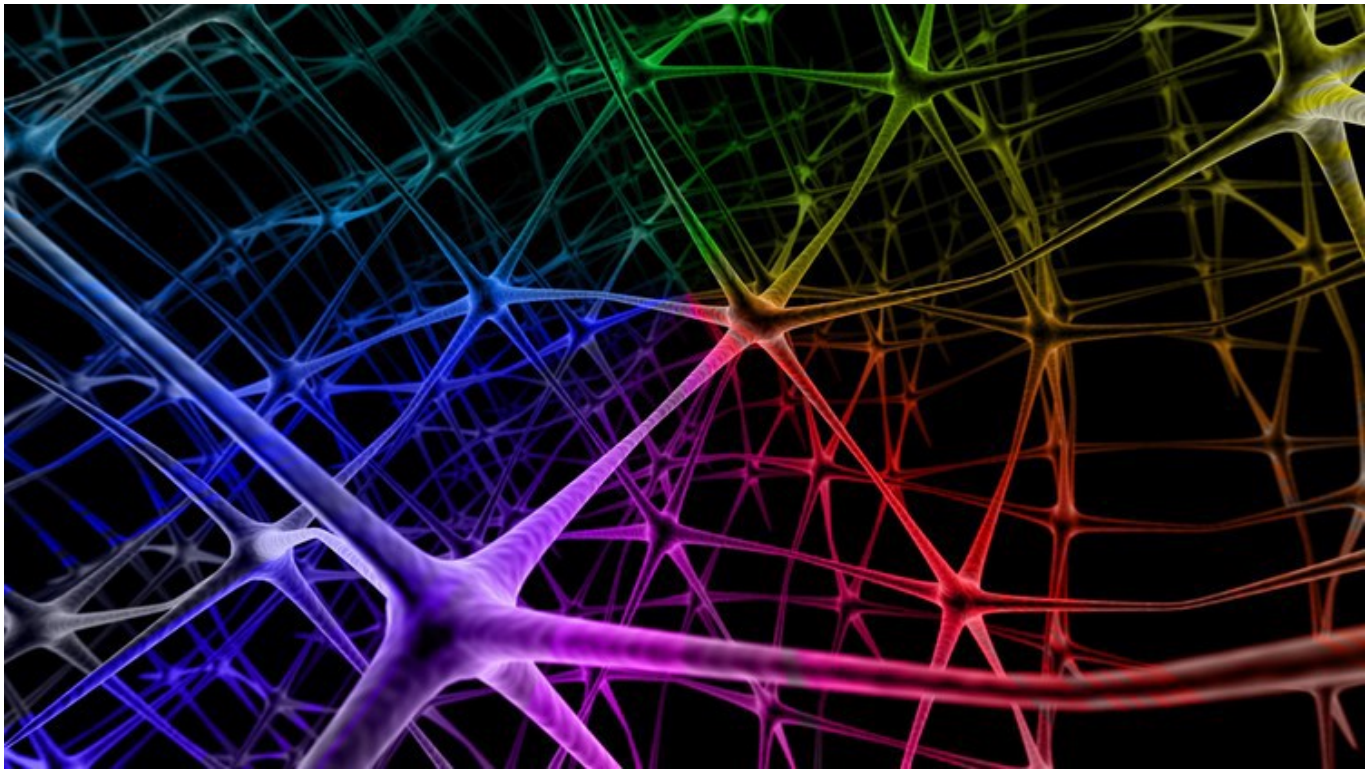
Autoencoders — Bits and Bytes of Deep Learning

Vindula Jayawardana Follow
Aug 4, 2017 · 5 min read



One way to think of what deep learning does is as “A to B mappings,” says Andrew Ng, chief scientist at Baidu Research. “You can input an audio clip and output the transcript. That’s speech recognition.” As long as you have data to train the software, the possibilities are endless, he maintains. “You can input email, and the output could be: Is this spam or not?” Input loan applications, he says, and the output might be the likelihood a customer will repay it. Input usage patterns on a fleet of cars and the output could advise where to send a car next.

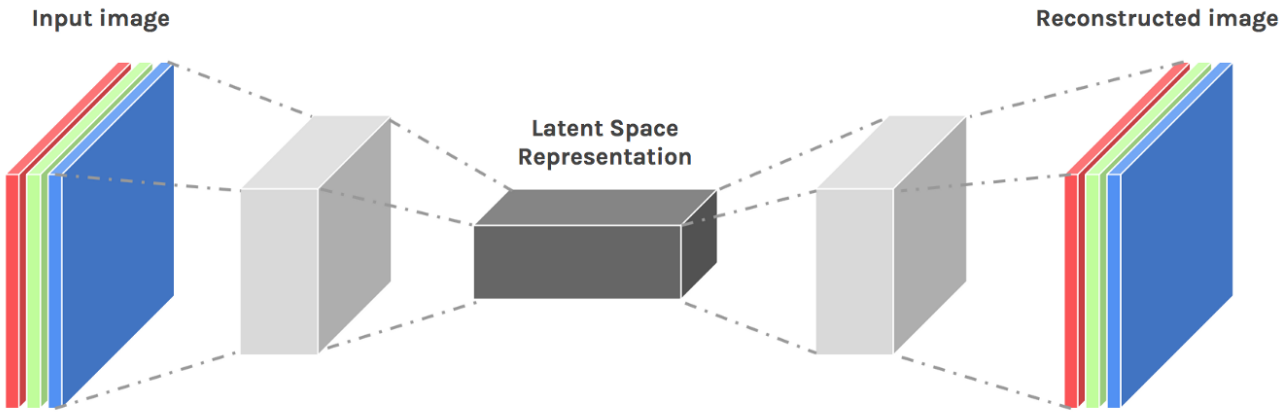
Rather making the facts complicated by having complex definitions, think of deep learning as a subset of a subset. Artificial Intelligence encircles a wide range of

technologies and techniques that enable computers systems to unravel problems in ways that at least superficially resemble thinking. Within that sphere, there is that whole toolbox of enigmatic but important mathematical techniques which drives the motive of learning by experience. That subset is known to be machine learning. Finally, within machine learning is the smaller subcategory called deep learning (also known as deep structured learning or hierarchical learning)which is the application of artificial neural networks (ANNs) to learning tasks that contain more than one hidden layer.

. . .

What's an Autoencoder?

Despite its somewhat initially-sounding cryptic name, autoencoders are a fairly basic machine learning model. Autoencoders (AE) are a family of neural networks for which the input is the same as the output. They work by compressing the input into a latent-space representation and then reconstructing the output from this representation.



Autoencoder architecture

In more terms, autoencoding is a data compression algorithm where the compression and decompression functions are,

1. **Data-specific:** Autoencoders are only able to compress data similar to what they have been trained on. An autoencoder which has been trained on human faces would not be performing well with images of modern buildings. This improvises the difference between autoencoders and MP3 kind of compression algorithms which

only hold assumptions about sound in general, but not about specific types of sounds.

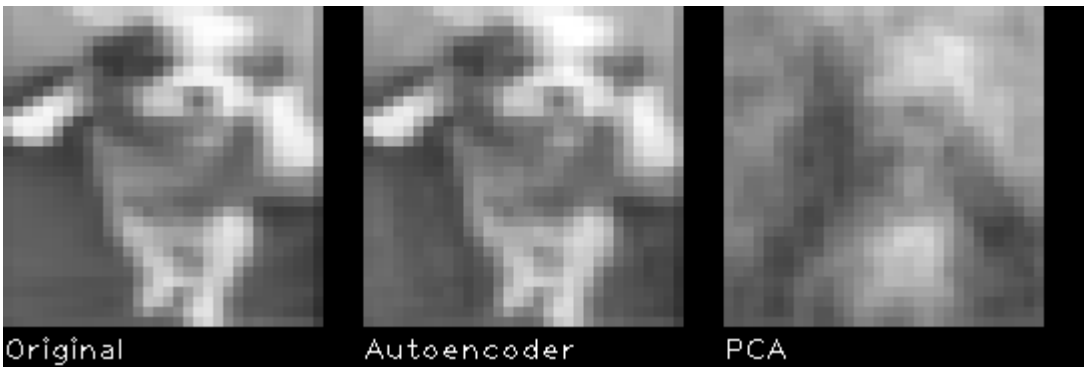
2. **Lossy:** This means that the decompressed outputs will be degraded compared to the original inputs. Just like what you see in JPEG or MP3.

3. **Learned automatically from examples:** If you have appropriate training data, it is easy to train specialized instances of the algorithm that will perform well on a specific type of input. It doesn't require any new engineering.

Additionally, in almost all contexts where the term "autoencoder" is used, the compression and decompression functions are implemented with neural networks.

Why Autoencoders?

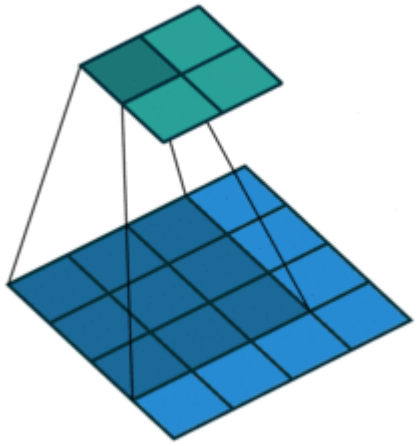
Despite the fact, the practical applications of autoencoders were pretty rare some time back, today **data denoising** and **dimensionality reduction for data visualization** are considered as two main interesting practical applications of autoencoders. With appropriate dimensionality and sparsity constraints, autoencoders can learn data projections that are more interesting than PCA or other basic techniques.



Performance comparison of Autoencoders and PCA

Convolutional Autoencoders

In the traditional architecture of autoencoders, it is not taken into account the fact that a signal can be seen as a sum of other signals. Convolutional Autoencoders (CAE), on the other way, use the convolution operator to accommodate this observation. Convolution operator allows filtering an input signal in order to extract some part of its content. They learn to encode the input in a set of simple signals and then try to reconstruct the input from them.



A convolution between a 4x4x1 input and a 3x3x1 convolutional filter.
The result is a 2x2x1 activation map. (Source)

Refer this for the use cases of convolution autoencoders with pretty good explanations using examples. We will see a practical example of CAE later in this post.

Building Autoencoders

We will start with the most simple autoencoder that we can build. In the latter part, we will be looking into more complex use cases of the autoencoders in real examples.

Following is the code for a simple autoencoder using keras as the platform. keras provided MNIST digits are used in the example.

With this code snippet, we will get the following output.

In the above image, the top row is the original digits, and the bottom row is the reconstructed digits. As you can see, we have lost some important details in this basic

example.

Since our inputs are images, it makes sense to use convolutional neural networks as encoders and decoders. In practical settings, autoencoders applied to images are always convolutional autoencoders as they simply perform much better.

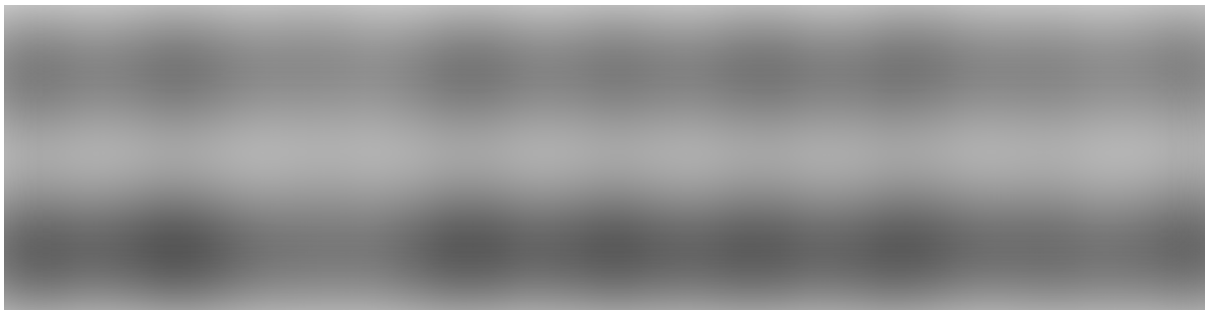
With the convolution autoencoder, we will get the following input and reconstructed output.

Autoencoder Practical Application: Image Denoising

In this section, we will be looking into the use of autoencoders in its real-world usage, for image denoising. We will train the convolution autoencoder to map noisy digits images to clean digits images.

We will generate synthetic noisy digits by applying a Gaussian noise matrix and clip the images between 0 and 1.

When autoencoder is trained, we can use it to remove the noises added to images we have never seen! And here is how the input and reconstructed output will look like.



Conclusion

An autoencoder is an artificial neural network used for unsupervised learning of efficient codings. In the modern era, autoencoders have become an emerging field of research in numerous aspects such as in anomaly detection. In this post, it was expected to provide a basic understanding of the aspects of what, why and how of autoencoders.

References

1. https://hackernoon.com/autoencoders-deep-learning-bits-1-11731e200694
2. https://blog.keras.io/building-autoencoders-in-keras.html
3. https://www.technologyreview.com/s/513696/deep-learning/

Machine Learning Deep Learning Autoencoder Artificial Intelligence Neural Networks