# ARTIFICIAL NEURAL NETWORKS - WEEK 10

Autoencoders

Dr. Aamir Akbar

Director of both AWKUM AI Lab and AWKUM Robotics, Final Year Projects (FYPs) coordinator, and lecturer at the department of Computer Science
Abdul Wali Khan University, Mardan (AWKUM)

# CONTENTS

# AUTOENCODERS

## What is Autoencoder?

An autoencoder is a neural network used for unsupervised learning. It learns to efficiently encode and then decode data. Essentially, it compresses the input into a lower-dimensional representation (known as Latent Space) and then tries to reconstruct the input from this representation.

1. Encoder: This part of the network compresses the input data into a latent space representation. It consists of one or more layers that gradually reduce the dimensionality of the input data, typically through a series of non-linear transformations such as activation functions.
2. Latent Space: The compressed representation of the input data in a lower-dimensional space is known as the latent space. It captures the most salient features of the input data while discarding noise and irrelevant information.
3. Decoder: The decoder component of the autoencoder aims to reconstruct the original input data from the latent space representation. It consists of one or more layers that upscale the data back to its original dimensionality.

# SUPERVISED LEARNING

**Data:** $(x, y)$

where $x$ is data, $y$ is label

**Goal:** learn a function to map,

# SUPERVISED LEARNING

**Data:** $(x, y)$

where $x$ is data, $y$ is label

**Goal:** learn a function to map,

$x \rightarrow y$

# SUPERVISED LEARNING

**Data:** $(x, y)$

where $x$ is data, $y$ is label

**Goal:** learn a function to map,

$x \rightarrow y$

**Examples:** Classification and Regression Tasks

1. Recognizing handwritten digits in the MNIST dataset.
2. Predicting the ETH-USD price based on historical data.

# UNSUPERVISED LEARNING

**Data:** $x$

where $x$ is data, no labels

**Goal:** learn the hidden or underlying structure of the data

# UNSUPERVISED LEARNING

**Data:** $x$

where $x$ is data, no labels

**Goal:** learn the hidden or underlying structure of the data

**Examples:** Clustering, features or dimensionality reduction.

# ENCODERS

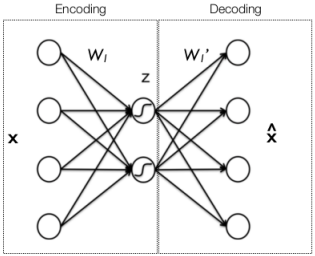
Encoders learns mapping of the data, $x$, to a lower-dimensional latent space, $z$.

# DECODERS

Decoders learns mapping back from latent space, $z$, to a reconstructed observation, $\hat{x}$.

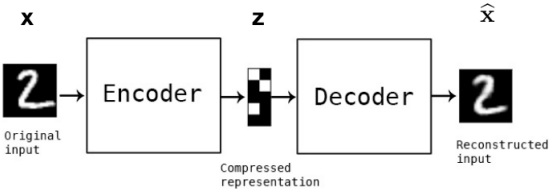
# AUTOENCODERS ARCHITECTURE

In such network, the output layer has the same number of nodes as the input layer.



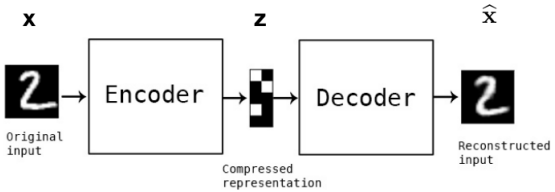

The network is often composed in such a way that the number of nodes in the hidden layer is smaller than the number of nodes in the input and output layers

# TRAINING AUTOENCODERS




To train such a network we do not need labels, that is why they can be used for unsupervised learning

# TRAINING AUTOENCODERS



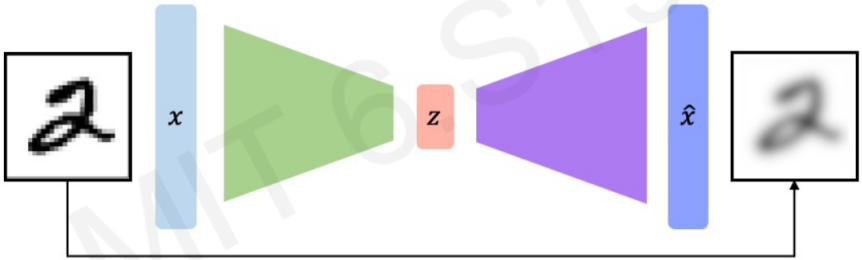

To train such a network we do not need labels, that is why they can be used for unsupervised learning

However, they are trained by using Backpropagation and minimizing the difference between $\hat{x}$ and $x$.

# TRAINING AUTOENCODERS

The mapping function $h_\theta(x)$ is trained to reconstruct its own inputs instead of predicting a target value.

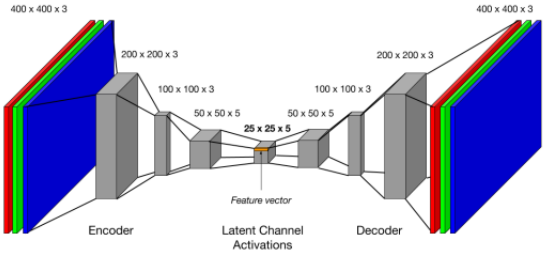
$$\hat{x} = h_\theta(x) \tag{1}$$



# DEEP CONVOLUTIONAL AUTOENCODERS

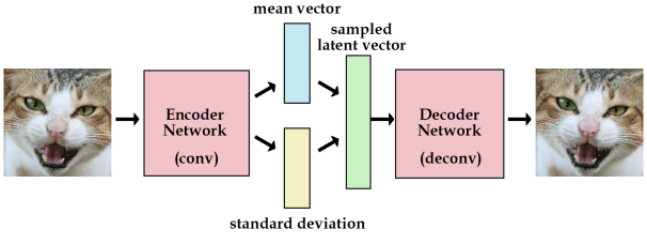
The encoder and the decoder can be convolutional and deconvolutional neural networks respectively.

The inner-layer is called the latent space representation of the data set.



# VARIATIONAL AUTOENCODER (VAE)

We encourage the encoder to produce latent representations that are distributed similar to a standard Gaussian (with mean 0 and variance 1).



# VARIATIONAL AUTOENCODER (VAE)

## Example: generating new images

- Generating new images is now easy: all we need to do is sample a latent vector from the unit gaussian and pass it into the decoder.
- Examples:

Fictional celebrities (by Alec Radford)

# RESOURCES

To download the source codes used in the previous slides, follow the link:

▸ Download Source Codes

Import the codes into your preferred development environment, such as Visual Studio Code (VS Code), to practice and explore further.

To learn programming in Python, follow my comprehensive 15-week Programming in Python course at:

▸ Programming in Python