A novel method for Handwritten Digit Recognition

ABSTRACT

This paper proposed a simple neural network approach towards handwritten digit recognition sing convolution. with machine learning algorithms like KNN,SVM/SOM, recognizing digits is considered as one of the unsolvable tasksdue to its distinctiveness in the style of writing. In this paper Convolution Neural Networks are implemented with anMNIST dataset of 70000 digits with 250 distinct forms of writings. The proposed method achieved 98.51% accuracy forreal-world handwritten digit prediction with less than 0.1 % loss on training with 60000 digits while 10000 under validation.

Key words

Convolution neural networks, MNIST dataset, Tensor Flow, OCR, Segmentation, ross in

1.INRODUCTION

Advancements in the field of computer vision using deep neural networks attract attention; thus, many A.I. practitioners are moving towards it.[1] One of the influencing projects that opted for deep learning is OCR (ObjectCharacter Recognition). OCR is a mechanism that converts printed or documented letters into encoded text. IntuitivelyOCR tool scans a document to extract the information and store it in a digital document. There are two major ways ofimplementing OCR, one by recognizing the patterns in thecharacters, and the other one is through segmentation.[1][10]Either way, this problem comes under the radar of machine learning. Handwritten digit recognition (HDR) is a snippet of

the pixels, and the weight carried by that neuron is calledactivation. The last layer, aka the output layer, contains neurons equal to the number of target classes; in case of the handwritten digit, recognition number of categories are ten ranging from 0 to 9 with probability values.[3][4] The number of input neurons and output neurons depends on the task, but the hidden layers are arbitrary and are not dependent on the task. That's why they are hidden layers, but the propagations between the hidden layers depend on the activation of the previous layers.[11] [12] Intuitively, the pattern of activations in the presentation layer causes some specific patterns in the next layer; thus,

the highest activation neuron is the network's choice of class. In this paper, we are implementing convolution neural network architecture with relu and sigmoid activation functions to predict a real-world handwritten digit by training the network with the MNIST dataset.

2.Neural network

Many techniques have been developed to recognize handwritten digits; most of the A.I. practitioners use this to test their model's performance. In the past decades, a segmentation-based approach was used to solve this problem later with the advancements in machine learning a segmentation less approach was introduced. Even though the implementation changes, the issue still remains the same and open for anyone to solve. Bailing Zhang [5] utilizes the ASSOM technique to provide numerical stability that can predict precise digits. Their main idea was to use the SOM clustering algorithm with auto encoder neural networks in a nonlinear approach. The modularity of SOM helped in extracting several features in a digit. For each digit, individual ASSOM was constructed and compared with ten several construction-related errors to minimize the misclassification. Their model shows promising results, even with small training samples. Sa proposed a technique for handwritten numerical strings of arbitrary length recognition using SVM and PCA, addressing the major challenge in word detection, which is overlapping characters. Their method uses hybrid PCA called PCAN et for segmentation and SVM for segmentation classification together called PCA-SVMN et. Their experiment shows high efficiency in recognizing unknown handwritten number classification without any segmentation method applied [15] Yue Yin; Wei Zhang [1] have concluded that out of all neural network implementations CNN method is valid for OCR based image classification systems. They claimed that OCR had become a preliminary technique in the field of computer vision; they state that if an image classification model performs well in OCR, then it can be used for any image classification

3.1 Pre-Processing

loading the data, we separated the data into X and y where X is the image, and y is the label corresponding to X. As shown in figure 3, the first layer/input layer for our model is convolution. Convolution takes each pixel as a neuron, so we need to reshape the images such that each pixel value is in its own space, thus converting a 28x28 matrix of greyscale values into 28x28x1 tensor. With the right dimensions for all the images, we can split the images into train and test fofurther steps [19][20][21].

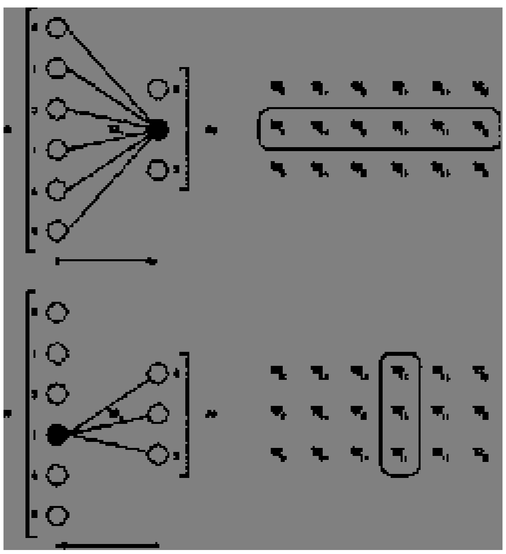


Figure 3.4 Forward and Backward propagation.

3.2 data en codion

Ross-categorical entropy as loss function; we have to specify the network that the given labels are categorical in nature.

3.3 Model Construction

After data encoding, the images and labels are ready to be fitted into our model [22] [23]. Summary of the model can be seen in Figure 4 Figure 3: Proposed model Our model is composed of feature extraction with convolution and binary classification. Convolution and max-pooling are carried out to extract the features in the image, and a 32 3x3 convolution filters are applied to a 28x28 image

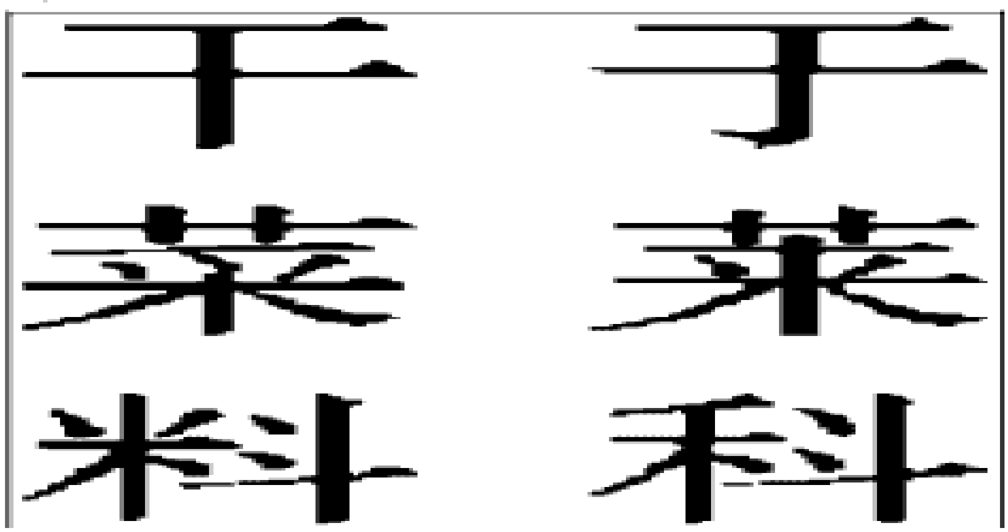
followed by a max-pooling layer of 2x2 pooling size followed by another convolution layer with 64 3x3 filters. In the end, we obtain 7x7 images to flatten. Flatten layer will flatten the 7x7 images into a series of 128 values that will be mapped to a dense layer of 128 neurons that are connected to the ca4.1 Accuracy score Our model stopped training at the 2. nd epoch as it reached 98.21% training accuracy and 98.51% validation accuracy with 5% tr ining loss and 4% validation loss. The progression of accuracy and loss are represented in Figure 5. igure 5: Loss and Accuracy Learning Curves From the above curve, we can observe that the loss and accuracy are cooperatively changed at every fold during k-fold cross-validation. Before two folds, efficiency almost reached 98%, and that's why the number of iterations stopped at the 2 nd epoch. Stability inaccuracy score can be observed from 2nd iteration.

4.1 Accuracy score Our model stopped training at the 2

nd epoch as it reached 98.21% training accuracy and 98.51% validation accuracy with 5% training loss and 4% validation loss. The progression of accuracy and loss are represented in Figure 5 igure 5: Loss and Accuracy Learning Curves From the above curve, we can observe that the loss and accuracy are cooperatively changed at every fold during k-fold cross-validation. Before two folds, efficiency almost reached 98%, and that's why the number of iterations stopped at the 2 nd epoch. Stability inaccuracy score can be observed from 2nd iteration.tegorical output layer of 10 neurons. N systems.

4.2 Training & Validation

After building the model [24], we compiled a model with adam optimizer and particular cross-entropy loss function, which are standard in making a convolution neural network. Once the model is successfully assembled, then we can train the model with training data for 100 iterations, but as the number of iteration increases, there is a chance for ver fitting Therefore we limit the training up to 98% accuracy, as we are using real-world data for prediction, test data was used to validate the model [16].



5 Model Evaluation & Prediction

For real-world image classification prediction, we need to do a little image pre-processing on the real-world images as model training was done with greyscale raster images. The steps of image pre-processing are,

1. Loading image

2. Convert the image to greyscale

3. Resize the image to 28x28

4. Converting the image into a matrix form

5. Reshape the matrix into 28x28x1

After pre-processing, we predict the label of the image by passing the pre-processed image through the neural network. The output we get is a list of 10 activation values 0 to 9, respectively. The position having the highest value is the predicted label for the image [18].

6. RESULTS AND DISCUSSION

Our model is built to work on real-world data, and real-world images are not even close to MNIST raster images, a lot of pre-processing was done to make a real image to look like a raster image.

7. CONCLUSION

The performance of CNN for handwritten recognition performed significantly. The proposed method obtained 98% accuracy and is able to identify real-world images as well; the loss percentage in both training and evaluation is less than 0.1, which is negligible. The only challenging part is the noise present in the real-world image, which needs to look after. The learning rate of the model is much dependent on the .nuber of dense neurons and the cross-validation measure.