PROJECT REPORT

Abdullah | 19-40521-1

1. Abstract

Handwritten digit recognition is an intricate assignment that is vital for developing applications, in computer vision digit recognition is one of the major applications. Human handwriting differ person to person. The target of this project to create a model to gain 98% or greater accuracy for Keras mnist dataset using CNN (Convolutional Neural Network) and Adam, SGD, RMSprop optimizer.

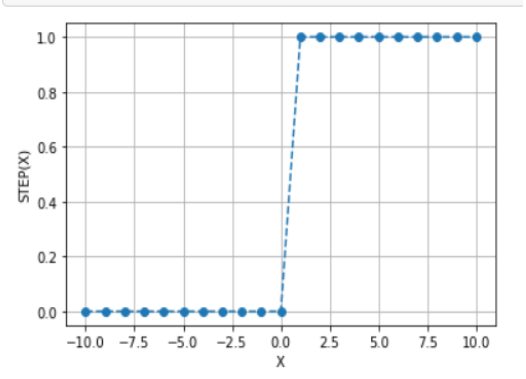
2. Introduction

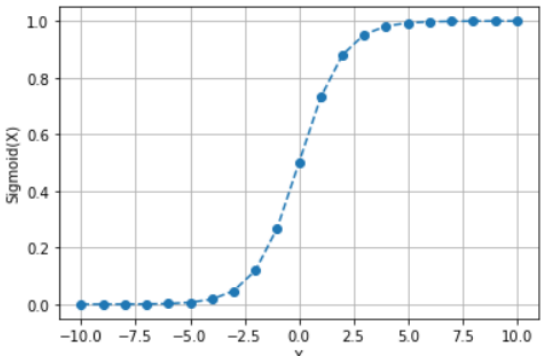
Artificial Intelligence has been witnessing a monumental growth in bridging the gap between the capabilities of humans and machines. Convolutional Neural Network (CNN) is deep learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. CNN has one or more convolutional layers and are used mainly for image processing, classification, segmentation and also for other auto correlated data. The CNN follows a hierarchical model which works on building a network, like a funnel, and finally gives out a fully-connected layer where all the neurons are connected to each other and the output is processed. The input and weight are represented in matrix manner like below:

$$\begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} W_{11} * x_1 + W_{12} * x_2 \\ W_{21} * x_1 + W_{22} * x_2 \end{bmatrix}$$
$$\Rightarrow \begin{bmatrix} W_{11} * x_1 + W_{12} * x_2 \\ W_{21} * x_1 + W_{22} * x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$
$$= \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

Figure 0: Matrix Representation of Weight and Input

This matrix can be calculated by one single operation using GPU. After calculation matrix multiplication bias is added. Then the output goes through activation function to make a range between some values. Step, Sigmoid, Tanh, ReLu, PReLu and Softmax are well known activation function. The process continues from input layer to hidden and till output layer.





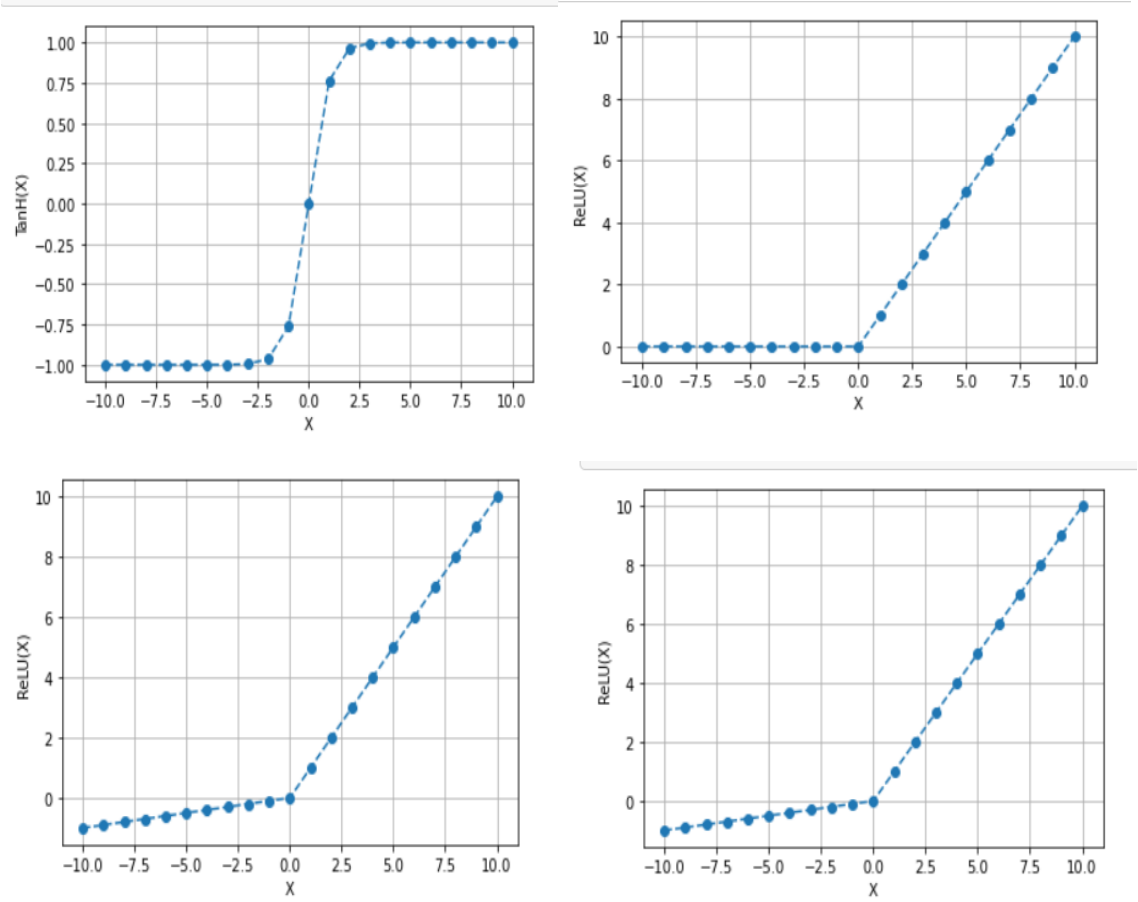


Figure 1: Graph of Different Types of Activation Function

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It is a python library and it can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow is a symbolic math library based on dataflow and differentiable programming. By using TensorFlow can create powerful model for deep learning easily.

The MNIST (Modified National Institute of Standards and Technology) database of handwritten digits, has a training set of 60,000 greyscale hand written images, and a test set of 10,000 greyscales hand written images. Each image contains 28*28 pixels where each pixel has a value from 0 to 255.

The model used for this project is a sequential model. This model has 4 layers such as Conv2D layers, MaxPooling2D layers, Flatten layer and Dense layers. Softmax activation function used in the output layer and ReLu used in all others layer. After creating the model, the model is compiled with 3 different optimizer such as Adam, RMSProp, SGD.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 24, 24, 32)        832
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 12, 12, 32)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 10, 10, 64)        18496
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
flatten_2 (Flatten)          (None, 1600)              0
_________________________________________________________________
dense_4 (Dense)              (None, 64)                102464
_________________________________________________________________
dense_5 (Dense)              (None, 10)                650
=================================================================
Total params: 122,442
Trainable params: 122,442
Non-trainable params: 0
_________________________________________________________________
```

Figure 2: Model Summary

3. Results

Result for Adam Optimizer:

Train Accuracy	Train Loss	Validation Accuracy	Validation Loss	Test Accuracy
99.76%	0.75%	98.91%	5.37%	98.99%

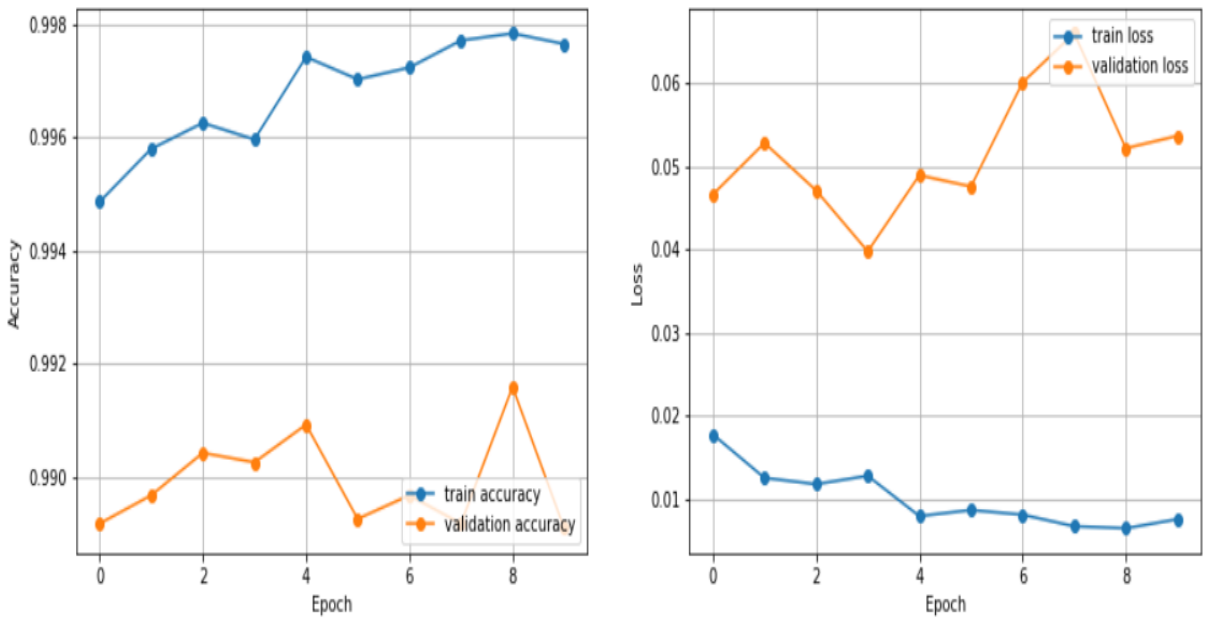


Figure 3: Graph for Adam Optimizer

Result for SGD Optimizer:

Train Accuracy	Train Loss	Validation Accuracy	Validation Loss	Test Accuracy
100%	134 x 10 ^ -4	98.28%	5.62%	99.19%

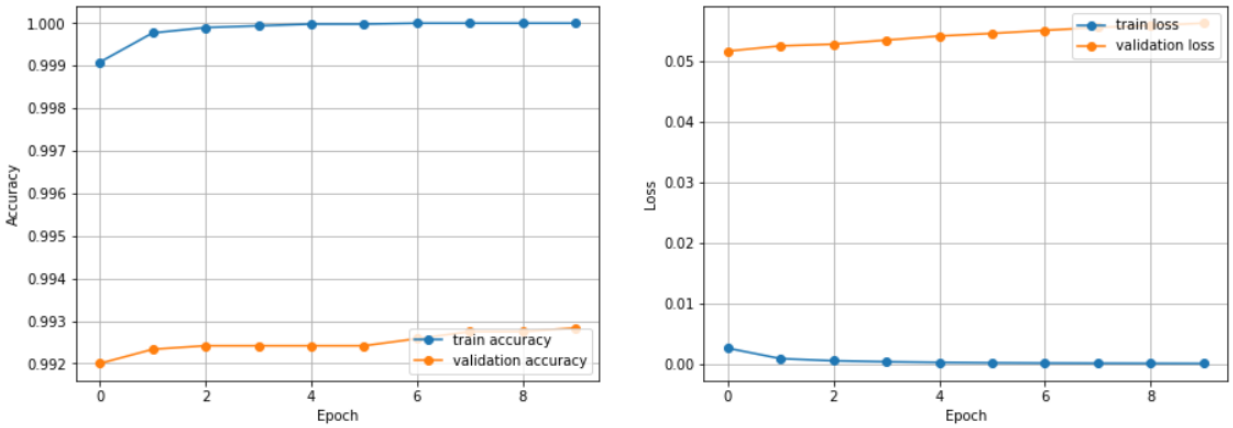


Figure 4: Graph for SGD Optimizer

Result for RMSProp Optimizer:

Train Accuracy	Train Loss	Validation Accuracy	Validation Loss	Test Accuracy
100%	150 x 10 ^ -4	99.20%	12.74%	99.22%

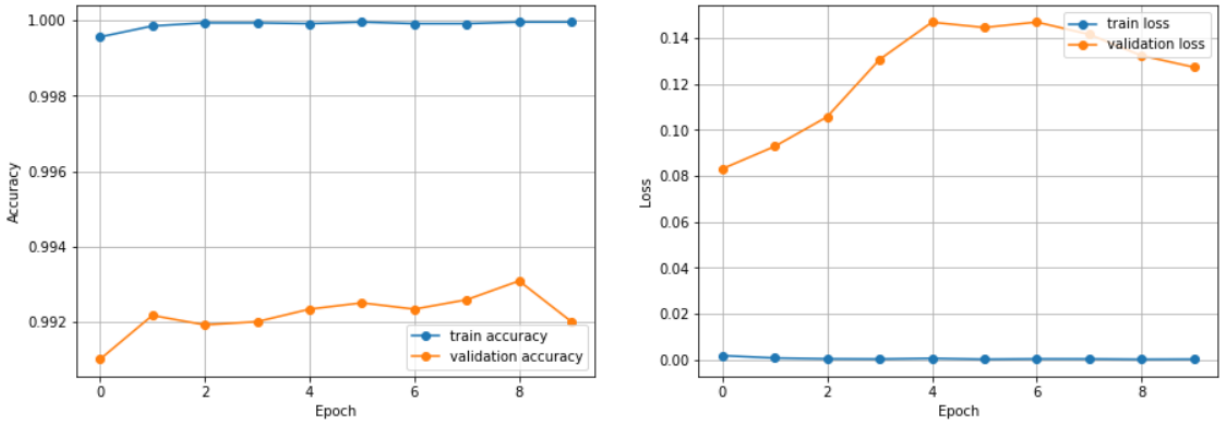


Figure 5: Graph for RMSProps Optimizer

4. Discussion

Using 3 different optimizer and 10 epoch the model is able to give more than 98% accuracy for mnist dataset. RMSProps and SGD optimizer performed very well in this model. Using RMSProps and SGD optimizer we can get 100% train accuracy and greater than 99% accuracy in testing. Here Adam optimizer has lowest accuracy rate in testing, training and validation. It has 99.76% accuracy in Training and 98.99% accuracy in testing.