Validation of Random Dataset using an efficient CNN model trained on MNIST handwritten Dataset

Adhesh Garg*, Diwanshi Gupta*, Sanjay Saxena*, Parimi Praveen Sahadev*,

Department of Computer Science & Engineering, IIIT Bhubaneswar, Bhubaneswar, Odisha

Corresponding Author: sanjay@iiit-bh.ac.in

Abstract: Image processing and Deep learning are two zones of excessive awareness to researchers and scientists around the world. It is having multiple applications fields such as robotics, medicine, and security and surveillance. Deep Learning is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text. MNIST data set is having a huge number of handwritten text data set and it is frequently used for training, testing, and validation of CNN deep model. In this article, we have created an efficient model with multiple convolutions, relu and pooling layers. Which is tested on MNIST data set with 98.45 % accuracy? Further, this model is tested on similar kind of random image data set which gives significant results in terms of accuracy.

Keywords—Convolutional Neural Network (CNN), Deep Learning, MNIST Dataset

I. Introduction

Recognizing handwritten digits has been of great importance and has various uses in online handwriting recognition on 21st century smartphones and tablets, to extract postal zip-codes on mail or letterheads, processing bank check amounts, numeric entries in forms (like - tax forms) filled up by hand, or to automatically identify license plates[1] etc. There are different challenges faced while attempting to solve this problem. The handwritten digits are generally different in strokes, size, thickness, orientation, and distance from the margins thereby increasing the complexity for recognition.[2] Some attempts on recognizing the handwritten digits have been made using ANN [3], by combining SVM using rule-based reasoning [4], and by applying multi-column Deep Neural Networks for Image Classification [5]. Furthermore, Lottery Digit Recognition Based Multi - feature [6], is an alternate implementation of digit recognition compared to NN and SVM, where the problem is in the implementation of the number of algorithms needed to implement. User Independent Online Handwritten Digit Recognition [7] is a locally based recognition system that characterized a digit by its stroke, where the disadvantage is the problem of classifying the strokes.[8] Our goal was to train a model that could classify a digit based on its pattern using CNN to recognize a similar pattern of the handwritten digit. In this paper, we have proposed a novel CNN model to achieve the best performance on the handwritten digit recognition task from random images and the MNIST dataset. Images are of English handwritten digits taken from a variety of resources, normalized to 28x28 size to fit into the model. We designed our model using four convolutional layers[9] in which after every two convolutional layers there are two max-pooling layers to extract the features of the digits in the image.[10]

This paper is organized as follows. Section II starts by explaining in detail about Deep Learning and Convolutional Neural Network. Section III focuses on the implementation of the model and the way dataset is trained. Results obtained on the trained dataset and it's testing on random images is discussed at the detail in Section IV, and Section V includes the conclusions of this work. Further, in section VI references are given.

978-1-7281-1380-7/19/$31.00 ©2019 IEEE

II. Deep Learning Convolutional Neural Network Model

2.1 Deep learning

On various data representations Deep learning is a part of machine learning methods.[11] Algorithms are designed by observing the human brain structure and functions. Deep learning includes Supervised, Semi-supervised or unsupervised learning . Deep learning architectures which are applied to computer vision, speech recognition, natural language processing, audio processing, social network filtering, machine translation, bioinformatics, medical image analysis fields includes Deep neural networks, Deep belief networks, Recurrent neural networks.[12]. The results they have produced are comparable to and sometimes even better than humans.[13]

2.2 Convolutional Neural network

Convolutional Neural Networks is like normal Neural Networks where data fed to the neural network is processed by convolution layer, they're created from neurons that have weights and biases which can be learned. Each somatic cell receives some inputs, performs a scalar product and optionally follows it with a non-linearity[14]. CNN is a series of layers, and each layer of a CNN transforms one activation to a different through a differentiable perform. Convolutional Layer, Pooling Layer, and Fully-Connected Layer are three main layers used to design CNN architecture. We will perform these layers in series to make a full CNN design. Unit pictures are input for CNN, the specific assumption done by CNN architecture allow us to encode some properties into architecture. These then build the forward perform additional economical to implement and immensely cut back the number of parameters within the network[15]

2.2.1 Convolutional (CONV) layer:

The framework of the Convolutional Layer accommodates numerous filters that learn and update themselves to reach the result. Every filter is a tiny spatial arrangement. During the pass, move each filter across input volume and compute dot products between filter and input[16]. The values of that particular filter is given in the form of two-dimensional activation map. When CNN sees a specific type of feature of the image, the filter learns to activate. A single two dimension Activation map is produced for each filter in every convolutional layer. On piling the activation maps against the depth of the neurons, one can achieve the result magnitude.

2.2.2 Spatial Arrangement:

The following three criterion determine the magnitude of the result: Depth, Stride, and Zero Padding.

1. Depth of a neural network is the number of layers identifying a specific feature in the input[17]. When the primary Convolutional Layer takes a raw image as input, then totally different somatic cells on the depth dimension could activate in presence of different edges. Set of neurons that are all looking at the same region of the input is depth column.[18]
2. Stride is used to move filters across pixels. If the stride is two, then two pixels are jumped by filters at once[19]. Larger the value of stride, smaller is the volume of output.
3. In zero padding, the input is padded with zeros all around the border to monitor the size of result.

2.2.3 ReLu(Rectified Linear Unit) layer: ReLu[20] refers to rectified linear unit. This layer is used to eliminate the negative values in a matrix, defined by function $f(x)=\max(0,x)$.

2.2.4 Pooling Layer: The Pooling layer is used to minimize the size of the matrix to make it easy for computation of future process. The general function used in pooling layer is max function. This function returns the maximum of elements of a filter.[21]

2.2.5 Fully Connected Layer: Each and every neuron of one layer is connected to all the neurons of the next layer. Such a layer is called a fully connected layer.

III. Implemented Model

We considered several models of Neural Network while making the model for the training of MNIST handwritten images. Complex Network, Convolutional neural network seems to be the best fit as it extracts features before feeding it to the fully connected layer. While implementing the model we optimized the accuracy by varying various parameters like Learning rate, Dropout and number of epochs. Figure 2 shows the architecture of the model.

We designed model with 4 convolutional layers, 32 filters of size 5x5 in first two and 64 filters of size 3x3 in the next two. We tried increasing the number of the filter by two in each layer ie. 32, 64, 128, 256 but we did not find any significant change in the accuracy. The image size is small so 32 and 64 filters do the job well. Firstly, we reduced the image in each convolution and two convolution layers were there, but the testing dataset was not giving good accuracy. Padding was kept the same for the process and we increased the layers to 4. There were 2 max-pooling layers each after 2 convolutional layers with strides of 2. They are reducing the image. The activation technique used after each convolutional network is Relu. There was a dropout of 0.25 after a pooling layer. After the features were extracted the final Matrix was flatten and fed as the input to the fully connected network. Activation technique used in the fully connected layer is Relu and Softmax with a dropout of 0.5. As the images are all grayscale then Relu is the best choice of the activation function.

We tried reducing the learning rate but the result was found optimal at 0.001. reducing the learning rate and increasing it was just reducing the validation accuracy. the model was divided into 2000 steps with a batch size of 128. Figure 1 shows the variation in accuracy with the variation in learning rate.

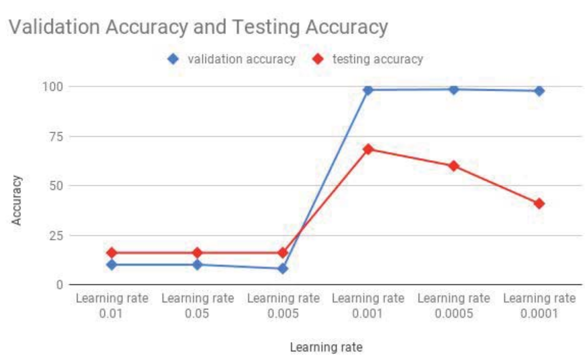


Fig 1: Validation and Testing Accuracy

We used the "Digit Recognizer" Dataset present on www.kaggle.com for training and validation process of the model. We trained on 40000 images and validated on 2000 images from the test images dataset. Then for the testing, we took several random images from the internet. We converted the images to grayscale and 28x28 pixels. Then the images were converted to a matrix and added to a CSV file used for testing.

We used Tensorflow libraries for coding and tested our observations on a public kernel on kaggle (https://www.kaggle.com/adheshgarg/testing-images). We used Kaggle GPU i.e. TESLA K80 that made our model very fast. RAM allocated to us for testing was 16GB. We have made a commit of our results at the mentioned link.

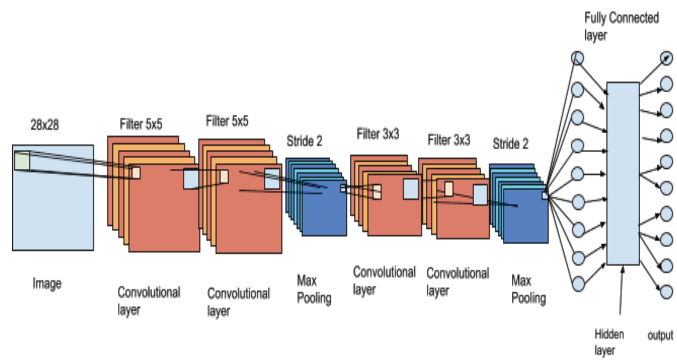
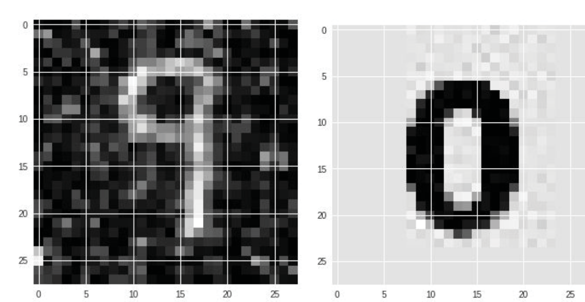


Fig 2: Architecture of the developed model

IV. Dataset and Results

MNIST "Digit Recognizer" dataset on kaggle has 42000 training images. We Trained our Model on 40000 images and Validated with 2000 images. The dataset has 10 classes of numbers from 0-9. We took 300 random images from the internet of different numbers handwritten and printed and tested them upon the optimized network. The validation accuracy for the set of 2000 image we received with 300 training epochs and 200 validation epochs is 98.45%. The system recognized about 95 percent of the 2000 images.



.

Fig. 3. Images used for testing

The testing accuracy of the random 300 images fed to the model with 200 epochs is 68.57%. While analyzing the data it can be observed that the model had a problem recognizing printed letters with sharp edges. The model was better with smooth-edged numbers. The total time for the training testing and validation of the system is 224.4 sec. Figure 3 shows 2 of the images used for testing the model.

The Model has proven to be a good model as identifying random images out of just small and only one type of training data. This testing accuracy can be further improved by increasing the training dataset with a different variety of images giving the system more features in the same class to work with.

V. CONCLUSION

In this paper, we have implemented a CNN model. That is the class of deep neural network with 4 convolutional layers. As we know that convolutional layers give a different kind of features of the image. Multiple convolution layers produce a diverse kind of topographies. We have used 4 convolutional layers which give the maximum accuracy. Further, 32 filters with the size 5x5 in first two layers and 64 filters of size 3x3 in the next two layers that gives the better accuracy size of the image data set is the reason of choosing 32 and 64 filters. In this work, we have downloaded the random image data set and validated on the implemented model and found the significant results as discussed in the previous section. In the future, our aim is to test this model with the diverse kind of data set. So that efficient usage of the CNN model is done.

VI. REFERENCES

[1] Z Selmi, M Ben Halima, A M. Alimi, “Deep Learning System for Automatic License Plate Detection and Recognition”, 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017, pp. 1132-1138, doi.org/10.1109/ICDAR.2017.187

[2] Y Bengio, A Courville, P Vincent, "Representation Learning: A Review and New Perspectives", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 35, issue 8, 2018, pp 1798–1828, doi.org/10.11,9/tpami.2013.50.

[3] J Schmidhuber, "Deep Learning in Neural Networks: An Overview", Neural Networks.Vol 61, 2015, pp 85–117, doi.org/10.1016/j.neunet.2014.09.003

[4] Y Bengio, Y LeCun, G Hinton, "Deep Learning". Nature 521, 2015, pp. 436–444, doi.org/10.1038/nature14539.

[5] D Ciresan, U Meier, J Schmidhuber, "Multi-column deep neural networks for image classification". 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp 3642–3649, doi.org/10.1109/cvpr.2012.6248110.

[6] L Deng, D Yu, "Deep Learning: Methods and Applications". Foundations and Trends in Signal Processing, 2014, Vol 7, pp 1–199, doi.org/10.561/2000000039.

[7] M Matusugu, M Katsuhiko, Y Mitari, Y Kaneda, "Subject independent facial expression recognition with robust face detection using a convolutional neural network", Neural Networks, 2013, Vol 16 issue 5, pp. 555–559, doi.org/10.1016/S0893-6080(03)00115-1.

[8] R Collobert, J Weston, "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning", Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 160–167, doi.org/10.1145/1390156.1390177

[9] P Le Callet, C Viard-Gaudin, D Barba, "A Convolutional Neural Network Approach for Objective Video Quality Assessment", IEEE Transactions on Neural Networks, 2006, Vol 17 Issue 5, pp. 1316–1327, doi.org/10.1109/TNN.2006.879766.

[10] Simard, Patrice, D Steinkraus, and J C. Platt. "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis." In ICDAR, vol. 3, pp. 958–962. 2003.

[11] LeCun, Yann, Bengio, Yoshua, "Convolutional networks for images, speech, and time series". In Arbib,

Michael A. The handbook of brain theory and neural networks, edition 2, 1995, pp. 276–278.

[12] D Yu, L Ma, H Lu, “Lottery Digit Recognition Based on Multi-features”, IEEE Systems and Information Engineering Design Symposium, 2007, pp 1-4, doi.org/10.1109/SIEDS.2007.4373986

[13] W Jian, Zheng-xing Sun, B Yuan, W Zheng, Wen-hui Xu, “User-Independent Online Handwritten Digit Recognition”, 2006 International Conference on Machine Learning and Cybernetics, 2006, pp 3359-3364, doi.org/10.1109/ICMLC.2006.258475

[14] K T Islam, G Mujtaba, R G Raj, H F Nweke, “Handwritten digits recognition with artificial neural network”, 2017 International Conference on Engineering Technology and Technopreneurship (ICE2T), Vol 3 Issue 8, 2017, pp 1-4, doi.org/10.1109/ICE2T.2017.8215993

[15] D. Gorgevik, D. Cakmakov, V. Radevski, “Handwritten digit recognition by combining support vector machines using rule-based reasoning”, Proceedings of the 23rd International Conference on Information Technology Interfaces ITI, 2001, pp. 139-144, doi.org/10.1109/ITI.2001.938010

[16] Dan Ciregan, Ueli Meier, Jürgen Schmidhuber, “Multi-column deep neural networks for image classification”, 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3642-3649, doi.org/10.1109/CVPR.2012.6248110

[17] Anuranjeeta, A., Saxena, S., Shukla, K., & Sharma, S, “Cellular Image Segmentation using

Morphological Operators and Extraction of Features for Quantitative Measurement. Biosciences”, Biotechnology Research Asia, 13(2), pp. 1101-1112. doi: 10.13005/bbra/2139

[18] Saxena S., Sharma S., & Sharma, N., “Parallel Image Processing Techniques, Benefits and Limitations. Research Journal of Applied Sciences”, Engineering and Technology, 12(2), 2016, pp. 223-238, doi: 10.19026/rjaset.12.2324

[19] Saxena, S., Sharma, N. Sharma, S., Singh, S., & Verma, A., “An Automated System for Atlas Based Multiple Organ Segmentation of Abdominal CT Images”. British Journal Of Mathematics & Computer Science, 12(1), 2016, pp. 1-14. doi: 10.9734/bjmcs/2016/20812

[20] Saxena S., Sharma S., Sharma N., Verma A., “An Intelligent Parallel System for Segmenting abdominal CT images using atlas-based allocation from Spine”, European Journal of Scientific Research, 137(3), 2016.

[21] Saxena S., Sharma N., Sharma S., “GPU Constructed Image Segmentation using First order Edge detection operators in CUDA Environment”, Journal of Chemical and Pharmaceutical Research, 8(2), 2016, pp. 379-387.