Handwriting recognition using Deep Learning in Keras

Shefali Arora
Division of Computer Engineering
Netaji Subhas Institute of Technology
Delhi,India

M.P.S Bhatia
Division of Computer Engineering
Netaji Subhas Institute of Technology
Delhi,India

***Abstract*— Nowadays, deep learning is playing an important role in the domain of image classification. In this paper, a Python library known as Keras, is used for classification of MNIST dataset, a database with images of handwritten images. Two architectures – feed forward neural networks and convolutional neural networks are used for feature extraction and training of model, which is optimized using Stochastic Gradient Descent. This paper gives an overview of multi-class classification of these images using these models, and their performance evaluation in terms of various metrics. It is observed that convolutional neural networks achieve a greater accuracy as compared to feedforward neural networks for classification of handwritten digits.**

Keywords— Neural networks,MNIST,classification

I. INTRODUCTION (*HEADING 1*)

In a real neuron system, neurons receive inputs from dendrites in the brain and send outputs along their axon[1]. The axon connects to other dendrites via synapses. In a computational model of neuron, these signals that travel along the dendrites (x_0) interact with other dendrites based on a multiplicative weight (w_0). These weights are essential to control the influence of neurons on each other . Thus a neural network model can be represented as follows:

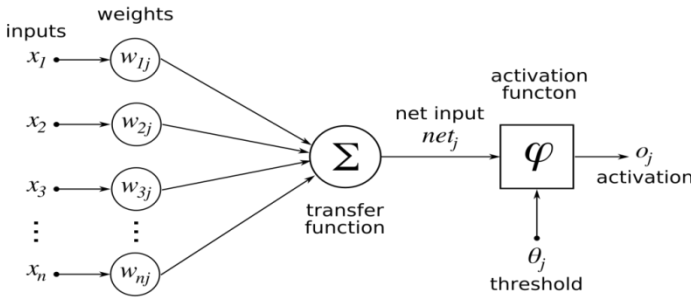

Fig.1. Architecture of neural network

$$y = f\left(\sum w_i\, x_i + b\right) \qquad (1)$$

Here b is the bias factor, and f is the frequency of output signals along the axon. In a neural network model, f is defined as activation function, which could be sigmoid or Rectified Linear Unit (ReLu) function[2].

A Convolutional Neural Network is a special case of neural networks, which involve various convolution layers, followed by fully connected layers as in a neural network[3].

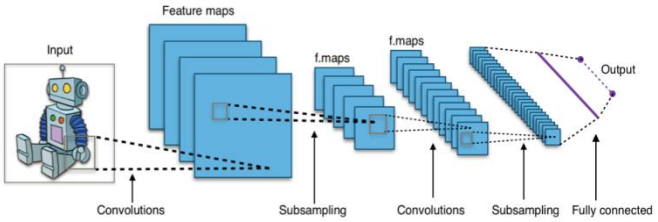

Fig.2. Convolutional neural network

In this figure, CNN has been applied to object recognition, which is performed very efficiently by the network. Each feature of receives inputs from small feature sets in its neighborhood. This helps to extract corners, points very easily and these are finally combined by the higher layers in the network. The process of feature extraction is done with the help of convolution filters.

The use of CNNs gives better performance in terms of complexity and memory requirements, as it combines the weights of convolution layers during feature extraction with fully connected layers used for the process of classification. Thus, these networks are commonly used for image recognition, speech recognition and video analysis.

In this paper, the application of neural networks and convolutional neural networks is evaluated on MNIST database[4]. MNIST is a handwritten digit dataset , which consists of 60,000 training images and 10,000 images in the test set. The digits are centered in a fixed size (28 x 28) image. These algorithms are employed to determine the accuracy with which these digits are classified..

The platform used for classification of digits in MNIST is Keras, a high level neural network API written in Python. It runs with Tensorflow or Theano in the backend. Keras was written to make the implementation of neural networks easier, as Tensorflow seemed rather complicated for the same[5].

ISBN: 978-1-5386-4119-4/18/$31.00 ©2018 IEEE

In the next section , the architecture and application of these networks on MNIST is explained. In Section 3, the results are evaluated in Keras based on their accuracy of classification. Section 4 concludes the paper.

II. LAYERS OF CNN

A. Convolutional layer

These layers make use of convolution to extract various features from the input. A filter is a small matrix ehich is used for detection of features. A convolved feature or activation map is obtained by sliding a filter over the image and computing the dot product. A filter of size n x n convolved with input of size N x N gives feature output with (N-n+1) x (N-n+1) elements. Starting from the top right corner, each filter is moved in downward direction, and then left to right until the whole input is covered.

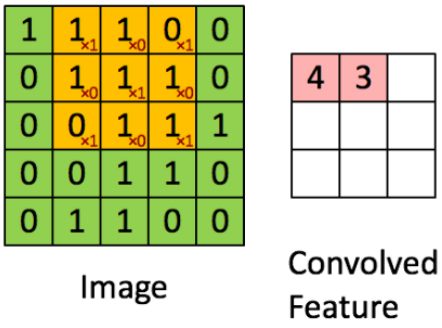

Fig.3. Convolution applied to layers

B. Pooling layer

The pooling layers reduce the resolution of features. It is the technique of moving window across the 2D window space , and the maximum value in the window is the output. This depends on the size of pooling layer taken by the user.

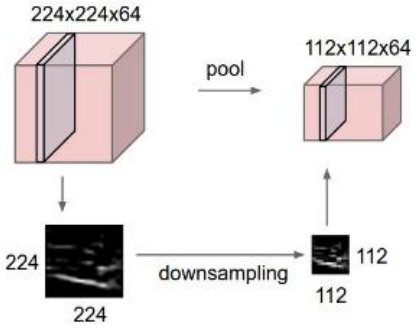

Fig.4. Max pooling

C. Non linear layer

ReLu, or rectified linear unit has been used as the activation function. It is a simple and efficient function, which helps to solve the problem of vanishing gradients in neural networks.

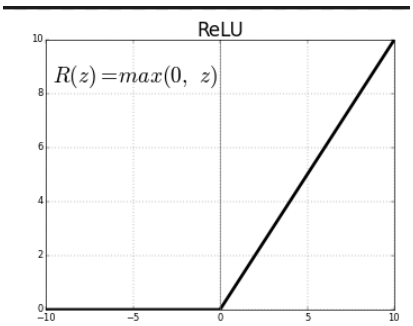

Fig.5. Graph depicting ReLu function

It removes any negative values from the output and makes sure that input and output layer sizes are the same

D. Fully Connected layer

These are the final layers of CNN , which sum weights of the previous layers and determine a specific target result. All elements in the previous layers get involved to give each element of output feature.

Convolutional neural networks are a better option for image recognition than simple feed forward neural networks as they are easier to train and are shift invariant, in case of change in position of images . Also memory requirements are reduced drastically in the case of CNNs as compared to feedforward neural networks.

E. Optimization of weights

Optimization of weights as been done using Stochastic Gradient Descent. The goal of optimization is to minimize the objective function. Traditionally, it is measured by finding the difference between the actual output t and predicted output .

$$E = \frac{1}{2}\,(t - f(w^T x))^2 \qquad (2)$$

We need to optimize w such that the function E is minimized. This is done using Stochastic Gradient descent, which is a better option over traditional batch gradient descent function. This is because while batch gradient descent scans all training examples at a time, SGD checks on a single training example and optimizes weights Thus it converges faster and avoids redundancy[7].

i)Randomly shuffle the dataset.
ii)Optimize weights by scanning one data point at a time

$$w_j = w_j - \eta\frac{\delta E}{\delta j} \qquad (3)$$

$$w_j = w_j - \eta(\,t - f(w^T x))\;\; x \qquad (4)$$

Thus SGD is not as memory intensive as the traditional gradient descent algorithm.

III. RECOGNITION OF DIGITS USING KERAS

The MNIST database consists of images of sizes 28 x 28. In case of feedforward neural network, we reshape the image matrix into a vector of size 784 x 1 and feed it as input. The two hidden layers consist of 512 neurons each, which feed output to a layer of 10 neurons. As there are 10 available classes for classification (i.e digits 0-9), we use 10 neurons in the output layer . Thus 60,000 examples are trained using these networks and 10,000 samples are tested to perform classification of digits.
The following images show the trained handwritten digits and an overview of the workflow using Keras:

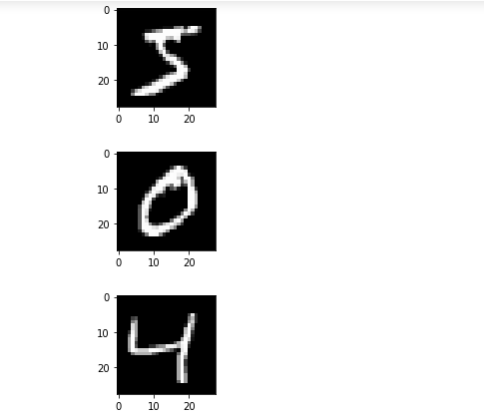

Fig.6. Trained handwritten images

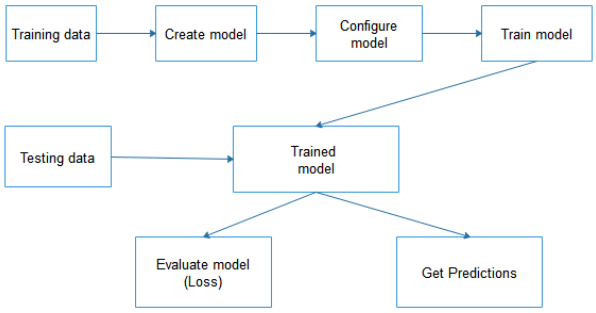

Fig.7. Workflow of Keras

- Sequential model is used for building the neural network, in which layers are stacked one by one.
- A Dense layer is used to connect one layer of neurons to the previous layer of neurons.
- ReLU activation function is added in each layer so the the network learns about non linear decision boundaries.
- In the output layer, sofmax function is used which helps to calculate probability with which an image falls in a particular class.
- One hot encoding is applied on labels of the image i.e digits 0-9. In one-hot encoding, there is only 1 in the array, whereas all other elements are zero.
- In case of Convolutional neural networks, 60,000 images are trained and shape of each image matrix is 28 x 28. The shape is in the form of (batch, height,width,channels). In our case, the number of channels is 1 as images are in grayscale, thus it is (60000,28,28,1). Images are rescaled to values between [0,1] in place of [0,255].
- We can specify the number and shape of filters using Conv2D function in Keras. These filters will be convolved with image matrix to give features.
- Along with ReLu layer, a layer with MaxPooling function is used , which helps to make assumptions about features, thus reducing overfitting and also the training time.
- Once model is compiled, loss (or the difference between actual output and predicted output) is calculated using cross entropy function.
- SGD or stochastic gradient descent is used to optimize weights using backpropogation.

A. Accuracy achieved

The accuracy of classification of digits using feedforward neural networks after
five iterations is 90 percent. Whereas with the use of Convolutional neural networks, the accuracy with which images are classified is 95.63%. The accuracy achieved in various iterations have been shown in the table below:

TABLE I. ACCURACY ACGIEVED IN VARIOUS ITERATIONS

Iterations	Type of Network	Accuracy achieved(%)
5	Feedforward Neural nets	90
5	CNN	95.63
20	Feedforward Neural nets	93.76
20	CNN	99.20

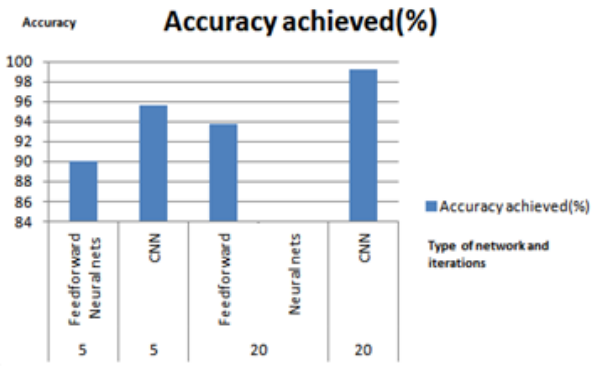

Fig.9. Accuracy achieved in both cases

IV. CONCLUSION AND FUTURE WORK

Convolutional neural networks give the best performance to classify or recognize images. In this paper, we have show the effectiveness of CNNs to classify handwritten digits. Their performance is observed to be better than feedforward neural networks. We would further work on image recognition using promising technologies of neural networks, like RNN[8] and LSTM which are finding a place in the list of emerging technologies. Also, we would extend the use of CNNs to recognize facial expressions and other biometric traits, which are used as means of identification in various applications these days.

REFERENCES

[1] Liu, D.: Advances in neural networks - ISNN 2007. Springer, Berlin (2007).

[2] Jiang, X., Pang, Y., Li, X., Pan, J., Xie, Y.: Deep neural networks with Elastic Rectified Linear Units for object recognition. Neurocomputing. 275, 1132-1139 (2018).

[3] Jaswal, D., V, S., Soman, K.: Image Classification Using Convolutional Neural Networks. International Journal of Scientific and Engineering Research. 5, 1661-1668 (2014).

[4] Classification performance analysis of MNIST Dataset utilizing a Multi-resolution Tech-nique. International Conference on Computing, Communication and Security (ICCCS). pp. 1-5. IEEE, Pamplemousses (2015).

[5] https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

[6] B. Kim and B. Zhang, "Hangul Handwriting Recognition using Recurrent Neural Networks", *KIISE Transactions on Computing Practices*, vol. 23, no. 5, pp. 316-321, 2017.

[7] P. Kumar, "Handwriting Recognition using Tensor Flow and Convolutional Neural Networks", *International Journal for Research in Applied Science and Engineering Technology*, vol. 6, no. 4, pp. 901-903, 2018.

[8] Bengio, Y.: Learning Deep Architectures for AI. Foundations and Trends® in Machine Learning. 2, 1-127 (2009).

[9] Improving training time of deep neural network with asynchronous averaged stochastic gradient descent. The 9th International Symposium on Chinese Spoken Language Processing. pp. 446-449. IEEE, Singapore (2014).

[10] Kim, M.: Multiple-concept feature generative models for multi-label image classification. Computer Vision and Image Understanding. 136, 69-78 (2015).