Design Autoencoder using BSnet (BSautonet)

Sing Kuang Tan
Email: singkuangtan@gmail.com

December 29, 2022

Abstract

In this paper, I am going to propose a design for an Autoencoder using BSnet. To take advantage of the BSnet design, the autoencoder will be easy to train with more convex training optimization function. The idea is to develop a simple and standard unsupervised machine learning model that can easily be used on most of the data without label. In the experiment result, the output is subjectively evaluated by a human and it has shown to achieve human level accuracy on denoising the MNIST human handwriting digits dataset.

1 Introduction

Given a set of images without label, the most convenient way is to learn a generative model to model the data. In this paper, we developed an autoencoder that will compress each image into a compress representation, which has many uses, e.g. as an embedding to find nearest match image given an input image. The autoencoder can be used to compress a image dataset, denoise the input images and understand the images intuitively without human to do labeling.

2 My Model

I developed a Boolean Structured network (BSnet aka BullShit net, LOL Laugh Out Loud). The link of my paper is https://vixra.org/abs/2212.0193.

The Autoencoder using BSnet has better performance than ordinary network. It has 10 layers design, with a compress center layer to compute the compressed codes of handwriting digit images. All previous layers will connect to the center layer, act like skip connections in Residual network. All layers after the center layer will connect to the last layer. No skip connection that connects from the first layer to last layer, so that the network can compute compressed codes in the center layer. Each input image pixel is converted to 8 bits binary number. So the input to the autoencoder is 784*8 dimensions where 784 is the number of pixels in each image. Note that each handwriting digit image is 28 by 28 in size, which is equal to 784 dimensions. The last layer will have sigmoid activation function, to output the 8 bit binary numbers of each pixel of output image. The 8 bit numbers are converted to real numbers and an absolute difference is computed between the real numbers and groundtruth pixel values. The absolute difference function (Mean Absolute Error (MAE) loss) is the loss function for training the deep learning autoencoder.

The figure 1 shown below is the network diagram of the autoencoder (BSautonet).

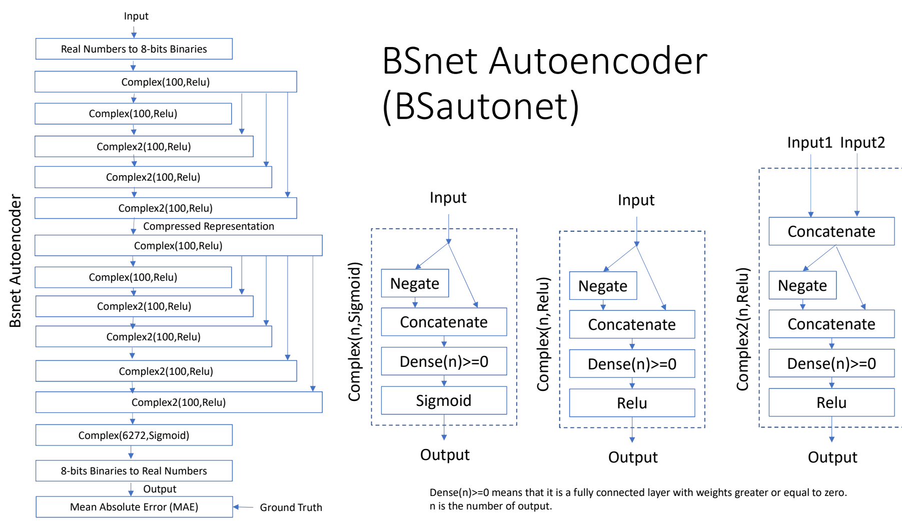

Figure 1: Network Diagram of BSnet Autoencoder (BSautonet))

3 Experiment Results

The autoencoder is trained on the MNIST dataset to evaluate its performance.

To make the problem difficult, 70% of the input image pixels (training set and testing set) are masked (set to maximum value of 1) and this makes the trained autoencoder more robust to noise.

Human evaluation of the output shows that the autoencoder is able to denoise the input handwriting digit image as good as human. U-Nets are often used in the literature and it uses 20 millions parameters. In contrast, my autoencoder uses only 2.8 millions parameters. BSnet can be used in the future for Generative AI (Artificial intelligence) such as diffusion model in image generation. Studying of the compressed representation codes shows that my autoencoder are able to generalize and compress handwriting digit images.

The figure 2 shown below shows samples of noisy inputs, denoised outputs and groudtruth from the autoencoder.

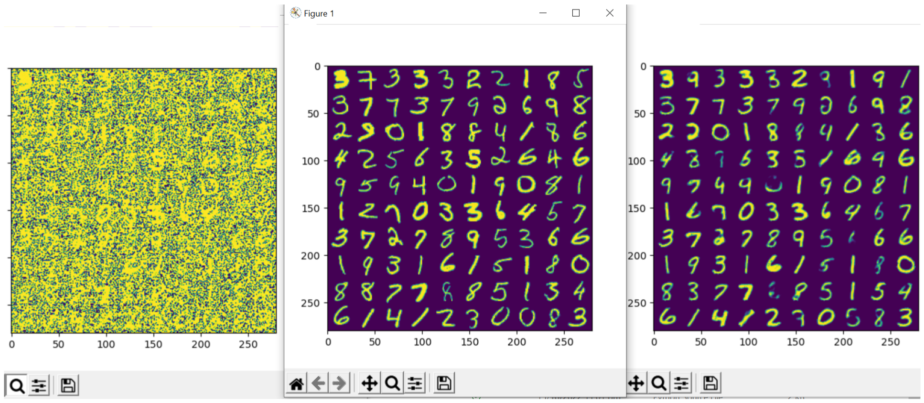

Figure 2: Noisy inputs (left), denoised outputs (right) and groundtruth (center)

From Figure 2, we can see that the autoencoder can denoise the noisy input images as good as a human.

Access my GitHub codes thru this link: https://github.com/singkuangtan/BSautonet

4 Conclusion

I have developed an autoencoder using BSnet. It can be used for most unsupervised learning e.g. learning embedding representation of a set of images. In the future, the BSnet design can be incorporated into more advanced generative network such as diffusion deep learning model and many self-supervised tasks or other difficult tasks with partial labels.