OPEN ACCESS
International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-6 Issue-5, June 2017

Image Classification using Convolution Neural Network

Tapan Bhavsar, Bhavinkumar Gajjar

Abstract: Convolution neural network has been mostly used for image classification in machine learning and computer vision. In simple neural network, single layer's feature may not contain enough useful information to predict image class correctly [6]. Using a feed forward CNN, misclassification rate can be reduced by some additional layers that contain acceptable information to predict image class. Also gradient based learning algorithm can be improved to synthesize complex decision that classify high dimensional pattern such as object edges and shape. In this paper, we make effort to modify standard neural network to transfer more information layer to layer. Moreover, already learned CNN model with training images are used to extract features from multiple layers. In this experiment, MNIST and CIFAR 10 dataset have been used to classify random images in 10 different classes labelled airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. In the addition, GPU can train CNN faster without giving the preference to hardware.

Keyword: Convolution Neural Network, CIFAR 10, gradient based learning algorithm, Image classification, MNIST, machine learning

I. INTRODUCTION

The field of machine learning has grown exponentially over the few decades, with the artificial neural network. We have studied most popular machine learning algorithm, Convolution Neural Networks, for image classification. Image classification is important topic in artificial vision system and has drawn significant amount of interest in recent times. Recently most researchers have started to develop efficient neural network in new trend of machine learning [2][3][4], which we call deep learning. In this paper, we want to implement CNN for image classification. Standard fully supervised convolution neural network will be used throughout the paper. However, when input dimensions in too large to use, the deep belief network takes long time to train. Sharing weights by convolution network solved this problem and improved the classification performance of various dataset. The convolution neural networks are inspired by animal visual cortex which is responsible for detecting light in receptive fields. These biologically inspired computational models are able to far exceed the performance of artificial intelligence in common machine learning tasks [1].

Manuscript published on 30 June 2017.
***** Correspondence Author (s)
Tapan Bhavsar, Student, Department of Electronics and Communication Engineering, Indus Institute of Technology and Engineering, Indus University, Ahmadabad (Gujarat), India.
E-mail: tapanbhavsar1601@gmail.com
Bhavinkumar Gajjar, Assistant Professor, Department of Electronics and Communication Engineering, Indus Institute of Technology and Engineering, Indus University, Ahmadabad (Gujarat), India. E-mail: bhavingajjar.ec@indusuni.ac.in

CNN are primarily used to solve difficult image driven pattern recognition tasks and with their precise yet simple architecture used.

However, when lots of images are coming to train model, it is too difficult problem to find feature from those. This is the one of the reasons that deep neural network is coming. Deep learning is form of machine learning that uses a model of computing that is very much inspired by the structure of brain. Deep learning comes in picture when the given neural network doesn't work well or its accuracy is unsatisfied. Deep learning is advance model in machine learning, which gives us more accurate output with compare to simple neural network.

Convolution neural network is feed-forward neural network, but if gradient based algorithms apply on it can increase accuracy and get proper predicted output. Normally in gradient based algorithm, cost function can be found by its equation and after that applying learning rate gradient decent will come out as output.

In this work standard dataset MNIST and CIFAR-10 used to test developed algorithm. MNIST dataset has images of 0-9 digits and each image is in gray scale. CIFAR-10 has images of 10 different classes which are based on animals and automobile machine.

The specific contributions of this paper are as follows: we have approached to use basics of convolution neural network and gradient based learning algorithm on two different dataset in section 2. Section 3 shows the experiment result of proposed architectural parameter change in CNN. Conclusion based on achieved result is discussed in section 4.

II. APPROACH USED

2.1. Convolution Neural Network (CNN)

In machine learning, there are mainly three types of learning supervised learning, unsupervised learning and reinforcement learning. CNN is mainly used in image processing domain of machine learning.

In the CNN, there are three types of layers except input and output layers has been created namely convolution layer (C1) (C2), pooling layer (S1)(S2) and full connected layer as shown in figure-1. First the neurons in input layer are based on input parameter. When input parameter increases, the accuracy of that model will also increase in simple neural network. This same thing works on CNN.

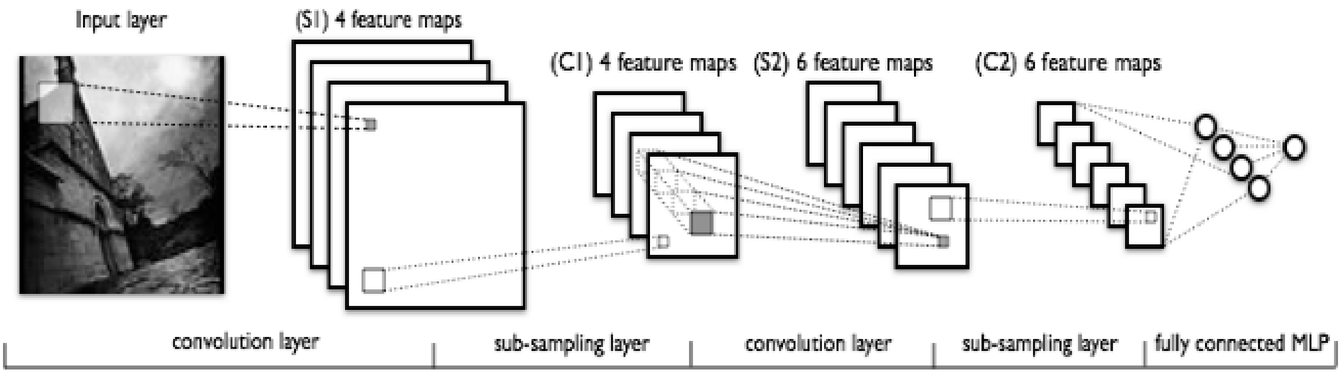

Figure 1 Simple architecture of CNN

The convolution layer's parameters consist of a set of learnable filter. Each filter contain features those can detect edges and shape of object. Every filter is small spatially, but extends through the full depth of the input volume. The new feature map can be obtained by first convolving input with learned filter and then applying an element-wise non-linear activation function.

The pooling layer aims to achieve shift-invariance by reducing the resolution of given map. Mainly, two types of pooling operation can be used namely max pooling and average pooling.

In the end dense layer has output prediction of classification using soft-max function *S(yi)*. Soft-max function is based on normalized exponential function [3]. That function is given below:

$$S(y_i) = \frac{e^{y_i}}{\sum e^{y_i}} \qquad (1)$$

Where $y_i$is predicted result of last layer or output layer of i^{th}neuron. From this equation we have made prediction range between 0 - 1. In image classification task, the different numbers of class has been given in chosen dataset. Ideally best prediction is 1.0 for single output node and probability of rest of output nodes are zero [8].

2.2. Gradient based learning algorithm

Gradient based learning algorithm can be also called by back propagation. In this method weights can be updated by using simple gradient decent and cross entropy or minimum loss of entropy algorithms [5].

Cross entropy $D(S(y_i),L_i)$ is loss of the given neural network. Using following function, we can find loss of that neural network at that iteration time.

$$D(S(y_i), L) = \sum L_i \log(S_i) \qquad (2)$$

where L_i is pre-defined i^{th} label in dataset and S_i is i^{th} neuron output of soft-max function. L_i is already labeled from dataset. It has range in between zero to number of classes. Cross entropy gives loss to train network with minimum loss of entropy. To minimize loss following function:

$$F = \frac{\sum D(S(y_i).L_i)}{N} \qquad (3)$$

Where N is numbers of sample in given dataset. Above equation is normalization of cross entropy to reduce loss.

Normally, pooling layer is inserted in between successive convolution layers. The main motto of pooling layer is, it reduces special size of representation to reduce the amount of parameters. It can also control over-fitting. Discarding pooling layers have also been found to be important in training good generative model.

Neural network has fully connected layer to convert two dimension neurons to one dimension dense layer. Neurons in fully connected layer have connection to all activations in the previous layer. Their activations can hence be computed with matrix multiplication with bias offset.

The general problem of minimizing loss with respect to a set of parameters is at the root of many issues in computer science [5]. Gradient decent is a first order iterative optimization algorithm to find local minima of the function. This is measured by gradient of loss function with respect to learning rate. Gradient decent function is given as following:

$$W_{(updated)} = W_{(previous)} - \alpha\Delta F \qquad (4)$$

$$B_{(updated)} = B_{(previous)} - \alpha\Delta F \qquad (5)$$

where α is learning rate of model and ΔF is minimum entropy loss and w is weight of CNN and B is bias of CNN[10]. With these procedures the parameters fluctuated around average trajectory, but usually convergence a second order method considerably fast with compare to gradient decent. Back propagation is by far most widely used neural network learning algorithm.

2.3. Dataset

Mostly newbie use MNIST dataset to understand how neural network can be trained. In MNIST dataset 0-9 digits images has been given as a binary image format or gray image format. Size of every image is 28x28x1 where this is represented as (width) x (height) x (depth). There are separable files of train data and labels and test data and labels. In train data 60000 images of digits and in test data 10000 images have been given with their considerable labels or class number. We have used MNIST dataset first to understand how neural network works. Because MNIST dataset has black and white 28x28x1 dimension images as we have explain above.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-6 Issue-5, June 2017

CIFAR 10 dataset contains 60000 colour images with 32x32x3 height, width and depth. There are 10 classes with 6000 images per each class. Dataset divide into 50000 training images and 10000 test images. Airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck are the 10 classes in the dataset.

III. EXPERIMENT

In this work, python library theano has been used to train neural network. This library is really good to train and study neural networks. In theano, if we want to train our neural network very fast using GPU, theano flags must be raisen to *gpu* or *cuda*. The codes for GPU and CPU are different from each other. In the addition, lasagne library is used in this experiment to make code of neural network easy using theano as backhand. Using lasagne library, it is easy to configure CNN because it gives freedom to work on layer and parameter modification.

First of all, training data with respect to labels is required to train CNN and before loading dataset we need to divide dataset in three parts: train data and train labels, validation data and validation labels, test data and test labels because training data is used to adjust the weights on the neural network. Validation set is just verified any increasing accuracy in train dataset and also used to minimized over-fitting. Test dataset is used to confirm the actual predictive power of the network. Normally, the ratio of training, validation and test dataset is respectively 5:1:1[9].

After dividing in three dataset we have standardize whole data. Standardization works as, data is subtracted by mean value of all data and after that the data is divided by maximum value of the data. Before apply for training, the data must be in 4 dimensions as respect to image sample number, number of channels, number of pixels in height, number of pixels in width. We have created three different model of neural network (NN)and take accuracy upto 100 epochs for each.

Following different types of neural network comparison for classification on MNIST dataset to classify 0-9 handwriting digits using 50000 images as training set, 10000 as validation set and 10000 images as testing set.4

Table 1 Effect on Accuracy by Changing Neural Network Model

	Simple Neural Network	Multilayer Neural Network	Convolution Neural Network
Layer 1	Soft-max[10]	Dense [800]	Conv1[3 x3 x 16]
Layer 2		Dense [800]	Pool1 [2 x2]
Layer 3		Soft-max[10]	Conv2[3 x3 x 32]
Layer 4			Pool2 [2 x 2]
Layer 5			Dense [256]
Layer 6			Soft-max[10]
Accuracy	92.61%	98.63%	98.72%
Time per epoch	0.26s	12s	33s

Using simple neural network we cannot classify some handwritten digit images because of low accuracy. In mnist dataset training, multilayer neural network and CNN accuracy is same but when testing on real time CNN more accurate with compare to multilayer neural network.

Following CNN comparison for classification on CIFAR-10 dataset to classify 10 different classes using 50000 images as training set, 10000 as validation set and 10000 images as testing set.Here, network-1 is learned using 3 x 3 kernel to detect basic shape, network-2 is pre-defined simple structure to learn. And in network-3 to detect high quality shape using double convolution layer.

Table 2 Effect on Accuracy by Changing Network Parameters

	Network-1	Network-2	Network-3
Layer 1	Conv1 [3 x 3 x 16]	Conv1 [5 x5 x 32]	Conv1 [5 x 5 x 16]
Layer 2	Pool1 [2 x 2]	Pool1 [2 x 2]	Conv2 [5 x 5 x 16]
Layer 3	Conv2 [3 x 3 x 32]	Conv2 [5 x 5 x32]	Pool1 [2 x 2]
Layer 4	pool2 [2 x 2]	Pool2 [2 x 2]	Conv3 [5 x 5 x 32]
Layer 5	Conv3 [3 x 3 x 32]	Conv3 [4 x 4 x 64]	Conv4 [5 x 5 x32]
Layer 6	Pool3 [2 x 2]	Pool3 [2 x 2]	Pool2 [2 x 2]
Layer 7	Conv4 [3 x 3 x64]	Dense [64]	Dense [512]
Layer 8	Pool4 [2 x 2]	Soft-max [10]	Dense [256]
Layer 9	Dense [256]		Soft-max [10]
Layer 10	Soft-max [10]		
Accuracy	64%	71%	68%
Time per epoch	75s	83s	92s

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

In above neural network models, every dropout consider as 0.5 and every mini iteration batch has shuffled 500 images from the dataset. Basically learning rate consider as 0.01 for CNN. Learning rate basically use in gradient decent to reach local minimum loss of the features. If learning we increase learning rate to 0.1 or above it. The output of local minima will be diverged and reduce accuracy of the model. If we decrease learning rate to 0.001 and below it gives same accuracy but with that number of epoch is increased and also increased time period to train model.

In this experiment, we see power of GPU, table shows GPU results. Its speed is 2 times more than speed of CPU. We have used GPU GTX 980 to train model. If the GPU is replaced by high features GPU, The time per epoch will be also decreased. I have considered i7 CPU because it really helps to speed up any neural network because of higher cores available.

IV. CONCLUSION

In this paper, we explored large convolution neural network models, train for image classification in a number of ways. We have compared every changed feature in CNN, how it effects on accuracy. Change features list in CNN as given: learning rate and number of epochs. In the end we have shown best results of our experiment.

We evaluate and compare our strategy to different model of CNNs on two benchmarks CIFAR-10 and MNIST. In MNIST training model, accuracy achieved is above 98% to predict any image of digits randomly. In CIFAR-10 training model, achieved accuracy is above 70 % to predict some random image of classes from CIFAR-10 dataset.

ACKNOWLEDGEMENTS

We are thankful to all officials from embedded department of Infinium Solutionz Pvt. Ltd, Ahmedabad, for giving valuable inputs in machine learning concepts.

REFERENCES

1. Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks."European conference on computer vision. Springer International Publishing, 2014.
2. Gu, Jiuxiang, et al. "Recent advances in convolutional neural networks."arXiv preprint arXiv:1512.07108 (2015).
3. Blot, Michael, Matthieu Cord, and Nicolas Thome. "Max-min convolutional neural networks for image classification."Image Processing (ICIP), 2016 IEEE International Conference on. IEEE, 2016.
4. Yim, Junho, et al. "Image classification using convolutional neural networks with multi-stage feature."Robot Intelligence Technology and Applications 3. Springer International Publishing, 2015. 587-594.
5. LeCun, Yann, et al. "Gradient-based learning applied to document recognition."Proceedings of the IEEE 86.11 (1998): 2278-2324.
6. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks."Advances in neural information processing systems. 2012.
7. Ciresan, Dan Claudiu, et al. "Flexible, high performance convolutional neural networks for image classification."Twenty-Second International Joint Conference on Artificial Intelligence. 2011.
8. Grave, Edouard, et al. "Efficient softmax approximation for GPUs."arXiv preprint arXiv:1609.04309 (2016).
9. Andreou, Panayiotis C., Chris Charalambous, and Spiros H. Martzoukos. "Robust artificial neural networks for pricing of European options."Computational Economics 27.2-3 (2006): 329-351.
10. Srivastava, Nitish, et al. "Dropout: A simple way to prevent neural networks from overfitting."The Journal of Machine Learning Research 15.1 (2014): 1929-1958.
11. Network, Back-Propagation. "Handwritten Digit Recognition with." (1989).

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)