International Journal of Computer Engineering in Research Trends

Multidisciplinary, Open Access, Peer-Reviewed and fully refereed

Research Paper | Volume-8, Issue-5, 2021 Regular Edition | E-ISSN: 2349-7084

Comparative Study of Algorithms to Recognise Handwritten Digits

Anushka Sharma[1], Prateek Dhawan[2],Swarnalatha P

[1 &3]*Computer Science Department, VIT University, Vellore, Tamil Nadu, India*
[2*]*Computer Science: Bioinformatics Department, VIT University, Vellore, Tamil Nadu, India*

E-mail: anushka.sharma2019@vitstudent.ac.in, prateek.dhawan2019@vitstudent.ac.in, pswarnalatha@vit.ac.in

Available online at: http://www.ijcert.org

Received: 27/05/2021, Revised: 02/06/2021, Accepted: 07/06/2021, Published: 09/06/2021

Abstract: The paper shows how different estimations can be applied to test the exactness of the neural associations. We separate the display of the Back spread estimation with changing getting ready plans and the ensuing power term in feed-forward neural associations. In a relationship, we moreover analyze the essential backslide estimation, which makes a choice based on the value of an immediate blend of the features. In this paper, Neural Associations are used with an MNIST dataset of 70000 digits and 250 assorted creating styles. Logistic Regression is a measurable model that in its fundamental structure utilizes a strategic capacity to demonstrate a twofold reliant variable, albeit a lot more intricate expansions exist. In relapse examination, calculated relapse (or logit relapse) is assessing the boundaries of a strategic model (a type of paired relapse). This comparable examination shows the exactness of these computations in distinguishing physically composed digits, with Backpropagation unequivocally expecting close 95.06% of the test dataset when it was run multiple times, and the essential logistic regression correctly anticipating close 99% with a hidden layer and 92% without a hidden layer.

Keywords: Backpropagation Network, MNIST data set, TensorFlow, Logistic Regression Algorithm, Matlab, Accuracy, Confusion Matrix.

---

1. Introduction

Seeing composed by hand numbers is a problematic task in PC vision systems, which is huge for some new applications. Artificial intelligence and PC vision experts have commonly used it to facilitate judicious applications, for instance, modernized agent's check number arrangement. We made an Artificial Neural Association (ANN) and set it up to see cost and precision at each accentuation.

To separate the show of estimations to set up the limits of the neural association our paper uses certain computations like Back Propagation and Linear Regression. Physically composed digits are not seen by hard figuring due to their unpredictable presence, whether or not they are, the unpredictability is unnecessarily high. With fitting preparation, these estimations can achieve a critical level of accuracy. This examination targets doing that decisively.

1.1 Understanding the Neural Network, and the Back-propagation algorithm

A neuron in the brain resembles neural association centers. Burdens changed during the artificial intelligence connection while planning to interface each center to another center. The value of each center is settled by the value and strategy for the past center. Forward multiplication is the term for this method. The last yield of the association is related with the target yield and the heaps are adjusted to restrict the dismissing limit that explains whether the association expected the result adequately. Back multiplication is the name for this strategy. The neural association uses different layers to construct multifaceted design and precision. There are a couple of layers between the totally related neural association information, yield, and cover layers. The centers of each layer of a totally related neural association are related with the centers and layers when it. This was to explain the backpropagation computation.

1.2 The logistic regression algorithm

Continuing forward, we examine the determined backslide estimation. Vital backslide is a go-to strategy for matched portrayal (issues with two class regards). Here, it is basic to appreciate what matched direct classifiers are. An immediate classifier makes a portrayal decision for a given insight reliant upon the value of a straight mix of the discernment's features. In an "equal" straight classifier, the discernment is organized into one of two potential classes using an immediate cutoff in the data feature space. To grasp twofold classifiers better, we should in like manner fathom straight backslide, which is a sort of a fundamental of the equal classifier: Vital backslide is a simulated intelligence model for twofold portrayal, for instance sorting out some way to orchestrate data centers into one of two groupings. It's a straight model, in that the decision depends just upon the spot aftereffect of a weight vector with a segment vector. This infers the game plan cutoff can be tended to as a hyperplane. It's an extensively used model through its own effort, and the general plan of straight followed-by-sigmoid is an average subject in neural associations.

Through this paper, we aim to shed light on the basic algorithms used to recognize handwritten digits. We use this for beginners getting started in the field of machine learning and deep learning. We hope to outline the algorithms, their level of accuracy, it's cost and further work in obtaining more accurate results.

2. Related Work

The penmanship acknowledgment of characters has been around since the 1980s. The errand of manually written digit acknowledgment with a classifier is of incredible significance and application, for example, web-based penmanship acknowledgment on PC tablets, acknowledgment of postal codes in the mailing station for arranging mail, preparing of Bank checksums, mathematical sections available filled structures (for example tax documents), and so forth There are a few difficulties in taking care of this issue. Written by hand numerals are not generally a similar size, thickness, or direction and position comparative with the edges. Our objective was to carry out an example grouping technique to perceive the written by hand digits given in the MNIST dataset of transcribed digit pictures (0 through 9). The informational index utilized for our application comprises 300 preparing pictures and 300 test pictures and is a subset of the MNIST informational collection (initially comprising 60,000 preparing pictures and 10,000 test pictures). Each picture has a dim degree of 28 x 28 (0 to 255) named portrayal of a solitary one Digit. The overall issue we anticipated was the similitude between digits like 1 and 7, 5 and 6, 3 and 8, 9 and 8, and so on Likewise, individuals compose similar numbers from numerous points of view. The digit '1' is composed as '1', '1', '1' or '1'. Also, 7 can be composed as 7, 7, or 7.

The assortment in various individuals' composing likewise influences the arrangement and presence of the finger. Velappa Ganapathy and Kok Leong Liew proposed a strategy utilizing the first multiscale neural preparation with alterations to the information preparing vectors to acquire its benefit in preparing. Higher goal character pictures are proposed and afterward, particular thresholding utilizing the base dispersing procedure is performed to improve the exactness of character acknowledgment. A test system program (a graphical UI) is planned with the goal that the characters can be anyplace on clear paper. The outcomes show that with a moderate level of preparing ages, such techniques can accomplish transcribed exactnesses of at any rate 85% and more prominent English capital letters. and digits. Mathias M. Adankon, Mohamed Cheriet, the LS-SVM classifier, as other piece machines, gives helpless speculation when hyperparameters are not proficiently coordinated. The creators proposed a model determination system for the LS-SVM, which is a variation of the well-known SVM classifier. They made the choice of the model based on the exact mistake basis as indicated by the LOO method. They applied a calculation to a penmanship acknowledgment issue with promising outcomes. Standard and the LOO strategy accomplished a better, they presume that the short LS-SVM with model determination would be a fascinating classifier elective for the SVM in design acknowledgment frameworks.

3. Methodology

Back propagation Neural Network

Data Visualization: In the initial segment of ex4.m, the code will stack the information and show it on a 2-dimensional plot (Figure 1) by calling the capacity show Information. There are 5000 preparing models in ex3data1.mat, where each preparation model is a 20 pixel by 20 pixels grayscale picture of the digit. Every pixel is addressed by a coasting point number showing the grayscale power at that area. The 20 by 20 lattice of pixels is \unrolled" into a 400-dimensional vector. Every one of these preparation models turns into a solitary line in our information lattice X. This gives us a 5000 by 400 framework X where each line is a preparation model for a manually written digit picture.

Model Representation: The neural organization has 3 layers { an info layer, a secret layer, and a yield layer. Our data sources are pixel upsides of digit pictures. Since the pictures are of size 20 * 20, this gives us 400 information layer units (not including the additional inclination unit which consistently yields +1). The preparation information will be stacked into the factors X and y by the ex4.m script. You have been furnished with a bunch of organization boundaries (Theta(1); Theta(2)) effectively prepared. These are put away in ex4weights.mat and will be stacked by ex4.m into Theta1 and Theta2. The boundaries have measurements that are estimated for a neural organization with 25 units in the subsequent layer and 10 yield units (relating to the 10 digit classes).

Stepwise algorithm applied:

Step 1: Loading & Visualizing Data:

We'll start the program by first loading and visualizi ng the dataset. We will be working with a dataset that contains handwritten digits. 100 data points are selected at random to display.

Step 2: Loading the parameters:

We load some preinitialized neural network paramet ers. We load the weights into variables Theta1 and Theta2.

Step 3: Apply feed forward part of the neural network to compute the cost:

To the neural network, we'll first start by implementing the feed forward part of the neural network that returns the cost only. The code is computed in n Cost Function. m to return the cost. After implementing the feed forward to compute the cost, we can now verify that our implementation is correct by verifying whether we get the same cost as ours for the fixed debugging parameters. The implementation of the feed forward cost is applied *without* regularization first, so that it will be easier for us to debug.

Step 4: Implement the Regularization:

Once our cost function implementation is correct, we'll now continue to implement the regularization with the cost.

Step 5: Implement the Sigmoid Gradient:

Before we start implementing the neural network, we will first implement the gradient for the sigmoid function. The code is computed in the sigmoid Gradient's file.

Step 6: Initializing parameters:

Now we will be starting to implement a two layer neural network that classifies digits. We can now implement a function to initialize the weights of the neural network (randInitializeWeights.m)

Step 7: Implement Back propagation :

Once the cost matches up with ours, we'll proceed to implement *the back propagation algorithm* for the neural network. The code written in nn CostFunction.m is added to return the partial derivatives of the parameters.

Step 8: Implement Regularization:

Once the back propagation implementation is correct, we can nowcontinue to implement the regularization with the cost an d gradient.

Step 9: Training NN:

We have now carried out all the code important to prepare a neural organization. To prepare the neural organization, we will presently utilize "fmincg", which is a capacity which works likewise to "fminunc". Review that these high level enhancers can prepare our expense capacities proficiently as long as we give them slope calculations. costFunction is a capacity that takes in just a single contention (the neural organization boundaries)

Step 10: Visualizing Weights:

We can now "picture" what the neural organization is learning by showing the secret units to perceive what highlights they are catching in the information.

Implement predict:

After training the neural network, we can now use it to predict the labels. The "predict" function is implemented to use the neural network to predict the labels of the training set. This will let us compute the training set accuracy.

Logistic Regression:

ANN was utilized as a classifier to develop an order model. ANN is intended to prepare the information base and assess the test execution of the network. The fundamental ANN contains two layers which are a secret layer and yield layer. When all is said and done, input neurons are the specific number of highlights vectors. As indicated by the quantity of preparation written by hand digit picture tests and the separated highlights per picture test, the info neurons is 784, and the information vector is 784-by-28000.

Just one secret layer with 100 neurons was taken, which was found to give the best presentation for the proposed application. The program we execute will mostly zero in on distinguishing 0-9 from fragmented pictures of manually written digits.

The contribution of our program is a dim level picture, the power level of which differs from 0 to 255. For straightforwardness, input pictures are pre-treated to be of a specific fixed size, and each info picture ought to contain just a single obscure digit in the center. These prerequisites are not very cruel in light of the fact that they can be accomplished utilizing straightforward picture preparing or PC vision methods. What's more, such pre-treated picture informational collections are not difficult to acquire. In my execution, the famous MNIST informational collection ([1]) is a decent decision. Each picture in MNIST is now standardized to 28x28 in the above sense and the informational collection itself is freely accessible. The MNIST informational index is actually a colossal one: it contains 60000 preparing tests and 10000 test tests. What's more, it has become a standard informational collection for testing different calculations.

The yield of our program will be the comparing 0-9 digit contained in the info picture. The info layer contains a similar number of units as the quantity of pixels in the information picture. For our situation it is 28x28=784. At that point the secret layer containing 100 units with sigmoid enactment is utilized to track down a reduced portrayal of info pictures. We need the yield units to give the contingent likelihood (subsequently the yield of every unit is somewhere in the range of 0 and 1, and the yields of every one of the 10 units will whole to 1) of each class to which each info has a place, and the unit that has the greatest yield will decide the class name. Subsequently, relu(ReLu) actuation is the alluring decision for yield units. Amended Direct Enactment work.

The vital piece of this venture is preparing the neural organization. Since our objective is ordinarily an order issue, the alluring target capacity will be the various class cross-entropy between the organization yield and the objective class names. There is a to some degree inconspicuous point in this program.

4. Results and Discussion

The back propagation learning algorithm portrayed an exactness of 95.06% of the test dataset when it was run multiple times with the same number of iterations. It is evident that with the increase in the number of iterations, the cost at every level decreases, and the level of accuracy increases.

The performance of BPNN can be enhanced by modifying the network itself. We utilized Matlab to reproduce the back propagation algorithm and Tensor flow in python for logistic regression In the logistic regression, relu (ReLU) is utilized as an enactment capacity to upgrade the presentation.

The general approval precision in the presentation is found 97%. The exactness is found at 91% at epoch 1, and epochs 2,3,4,5 have a precision of 95%, 96%, 97%, and 99% separately.

Interestingly, without a secret layer, we have: epoch 1 at 87%, epoch 2, epoch 3, epoch 4 and epoch 5 at 91%, 92% 92.4% and 92.5% each. We can see that adding a secret layer has expanded the exactness of our calculation. We use seaborn from python to portray our outcomes as a confusion matrix, as is seen ahead.

Hence it can be concluded that our proposed system delivered an accuracy level of above 90% and will take less time in training due to the efficient use of these algorithms.

Table 1. Performance of the proposed algorithm

ALGORITHM	ACCURACY
BACK PROPAGATION	95.06 %
LOGISTIC REGRESSION(WITHOUT HIDDEN LAYER)	92%
LOGISTIC REGRESSION(WITH HIDDEN LAYER)	99%

Back propagation:

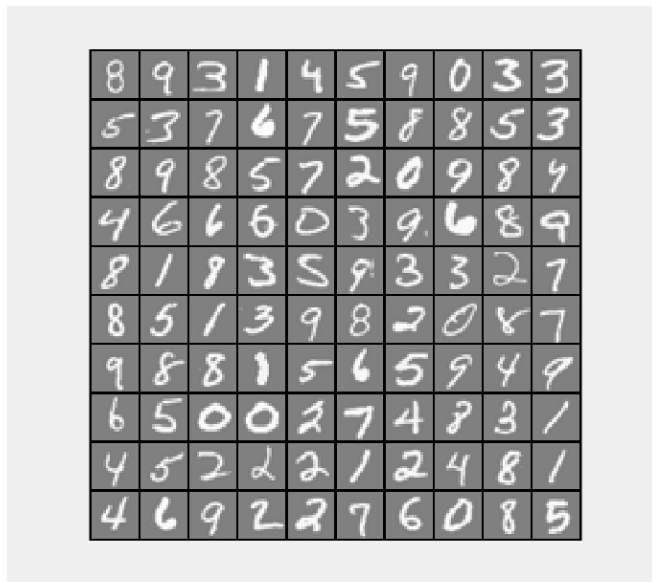

Figure 1. *(loading and visualization of the dataset from MNIST which consists of images containing handwritten digits)*

```
Feedforward Using Neural Network ...
Cost at parameters (loaded from ex4weights): 0.287629
(this value should be about 0.287629)

Program paused. Press enter to continue.

Checking Cost Function (w/ Regularization) ...
Cost at parameters (loaded from ex4weights): 0.383770
(this value should be about 0.383770)
Program paused. Press enter to continue.
```



```
Initializing Neural Network Parameters ...

Checking Backpropagation...
  -0.0093   -0.0093
   0.0089    0.0089
  -0.0084   -0.0084
   0.0076    0.0076
  -0.0067   -0.0067
  -0.0000   -0.0000
   0.0000    0.0000
  -0.0000   -0.0000
   0.0000    0.0000
  -0.0000   -0.0000
  -0.0002   -0.0002
   0.0002    0.0002
  -0.0003   -0.0003
   0.0003    0.0003
  -0.0004   -0.0004
  -0.0001   -0.0001
   0.0001    0.0001
  -0.0001   -0.0001
   0.0002    0.0002
  -0.0002   -0.0002
   0.3145    0.3145
```

The backpropagation algorithm is implemented to match the cost at the output layer and the data that has been forwarded from the input layer.

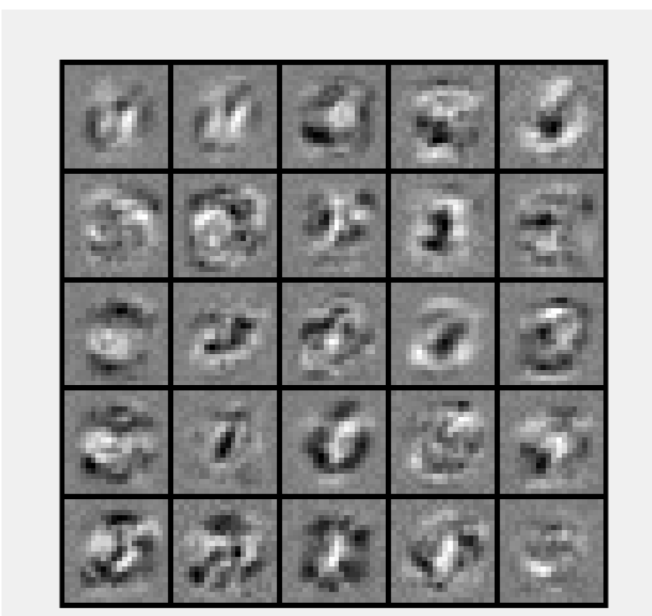

Figure 2: (visualizing the neural network at the hidden layer, to see the features captured in the data)

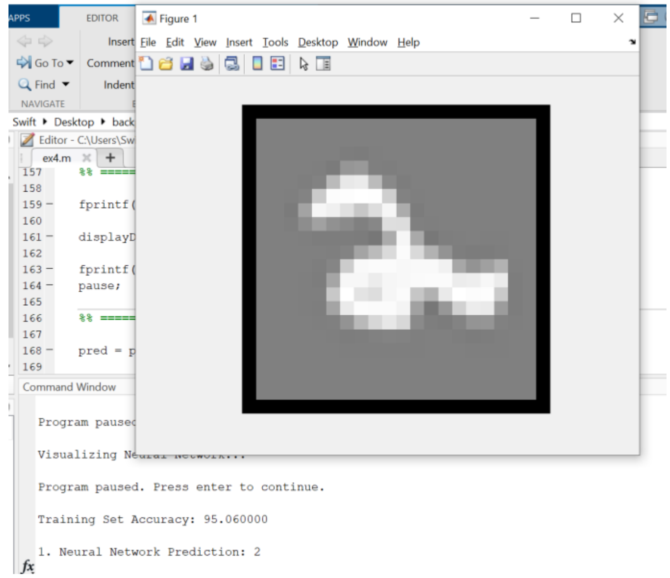

Figure 3(a):

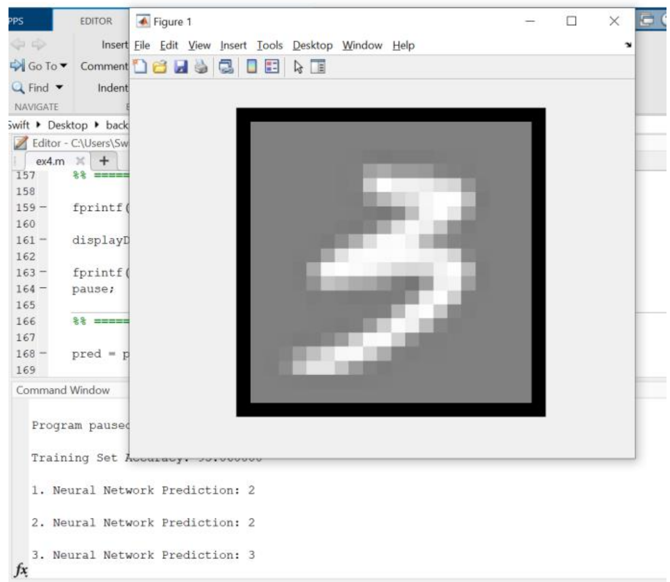

figure 3(b)

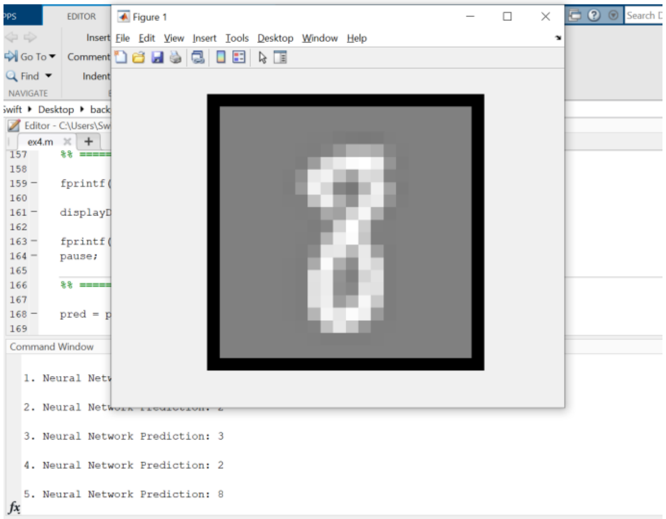

figure 3(c)

Figure 3(a),(b),(c) :predicting the data at the neural network, to predict the accuracy & the sample dataset

We accomplished a precision near 95.06% of the test dataset when it was run with 50 cycles. The expense and the degree of exactness increment with the increment in the emphases.

Logistic Regression:

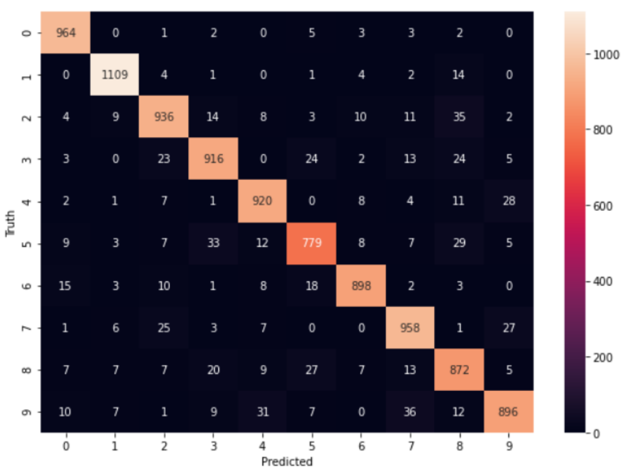

Figure 4. *confusion matrix of a logistic regression algorithm without a hidden layer*

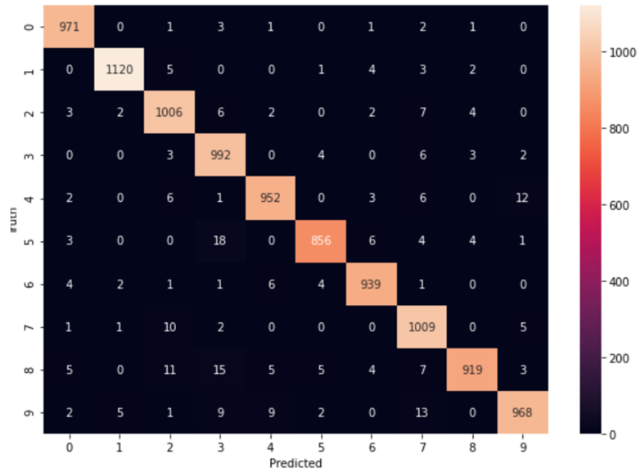

Figure 5. *after adding a hidden layer, we see that the number of zeroes has increased in the confusion matrix, meaning the accuracy has increased.*

We also see the epoch values increasing for the hidden layer, in contrast to without a hidden layer.

```
Epoch 1/5
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2967 - accuracy: 0.9179
Epoch 2/5
1875/1875 [==============================] - 3s 2ms/step - loss: 0.1398 - accuracy: 0.9588
Epoch 3/5
1875/1875 [==============================] - 3s 2ms/step - loss: 0.1005 - accuracy: 0.9695
Epoch 4/5
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0792 - accuracy: 0.9765
Epoch 5/5
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0636 - accuracy: 0.9808
```

(with hidden layer)

```
Epoch 1/5
1875/1875 [==============================] - 2s 1ms/step - loss: 0.4899 - accuracy: 0.8769
Epoch 2/5
1875/1875 [==============================] - 2s 1ms/step - loss: 0.3061 - accuracy: 0.9155
Epoch 3/5
1875/1875 [==============================] - 2s 1ms/step - loss: 0.2852 - accuracy: 0.9203
Epoch 4/5
1875/1875 [==============================] - 2s 1ms/step - loss: 0.2748 - accuracy: 0.9243
Epoch 5/5
1875/1875 [==============================] - 2s 1ms/step - loss: 0.2679 - accuracy: 0.9257
```

(without hidden layer)

We accomplish a precision of 99% when we run 10 epochs of the strategic relapse calculation with a secret layer, rather than 98% in a 5 emphasis run.

```
Epoch 1/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2867 - accuracy: 0.9206
Epoch 2/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.1332 - accuracy: 0.9614
Epoch 3/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0954 - accuracy: 0.9717
Epoch 4/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0744 - accuracy: 0.9775
Epoch 5/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0616 - accuracy: 0.9813
Epoch 6/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0500 - accuracy: 0.9845
Epoch 7/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0414 - accuracy: 0.9876
Epoch 8/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0342 - accuracy: 0.9894
Epoch 9/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0290 - accuracy: 0.9911
Epoch 10/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0247 - accuracy: 0.9927
```


5. Conclusion and Future Scope

We can see that the calculated relapse calculation gives 99% accuracy in perceiving manually written digits, as well as 95% given by the backpropagation calculation. Having said that, it's as yet not adequate for practical use. Compelled to the trouble of acquiring ground truth information, we here just direct trials on a moderately little dataset. In future work, we wish we can prepare these strategies with more information and have further testing. For future work, we propose the accompanying things:

1. We utilized crude pixel force as our highlights. However, we can utilize various strategies to remove highlights which may improve our presentation.
2. In the Neural Network in linear regression, we utilized just one hidden layer. Neural Organizations are computationally over the top expensive and it requires some investment to prepare in our machines. Along these lines, we can attempt to run a Neural Network with more secret layers on superior figuring machines.

We can foster an Android/iOS application that distinguishes the penmanship continuously by clicking an image of it.

References

[1] S. Mori, C.Y. Suen, and K. Kamamoto, "Historical review of OCR research and development," Proc. of IEEE, vol. 80, pp. 1029-1058, July 1992.

[2] S. Impedovo, L. Ottaviano and S. Occhinegro, "Optical character recognition", International Journal Pattern Recognition and Artificial Intelligence, Vol. 5(1-2), pp. 1-24, 1991.

[3] V.K. Govindan and A.P. Shivaprasad, "Character Recognition – A review," Pattern Recognition, vol. 23, no. 7, pp. 671- 683, 1990 International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011 37.

[4] R. Plamondon and S. N. Srihari, "On-line and off-line handwritten character recognition: A comprehensive survey," IEEE. Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 63-84, 2000.

[5] N. Arica and F. Yarman-Vural, "An Overview of Character Recognition Focused onOff-line Handwriting", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2001, 31(2), pp. 216 - 233.

[6] G. Anbarjafari, "1. Introduction to image processing," Sisu@UT. [Online].

[7] S. Tanner, "Deciding whether Optical Character Recognition is feasible" [Online].

[8] H. F. Schantz, The history of OCR, optical character recognition. Manchester Center, VT: Recognition Technologies Users Association, 1982. [4] History of Computers and Computing, Internet, Dreamers, Emanuel Goldberg. [Online].

[9] "Optical Character Recognition: What you Need to Know". Phoenix Software International.

[10] S. N. Srihari, & E. J. Kubert. Integration of Hand-Written Address Interpretation Technology into the United States Postal Service Remote Computer Reader System.

[11] J Pradeep, E.Srinivasan, S.Himavathi, "Diagonal Based Feature Extraction for Handwritten Alphabet Recognition System Using Neural Network", International Journal of Computer Science & Information Technology (IJCSIT), Vol. 3, No 1, pp 1171-1181, Feb 2011.

[12] Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, "Human-Computer Interaction - Third Edition", Pearson Education Limited 2004

Authors Profile

1. Ms.Anushka Sharma - Vellore Institute of Technology, Vellore Machine Learning, Natural Language Processing, Deep Learning, Data Science

2. Prateek Dhawan – Vellore Institute of Technology, Vellore Machine Learning, Artificial Neural Network, Deep Learning, Data Science.

