Preprocessing for Image Classification by Convolutional Neural Networks

Kuntal Kumar Pal, Sudeep K. S.

***Abstract*— In recent times, the Convolutional Neural Networks have become the most powerful method for image classification. Various researchers have shown the importance of network architecture in achieving better performances by making changes in different layers of the network. Some have shown the importance of the neuron's activation by using various types of activation functions. But here we have shown the importance of preprocessing techniques for image classification using the CIFAR10 dataset and three variations of the Convolutional Neural Network. The results that we have achieved, clearly shows that the Zero Component Analysis(ZCA) outperforms both the Mean Normalization and Standardization techniques for all the three networks and thus it is the most important preprocessing technique for image classification with Convolutional Neural Networks.**

Keywords*— *Preprocessing, Convolutional Neural Network, Normalization, Standardization, ZCA, CIFAR10.

I. INTRODUCTION

The Convolutional Neural Network(CNN) was first introduced by Yann LeCun et al. [1]. Since then it has been applied in lots of area of research like Document Classification [2], Speech Recognition [3], Modelling Sentences [4], Facial Recognition[5][6], Recognition ofTraffic Signs[7] and Hand-written Digits[8][7]. However, Krizhevsky et al.[9] in 2012 was able to achieve remarkable improvement in object classification standards with top-5 test error rate of 15.3% on Imagenet dataset. Since then, the use of CNN in object classification gained popularity and various models have been proposed. Some of them being LeNet[2], Boltzmann machine[10] by Nair and Hinton, Multicolumn Deep Neural Network by Ciresan et al.[7], Dropconnect[8], GoogLeNet[11] and Fractional Max-pooling Network by Benjamin Graham[12].

As CNN was introduced there was the need for a labelled dataset which can be used as standard for image classification. Krizhevsky and Hinton in 2009 introduced the CIFAR10 dataset [13] which has challenged a lot of researchers since then. Other datasets have been introduced over time but CIFAR10 remains the most common among them, mainly for its small size of images and less number of classes.

It has not been possible for the architectures alone of the CNN models proposed, to achieve good classification accuracyon any dataset. The preprocessing techniques play a vital role in achieving the state of art on any dataset. Here we will provide a comparison based analysis of some of the well-known preprocessing techniques on three convolutional architectures.

Kuntal Kumar Pal, Computer Science and Engineering, National Institute of Technology, Calicut, Kerala, India (kuntal.octo@gmail.com)

Sudeep K.S, Computer Science and Engineering, National Institute of Technology, Calicut, Kerala, India (sudeep@nitc.ac.in)

II. METHODOLOGY

The Convolutional Neural Networks considered here is inspired from the works of Krizhevsky, Hinton and Sutskever[9]. The architecture of this network is similar to single layer neural network with some additional layers. The basic architecture includes a convolutional layer consisting of predefined number of filters of size 3x5x5, called the feature maps, which learn features from the input image. Here neurons are only connected to a small area (locally connected) having size of the filters. This is followed by a *pooling layer* of fixed size 2x2. For each of the models we have used max-pooling. This max pooling layer down samples its input along its width and height. This layer is followed by a fully-connected layer where all the neurons considers every activation in the previous layer. Each layers of the model learns its weights and biases (unknown parameters) using gradient descent in small mini-batches [14] of training samples.

The initial values of weights and biases of all the feature maps are assigned with random normal distribution. There are a number of hyper-parameters like learning rate(η), decay of learning rate, cost regularization constant(λ), minibatch size, number of neurons in fully-connected layer which plays a vital role in achieving good performance. The architectures of the three networks have been provided below.

A. Convolutional Neural Network 1

First Convolutional Network has 3 layers. A convolutional and pooling layer combined, a fully connected layer and a softmax layer with sigmoid function used as activation. We will represent this architecture as CNN-1 throughout the rest of the paper.

B. Convolutional Neural Network 2

Second Convolutional Network is similar to first (CNN-1) with one exception that is Rectified Linear Units (ReLUs)[9] used as activation function $f(x) = max(0, x)$. Nair and Hinton[10] used them to improve their Boltzmann Machine.The advantage of using ReLU is that it operates much faster thansigmoid activation as it refrains from calculating complex operations. Thus the training time reduces considerably. We will represent this architecture as CNN-2 throughout the rest of the paper.

C. Convolutional Neural Network 3

Third Convolutional Network has 4 layers. Two combined layers of convolutional and pooling, a fully connected layer and a softmax layer with ReLU used as activation function. We will represent this architecture as CNN-3 throughout the rest of the paper.

978-1-5090-0774-5/16/$31.00 © 2016 IEEE

III. DATASET

We have taken CIFAR10 dataset for our experiments. This is the contribution of Alex Krizhevsky and Geoffrey Hinton [13]. This dataset has 60000 colored images of size 32x32 divided into 10 classes namely airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck. Among these images, 50000 has already been segregated for training and remaining 10000, for testing purposes. We have used first 10000 samples from training set as validation set. 10 sample images from each of the 10 classes of the dataset have been shown in fig. 1.

Fig.1: 10 raw sample images from each of 10 different classes of CIFAR10 namely airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck

IV. PREPROCESSING

Raw data if applied to any classification methods does not produce good accuracy as can be verified from the results we achieved. Our goal is to show how much the accuracy varies with the application of some well-known preprocessing techniques on some simple convolutional networks. Some of the preprocessing techniques are mentioned below.

A. Mean Normalization

The mean along each of the features (dimensions of images) of training samples is computed and subtracted from each of the images. This way the whole training set is turned into organized data. Thus the brightness of the whole training set is normalized with respect to each dimension. This is done by(1).

$$X' = X - \mu. \qquad (1)$$

where X' is the normalized data, X is the original data and μ is the mean vector across all the features of X.

B. Standardization

The data is first mean normalized and then the standard deviation along each of the features of training samples is computed and divided. This makes the final data mean and variance normalized. Thus the raw data is organized with respect to both mean and variance of each of the dimension of the training set. This is standardization of input data X done by (2).

$$X' = (X - \mu) / \Sigma. \qquad (2)$$

where X' is the normalized data, μ is the mean vector across all the features and Σ is the standard deviation vector across all the features

C. Zero Component Analysis

Zero Component Analysis (ZCA) was first applied on training data by Krizhevsky [13] and later by Coates et al. [15]. The ZCA transformation makes the edges of the objects more prominent as can be seen from fig. 2. The convolutional layers detect various features through the feature maps based on these edges.

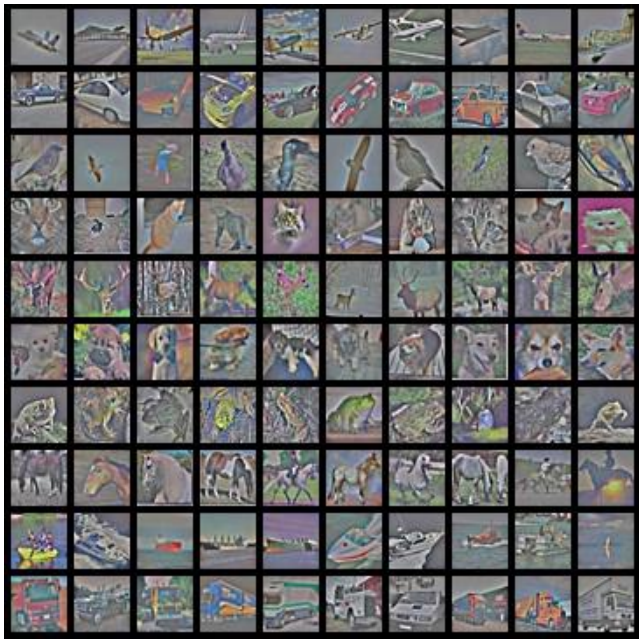

Fig.2:ZCA-whitened images of sample images of Fig. 1 with ε=0.1

The method of performing ZCA is mentioned in details. First the data(X) is size normalized by feature scaling with (3).

$$X' = X / 255. \qquad (3)$$

Then it is mean normalized by (1) and later the singular value decomposition of the covariance matrix of the mean normalized data is calculated. Finally, the whitening is performed by (4).

$$X_{ZCA} = U.\mathrm{diag}(1 / \sqrt{\mathrm{diag}(S) + \varepsilon}).U^T.X'. \qquad (4)$$

where diag(a) represents the diagonal matrix of given matrix a, U is the Eigen vector matrix and S is the Eigen value matrix of singular value decomposition of covariance matrix, U^T is the transpose of the Eigen vector matrix U, ε is the whitening coefficient. This method of preprocessing introduces another hyper-parameter whitening coefficient(ε).

V. TRAINING AND TESTING

The training and testing is done on GPU (GeForce820M) with python 2.7 and theano with cuda compilation tools (release 5.5, V5.5.0) on a machine having 8GBRAM and Intel core i3 processor

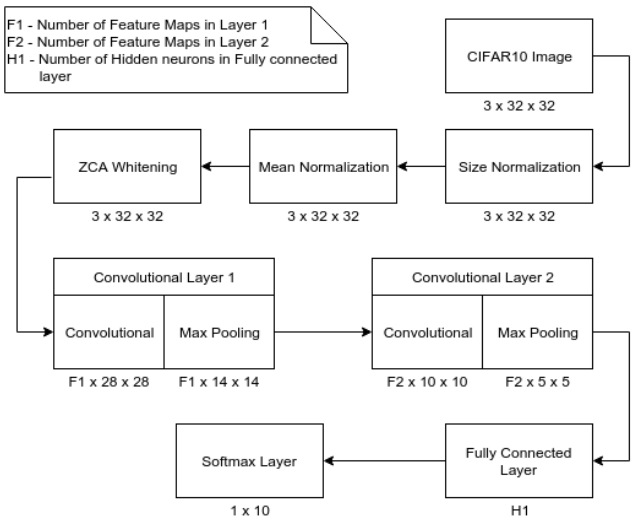

Fig.3: Block Diagram of training and testing process of CNN-3 with ZCA

The training and testing method on CNN-3 with ZCA preprocessing have been demonstrated by the block diagram in fig. 3. Here the output structure of the data has been shown under each block. During training it learns the unknown parameters and during testing it predicts the output class.

The training is done using randomized set of the 40000 data to generalize the process. All the data is segregated into 3x32x32 dimensions since they are colored (R, G, B) images of size 32x32. We have used a minibatch of size 50 for the minibatch stochastic gradient descent [14]. Due to this randomization and initial random parameter values selected for each layers (values for weights and biases), the output accuracy will vary with each execution. So we have taken the mean accuracies of three successful executions for generalization keeping the values of the hyper-parameters (learning rate, regularization cost, minibatch size) constant. The learning rate have been used with the decay factor of 0.1 after 10 unsuccessful increment of validation accuracies, for fine tuning the test accuracy.

We have tested the dataset on the three networks with a set of six different feature maps 16, 20, 40, 60, 80, 100 in the convolutional layer for 100 epochs with 100 neurons being used in the fully-connected network. For simplicity we have kept the number of feature maps same in both the convolutional layers of CNN-3.

VI. RESULTS

The results that we have achieved on the convolutional models with three preprocessing techniques have been shown in Table. 1. This represents the changes in test accuracies with the increase in feature maps of the convolutional layer with three preprocessing techniques on CNN-1, CNN-2, CNN-3.

These results have been shown in graphical representations for proper visualization in fig. 4 for CNN-1, fig. 5 for CNN-2 and fig. 6 for CNN-3. In each of the figures, red(bottom), grey(second from bottom), yellow(second from top) and blue(top) line represents the accuracies achieved using no preprocessing, mean normalization, standardization and ZCA normalization respectively

As can be seen from the graph in fig. 4, raw data on CNN-1 achieves accuracies in range 51-56%. When we used mean normalization, the accuracies increase negligibly to range 55-58%. Standardization however turns out to be better than the other two, achieving 63-66% accuracies. This is expected as standardization organizes the raw data more than mean normalization. The accuracies further increases if ZCA is applied, achieving accuracies in range 64-68%.

TABLE I:TEST ACCURACIES ON THREE CONVOLUTIONALARCHITECHTURES WITH DIFFERENT PREPROCESSING TECHNIQUES

Preprocessing	Features	CNN-1(%)	CNN-2(%)	CNN-3(%)
None	16	51.26	43.54	50.12
	20	52.14	43.19	52.79
	40	55.16	46.55	55.17
	60	55.64	47.54	55.63
	80	53.43	46.64	55.94
	100	54.47	49.06	55.84
Mean Normalization	16	55.01	57.72	61.76
	20	55.21	57.57	63.25
	40	56.97	56.50	65.91
	60	57.23	56.21	63.38
	80	56.46	55.99	62.55
	100	57.76	55.24	64.24
Standardization	16	63.29	64.75	65.75
	20	64.27	64.87	66.53
	40	65.59	63.86	68.36
	60	64.84	66.37	66.50
	80	63.96	66.74	65.67
	100	64.80	64.92	66.67
ZCA	16	67.03	68.32	67.78
	20	67.34	67.39	69.39
	40	66.98	67.51	72.06
	60	67.02	68.57	72.67
	80	65.64	67.89	71.61
	100	64.54	68.65	72.12

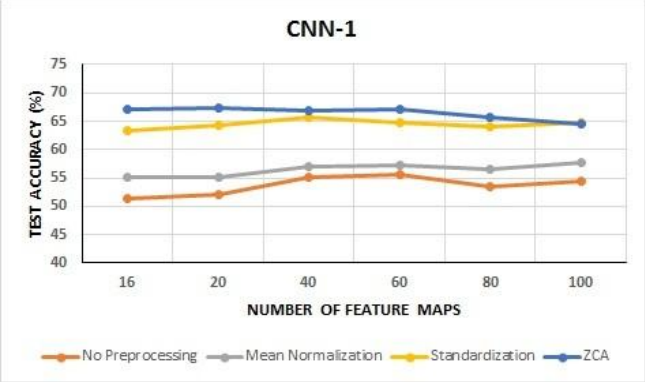

Fig.4:Test Accuracy vs Number of Convolutional layer Feature Maps for different preprocessing techniques for CNN-1 architecture

In fig. 5 we have shown how the preprocessing techniques affects the CNN-2 network. Here also we found that ZCA(67-69%) comfortably outperforms the accuracies achieved with no preprocessing(43-50%) and mean normalization technique (55-58%) but edges past the standardization(63-67%).

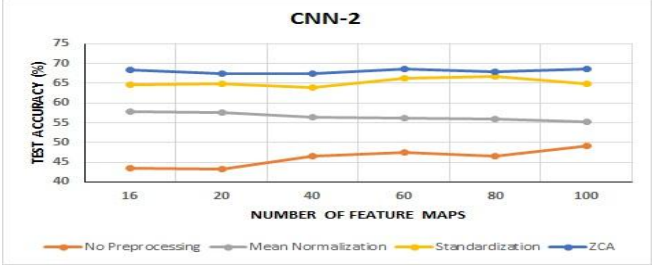

Fig.5:Test Accuracy vs Number of Convolutional layer Feature Maps for different preprocessing techniques for CNN-2 architecture.

From fig. 6 it can be clearly seen that for CNN-3 also, the best performance is achieved by ZCA (67-73%), outperforming accuracies provided by all the other techniques, without preprocessing (50-56%), mean normalization (61-66%) and standardization(65-69%).

VII. CONCLUSION

From the results that we have achieved, we conclude that preprocessing raw image data with ZCA achieves the best performance for convolutional neural network and the performance increases even more with the increase in convolutional layers in the architecture

We have also noticed that with the increase in one more convolutional layer, from CNN-2 to CNN-3, the performance increases for each preprocessing techniques as can be seen from the results in Table. 1

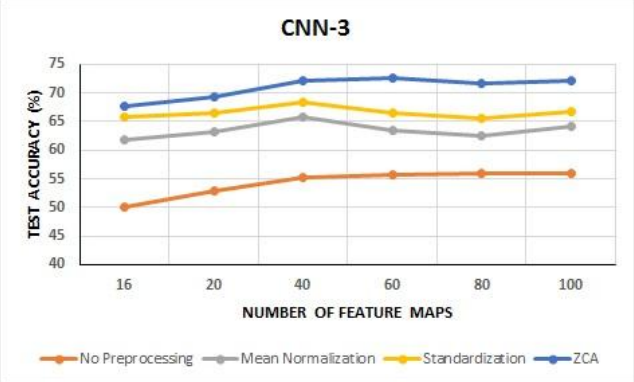

Fig.6:Test Accuracy vs Number of Convolutional layer Feature Maps for different preprocessing techniques for CNN-3 architecture.

REFERENCES

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural computation, vol. 1, no. 4, pp. 541–551, 1989.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[3] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012, pp. 4277–4280.

[4] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," arXiv preprint arXiv:1404.2188, 2014.

[5] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708.

[6] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in Advances in Neural Information Processing Systems, 2014, pp. 1988–1996.

[7] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, pp. 3642–3649.

[8] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in Proceedings of the 30th International Conference on Machine Learning (ICML-13), 2013, pp. 1058–1066.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

[10] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 807–814.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[12] B. Graham, "Fractional max-pooling," arXiv preprint arXiv:1412.6071, 2014.

[13] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[14] Michael. A. Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.

[15] A. Coates, A. Y. Ng, and H. Lee, "An analysis of singlelayer networks in unsupervised feature learning," in International conference on artificial intelligence and statistics, 2011, pp. 215–223