A Deep Learning Approach to Handwritten Image Recognition

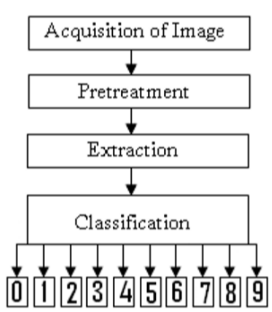
K.Yogeswara Rao, SatyanarayanaMurthy Gorti
Department of computer science and engineering
Aditya Institute of Technology and Management,Tekklali India

Abstract - In this paper we have presented an approach for recognition of the image using Deep Learning with maximum accuracy result for the dataset inserted. The vary Learning rate introduced in this work is to overcome skipping of the data. Dropout used in this work is to overcome over fitting problem, where random neurons are dropped from the network with all their weights and biases.

Keywords - Handwritten recognition, Deep Learning, deep neural network, Convolution Neural Network, Cross Entropy

I. INTRODUCTION

The neural networks unit wide used for the recognition of characters [1, 2, 3, 4, 5, 6, 7, and 8]ii. throughout this workwe have a bent to coach and check a neural network classifier pattern MNIST info, the necessarystep inside the popularityis learning, we've got a bent to used descent of the gradient formula. inside the coaching job methodology the junction weight of the connections between the neurons is modified. the first layers contain connected binary neurons whereasnot editable connections [1].The Multi-Layer Perception contains three layers of neurons; the first layer corresponds to the membrane. In technical terms it matches the input image. The second layer (hidden layer) corresponds to the extraction of characteristics subsystems. The third layer corresponds to the output system. each somatic cell throughout this layer corresponds to one of the output classes. inside therecognition task of written digits, this layer contains 10 neurons such as the digits zero ... 9 (Fig. 1). the primary weights of the network haphazardly connections. The weights unit changedthroughout the formation of the perceptron. The rule of modification of weight corresponds to the academic formula



II. MOTIVATION

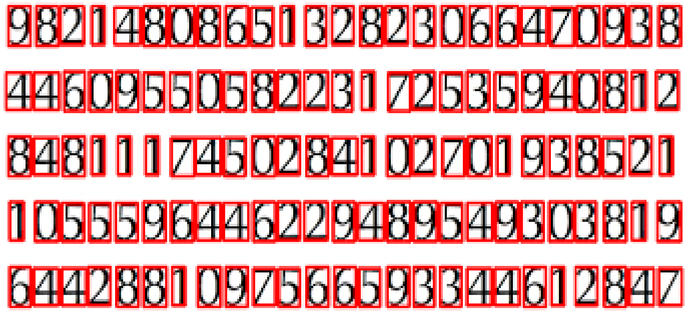
Handwritten digit recognition is a very important downside in optical character recognition, and it's been used as a legal action for theories of pattern recognition and machine learning algorithms for several years. an honest image recognition technique will act as a precursor for several applications, like image to speech app for visually impaired individuals, which might facilitate them to navigate and acknowledge their destination. It also can be accustomed capture serial numbers within the product, hand written prescription numbers, etc.

III. DEEP LEARNING

Deep Learning is AN apparently field of computer science [2] Deep learning evolved by the machine learning, the thought behind deep learning to create learning rule that mimic brain. Deep learning could be a set of machine learning techniques, will teach its self perceive things like pictures, speech a while higher than even folks. Convolutional Neural Networks (CNNs) square measure analogousto ancient ANNs therein they're comprised of neurons that self-optimize through learning. every somatic cell can still receive AN input and perform a operation (such as a real followed by a non-linear function) - the idea of multitudinous ANNs. From the input raw image vectors to the ultimate output of the category score, the complete of the network can still categorical one perceptive score operate(the weight). The last layer can contain loss functions related to the categories, and every one of the regular tips and tricks developed for ancient ANNs still apply. the sole notable distinction between CNNs and ancient ANNs is that CNNs square measure primarily utilized in the sphere of pattern recognition at intervals pictures. this permits America to encrypt image- specific options into the design, creating the network additional suited to image-focused tasks - whileany reducing the parameters needed to line up the model [3].

IV. OCR (OPTICAL CHARACTER RECOGNITION)

Optical character recognition is a crucial application of machine learning wherever Associate in Nursing algorithmic rule is trained on dataset of illustrious digits and may learn to classify digits. With Associate in Nursing optical character recognition system of written digit, with the utilization of neural networks. And a technique of extraction of characteristics supported digit kind, this technique is tested on the MNIST written isolated digit information. Optical character recognition is a crucial application of machine learning wherever Associate in Nursing algorithmic rule is trained on dataset of illustrious digits and may learn to classify digits. With Associate in Nursing optical character recognition system of written digit, with the utilization of neural networks. And a technique of extraction of characteristics supported digit kind, this technique is tested on the MNIST written isolated digit information.



V. RELATED WORK

"Image recognition victimisation Deep Learning" could be a methodology planned application that takes input within the kind of immense quantity information sets specifically coaching and testing dataset and returns image reorganization accuracy. during this work MNIST information set area unit accustomed acknowledge the written digits gift within heyann.lecun.com that contains sixty,000 coaching datasets and ten,000 take a look at datasets. the most accuracy tells the effectiveness of the system.

By victimisation k-nearest neighbors algorithmic rule (k-NN) the most accuracy is gained is ninety nine.28%. [4]

By victimisation Support Vector Machine (SVM) the most accuracy is gained is ninety nine.24%.

By victimisation Neural Network, the most accuracy is gained is ninety eight.4%.

By victimisation Deep Neural Network, the most accuracy is gained is ninety nine.0%.

VI. PROPOSED SYSTEM

Recognition of the image victimisation Deep Learning that displays most accuracy result for the dataset inserted. within the projected system MNIST datasets square measure used for coaching and testing. By victimisation CNN (Convolution Neural Network) with a pair of Stride the datasets square measure classified, cross entropy is taken as loss perform together with the activation perform ReLU (Rectifier Linear Unit). The vary Learning rate is introduced during this system to beat skipping of the information. Dropout is employed during this system to beat over fitting downside, wherever random neurons square measure born from the network with all their weights and biases.

VII. ALGORITHM

Input: Training dataset and Test dataset from MNIST Dataset

Output: Accuracy percentage in different epoch, final maximum accuracy percentage.

Steps: -

1. Imported the MNIST dataset from yann.lecun.com by using " from tensorflow.examples.tutorials.mnist import input data as mnist_data"

Mnist Data are then readied.

2. SoftMax, cross-entropy mini- batches

3. The data is then passed to the convolutional network using activation function RELU.

y=tf.nn.relu(tf.matmul(X,W)+b)

X->100 images, one per line; 100x784.

Y->weight 784x10.

b->bias 100x10.

4. . Cross entropy is choose and applied (to measure distance between the pixel). We added "-" sign as all the logarithm value of computed probabilities is negative.

i.e. Cross entropy= -tf.reduce_sum(Y_*tf.log(y))

5. Now, Actual heart of what tensor flow do is Learning rate decay.

optimizer =tf.train.GradientDescentOptimizer(value) # value= Learning rate e.g. 0.003

trains=optimizer.minimize(Cross_entropy) #where cross entropy act like loss function

6. To increase the efficiency, the SoftMax is replaced with Relu which increases the efficiency and the Learning rate decay is decreased from 0.003 to 0.001.

Y1 = tf.nn.relu(tf.nn.conv2d(X, W1, strides=[1, stride, stride, 1], padding='SAME') + B1) # stride if 1 gives out as 28 x 28

7. All weights, biases are calculated along with accuracy and loss

allweights = tf.concat([tf.reshape(W1, [-1]), tf.reshape(W2, [-1]), tf.reshape(W3, [-1]), tf.reshape(W4, [-1]), tf.reshape(W5, [-1])], 0)

allbiases = tf.concat([tf.reshape(B1, [-1]), tf.reshape(B2, [-1]), tf.reshape(B3, [-1]), tf.reshape(B4, [-1]), tf.reshape(B5, [-1])], 0)

8. Final out is display showing the accuracy of recognition of the test dataset character in MNIST but problem arises over fitting which can be solved by regression function called dropout.

p=tf.placeholder (tf.float32)

Y=tf.nn.dropout(y,p)

VIII. IMPLEMENTATION

WeightsandBiases

Weight is that he strengthof the association. If I increase the input then what proportion influence will it wear the output.

Weights close to zero mean dynamical this input won't amendment the output. several algorithms can mechanically set those weights to zero so as to change the network.

Bias as suggests that however distant our predictions ar from real values. usually constant algorithms have a high bias creating them quick to find out and easier to grasp however usually less

versatile. Successively they're have lower prognosticative performance on advanced issues that fail to fulfill the simplifying assumptions of the algorithms bias.
Low Bias: Suggests a lot of assumptions concerning the shape of the target perform.
High-Bias: Suggests fewer assumptions concerning the shape of the target perform.
Weights and biases ar the learnable parameters of your model. in addition as neural networks, they seem with identical names in connected models reminiscent of rectilinear regression. Most machine learning algorithms embody some learnable parameters like this.
The values of those parameters before learning starts ar initialized arbitrarily (this stops all of them convergency to one value). Then once conferred with information throughout coaching, they'r adjusted towards values that have correct output.
we don't have to be compelled to give values before coaching, though we have a tendency to might want to choose things reminiscent of what number parameters there ought to be (in neural networks that's controlled by the scale of every layer). Tensor Flow calculates the values mechanically, throughout coaching. once
we have associate degree already-trained model and wish to re-use it, then we are going to wish to line the values directly e.g. by loading them from file.

$$y = f(x) = \sum x_i w_i$$

In Tensor Flow [9] weight and biases are formulated as:
W = tf.Variable(tf.truncated_normal([6, 6, 1, 6], stddev=0.1))
B = tf.Variable(tf.constant(0.1, tf.float32, [6]))
CNN for Handwritten Digit Recognition the CNN for Handwritten Digit Recognition works in three main phases.
1. Input MNIST Data1: The first phase is to input the MNIST data. The MNIST data is provided as 784-d array of pixels. So firstly, we convert it to grayscale images using 28x28 matrix of pixels.

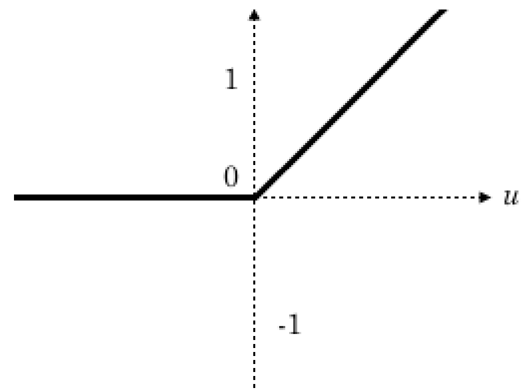
2. Building Network Architecture: In the second phase, we define the models to be used to build a convolutional neural network. Here, we use the Sequential class from Kera‟s to build the network. In this network, we have three-layer sets of layers.

"CONV =>ReLU=> POOL".

a)First Convolution Layer: In the first layer, we take 20 convolutional filters that go as a sliding window of size 5x5 over all the images of 28x28 matrix size and try to get the pixels with most intensity value.
b)ReLU Function: The most successful and widely-used activation function is the Rectified Linear Unit (ReLU)[8]. We know that convolution is a method that uses Back Propagation. So, using the ReLU function as the activation function just after the convolutional layer reduces the likelihood of the vanishing gradient and avoids sparsity. This way we don‟t lose the important data and even get rid of redundant data like a lot of 0‟s in the pixels.

ReLU can be expressed as,
f(x)= x if x>0
f(x)=0 if x<=0

$$f(u) = \max(0, u)$$



ReLU Activation function
In tensorflow we have a ReLU function i.e.: -
Y=tf.nn.relu(tf.matmul(X,W1)+B1)
c) Pooling Layer: The pooling layer gets the data from the ReLU function and down-samples the steps in the 3D tensor. In short it pools all the pixels obtained from previous layers and again forms a new image matrix of a smaller size. These images are again input into the second set of layers i.e. "CONV =>ReLU=> POOL" and this process goes on till we get to a smallest set of pixels from which we can classify the digit.
3. Fully Connected Layer: The fully connected layer is used to connect each of the previous layers to the next layers. This layer consists of 500 neurons. Finally, we apply a SoftMax Classifier that returns a list of probabilities for each of the 10 class labels. The class label with the largest probability is chosen as the final classification from the network and shown in the output.
This output received is used to make the confusion matrix for the model. In this we can add more number of layers but adding more layers might affect the accuracy of the system. Since, it uses multiple layers, so it‟s called a Deep Learning system.

Loss Function Cross Entropy
The effects the choice of the classification loss function has on deep neural networks training as well as their final characteristics[5], so choosing correct function is very important.
Cross Entropy(CE) is a function of weights, biases, pixels of the training images and its known labels.
The basic idea behind the Cross-Entropy method is to transform the original (combinatorial) optimization problem to an associated stochastic optimization problem, and then to tackle

the stochastic problem efficiently by an adaptive sampling algorithm. By doing so one constructs a random sequence of solutions which converges (probabilistically) to the optimal or at least a reasonable solution. Once the associated stochastic optimization is defined, the CE method alternates the following two phases:

1. Generation of a sample of random data (trajectories, vectors, etc.) according to a specified random mechanism.
2. Update of the parameters of the random mechanism, on the basis of the data, in order to produce a "better" sample in the next iteration.

Cross entropy (CE)= -Σ Yi'.log(Yi)

Where Y" is „one-hot encoded" that means that if one wants to represent label "2" by using a vector of 10 values, all zeros but 2th value which is 1. It is very handy as the format is very similar to the neural network outputs its prediction as vector of 10 values.
Eg: if one wants to represent 2 in one hot encoded system the vector will be as shown in figure below:

0	1	0	0	0	0	0	0	0	0

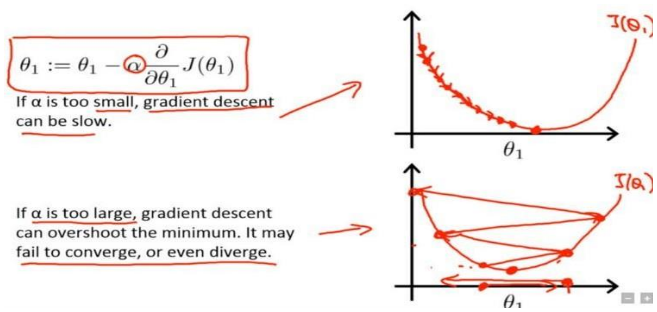


Fig.1: one-hot encoded for 2

And Y is the computed probabilities can be represented as for the recognition of digit 2:

0.1	0.9	0.2	0.1	0.1	0.2	0.3	0.1	0.3	0.2

Fig.2: computed probabilities for 2

In tensor flow cross entropy can be calculated as:
CE=-tf.reduce_sum(Y_*tf.log(Y))

Learning Rate

The learning rate is that the most significant hyper-parameter to tune for coaching deep neural networks [7]. Learning rate may be a decreasing operate of your time. 2 forms that ar ordinarily used ar a linear operate of your time and a operate that's reciprocally proportional to the time t.
The learning rate is one among the foremost necessary hyper-parameters to tune for coaching deep neural networks.
Deep learning models ar usually trained by a random gradient descent optimizer. There ar several variations of randomgradient descent: Adam, RMSProp, Adagrad, etc. All of them allow you to set the educational rate. This parameter tells the optimizer however way to maneuver the weights within the direction of the gradient for a mini-batch.
If the educational rate is low,
then coaching is additional reliable, however improvement can take loads of your timeas a result of steps towards the minimum of the loss operate ar little.
If the educational rate is high, then coaching might not converge or maybe diverge. Weight changes will be thereforehuge that the optimizer overshoots the minimum and makes the loss worse. Leslie N. Smith describes a powerful technique to select a range of learning rates for a neural network in section 3.3 of the 2015 paper "Cyclical Learning Rates for Training Neural Networks". The trick is to train a network starting from a low learning rate and increase the learning rate exponentially for every batch.

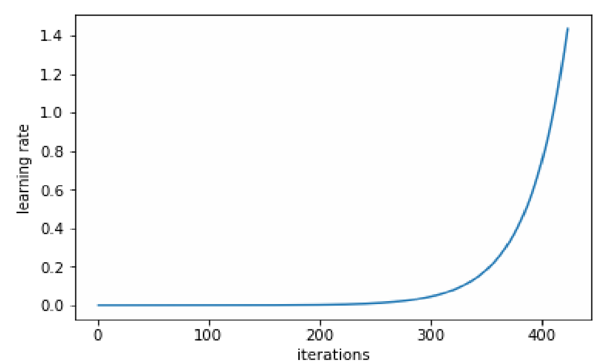
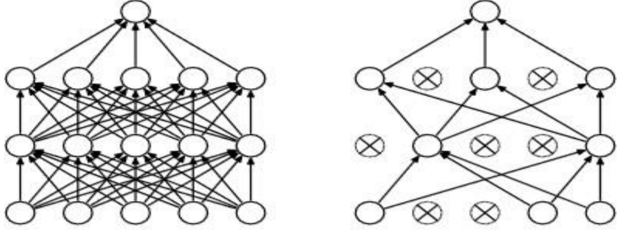


Fig.3: LR increases after each mini-batch



(a) Standard Neural Net (b) After applying dropout.

In TensorFlow it is evaluated as: -

```
max_lr = 0.003
).0001
/_speed = 2000.0
LR = min_lr + (max_lr – min_lr) * math.exp(-
i/decay_speed)
```

DROPOUT OVERFITTING SOLUTION

Deep neural networks contain multiple non-linear hidden layers and this makes them terribly communicatory models which will learn terribly difficult relationships between their inputs and outputs. With restricted coaching information, however, several of those difficult relationships are the results of sampling noise, in order that they can exist within the coaching set however not in real check information though it's drawn from constant distribution[6]. This ends up in overfitting and lots of strategies are developed for reducing it. These embrace stopping the coaching as presently as performance on a validation set starts to induce worse, introducing weight penalties of varied types similar to L1 and L2 regularization and soft weight sharing (Nowlan and Hinton,1992). In dropout, at every coaching iteration, we tend to drop random neurons from the network. we elect a chance (P) for a somatic cell to be unbroken, typically between five hundredth and seventy fifth, and so at every iteration of the coaching loop, we tend to haphazardly take away neurons with all their weights and biases. completely different neurons are born at every iteration (and you furthermore mght ought to boost the output of the remaining neurons in proportion to create positive activations on ensuing layer don't
shift). once testing the performance of your network after all you place all the neurons back (P=1).

TensorFlow contains a dropout function which randomly zeroes-out some of the outputs and boosts the remaining ones by 1/P.

```
aceholder(tf.float32)

Y=tf.nn.dropout(tf.relu(tf.matmul(X,W1)+B1), P)
```

DATA SET:

The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples[8]. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image. The dimensionality of each image sample vector is 28*28 i.e. 784, where each element is binary.

IX. RESULT ANALYSIS

This paper is used to find out accuracy of the test dataset which is 10,000 images. Below table contains the result of 100 epochs where each epochs contains 100 evaluated, computed result

Result Evaluation

Graphical analysis

A graph is plotted accuracy percentage vs epochs, where epochs are X axis and Accuracy percentage are Y axis. We can see that the started at minimum and reached to constant maximum of 99% and the maximum accuracy in it is 99.36%.

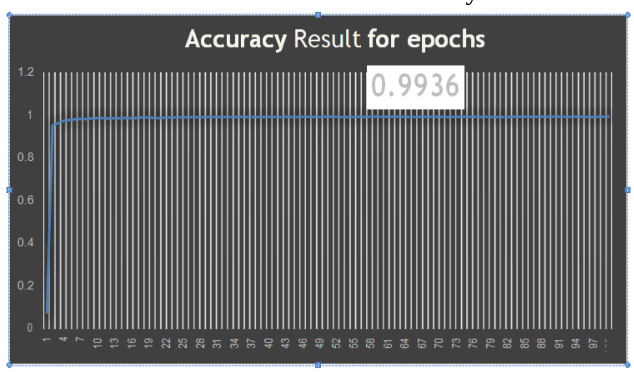


Fig.6.3.3.1. Graphical representation of result

X. CONCLUSION

In this work, Deep Convolution Neural Network is employed for classification of the quality base MNIST isolated. the popularity rate is ninety nine.36% with a check info containing sixty,000 images. This rule will be utilized in recognizing communicating codes, phone digits, vehicle range, and alternative fields. Even by exploitation same technique projected during this work, we will classify alternative pictures like face recognition, object detection, etc. The CNN with additional layers tends to supply a lot of accuracy. what is more, we will apply batch nomination to extend the potency to most.

XI. REFERENCES

[1]. Katarzyna Janocha1, Wojciech Marian Czarnecki2,1 1Faculty of Mathematics and Computer Science, Jagiellonian University, Krakow, Poland 2DeepMind, London, UKOn Loss Functions for Deep Neural Networks in Classification. 18 Feb 2017, arXiv:1702.05659v1 [cs.LG].

[2]. M. Abadi, A. Agarwal, et. al. and Xiaoqiang Zheng, "Tensor Flow: Large-scale machine learning on heterogeneous systems", (2015). Software available from tensorflow.org.

[3]. An Introduction to Convolutional Neural Networks Keiron O'Shea1 and Ryan Nash2 1 Department of Computer Science, Aberystwyth University, Ceredigion, SY23 3DB keo7@aber.ac.uk 2 School of Computing and Communications, Lancaster University, Lancashire, LA1

[4]. Norhidayu binti Abdul Hamid, Nilam Nur Binti Amir Sjarif, Handwritten Recognition Using SVM, KNN and Neural Network.

[5]. Katarzyna Janocha, Wojciech Marian Czarnecki,On Loss Functions for Deep Neural Networks in Classification, 18 Feb 2017, arXiv:1702.05659v1 [cs.LG].

[6]. Nitish Srivastava nitish@cs.toronto.edu Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever ilya, Ruslan Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting Journal of Machine Learning Research 15 (2014) 1929-1958, Published 6/14.

[7]. Leslie N. Smith, Cyclical Learning Rates for Training NeuralNetworks, rXiv:1506.01186v6 [cs.CV] 4 Apr 2017.

[8]. The MNIST Database of Handwritten Digit Images for Machine Learning Research, Li Deng, [best of THE WEB.

[9]. http://yann.lecun.com/exdb/mnist.

K Yogeswara rao received his M.tech From Andhra University,Visakhapatnama, AP India , and Ph.D in Jan 2017 in Computer Science and Engineering and area of Specialization is Text Mining. He is working as a Associate Professor in Aditya Institute of Technology and Management, Tekkali-532201, AP,India. He is having 19+ Years of experience in teaching. He published good number of papers in International Journals with good impact factor and Scopus Indexed. He is a Member of CSI.

Satyanarayana Murty Gorti received his Master's Degree from JNTU Kakinada, AP India , and Ph.D in May 2014 in Computer Science and Engineering and area of Specialization is Image Mining. He is working as a Professor in Aditya Institute of Technology and Management, Tekkali-532201 , AP,India. He is working in this Institute since August 2005 and having 20+ Years of experience in teaching. He acted as a reviewer for various International conferences and Journals. He acted as a Convener for AICTE Sponsored FDPs and Conferences. He published good number of papers in International Journals with good impact factor and Scopus Indexed. He is a Life Member of CSI and ISTE.