Implementation of Deep Neural Network

Group 10
Akshat Doshi, Faheem Samol, Neha Aytoda, Rajvi Prajapati, Ratnesh Shah
School Of Engineering & Applied Sciences, Ahmedabad University.

***Abstract*—It is a field of study that investigates how simple models of biological brains can be used to solve difficult computational tasks like the predictive modeling tasks we see in machine learning. The goal is not to create realistic models of the brain, but instead to develop robust algorithms and data structures that we can use to model difficult problems. The power of neural networks come from their ability to learn the representation in our training data and how to best relate it to the output variable that we want to predict.**

***Keywords*– Convolution Neural Network, Deep Neural Network, Neurons, Activation Function**

I. Deep Neural Network

Deep neural networks is also called as "stacked neural networks"- A set of networks composed of several hidden layers between the input and output layers. DNN can be trained on labeled data and then applied to unstructured data. Its training methods are derived from probabilistic interpretation of networks as generative models. There are 3 steps for the learning process of a Neural Network:
1. input * weight + bias = guess
2. ground truth - guess = error
3. error * weight's contribution to error = adjustment
This process of updation of weights is known as Back Propagation. The activation functions such as sigmoid, ReLU are used to determine what output a node will generate based upon it's input.

II. Results

A. Handwritten Digit Recognition

We have used MNIST dataset in which each image is a 28 x 28 pixel square (784 pixels total). 60,000 images are used for training whereas a new set of 10,000 images are used for testing. Since there are 10 digits we have 10 classes to predict.

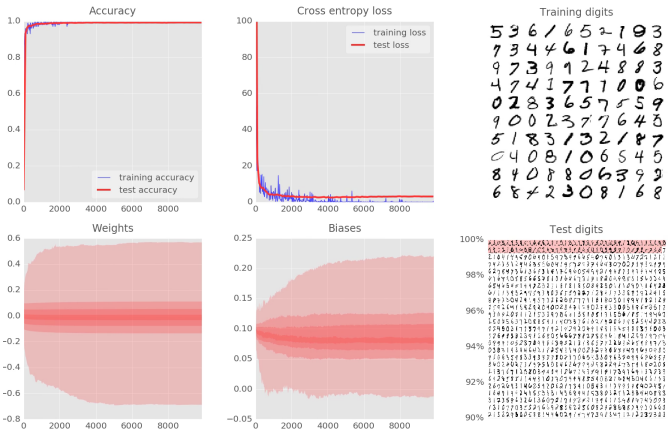

Fig 1: Handwritten Digit Recognition

Using a simple neural network with 10 neurons and a softmax function for classification we achieved an accuracy of 92% which is quite less in terms of real life scenarios.
To increase the accuracy we moved towards deep learning, now we have used three convolution layers and two fully connected layers through which we obtained an accuracy of 98.2% using ReLU as an activation function for the hidden layers and softmax function for output layer.
After doing this we found that the error function was not converging properly after certain no. of iterations this happens due to over fitting, for this dropout was the solution we found. Dropout as the name suggests drops neurons randomly with certain probability during each iteration of training. This finally increased our accuracy to 99.2%.

B. Diabetes Dataset

We have used the Pima Indians diabetes data set. It contains 5 years diabetic medical history of patient. It is a binary classification problem onset of diabetes as 1 or not as 0. There are 8 attributes and a total of 768 entries in the data. The below flowchart shows our implemented model of neural network.

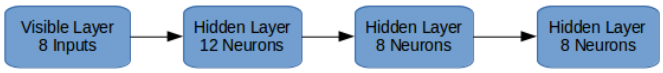

Fig 2: Deep Neural Network Model

If we take the first case where we used 100 entries of data in input out of the 67% of entries were used for training and rest 33% entries were used for testing.

Entries/768	Training Accuracy(%)	Testing Accuracy(%)
100	76.12	54.3
300	71.74	61.7
500	74.33	71.52
768	77.82	75.20

Epochs	Accuracy(%)	Time to Compute (Sec)
10	66.14	3.881
50	70.87	6.08
150	72.83	14
200	75.59	18.89

References

[1] codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#0.
[2] Deep Learning with python; Develop Deep Learning Models on Theano and Tensorflow Using Keras.