Journal of Current Research in Engineering and Science
JCRES
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume 6- Issue 1, Paper 16,January 2023

Various optimizers for training Convolutional neural network (CNN)

Nitin

Chandigarh University, Punjab

nitinkoundal09@gmail.com

Abstract-Convolutional Neural Networks (CNN) calculation has been generally utilized and has shown victories in different PC acknowledgment assignments. It additionally shows promising outcomes on many data models acknowledgment however we want to deal with it exactness, training time and accuracy that can be just overseen by the various optimizers of the CNN. Composed by hand digit affirmation is getting a colossal interest in the piece of PC acknowledgment. We will execute a better and precise way than manage see and anticipate genuinely formed numbers from zero to nine. A multilayer class of forward support considered called convolutional. The network enjoys upper hand upon other NeuralArtificial associations at utilizing and eliminating the features of data, working on the data on two dimenssional shapes with more sensitive degree of unvarying and accuracy to scaling, interpretation and various turns. LeNet planning was first invented by Yan LeCun in his research paper. The producers representation of LeNet was fundamentally revolved around numbers and symbols acknowledgment. This planning is fundamental and basic simplifying that for execution of CNN. We'll took the dataset that is MNIST dataset for affirmation, planning. Fundamental place for dataset is to arrange that translated numbers from zero to nine. We get a sum of seventy thousands pictures to testing and planning. Every digit tends to a twenty eight by twenty eight dim pixel powers for good output results. The numbers go through input layers of CNN LeNet and a while later into the mysterious layer that contains 2 game plans of pooling layers, convolutional and initiation. Finally it's arranged into the totally related layers and gives a classifier named as softmax to manage the number. Finally executionof this association with the help of keras significant learning automatic python library.

1. INTRODUCTION

Optimizers are basically an method or algorithms which is mainly used for change different attributes or values of a neural networks like its learning rate so we can get a result with less loss and great accuracy, The globe of human brain its half glob is mainly called asV1 which accommodate a lot no. of neurons with trillions of fellowship among them and they using to think about the digit or develop system that will helps to acquire among those. Towards the completion of neural framework uses the examples to normally develop steps for seeing written by hand counts. Total two kinds of neutrons open in mind are sigmoid neuron, perceptron. Now for figure the yield we'll introduce loads processing meaningful different commitments for the yield. Those neutron's gave a yield of zero or one else the weighing total beneath and above little margin esteem. Distinctive dynamic models are framed by various edges esteems and loads. During organization that principal layers of perceptron settle on straightforward choices made by reproducing the heap with data sources and this manner shows perceptron at subsequent layers could determine on remarkably larger complex choice among the perceptrons at the principal layers. Layer among them are the main layers achieved on logically greater complicated choices made with the primary layers. For learning reason we ought to constantly change the loads so the organization discovers the total and contrasts it and an edge worth of inclination. On the off chance that a little change in the loads alters the yield toward the path we need to continue then we can utilize little loads or we can take enormous loads for preparing,this technique resembles hit and preliminary which we use in tackling more serious level These information layers of organization contain information by encoding of neurons the qualities of the picture is getting from pixels input that is written by our hand in the form of digits. The information that we have prepared is gotten from the MNIST informational collection contain numerous twenty eight by twenty eight pixels pictures thus layers contain seven hundred eighty four information neutrons. In our organization the 2nd layer is the secret one, it also took the collected yield for the 1st layer after applying actuation capacity for identifyingthe example for information pictures. We'll explore various avenues regarding various quantity of the qualities of neutrons at secret one. After this the yield layers of that organization contain ten neutrons, each neutron whenever terminated gave an yield between zero to nine , that was the fundamental execution for organized neural of a picture. Similarly of deception organized neural with the little alteration covers underthe network that is Deep learning are similar named to be CNN like a more serious level for precision of arrangement. Deep organizations highlights were separated not only the whole area of information but also rather a few highlights are removed thr parts from the space. Profound organizations used Le-Net design that empowers organizations quick for preparation also getting good aftereffects for arrangement. Basically Keras which is a highlevel profound organizations uses python written library and equipped by utilizing the running through Theano or Tensor Flow. Organizations created this for the purpose that zeroing in basically to determining quick preparing. It Supports both intermittent and convolutional organizations also mix both of them too. Fundamental guidelines for that library was seclusion, that was perceived a method of grouping completely configurable libraries which could be consolidated along some limitations on them. Specifically streamlining agents, cost capacities, streamlining agents neural layers, actuation capacities are generally modules that you can join to shape another organization. The major point of keras is seclusion method of designing the layers. Henceforth using keras profound learning modules for carrying out the design..

2. LITERATURE SURVEY

2.1. Existing System

A few OCR frameworks have been created and every one of them shifts dependent on the picture preparing, highlight extraction, arrangement procedures utilized. The essential picture prepreparing rules are constant in all character acknowledgment programming. These means includes:

(i) They Convert RGB based picture to greyscale.
(ii) They Convert a grayscale picture to a binariy design.
(iii) They Identifying the Things and increase or decrease the size so the model can fit in which that model is prepared for.

The less or limited quantity of commotion gets neglected while changing over the image from grayscale to the binary design because of edge esteem sets. For include characterization procedures and extraction, different techniques has been suggested. That strategies is basically performed and proposed by SK Patel and J. Jha [1] incorporates essential advances should have been carried out in the OCR frameworks. A few AI strategies like closest neighbors, SVM, RNNs and HMMs have been analysed to get exact results. RNN(Recurrent neural network) is arisen as probably the best technique to the arrangement. It is One of the frameworks which has utilized ANN(Artificial Neural Networks) to perceive numbers from the pictures.[2] After that an advance has been taken in the paper by [3]. They looks at different neural organization method utilized in the optical character recognition(OCR) frameworks. Outcome gives a profound CNO(convolutional neural organization) (DCNN) gives better ability which can remove highlights. The CRNN methodology have been carried out fom the paper introduced by C. Yaowhich and B. Shi, X. Bai [4] utilizes LSTM for consecutive handling the picture with accordingly lessens conceivable wrong blunders.

2.2. Critical Analysis of Existing System

(i) The current frameworks experience issues in separating among commotion a word. The little spot that is really commotion additionally got distinguished as characters.

(ii) If the written text was on any governed line papers that lines would be stuided as some possible word.

(iii) Iincreased and decreased size of words are not again handled in current framework. There was a deficiency in pixels while increasing or decreasing the size and subsequently the character in word break are not arranged as expected by model, diminishing that exactness.

(iv) Framework that uses CRNN engineering yet doesn't utilizes appropriate translating calculations which causes etymological and logical mistakes.

2.3. Proposed System

(i) Ruled line removal

Governed line can be recognized utilizing Hough lines change introduced in a paper [5]. Binary picture go through some calculation. That calculation works on checking the pixels shaping Review on Literature "Handwritten Text Recognition" Swati Sinha1, Rajeshri Jadhav2 Yash Gurav3 , Priyanka Bhagat3 , Yash Gurav4 1Professor, DIT VIT, Mumbai, India 2,3,4Students DIT, VIT, Mumbai, IndiaInternational Journal Research in aScience,Engineering and Management Issue-2 Volume-3, , Feb-2020 | International Standard Serial Number : 2581-5792 617 straight lines, for various pictures take collect from the various source, its hard to set solitary edge esteem. The straightforward answer for increase or decrease the picture to a constant size prior to beginning picture handling on it. With the calculation viably recognize line vigilant edges identification should be utilized.

(ii) Processing on a segmented words

InterestsArea can distinguished utilizing a locale developing calculation should possible in the python used for findContours() strategy. In this manner, the ROI can be increased or decreased in size to a foreordained sizes. Subsequent to resizing, performing widening utilizing proper channel thickens of the words limit and fixe little breaks in the words. Thick word can viably expand proficiency of acknowledgment.

(iii) CRNN architecture

Venture utilizes convolutional recurrent neural network model for perceiving words. It is a LSTM design [7] for perceiving successive highlights the information picture. Convolutional recurrent neural network have been used for different settings including object acknowledgment, text acknowledgment and so on Profound CNNs are teachable and yield highlight maps that used for the different visually acknowledge tasks. RNN invaluable regarding logical agreement and for semantic rightness which will not be conceivable in the event that we interaction each character in turn. It works on successions of subjective height from one side to another side that is left to right side. The pictures were retailed and ascend to a similar tallness. It is inputted into the convolutional recurrent neural network engineering. CNN design incorporates pooling layer, convolutional layers, pooling and use RELU actuation work that yields component arrangement in the size of 32x256. RNN engineering took include arrangement in info, with each 32 time stamps. The component succession contain 256 highlights for each time steps, in RNN spreads pertinent data through these type of arrangement. Every segment of element vectors are relates to rectangular district of first picture. A RNN yields forecasts every one of 32 casings which is changed over in a likelihood for the eightyname groupings. Mark succession with the most noteworthy likelihood is yielded. The framework is prepared with the CTC misfortune work which empowers preparing of information and targets E. Unraveling calculation The interpreting of the expectations in the relating name arrangement is finished utilizing translating calculations like a Words bar Search, Vanilla shaft search. Is prospective by Harald scheidl in a paper [7], words pillar search authentic than the vanilla shaft search because of its accuracy and its requirement for word to utilized from word reference, accordingly diminishing the blunder rate. A WBS utilizes prefix tree to additional yield the right marks. Framework is consequently evolved utilizing the CRNN(convolutional recurrent neural network) engineering and Word Beam Search Decoding calculation.

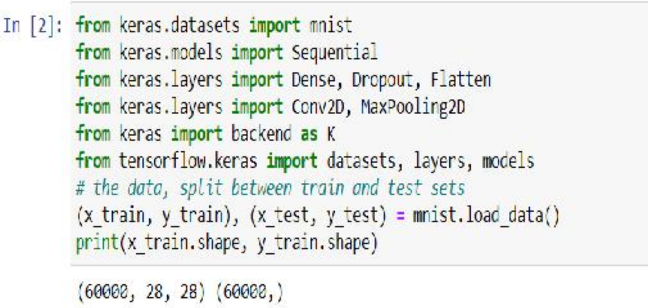
3. METHODOLOGY

In this chapter the implementation will be described and how each functionality is working. All tools that were used will be described as well.

3.1. STEPS FOR IMPLEMETATION

(i) Importing libraries and loading the dataset

To begin with, we’ll import every modules that we we’ll required for creating model so we take a library that is Keras now take datasets that is MNIST dataset. So we can undoubtedlyimporting the data and start work with it. The data set named as mnist.load_data() This will give us a preparation information, it is marks and furthermore testing information and the names.

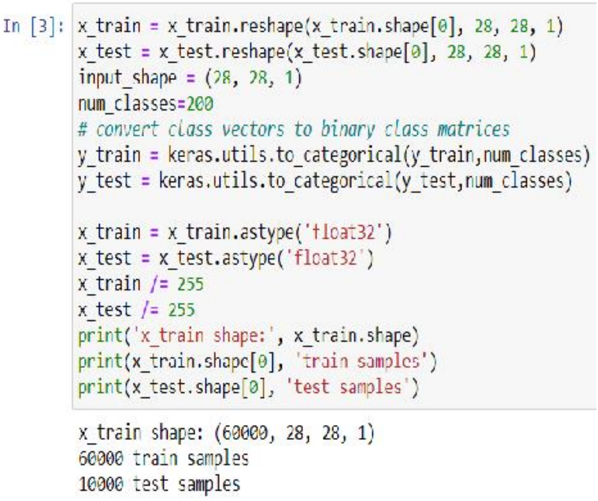


```
In [2]: from keras.datasets import mnist
        from keras.models import Sequential
        from keras.layers import Dense, Dropout, Flatten
        from keras.layers import Conv2D, MaxPooling2D
        from keras import backend as K
        from tensorflow.keras import datasets, layers, models
        # the data, split between train and test sets
        (x_train, y_train), (x_test, y_test) = mnist.load_data()
        print(x_train.shape, y_train.shape)

        (60000, 28, 28) (60000,)
```

Figure 1. Importing all the library required for our project

(ii) Preprocessing the data

The image info can’t be take care of the straight forwardly in the model that’s why we bring to a close certain cycle and their activities info to prepare it for the neural organization. The parts of preparation information is (60000.28.28). The convolutional neural network model will lack so we require more measurement for that we reshape the grid to shape (60000.28.28.1)

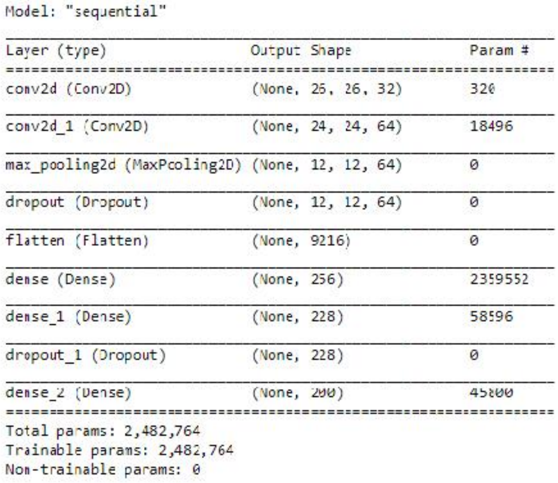


```
In [3]: x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
        x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)
        input_shape = (28, 28, 1)
        num_classes=200
        # convert class vectors to binary class matrices
        y_train = keras.utils.to_categorical(y_train,num_classes)
        y_test = keras.utils.to_categorical(y_test,num_classes)

        x_train = x_train.astype('float32')
        x_test = x_test.astype('float32')
        x_train /= 255
        x_test /= 255
        print('x_train shape:', x_train.shape)
        print(x_train.shape[0], 'train samples')
        print(x_test.shape[0], 'test samples')

        x_train shape: (60000, 28, 28, 1)
        60000 train samples
        10000 test samples
```

Figure 2. Preprocessing the data of our project

(iii) Creating a model

Presently we will make our CNN model in Python(Jupyter notebook) project. A CNNmodel for the most part compare of pooling layer and convolutional layer. It turns out better for info that are addressed as lattice structure, this is the motivation behind why CNN functions admirably for picturearrangement issues.Dropout layers are maily used for deactivating a portion of neurons and for keep in mind that preparation, it decreases offer fitting of the model. We will at that point gather the model to enhancer with the help of differernt optimizers like Adadelta, Adam, SGD, Adagrad, Nadam, RMS prop, Adamax.



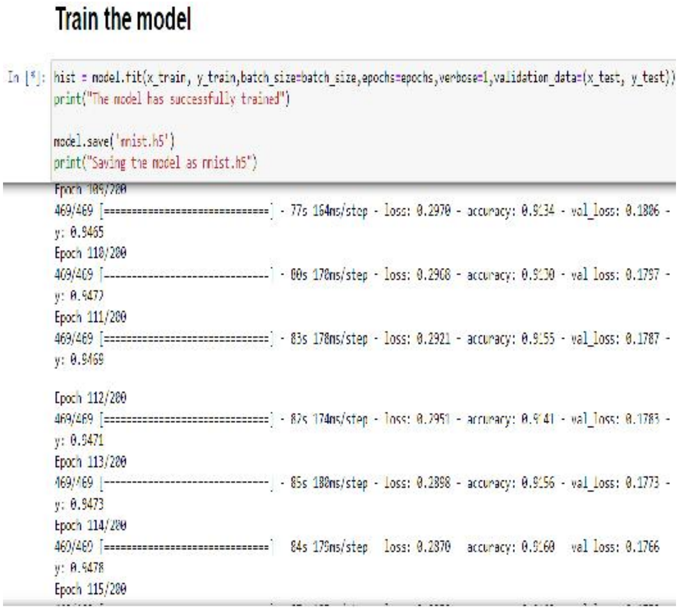
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 26, 26, 32)        320
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 24, 24, 64)        18496
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 12, 12, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 12, 12, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 9216)              0
_________________________________________________________________
dense (Dense)                (None, 256)               2359552
_________________________________________________________________
dense_1 (Dense)              (None, 228)               58596
_________________________________________________________________
dropout_1 (Dropout)          (None, 228)               0
_________________________________________________________________
dense_2 (Dense)              (None, 200)               45800
=================================================================
Total params: 2,482,764
Trainable params: 2,482,764
Non-trainable params: 0
```

Figure 3. Creating CNN model

(iv) Training our model

Model.fit() capacity of Keras library will begin withthe preparation of model. It returns with the preparation information, approval info, ages, and bunch size. It need some investment to create or prepare the model. Subsequent for preparing, we loads, save and

model interpretation in the 'mnist.h5'document



Train the model

```
In [*]: hist = model.fit(x_train, y_train,batch_size=batch_size,epochs=epochs,verbose=1,validation_data=(x_test, y_test))
        print("The model has successfully trained")

        model.save('mnist.h5')
        print("Saving the model as mnist.h5")
```

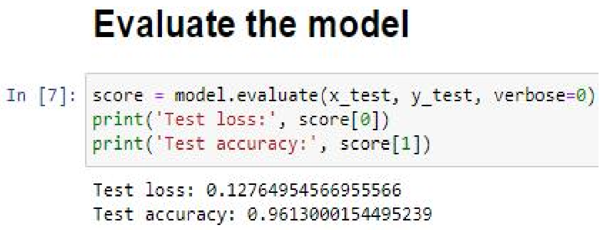
```
Epoch 109/200
469/469 [==============================] - 77s 164ms/step - loss: 0.2970 - accuracy: 0.9134 - val_loss: 0.1806 -
y: 0.9465
Epoch 110/200
469/469 [------------------------------] - 80s 170ms/step - loss: 0.2908 - accuracy: 0.9130 - val loss: 0.1797 -
y: 0.9477
Epoch 111/200
469/469 [==============================] - 83s 178ms/step - loss: 0.2921 - accuracy: 0.9155 - val_loss: 0.1787 -
y: 0.9469

Epoch 112/200
469/469 [==============================] - 82s 174ms/step - loss: 0.2951 - accuracy: 0.9141 - val_loss: 0.1783 -
y: 0.9471
Epoch 113/200
469/469 [------------------------------] - 85s 180ms/step - loss: 0.2898 - accuracy: 0.9156 - val_loss: 0.1773 -
y: 0.9473
Epoch 114/200
469/469 [==============================]   84s 179ms/step   loss: 0.2870   accuracy: 0.9160   val loss: 0.1766
y: 0.9478
Epoch 115/200
```

Figure 4. Training our model

(v) Evaluating our model

In MNIST dataset there are 10,000 pictures which will be utilized to see that how great our model works. The testing info was not similar with preparation of the info subsequently, it is new info for our model.

MNIST dataset is so good so we can get the accuracy in 99% so we are going to check which optimizer gives us the best result.



Evaluate the model

```
In [7]: score = model.evaluate(x_test, y_test, verbose=0)
        print('Test loss:', score[0])
        print('Test accuracy:', score[1])

        Test loss: 0.12764954566955566
        Test accuracy: 0.9613000154495239
```

Figure 5. Evaluating how good our model work.

Evaluating our model for all types of optimizers and then match their level of accuracy and how much time they have consumed.

4. RESULT ANALYSIS

After evaluation we understand that adam nadam and RMS prop shows similar results and adamax

SGD, adagrad, adadelta shows less favourable result.

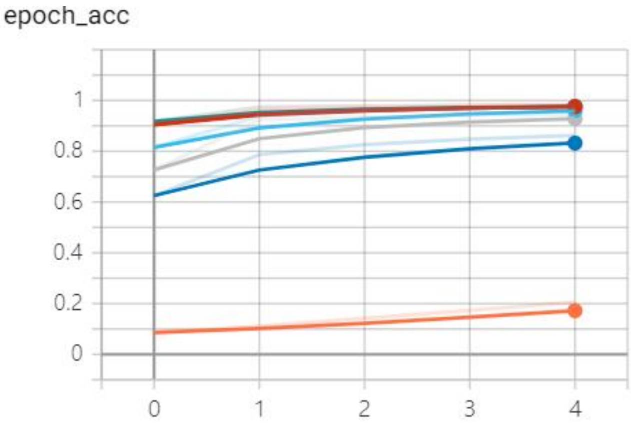
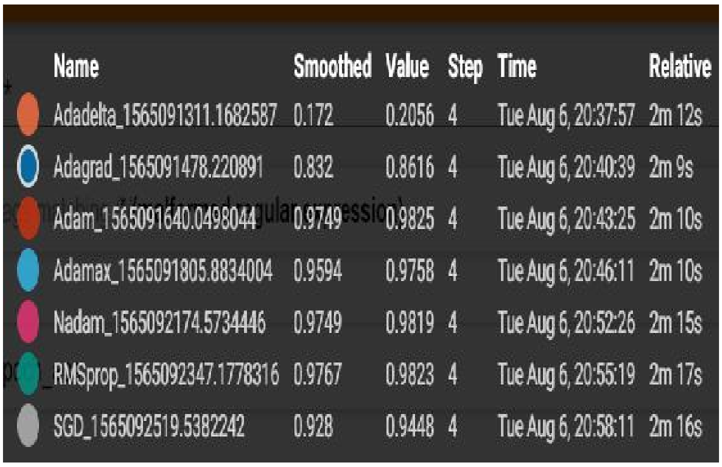


Figure 6. Evaluation shown in the form of graph



	Name	Smoothed	Value	Step	Time	Relative
●	Adadelta_1565091311.1682587	0.172	0.2056	4	Tue Aug 6, 20:37:57	2m 12s
○	Adagrad_1565091478.220891	0.832	0.8616	4	Tue Aug 6, 20:40:39	2m 9s
●	Adam_1565091640.0498044	0.9749	0.9825	4	Tue Aug 6, 20:43:25	2m 10s
●	Adamax_1565091805.8834004	0.9594	0.9758	4	Tue Aug 6, 20:46:11	2m 10s
●	Nadam_1565092174.5734446	0.9749	0.9819	4	Tue Aug 6, 20:52:26	2m 15s
●	RMSprop_1565092347.1778316	0.9767	0.9823	4	Tue Aug 6, 20:55:19	2m 17s
●	SGD_1565092519.5382242	0.928	0.9448	4	Tue Aug 6, 20:58:11	2m 16s

Figure 7. Evaluating the model by its accuracy value and time as shown in the figure.

5. CONCLUSION

Among different types of optimizer, I have confirmed RMSprop as well as Adam is really effective in terms ofConvolutional Neural Network sequentail model.

Adam = Nadam = RMSProp > Adamax > SGD > Adagrad >> Adadelta

Execution of RMSprop and Adam relies upon numerous elements for example less memory prerequisites, lessrun time and better exactness albeit in research paper this principally centered around improving precision rate for grouping. Before Artificial neurons would be wise to exactness however now the piece of PC sight majorly relies on profound learning which highlights for example convolutional neural organizations. Exploration means going on in this area and investigation have been created numerous types of Le-Net engineering LeNet:1, LeNet:4, Boosted Le-Net:4 and furthermore mixing helps numerous strategies like LeNet;4 with K- Nearest Neighbors Algorithm , yet seemingly forever Le-Net design is examined as a condition of the craft. Numerous techniques for example Tangent Distance Classifier is created for making use of Le-Net design. Fundamental point of this research paper manages 1 of the strategy that was very well may carried out , there were a few strategies where they should be possible and utilizing various systems for example MATLAB, OCTAVE. The piece of PC sight at man-made assuming important thinking process was to build an organization which is faster to every presentation gauge and gives result to a wide range of dataset that can be perceived and prepared.

6. FUTURE SCOPE

Expand this model for work on different dataset which increase the precision further by executing greater number of secret layers and epochs which detect custom hand written numbers use convolutional neural network with less number of layers to get good precision. Stable sized CNN's has been applied to many applications for example manually written digit citation , machine printed characters citation and online penmanship citation,then they can in addition to be valuable for stamp. More the devising models, more is the accuracy of the organizations. Unsupervised Artificial Intelligence was made simpler making use of CNN organizations, the portion of objects to come works possibly to carry out by Convolutional Neural Network's are packing or taking same result from more modest organizations with the help of improvement stunts , more equable parts of learning with the end aim shows that the information pictures doesn't gets twisted. This significant 3-Dimensional vision networks is an extension for investigates to create utilizing LeNet engineering and all the more organically concordant strategies, an expectation for future is Unsupervised Convolutional Neural Network 's.

REFERENCES

[1] S. R. Patel, and J. Jha, "Handwritten Character Recognition using Machine Learning Approach - A Survey" International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), 2015.

[2] K. T. Islam, G. Mujtaba, R.G. Raj, H. F. Nweke, "Handwritten Digits Recognition with Artificial Neural Network" International Conference on Engineering Technology and Technopreneurship (ICE2T), 2017.

[3] M. M. J. Ghosh, A. Y. Maghari "A comparative study on handwritten digit recognition using neural networks," International Conference on Promising Electronic Technologies, 2017.

[4] B. Shi, X Bai and C Yaowhich, "An End-to-End Trainable Neural Network for Imagebased Sequence Recognition and Its Application to Scene Text Recognition," 2015

[5] T. Dozat, Incorporating Nesterov Momentum into Adam, International Conference on Learning Representations, San Juan, Puerto Rico, 1(2016) 2013-2016.

[6] Mohammed A. A. Refaey, "Ruled Lines Detection and Removal in Grey Level Handwritten Image Documents",6th International Conference on Information and Communication Systems (ICICS), 2015.

[7] Zhang Da-kun, "An Extended Opening Operation and Its Application in Image Processing", International Conference on Multimedia and Information Technology, 2008.

[8] H. Scheidl, S. Fiel, R. Sablatnig, "Word Beam Search: A Connectionist Temporal Classification Decoding Algorithm",16th Int

[9] Yann LeCun, Leon Bottou,Yoshua Bengio and Patrick Haffner , paper on "Gradient Based Learning Applied to Document Recognition" , Proc of the IEEE , NOVEMBER 1998

[10] Y.LeCun,L.Jackel,L.bottom,A.brunot,C.Cortes,J.Denker ,H.Drucker,I.guyon,U.muller paper on “Comparision of Learning Algorithms for handwritten digit recognition”

[11] Y.LeCun,B.Baser,J.S.Denker,D.Henderson ,R.E.Howard,R.Hubbard,and L.D.Jackel , Handwritten digit recognition with a back- propogation network in D.Tourezky, Advances in Neural Information Processing Systems 2, Morgan Kaufman(1990)

[12] Corima Cartes and Vladimir Vapnik The Soft Margin Classifier , Machine Learning to appear(1995)

[13] Haider A.Alwzwazy, Hayder M. Albehadili, Younes S.Alwan ,Naz E Islam paper on “Handwritten Digit Recognition Using Convolutional Neural Networks” Vol 4 ,Issue 2,February 2016

[14] Xuan Yang , Jing Pu paper on “Mdig:Multi-digit Recognition using Convolutional Neural Network on Mobile

[15] Saeed Al-Mansoori paper on “Intelligent Digit Recognition using Artificial Neural Networks Vol 5, Issue 5, (Part-3) May 2015 , pp 46-5

[16] H. Huang, C. Wang and B. Dong, Nostalgic Adam: Weighing more of the past gradients when designing the adaptive learning rate, arXiv preprint arXiv:1805.07557, 2018

[17] S. Ruder, An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747, 2016.

[18] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations, San Diego, CA, USA, 2015.

[19] L. Luo, Y. Xiong, Y. Liu and X. Sun, Adaptive gradient methods with dynamic bound of learning rate, International Conference on Learning Representations, New Orleans, 2019.

[20] L. Bottou, F.E. Curtis and J. Nocedal, Optimization methods for large-scale machine learning, Siam Review, 60(2) (2018) 223–311.

[21] H. Hayashi, J. Koushik and G. Neubig, Eve: A gradient based optimization method with locally and globally adaptive learning rates, arXiv preprint arXiv:1611.01505, 2016

[22] M.C. Mukkamala and M. Hein, Variants of RMSProp and Adagrad with logarithmic regret bounds, in Proc. 34th International Conference on Machine Learning, Sydney, Australia, 70 (2017) 2545–2553.

[23] M. Zaheer, S. Reddi, D. Sachan, S. Kale and S. Kumar, Adaptive methods for nonconvex optimization, Advances in Neural Information Processing Systems, 31 (2018), pp. 9793–9803.

[24] I. Sutskever, J. Martens, G. Dahl and G. Hinton, On importance of initialization and momentum in deep learning, International Conference on Machine Learning, Atlanta, USA, 2013, pp. 1139–1147.

[25] D. Soham, M. Anirbit and U. Enayat, Convergence guarantees for RMSProp and Adam in non-convex optimization and an empirical comparison to Nesterov acceleration, arXiv preprint arXiv:1807.06766, 2018.

[26] N. Shazeer and M. Stern, Adafactor: Adaptive learning rates with sublinear memory cost, arXiv preprint arXiv:1804.04235, 2018