Project Presentation on Denoising Auto-Encoders for Image Denoising

Karan Goel

2011EE50555

26th April 2015

Denoising Auto-Encoders

f_θ maps input to the hidden representation $\mathbf{y}$, using a mapping

$$\mathbf{y} = f_\theta(\tilde{\mathbf{x}}) = s(\mathbf{W}\tilde{\mathbf{x}} + \mathbf{b})$$

$\mathbf{y}$ is then mapped to denoised output $\mathbf{z}$, by $g_{\theta'}$ where

$$\mathbf{z} = g_{\theta'}(\mathbf{y}) = s(\mathbf{W}'\mathbf{y} + \mathbf{b}')$$

Learning carried out using stochastic gradient descent with back-propagation. Cross-entropy loss is being used

$$L_H(\mathbf{x}, \mathbf{z}) = -\sum_{k=1}^{d}[\mathbf{x_k} \log \mathbf{z_k} + (\mathbf{1} - \mathbf{x_k}) \log(\mathbf{1} - \mathbf{z_k})]$$

Denoising Auto-Encoders

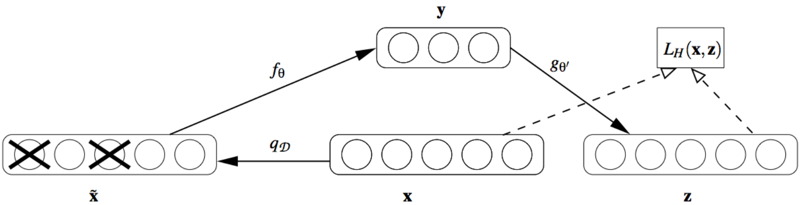

Figure 1: Architecture of the Basic Denoising Auto-Encoder

Outline

Aim is to investigate

- Effect of Hidden Units
- Effect of Noise Model
- Effect of Noise Parameters
- Usefulness with Different Kinds of Images [MNIST, CIFAR]
- Effect of Loss Formulation

Effect of Hidden Units

Dataset: MNIST
Noise Model: Masking Noise
Fraction of Masked Pixels in Training: 30%
Fraction of Masked Pixels in Testing: 30%
PSNR (Image 1, Noisy) = 61.68
PSNR (Image 2, Noisy) = 61.44

Hidden Units	**Image 1**	**Image 2**
1	60.91	60.72
10	62.41	62.26
100	69.11	68.07
500	70.59	70.23
1000	70.87	**70.90**
5000	**71.38**	70.78

Table 1: Comparison of Reconstructed PSNR by changing Hidden Units

10 Hidden Units

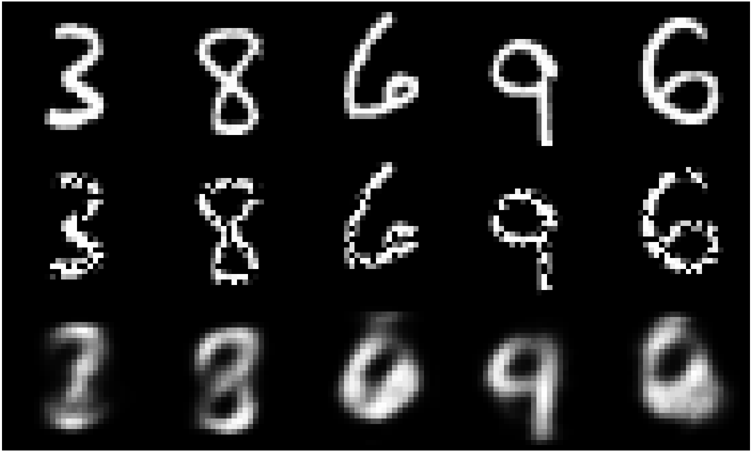

Figure 2: Top: Original, Middle: Noisy, Bottom: Reconstructed

500 Hidden Units

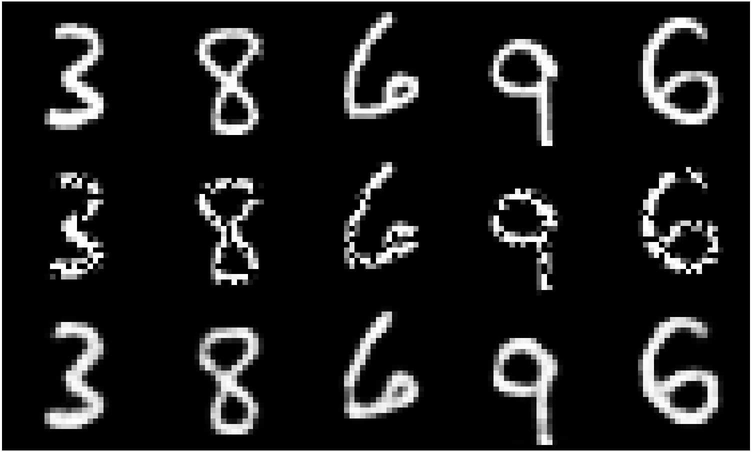

Figure 3: Top: Original, Middle: Noisy, Bottom: Reconstructed

Effect of Noise Model

Dataset: MNIST
Training Noise Model: Masking Noise
Fraction of Masked Pixels in Training: 30%
Hidden Units: 500
Testing Noise Model: Gaussian(0, 25)

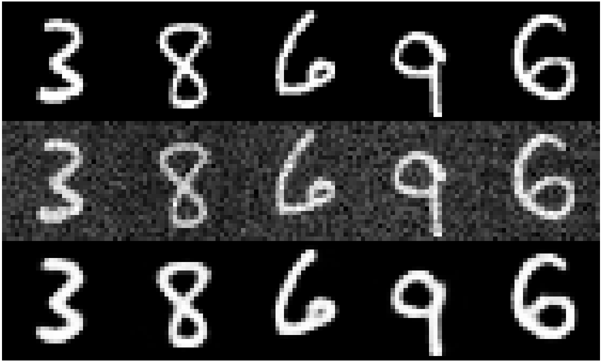

Figure 4: Top: Original, Middle: Noisy, Bottom: Reconstructed

Effect of Noise Model

Dataset: MNIST
Training Noise Model: Masking Noise
Fraction of Masked Pixels in Training: 30%
Hidden Units: 500
Testing Noise Model: Salt and Pepper Noise
Fraction of Masked Pixels in Testing: 30%

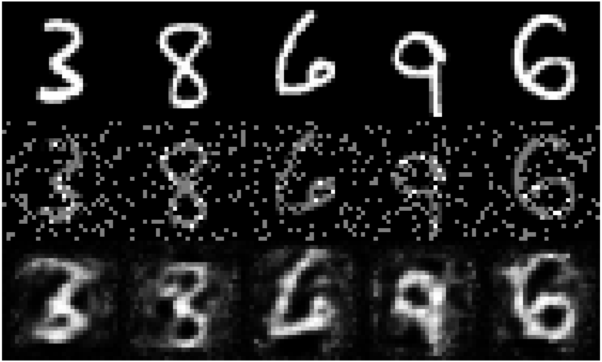

Figure 5: Top: Original, Middle: Noisy, Bottom: Reconstructed

Effect of Noise Model

Dataset: MNIST
Training Noise Model: Gaussian Noise(0,25)
Hidden Units: 500
Testing Noise Model: Masking Noise (30%)

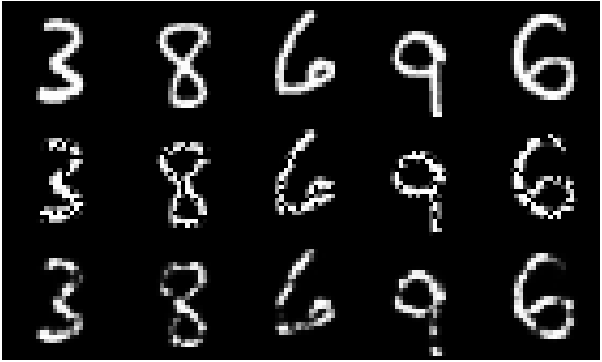

Figure 6: Top: Original, Middle: Noisy, Bottom: Reconstructed

Effect of Noise Model

Dataset: MNIST
Training Noise Model: Salt-and-Pepper Noise (30%)
Hidden Units: 500
Testing Noise Model: Gaussian Noise(0,25) + Masking Noise (30%)

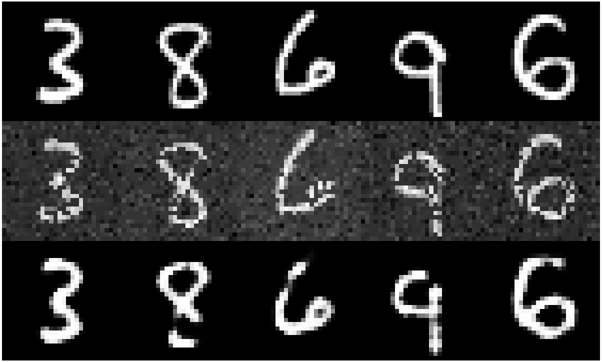

Figure 7: Top: Original, Middle: Noisy, Bottom: Reconstructed

Effect of Noise Parameters

Dataset: MNIST
Training Noise Model: Masking Noise
Fraction of Masked Pixels in Training: 0%
Hidden Units: 500
Testing Noise Model: Masking Noise (30%)

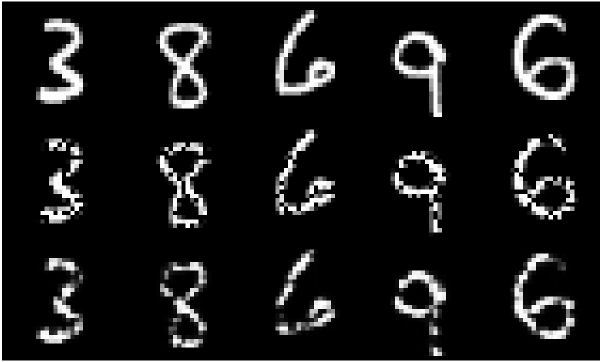

Figure 8: Top: Original, Middle: Noisy, Bottom: Reconstructed

Effect of Noise Parameters

Dataset: MNIST
Training Noise Model: Masking Noise
Fraction of Masked Pixels in Training: 30%
Hidden Units: 500
Fraction of Masked Pixels in Testing: 50%

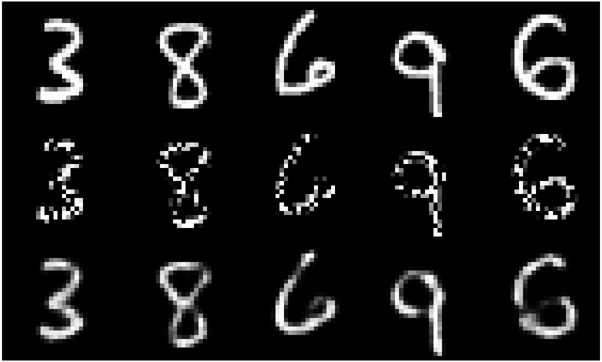

Figure 9: Top: Original, Middle: Noisy, Bottom: Reconstructed

Effect of Noise Parameters

Dataset: MNIST
Training Noise Model: Masking Noise
Fraction of Masked Pixels in Training: 50%
Hidden Units: 500
Fraction of Masked Pixels in Testing: 0%

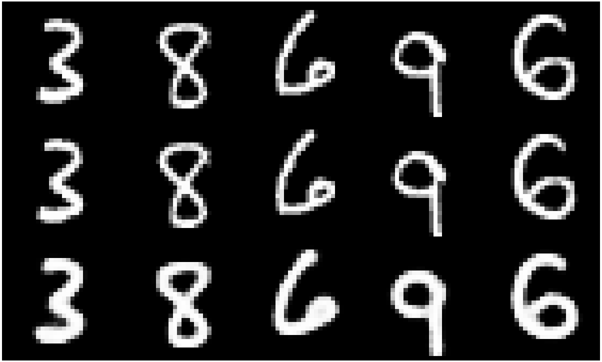

Figure 10: Top: Original, Middle: Noisy, Bottom: Reconstructed

Tiny Natural Images Dataset

Dataset: CIFAR

Training Noise Model: Gaussian(0, 25)

Hidden Units: 4000

Testing Noise Model: Gaussian(0,25)

Figure 11: Top: Original, Middle: Noisy, Bottom: Reconstructed

Tiny Natural Images Dataset

Dataset: CIFAR

Training Noise Model: Gaussian(0, 25)

Hidden Units: 8000

Testing Noise Model: Gaussian(0,25)

Figure 12: Top: Original, Middle: Noisy, Bottom: Reconstructed

Tiny Natural Images Dataset (Squared Error)

Dataset: CIFAR

Training Noise Model: Gaussian(0, 25)

Hidden Units: 4000

Testing Noise Model: Gaussian(0,25)

Figure 13: Top: Original, Middle: Noisy, Bottom: Reconstructed

Squared Error Formulation

Dataset: MNIST
Training Noise Model: Masking Noise
Fraction of Masked Pixels in Training: 30%
Hidden Units: 500
Testing Noise Model: Salt and Pepper Noise
Fraction of Masked Pixels in Testing: 30%

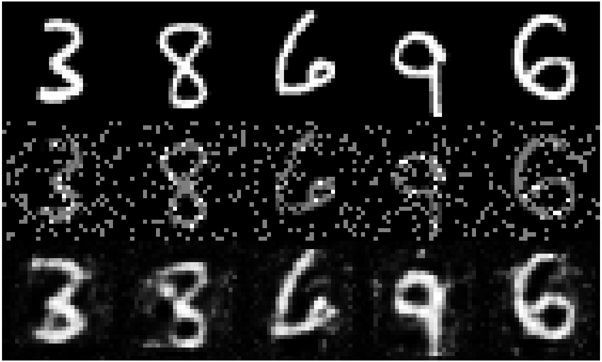

Figure 14: Top: Original, Middle: Noisy, Bottom: Reconstructed

Conclusion

- Effective on the MNIST dataset (limited variety of images), easy to generalize with 1 hidden layer
- Poor performance with CIFAR; complex natural images, require deeper representation
  - Even 8000(!) units doesn't help
- Squared loss is more robust to changes in the noise model
- Denoising is widely applicable in Multimedia Systems
  - Wireless Image Sensors with noisy data
  - Preprocessing/postprocessing step in compression
  - Used in Space Imaging Applications
  - Medical Imaging Applications

References

[1] Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *The Journal of Machine Learning Research* 11 (2010): 3371-3408.