Image Denoising Autoencoders

Applied Machine Learning Final Project Final Report

Amir Mehrpanah

June 2020

Contents

1 Abstract

An autoencoder is an unsupervised machine learning algorithm that takes an image as input and tries to reconstruct it using fewer number of variables from the bottleneck also known as latent space (see Figure 1). The image is majorly compressed at the bottleneck. The compression in autoencoders is achieved by training the network for a period of time and as it learns, it tries to best represent the input image at the bottleneck. [8] Sometimes, the input images for autoencoders can be noisy. In that case, the deep learning autoencoder has to denoise the input images, get the hidden code representation, and then reconstruct the original images. [6] Denoising is one of the useful applications of autoencoders in deep learning. Denosing autoencoders are great tools for improving image quality in satellite imagery [4], medical imagery[2] and many other areas.

2 Introduction

Our major goal in this project was familiarizing the student with image processing basics and help him join the GNN research group at Shahid Beheshti

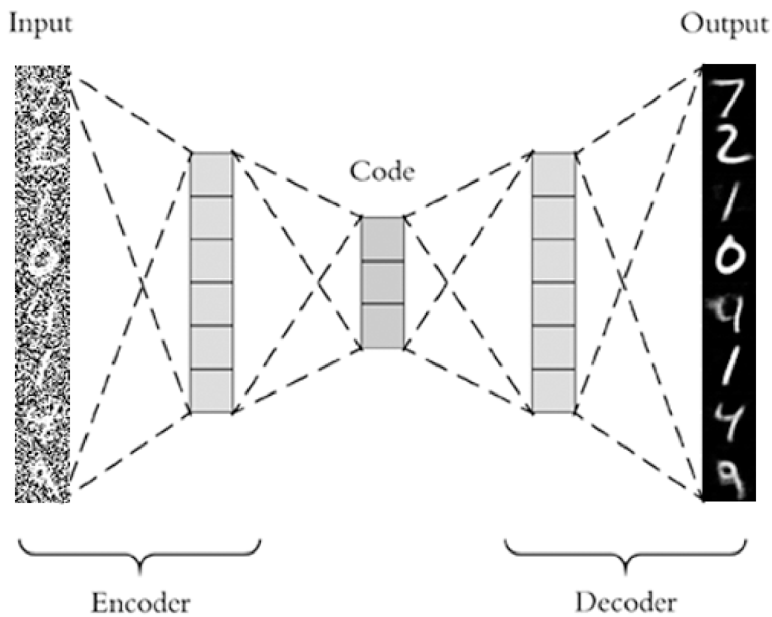

Figure 1: A conceptual map of an autoencoder [7]

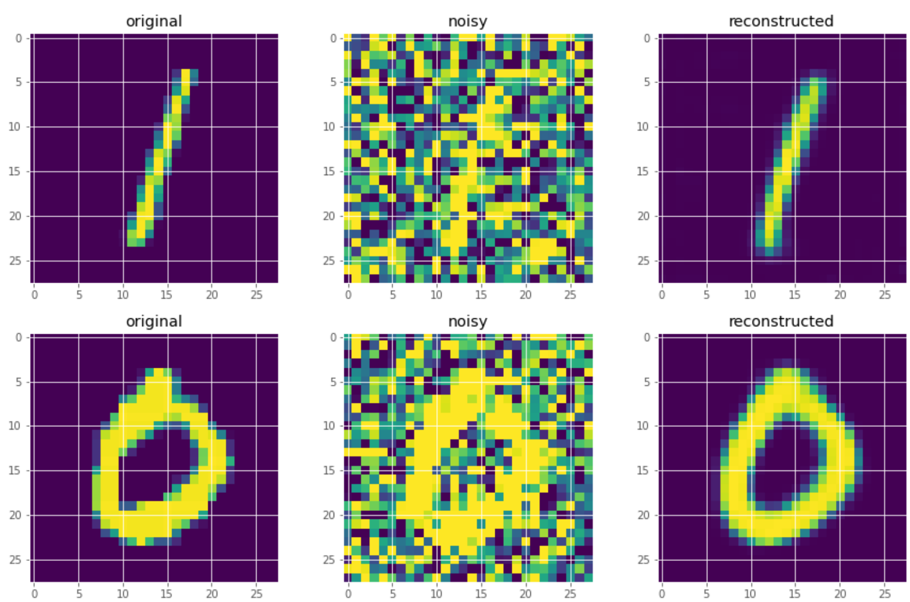

Figure 2: Sample input and output of the selected network [7]. The right image is the denoised version, the left image is the origial version and the middle one is the noisy image fed to the network chosen randomly from MNIST data set.

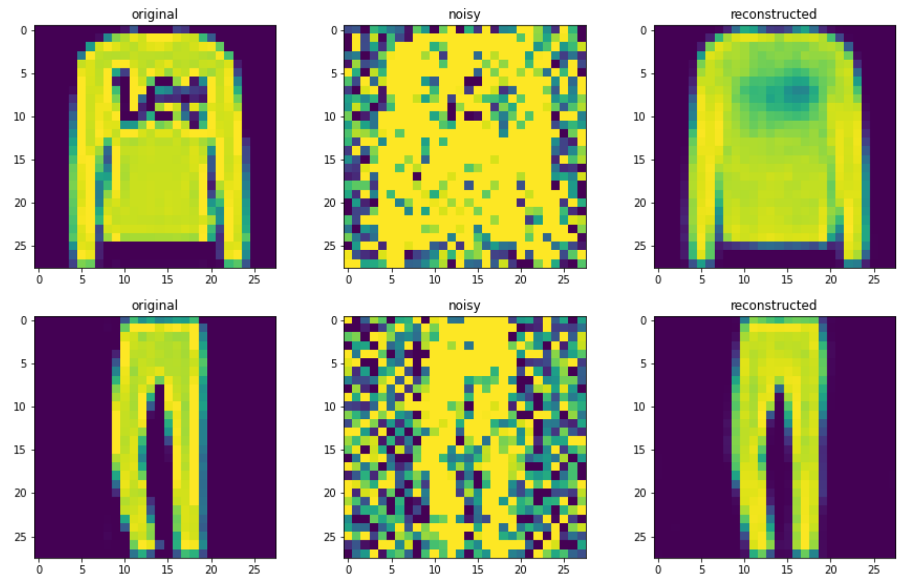

Figure 3: Sample input and output of the selected network [7]. The right image is the denoised version, the left image is the origial version and the middle one is the noisy image fed to the network chosen randomly from Fashion-MNIST data set.

University of Tehran. Furthermore we tried to gain basic insight into autoencoders both theoretically and practically. Due to time and hardware limitations we had to restrict our project scope to small data sets[1]. However, our approach in this project was, to a great extent, exploratory. We used Tensorflow Keras api equipped with some of sickit learn's helper functions and python 3.6 on Google (or Kaggle) cloud servers for cells with large run times. We also took advantage of Tensorboard for graph visualization and comparing the metrics.

3 Results and Evaluation

According to the project plan offered in our proposal, we first tried to find the best performance on an small data set among the available networks. In the first step, four different architectures [3],[7],[1],[8] implemented, trained and the best one [7] selected (see Figures 2 and 3.) based on metrics recorded by Tensorboard. After determining best available architecture, we then prodceeded smoothly to larger data sets [2], on which the network performed poorly (see Figure 4). We wanted to identify the cause of this problem (high noise factor or low network complexity) so we fed the original images (without noise) to the autoencoder

[1]The term "small data sets" may refer to MNIST, Fashion MNIST.

[2]Such as CIFAR as it is slightly larger than MNIST and Fashion-MNIST.

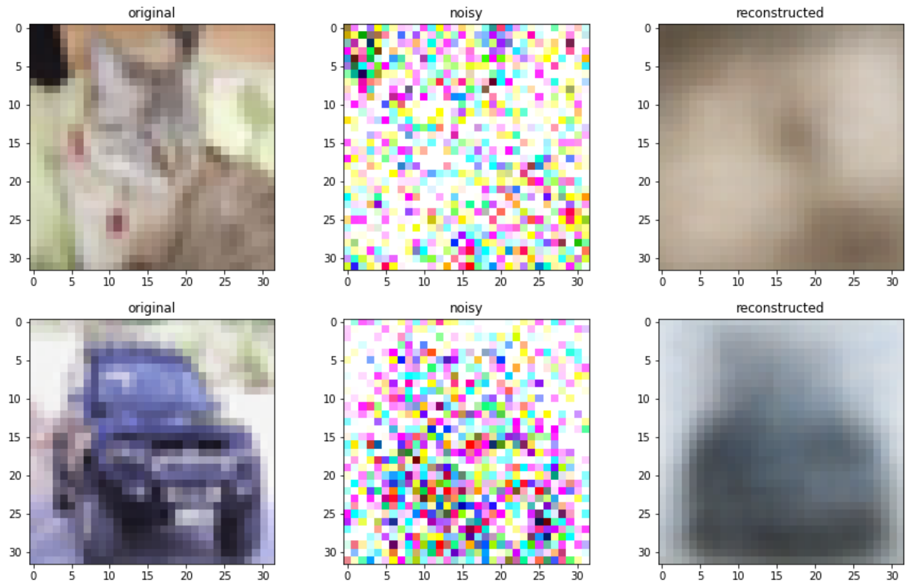

Figure 4: Sample input and output of the selected network [7]. The right image is the denoised version, the left image is the origial version and the middle one is the noisy image fed to the network chosen randomly from CIFAR data set. As you can see network failed to reconstruct the noisy image.

Figure 5: Recorded metrics of training the selected network (without any change in layers and nodes) by Tensorboard. Red line shows metrics on CIFAR images without noise and orange line shows metrics on noisy images.

and got similar results (see Figure 5) which indicates that the problem lies more in the low network complexity and less in the high noise factor. In the next step we increased network complexity to cover CIFAR as well.

generating new samples was a side mission in our proposal that we accomplished. Two representations obtained by applying only the trained encoder on two random samples, then connected by a linear interpolation and fed to the trained decoder[3]. We refer the reader to our Github repository for more information on the results.

4 Future Work

There is an infinite space for exploration, tweaking parameters, trying totally new structures and gaining deeper knowledge about autoencoders. Although we tried to search, as long as time frames allowed us, and analyze different structures, there is still enormous untested novel possibilities. We also reviewed papers about the cutting-edge technologies such as denoising generative adversarial networks [5] as a path for future work.

5 Supplementary Information

Tanks to Mr.Y.Taheri Yeganeh, my former proposal (Binary Tree Integrated Ensembles), after reviewing relevant literature, revised substantially to lie within the field of image processing. Due to COVID-19 pandemic we could not use of university's computing clusters and suffered from a serious lack of resource and minor changes took a long time.

[3]This process done for MNIST and Fashion-MNIST data sets.

References

[1] Chris. *Building an Image Denoiser with a Keras autoencoder neural network*. 2019. URL: https://www.machinecurve.com/index.php/2019/12/20/building-an-image-denoiser-with-a-keras-autoencoder-neural-network/.

[2] L.Gondara. "Medical Image Denoising Using Convolutional Denoising Autoencoders". In: (2016), pp. 241–246.

[3] Nidhi Mantri. *Build and use an Image Denoising Autoencoder model in Keras*. 2020. URL: https://iq.opengenus.org/image-denoising-autoencoder-keras/.

[4] Saikat Basu; Sangram Ganguly; Supratik Mukhopadhyay; Robert DiBiano; Manohar Karki; Ramakrishna Nemani. "DeepSat – A Learning framework for Satellite Imagery". In: (2016).

[5] Zailiang Chen; Ziyang Zeng; Hailan Shen; Xianxian Zheng; Peishan Dai; Pingbo Ouyang. "DN-GAN: Denoising generative adversarial networks for speckle noise reduction in optical coherence tomography images". In: *Elsevier BV* 55 (2020), p. 101632. DOI: https://doi.org/10.1016/j.bspc.2019.101632.

[6] Sovit Ranjan Rath. *Autoencoder Neural Network: Application to Image Denoising*. 2020. URL: https://debuggercafe.com/autoencoder-neural-network-application-to-image-denoising/.

[7] Adrian Rosebrock. *Denoising autoencoders with Keras, TensorFlow, and Deep Learning*. 2020. URL: https://www.pyimagesearch.com/2020/02/24/denoising-autoencoders-with-keras-tensorflow-and-deep-learning/.

[8] Aditya Sharma. *Implementing Autoencoders in Keras: Tutorial*. 2018. URL: https://www.datacamp.com/community/tutorials/autoencoder-keras-tutorial.