IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 5.731

Volume 10, Issue 1, October 2021

A Review on Autoencoder and its Application

Meeth Shetty

Student M.Sc., Department of Information Technology

Sir Sitaram and Lady Shantabai Patkar College of Arts and Science, Mumbai, India

meethshetty46@gmail.com

Abstract: *Deep Learning is gaining a lot of attention due to the availability of high computation power and better optimization techniques, which has helped them to achieve accuracies greater than a human. To train supervised models we need to have a large amount of labeled data, which is not the case with unsupervised methods. Unsupervised learning methods benefit from the lack of necessity for labeled training data. Autoencoders are a very popular implementation of the unsupervised model. They learn only the most important features of the input; hence they can represent the data well. They can be used for many tasks like dimensionality reduction, anomaly detection, etc. This review aims to provide a basic idea about autoencoder, its variants, and applications.*

Keywords: Autoencoder, deep learning, unsupervised learning.

I. Introduction

Artificial Intelligence can be referred to as the development of systems that are intelligent enough to make decisions like humans. There has been great progress in AI today due to the availability of high computation power and better optimization techniques. In recent years, Deep Learning has gained a lot of traction due to its high accuracy compared to other AI techniques. This is due to its ability to learn under unsupervised conditions. Unsupervised learning is gaining a lot of attention in recent years, as training deep learning models require a lot of training data, which is hard to create.

An autoencoder is a kind of deep learning network used to learn efficient data representation in an unsupervised manner. The expected output of an autoencoder is a representation that is close enough to the input. It is an unsupervised learning method, although technically, it is trained using supervised learning methods. The computing paradigm has now shifted from hardcoded program to soft computing program where the computer is the only told what to do and how to do. It has to predict the answers based on what it had learned from previous experience. Nowadays people want to provide an artificial brain to a computer that why AI is closely related to computing.

II. Autoencoder

Autoencoder is an unsupervised learning algorithm that aims to learn a representation of the input so that it can produce an output similar to the input. An autoencoder performs two activities, encoding, and decoding.

An Autoencoder encodes the input X into a representation using function g. It then decodes the representation using function f and tries to recover the input with minimum loss.

Autoencoder aims to minimize the reconstruction error between this encoding and decoding of the data. This is done by making sure that the model doesn't just copy the input to output. It learns the most important features from the input that can help it reconstruct the data with some minimal loss.

There are three components in an Autoencoder, the encoder that encodes the input, the hidden representation that learns the general representation of this encoded input, and a decoder that decodes an output that is close enough to the actual input.

Based on the type of input we decide the function that will be used to obtain the output. If the inputs are binary, a logistic function makes for sense for the output. If the inputs are real-valued, a linear function or tanh function makes more sense for the output activation function. The function for the output of hidden representation is typically the sigmoid function. For the loss function, the loss function is ideally squared error loss function in case of real-valued

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

IJARSCT
Impact Factor: 5.731

Volume 10, Issue 1, October 2021

output and input; it is sigmoid in case of binary values.

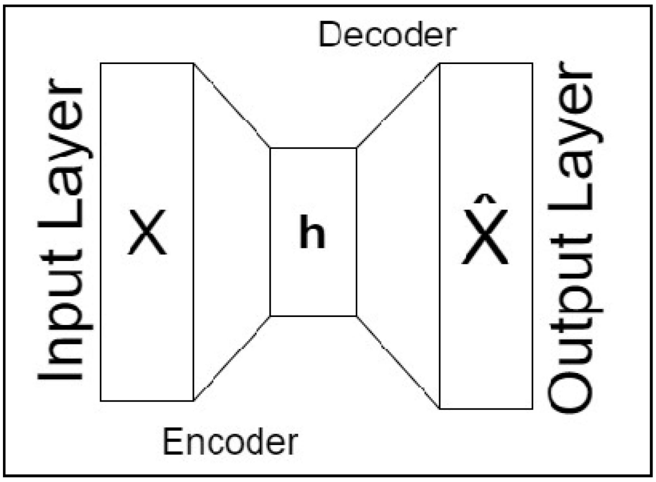

Figure 1: Schema of a basic Autoencoder

III. Types of Autoencoder

3.1 Under Complete Autoencoders

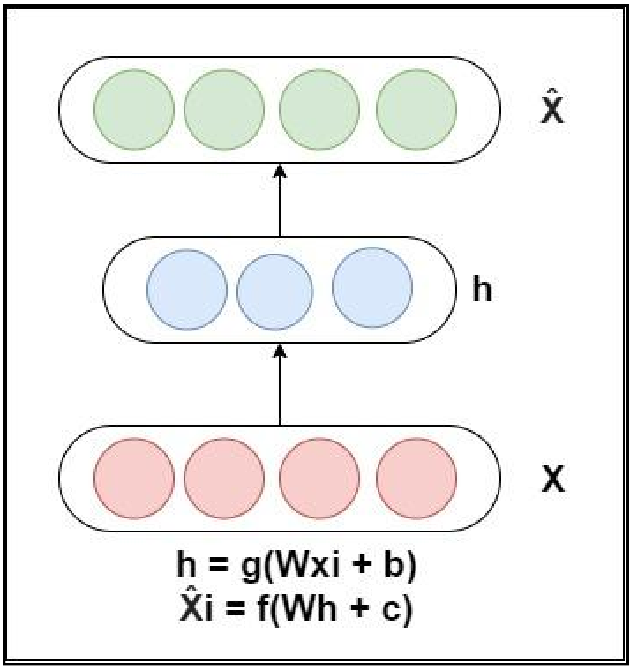

Figure 2: Undercomplete Autoencoder

Under-complete autoencoder's objective is to capture only the most important features of the input. It has a smaller dimension of the hidden layer which makes sure that the hidden layer doesn't simply copy the input. Learning an under-complete representation of data forces the autoencoder to learn only the most important features that will help in the reconstruction of the input with some small reconstruction loss. When the output of the decoder is obtained using a linear function, the autoencoder performs the task of Principal Component Analysis (PCA) which reducing the dimension of the data.

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Volume 10, Issue 1, October 2021

Impact Factor: 5.731

3.2 Overcomplete Autoencoder

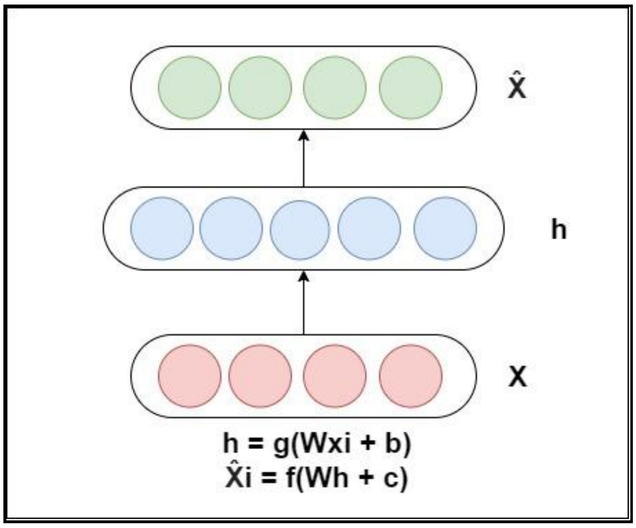

Figure 3: Overcomplete Autoencoder

Overcomplete Autoencoders have more dimensions in the hidden layer than the dimension of input. As there is no restriction on the hidden layer, it may not learn important features, and most likely will copy the input to output. With regularization techniques, it can be used to learn salient features of the input.

3.3 Sparse Autoencoder

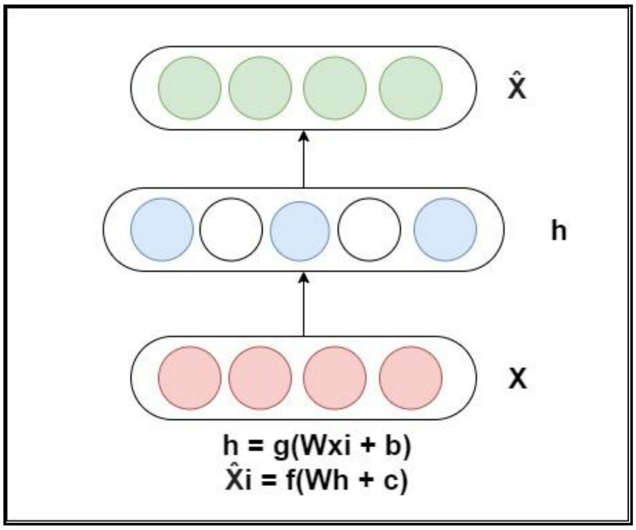

Figure 3: Sparse Autoencoder

Sparse Autoencoders are autoencoders that apply a sparsity regularization on the hidden layer. It aims to keep most of the neurons except the ones which are most likely to learn input features inactive. A neuron is considered inactive if its output is zero or close to zero, and it is considered active when its output is close to 1. This sparsity is introduced by adding the sparsity parameter to the loss function.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 5.731
Volume 10, Issue 1, October 2021

3.4 Denoising Autoencoder

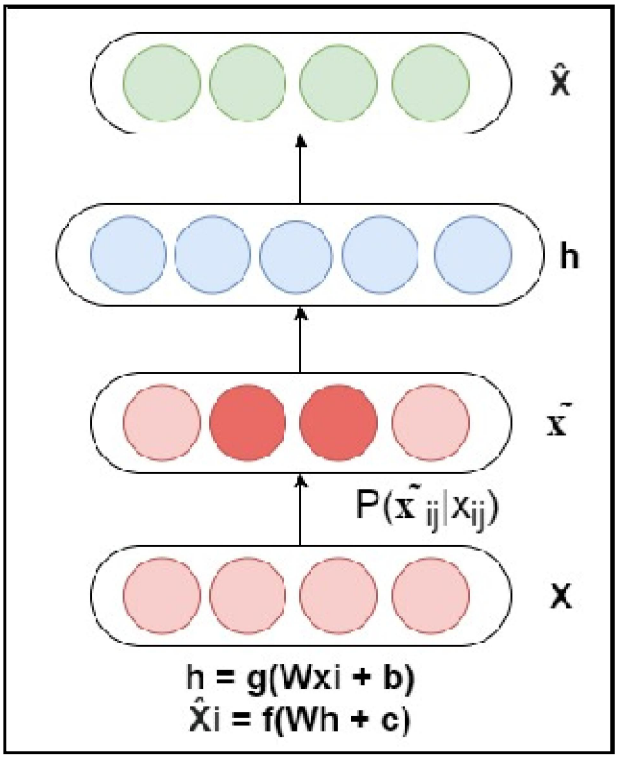

Figure 4: Denoising Autoencoder

In Denoising Autoencoder, to ensure that the hidden layer does not simply copy the input, the input is corrupted using some noise. The corrupted input is fed to the autoencoder, and it is trained to learn from the data to output the original noise-free data. The input is corrupted using a probabilistic process ($P(\tilde{x}_{ij}| x_{ij})$). Now the model will receive a big penalty from the loss function if it simply copies the corrupted input to output. This is for the autoencoder to learn features from the input, and it will learn to output a representation that is close to the noise-free input.

3.5 Contractive Autoencoder

Contractive autoencoders try to prevent the hidden layer from copying the input to output. It makes sure that the autoencoder is less sensitive to small variations in input data and more sensitive to important variations. This ensures that only the most important features of the input are learned. This is done by adding a regularization to the loss function, which forces the network to learn from data. The regularization corresponds to the Frobenius norm of the Jacobian matrix of the hidden layer which learns the features.

3.6 Variational Autoencoder

Variational autoencoders are similar to Generative Adversarial Networks (GANs). They work similar to other autoencoders, the only difference is that the decoder works like a generator model. Generative models are models that are trained on some set of data, from which they are expected to learn some important representations using which they are expected to generate similar output which it has not seen before. It shows promise for many complicated generative tasks like handwritten digits, the generation of new faces, etc.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 5.731
Volume 10, Issue 1, October 2021

IV. Pros and Cons

Autoencoder	Pros	Cons
Undercomplete	• No need for regularization, as it does not copy input to output.	• Can sometimes fail to generalize. • Can overfit when training data is less.
Overcomplete	• Can be better generalized using regularization.	• More likely to copy input to output.
Sparse	• Less likely to overfit. • Only nodes that will learn important features are activated.	• We need to train the nodes to be data dependent.
Denoising	• We achieve a good representation of data. • Easy to implement	• Preliminary mapping is required between corrupted input and output.
Contractive	• More appropriate for feature extraction.	• Traditional contractive autoencoders have high reconstruction error.
Variational	• Useful for generating work.	• A lot of time needs to be given during the sampling process

V. Application of Autoencoder

5.1 Dimensionality Reduction

Data like images or text is usually in a spare high-dimensional representation. The high dimensionality of data can cause the need for higher computation power, hence there is a need for reduction of data dimension. Principal Component Analysis is the traditional method used for this task, but autoencoder can also be used.

As under complete autoencoder has fewer dimensions in the hidden layer, it can learn useful features of data. If the output function of the autoencoder is linear, it acts similar to PCA, hence it can be used for dimensionality reduction.

5.2 Image Denoising

Noise in data can be described as variations in the original data due to corruption of data. These variations in data are not a desirable feature. As the autoencoder is trained to learn the most important features of the data, it can ignore these small variations and output the original image with some minimal loss.

5.3 Image Compression

Image compression is used to deal with the storage issue of the data. In image compression, some of the data in the image are removed to decrease the overall size of the image, at the cost of loss of some data from the image. Autoencoder does dimensionality reduction in the encoding step, which is what the task of image compression is. Though it should be noted that autoencoder can do image compression for only the type of image on which it was trained.

5.4 Feature Extraction

Feature extraction is the task of extracting important features from the data. Autoencoder is built with the task of extracting the most important features from the data, so they are very appropriate for this task.

5.5 Anomaly detection

Anomaly detection is an unsupervised task where the model is trained with some data and then it is expected to identify the data which is different from the one it was trained on. As autoencoders learn the most important features of the data, if data that does not have these learned features arrives, it can notify us. Hence, autoencoders are suitable to be used for anomaly detection tasks like fraud detection, network monitoring, system monitoring, etc.

5.6 Recommendation system

A recommendation system is expected to suggest the customers; products or services based on the products or services they have used in the past. It is a very important part of many e-commerce websites, which helps the business to bring leads. Traditionally Collaborative filters are used to build recommendation systems. AutoRec is an autoencoder model developed for purpose of a recommendation system. There are two types of AutoRec: User-based AutoRec which learns the representation of item preference for certain users and Item-based AutoRec which learns user preference representation for certain items.

VI. Conclusion

Deep Learning methods have one drawback of lack of labeled training data, unsupervised methods can counter this drawback. Autoencoder learns the most important latent representation of the input. This helps in tasks like dimensionality reduction, which is essential if we want to reduce the computation requirement or storage requirement. It can also be applied to tasks like recommendation system, anomaly detection, image denoising, etc. Using autoencoder we can perform generative tasks as well like generating new faces, handwritten numbers, etc.

References

[1]. Sedhain, S., Menon, A.K., Sanner, S., Xie, L.: Autorec: Autoencoders meet collaborative filtering. In: Proceedings of the 24th International Conference onWorldWideWeb Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume, pp. 111–112 (2015).

[2]. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. science 313(5786), 504–507 (2006).

[3]. Wang,W., Huang, Y.,Wang, Y.,Wang, L.: Generalized autoencoder: A neural network framework for dimensionality reduction. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 490–497 (2014).

[4]. Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 733–742 (2016).

[5]. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., van den Hengel, A.: Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection (2019).

[6]. Sedhain, S., Menon, A.K., Sanner, S., Xie, L.: Autorec: Autoencoders meet collaborative filtering. In: Proceedings of the 24th International Conference onWorldWideWeb Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume, pp. 111–112 (2015).

[7]. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. CoRR abs/1312.6114 (2013)

[8]. Hou, X., Shen, L., Sun, K., Qiu, G.: Deep feature consistent variational autoencoder. CoRR abs/1610.00291 (2016). URL http://arxiv.org/abs/1610.00291

[9]. Mitesh Khapra,(2017), Deep Learning-IIT Madras, noc21_cs76.

[10]. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, ICML ’08, pp. 1096–1103. ACM, New York, NY, USA (2008). DOI 10.1145/1390156.1390294. URL http://doi.acm.org/10.1145/1390156.1390294.