ISSN No. 0976-5697

Volume 8, No. 5, May-June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

HDNcfm: Handwritten Digit Recognition System Using No Combination of Feature Maps

Drashti Dobariya
Department of Computer Engineering
Institute of Technology, Nirma University
Gujarat, India

Dhwani Prajapati
Department of Computer Engineering
Institute of Technology, Nirma University
Gujarat, India

Sudeep Tanwar*
Department of Computer Engineering
Institute of Technology, Nirma University
Gujarat, India

Abstract: Convolutional Neural Network (CNN) has proved its significance as a robust method for classification of various image based applications such as face recognition and handwritten digit recognition. Traditional neural network's input feature maps are convolved along with the kernel, which aids in increasing the performance of the system. However, the problem of some feature lost arise in traditional neural network while combining the feature maps, and also the application does not tend to do well at a large scale networks. Hence, to address these issues, in this paper, we proposed, *HDNcfm: Handwritten Digit Recognition System using Combination of Feature Maps*, which tend to converge faster than Combination of Feature Map(CFM) method, and also upgrade its performance. In this method as the input features doesn't combined, the number of input feature and output feature can be same and therefore, this method can also be applied for relatively smaller dataset. Performance of the proposed approach has been evaluated using the parameters such as-convergence rate, and error rate. Results obtained clearly show the superior performance of the proposed scheme as compared to the traditional neural network based schemes.

Keywords: Handwritten Digit Recognition; NCFM; CNNs; Multilayer Perceptron model; Artificial neural network.

I. INTRODUCTION

NOWADAYS, Convolutional Neural Networks (CNNs) have become an emergent topic in machine learning. It has lead its importance in various different applications such as music and speech recognition, image processing, and sentence classification to name a few. We need to achieve high accuracy rate close to human brain's neural network, to gain high accuracy as close to 100% as possible and minimization of the error rate. Various techniques have been proposed over the years to achieve these goals. Neural networks without the convolution layers results into an error rate of 0.35% [1-3]. Introduction of elastic distortion [4] lead to expansion of the dataset resulting in 0.4% error rate. We can get error rate of 0.23% by using multi-column deep neural networks. Largest dataset can also be easily over-fitted with big network contains millions of parameters. Figure 1 show the revolution in neural network.

This paper introduced a replacement No combination of feature maps(NCFM) technique for abstraction of features. Every input feature map will generate an output feature map without combining, which lead to minimal data loss and achieve high precision rate. On analysis of results we can conclude that high performance can be achieved and that we can accomplish the acceptable precision rate on the dataset.

The organization of the paper is as follows: Section II describes the outline of all the related previous research papers then follows introduction of NCFM technique to enhance the performance of the CNN. Section III then describes NCFM technique thoroughly and discusses the benefits. Section IV describes the architectures of the CNN with NCFM and CFM, that are used throughout section V. Section VI designs 3 experiments to verify the validity and analysis of the Ncfm technique. Finally, Section VI presents our conclusions.

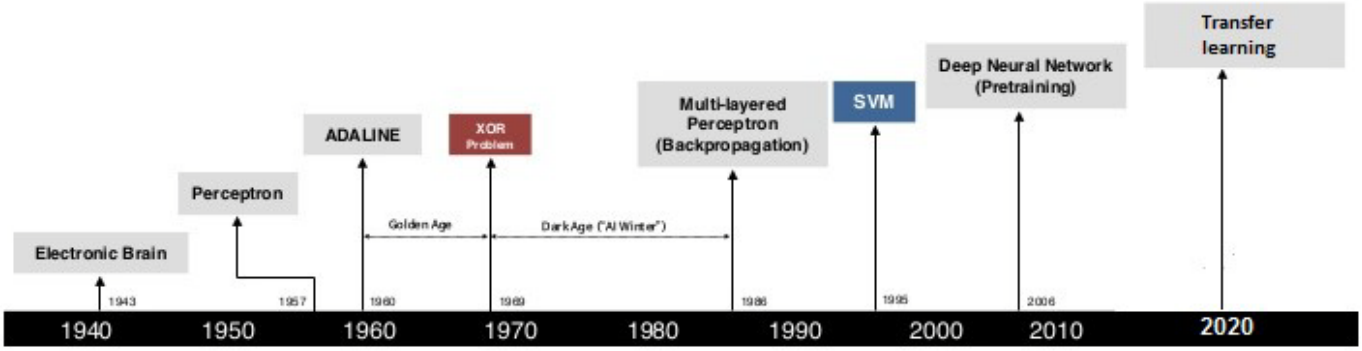

Fig. 1. Revolution in Neural Network

© 2015-19, IJARCS All Rights Reserved

TABLE I

LITERATURE SURVEY SUMMARY

Method	Accuracy	Description
Hand printed symbol recognition.[5]	97% overall.	Extract geometrical, topological and local measurements required to identify the character.
OCR for cursive handwriting. [6]	88.8% for exicon size 40,000.	To implement segmentation and recognition algorithms for cursive handwriting.
Recognition handwritten numerals based upon fuzzy model.[13-14]	95% for Hindi and 98.4% for English numerals overall.	The aim is to utilize the fuzzy technique to recognize handwritten numerals for Hindi and English numerals.
Combining decision multiple connectionist classifiers for Devanagari numeral recognition. [7]	89.6% overall.	To use a reliable and an efficient technique for classifying numerals.
Hill climbing algorithm for handwritten character recognition. [10]	93% for uppercase letters.	To implement hill climbing algorithm for selecting feature subset.
Optimization of feature selection for recognition of Arabic characters. [11]	88% for numbers and 70% for letters.	To apply a method of selecting the features in an optimized way.
Handwritten numeral recognition for six popular Indian scripts. [12]	99.56% for Devanagari, 98.99% for Bangla, 99.37% for Telugu, 98.40% for Oriya, 98.71% for Kannada and 98.51% for Tamil overall.	To find out the recognition rate for the six popular Indian scripts.

II. RELATED WORK

This section describe the previous work done by researchers in this domain with their merits and demerits. In handwritten character recognition area, an early novel attempt was made by Grimsdale in 1959. In the early sixties, an approach named analysis-by-synthesis technique by Eden(1968) was proposed. This method presented the fact that all handwritten characters can be recognized with the help of a definite number of schematic features. This laid the foundation for all the later approaches of character recognition.

K. Gaurav *et al.* [5], proposed different pre-processing techniques which were involved with the recognition of the handwritten digits and characters. The dataset collected by the author consisted of various colored documents, some with highly intensified background and some form documents. Various methods like normalization, binarization, segmentation, morphological techniques were discussed in it. He further suggested that by combining all the above mentioned pre-processing techniques, we can achieve better precision results. Salvador *et al.*[6], suggested HMM model for unconstrained handwritten text recognition. Markov chain was used to model structural part of the system with the help of multilayer perceptron, for the estimation of the emission probabilities. Pre-processing such as removal of slope, slant, and the normalization of the images was done with the help of the supervised learning. It lead to development of a recognition system with high pre-processing and precision rate(based on ANN).

U. Palet*et al.*[7], suggested a modified model of quadratic classifier which was used for recognition of the off-line handwritten numeric values of six widespread Indian script. Anita Pal *et al.* [8], extracted features using Boundary tracing and fourier descriptors. Identification of the characters was done based on the compared features which distinguished the different characters. For achieving a high and better performance, an analysis for calculation of number of the hidden layer nodes have been done. Here, accuracy of 94% with minimal training time was achieved. J. Pradeep *et al.* [9] proposed a diagonal feature extraction method based on ANN model. The comparison of the two methods- diagonal feature extraction, and the traditional one i.e., vertical and horizontal feature extraction was done on two approaches with 54 and 69 feature respectively. The diagonal feature extraction method yield precision rate of 97.8% and 98.5% for 54 and 9 feature respectively.

A. Brakensiek*et al.* [10], introduced a system of recognition of off-line cursive handwriting that was predicated using discrete and hybrid modeling techniques of Hidden Markov Models (HMM). Experiments included a discrete and two totally different hybrid approaches, that included semi-continuous structures, and discrete structure, were compared. System was developed using a segmentation free approach. It has been found that performance, and recognition rate of a hybrid modeling technique for HMMs can be improved by comparing neural vector quantizer and discrete HMMs. R. Bajaj *et al.* [11], used three kinds of features for classification of Devanagari Numerals, first, the moment features, second, descriptive component features and finally, density features. Multi-classifier connection architecture was used for enhancing the recognition predictability and achieved 89.6% accuracy for handwritten Devanagari numerals.

Sandhya Aroral *et al.*[12], combined intersection, chain code histogram, straight line, and shadow feature features for recognition of the Devanagari characters using 4900 samples as training set. The features were calculated after the segmentation of the image, except the shadow features which were computed globally. The final results concluded

© 2015-19, IJARCS All Rights Reserved

the precision rate of 92.80%. L T. Som*et al.* [15], proposed a fuzzy membership function for handwritten character recognition. First of all, an image was normalized to 20 pixels. Then from every ten pictures of each character, an fused image was created. With the help of vertical and horizontal projection of each character, a bounding box was set and image was then resized(10 X 10 size). Finally, on the basis of comparison of the test cases with the fused image, the characters were classified.

Renata F. *et al.* [16], have proposed off-line handwritten digit recognition using support vector machines. He concluded that SVM performs better than the Multilayer perceptron classifier model. They carried out experiments on NIST SD19 dataset [17]. Yoshimasa Kimura *et al.* [18] proposed a method for selection of features with the help of Genetic Algorithms. The genes which have recognition rate greater than pre-determined threshold of the parent gene and thereby gradually decreases the number of features used in the recognition process.

Nafiz Arical *et al.* [19], proposed a technique which can be used with minimum number of preprocessing operations. He further proposed a powerful segmentation algorithm which utilizes character boundaries, slant angle, local maxima and minima, stroke height and breadth, higher and lower baselines, and descenders and ascenders for the search of the optimum segmentation path which leads to decrease in over-segmentation. HMM was used to estimate model parameters as well as global and space parameters. M. Hanmandlu *et al.* [20] used exponential membership function for their fuzzy model for representation of Hindi and English numerals. The recognition was based of the modification of the exponential membership function best fit to the fuzzy sets which were derived from normalized distances(features) with the help of box method. The precision rate of their recognition system was found to be 95% and 98.4% for hindi and english numerals respectively.

K. H. Aparna *et al.* [21], proposed a technique for constructing a handwritten Tamil character recognition by execution of sequence of strokes. Shape-based representation of each stroke was used to represent a

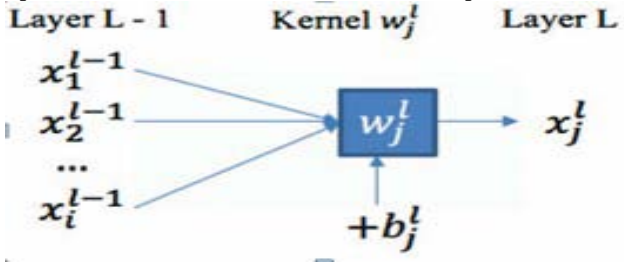

Fig. 2. A single neuron

stroke as a string of shape features. Unknown stroke was identified by comparing it with a database of strokes using a string matching procedure by using this representation.

A. *Motivation*

There are many applications that needs to be digitized but because of handwritten digit we can't able to make it digitized. Although CNN alone is not appropriate model to identify handwritten digit recognizer, because of it has nature of free flowing, overlapping digit, variable aspect ratio and noisy characteristics. Second problem is that it is not always of same size, thickness, orientation and position relative to the margin. So better option is to combine CNN with Ncfm to recognize the handwritten digits.

B. *Research Contributions*

Research contributions of the paper are as follows:

- We have proposed CNN with NCFM technique to enhance precision rate and speeding up data convergence.
- We have compared the CFM(Combination of feature maps) method with NCFM and conclude that NCFM converges quicker than CFM, and also NCFM performs higher than CFM with less convolutional filters.

III. NO COMBINATION OF FEATURE MAPS

This approach is divided into four parts. Part A gives description of the feed forward CNN, part B describes back-propagation in detail while part C shows about the application of the NCFM into the neural network layers and then discuss its benefits.

A. *Feed Forward CNN*

The description of the NCFM technique starts with description of a single CNN layer as shown in figure 2. Where w_j^l is convolutional computational unit and $b^l{}_j$ is a bias value, combines all the feature maps and gives an output, as indicated by equation 1:

$$x_j^l = f(u_j^l), u_j^l = (W_j^I)^T X^{l-1} + b_j^l = \sum w_j^l x_j^{l-1} + b_j^l \quad (1)$$

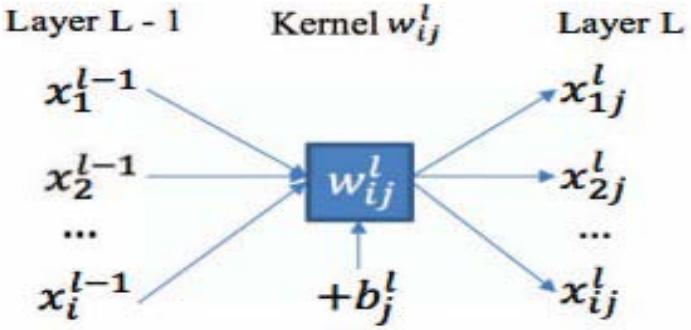

Fig. 3. Neuron example without combining inputs

Where*f* denotes the activation function while *b* and *w*are parameters of neuron. Figure 3 show example when one does not combine feature maps. Here, each input x_i^{l-1} gets convolved with w_{ij}^l to produce output denoted by $x^l{}_{ij}$

$$x_{ij}^l = f(u_{ij}^l), u_{ij}^l = w_{ij}^l x_j^{l-1} + b_j^l \quad (2)$$

Here we are using MNIST dataset for images of numerals where our aim is to classify these images into 10 different classes. We have used Softmax Regression for this classification. The cost function for this regression model is as described in equation 3.

$$E = \frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} l\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_j^T x^{(i)}}}\right] \quad (3)$$

Where k are the total number of class label, m is total number of class training examples, θ is the feature of NN while $y^{(i)}$ is the class label.

© 2015-19, IJARCS All Rights Reserved

B. Back-PropagationIn CNN

Back-propagation pass is required in order to reduce the value of the cost function which can be calculated from the equation 3. It minimizes the value of *b* and *w* by ancient algorithms such as gradient descent and then again trains the neural network with the new value obtained. The modified value of *b* and *w*can be obtained by applying gradient descent with the following formulas:

$$w_{ij}^{l} = w_{ij}^{l} + \alpha \frac{\partial E}{\partial w_{ij}^{l}} \quad (4)$$

$$b_{j}^{l} = b_{j}^{l} + \alpha \frac{\partial E}{\partial b_{j}^{l}} \quad (5)$$

Where learning rate is denoted by α. Based on Equation 4 and 5, we update the existing value of $b^{l}{}_{j}$and that of $w_{ij}{}^{l}$. Then assuming we define, $\frac{\partial E}{\partial b_j^l}$ as δ_j^l, we get the following formula's:

$$\delta_j^l = \frac{\partial E}{\partial b_j^l} = \frac{\partial E}{\partial u_j^l}\frac{\partial u_j^l}{\partial b_j^l} = \frac{\partial E}{\partial u_j^l} \quad (6)$$

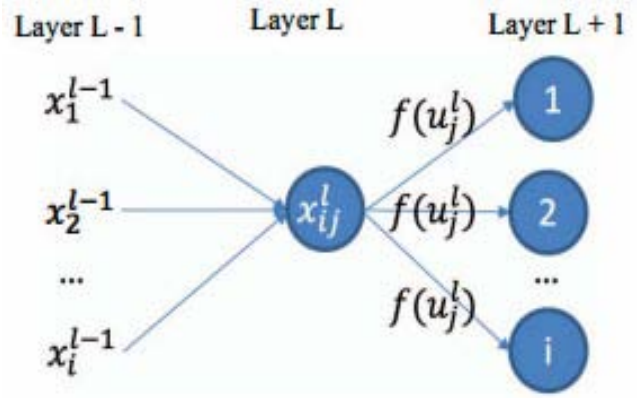

Fig. 4. Combination of neuron inputs and complete connection with the next layer.

From figure 3, $x^{l}{}_{ij}$combined all the input from the previous layer(L-1) and connected completely to the next layer(L+1). Thus, the value of $\delta_j{}^{l}$can be obtained from the high layer's $\delta_j{}^{l+1}$:

$$\delta_j^l = \frac{\partial E}{\partial u_j^l} = \sum_k \frac{\partial E}{\partial u_k^{l+1}}\frac{\partial u_k^{l+1}}{\partial u_j^l} = \sum w_{jk}^{l+1}\delta_k^{l+1} f(u_j^l) \quad (7)$$

According to equation 4 and figure 3, we need to derivate the loss function(3) to get the updated value of $\delta_j{}^{l}$:

$$\frac{\partial E}{\partial w_{ij}^l} = \frac{\partial E}{\partial u_j^l}\frac{\partial u_j^l}{\partial w_{ij}^l} = \delta_j^l x_i^{l-1} \quad (8)$$

Where $u^{l}{}_{j}$can be referred from equation 1. If we do not combine the inputs then, we will get the same number of output neurons as that of the input neuron. The same concept is given in the figure 4. Thus, Equation 6 becomes

$$\frac{\partial E}{\partial b_j^l} = \sum_i \frac{\partial E}{\partial u_{ij}^l}\frac{\partial u_{ij}^l}{\partial b_j^l} = \sum_i \frac{\partial E}{\partial u_{ij}^l} = \sum_i \delta_{ij}^l \quad (9)$$

and Equation 7 becomes

$$\delta_{ij}^l = \frac{\partial E}{\partial u_{ij}^l} = \sum_k \frac{\partial E}{\partial u_k^{l+1}}\frac{\partial u_k^{l+1}}{\partial u_{ij}^l} = \sum_k w_{jk}^{l+1}\delta_k^{l+1} f'(u_{ij}^l) \quad (10)$$

and Equation 8 becomes

$$\frac{\partial E}{\partial w_{ij}^l} = \sum_i \frac{\partial E}{\partial u_{ij}^l}\frac{\partial u_{ij}^l}{\partial w_{ij}^l} = \sum_i \delta_{ij}^l x_i^{l-1} \quad (11)$$

We finally update the value of $w_{ij}{}^{l}$and $b^{l}{}_{j}$from the Equations 9, 10, and 11.

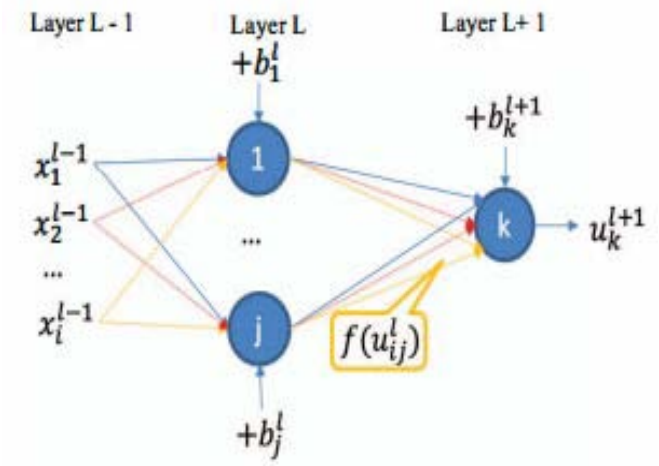

Fig. 5. Fully connecting with the next layer without combining inputs

C. Comparison of CFM with NCFM in CNN

There are two main advantages of using NCFM technique over CFM technique:

1. From equation 11, we can infer that learning will occur only for non-zero and positive values of *x* when applying CFM technique. However, we can use the NCFM technique when only one non-zero and positive value of input. This property ensures faster convergence.
2. While applying the NCFM technique, one input neuron gives$\prod_{i=1}^{n} c_i$ and$\sum_{i=1}^{c} c_i$ by using the CFM technique. Thus, NCFM method needs fewer convolutional filters than the CFM technique as it extracts more features in comparison.

IV. SYSTEM ARCHITECTURE AND COMPONENTS

This section deals with the architectures of CNN with Ncfm and Cfm. These are used widely throughout section V following this section. Figure 6 show the detailed architecture of CNN after applying Ncfm technique on convolutional layers. In the case of first convolutional layer, there are C_1 filters with 5 X 5 convolutional kernel each. From the input image C_1@28*X*28, C_1 feature maps with width 24 and height 24 are extracted from the filters. Once the extraction of features is done, the max pooling layer transforms the C1@24X24 input feature maps into C_1@12*X*12 output maps.

The second layer basically transforms the convolutional layer with C_2@5*X*5 filters, its every input feature map into C2@8X8 output feature maps. Once the extraction of max pooling layer is complete, we obtain C_i@4*X*4 output feature maps. All the fully-connection layers with respective neurons in each case are connected with all the previous

© 2015-19, IJARCS All Rights Reserved

layers' output features. Additionally, the first connection-layer is identical to the second fully-connection layer and the second fully-connection layer weight matrix is F_1XF_2.

The final layer also known as Softmax layer consists weight matrix of F_2X10 and 10 neurons. The output is obtained by estimation of prediction result. Figure 7 show applying Cfm technique on convolutional layers. Comparing Cfm and Ncfm, the configuration of Cfm is identical to Ncfm and the only difference between the two is the output size of the max pooling layer and the second convolutional layer.

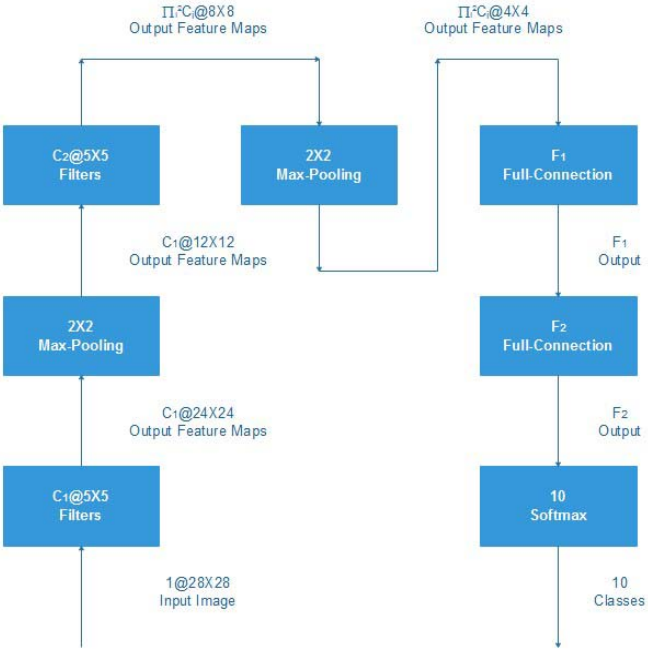

Fig. 6. Applying Ncfm technique on convolutional layers.

V. RESULT AND DISCUSSION

In order to affirm the validity of Ncfm methodology, we have selected three experiments to validate our approach. The first two subsections, depicts the experimental setup such as weight initialization and dataset. The third subsection, includes an experiment to verify fast convergence. The fourth subsection, shows a comparative study of both the techniques: Ncfm and Cfm, with the outcome that Ncfm method is better as compared to Cfm with just few filters. Finally, the last subsection describe the best performance outcome using MNIST dataset.

A. Data Augmentation and Dataset

We utilize the MNIST dataset for the following experiments. The MNIST dataset is a standard dataset containing images of 28X28 pixels with fragmented handwritten digits. Totally there are 70,000 examples from which 10,000 examples are for testing and the remaining 60,000 examples are training examples.

Label-preserving transformation can be used to enlarge the available dataset artificially in order to minimize over fitting on image data which can cause adverse effects on the final outcome. We enlist various unique forms of data augmentation:

- Elastic Distortions: An updated target location is given with reference to the previous position:

$$(u,v)^{Target} = (u,v)^{Ori} + (\Delta u, \Delta v) \quad (12)$$

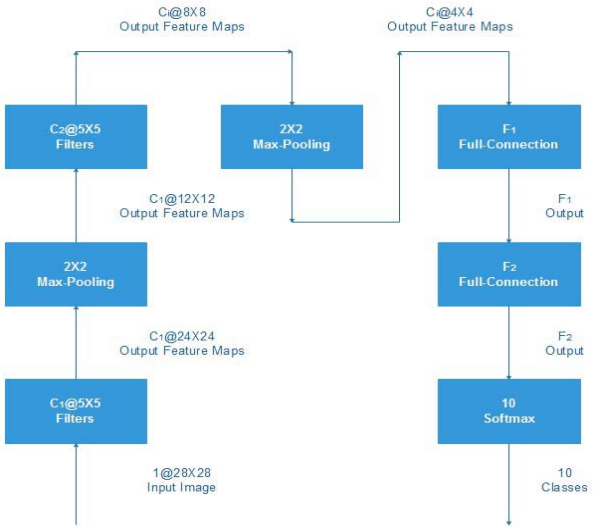

Fig. 7. Applying Cfm technique on convolutional layers.

where $|\Delta u| \leq 1.0$ and $|\Delta v| \leq 1.0$. Matrices Δu and Δv can be randomly initialized and later by using Gaussian filter we can reduce noise.

- Affine Transformation: We make use of scale and rotation transformation to expand our dataset. The scale ranges from −12% to 12% of origin size of pixel and the rotation ranges from -12° to 12°. This method is generally used to correct the deformations caused due to non-ideal angles.

B. Initializing Neural Network Weights

An algorithm that is used mostly while training a neural network is Mini-batch gradient descent. So, all the experiments utilize this algorithm. In this, uniform distribution or normal distribution is used to initialize the weight of the layers in a neural network. Convolutional Layers.

$$U \sim Normal(0,0.1) \quad (13)$$

Fully-connected Layers

$$U \sim \text{Uniform}[-b,b], b = \sqrt{(6.0/\sqrt{w+h})} \quad (14)$$

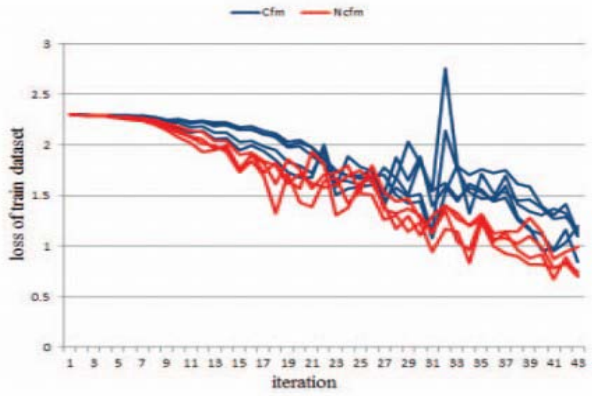

Fig. 8. Convergence Rate of Ncfm and Cfm for multiple filters.

Softmax Layers

$$U \sim Uniform[-0.01,0.01] \quad (15)$$

Where, w = width and h = height as per equation 3.

© 2015-19, IJARCS All Rights Reserved

C. *Convergence Speed*

Back-propagation algorithm was introduced back in 1970s. It is a common technique to train the neural networks. This algorithm is quite simple conceptually and very efficient computationally. It is a key necessity to achieve a fast convergence for back-propagation learning. Figure 8 describes, Ncfm technique can get more number of features that can shape the direction of extreme more efficiently and accurately so that we achieve faster convergence. We applied Ncfm methodology to the second convolutional layer and compared it with Cfm methodology to analyze fast convergence feature. The configuration of neural network is: C_1 = 10,C_2= 20,F_1 = 256,F_2 = 256. According to our observation, the weight scales are not unique of the first fully-connected layers. Figure 8 also show the loss function values as the number of iterations varies. We observe that the neural network with Ncfm converges faster than Cfm technique.

D. *Need Fewer Convolutional Layers*

Using Ncfm technique, more features are extracted as compared to Cfm technique to achieve a significant performance with less convolutional filters. The neural network configuration here is: C_1 = 10,F_1= 256 and the value of C_2 ranges from 2 to 11. From the figure 9, we come to a conclusion that if we use the Ncfm technique then the neural network performs with good consistency, whereas the error rate of Cfm increases with decrease in number of filters.

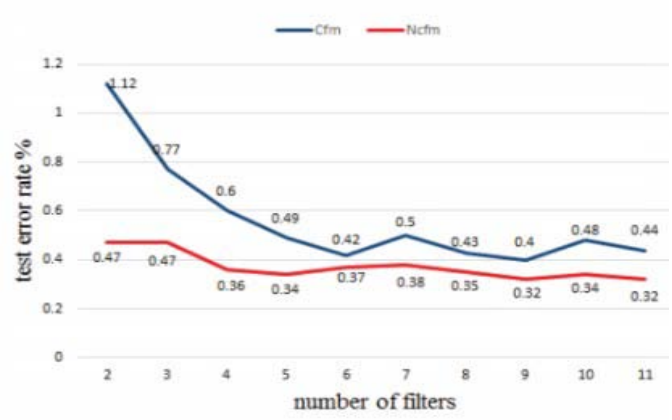

Fig. 9. Need of few convolutional filters

TABLE II
COMPARING NCFM WITH CFM (ERROR RATE %)

Trial	Ncfm	Cfm
1	0.24	0.43
2	0.32	0.33
3	0.27	0.37
4	0.28	0.36
5	0.27	0.36
Avg	0.276± 0.0258	0.37± 0.0329
Vote	0.19	0.29

E. *Comparison of Ncfm and Cfm*

In order to analogize the two techniques: Ncfm and Cfm, the input feature maps are directly augmented in the second convolutional layer of the neural network. The configuration of Ncfm technique is: C_1 = 10,C_2= 20,F_1 = 256,F_2 = 256, and the configuration of Cfm technique is:
C_1 = 10,C_2= 200,F_1 = 256,F_2 = 256.

The same number of feature maps are extracted to fully-connected layer by the convolutional layer. Table II show the Ncfm technique outperforms even without enforcing sparse combinations or response normalization techniques.

Wan L. *et al.* [5] obtained an error rate of 0.28%, when using a single model and an error rate of 0.21 % with voting that uses both the above mentioned techniques: enforcing sparse combinations or response normalization techniques. Hence, we obtained an optimum result without using these techniques as Ncfm technique performs efficiently.

VI. CONCLUSION AND FUTURE SCOPE

This paper outlines Ncfm methodology applied to hand written digits recognition. As every input learns and executes better with less number of convolutional filters, this methodology can increase convergence. Using this technique, we evidence state-of-the-art accuracy with 98.76% on MNIST database. Ncfm technique is applied to the convolutional layers and also give great significance to handwritten digits recognition. We proposed this approach in order to obtain higher abstract information and also to avoid loss of information. In future, we will apply the same technique on a larger available standard datasets to procure a significant change and optimize our training dataset because of different input features.

REFERENCES

[1] Kai Ding, Zhibiniu, ianwen Jin, Xinghua Zhu, A Comparative study of GABOR feature and gradient feature for handwritten chinese character recognition, *International Conference on Wavelet Analysis and Pattern Recognition*, pp. 1182-1186, Beijing, China, 2007.

[2] Pranob K Charles, V.Harish, M.Swathi, CH. Deepthi, "A Review on the Various Techniques used for Optical Character Recognition", *International Journal of Engineering Research and Applications*,2(1), pp. 659-662, 2012.

[3] Bhatia Neetu, Optical Character Recognition Techniques, *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(5), pp. 1219-1223, 2014.

[4] Iana M. origo and VenuGovindaraju, Offline Arabic Handwriting Recognition: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5), pp. 712-724, 2006.

[5] K. Gaurav and Bhatia P. K., Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition, *2 nd International Conference on Emerging Trends in Engineering & Management, ICETEM*, pp. 14-22, 2013.

[6] Salvador Espaa-Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4), pp. 767-779, 2011.

[7] U. Pal, T. Wakabayashi and F. Kimura, Handwritten numeral recognition of six popular scripts", *Ninth International conference on Document Analysis and Recognition ICDAR*, 07(2), pp.749-753, 2007.

© 2015-19, IJARCS All Rights Reserved

[8] Anita Pal &Dayashankar Singh, Handwritten English Character Recognition Using Neural, *Network International Journal of Computer Science & Communication*, 1(2), pp. 141-144, 2010.

[9] J. Pradeep, E. Srinivasan and S. Himavathi, Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network, *IEEE International Conference*, ISSN 978-1-4244-8679-3, pp. 364-368, 2011.

[10] A. Brakensiek, J. Rottland, A. Kosmala and J. Rigoll, Offline Handwriting Recognition using various Hybrid Modeling Techniques & Character N-Grams, *Available at http://irs.ub.rug.nl/dbi/4357a84695495.*

[11] Reena Bajaj, ipikaDey, and S. Chaudhury, Devnagari numeral recognition by combining decision of multiple connectionist classifiers, *Sadhana*, 27(1), pp.-59-72, 2002.

[12] Sandhya Arora, Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition, *IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur*, pp. 1-6, 2008.

[13] Mohammed Z. Khedher, Gheith A. Abandah, and Ahmed M. AlKhawaldeh, Optimizing Feature Selection for Recognizing Handwritten Arabic Characters, *proceedings of World Academy of Science Engineering and Technology*, 4, ISSN 1307-6884, pp. 1038-1041, 2005.

[14] SushreeSangitaPatnaik and Anup Kumar Panda, Particle Swarm Optimization and Bacterial Foraging Optimization Techniques for Optimal Current Harmonic Mitigation by Employing Active Power Filter", *Applied Computational Intelligence and Soft Computing*, Article ID 897127, 2012.

[15] T.Som, SumitSaha, "Handwritten Character Recognition Using Fuzzy Membership Function", *International Journal of Emerging Technologies in Sciences and Engineering*, 5(2), pp. 11-15, 2011.

[16] Renata F. P. Neves, Alberto N. G. opesFilho, Carlos A.B.Mello, CleberZanchettin, A SVM Based Off-Line Handwritten Digit Recognizer, *International conference on Systems, Man and Cybernetics*, pp. 510-515, 9-12 Oct, 2011.

[17] G. Pirlo and D. Impedovo, Fuzzy Zoning Based Classification for Handwritten Characters,*IEEE Transaction on pattern Recognition and Machine Intelligence*, 19(4), pp.780-785, 2011.

[18] Yoshimasa Kimura, "Feature Selection for Character Recognition Using Genetic Algorithm", *Fourth International Conference on Innovative Computing, Information and Control*, 978-0-7695-3873-0/09 2009.

[19] Nafiz Arica, and Fatos T. Yarman-Vural, "Optical Character Recognition for Cursive Handwriting", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6), pp. 801-113, 2002.

[20] M. Hanmandlu, O.V. Ramana Murthy, Fuzzy model based recognition of handwritten numerals, *pattern recognition*, 40, pp.1840-1854, 2007.

[21] K. H. Aparna,, Vidhya Subramanian, M. Kasirajan, G. Vijay Prakash, V. S. Chakravarthy, SriganeshMadhvanath, "Online Handwriting Recognition for Tamil", IWFHR, 2004, Proceedings, *Ninth International Workshop on Frontiers in Handwriting Recognition, Proceedings*, pp. 438-443, 2004, doi:10.1109/IWFHR.

© 2015-19, IJARCS All Rights Reserved