Deep Neural Network Implementation using TensorFlow

Hardil Mehta (1401018), Chintan Saraviya (1401026),
Kashish Shah (1401048), Rupande Shastri (1401102), Kishan Raval (1401117)

***Abstract*—Neural Networks are paradigms modeled on the human brain used for various applications such as pattern recognition and data classification. Deep Neural networks have more than one layer of neurons for better accuracy. Additional supporting mechanisms include RELU (to include non-linearity), Learning Rate Decay for better convergence and dropout to avoid overfitting. As part of implementation we have worked on the MNIST data set using a five layer DNN with all these mechanisms. We have additionally implemented a three layer DNN on various data sets for classification.**

I. Introduction

A neural network consists of a set of neurons which are trained to learn in a dynamic environment. These neurons learn by continuously adapting to the changes in their environment, that is, their inputs and feedback. A neural network that has more than one layers is called a Deep Neural Network (DNN). A neural network generally has two processes which completes the learning process: feed-forward and backpropogation. Feed forward is when the input goes through the layer(s) of the neural network to get processed. Backpropogation is when the outputs are fed back to the system to for improvisation.

A. *Algorithm Used*

Softmax Regression is used to predict and classify the images into their respective categories. In softmax they first add up all the evidence for and against the data object belonging to a certain class, and then converts the evidence into probabilities that add up to one when added for each class. The activation function is a weighted sum of inputs along with bias, given as $Y = WX + b$, where W is the set of weights and b is the set of biases for input X and output Y.

For this first layer of our neural network, we have 10 neurons (one for each category). Each is fed with all images as input and gives out the probability of each image being of that category. The output of this layer of neurons is a 10xn dimensional vector with probabilities of each of the n images belonging to each of the categories. These vectors are one-hot coded to classify each image into exactly one category of the highest probability.

B. *Deep Neural Network - Adding hidden layers*

In order to increase accuracy we add more hidden layers in our neural network. The layer that was built for classification is last, preceded by hidden layers that have their own activation function. The last layer now takes as input the output of its preceding hidden layer.

TABLE I. Simulation Results: 3 layer DNN

Data Set Taken	Breast Cancer		Wine Reco. Data	iris Data
No. of Features	30	30	13	4
Training Set	400 Samples	400 Samples	100 Samples	120
Test Set	170 Samples	170 Samples	78 Samples	30
No. of Iterations	500	2000	500	200
Accuracy	93%	95%	82%	96.66%

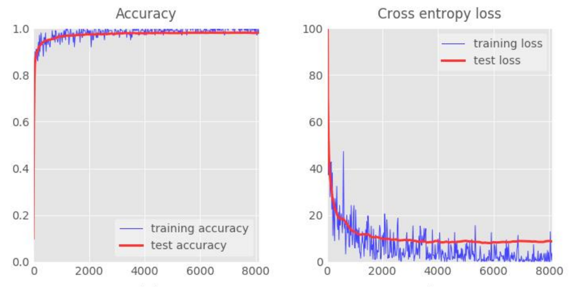
The classic sigmoid activation function tends to cause problems in deep neural networks. Modern neural networks use RELU (Rectified Linear Unit). RELU helps in including non-linearity without compromising on simplicity of derivatives and other processes on the function.

Keeping a constant learning rate in the optimizing function makes the accuracy fluctuate after some time. To solve this, we reduce the learning rate up to a certain small value to make sure the process converges and to stabilize the change in accuracy. This process is called learning rate decay.

On observation we find that at a certain point, the training accuracy increases faster than the test accuracy while learning. This is when the test accuracy tends to become constant, and the learning is no longer productive - commonly known as a case of overfitting. In such cases, we want to reduce the training accuracy to match the test accuracy to allow generality into our learning. This can be done using a phenomenon called 'dropout' where we drop some fraction of the neurons at each iteration.

II. Implementation

Part 1: Implementing a five layer DNN with RELU, L-R decay and dropout on the MNIST data set



Part 2: Implementing a three layer DNN with RELU on different data sets varying the step size and comparing accuracy. The results are shown in Table I.