Deep Neural Network

Philip Hoddinott

December 16, 2018

Abstract

Remove pasive voice, will, chec out rubirc The purpose of this report is to develop a neural net that can identify handwritten digets in the MNIST database at near human levels of accuracy. The neural net will be developed without the assistance of libraries such as Python's tensor flow or MATLAB's Deep Learning.

solve the MNIST on the mnist database .

The author would like to express his gratitude to Professor Hicken

for his suggestion of this project and his assistance with the methods of orbital determination through out the semester.

Contents

List of Figures

1 Introduction

Have it solve the MNIST with a simple simple thing then try diffrent layers and stuff

Go over tan h vs sigmoid Explain batch testing

1.1 The MNIST database

The Modified National Institute of Standards and Technology database or MNIST database [4] is a database of handwritten numbers used to train image processing systems. It contains 60,000 training images and 10,000 testing images.

A number of attempts have been made to get the lowest possible error rate on this dataset. As of August 2018 the the lowest achieved so far is a error rate of 0.21% or an accuracy of 99.79%. For comparison human brains that are hardwired for pattern recognition 1.5 Come back to [5].

The database is comprised of images that are made up of a grid of 28x28 pixels, as seen in figure 1.

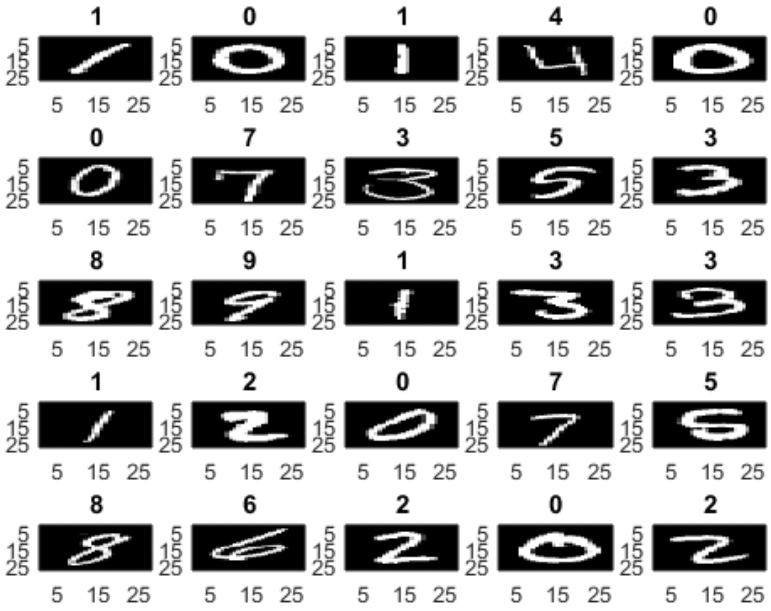

Figure 1: Sample numbers from MNIST [1].

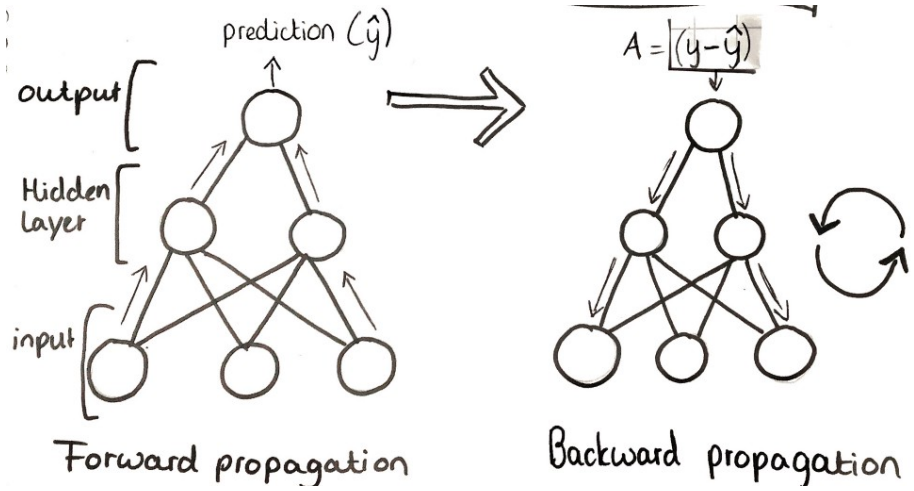

Figure 2: A visulization of forward and backward propogation [2].

1.2 Artificial neural network

An artificial neural network (referred tp as a neural network in this paper) is a computation system that mimics the biological neural netwroks found in animal brains. A nerual network is not an explict A Neural networks may be trained for tasks, such as the number recognition in this report.

1.2.1 Nerual Network Walk through

Forward and backward propogation are visulized in figure 2

1.2.2 Forward Propagation

In a neural network forward propagation is the

1.2.3 Backward propgration

After going forward the nerual net in the forward propgration, the next step is backwards propagation. Backwards propagation is the updating of the weight parameters via the gradient of the error. The graident of the error

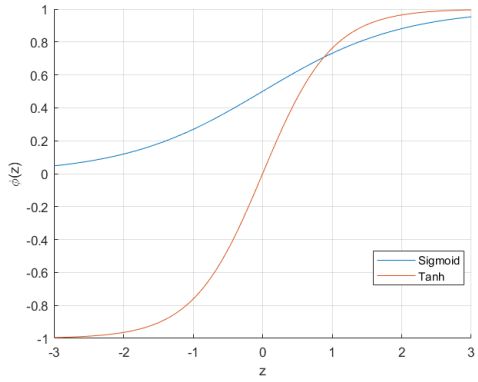

Figure 3: Visualization of sigmoid and Tanh function

1.3 Gradient Decent

1.3.1 Activation Function

The two most common activation functions used in neural nets for the gradient decent are sigmoid and hyperbolic tangent (Tanh). The formula for Tanh is seen in equation 1, and the derivative of Tanh is seen in equation 2

$$\phi(z) = \frac{1 - e^{-2z}}{1 + e^{-2z}} \tag{1}$$

$$\phi'(z) = \frac{4}{(x^{-z} + e^{z})^2} \tag{2}$$

The equation for the sigmoid function is seen in equation 3, and it's derivative is seen in equation 4.

$$\phi(z) = \frac{1}{1 + e^{-z}} \tag{3}$$

$$\phi'(z) = \frac{e^{-z}}{(e^{-z} + 1)^2} \tag{4}$$

The sigmoid and Tanh function are visualized in figure 3.

Both functions have relativly simple mathematical formulas and are differentiable. In this paper the sigmoid function is used over the Tanh function,

The sigmoid function function is used over the Tanh function as it does not pass through the zero. and

Expalain importance of activation function

ID	architecture (number of neurons in each layer)	test error for best validation [%]	best test error [%]	simulation time [h]	weights [milions]
1	1000, 500, 10	**0.49**	0.44	23.4	1.34
2	1500, 1000, 500, 10	**0.46**	0.40	44.2	3.26
3	2000, 1500, 1000, 500, 10	**0.41**	0.39	66.7	6.69
4	2500, 2000, 1500, 1000, 500, 10	**0.35**	0.32	114.5	12.11

Figure 4: Run times for various neural network architectures [3].

1.4 Pitfalls

The most important thing to stear clear of is over traning. Overtraning occurs when the neural net trains too much to the traning data. While it will have a high accuracy for the traning data, it's performance for the test data will decay, as it has become too well attuned to the training data.

The other problem is the time it takes to train. A three layer neural net can be trained to 97% accuracy within 10 minutes, however it will not improve far beyond that. Larger nets will take longer to train, but will take far longer to train.

2 Implementation

The code written for this project was diuesgend so that the For a three layer neural net a hidden layer of 250 neurons seems to work the best [1].

Go over how it was implemented Go over batch testing

restuls, comparison of diffrent arctiectures

Go over the way this was implmented for the best way

3 Results

3.1 Simple Neural Net

For a simple, $784 \times 250 \times 10$ neural net with a learning rate of 0.1 a test accuracy of 98% was achieved. Looking at the results, It was discovered that

4 Conclusion

References

[1] Loren Shure. Artificial neural networks for beginners. https://blogs.mathworks.com/loren/2015/08/04/artificial-neural-networks-for-beginners/, August 2015.

[2] Daphne Cornelisse. https://medium.freecodecamp.org/building-a-3-layer-neural-network-from-scratch-99239c4af5d3?fbclid=IwAR1jjh1IsEVdvN0pIeMygzfY2ZG3K-Zata3z_jqlfLtQZKK0X6QEbNZABzw, February 2018.

[3] Luca Maria Gambardella Jurgen Schmidhuber Dan Claudiu Ciresan, Ueli Meier. Deep big simple neural nets excel on handwritten digit recognition. https://arxiv.org/pdf/1003.0358.pdf, March 2010.

[4] Christopher J.C. Burges Yann LeCun, Corinna Cortes. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

[5] John Denker Patrice Simard, Yann Le Cun. Efficient pattern recognition using a new transformation distance. https://papers.nips.cc/paper/656-efficient-pattern-recognition-using-a-new-transformation-distance.pdf.

Appendix 1 - MATLAB code