International Journal for Modern Trends in Science and Technology, 6(8S): 160-165, 2020

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTSTCIET31
Available online at: http://www.ijmtst.com/ncracse2020.html

Data Classification with Deep Learning Using Tensor Flow

Dr.Keshetti Sreekala | A.S.Shalini

[1]Assistant Professor, Dept of CSE, Mahatma Gandhi Institute of Technology, Hyderabad, Telangana, India
[2] UG Student, Mahatma Gandhi Institute of Technology, Hyderabad, Telangana, India

To Cite this Article
Dr.Keshetti Sreekala and A.S.Shalini, "Data Classification with Deep Learning Using Tensor Flow", *International Journal for Modern Trends in Science and Technology*, 6(8S): 160-165, 2020.

Article Info
Received on 16-July-2020, Revised on 15-August-2020, Accepted on 25-August-2020, Published on 30-August-2020.

ABSTRACT

Handwritten digits recognition is method of recognizing and classifying handwritten digits from 0 to 9. We have implemented recognizing hand written digits in a better and accurate approach to identify manually written digits from 0 to 9. For this we have used a multilayer sustain forward system called Convolutional network is taken into consideration. We have used Tensorflow, one of the most popular deep learning libraries to classify MNIST dataset, which is frequently used in data analysis studies. By Using Tensorflow, CNN has achieved an accuracy of 99% on data. The functions used for implementing the digit recognition are Rectified Linear Unit (ReLu), Hyperbolic Tangent (tanH), Exponential Linear Unit (eLu), sigmoid, softplus and softsign. In this Study, Convolutional Neural Network (CNN) and SoftMax classifier are used as deep learning artificial neural network. The results show that the most accurate classification rate is obtained with the use of the ReLu activation function.

KEYWORDS: *Deep Learning, RELU, TensorFlow, CNN*

INTRODUCTION

Handwriting recognition has become one of the best directions in the field of image processing and pattern recognition. New and new technologies are being proposed everyday for recognizing the handwritten characters. The recognition method involves acquiring the image from the user and then transmitting it to a server where the recognition process takes place whereas in offline recognition all the necessary processes takes place on the users system. The problem we predicted we would face in this digit classification problem was the similarity between the digits like 1 and 7, 5 and 6, 3 and 8, 9 and 8 etc. Also many people write the same digit in many different ways - the digit'1' is written as'1', 'l' or '|'. The uniqueness in the handwriting of different individuals also influences the formation and appearance of the digits.

In handwritten recognition digits, digits are given as input. The model can be recognized by the system. A simple convolutional neural network (CNN) has an input layer, an output layer and some hidden layers between the input and output layer. CNN has a very similar architecture as ANN. There are several neurons in each layer in ANN. The sum of all the n layer becomes the input of a neuron of the next layer adding a biased value. In CNN the layer has three dimensions. Here all the neurons are not fully connected. Instead, every neuron in the layer is connected to the local receptive field. A cost function generates in order to train the network. It compares the output of the network

with the desired output. The signal propagates back to the system, again and again, to update the shared weights and biases in all the receptive fields to minimize the value of cost function which increases the network's performance. The goal of this article is to observe the influence of hidden layers of a CNN for handwritten digits.

The Classifier to be used should able to classify digit with high accuracy and should take less training time during classification. We have applied a different type of Convolutional Neural Network algorithm on Modified National Institute of Standards and Technology (MNIST) dataset using Tensorflow, a Neural Network library written in python. The main purpose of this paper is to analyze the variation of outcome results for using a different combination of hidden layers of Convolutional Neural Network.

RELATED WORK

Hasbi Ash Shiddieqy et al.[1] proposed "Implementation of Deep-Learning based Image Classification on Single Board Computer", Deep learning, as a new subset from machine learning, refers to deeply linear combinations of several simple layers or functions parameterized by variables. In this paper, a deep-learning algorithm based on convolutional neural network is implemented using python and tflearn for image classification, in which two different structures of CNN are used, namely with two and five layers and It conclude that the CNN with higher layer performs classification process with much higher accuracy.

Rui Wang et al.[2] proposed "Blur Image Classification based on Deep Learning", shown that along with the layers of a CNN, an image can be represented at different levels of abstraction from low to high-level features, which can provide greater robustness to intra-class variability and the presence of blurred spots and illumination variation. In case of blur situations, described the performance of a Simplified-Fast-AlexNet (SFA) to classify different types of blurred images, such as defocus, Gaussian, and motion blur. In this paper, a convolution neural network (CNN) of Simplified-Fast-Alexnet (SFA) based on the learning features is proposed for handling the classification issues of defocus blur, Gaussian blur, haze blur and motion blur four blur type images. The experiment results demonstrate that the performance of classification accuracy of SFA, which is 96.99% for simulated blur dataset and 92.75% for natural blur dataset, is equivalent to Alexnet and superior to other classification methods.

Hongzheng Fang et al.[3] proposed "Complex System Fault Diagnostic Method Based on Convolutional Neural Network", Deep learning is characterized by multi-level learning to obtain different abstraction levels of raw data, thus improving the accuracy of the tasks such as classification and prediction. In this paper, It brings a new idea of the complex system fault diagnostics and prognostics. Combining the characteristics of complex system test data and the advantages of deep learning, a fault diagnostics method based on convolutional neural network is proposed, including preprocessing, model training and optimization. Then a complex system fault diagnostic algorithm platform based-on deep learning method is realized. The simulation method of an aero-engine gas path test proves that the proposed method has good feasibility and effect, can fully utilize the advantages of deep learning, and is more suitable for characterizing the complex and varied characteristics hidden inside the complex system data. It can provide the support for the design, test and management of the complex system, and improve the safety and effectively reduce the life cycle costs of the complex system.

Rosanna C. Ucat et al.[4] proposed "Postharvest Grading Classification of Cavendish Banana Using Deep Learning and Tensorflow", In the Philippines' banana export industry, the postharvest classification of Cavendish banana is an important factor which affects one of the country's revenue. The present set up uses a quality inspector and is solely dependent on its visual capability. This may result in an error and inconsistent results grading classification. In this paper, The use of image processing with deep learning approach to classify Cavendish banana grade by the basis of its standard requirement. The final classification in all trained, validation, and test data showed above 90% accuracy.

Ye Tao et al.[5] proposed "Deep Learning in Photovoltaic Penetration Classification", Solar panels installation increases dramatically in recent years, especially in the areas with rich illumination. In this paper proposed a deep learning based algorithm to differentiate photovoltaic events from other grid events, and it conclude that a deep convolutional neural network

can achieve higher classification accuracy than a fully connected model. The proposed classification framework can identify event types and help operators adopt better strategies. In addition, this framework is able to exploit the potential of the historical data since deep learning can reduce the error rate as the dataset increases In contrast, we also present a shadow-learning framework using fully-connected network for comparison. The photovoltaic event data used in the experiments are generated from a smart inverter model.

DESIGN METHODOLOGY

A. Architecture

The images contained in the MNIST data set are stored in the IDX file format. This is the simple format for vectors and multidimensional matrices of various digital types. There are files in the dataset, including image and label information for testing and training. The parameters for the Tensorflow Library for deep learning in the second phase are specified. For this study, 100,000 iterations were processed. As the each image is composed of 28x28 pixels, the neuron input parameter value is taken as 784. Output class is selected as 10, because the digits are expected to be within digits from 0 to 9.

As shown in the Fig in determining the weights in the third stage, 5x5 convolutional layer and 32 outputs were selected. Then a 5x5 convolutional layer and 64 outputs were applied again. In the fully connected layer, values of 3136 and 1024 are selected as input and output, respectively. In the output layer, the output value in the previous layer is 1024, which is input and taken as 10 outputs with the total number of classes. Bias values will be taken as output values in the weight layers.

In the fourth step, the performance of the classifier will be measured during the selected iteration by selecting different activation functions. The accuracy rates achieved by using different activation functions are different. The accuracy values obtained according to the iteration numbers of the ReLu activation function obtained as the best result.

Different activation functions were selected in the system to test the accuracy of the classification of the system. ReLu, eLu, tanH, sigmoid, softPlus and softsign activation functions were used for this purpose. SoftMax is used as the classifier function. It has been observed that the best accuracy is achieved when the ReLu activation function is selected. In this ReLu activation function, 98.43% classification accuracy is obtained on the test data.

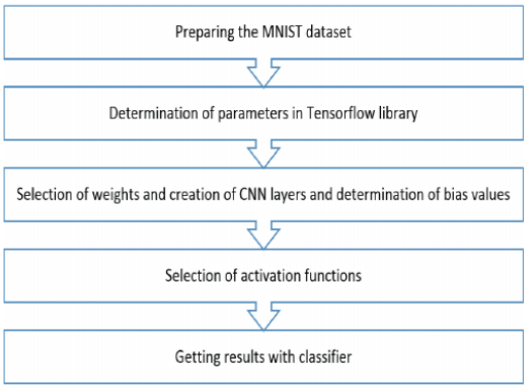

Fig :Architecture of Data Classification with Deep learning Using Tensor Flow

B. Modules of Data Classification with Deep learning Using Tensor Flow

The module is composed of the following sections:

- Input MNIST Data
- Building Network Architecture
- Fully Connected Layer

Input MNIST Data:

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. The MNIST database contains 60000 training set and 10000 testing set handwritten digits ranging from 0 to 9 .Each digit is normalized and centered in a gray-level image with size 28X28. A total of 10000 samples will be taken both for training and testing set respectively. Pre-processing may be required for pre-captured photo sequences. The dimensions are matched and made sure that each image contains only one digit. We use MNIST data sets for training, recognition of handwritten digits. MNIST is a simple computer vision dataset. The pre-processing additionally characterizes a smaller portrayal of the example. Binarization changes over a gray scale image into a binary image. The initial approach to the training set images that are to be processed in order to reduce the data, by thresholding them into a binary image.

Building Network Architecture:

In this I define the models to be used to build a convolutional neural network. Here, we use the

Sequential class from Keras to build the network. In this network, the three layers are present sets of layers “CONV =>ReLU=> POOL”.

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 .

ReLU Function: It is called as rectified linear unit and its function is to ensure that output is positive as negative values are defined to zeros. 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.

Pooling Layer: The pooling layer gets the data from the ReLU function and down-samples the steps in the 3D tensor. Images are again given input into the second layers and this process continue until we get to a smallest set of pixels from which we can classify the digit.

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 and shown in the output. we can add more number of layers but adding more layers might affect the accuracy of the system. since it uses multiple layers it is called a Deep Learning system.

DESIGN OF DATA CLASSIFICATION WITH DEEP LEARNING USING TENSOR FLOW

As shown from the below Fig, it describes that it is the simplest is representation of a user's interaction with the system that shows the relationship between the user and different use cases in which the user is involved. The use case will identify the different types of users of a system and the different use cases .

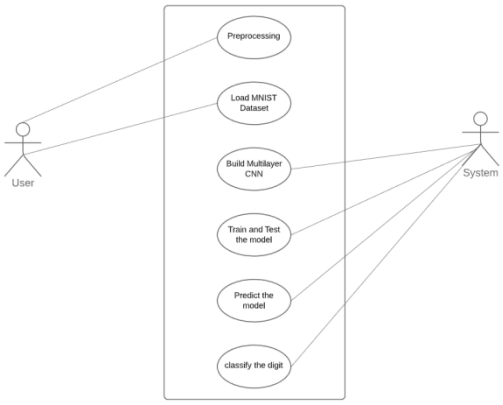

Fig: Use case diagram of data classification with deep learning using tensor flow

As shown from the below Fig, it describes i.e, it describes the activity diagram of the process which is involved to classify the hand written digits using machine learning algorithm and Training and testing is done next prediction of labels id done finally we will get the recognized digit .

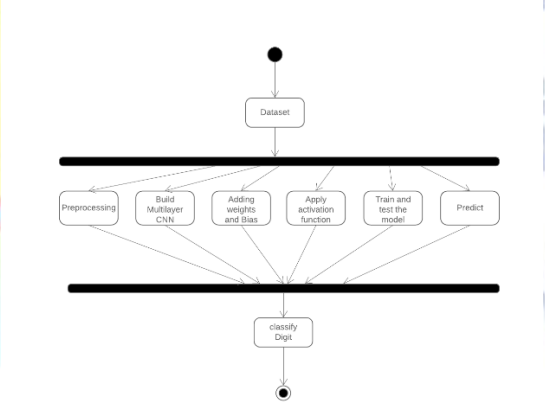

Fig: Activity diagram of data classification with deep learning using tensor flow

As shown from the below Fig, it describes i.e., the user tries to import the modules and import the dataset and Build the multilayer CNN and add the weights and bias and next selection of activation function ReLu is used next train and test the model later predict the result.

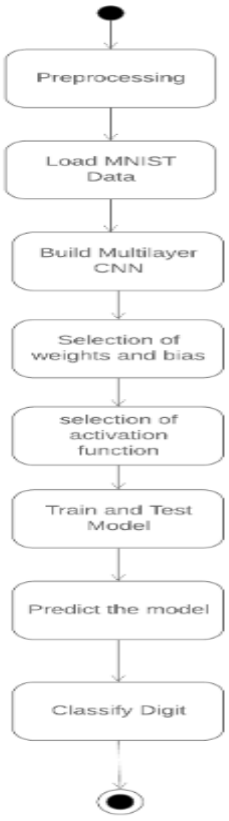

Fig: State diagram of data classification with deep learning using tensor flow

RESULTS

As shown from the below Fig, we are building the Multilayer CNN which takes the input as 28*28 image.

```
w1 = tf.Variable(tf.truncated_normal([PATCH, PATCH, CHANNELS, DEPTH], stddev=0.1))
b1 = tf.Variable(tf.zeros([DEPTH]))
w2 = tf.Variable(tf.truncated_normal([PATCH, PATCH, DEPTH, 2*DEPTH], stddev=0.1))
b2 = tf.Variable(tf.constant(1.0, shape=[2*DEPTH]))
w3 = tf.Variable(tf.truncated_normal([WIDTH // 4 * WIDTH // 4 * 2*DEPTH, HIDDEN], stddev=0.1))
b3 = tf.Variable(tf.constant(1.0, shape=[HIDDEN]))
w4 = tf.Variable(tf.truncated_normal([HIDDEN, LABELS], stddev=0.1))
b4 = tf.Variable(tf.constant(1.0, shape=[LABELS]))

def logits(data):
    # Convolutional Layer 1
    x = tf.nn.conv2d(data, w1, [1, 1, 1, 1], padding='SAME')
    x = tf.nn.max_pool(x, [1, 2, 2, 1], [1, 2, 2, 1], padding='SAME')
    x = tf.nn.relu(x + b1)
    # Convolutional Layer 2
    x = tf.nn.conv2d(x, w2, [1, 1, 1, 1], padding='SAME')
    x = tf.nn.max_pool(x, [1, 2, 2, 1], [1, 2, 2, 1], padding='SAME')
    x = tf.nn.relu(x + b2)
    # Fully connected Layer
    x = tf.reshape(x, (-1, WIDTH // 4 * WIDTH // 4 * 2*DEPTH))
    x = tf.nn.relu(tf.matmul(x, w3) + b3)
    return tf.matmul(x, w4) + b4

# Prediction:
tf_pred = tf.nn.softmax(logits(tf_data))
```

Fig: Building CNN

The below Fig,is after training the model, start the session and run the session for the number of iterations

```
ss = ShuffleSplit(n_splits=STEPS, train_size=BATCH)
ss.get_n_splits(train_data, train_labels)
history = [(0, np.nan, 10)] # Initial Error Measures
for step, (idx, _) in enumerate(ss.split(train_data,train_labels), start=1):
    fd = {tf_data:train_data[idx], tf_labels:train_labels[idx]}
    session.run(tf_step, feed_dict=fd)
    if step%500 == 0:
        fd = {tf_data:valid_data, tf_labels:valid_labels}
        valid_loss, valid_accuracy = session.run([tf_loss, tf_acc], feed_dict=fd)
        history.append((step, valid_loss, valid_accuracy))
        print('Step %i \t Valid. Acc. = %f \n'%(step, valid_accuracy))
```

```
Step 500        Valid. Acc. = 94.300003
Step 1000       Valid. Acc. = 97.000000
Step 1500       Valid. Acc. = 96.100006
Step 2000       Valid. Acc. = 97.599998
Step 2500       Valid. Acc. = 97.299995
Step 3000       Valid. Acc. = 97.500000
Step 3500       Valid. Acc. = 97.699997
Step 4000       Valid. Acc. = 98.099998
Step 4500       Valid. Acc. = 98.099998
Step 5000       Valid. Acc. = 98.299995
Step 5500       Valid. Acc. = 97.799995
Step 6000       Valid. Acc. = 98.500000
Step 6500       Valid. Acc. = 97.399994
Step 7000       Valid. Acc. = 97.599998
Step 7500       Valid. Acc. = 98.099998
Step 8000       Valid. Acc. = 97.699997
```

Fig: Run the session

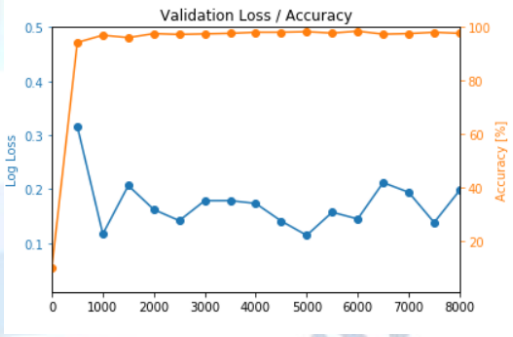

Fig:Visualizing the Training Accuracy

The above Fig,shows validation accuracy with varying number of iterations. There is a parallel increment in accuracy of test as well as validation data respectively.

CONCULSION

In this hand written digit recognition we have discovered the solution for MNIST handwritten digit recognition problem using deep learning models developed in Python using the Tensor flow library that are capable of achieving excellent results. By Using the Tensorflow, CNN has

achieved an accuracy of 99.4% on train set whereas 98.4% on the test data. The key parameters that are included are Building the multilayer CNN and the selection of best activation function that is ReLu for this problem. The results show that the most accurate classification rate is obtained using the ReLu activation function. The result of our work shows that the maximum accuracy 99.2% was obtained in MNIST dataset using deep learning technique and an appropriate learning rate at 8000 iterations. It is observed that accuracy slowly starts decreasing or remains constant after 8000 iterations. Finally we have learnt how to apply and implement a handwritten digit recognition system by using convolutional neural network and also learnt how to improve the performance to give the high recognition accuracy.

REFERENCES

[1] Hasbi Ash Shiddieqy, Farkhad Ihsan Hariadi, Trio Adiono "Implementation of Deep-Learning based Image Classification on Single Board Computer", International Symposium on Electronics and Smart Devices (ISESD), ISBN 978-1-5386-2779-2, pp. 133-137, 2017.

[2] Rui Wang, Wei Li, JinZhong Wu "Blur Image Classification based on Deep Learning", IEEE International Conference on Imaging Systems and Techniques (IST), 2017, DOI: 10.1109/IST.2017.8261503 ·

[3] Hongzheng Fang "Complex System Fault Diagnostic Method Based on Convolutional Neural Network", Prognostics and System Health Management Conference, PHM-Paris-2019, pp 150-155.

[4] Rosanna C. Ucat, Jennifer C. Dela Cruz "Postharvest Grading Classification of Cavendish Banana Using Deep Learning and Tensorflow", International Symposium on Multimedia and Communication Technology (ISMAC), 2019.

[5] Ye Tao, Ming Zhang, Mark Parsons "Deep Learning in Photovoltaic Penetration Classification", IEEE Power & Energy Society General Meeting, ISBN 978-1-5386-2213-1, pp. 1-5, 2017.

[6] A. Dutta and A. Dutta, Handwritten digit recognition using deep learning, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 6, no. 7, July 2017.

[7] Wu, Ming & Zhang, Zhen. (2019). Handwritten Digit Classification using the MNIST Data Set.

[8] Mishra, Shashank&Malathi, D &Senthilkumar, K. (2018). DIGIT RECOGNITION USING DEEP LEARNING

[9] Mahmoud M. Abu Ghosh, Ashraf Y. Maghari, "Promising Electronic Technologies" (ICPET), 2017 International Conference on, Comparative Study on Handwriting Digits recognition Using Neural Networks, 16-17 Oct. 2017.

[10] Hayder M. Albeahdili, Haider A Alwzwazy, Naz E. Islam, "Robust convolutional neural networks for image recognition", (IJACSA), pp 10-21, Volume 6, Issue 11, 2015.