AN INTRODUCTION TO AUTOENCODERS

3 authors:

Dibyendu Barman
Government College of Engineering & Textile Technology, Berhampore, WB, India
22 PUBLICATIONS 84 CITATIONS
SEE PROFILE

Abul Hasnat
Government College of Engineering and Textile Technology, Berhampore, West Bengal
56 PUBLICATIONS 281 CITATIONS
SEE PROFILE

Rupam Nag
Government College of Engineering and Textile Technology, Berhampore
2 PUBLICATIONS 0 CITATIONS
SEE PROFILE

AN INTRODUCTION TO AUTOENCODERS

Dibyendu Barman, Abul Hasnat, Rupam Nag

Department of Computer Science and Engineering,
Government College of Engineering and Textile Technology, Berhampore, West Bengal, India
Email: dibyendu.barman@gmail.com, email.abulhasnat@gmail.com, nagrupam03@gmail.com

Abstract: The neural network, which resembles the neuronal architecture of the human brain, is at the heart of deep learning. Neural networks excel in identifying detailed patterns and representations in large datasets, allowing them to make predictions, categorize data, and produce new insights. Auto-Encoders emerge as an intriguing subclass of neural networks, providing a distinct approach to unsupervised learning. Auto-Encoders are a versatile and robust family of designs for the dynamic area of deep learning, where neural networks continually evolve to discover complex patterns and representation. These unsupervised learning models have gained a lot of attention for their capacity to develop appropriate data representations and are beneficial in a range of fields, including image processing and anomaly detection.

Keywords: Autoencoder, Encode, Decode, Dimentionality Reduction, Neural Network.

I. INTRODUCTION

An Autoencoder is a type of artificial neural network used to learn data encodings in an unsupervised manner. The aim of an autoencoder is to learn a lower-dimensional representation (encoding) of a higher-dimensional data, typically for dimensionality reduction, by training the network to capture the most important parts of the input image. Autoencoders are a specialized class of algorithms that can learn efficient representations of input data with no need for labels. It is a class of artificial neural networks designed for unsupervised learning [1-2]. Learning to compress and effectively represent input data without specific labels is the essential principle of an automatic decoder. This is accomplished using a two-fold structure that consists of an encoder and a decoder. The encoder transforms the input data into a reduced-dimensional representation, which is often referred to as “latent space” or “encoding”. From that representation, a decoder rebuilds the initial input. For the network to gain meaningful patterns in data, a process of encoding and decoding facilitates the definition of essential features [3-4].

This article is organized as follows: Section II discusses in details of the algorithms. Section III shows the comparative study between PCA and Auto-Encoders. Section IV shows the types of Auto-Encoders. Lastly Section V shows the applications of aspects analyzed in this paper.

II. AUTOENCODERS

The general architecture of an Auto-Encoder includes an Encoder, Decoder, and Bottleneck layer.

A. Encoder: The input layer receives raw input data. Hidden layers gradually lower the dimensionality of the input, capturing significant characteristics and patterns. These layers constitute the encoder. The final hidden layer, the bottleneck layer (latent space), has a drastically decreased dimensionality. This layer represents the compressed encoding of the incoming data [5].

B. Bottleneck: It is a module that stores the compressed knowledge representations and it is the most significant component of the network [5].

C. Decoder: The bottleneck layer converts the encoded representation back to the original input's dimensions. The hidden layers gradually increase their dimensionality with the goal of reconstructing the original input. The output layer generates the reconstructed output, which ideally should be as near to the input data as feasible [6].

During training, the loss function is commonly a reconstruction loss, which measures the difference between the input and the reconstructed outputs. For continuous data, the Mean Squared Error (MSE) is a popular choice; the binary cross-entropy is used for binary data. During training, the Autoencoder learns to minimize reconstruction loss, causing the network to capture the most relevant aspects of the input data in the bottleneck layer [4].

Once the training process is complete, the encoder portion of the Auto-Encoder is kept to encode data of same type that is used for training. The methods to limit the network are-

a) Keep hidden Layers tiny: If each hidden layer is maintained as tiny as possible, the network will be compelled to take up just the most representative aspects of the input, resulting in data encoding [7].

b) Regularization: In this approach, a loss factor is introduced to the cost function, encouraging the network to train in ways other than just duplicating the input [7].

c) Denoising: Another method of restricting the network is to introduce noise into the input and train the network how to remove it from the data [8].

d) Tuning the activation functions: This strategy entails modifying the activation functions of various nodes such that the bulk of them remain inactive, hence lowering the size of the hidden layers.

III. AUTOENCODER VS PRINCIPAL COMPONENT ANALYSIS (PCA)

Although PCA is basically a linear transformation, auto-encoders may represent complex non-linear processes. Because PCA features are projections onto an orthogonal basis, they are totally linearly independent. However, because auto encoded features are solely trained for accurate reconstruction, they may exhibit correlations. PCA is faster and cheaper to compute than Autoencoders. PCA behaves similarly to a single layered Autoencoder with a linear activation function. Because of the enormous number of parameters, the Autoencoder is susceptible to overfitting. However, regularization and proper planning may be helpful to avoid it. Comparative study between AE and PCA is shown in Table I.

TABLE I: Comparative study between Auto-Encode and PCA

SL	Auto-Encoder (AE)	Principal Component Analysis (PCA)
1	It is non-linear mapping. It is prone to over-fitting.	It is linear mapping. PCA is faster and cheaper than Auto-Encoder.
2	It is generalization.	It is only about linear mapping.
3	AE can extract non-linear structure in the input data.	It extracts linear structure.
4	It is more powerful than PCA.	It is less powerful.
5	AE gives much better representation in reduced dimension of data.	PCA gives linear representation.

IV. DIFFERENT TYPES OF AUTOENCODER

An Autoencoder is a form of Artificial Neural Network that learns effective data coding without supervision.

Generally Auto-encoders are of the following types-

1) De-noising Auto-Encoder
2) Sparse Auto-Encoder
3) Deep Auto-Encoder
4) Contractive Auto-Encoder
5) Under complete Auto-Encoder
6) Convolutional Auto-Eencoder
7) Variational Auto-Eencoder

IV.A Denoising Autoencoder

The Denoising Autoencoder trains on a partially corrupted input to recover the original, undistorted picture. As previously stated, this strategy is an excellent way to prevent the network from merely replicating the input and instead learning the underlying structure and significant properties of the data. Figure 1 illustrates the denoising/reconstruction process [8].

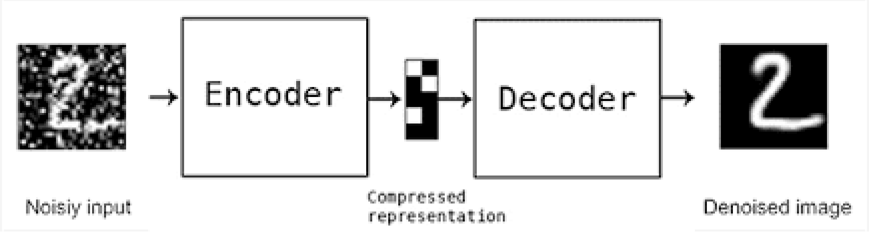

Figure 1. Denoising process using Denoising Autoencoder

The pros and cons of the Denoising Autoencoder are discussed below.

Advantages:

i) This form of Autoencoder may extract critical characteristics while removing noise or unnecessary features.

ii) De-noising Autoencoders may be used for data augmentation, with recovered pictures serving as enhanced data, yielding extra training examples.

Disadvantages:

i) Selecting the appropriate type and degree of noise to add may be difficult and may need specialized expertise.

ii) Denoising can cause the loss of some important information from the original input.

IV.B Sparse Autoencoder

This type of Autoencoder often has more hidden units than the input, but only a few can be active at the same time. This attribute is referred to as the network's sparsity. The network's sparsity can be modified by manually zeroing the requisite hidden units, tweaking the activation functions, or introducing a loss element into the cost function [9]. Basic structure of Sparse Autoencoder is shown in figure 2.

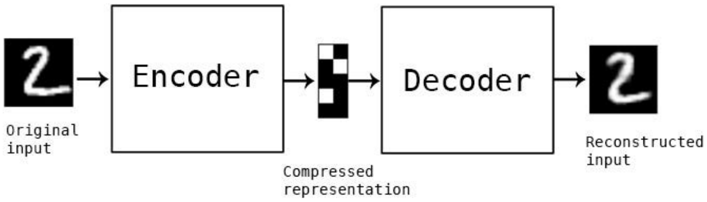

Figure 2. Basic structure of Sparse Autoencoder

Merits and demerits of Sparse Autoencoders are discussed below.

Advantages:

i) The sparsity requirement in sparse Autoencoders aids in filtering out noise and extraneous features during the encoding process.

ii) Because of their focus on sparse activations, these Autoencoders frequently learn significant and meaningful features.

Disadvantages:

i) The choice of hyper-parameters has a considerable impact on the performance of the sparse Autoencoder. Distinct inputs should activate distinct nodes of the network.

IV.C Deep Autoencoder

Deep Auto-encoders are made up of two identical deep belief networks, one for encoding and the other for decoding. Deep Autoencoders typically feature 4 to 5 encoding layers, followed by 4 to 5 decoding levels. For this model, unsupervised layer-by-layer pre-training is done. The layers are Restricted Boltzmann Machines (RBM), which are the foundation of Deep Belief Networks (DBN). A deep Autoencoder would apply binary changes after each RBM when processing the MNIST benchmark dataset. Deep Autoencoders are useful for topic modelling, which is the statistical modelling of abstract subjects scattered over a collection of texts [9]. They can also compress photos into 30 number vectors. Basic structure of Deep Autoencoder is shown in figure 3.

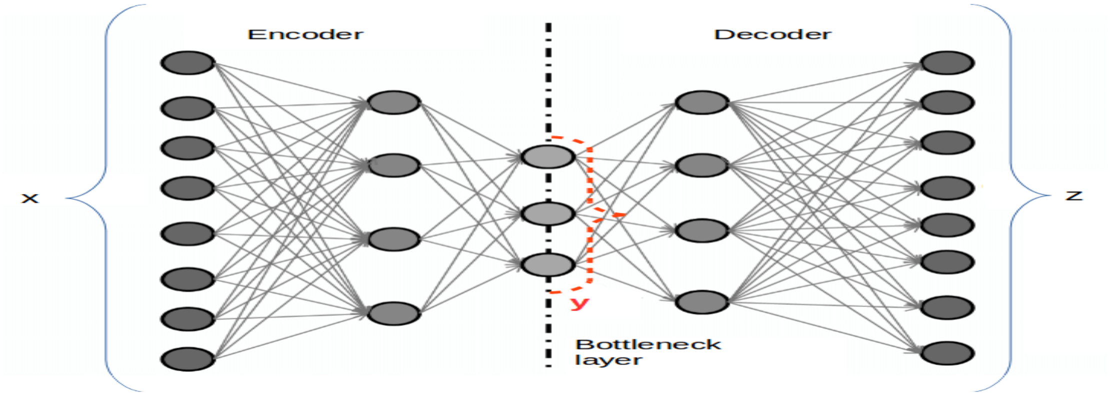

Figure 3. Basic structure of Deep Autoencoder

Merits and demerits of Deep Autoencoders are discussed below.

Advantages:

i) Deep Autoencoders may be utilized for different types of datasets containing real-valued data, such as those using Gaussian corrected RBMs.

ii) The final encoding layer is small and quick.

Disadvantages:

i) There are more parameters than input data, increasing the likelihood of overfitting.

ii) Training the data may be a complication since during the stage of the decoder's backpropagation, the learning rate should be decreased or made slower depending on whether binary or continuous data is being handled.

IV.D Contractive Autoencoder

A Contractive Autoencoder's goal is to provide a robust learnt representation that is less susceptible to tiny variations in the input. By adding a penalty term to the loss function, the data representation becomes more robust. Contractive Autoencoders, like sparse and denoising Autoencoders, are forms of regularization. This regularize, however, corresponds to the Frobenius norm of the encoder activations' Jacobian matrix with regard to the input. The Frobenius norm of the Jacobian matrix for the hidden layer is computed with regard to input and is essentially the sum of all components squared [10]. Basic structure of Contractive Autoencoder is shown in figure 4.

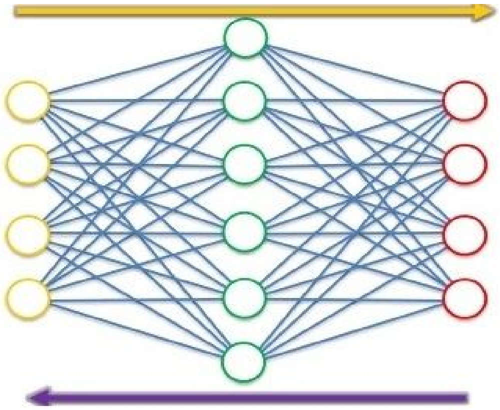

Figure 4. Basic structure of Contractive Autoencoder

Advantages:

To learn meaningful feature extraction, employ a Contractive Autoencoder rather than a Denoising Autoencoder. This model learns an encoding for comparable inputs. As a result, we are driving the model to learn how to condense a neighborhood of inputs into a smaller neighborhood of outputs.

IV.E Under-complete Autoencoder

The purpose of an Under-complete Autoencoder is to capture the most essential elements in the data. Under complete Autoencoders have a reduced hidden layer dimension relative to the input layer. This aids in extracting key aspects from the data. It reduces the loss function by punishing g(f(x)) for being different from the input x. Basic structure of Under-complete Autoencoder is shown in figure 5.

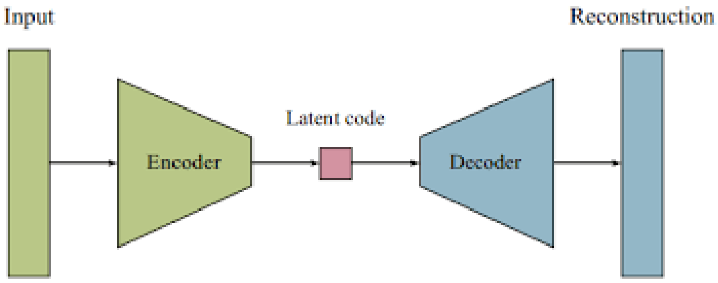

Figure 5. Basic structure of Undercomplete Autoencoder

The pros and cons of Under-complete Autoencoder are discussed below.

Advantages:

Under-complete Autoencoders require no regularization since they optimize data probability rather than replicating input to output.

Disadvantage:

Overfitting can result from in an over parameterized model due to insufficient training data.

IV.F Convolutional Autoencoder

Convolutional Autoencoders employ Convolutional Neural Networks (CNNs) as building blocks. The encoder comprises of many layers that take an image or a grid as input and send it through various convolution layers, resulting in a compressed representation of the input. The decoder is the encoder's mirror image; it de-convolves the compressed representation and attempts to recover the original picture [12]. Basic structure of Convolutional Autoencoder is shown in figure 6.

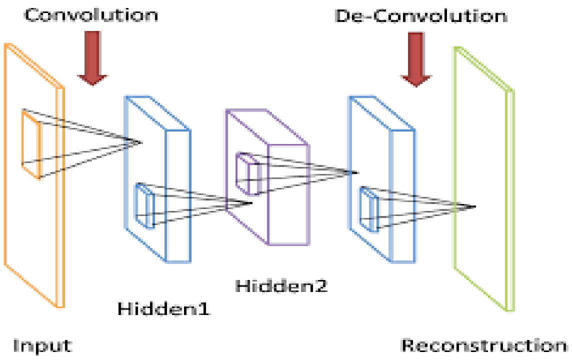

Figure 6. Basic structure of Convolution Autoencoder

Merits and demerits of Convolutional Autoencoder are discussed below.

Advantages:

i) The high-dimensional image data may be reduced to lower-dimensional data using Convolutional Autoencoders. This enhances the efficiency of image storage and transmission.

ii) A Convolutional Autoencoder can recreate missing pieces of a picture. It can also handle photos with modest changes in object location and orientation.

Disadvantages:

i) This type of Autoencoder tends to overfit. Regularization techniques should be utilized to address this issue.

ii) Data compression can cause data loss, resulting in a low-quality output (i.e. reconstructed image).

IV.G Variational Autoencoder

The Variational Autoencoder makes assumptions about the distribution of latent variables. It is trained using the Stochastic Gradient Variational Bayes estimator [11]. Figure 7 illustrates the basic structure of Variational Autoencoder.

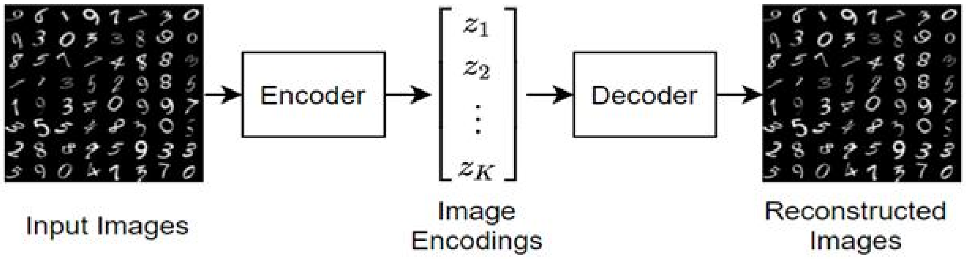

Figure 7. Basic structure of Variational Autoencoder

The pros and cons of Variational Autoencoder are discussed below.

Advantages:

i) Variational Autoencoder is used to create new data points that closely mirror the original training data. These samples are acquired from the latent space.

ii) Variational Autoencoder is a probabilistic framework that learns a compressed representation of data. It captures the underlying structure and variability. It is helpful for anomaly detection and data exploration.

Disadvantages:

i) Variational the Autoencoder uses approximations to estimate the real distribution of the latent variables.

ii) The approximation has some limitations that might impair the quality of the produced samples. The produced samples may represent only a portion of the real data distribution. Thus it may lead to a lack of variety in produced samples.

V. APPLICATIONS OF AUTOENCODER

A few applications of AE may be summarized as below-

1) Autoencoders have the ability to reduce the dimensionality of datasets with high dimensions. The essential features can be captured by an AE and it provides a lower-dimensional data representation [1][2].

2) Autoencoders may be used to create new data points that is similar to the training set. This can be achieved by using the decoder to generate fresh data after sampling from the compressed representation.

3) Autoencoders can be used to minimize noise in data. For this purpose, we may use an Autoencoder to extract the original data from a noisy version of the data[6].

4) Compression of images and audio: Autoencoders are capable of compressing large images or audio files while preserving the majority of the essential information in the original data [8].

5) Autoencoders can be used to find abnormalities or outliers in datasets.

VI. DEMERITS OF AUTOENCODER

Drawbacks with Autoencoders may be listed as below-

i) The computational cost of Autoencoders (the cost of computer vision) is a major concern, especially when working with huge datasets and intricate models.

ii) Autoencoders is vulnerable to overfitting, a phenomenon in which the model picks up noise or other artifacts from the training dataset that do not translate well to fresh data.

VII. CONCLUSION

Strong neural network models called Autoencoders are useful for learning data coding's unsupervised. They are helpful for different kinds of jobs, including as a) anomaly detection, b) feature learning, and c) dimensionality reduction. Autoencoders are still a valuable tool in the toolbox of machine learning practitioners, despite their shortcomings, particularly when dealing with complicated data formats like text and images.

References

[1] Rami Al-Hmouz, Witold Pedrycz, Medhat Awadallah, Ahmed Al-Hmouz, "Fuzzy relational representation, modeling and interpretation of temporal data", Knowledge-Based Systems, vol.244, pp.108548, 2022.

[2] Haoxuan Zhou, Xin Huang, Guangrui Wen , Zihao Lei, Shuzhi Dong, Ping Zhang and Xuefeng Chen , "Construction of health indicators for condition monitoring of rotating machinery: A review of the research", Expert Syst. Appl, vol. 203, pp. 117297, 2022.

[3] Hmrishav Bandyopadhyay, Autoencoders in Deep Learning: Tutorial & Use Cases [2022], 2022.

[4] Dor Bank, Noam Koenigstein and Raja Giryes, Autoencoders, pp. 1-22, 2021.

[5] Håkon Hukkelås, Rudolf Mester and Frank Lindseth, "Deepprivacy: A generative adversarial network for face anonymization", 2019.

[6] Junhai Zhai, Sufang Zhang, Junfen Chen and Qiang He, "Autoencoder and Its Various Variants", IEEE International Conference on Systems Man and Cybernetics (SMC), pp. 415-419, 2018.

[7] Cheng-Yu Chen, Jenq-Shiou Leu and Setya Widyawan Prakosa, Using Autoencoder to Facilitate Information Retention for Data Dimension Reduction, IEEE, pp. 1-5, 2018.

[8] Bhawna Goyal, Sunil Agrawal and BS Sohi, "Noise Issues Prevailing in Various Types of Medical Images", Biomedical Pharmacology Journal, vol. 11, pp. 1227, 2018.

[9] Jiawei Chen, Janusz Konrad and Prakash Ishwar, "Vgan-based image representation learning for privacy-preserving facial expression recognition", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2018.

[10] N. Raval, A. Machanavajjhala and L. P. Cox, "Protecting visual secrets using adversarial nets", 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1329-1332, 2017.

[11] Rami Al-Hmouz, Witold Pedrycz, Medhat Awadallah, Ahmed Al-Hmouz, "Fuzzy relational representation, modeling and interpretation of temporal data", Knowledge-Based Systems, vol.244, pp.108548, 2022.

[12] Haoxuan Zhou, Xin Huang, Guangrui Wen , Zihao Lei, Shuzhi Dong, Ping Zhang and Xuefeng Chen , "Construction of health indicators for condition monitoring of rotating machinery: A review of the research", Expert Syst. Appl, vol. 203, pp. 117297, 2022