Lecture-11

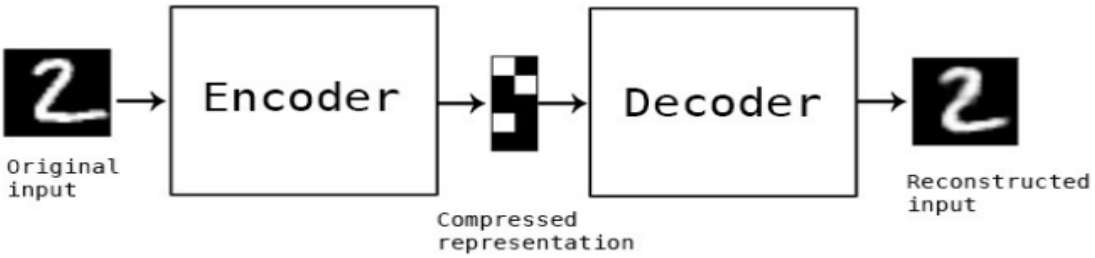
Name: Vaibhav Thakur

Roll No: 15MI412

Unsupervised learning is the training of an artificial intelligence (AI) algorithm using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance.
In unsupervised learning, an AI system is presented with unlabeled, uncategorised data and the system's algorithms act on the data without prior training. The output is dependent upon the coded algorithms. Subjecting a system to unsupervised learning is one way of testing AI.
Unsupervised learning algorithms can perform more complex processing tasks than supervised learning systems. However, unsupervised learning can be more unpredictable than the alternate model. While an unsupervised learning AI system might, for example, figure out on its own how to sort cats from dogs, it might also add unforeseen and undesired categories to deal with unusual breeds, creating clutter instead of order.

AUTO-ENCODER:

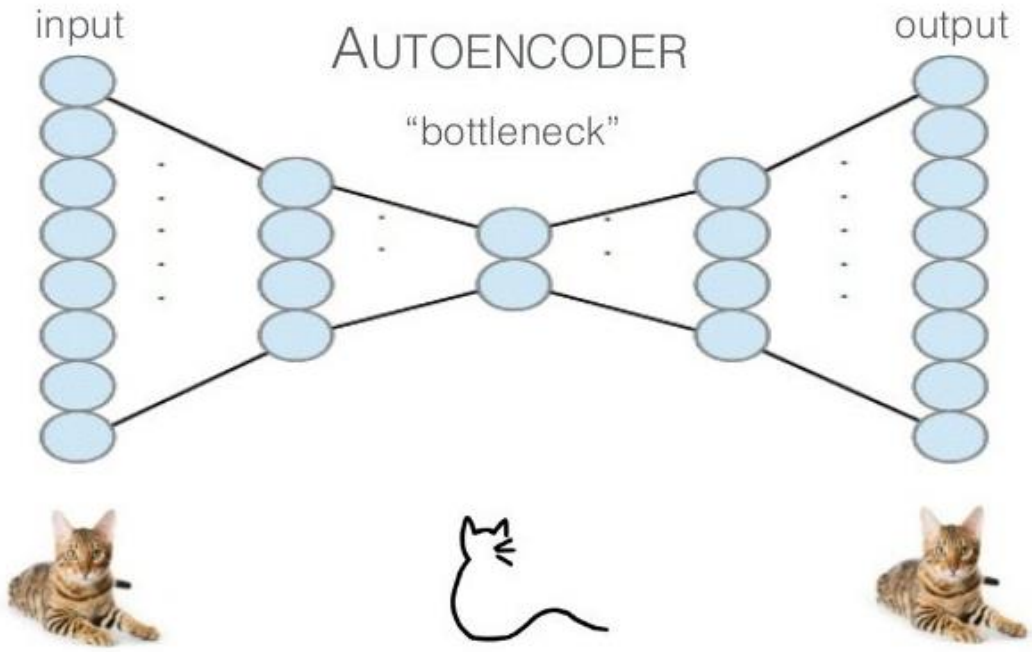
Autoencoder is an unsupervised artificial neural network that learns how to efficiently compress and encode data then learns how to reconstruct the data back from the reduced encoded representation to a representation that is as close to the original input as possible.
Autoencoder, by design, reduces data dimensions by learning how to ignore the noise in the data.
Here is an example of the input/output image from the MNIST dataset to an autoencoder.



Autoencoder for MNIST

Design of Auto-Encoders:

They are actually traditional neural networks. Their design make them special. Firstly, they must have same number of nodes for both input and output layers. Secondly, hidden layers must be symmetric about center. Thirdly, number of nodes for hidden layers must decrease from left to centroid, and must increase from centroid to right.



The key point is that input features are reduced and restored respectively. We can say that input can be compressed as the value of centroid layer's output if input is similar to output. I said similar because this compression operation is not lossless compression.

Left side of this network is called as autoencoder and it is responsible for reduction. On the other hand, right side of the network is called as autodecoder and this is in charge of enlargement.

Autoencoder Components:

Autoencoders consists of 4 main parts:

1- **Encoder**: In which the model learns how to reduce the input dimensions and compress the input data into an encoded representation.

2- **Bottleneck**: which is the layer that contains the compressed representation of the input data. This is the lowest possible dimensions of the input data.

3- **Decoder**: In which the model learns how to reconstruct the data from the encoded representation to be as close to the original input as possible.

4- **Reconstruction Loss**: This is the method that measures measure how well the decoder is performing and how close the output is to the original input.

The training then involves using back propagation in order to minimize the network's reconstruction loss.

AUTO-ENCODER:- ENCODER AND DECODER

Encoder

Produces Code or Latent Representation

$$\mathbf{h} = s(\mathbf{W}\mathbf{x} + \mathbf{b}) = f(\mathbf{x})$$

Decoder

Produces Reconstruction of the input

$$\hat{\mathbf{x}} = s(\mathbf{W}'\mathbf{h} + \mathbf{b}') = g(\mathbf{h})$$

Tied weights when $\mathbf{W}' = \mathbf{W}^T$

AUTO-ENCODER:- LOSS FUNCTION

Given the output $\hat{\mathbf{x}} = g(f(\mathbf{x}))$
We want to minimize some reconstruction loss:

$$\mathcal{L}(\mathbf{x}, g(f(\mathbf{x})) = \hat{\mathbf{x}})$$

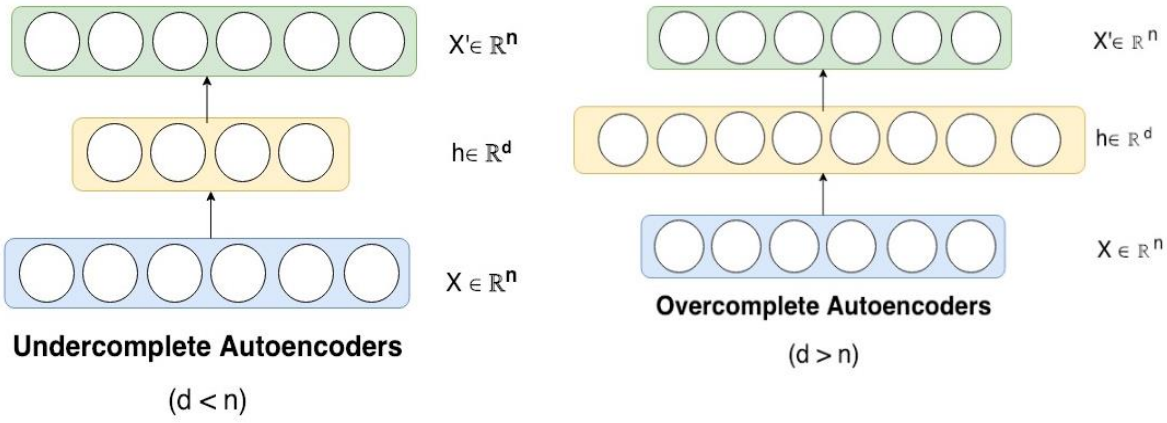
Cross entropy (bits or probability vectors)

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \mathbf{x} \log \hat{\mathbf{x}} + (1 - \mathbf{x}) \log(1 - \hat{\mathbf{x}})$$

Mean squared error (continuous values)

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = ||\mathbf{x} - \hat{\mathbf{x}}||^2$$

UNDERCOMPLETE AND OVERCOMPLETE AE



Undercomplete

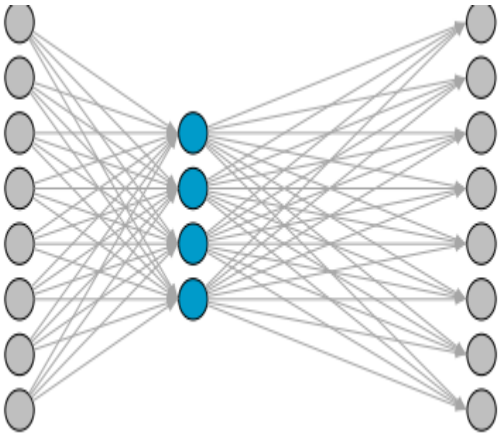
- Bottleneck layer produces code **h** with less dimensions then input **x**

Overcomplete

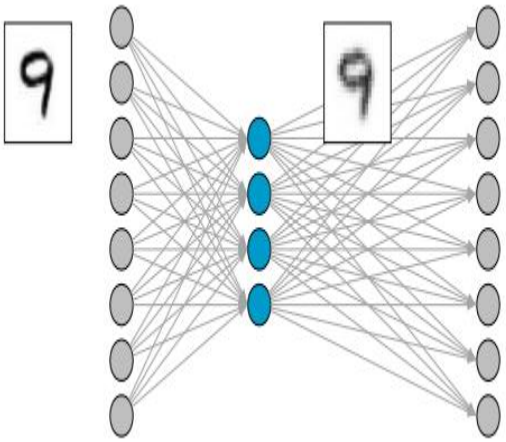
- Code **h** has more dimensions then the input **x**
- Different versions e.g. sparse, denoising, contractive.

UNDERCOMPLETE AUTO ENCODERS:

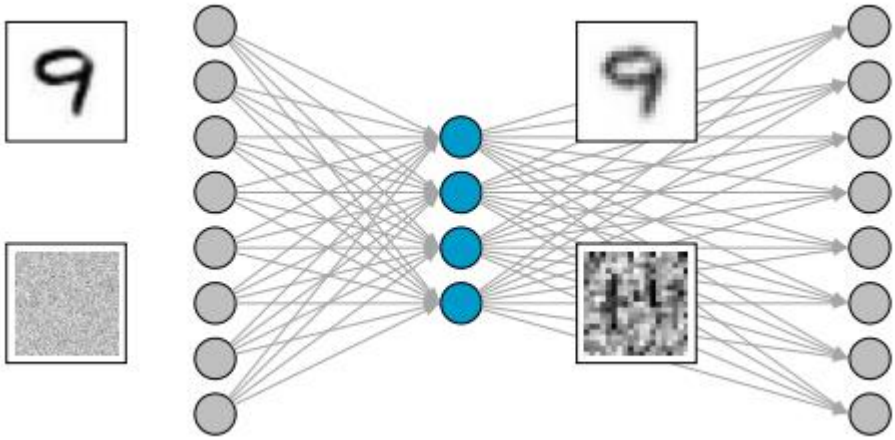
- Has a “bottleneck” layer.
- Can be used for Dimensionality Reduction — often compared to Principal Component Analysis (PCA).
- Often code is a good representation for the training data only.



1.



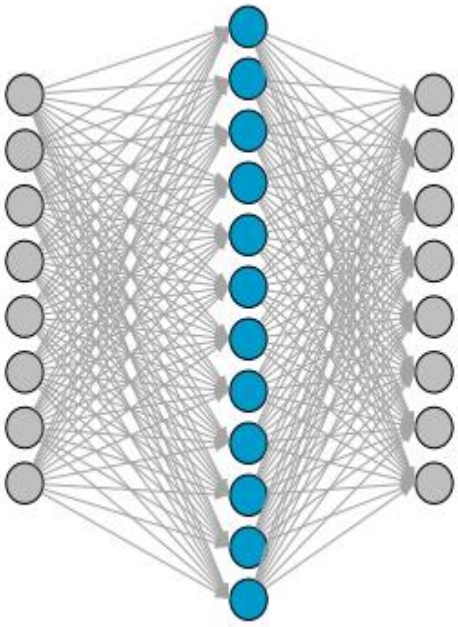
2.



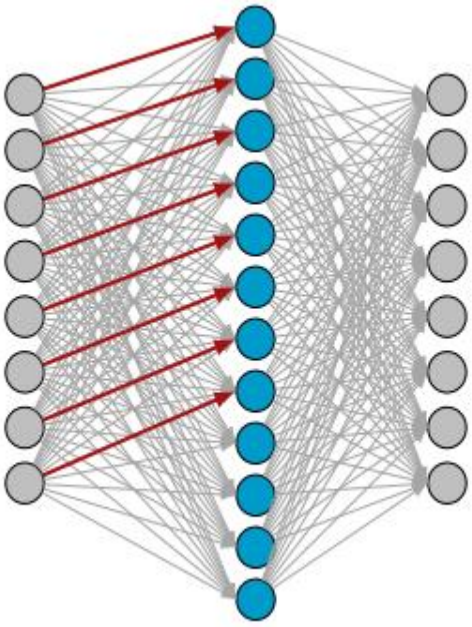
3.

UNDERCOMPLETE AUTO ENCODERS:

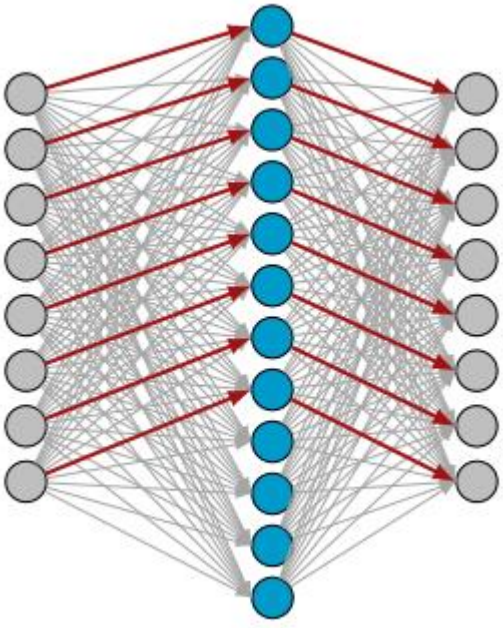
- High dimensional intermediate layer.



1.



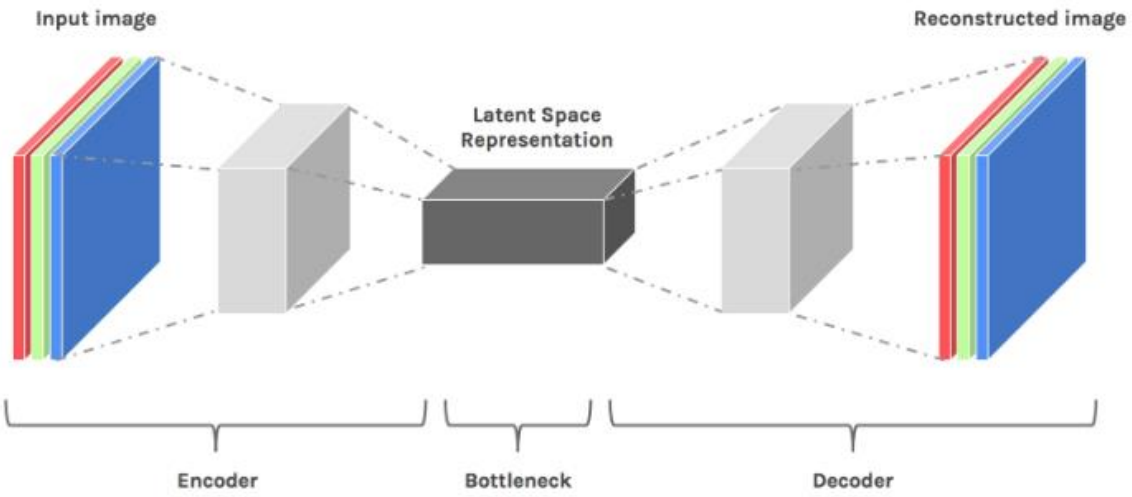
2.



3.

LATENT SPACE:

An autoencoder is made of two components, here's a quick reminder. The ***encoder*** brings the data from a high dimensional input to a ***bottleneck*** layer, where the number of neurons is the smallest. Then, the ***decoder*** takes this encoded input and converts it back to the original input shape — in our case an image. The ***latent space is*** the space in which the data lies in the bottleneck layer.



Convolutional Encoder-Decoder architecture

The latent space contains a **compressed** representation of the image, which is **the only information** the decoder is allowed to use to try to reconstruct the input **as faithfully as possible**. To perform well, the network has to learn to extract the **most relevant** features in the bottleneck.