Handwritten Digit Recognition using DNN, CNN and RNN

Subhi Jain[1], Rahul Chauhan[2]
2510shubhijain@gmail.com[1], chauhan14853@gmail.com[2]

[1]Graphic Era Hill University, Computer science and Engineering Department, Dehradun.
[2]Graphic Era Hill University, Electronics & Communication Engineering Department, Dehradun.

Abstract- Deep learning is the domain of machine learning that implements deep neural architectures to mimic the functions of the human brain deep neural networks are the same artificial neural networks with many hidden layers In contrast to shallow networks, deep neural networks are implemented using deep learning algorithms. The network learns from multiple levels of representation and accordingly responds to different levels of abstraction, where each layer learns patterns. Handwritten digit recognition is a classic machine learning program often said as the “Hello World !” of machine learning datasets for implementation of algorithms. There have been numerous machine learning techniques that have been applied on this problem. This paper focuses on the implementation neural networks and deep learning algorithms. The NN algorithms such as DNN, CNN and RNN are used along with the basic shallow neural network. The algorithms are implemented on various deep learning frameworks and the performance is compared and evaluated in terms of accuracy of the models. The best accuracy according to the results has been of CNN 99.6%. On the other hand, the remaining algorithms have an error rate ranging from 2-3%.

Keywords- Deep Learning, Convolutional neural networks, Recurrent neural networks, Deep neural networks, Shallow neural networks.

1. Introduction

Hand written digit recognition is a classic problem in the field of image recognition. The shape of the digits and its features help identify the digit from the strokes and boundaries. There have been great achievements in recent years in the field of pattern recognition, particularly in the field of Hand written digit recognition problem [1]. Hand writing recognition is the ability of a device to take handwriting as input from sources. The handwriting taken as input can be used to verify signatures, used to interpret text and OCR (optical character recognition) to read the text and transform it into a form which can be manipulated by computer [3]. The traditional machine learning algorithms are shallow learning algorithms and incapable of extracting multiple features. In the era of big data, deep learning algorithms have performed efficiently in digit recognition tasks on MNIST dataset [2]. Fig.1 shows the samples of digits in MNIST dataset. Neural networks and deep learning have proven to perform exceptionally well in recognising handwritten digits.

Artificial neural networks are machine leaning algorithms that are an implementation of the neural structure of brain. Similar to the brain, the neural networks take input and each input has a weight. The weight and bias along with the input are fed to hidden layers, and then based on activation values of the calculations the calculations are forwarded to the output layer. The neural networks with one hidden layer are called shallow neural networks. The shallow neural networks are incapable of training datasets that require multiple feature extraction. Hence, Deep neural networks were introduced. Deep neural networks are the neural networks with more than one hidden layers. In this each hidden layer learns a different feature. The state of art neural networks recently evolved into Deep learning algorithms which mimic the functions of human cerebral cortex in their implementation [4].

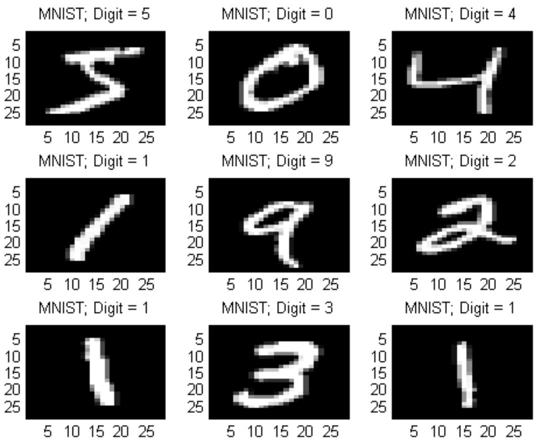


Figure 1. Sample of handwritten digits in MNIST dataset

In this paper, various neural network and deep learning algorithms are implemented to perform handwritten digit recognition. The implementations discussed in this paper are of shallow neural network, deep neural networks and deep learning algorithms. The architectures of the neural networks are different and each algorithm performs differently on the dataset. The experiments were conducted and results are verified with algorithms and their performance on the dataset.

The paper is structured as follows: Introduction is followed by the literature survey. The dataset is discussed next followed by the learning models implemented on the datasets for digit recognition. The implementation begins with shallow neural networks, followed by a 3 layer and 4 layer neural networks. The deep neural networks are followed by deep learning algorithms, namely, Convolutional neural networks and Recurrent neural networks. The results are in the next section in form of tabular data and graphs.

2. Literature Survey

There have been many notable works in handwriting recognition. Other than that, deep learning and neural networks have also witnessed great strides. Some of the works have been listed below:

Norhidayu binti Abdul Hamid et al. [3] have performed hand written digit recognition over MNIST dataset using CNN (Convolutional neural networks) , SVM (Support Vector Machines) and KNN (K- Nearest Neighbor) classifiers. In their work, KNN and SVM predicted the outcomes correctly on datasets but Multilayer perceptron fail to recognize the digit 9 due to non convex function as it gets stuck in the local minima. It was concluded that th eaccuracy would improve by using CNN with Keras.

Mahmoud M. Abu Ghosh et al. [2] have performed a comparative study on digit recognition using neural networks. They implemented Deep neural networks(DNN), Convolutional neural networks (CNN) and Deep belief networks. The maximum accuracy is of DNN i.e 98.08% as evaluated by the model. They have also compared the execution time and shown the error rates with various digits that may appear similar.

Youssouf Chherawala et al. [5] in their article stated that the word image is used to extract features and then the handwriting is recognised from those features. They developed the model as an application of recurrent neural networks. The RNN classifier used a weighted vote combination, where the significance of feature sets is recognised by the weights and their combination.

Nurul Ilmi et al. [6] in their article use local binary patterns for feature extraction and KNN classifier for the recognition of hand written digits. The testing result on MNIST data had an accuracy of 89.81% and the C1 form data had an accuracy of 70.91%. The C1 form data was used by the General Elections in Indonesia.

3. Proposed methods

3.1. Datasets

The handwritten digit recognition system uses the MNIST data set [7]. It has 70,000 images that can be used to train and evaluate the system. The train set has 60,000 images and the test set has 10,000 images. It is the subset of NIST dataset (National institute of standards and technology) , having 28 x 28 size input images and 10 class labels from (0-9). Therefore, the size of image is 28 x 28 pixel square i.e 784 pixels.

The dataset is fed to 4 classification algorithms, namely : shallow neural networks, Deep neural networks (3-layer), Convolutional neural networks, and Recurrent neural networks.

3.2 Classifiers

3.2.1 Deep Neural networks

The Deep neural networks are implemented in two ways: 4-layer. The 4-layer deep neural network uses a multilayer perceptron classifier or a deep neural network with 3 hidden layers and one output layer. The hyper parameters for the model are : number of neurons in hidden layers is 200,150 and 100 respectively, learning rate 0.005, batch size of 128 and number of epochs is 10. The number of neuron in the output layer is 10. The architecture uses Relu activation for the hidden layers and softmax activation for output layer. Fig 4. shows the 4-layer architecture, fig.5. Shows the calculated accuracy i.e. 97.8 %..

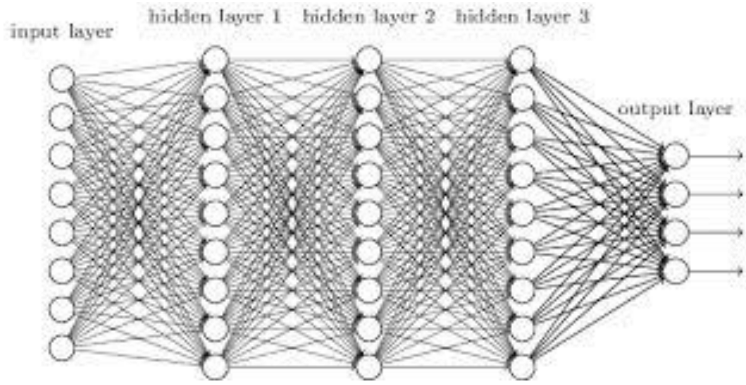


Figure.4. 4-layer neural network architecture

```
Epoch:  0
Epoch:  1
Epoch:  2
Epoch:  3
Epoch:  4
Epoch:  5
Epoch:  6
Epoch:  7
Epoch:  8
Epoch:  9
Accuracy:  0.978
done
Neural Network predicted 2
Real label is: 2

Process finished with exit code 0
```

Figure 5. Accuracy of Deep Neural network (4 layer)

3.2.3 Convolutional Neural Networks

Convolutional neural networks deal with image data, typically 2D data and uses convolution, pooling and fully connected layers to classify the data and produce the output. The three main features of CNN are : Local Receptive field, shared weights and biases and pooling. The convolution layers use the convolution operation between the input image and the filter or kernel. The filter or kernel is also a 2D matrix that is responsible for generating feature maps using the local receptive field. Local receptive field is a small localised field of the input image connected to a single neuron in the feature map. The number of feature maps is dependent on the number of features to be classified. The kernel acts as a weight matrix and learns the weights after the feature map detects the features. The shared weights and bias are connected to local receptive field and the output is given as [8,9,10] :

$$O = \sigma(v + \sum_{i=0}^{4} \sum_{j=0}^{4} w_{i,j}\, h_{a+i,b+j}) \qquad (5)$$

Here, O= output , σ = sigmoidal function, v = shared bias value, $w_{i,j}$ = 5 x5 array of shared weights and $h_{x,y}$ is the input activation at position x ,y . It means that first hidden layer neurons detect the same feature across the entire image. The pooling layers simplify the output after convolution. There are two types of pooling: Max pooling and L2 pooling. In max pooling the maximum activation output is pooled into a 2 x 2 input region and L2 pooling takes the square root of the sum of squares of the activation 2x2 region. Finally, the fully connected layers connect each layer of max pooling layer to the output neurons. The architecture of the developed model is as follows:

Convolution_layer 1 → Relu → Max_pool → dropout → Convolution_layer 2 → Relu → Max_pool → dropout → Convolution_layer 3 → Relu → Max_pool → fully_connected → dropout → output_layer → Result

Dropout is a regularization parameter that prevents overfitting of the data. It randomly dead some nodes in the network using the keep_prob parameter. Keep_prob is the probability of the hidden nodes to be in the network. The 28 x 28 input image is taken by the model and passed to the various layers. The first filter is of size 5x5x1x32 (32 features to learn in first hidden layer), 3x3x32x64 for the second convolution layer, 3x3x64x128 for the third layer, (128*4*4,625) for the fourth layer and (625,10) for the last layer. The stride is 1 for convolution layer and 2 for max pooling layers. The padding is SAME. The optimizer used is RMS optimizer with a learning rate of 0.001 and the parameter β is 0.9. The keep_prob value is 0.8. The accuracy of this model is the highest amongst all i.e. 99.6%. Fig. 6 shows the CNN architecture and Figure &. shows the accuracy.

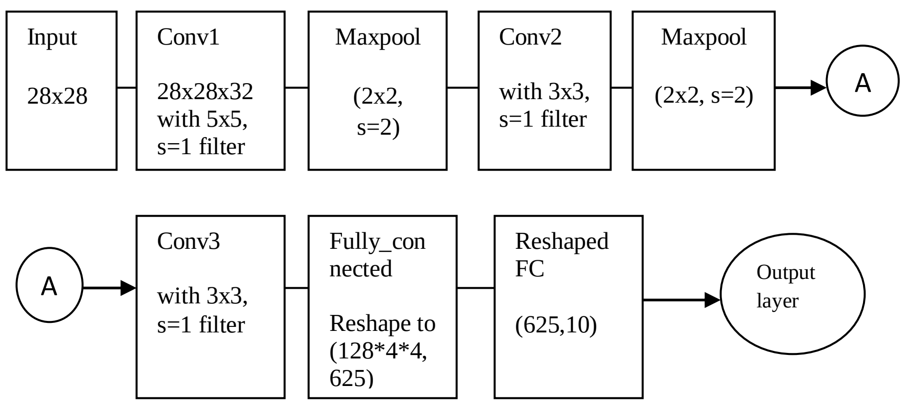


Figure. 6. CNN model for the MNIST dataset

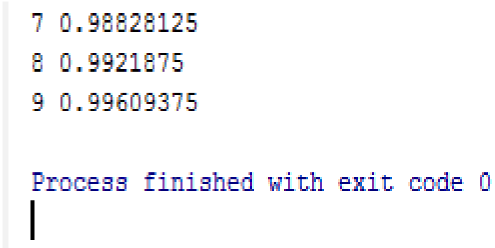


Figure.7. Accuracy on CNN model.

3.3.4 Recurrent Neural network

RNN allow the information to be continuous and lasting by using a loop. It process the input one at a time in sequence and updates the state of the vector which has data about past elements. Traditionally, the neural networks give input simultaneously, are independent of one another and have different parameters. The recurrent neural networks process one input at a time, the weight and shared bias parameters are same and a dependent on one another. Here, we have used a bidirectional RNN which states that output will depend on previous and future data elements in sequence [11]. The RNN run in opposite directions and the outputs of both are mixed. One executes the process in a direction and the other runs in opposite direction. The architecture of the model has input number as 28, number of steps as 28, number of hidden neurons as 128 and output labels as 10. The learning rate is 0.001, training iterations are 100000, batch size is 128 and display step is 10. The optimizer is Adam optimizer with default values. Two LSTM cells are defined in the model and the model is trained. Figure.8. shows the bidirectional RNN architecture and Figure 9. shows the accuracy which is 99.2%.

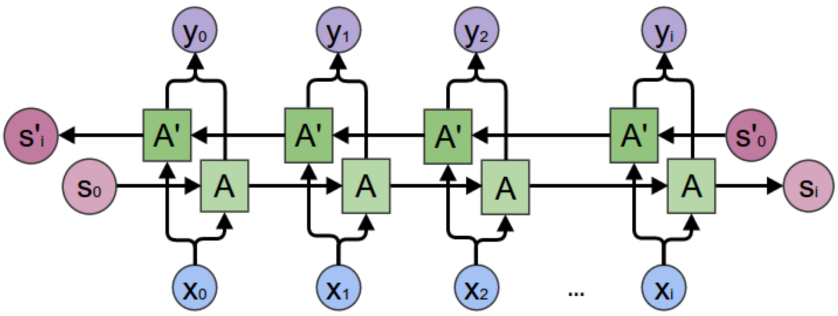
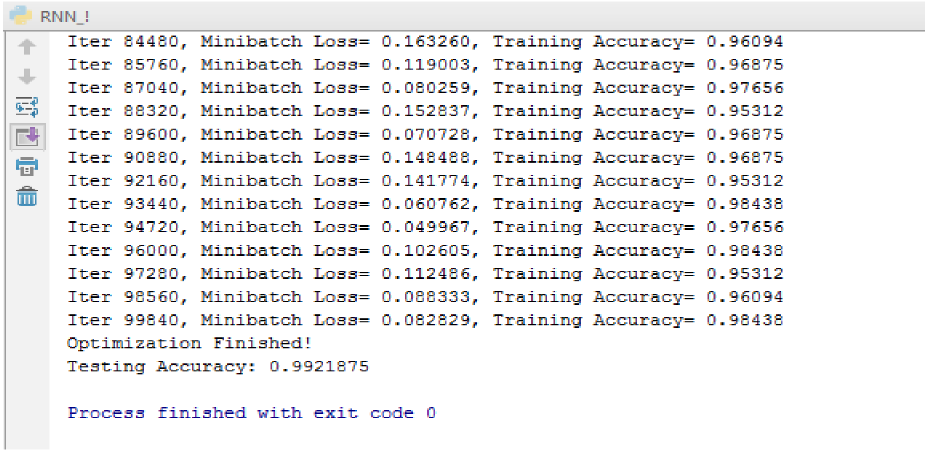


Figure.8. Bidirectional RNN architecture



```
RNN_!
Iter 84480, Minibatch Loss= 0.163260, Training Accuracy= 0.96094
Iter 85760, Minibatch Loss= 0.119003, Training Accuracy= 0.96875
Iter 87040, Minibatch Loss= 0.080259, Training Accuracy= 0.97656
Iter 88320, Minibatch Loss= 0.152837, Training Accuracy= 0.95312
Iter 89600, Minibatch Loss= 0.070728, Training Accuracy= 0.96875
Iter 90880, Minibatch Loss= 0.148488, Training Accuracy= 0.96875
Iter 92160, Minibatch Loss= 0.141774, Training Accuracy= 0.95312
Iter 93440, Minibatch Loss= 0.060762, Training Accuracy= 0.98438
Iter 94720, Minibatch Loss= 0.049967, Training Accuracy= 0.97656
Iter 96000, Minibatch Loss= 0.102605, Training Accuracy= 0.98438
Iter 97280, Minibatch Loss= 0.112486, Training Accuracy= 0.95312
Iter 98560, Minibatch Loss= 0.088333, Training Accuracy= 0.96094
Iter 99840, Minibatch Loss= 0.082829, Training Accuracy= 0.98438
Optimization Finished!
Testing Accuracy: 0.9921875

Process finished with exit code 0
```

Figure.9. Accuracy of Bidirectional RNN model

4. Results of Experiment

DNN, CNN and Bidirectional RNN are implemented on MNIST dataset with varying accuracy. The accuracy of the algorithms is tabulated below:

Algorithm	Accuracy
Deep Neural Networks (4-Layer)	97.74%
Convolutional neural networks	99.6%
Bidirectional Recurrent neural networks	99.2%

Table.1. Accuracy of algorithms

5. Conclusion

The results conclude that CNN performed has the best accuracy on MNIST dataset of 99.6%. Bidirectional RNN has the accuracy of 98.43% on training dataset and 99.2% on testing dataset. The 4-layer DNN has the least accuracy of 97.4%. This is because the convolution neural network use feature maps to learn the features from an image. The features and stroke are more helpful in predicting the digits accuracy rather than the hidden layers in DNN. Bidirectional RNN also perform well as they use previous layer output as input. Thus, CNN classifies the model with the best accuracy.

References:

1. Ping kuang, Wei-na cao, Qiao wu, “Preview on Structures and Algorithms of Deep Learning”,IEEE, 2014.
2. Mahmoud M. Abu Ghosh ; Ashraf Y. Maghari, “A Comparative Study on Handwriting Digit Recognition Using Neural Networks”, IEEE,2017.
3. Norhidayu binti Abdul Hamid, Nilam Nur Binti Amir Sjarif, “Handwritten Recognition Using SVM, KNN and Neural Network”, www.arxiv.org/ftp/arxiv/papers/1702/1702.00723.
4. Yann LeCun,Yoshua Bengio, Geoffery Hinton, “Deep Learning”, NATURE,VOL 521, May 2015.
5. Youssouf Chherawala, Partha Pratim Roy and Mohamed Cheriet, “Feature Set Evaluation for Offline Handwriting Recognition Systems: Application to the Recurrent Neural Network,” IEEE TRANSACTIONS ON CYBERNETICS, VOL. 46, NO. 12, DECEMBER 2016.
6. Nurul Ilmi, Tjokorda Agung Budi W and Kurniawan Nur R, “Handwriting Digit Recognation using Local Binary Pattern Varience and K-Nearest Neighbor,” 2016 Fourth International Conference on Information and Communication Technologies (ICoICT).
7. http://yann.lecun.com/exdb/mnist/
8. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich,A., “Going deeper with convolutions”. *CoRR*, abs/1409.4842, 2014.
9. Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., and Yan, S. CNN: Single-label to multi-label. *CoRR*,abs/1406.5726, 2014.
10. Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901,2013. Published in Proc. ECCV, 2014.
11. Mike Schuster and Kuldip K. Paliwal, “Bidirectional Recurrent Neural Networks”, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 45, NO. 11, NOVEMBER 1997.