Denoising Auto Encoder with Intermittency Chaos

Ryuta YOSHIMURA, Shinsaburo KITTAKA, Yoko UWATE, and Yoshifumi NISHIO
Dept. Electrical and Electronic Engineering,
Tokushima University
2-1 Minami-Josanjima, Tokushima 7708506, Japan
Email: yoshimura,kittaka,uwate,nishio@ee.tokushima-u.ac.jp

I. Introduction

In the study,we investigate denoising auto encoder with chaotic noises in order to obtain good parameters in deep learning. The noise makes up from logistic map. We confirm effectiveness by simulation.

II. Proposed Method

We use chaotic noise with denoising auto encoder to obtain good features. It uses input data with the noise and defines original data as answer. Denoising auto encoder reconstructs the original data by adjusting weights and obtains good parameters for deep learning.

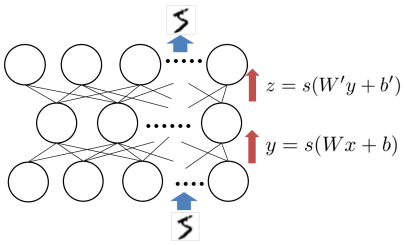


Fig. 1. Denoising auto encoder.

$$\begin{cases} y = s(Wx + b) \\ z = s(W'y + b') \end{cases} \tag{1}$$

Equation (1) shows encode and decode. x means input data. y means encoded information. z means reconstructing data from x. s means sigmoid function. W and W' mean weight. We define W_x equal W'_x by tied weight. We update weight so that output z nears input x with calculating an error function. After updating the parameters W, b and b' the network calculates to minimize error function. The cross entropy which is shown by Eq. (2) is used as error function.

$$L_H(x, z) = -x \log z - (1 - x) \log(1 - z) \tag{2}$$

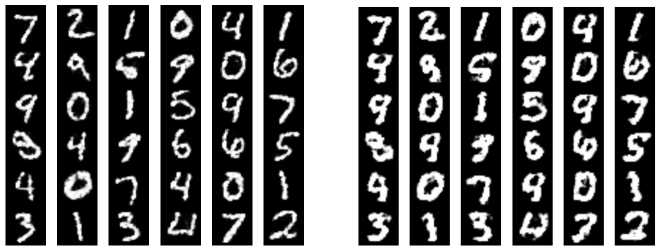
In order to minimize the error function, network should calculate a slope of L_H for W, b and b'. After calculating, network obtains new parameters.

$$f(x_{n+1}) \quad = \quad ax_n(1 - x_n) \tag{3}$$

Equation (3) shows logistic map. Parameter a changes the logistic map behavior. We set the parameter a as 3.828327 and use the intermittency chaos. We generate a random number by the logistic and compare the number with the threshold. When it exceeds the threshold, it output 0. The output is multiplied by each pixel of the input data. The pixel which is multiplied 0 is painted black and becomes a noise.

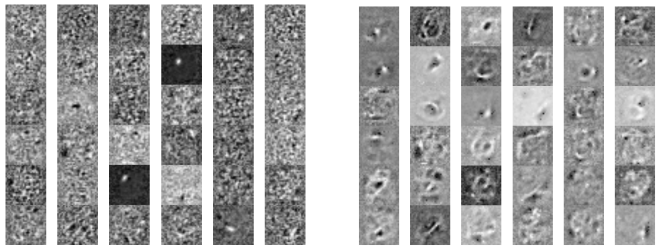
III. Simulation Results

We put a different percentage of noise into the network to show difference of percentage of noise. We find out that the error function is decreased and denoising auto encoder can learn itself. We show a visualization of what kind of information the weight holds and reconstructed images.



(a) (b)

Fig. 2. Reconstructed image due to percentage of noise. (a) 20%. (b) 80%.



(a) (b)

Fig. 3. Weight visualization due to percentage of noise. (a) 20%. (b) 80%.

Figure 2 shows reconstructed images due to difference of percentage of noise. The letters of (b) are visually thicker than one of (a). Figure 3 shows that tendency of weight that is different due to percentage of noise. When the rate is 80%, it seems the weight has features like letter of numbers.

IV. Conclusions

As the rate of noise was increased, the letter of reconstructed image was thick and clear. This was because a tendency to correct black noise to white becomes strong. Some of the visualized weights showed features like number.