ORCHID INTERNATIONAL College

A Final Year Project Work Mid Defence

On

MNIST based Handwritten Digit Recognition using Deep Learning

Presented By:

Biswas Poudyal(10526/073)
Krishna Khanal(10691/073)
Sameer Kattel(10713/073)
Subina Lama(10720/073)

May, 2020

Presentation Outline

- Introduction
- Objectives
- System Analysis
- System Design
- Visualization
- Output and Analysis
- Conclusion and Further Work

Introduction

- Handwritten Digit Recognition is an approach of enabling computers to recognize human written digits through a complicated process of learning called Machine Learning.
- The project intends to implement a Feedforward Neural Network model from scratch, that will be able to recognize the handwritten digits with high accuracy.

Objectives

- To design a Multi-Layer FFNN Model that recognizes Handwritten digits (one at a time) with high accuracy.
- To properly analyze and optimize the designed model using different model improvement techniques and optimization algorithms.
- To implement the components of the project from scratch.

System Analysis

Functional Requirements

- The model should be able to classify any new image of a handwritten digit with maximum accuracy.

System Analysis

Data Description

- MNIST Database used.
- The database was consists of four files in binary(idx) file format
- Data after processing is split into Train/ Dev and test set

Dataset	Processed Array size	Original Array Size
Training Set Image	(784, 60000)	(60000, 28, 28)
Training Set Label	(10, 60000)	(1, 60000)
Dev Set Image	(784, 5000)	(5000, 28, 28)
Dev Set Label	(10, 5000)	(1, 5000)
Test Set Image	(784, 5000)	(5000, 28, 28)
Test Set Label	(10, 5000)	(1, 5000)

System Analysis

Process Modeling

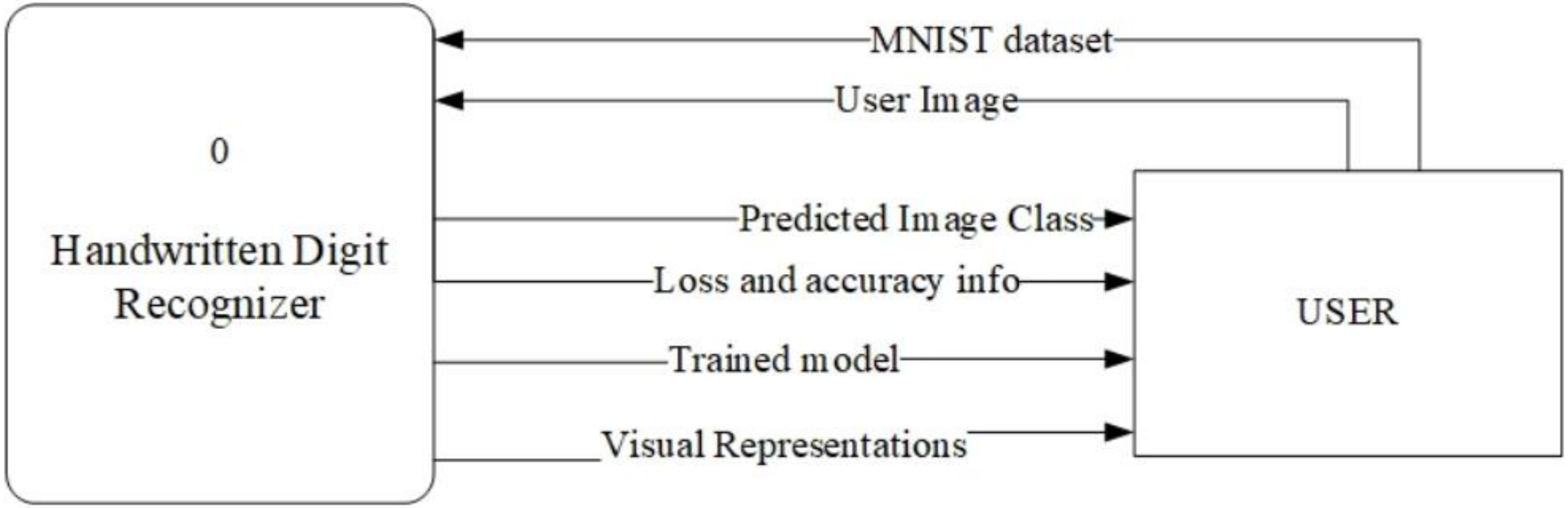
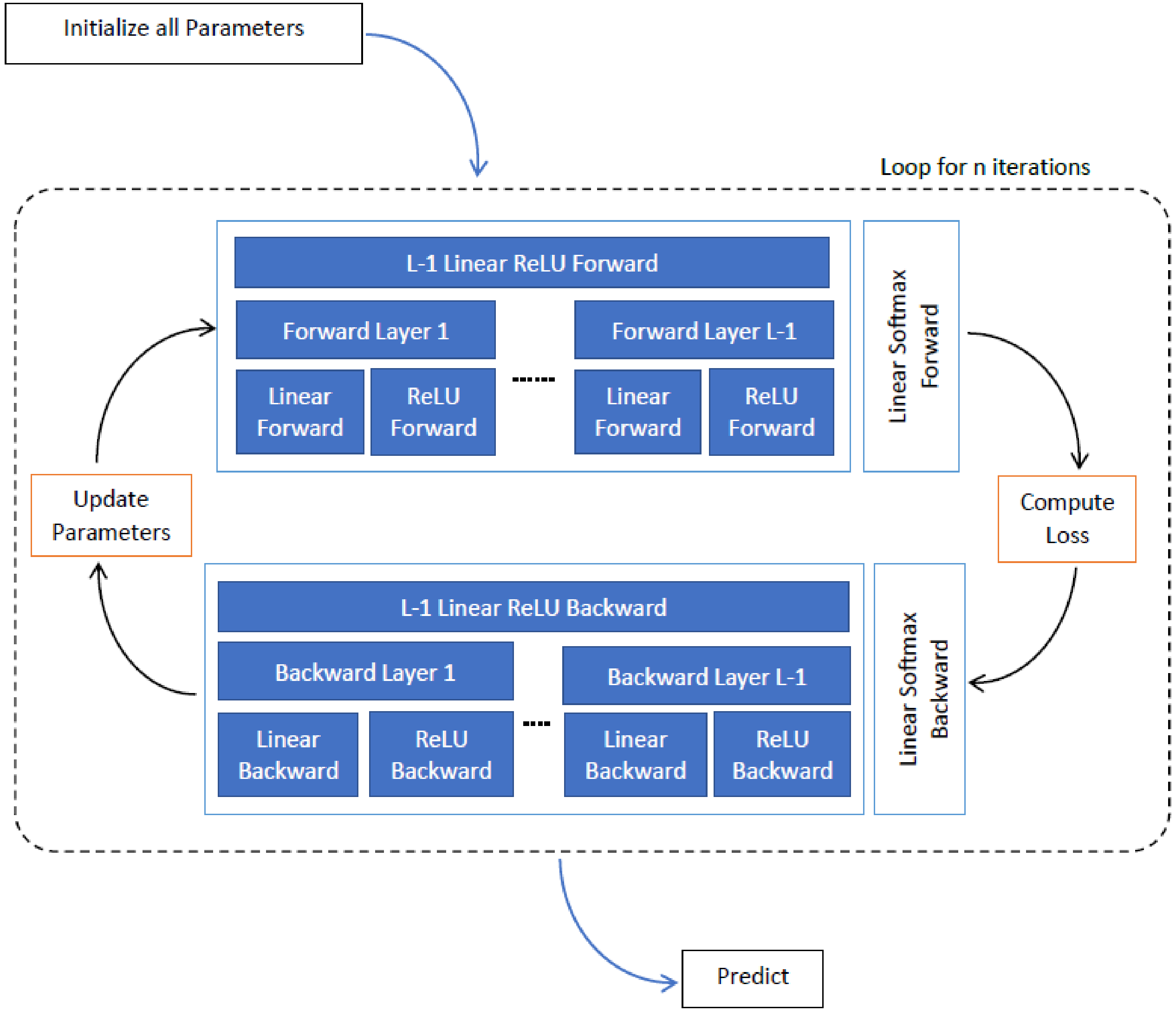


Fig: Context Diagram

System Design

Base Model Architecture

- Number of Layers: 3 ; 2 Hidden and 1 Output layers
- Layer 1: 32 Hidden Units
- Layer 2: 16 Hidden Units
- Layer 3: 10 Units
- Learning Rate: 0.1
- Number of Epoch: 10000

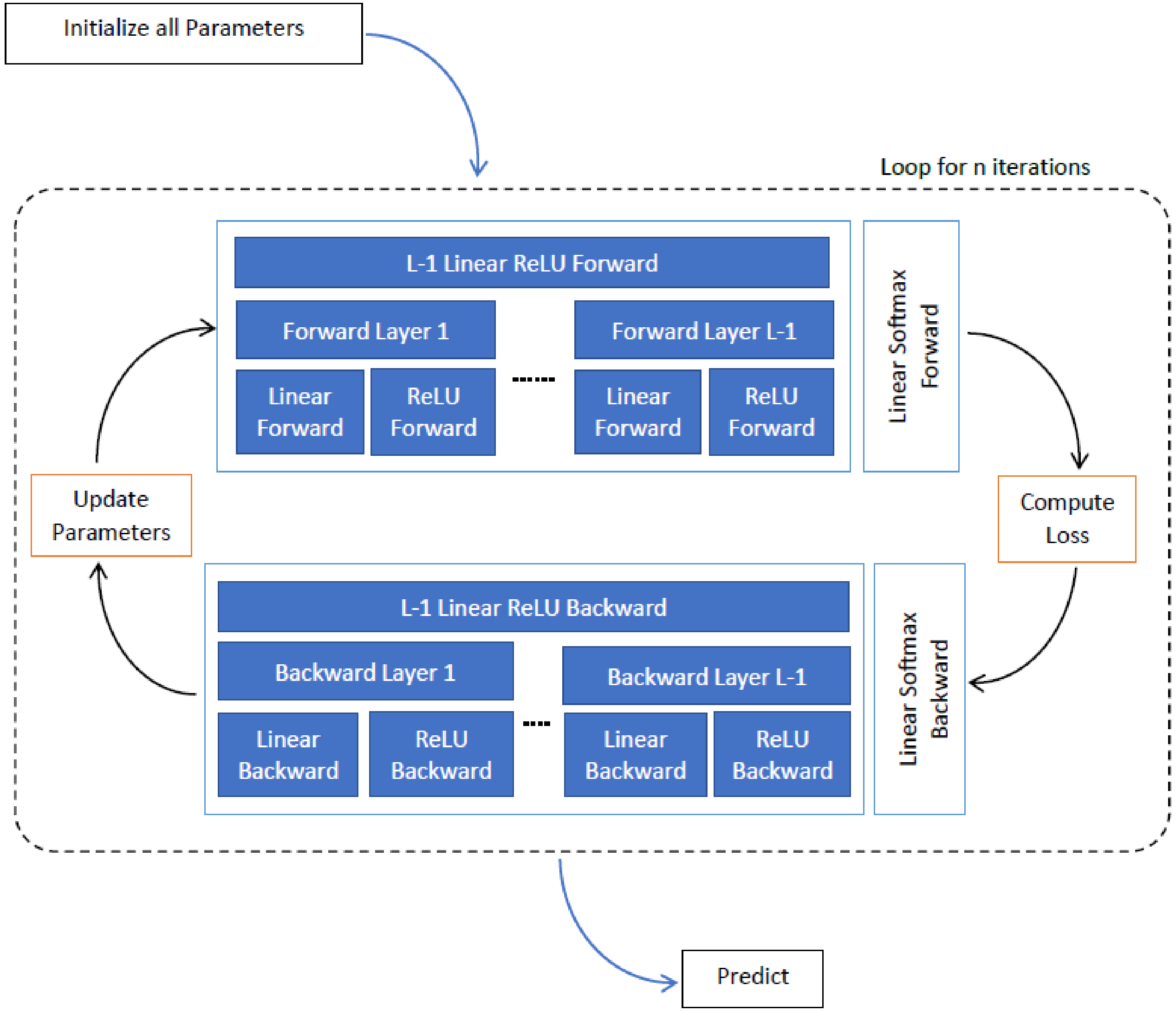


Feedforward Neural Network Model

System Design

Base Model Architecture

- Activation Functions: ReLU in Hidden Layers and Softmax in Output Layer
- Cost Function: Softmax Cross Entropy Cost Function
- Algorithm: Batch Gradient Descent with Back Propagation
- Initialization: Random



Feedforward Neural Network Model

Visualization

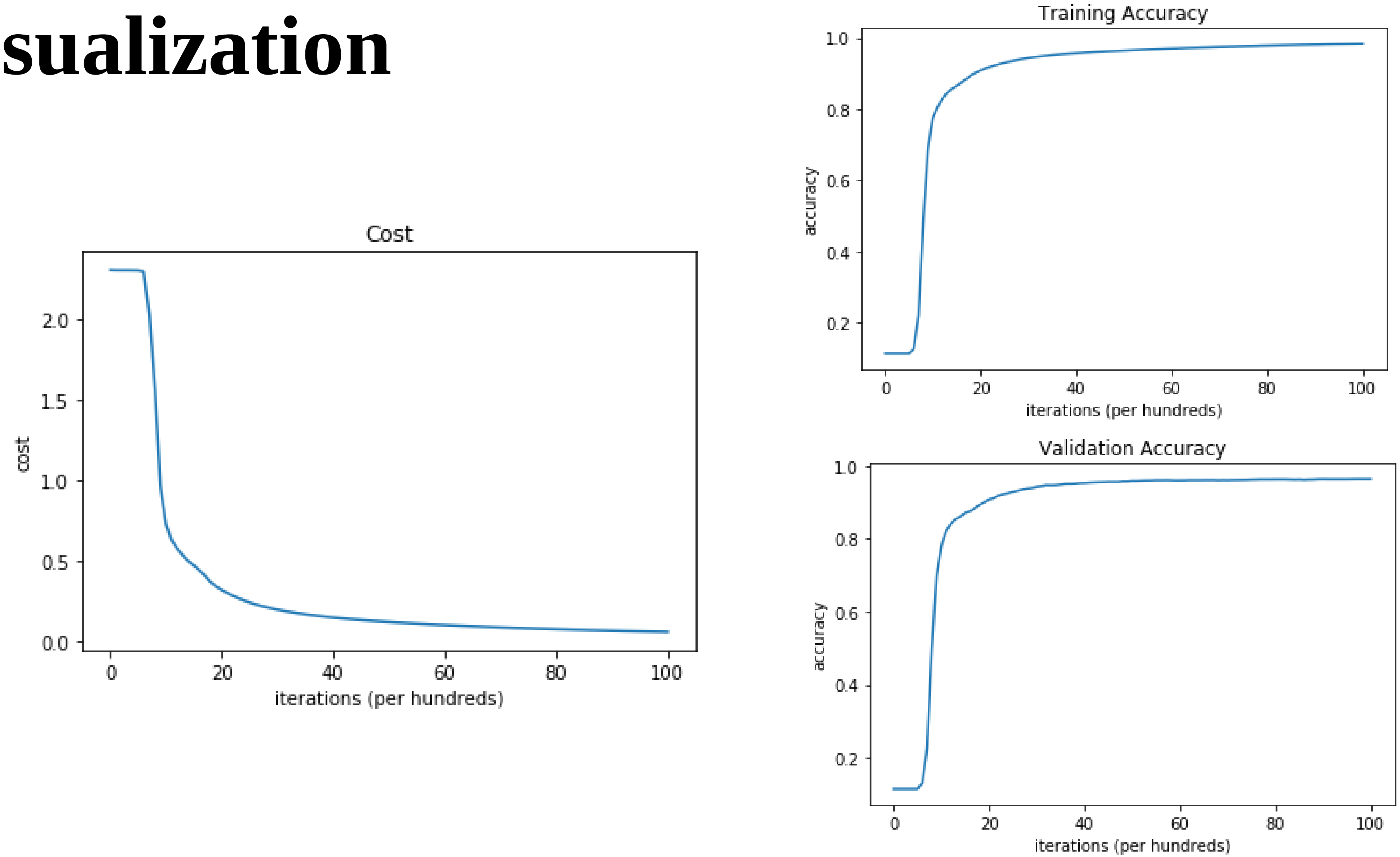


Fig: Visualization of Cost and Accuracy during Training

Visualization

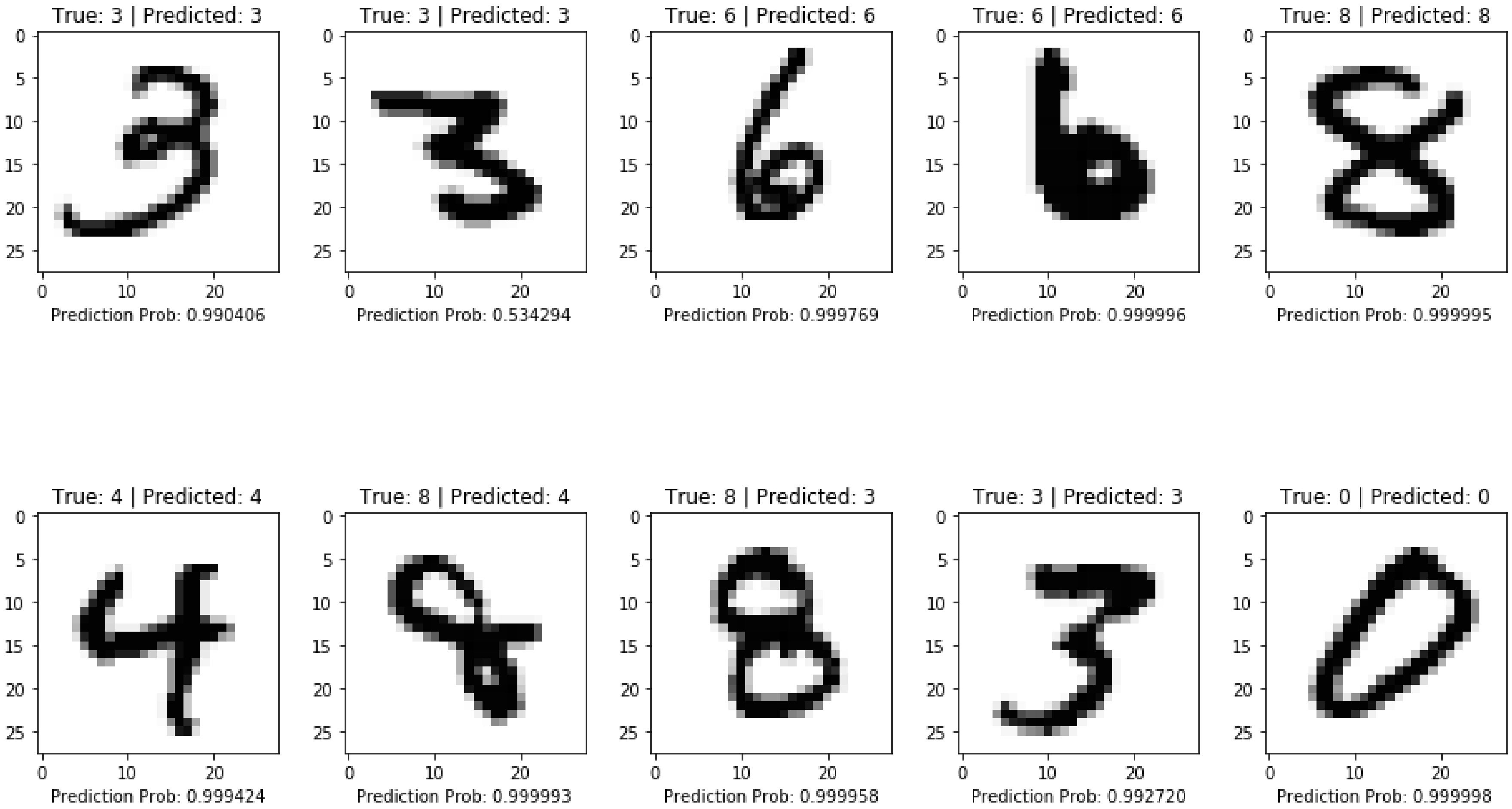


Fig: Prediction of Dev Set Data

Output and Analysis

There seems a problem of high bias and high variance

To reduce high Bias:

- Try advanced optimization algorithms
- Train Larger Network
- Tune Hyper Parameter

To reduce high variance:

- Regularization should be carried out

Dataset	Accuracy	Error
Training Set	98.39%	1.61%
Dev Set	96.32%	3.680%
Test Set	96.68%	3.32%

Conclusion and Further Work

- Algorithms like Mini-batch Gradient Descent and Adam will be used
- Regularization methods like Dropout and L2 regularization will be used
- The model will be trained in larger Network
- He- Initialization would be used to initialize the parameters
- Tuning of Hyper Parameters would be done to obtain better result