Extracting Robust Features with Denoising Autoencoder

{Annoop Shahhare , Gandhar Vaidya, & Gururaj Mulay} CS545 - Machine Learning

Colorado State University

1. Objectives

Implement an unsupervised learning model using stacked Denoising Autoencoders to initialize deep architectures.
Test the algorithm on MNIST dataset and demonstrate its advantages over deep networks that use ordinary autoencoders.

2. Introduction

Auto-encoders
Auto-encoder forms a distributed representation for the input data, typically for the purpose of dimensionality reduction.

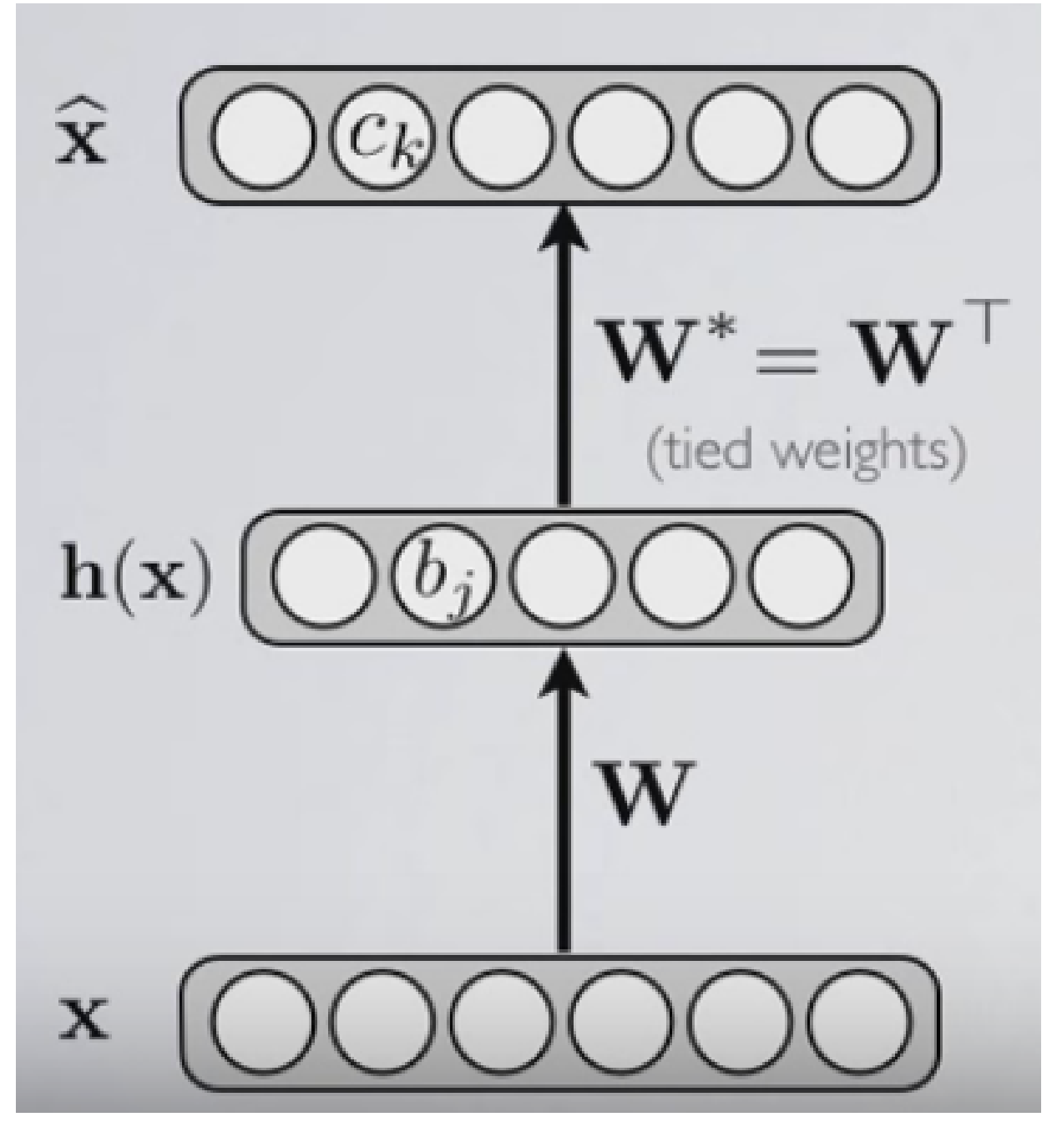


Figure 1: Auto-encoder

Denoising Auto-encoders
Denoising auto-encoder(DAE), a variant of autoencoder, encodes partially corrupted inputs and tries to undo the effect of a corruption by reconstructing the original inputs. **DAE ensures the autoencoder gets properly trained by having partial data instead of merely getting a copy of the input**

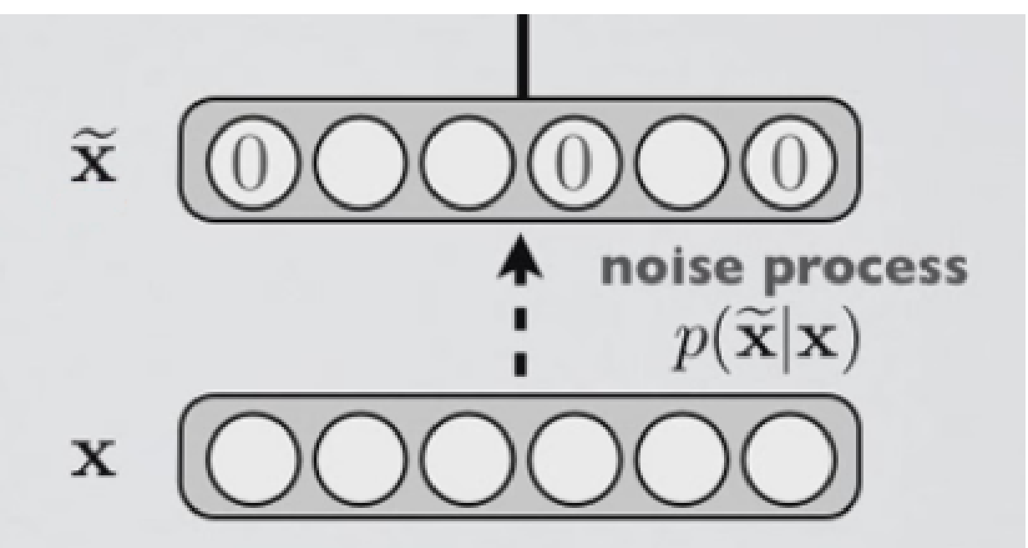


Figure 2: Denoising Auto-encoder

Stacked Denoising Auto-encoders
Deep networks can achieve a better solution if they are 'pre-trained' and initialized using denoising autoencoder. It enables the network to learn robust features leading to higher accuracy in prediction. For this we use a stack of denoising autoencoder to initialize the weights at each layer of deep network.

3. Methods

The basic framework of a denoising autoencoder is shown below.

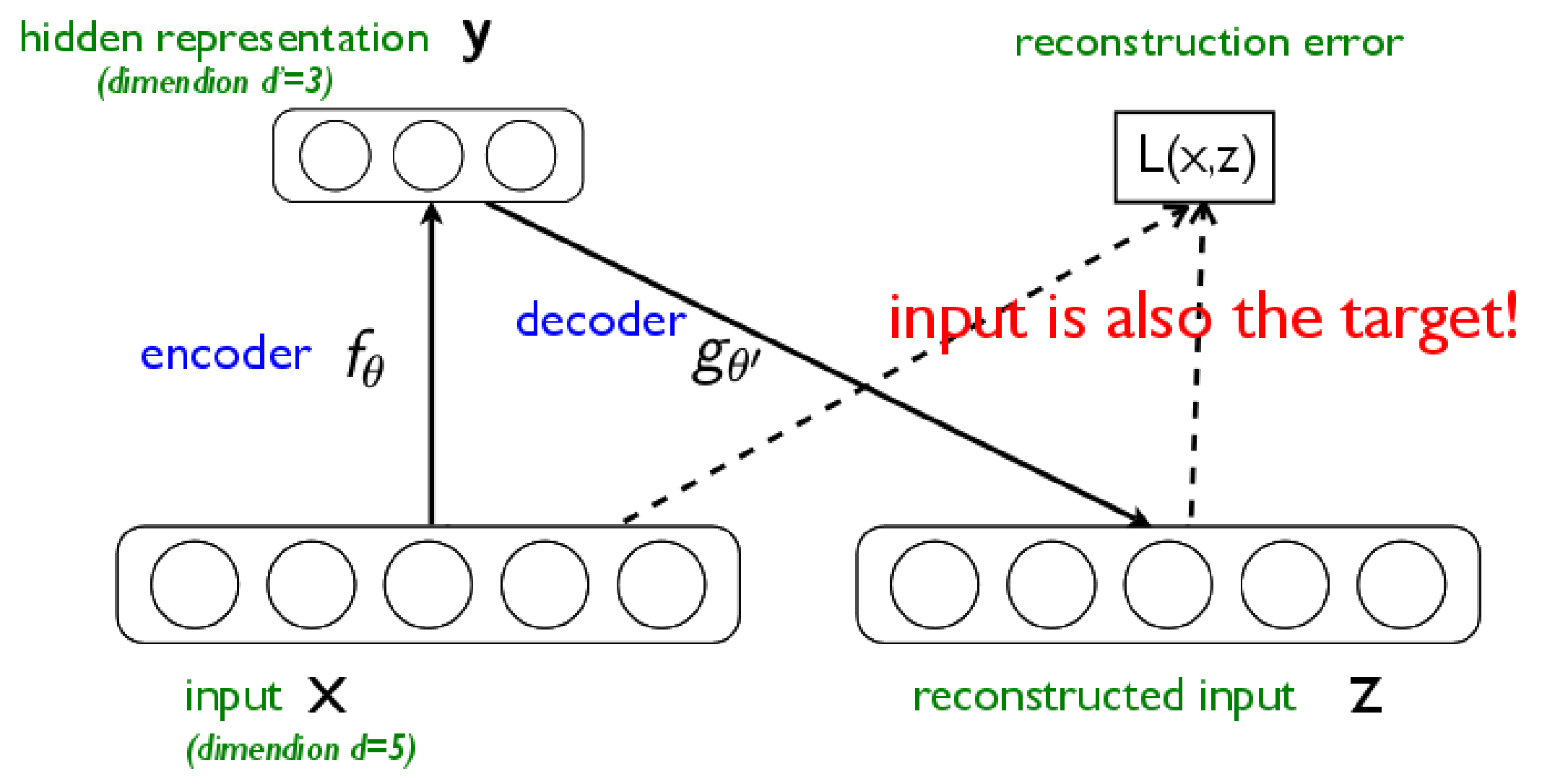


Figure 3: Framework for DAE

The input x is corrupted to produce x' by replacing some data with 0 values and the autoencoder is trained to fill in these missing values.
Our aim is to have z as close as possible to x i.e. the original uncorrupted input. This means we have to minimize the reconstruction error.
Error can be computed using squared error or a log-likelihood function.

4. Algorithm

1. Take an input vector x which has binary values (0,1) of size d
2. Project this vector x to y-
   y= $f_\theta(x)$ = s(Wx + b)
3. y is then mapped back to z in input space
   $z = g'_\theta(y) = s(W'y + b')$ with θ'=(W',b').
4. Next , the algorithm focuses on minimizing the reconstruction error.
   argmin$\frac{1}{n}$ $\sum_{i=1}^{n} L(x^i, z^i)$ where L(x,z) = $||x - z||^2$(squared error)

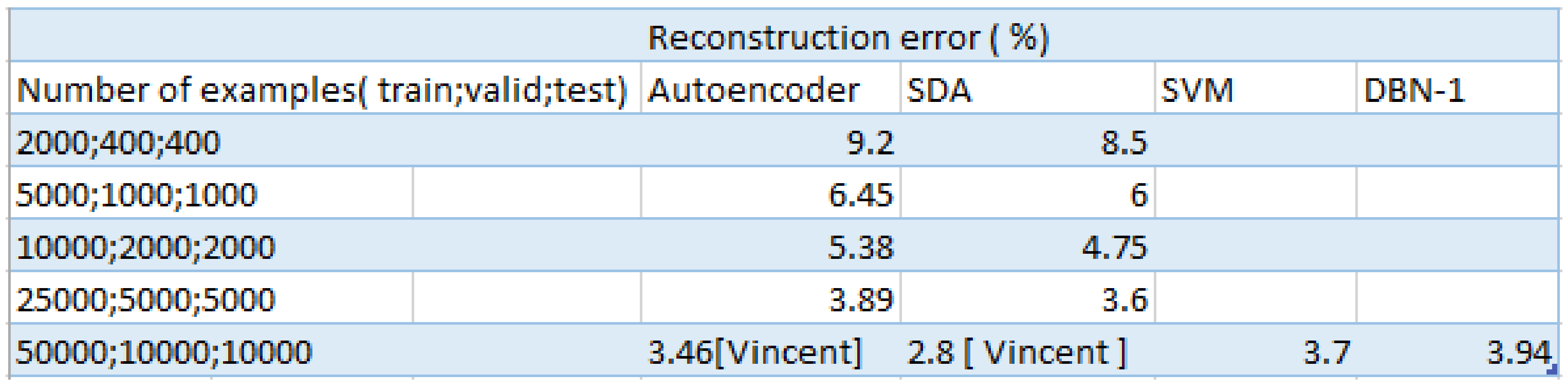
5. MNIST Dataset

MNIST: Mixed National Institute of Standards and Technology

It's a large set of handwritten digits used for NN training. It has 50,000 training images, 10,000 validation images, and 10,000 testing images.
Each images has a size of 28 X 28 pixels.
The digits have been size-normalized and centered in a fixed-size image.

6. Results

Following table shows variation of test error against number of training examples on stacked DAE with 3 weight vector layers with 10% noise.



	Reconstruction error (%)			
Number of examples(train;valid;test)	Autoencoder	SDA	SVM	DBN-1
2000;400;400	9.2	8.5		
5000;1000;1000	6.45	6		
10000;2000;2000	5.38	4.75		
25000;5000;5000	3.89	3.6		
50000;10000;10000	3.46[Vincent]	2.8 [Vincent]	3.7	3.94

Following images show the filter representation of some selected rows of weight vector W for one of the the stacked autoencoder using different levels of input corruption.

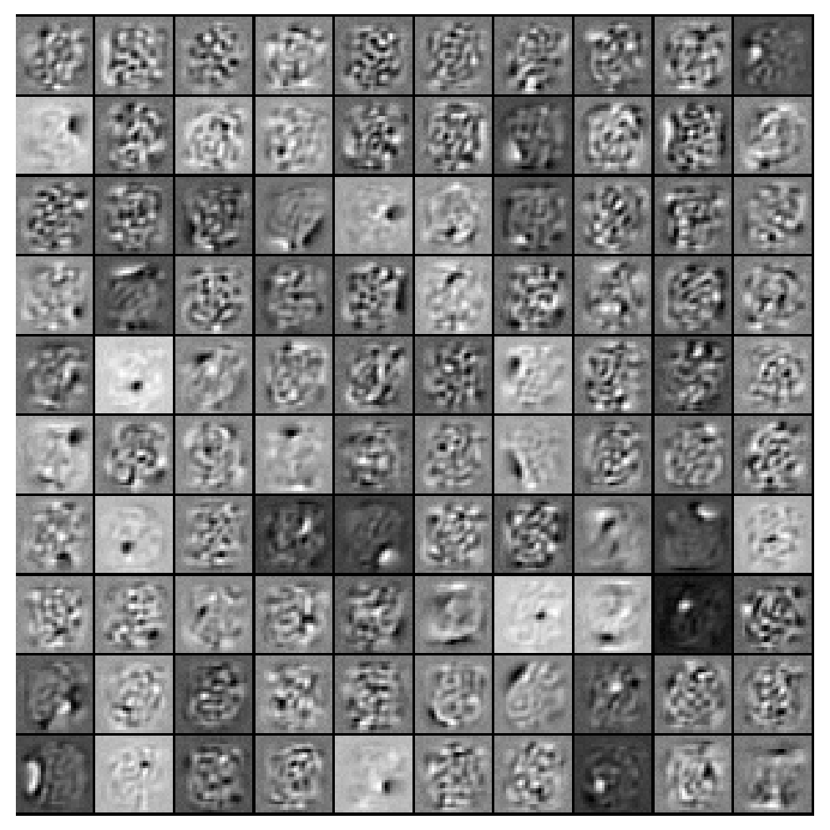


Figure 4: result without corrupting data

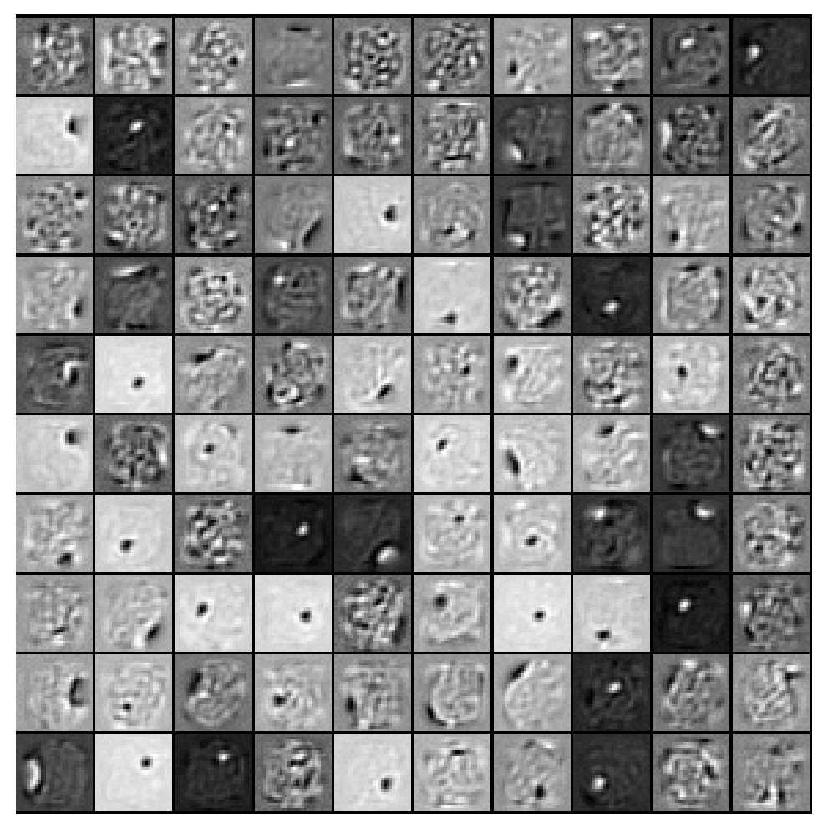


Figure 5: result when corruption level is 20%

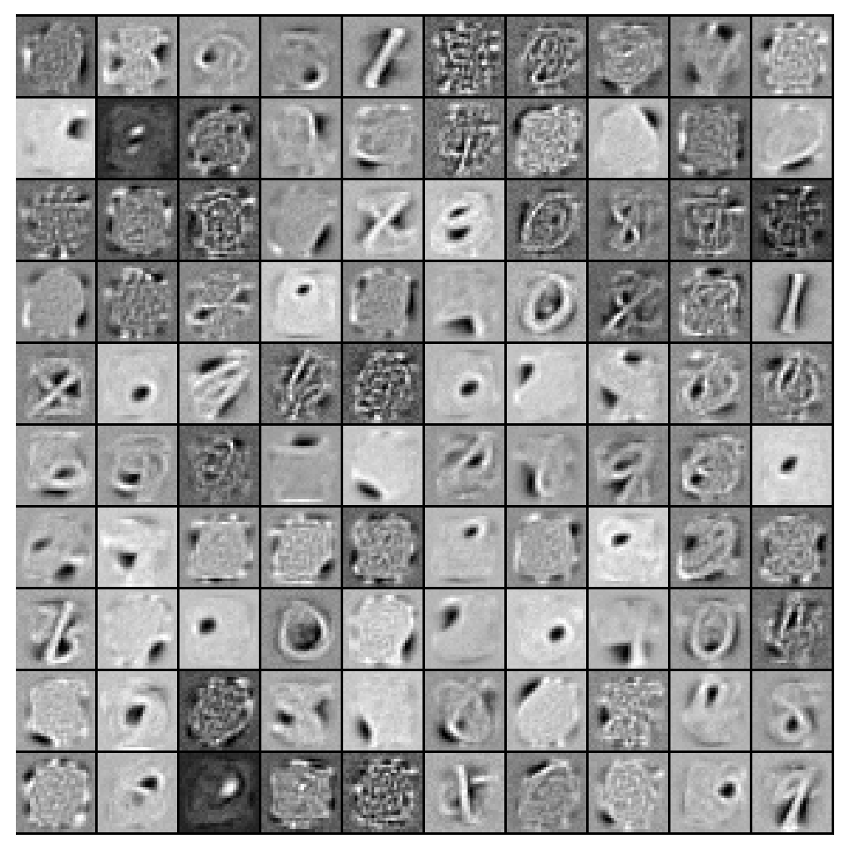


Figure 6: result when corruption level is 70%

We observe that in case of added noise, the autoencoder is able to extract interesting features that are not local to input dimension as opposed to that we see in the case of ordinary autoencoder in second figure. **They behave in less local manner and learn the features that are spread out over the larger input dimension (784) by becoming more sensitive to them.**

7. Conclusions

- Corrupting the inputs before feeding to autoencoder leads to a better representation of input dataset compared to ordinary autoencoder.
- Pre-training the deep networks with stacked denoising autoencoder and then fine-tuning the weights using different algorithms gives better accuracy on test data. This method avoids poor solutions obtained with random initialization of weight vectors.
- As we increase the noise level, denoising training forces the filters to differentiate more, and capture more distinctive features. Higher noise levels tend to induce less local filters.

8. References

- Vincent et al., JMLR-11 (2010)
- Vincent et al., ICML-25 (2008)
- Hinton et al., Science Magazine (2006)
- MNIST link: http://yann.lecun.com/exdb/mnist/
- http://www.deeplearning.net/tutorial/dA.html