A Better Autoencoder for Image: Convolutional Autoencoder

Yifei Zhang[1[u6001933]]

Australian National University ACT 2601, AU
u6001933@anu.edu.au

Abstract. Autoencoder has drawn lots of attention in the field of image processing. As the target output of autoencoder is the same as its input, autoencoder can be used in many useful applications such as data compression and data de-nosing[1]. In this paper, we compare and implement the two auto encoders with different architectures. The first autoencoder is the simple autoencoder(SAE) with one hidden layer. Another autoencoder is and convolution autoencoder[9]. We compare these two autoencoders in two different tasks: image compression and image de-noising. We show that convolutional autoencoder performs better than the simple autoencoder.

Keywords: Autoencoder · Convolutional Autoencoder · Deep learning

1 Introduction

In machine learning, autoencoder is an unsupervised learning algorithm with the input value as the same as the output value aiming to transform the input to output with least distortion[1]. This technique is widely used in compressing data[12], which helps to reduce the storage usage, fasting required time for the same computation, improving performance by removing redundant variables[7], visualizing high dimensional data[11] and removing noise from the original data. Recently year deep learning has been achieved great success in the field of computer science[14][13]. One of the deep learning architecture convolution neural network show amazing ability to extracting features of images[13]. We wonder if we can leverage the power of convolution neural network to improve the performance of simple autoencoder. In this paper, we introduce a more sophisticated autoencoder using convolution layers[9], we compare convolution autoencoder to the simple autoencoder in different tasks: image compression and image de-noising. We show that convolution autoencoder outperforms the simple one.

We organize this paper in the following way: Sec.2 details the method which includes the dataset, the architecture of convolution autoencoder and simple autoencoder and different tasks to be tested on. Sec.3 shows the setup and procedure of the experiment. We conclude the result and finding in Sec.4. Future work is detail in Sec.5

2 Method

2.1 Dataset

The dataset we chose is LFW (labeled face in the wild home) [10]. It contains more than 13,000 face picture of different people. We random sample 300 of them. Since this paper focuses on dealing with the human face, we manually crop the main face out in order to reduce the noise in the background.

2.2 Tasks

Image compression Autoencoder can be used for image compression. In this task, the size of hidden layer in the autoencoder is strictly less than the size of the output layer. Training such autoencoder using backpropagation with the input values as the exactly the same as the target values force autoencoder to learn the low dimension representation of the input data. The activation of hidden layer is the compressed data. The preceding part of this network is the encoder of the image and the last part of this network is treated as the decoder. Fig.2.2

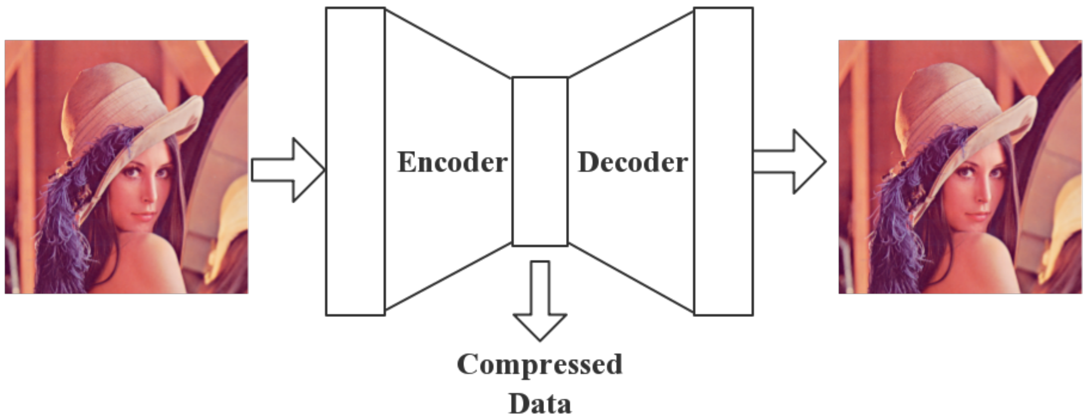

Fig. 1. Illustration of image compression

Image de-noising Another application of autoencoder is that it can be used for image de-noising. In the image de-noising task, we treat the autoencoder as a non-linear function that can eliminate the effect of noises in the image. We train this network by feeding the image with random noise (Gaussian noise) and the target of output is the original image without noise. This encourages the autoencoder to learn a function that removes the noise and reconstruct the image. Fig.3

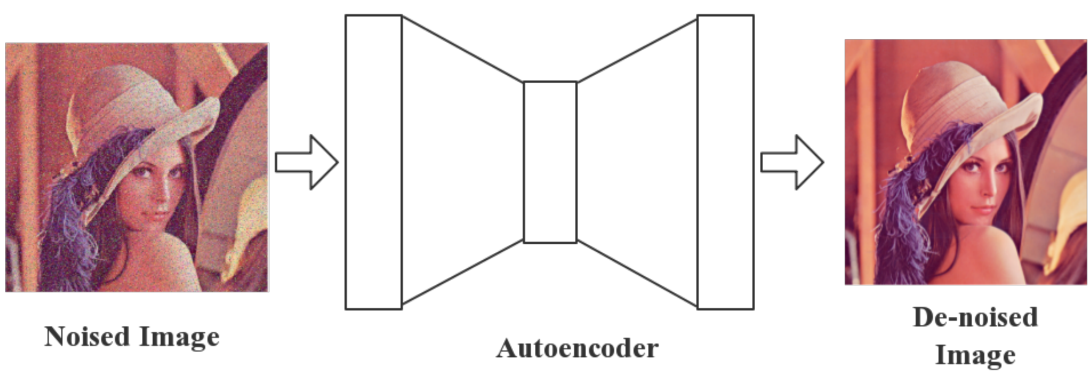

Fig. 2. Illustration of image de-noising

2.3 Different Autoencoder architecture

In this section, we introduce two different autoencoders: simple autoencoder with three hidden layers(AE), convolutional (CAE) autoencoder.

Simple Autocoder(SAE) Simple autoencoder(SAE) is a feed-forward network with three 3 layers. The connections between layers are fully connected. Units in the previous layers are connected to all units in the next layer. The size of the input layer and output layer are equal to the image size which is 64×64 in our case. The size of hidden layer is 32×64. By making the target of the output be the same as the original image, the autoencoder is forced to learn the compressed representation with no information lost. Here, the compressed representation is 32×64 dimension.

Convolutional Autoencoder(CAE) Convolutional autoencoder extends the basic structure of the simple autoencoder by changing the fully connected layers to convolution layers. Same as the simple autoencoder, the size of the input layer is also the same as output layers but the network of decoder change to Convolution layers and the network of decoder change to transposed convolutional layers. We show this network in Fig.3. and detail the hyper-parameters of each convolution layers that we implement (kernel size, stride, padding) in Table 1.

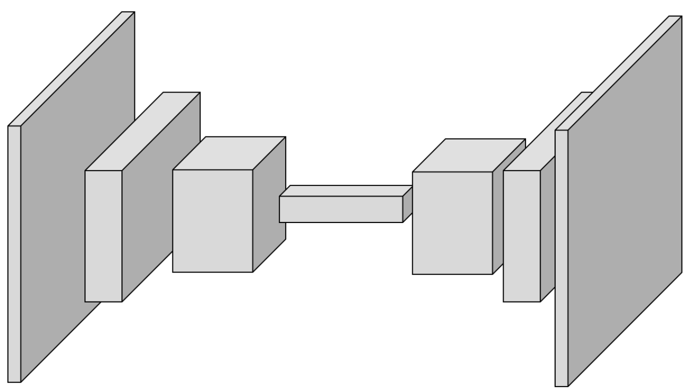

Fig. 3. Structure of convolutional autoencoder: each gray box represent the one convolution layer, we ignore the pooling layer this picture

Table 1. Detail of different layers o the convolutional autoencoder

Layer	Type	Window size	stride	padding
1	Conv	3	1	1
2	Pool(max)	2	1	1
3	Conv	3	1	1
4	Pool(max)	2	1	1
5	Conv	3	1	1
6	Pool(max)	2	1	1
7	TransConv	5	2	1
8	TransConv	3	2	1
9	TransConv	2	2	1
10	TransConv	3	1	1

3 Experiment and Discussion

Image compression In order to make the result comparable, we manually set the compression representations of both autoencoder the same dimension by adjusting the number of units and channels in the hidden layers. For the convolutional autoencoder, we follow the same setting described in Table 1 and Fig.3. For simple autoencoder, we change the hidden layer size to 64×32. We chose learning rate 10^{-3} and train two autoencoders 3000 epochs. We record the loss of each epochs 5 and display the reconstructed image of the last epoch in Fig 4. Clearly, the reconstruction result of CAE is clearer than the SAE, which indicating CAE can compress data with less information lost even if the free parameters of CAE is much less than the number of free parameters of SAE. In addition to the restore quality of the image, CAE take less training time to get the tolerable loss Fig 5.

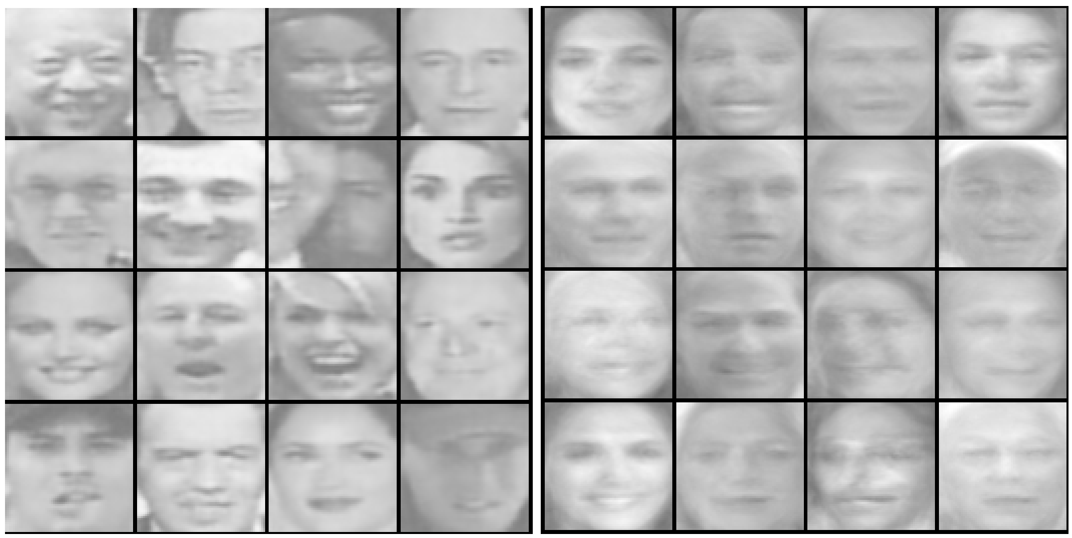

Fig. 4. 16 reconstructed images in the last epoch,left is the output of CAE; the right is the output of SAE

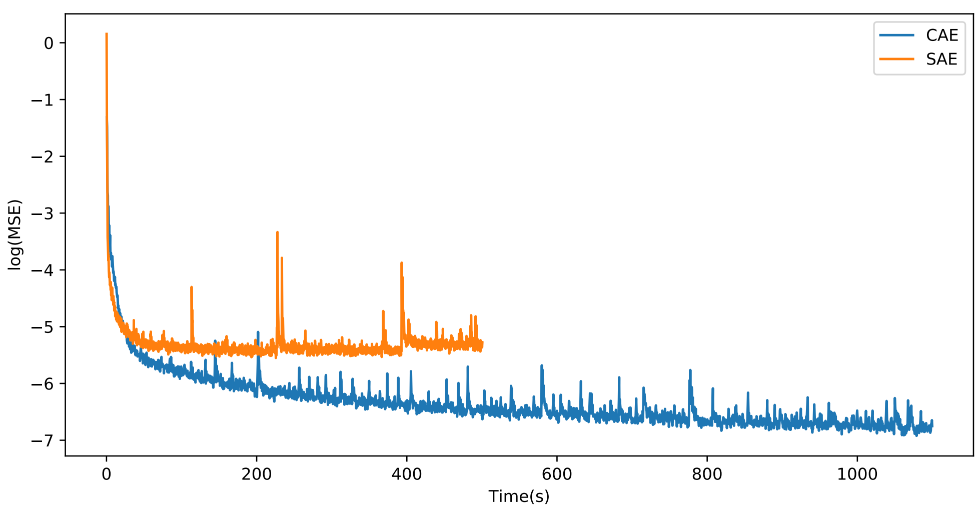

Fig. 5. Loss curve of CAE and SAE with time

Image De-noising We further compare these two autoencoders in the image de-noising task. We add Gaussian noise to the images. Since we do not compress data anymore, there is no need to make the size of the hidden layer be strictly less than the input layers. We change the size of the hidden layer to 64×64 of both autoencoders which enables a bigger search space to learn the noise pattern. We follow the same setting used in image compression and train the two autoencoders. We show the noised image and de-noised picture in Fig 6 and Fig7. We can see that the face in the SAE is more

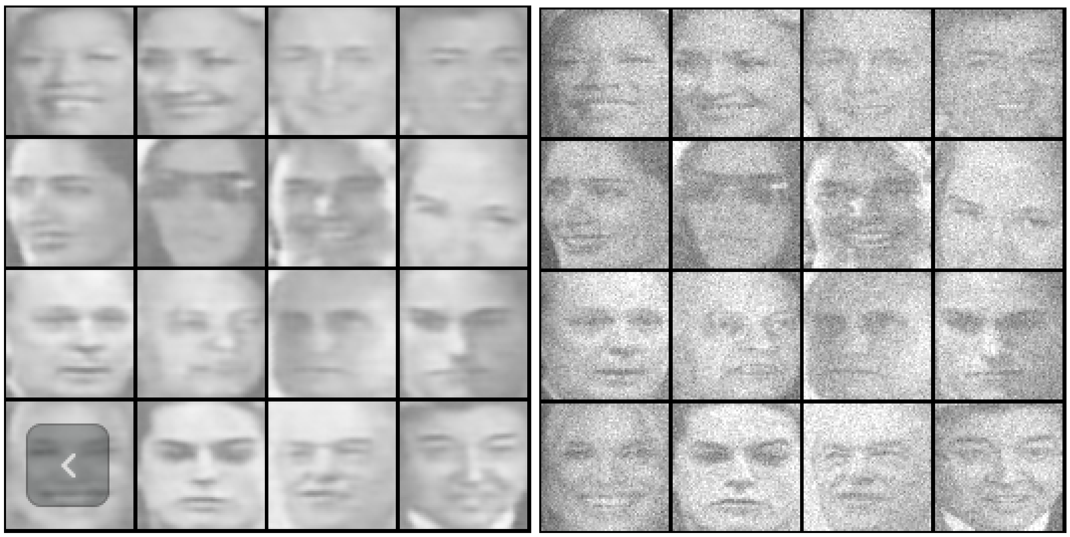

Fig. 6. De-noised result of CAE: left is the de-noised image and right is noised image

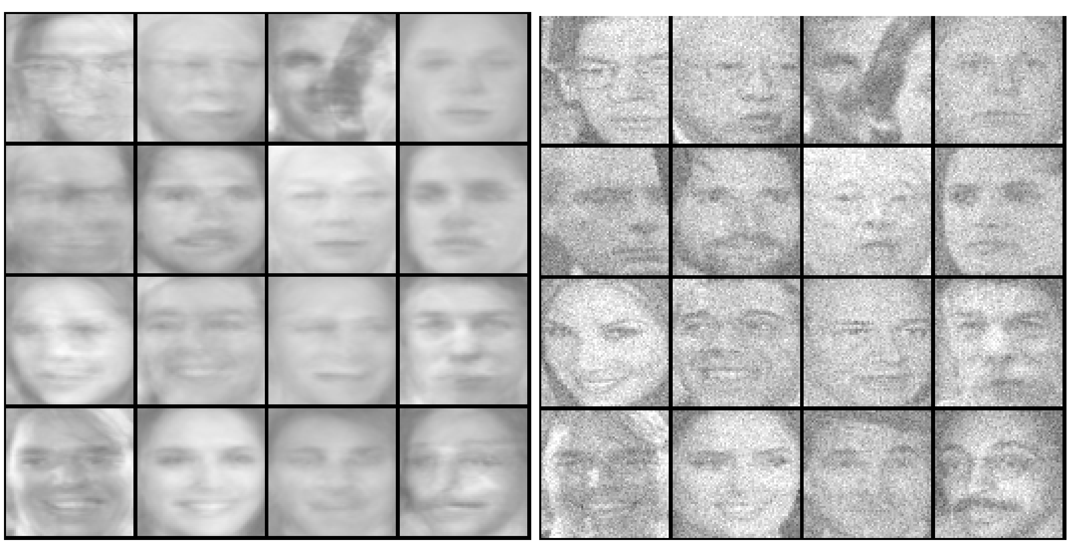

Fig. 7. De-noised result of SAE: left is the de-noised image and right is noised image

blur and the faces overlap each other. For some face, we could not even tell if the person is the same as the input person. However, the face of CAE are much more clear and we could still recognize it is

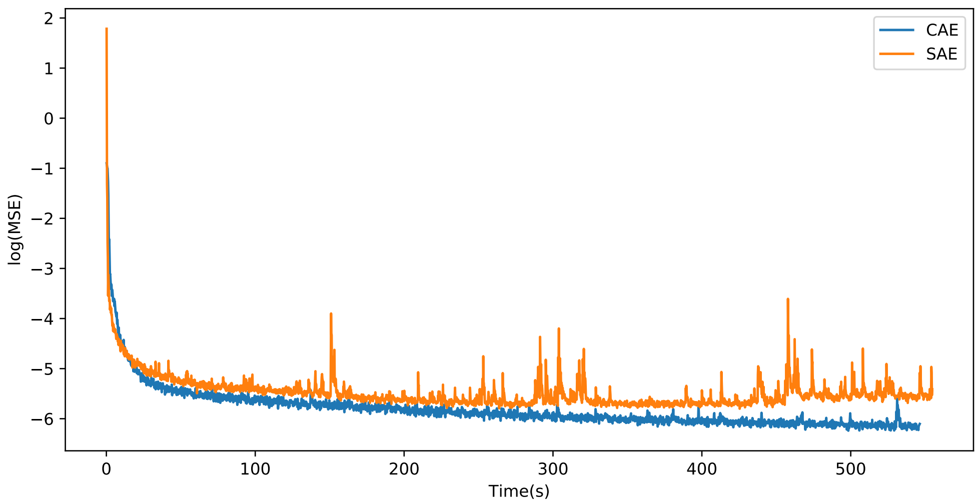

Fig. 8. loss curve of the SAE and CAE in image de-noising task

the same person in the input image. In addition to that, training using the same number of epochs with the same batch size, the MSE loss of CAE is less than SAE. Thus, for image de-noising, CAE is better than SAE.

4 Conclusion

From the experiment above, we see a improvement in both image compression and image de-noising by using convolution autoencoder. Some reason account for the improvement would be:

- Since we use image data, convolution layer is better to capture the spatial information in the image.
- Rather than using one hidden layer in the SAE, CAE use multiple layers to extract the high-level features in the image, which give a better representation.
- The number of free parameter in the fully connected layer is larger than the number of free parameter in multi convolution layer, which makes the simple autoencoder hard to train and cost of time.

5 Future Work

Beside the convolutional autoencoder, Variational autoencoder(VAE)[7] is another autoencoder that worth investigating. Unlike the autoencoder of CAE and SAE. VAE encoder data into a distribution. It would be interesting to explore it in the future work

References

1. Gedeon, T. D., Harris, D.: Progressive image compression. In Neural Networks, 1992. IJCNN., International Joint Conference on (Vol. 4, pp. 403-407). IEEE (1992).
2. Gedeon, T. D., Harris, D.:Network reduction techniques. In Proceedings International Conference on Neural Networks Methodologies and Applications, Vol. 1, pp. 119-126 (1991)
3. Jolliffe, I. T.: Principal component analysis and factor analysis. In Principal component analysis pp. 115-128. Springer, New York, NY(1986)..
4. Nasrabadi, N. M. : Pattern recognition and machine learning. Journal of electronic imaging, 16(4), 049901 (2007).
5. Dua, D., Karra Taniskidou, E.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science(1986).
6. Sra, S., Dhillon, I. S.:Generalized nonnegative matrix approximations with Bregman divergences. In Advances in neural information processing systems pp. 283-290 (2006).
7. Kingma, D. P., Welling, M.:Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013).
8. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
9. Mao, X., Shen, C., Yang, Y. B.:Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In Advances in neural information processing systems pp. 2802-2810 (2016).
10. Huang, G. B., Ramesh, M., Berg, T., & Learned-Miller, E. (Labeled faces in the wild: A database for studying face recognition in unconstrained environments (Vol. 1, No. 2, p. 3). Technical Report 07-49, University of Massachusetts, Amherst (2007).
11. Maaten, L. V. D., & Hinton, G. Visualizing data using t-SNE. Journal of machine learning research, 9(Nov), 2579-2605(2008).
12. Baldi, P. Autoencoders, unsupervised learning, and deep architectures. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning (pp. 37-49)(2012, June).
13. Krizhevsky, A., Sutskever, I., & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105)(2012).
14. LeCun, Y. (2015). LeNet-5, convolutional neural networks. URL: http://yann. lecun. com/exdb/lenet, 20.