Autoencoder

Deep Learning model

By: Theepana Govintharajah

Autoencoder

- Autoencoders are a type of neural network architecture used in unsupervised learning tasks that learns to compress data and then reconstruct it from its compressed form.

- Autoencoders are only able to meaningfully compress data similar to what they have been trained on. Since they learn features specific for the given training data, we can’t expect an autoencoder trained on handwritten digits to compress landscape photos.

By: Theepana Govintharajah

Autoencoder Architecture

An Autoencoder consists of three layers:

- Encoder
- Latent Space/ Bottleneck/ Code
- Decoder

Encoder

- Responsible for compressing the input data into a lower-dimensional representation.

- It consists of one or more layers of neurons that transform the input data into a compressed or encoded representation. The encoder's output is often referred to as the "encoding" or "latent space."

By: Theepana Govintharajah

Autoencoder Architecture

Latent Space:

- lower-dimensional space is often called the "latent space" or "feature space." The goal is to capture the essential features of the input data in this space.

Decoder:

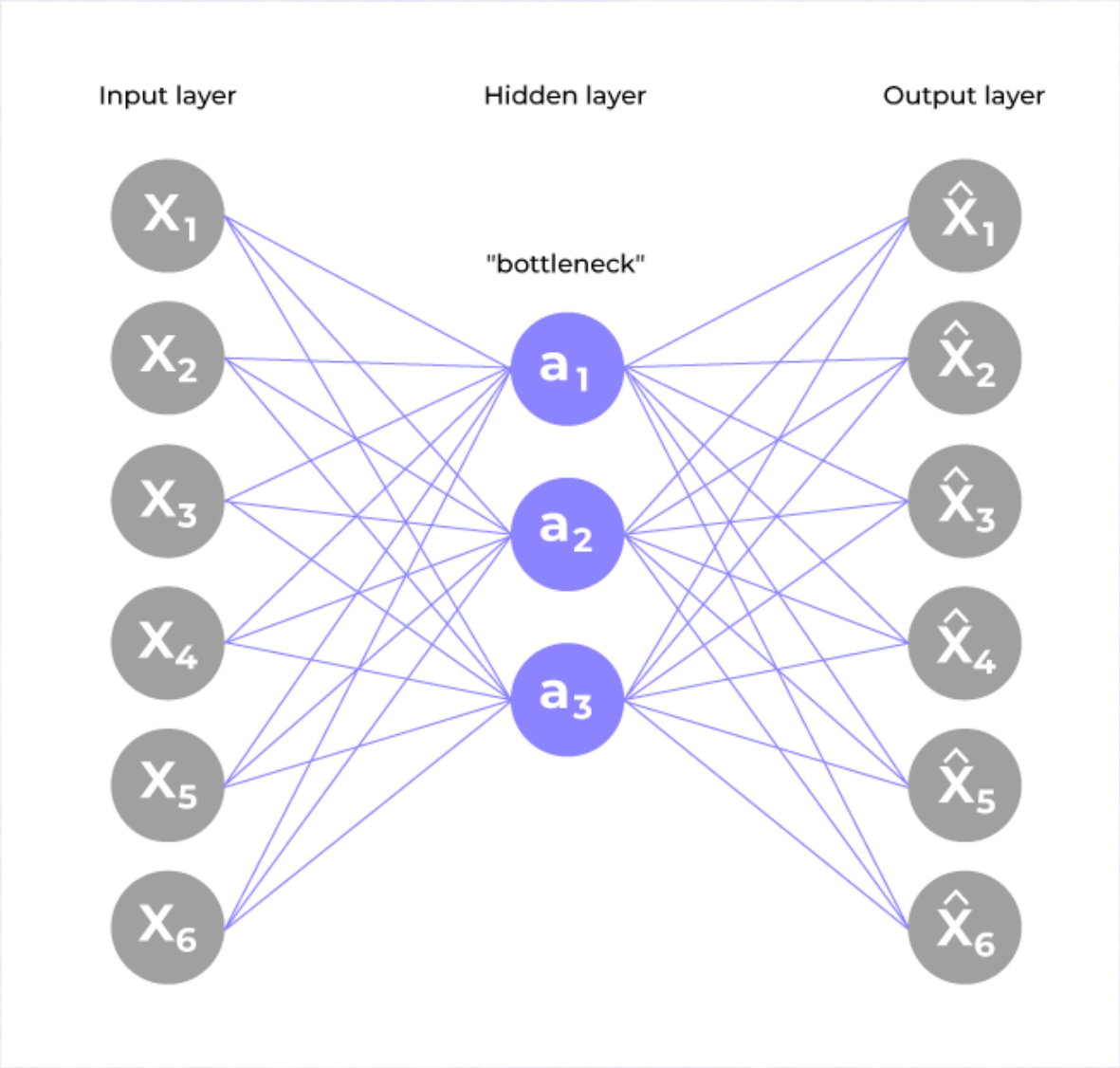
- Tasked with reconstructing the input data from the compressed representation generated by the encoder.

- Like the encoder, the decoder consists of one or more layers of neurons.

- It takes the encoded representation and transforms it back into the original input data.

By: Theepana Govintharajah

Autoencoder



By: Theepana Govintharajah

Autoencoder

- During training, the autoencoder is given examples of input data. It tries to encode the data into the latent space and then decode it back into the original form.

- The difference between the original data and the reconstructed data is called the reconstruction error.

- The loss function used during training is typically a reconstruction loss / reconstruction error.

- The autoencoder is trained to minimize this error, so it learns to encode the data in a way that can be accurately reconstructed forcing the network to capture the most important features of the input data in the bottleneck layer.

By: Theepana Govintharajah

Hyperparameters that you need to set

Code size:

- The code size or the size of the bottleneck is the most important hyperparameter used to tune the autoencoder.
- The bottleneck size decides how much the data has to be compressed.This can also act as a regularisation term.

Number of layers:

- Depth of the encoder and the decoder.

Number of nodes per layer:

Loss function:

- Common choices include mean squared error (MSE) for continuous data or binary cross-entropy for binary data.

By: Theepana Govintharajah

Types of Autoencoder

1. Undercomplete autoencoders
2. Sparse autoencoders
3. Denoising autoencoders
4. Variational Autoencoders (for generative modelling)

Undercomplete autoencoders:

- An undercomplete autoencoder is one of the simplest types of autoencoders.
- The way it works is very straightforward
- This form of compression in the data can be modeled as a form of **dimensionality reduction**.
- When we think of dimensionality reduction, we tend to think of methods like PCA (Principal Component Analysis), however PCA can only build linear relationships

By: Theepana Govintharajah

Types of Autoencoder

Sparse autoencoders:

- Sparse autoencoder is regulated by changing the number of nodes in bottleneck layer. Sparse autoencoders work by penalizing the activation of some neurons in hidden layers.

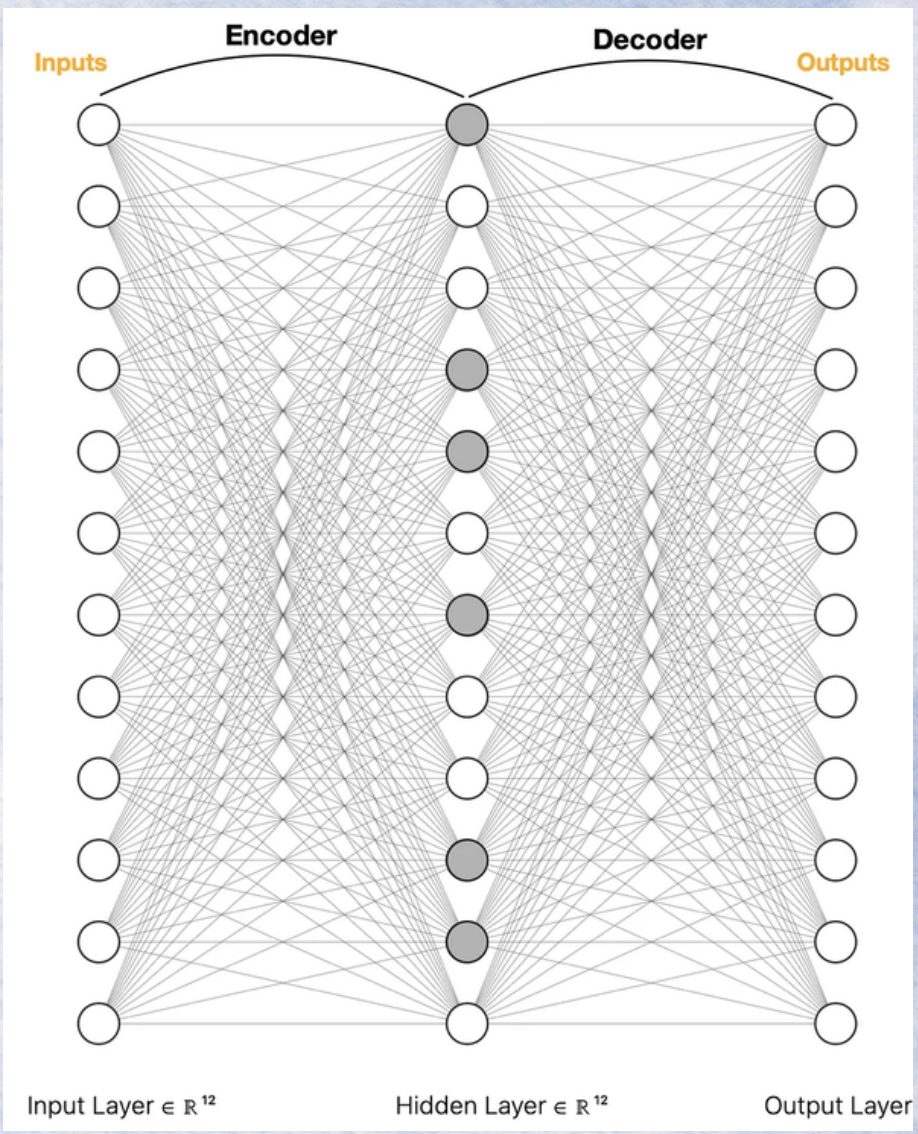
- As a means of regularizing the neural network, the sparsity function prevents more neurons from being activated.

- There are two types of regularizers used:

1. The **L1 Loss method** is a general regularizer we can use to add magnitude to the model.
2. The **KL-divergence method** considers the activations over a collection of samples at once rather than summing them as in the L1 Loss method. We constrain the average activation of each neuron over this collection.

By: Theepana Govintharajah

Sparse autoencoders



Legend

Active node (node output ≠ 0)

Inactive node (node output = 0)

Node activations vary depending on the inputs. Hence, it is not always same nodes that become inactive.

By: Theepana Govintharajah

Types of Autoencoder

Denoising autoencoders:

- Denoising autoencoders, as the name suggests, are autoencoders that remove noise from an image.

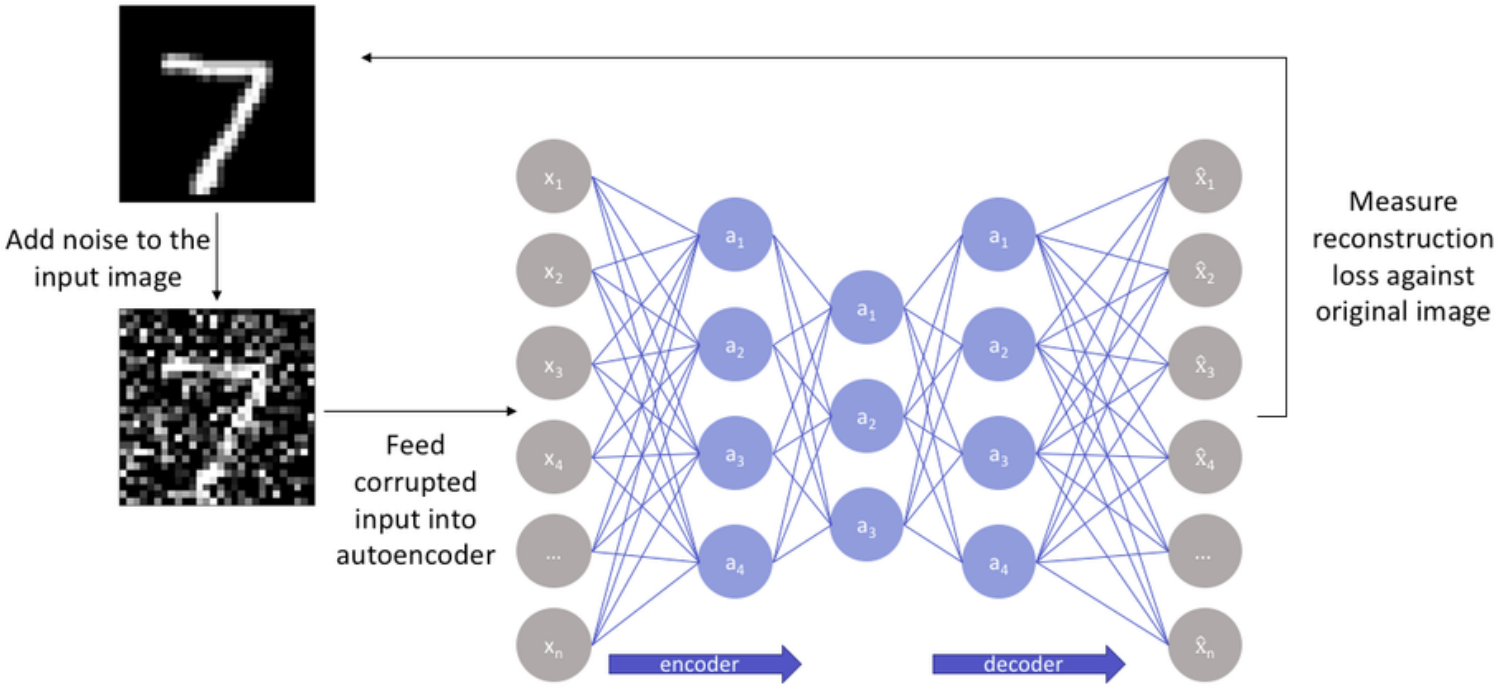
- They differ because they don't have the input image as their ground truth. Instead, they use a noisy version.

- The denoising autoencoder gets rid of noise by learning a representation of the input where the noise can be filtered out easily.

- The loss function generally used in these types of networks is L2 or L1 loss.

By: Theepana Govintharajah

Denoising autoencoders



Applications of Autoencoder

- Data denoising image and audio
- Dimensionality reduction
- Anomaly detection: autoencoders can identify data anomalies using a loss function that penalizes model complexity
- Image inpainting: autoencoders have been used to fill in gaps in images by learning how to reconstruct missing pixels based on surrounding pixels. For example, if you're trying to restore an old photograph that's missing part of its right side, the autoencoder could learn how to fill in the missing details based on what it knows about the rest of the photo.
- Variational Autoencoders can be used to generate both image and time series data.

By: Theepana Govintharajah

Follow me for more similar posts

By: Theepana Govintharajah