A Deep Learning Approach of Recognizing Handwritten Digits

Subham Pradhan 22mdsa48 Ashutosh Patra 22mdsa68

Himanshu Sekhar Panigrahi 22mdsa76
MSc.(Data Science), Silicon Tech, Bhubaneswar, Odisha, India

May 31, 2023

Abstract

Humans always wanted to make life easier by making machines do their regular chores. Although we have done more progress but to do works efficiently, we need to train machines as we do to train humans by giving examples. To simulate the process of learning and take necessary steps as per situations properly we(humans) tried to mimic the activation of biological neurons in our brain in machines by incorporating artificial neural networks (ANN). This breakthrough in human achievements enabled machines to learn almost anything. To make computers see what we humans see and as well as distinguish efficiently the convolution operation was designed for processing data with grid-like topology such as images. This version of ANN was called Convolutional Neural Networking (CNN). CNN can learn to identify the patterns in images even though patterns are unlabelled. This makes CNN suitable for image classification. CNN is nothing new as it was developed in the 1980s and induced in US postal system to detect ZIP codes but slowly lost its shine. CNN was again resurrected in 2005 in the name of deep learning and still then it is booming and an industry-leading technique in Machine Learning. It is widely used for classification as it is proved that perceptron will converge[1].

The goal of this paper is to generate a deep-learning architecture to solve the problem of recognizing handwritten digits. Deep learning neural networks have revolutionized the field of handwriting recognition, and are now able to recognize handwritten digits with over 0.99 accuracy. This is a significant improvement over previous methods and is due to the ability of deep-learning neural networks to learn to recognize patterns in handwritten characters. They do this by being trained on large datasets of handwritten digits. The more data they are trained on, the better they become at recognizing handwritten characters.

1 Introduction

The rise of deep learning has the potential to solve real-world problems efficiently. The use of the convolutional neural network(CNN) makes computers do trivial tasks like detecting objects, animals, or flowers in images as well as recognizing speeches and faces for identification. The study of CNN emerged from the study of the brain's visual cortex, and they have been in use since the 1980s. Thus, the systems trained with CNN can provide services in automatic video classification, self-driving cars, image search services, and more. In 1958 D.H. Hubel and T. Wiesel [2] performed a series of experiments on cats and in 1959 on monkeys. this gave pivotal inferences into the structure of the visual cortex for which the duo got the Nobel Prize in 1981 in medicine. They showed that many neurons in the visual cortex have a small receptive field. The receptive fields of different neurons may overlap and together they tile the whole visual field. This research paper created quite a buzz in the world and people started to think about how to artificially generate such cortex that will enable computers to see and recognize patterns to tile them up to understand the visual objects. So they come up with the idea of topology as grid-like structures create the whole image. So such a neural system was developed to slide over the grid and evaluate the pixels for understanding the label of the image. Thus the process was called Convolution and this particular version of ANN was named the convolutional neural network (CNN).

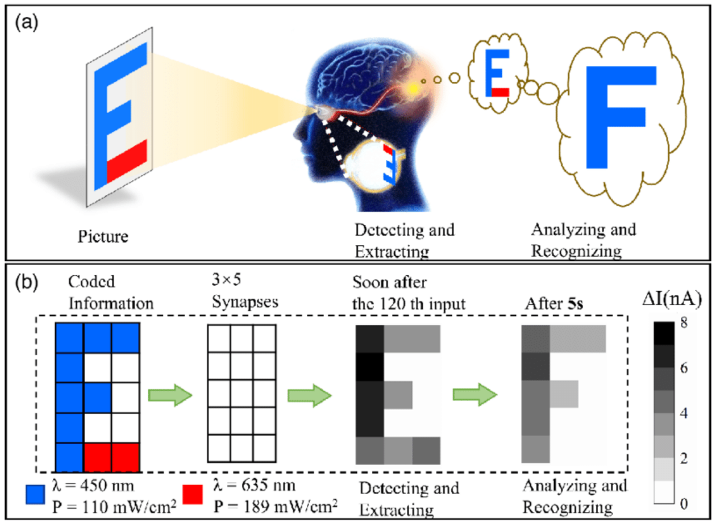

Figure 1: Biological neurons in visual cortex respond to specific patterns)

2 Literature Review

There are different approaches taken by famous scientists in the sector of character recognition and we also took our interest in research from those papers. If we summarize all those here then our initial view was on the leNet architecture by Y.Lecun et.al [3] and his paper that architecture was created in 1994 and gave us a concise idea about implementation. Our next model of interest was the paper by Alex Krizhevsky and Geoffrey Hinton et.al[4] This paper gave us an idea about deep CNN for digit recognition using the AlexNet architecture as this is a great model which achieved 99.7% pretty amazing accuracy. Then we read and studied the paper by Hinton et.al which used RNN(Recurrent NN)[5] and achieved a great result. For architecture and to achieve accuracy we followed Alexnet architecture and GoogLenet architecture although the above 2 architectures for the cifar 1000 label image classification problem those gave us a sound understanding of Deep Neural Networking.

3 Motivation for the project

On our initial days of research we created a model with the MNIST dataset with pretty good accuracy. The data was created days back in the US by taking data of the US zip code data so for a product-based and more generalized model we gathered data from various sources like the internet and wrote some on our own. Then we created a model using it.

4 Data Collection and Processing

4.1 Data Collection

As we have expressed in our motivation behind this project it was inevitable to achieve the goal without getting data in hand. We used web scrapping by which we get handwritten characters as well as some images of digits. But that number was quite small in comparison to the already existing MNIST dataset comprising a total of 70,000 rows. The number was small as some data points were skewed and somewhat flipped or rotated so instead of rectifying those individual points we added some synthetic data to increase the size and surprisingly we had more than 100,000 data points.

4.2 Data Preprocessing

An unprocessed data directly fed to the algorithm may not give promising results as it comprises anomalies, irrelevant data, etc. As we are working with images all images were not of the same dimension and colormap.

1. colormap was changed to RGBA as pixel values of some images were getting distorted while making grayscale
2. Made all data points to the dimension [28,28,4] which will be relevant to MNIST standard
3. To increase size we applied data augmentation without allowing horizontal and vertical flips and the rotation range was set up to 10 degrees.

5 Feature Works

The first model of ours was based on leNet5 architecture using MNIST which was simple and easy to implement but in that model activation functions were tanh and at the output layer the activation was RBF so we modified that hidden layer activation to ReLu and output layer to softmax and the result was promising for a small network like that.

The next model was created again using MNIST on the basis of Alexnet architecture which we tweaked and also got pretty good results. Since we got a promising accuracy of 98% with 20% validation but when the model was tested on real handwritten digits it resulted in grave mistakes e.g mostly it was confusing in recognizing 3 and 8 also in some cases for 1 and 2

This error was indicating somewhat the model needs further training so we thought of getting a large dataset created on our own for better precision. The model currently created is limited to 90% accuracy with the same 20% validation and further improvement is expected.

6 Architecture

Our architecture design is provided here which is built by getting inspiration of AlexNet with some parameter tuning and some tweaks.

6.1 Tweaks

T1. Used a bottleneck approach like GoogLeNet by using stride 1 always
T2. instead of using softmax activation at output layer, we used linear by making a change to loss.
T3. Our new loss function for compilation is
$loss = tf.keras.losses.SparseCategorigacalCrossentropy(from_logits = True)$
this code incorporated linear activation used for maximum likelihood estimation
T4. Use of dropout layer by 50% to reduce overfitting
T5. Use of Batch normalization before flattening

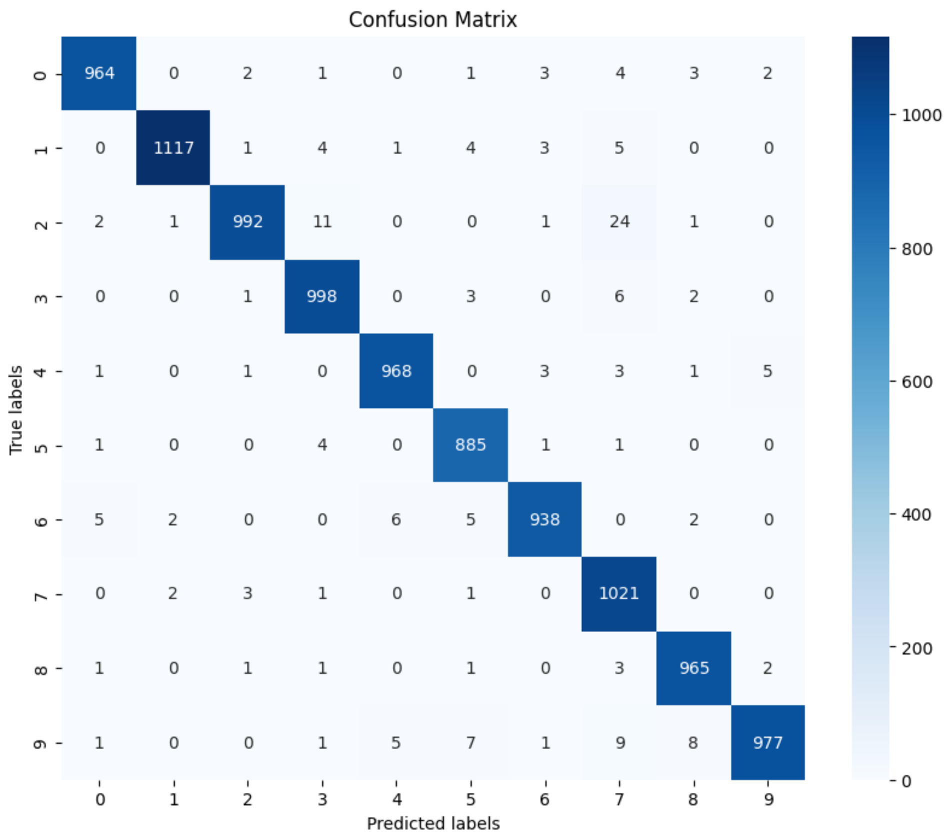

Figure 2: (Confusion matrix for the model evaluation)

Layer	Type	filters	kernel size	stride	Axtivation	Padding
Out	Fully Connected	10	n		linear	-
F13	Fully Connected	64	n		ReLu	-
F12	Fully Connected	128	n		ReLu	-
S11	Flatten	-	n			-
S10	Maxpooling	-	n			same
S9	Batch normalization	-	n			-
C8	Convolution	256	3		ReLu	same
C7	Convolution	256	3		ReLu	same
S6	Maxpooling	-	n			same
C5	Convolution	128			ReLu	same
C4	Convolution	128	3		ReLu	same
S3	Maxpooling	-	n		ReLu	same
C2	Convolution	64	3		ReLu	same
C1	Convolution	32	7		ReLu	same
In	Input	1	-			

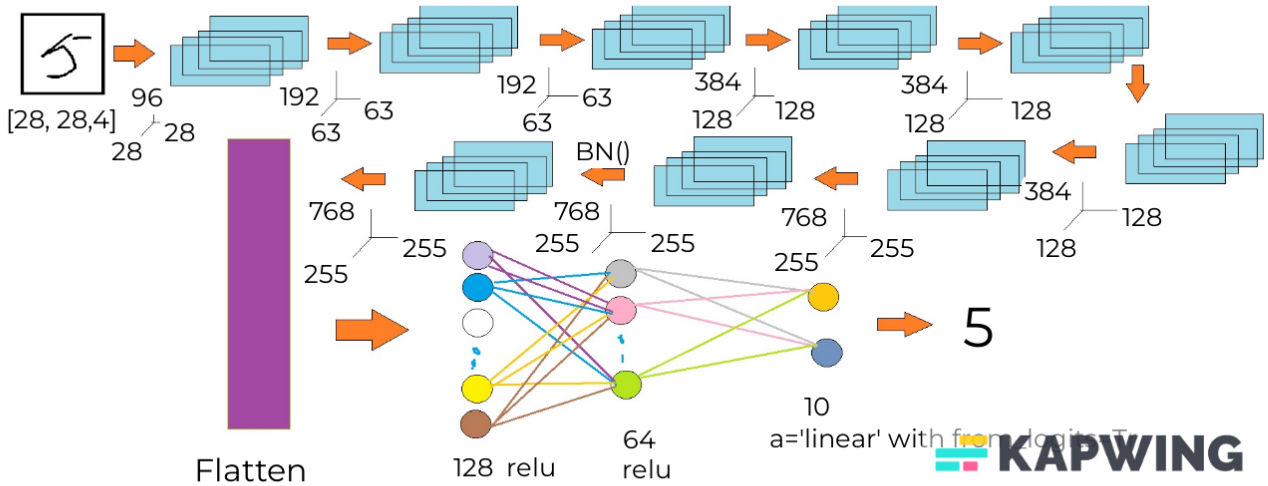

Figure 3: (Deep Neural network architecture for digit classification)

7 Future Work

1. Investigating more into deep learning architectures for better model creation
2. Doing more research and implementing Regularizations like Dropout and local response normalization
3. Looking more into the G.Hinton et.al RNN model[5]

8 References

1. Perceptron Convergence Theorem
2. "Receptive fields of single neurons in the cat's striate cortex" by D. H. Hubel and T. N. Wiesel 1959
3. "Handwritten Digit Recognition with a Back-Propagation Network " by Denker, Y. LeCun et. al
4. "Convolutional neural networks for image recognition" by Alex Krizhevsky, G. Hinton et.al
5. "A fast learning algorithm for deep belief nets" by G. Hinton et. al