Autoencoders

ML Instruction Team, Fall 2022

CE Department
Sharif University of Technology

Mahsa Yazdani
Arian Amani

The Power Of Unsupervised Learning

- Huge datasets compared to supervised learning (No need for labeling)
- Can find previously unknown patterns in data that are impossible with supervised learning

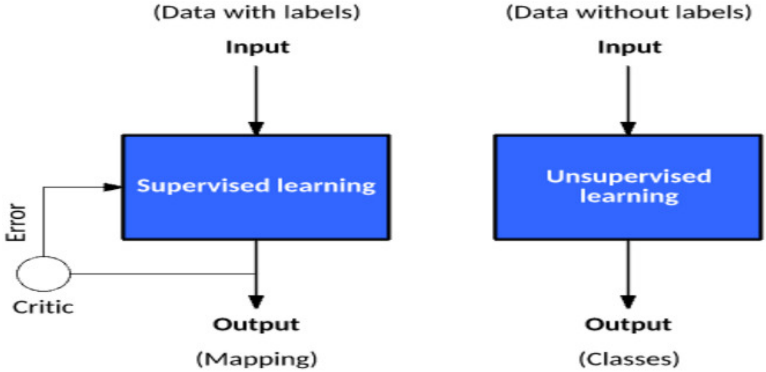

Figure: Types of Machine learning: Deep learning (supervised and unsupervised learning)(Jones [2017]), Source

Applications of Autoencoders

- Compression:
  - AEs can compress our input into a lower dimensional vector and try to reconstruct the original input from that vector

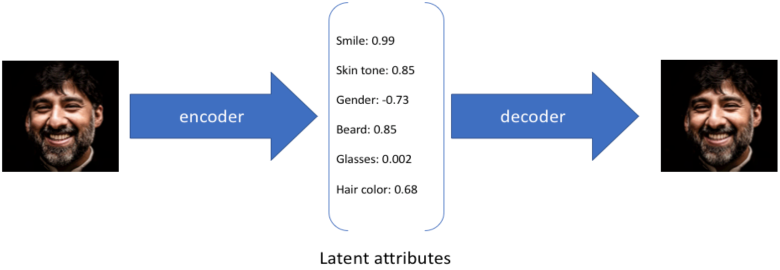

Figure: Source

Applications of Autoencoders

- Dimensionality Reduction:
  - AEs can perform dimensionality reduction

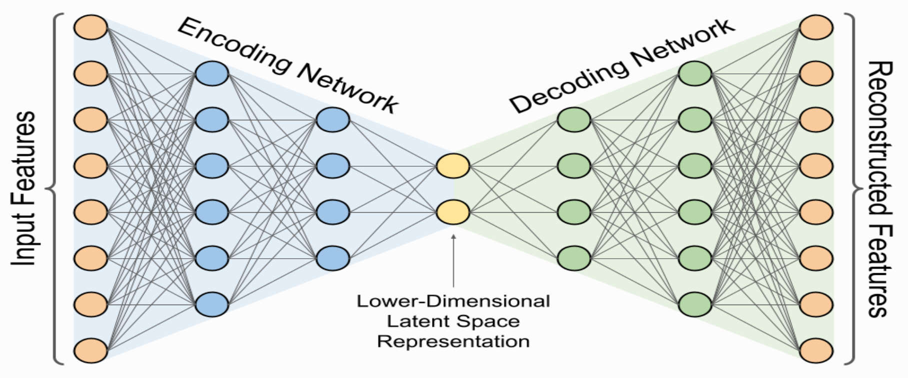

Figure: Source

Applications of Autoencoders

- Image coloring and noise reduction

Figure: Source

Applications of Autoencoders

- Super-Resolution

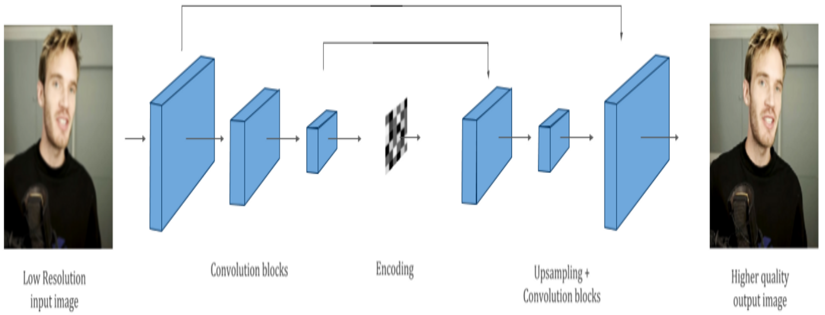

Figure: Source

What is an Autoencoder?

- An autoencoder is a type of artificial neural network, capable of learning a low dimensional representation of the input data (codings), without supervision (unlabeled training data - unsupervised learning)
- Autoencoders take an input X and try to predict X. We use a bottleneck layer with a smaller dimension compared to the input, to use as the coding.

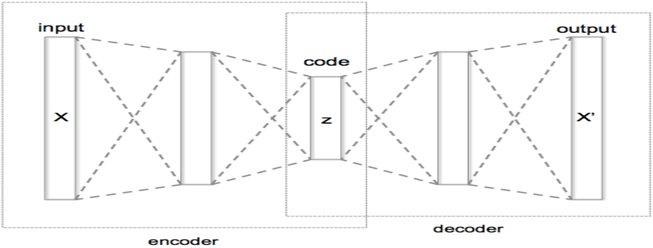

Figure: Schematic structure of an autoencoder with 3 fully connected hidden layers. The code (z) is the most internal layer, Source

Autoencoders: Structure

- Encoder: compress input into a latent-space of usually smaller dimension. $h = f(x)$
- Decoder: reconstruct input from the latent space. $r = g(f(x))$ with r as close to x as possible

Autoencoders: Applications

- Autoencoders can
  - act as feature detectors
  - be used for unsupervised pretraining of deep neural networks

Autoencoders: Applications

- Autoencoders can be used as generative models. (will be discussed more later in this chapter.)

Autoencoders: Applications

- Watermark removal

Autoencoders: Applications

- Noise reduction

Stacked Autoencoders

- Autoencoders can have multiple hidden layers. In this case they are called *stacked autoencoders* (or *deep autoencoders*).
- Adding more layers helps the autoencoder learn more complex codings. but be careful about overfitting!

Autoencoders & Images

Are normal Autoencoders suitable for working with images?

Autoencoders & Images

- Convolutional Autoencoders

Denoising Autoencoders

- Another way to force the autoencoder to learn useful features is to add noise to its inputs.
- Denoising autoencoders train to minimize the loss between x and $g(f(x+w))$, where w is random noise.
- Denoising autoencoders, with Gaussian noise (left) or dropout (right):

Denoising Autoencoders

- A few noisy images (with half the pixels turned off), and the images reconstructed by the dropout-based denoising autoencoder. Notice how the autoencoder guesses details that are actually not in the input, such as the top of the white shirt (bottom row, fourth image).

Denoising Autoencoders

- Intuitively, a denoising autoencoder learns a projection from a neighborhood of our training data back onto the training data.

Autoencoder Generative Models

How can we generate NEW data with Autoencoders??

hint: Autoencoder learns the feature space!

Walking through an example

- We want to reconstruct some shapes.

Walking through an example

- Not all of the points in latent space have meaningful reconstructions.

Walking through an example

- What we want is something like the following picture. So that with sampling from the latent space, we can generate new shapes.

Variational Autoencoders

- instead of directly producing a coding for a given input, the encoder produces a mean coding μ and a standard deviation σ. The actual coding is then sampled randomly from a Gaussian distribution with mean μ and standard deviation σ

Variational Autoencoders

Test

- One
  - One
  - Two
  - Three
- For two-dimensional tensors, we have a corresponding sum with indices (a, b) for f and $(i - a, j - b)$ for g, respectively:

$$(f * g)(i, j) = \sum_a \sum_b f(a, b) g(i - a, j - b)$$

- It is given by,

$$w_{t+1} = w_t - \left(\alpha_t / \sqrt{(v_t}\right) + e\Big) * (\delta L / \delta w_t)$$

where,

$$v_t = \beta * v_t + (1 - \beta) * (\delta L / \delta w_t)^2$$

Image References

- https://lilianweng.github.io/posts/2018-08-12-vae/
- https://emkademy.medium.com/1-first-step-to-generative-deep-learning-with-autoencoders-22bd41e56d18
- Aurelien Geron. 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd. ed.). O'Reilly Media, Inc.
- https://github.com/wojciechmo/vae
- https://ai.googleblog.com/2017/08/making-visible-watermarks-more-effective.html
- https://medium.com/@harishr2301/denoising-autoencoders-996e866e5cd0
- https://subscription.packtpub.com/book/big-data-and-business-intelligence/9781787121089/4/ch04lvl1sec51/setting-up-stacked-autoencoders
- https://www.analyticsvidhya.com/blog/2021/01/auto-encoders-for-computer-vision-an-endless-world-of-possibilities/
- https://towardsdatascience.com/convolutional-autoencoders-for-image-noise-reduction-32fce9fc1763
- https://ift6266h17.files.wordpress.com/2017/03/14_autoencoders.pdf
- https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

References

- Aurelien Geron. 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd. ed.). O'Reilly Media, Inc.
- https://www.cs.toronto.edu/ rgrosse/courses/csc321_2017/slides/lec20.pdf
- https://www.math.purdue.edu/ buzzard/MA598-Spring2019/Lectures/Lec16%20-%20Autoencoders.pptx
- http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture11.pdf

Thank You!

Any Question?