RECOGNITION OF HANDWRITTEN DIGIT USING CONVOLUTIONAL NEURAL NETWORK IN PYTHON

M. Sivaranjani*1, Mr. R. Sathish Kumar*2

*1,2-Asst. Prof. Department of MCA & Krishnasamy College of Engineering & Technology, Cuddalore, India.

ABSTRACT

In recent times, with the increase of deep learning has brought a dramatic twist in the field of machine learning by making it more artificially intelligent. Deep learning is remarkably used in vast ranges of fields because of its diverse range of applications such as surveillance, health, medicine, sports, robotics, drones, etc. In deep learning, Convolutional Neural Network (CNN) is at the center of spectacular advances that mixes Artificial Neural Network (ANN) and up to date deep learning strategies. It has been used broadly in pattern recognition, sentence classification, speech recognition, face recognition, text categorization, document analysis, scene, and handwritten digit recognition. The goal of this paper is to observe the variation of accuracies of CNN to classify handwritten digits using various numbers of hidden layers and epochs and to make the comparison between the accuracies. For this performance evaluation of CNN, we performed our experiment using Modified National Institute of Standards and Technology (MNIST) dataset. Further, the network is trained using stochastic gradient descent and the back-propagation algorithm.

Keywords: Handwritten Digit Recognition, Convolutional Neural Network (CNN), Deep Learning, MNIST Dataset, Epochs, Hidden Layers, Stochastic Gradient Descent, Back-Propagation.

I. INTRODUCTION

With time the numbers of fields are increasing in which deep learning can be applied. CNN is exceptionally used in deep learning for visual imagery analyzing, object detection, face recognition, robotics, video analysis, segmentation, pattern recognition, natural language processing, image classification, spam detection, speech recognition, topic categorization, regression analysis, etc. In these arenas, the accuracies in these fields which include handwritten digits recognition using Deep Convolutional Neural Network (DCNN) have reached close to the humanoid level accomplishment. Mammalian visual systems' biological representation is the one by which the architecture of the CNN is inspired. Within the cat's visible cortical region cells are sensitized into a tiny region of the visual arena which is known as the receptive field. In 1962, it was discovered by D. H. Hubel et al. The recognition, was the primary computer vision which was motivated by the work of D. H. Hubel et al. It was introduced by Fukushima in 1980. In 1998, the framework of CNNs is designed by LeCun et al. which had seven layers of convolutional neural networks. It was then practiced with the handwritten digit's classification direct after values of pixel images. Gradient descent also back-propagation algorithm is used for training the model. In handwritten recognition digits, characters are given as input. The model can be recognized by the system. A simple Artificial Neural Network (ANN) consists of an input layer, an output layer and several hidden layers among the input and output layers. The architecture of CNN is comparable to the architecture of ANN. In every layer of ANN, quite a lot of neurons are present. The sum of all the weighted neurons in the layer converted into the input of a neuron in the next layer by the accumulation of biased value. The layers of the CNN comprise of three dimensions where all the neurons are not fully connected rather each and every neuron in the layer is associated to the local receptive field. In order to train the network, a cost function is generated which relates the output of the network with the desired output. To minimize the rate of the cost function, the signal propagates back to the system, again and again, to appraise the shared weights as well as biases in all the receptive fields. In this article, the goal is to perceive the influence of hidden layers of a CNN for handwritten digits. Different types of CNN algorithm are applied on MNIST dataset using TensorFlow, a neural network library in python. The key objective of this project is to analyze the variation of outcome results for using a different combination of hidden layers of CNN. For training, the network stochastic gradient and back-propagation algorithm are used and for testing the forward algorithm is used.
II. LITERATURE SURVEY

2.1: Handwritten digit recognition with a back-propagation network

**Author:** Y LeCun, B Boser, J Denker, D Henderson -1990

**Description:**
We present an application of back-propagation networks to handwritten digit recognition. Minimal preprocessing of the data was required, but architecture of the network was highly constrained and specifically designed for the task. The input of the network consists of normalized images of isolated digits. The method has 1% error rate and about a 9% reject rate on zipcode digits provided by the U.S. Postal Service.

2.2: Neural Network Recognizer for Hand-Written Zip Code Digits

**Author:** JS Denker, WR Gardner, HP Graf, D Henderson – 1989

**Description:**
This paper describes the construction of a system that recognizes hand-printed digits, using a combination of classical techniques and neural-net methods. The system has been trained and tested on real-world data, derived from zip codes seen on actual US Mail. The system rejects a small percentage of the examples as unclassifiable, and achieves a very low error rate on the remaining examples. The system compares favourably with other state-of-the-art recognizers.

2.3: LeNet-5, convolutional neural networks

**Author:** Y LeCun -2015

**Description:**
Convolutional Neural Networks are special kind of multi-layer neural networks. Like almost every other neural network they are trained with a version of the back-propagation Convolutional Neural Networks are designed to recognize visual patterns directly from pixel images with minimal pre-processing. They can recognize patterns with extreme variability (such as handwritten characters), and with robustness to distortions and simple geometric transformations. LeNet-5 is our latest convolutional network designed for handwritten and machine-printed character recognition. Here is an example of LeNet-5 in action.

III. PROPOSED SYSTEM

In recent times, with the increase of deep learning. This project is to observe the variation of accuracies of CNN to classify handwritten digits using various numbers of hidden layers and epochs and to make the comparison between the accuracies. This performance evaluation of CNN, we performed our experiment using Modified National Institute of Standards and Technology (MNIST) dataset.

**Module Description:**

**Module 1: Dataset Collection**

A dataset (or data set) is a collection of data, usually presented in tabular form. Each column represents a particular variable. Each row corresponds to a given member of the dataset in question. It lists values for each of the variables, such as height and weight of an object. Each value is known as a datum.

We have chosen to use a publicly-available patient’s data which contains a relatively small number of inputs and cases. The data is arranged in such a way that will allow those trained in disciplines to easily draw parallels between familiar statistical and novel ML techniques. Additionally, the compact dataset enables short computational times on almost all modern computers. Datasets are collected from Kaggle opensource website.

Here, we are using the data collection has online written numbers. We will draw the number in the text box and that drawn number will be finally, predicted. The values between 0 to 9.

**Module 2: Pre-Processing**

In this step, we use various types of pre-processing techniques to handle the missing, noisy and inconsistent data. There are a number of pre-processing techniques such as case folding dam character erase, tokenization, slang word handling, stop word removal, stemming and number handling.
The `sklearn` preprocessing package provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators. It is a process of reducing null values, noisy data, etc. Using this pre-processing method, we can able to convert our data's to structured format.

Data preprocessing involves transforming raw data to well-formed data sets so that data mining analytics can be applied. Raw data is often incomplete and has inconsistent formatting.

**Module 3: Model Implementation**

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.

CNN is a neural network that extracts input image features and another neural network classifies the image features. The input image is used by the feature extraction network. The extracted feature signals are utilized by the neural network for classification.

Modified National Institute of Standards and Technology (MNIST) is an enormous set of computer vision dataset which is extensively utilized for the purpose of different training and testing classifications. It was created from the two special datasets of National Institute of Standards and Technology (NIST) which holds binary images of handwritten digits. The training set contains handwritten digits from 250 people, among them 50% training dataset was employees from the Census Bureau and the rest of it was from high school students.

For this project we are using deep learning method using CNN for predict the numbers.

**CNN:**
- A convolutional neural network, or CNN, is a deep learning neural network sketched for processing structured arrays of data such as portrayals.
- CNN are very satisfactory at picking up on design in the input image, such as lines, gradients, circles, or even eyes and faces.
- This characteristic that makes convolutional neural network so robust for computer vision.

**How Do Neural Networks work?**
- A network comprises layers of neurons. It associates each neuron with a random number called the bias.
- Neurons present in each layer transmit information to neurons of the next layer over channels.
- These channels are associated with values called weights.
- The weights, along with the biases, determine the information that is passed over from neuron to neuron.
- Neurons from each layer transmit information to neurons of the next layer.
- The output layer gives a predicted output.

Let’s go ahead and build a neural network to predict bike prices based on a few of its features.
- The input features such as cc, mileage, and abs are fed to the input layer.
- The hidden layers help in improving output accuracy.
- Each of the connections has a weight assigned to it. The neuron takes a subset of the inputs and processes it.

**Module 4: Flask**

Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Pocco. Flask is based on the Werkzeug WSGI toolkit and the Jinja2 template engine. Both are Pocco projects.

Flask is a web framework, it’s a Python module that lets you develop web applications easily. It’s having a small and easy-to-extend core: it’s a microframework that doesn’t include an ORM (Object Relational Manager) or such features. It does have many cool features like URL routing, template engine. It is a WSGI web app framework A Web Application Framework or a simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low-level details such as protocol, thread management, and so on.
WSGI

The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications.

What is Flask used for?

Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier. It gives developers flexibility and is a more accessible framework for new developers since you can build a web application quickly using only a single Python file.

Module 5: Predicted Output

Finally, we will get the predicted values of accuracy. We can draw any number between 0 to 9. That value will be predicted.

IV. SYSTEM ARCHITECTURE

For the recognition of handwritten digits, a seven-layered CNN is designed with one input layer, five hidden layers, and one output layer as demonstrated in figure 1.

**Fig. 1**: A seven-layered CNN Architecture for handwritten digit identification

**MNIST DATASET**

Modified National Institute of Standards and Technology (MNIST) is an enormous set of computer vision dataset which is extensively utilized for the purpose of different training and testing classifications. It was created from the two special dataset of National Institute of Standards and Technology (NIST) which holds binary images of handwritten digits. The training set contains handwritten digits from 250 people, among them 50% training dataset was employees from the Census Bureau and the rest of it was from high school students.

**Fig. 2**: Dataset collection
V. DATA FLOW DIAGRAM

Fig. 3: DFD Diagram

VI. CONCLUSION

In this section, the variations of accuracies for handwritten digit were observed for 15 epochs by varying the hidden layers. The accuracy curves were generated for the six cases for the different parameter using CNN MNIST digit dataset. The six cases perform differently because of the various combinations of hidden layers. The layers were taken randomly in a periodic sequence so that each case behaves differently during the experiment. The maximum and minimum accuracies were observed for different hidden layers variation with a batch size of 100. Among all the observation, the maximum accuracy in the performance was found 99.21% for 15 epochs in case 2 (Conv1, pool1, Conv2, pool2 with 2 dropouts). Such type of higher accuracy rate in digit recognition will help to speed up the performance more competently. However, the minimum accuracy among all observation in the performance was found 97.07% in case 6 (Conv1, pool1, Conv2, pool2 with 1 dropout). Moreover, among all the cases, the total highest test loss is 0.049449, which is found in case 3 excluding dropout and the total lowest test loss is 0.026303 found in case 2 including dropout.

VII. REFERENCE