using-pytorch-9d9211a696e