Progress Report 2: Generative Models Using Probabilistic Principal Component Analysis

Group 9

Aashima Yuthika(1401071), Deep Patel(1401010), Deval Shah(1401060), Kirtan Modi(1401122), Yash Kotadia(1401114)

School of Engineering and Applied Sciences, Ahmedabad University

Subject: Machine Learning & Algorithms and Optimisation for Big Data

March 26, 2017

***Abstract*—The main aim of this project is to be able to build Generative models using the concept of Probabilistic Principal Component Analysis or PPCA and Deep Neural Networks. We will be using Deep Neural Networks for training the model and then having it generate images similar to the real ones in the Principal Component domain, hence implementing Generative Models.**

I. Introduction

Generative models are being largely studied now by the Machine Learning Community and there are many researchers working on this problem as well. For this project we will be making use of the concept of PPCA to try and achieve this. Along with PPCA we will be using Deep Neural Networks. The basic concept of Generative models using two parallel systems - The Discriminator Network and The Generator Network. The Discriminator Network tries to estimate whether the data is real or not, and The Generator Network tries to introduce fake data that fools The Discriminator. For this part of the report we have tried to understand the Deep Neural Network using TensorFlow. We had explored tools like TensorFlow, Theanos and Keras. TKeras uses TensorFlow in the Background so we have gone with TensorFlow and in general it's easier. Thus, we have tested our codes on the MNIST Dataset.

II. Neural Networks

Neural Networks in Machine Learning is inspired from the human brain. According to literature, A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

1) Knowledge is acquired by the network from its environment through a learning process.
2) Inter Neuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

A Neural Network (like in the case of a human brain) contains many nodes. A node combines input from the data with a set of coefficients, or weights,thus assigning significance to inputs. These input-weight products are summed and the sum is passed through a nodes activation function, to determine whether and to what extent that signal progresses further through the network to affect the ultimate outcome. A node layer is a row of those neuronlike switches that turn on or off as the input is fed through the net. Each layers output is simultaneously the subsequent layers input, starting from an initial input layer receiving your data.

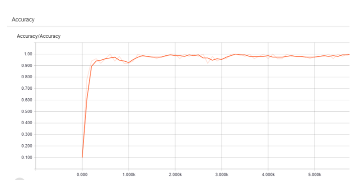
III. Deep and Convolutional Neural Networks

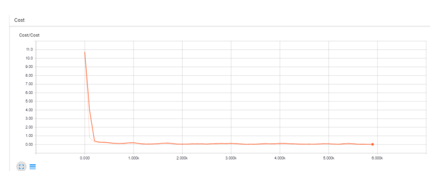
Deep-learning networks are distinguished from the more commonplace single-hidden-layer neural networks by their depth; that is, the number of node layers through which data passes in a multistep process of pattern recognition. More than three layers (including input and output) qualifies as Deep Learning. In deep-learning networks, each layer of node trains on a distinct set of features based on the previous layers output. The further one advances into the neural net, the more complex the features the nodes are capable of recognizing, since they aggregate and recombine features from the previous layer.

A Convolutional NN architecture is formed by a stack of distinct layers that transform the input volume into an output volume through a differentiable function. The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2-dimensional activation map of that filter. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input.

IV. Results

We have used TensorFlow to implement Deep Convolution Neural Network on the MNIST Dataset, which contains binary images of handwritten digits. The network has 1 input layer, 2 convolution layers, 1 fully connected layer, and 1 output layer. The accuracy for Deep CNN is 98.62 % which is better than Simple Neural Network analysis which gave around 92 %. Images: Accuracy and Cost graphs.





References

[1] Simon Haykin, *Neural Networks - A Comprehensive Approach*, 2 Ed., Pearson Prentice Hall, 23-59.

[2] https://deeplearning4j.org/neuralnet-overview

[3] http://yann.lecun.com/exdb/mnist/