simplilearn | PURDUE UNIVERSITY

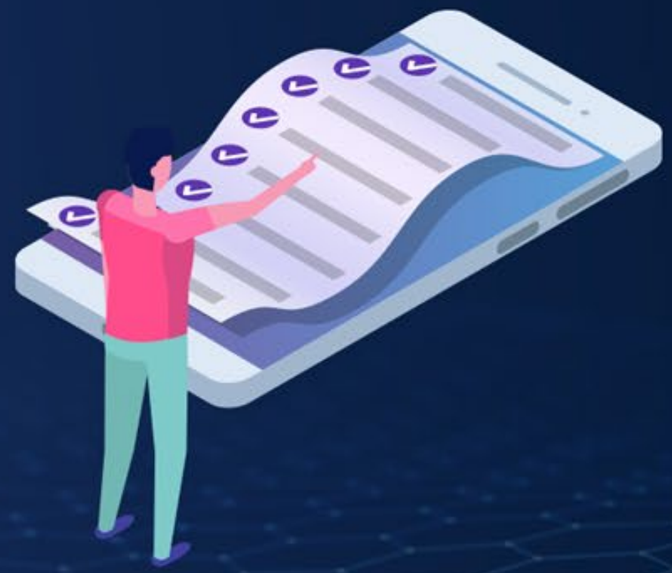
Deep Learning with Keras with TensorFlow

Autoencoders

Learning Objectives

By the end of this lesson, you will be able to:

- Explore autoencoders and their functionalities
- Create autoencoder model to detect anomalies

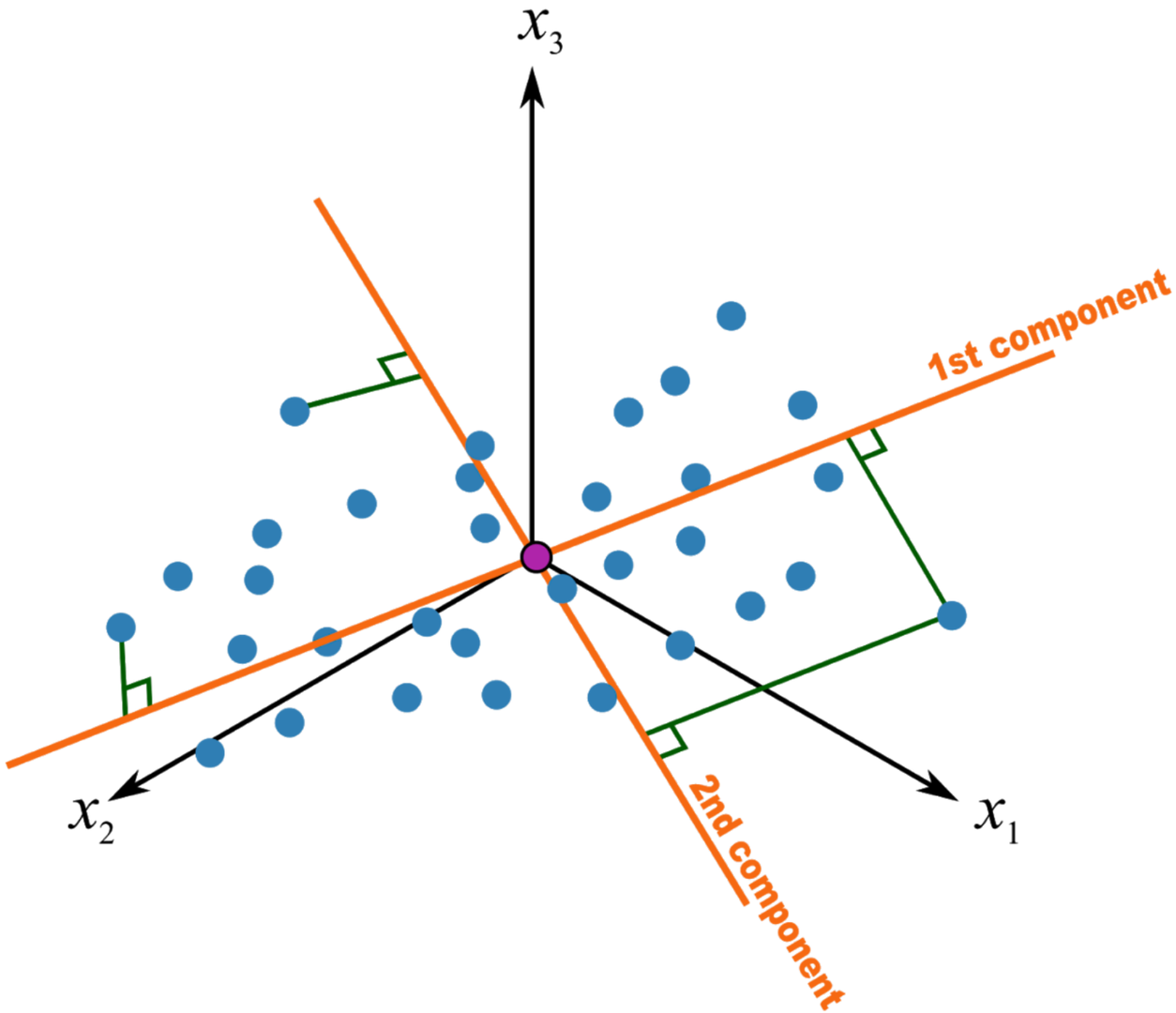


Unsupervised Learning

> Unsupervised learning is to learn the structure of the data and correlations between features from input data with no labels. For example, clustering, compression, feature and representation learning, dimensionality reduction, generative models, etc.

Principal Component Analysis (PCA)

- Invented by Karl Pearson in 1901
- Is a statistical approach for data compression and visualization
- Has a drawback as it analyzes linear components only

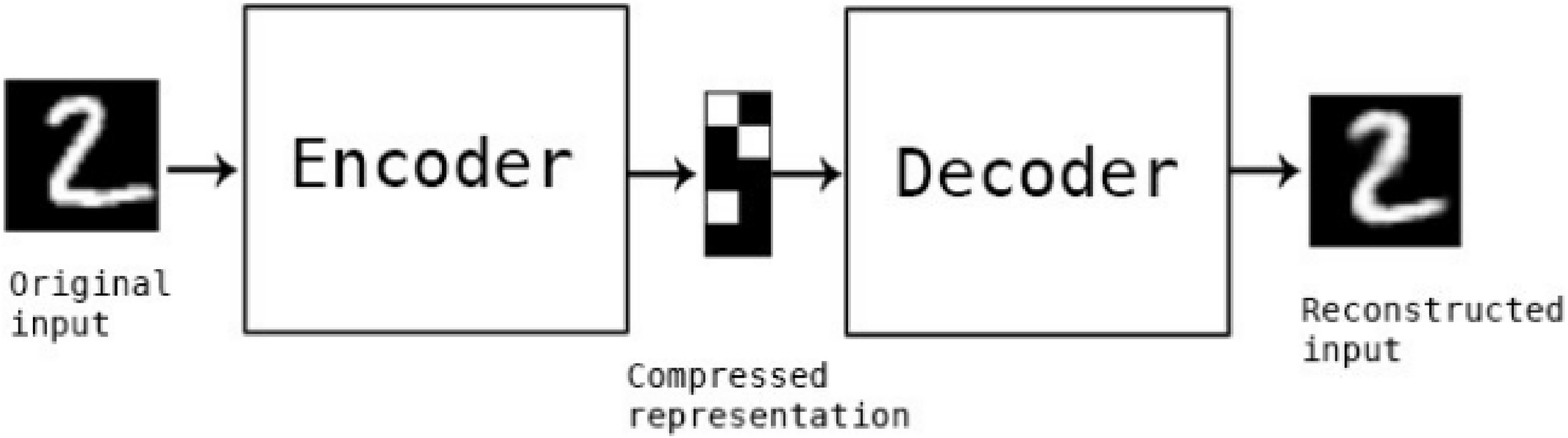


Autoencoders

> An autoencoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner.

Kramer, Mark A. (1991). "Nonlinear principal component analysis using auto associative neural networks"

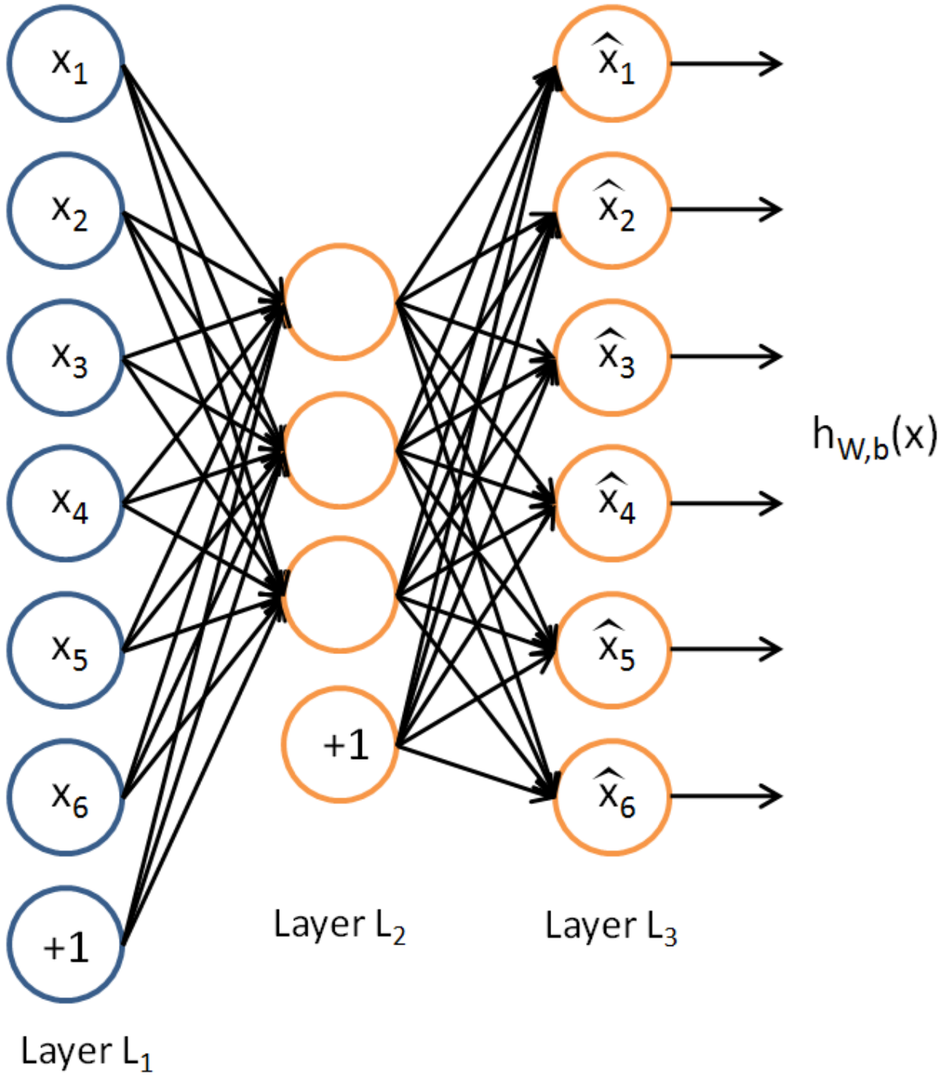
Autoencoder: Example



Autoencoder for MNIST

Autoencoders

- Analyzes nonlinear components unlike PCA
- Achieves the PCA capacity even without activation functions



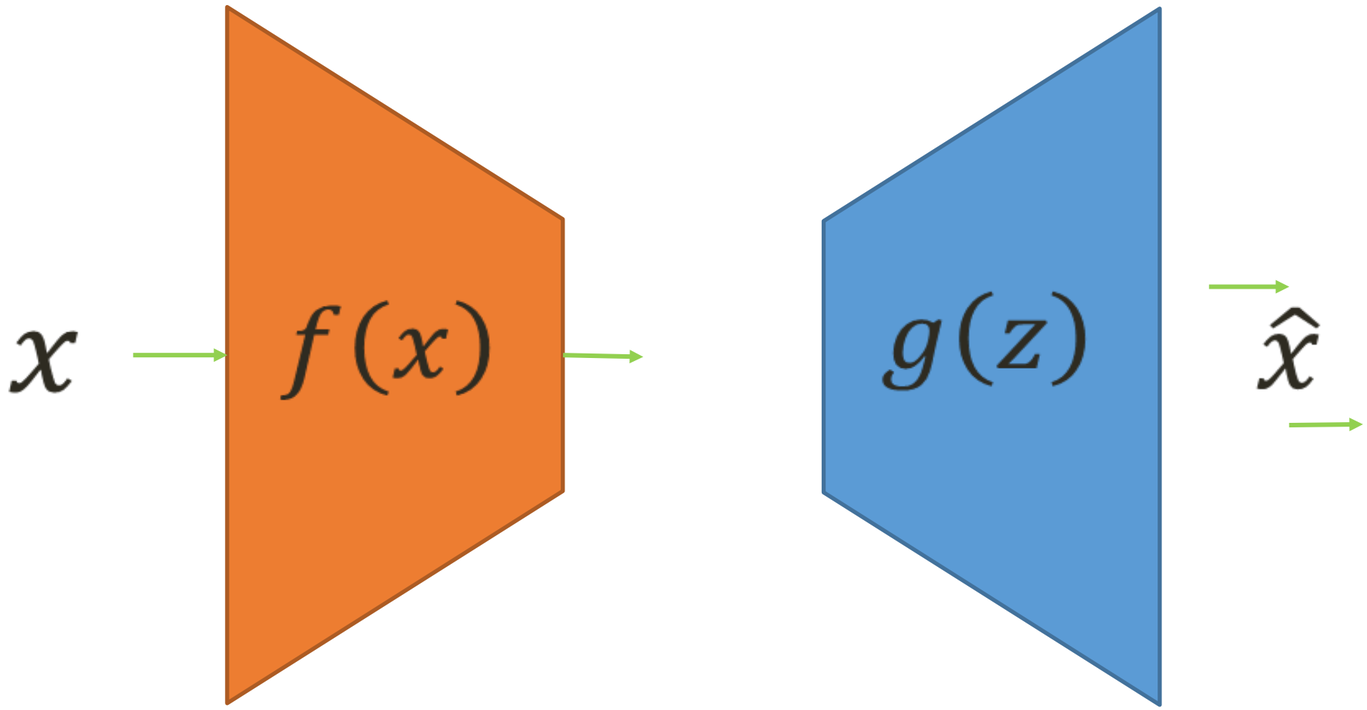
Uses of Autoencoders

Components of an Autoencoder

Encoder	Reduces the input dimensions and compresses the input data into an encoded representation
Bottleneck	Contains the compressed representation of the input data in its lowest possible form
Decoder	Reconstructs the data from the encoded representation to the original as same as possible
Reconstruction Loss	Measures the performance of the decoder and the similarities between the input and the output

Autoencoder Formulation

Data with no label x goes through the function encoder function f(X) and decoder g(z) and converts into $\hat{x}$.



Autoencoder Formulation

$$f(x) = s(wx + b) = z$$

and

$$g(z) = s(w'z + b') = \hat{x}$$

such that , $h(x) = g\big(f(x)\big) = \hat{x}$

z is a latent representation or code and s is a nonlinearity such as the sigmoid

$\hat{x}$ is x's reconstruction

h is an approximation of the identity function.

Training an Autoencoder

- To train the model, gradient descent is used

- Use squared loss function

$$L(x, \hat{x}) = \|x - \hat{x}\|^2$$ e number of hidden neurons or regularization) we can learn information about the structure of the data.

- If input is interpreted as bit vectors or vectors of bit probabilities, the cross entropy can be used as a loss function

$$H(p,q) = -\sum_x [p(x) \log q(x)]$$

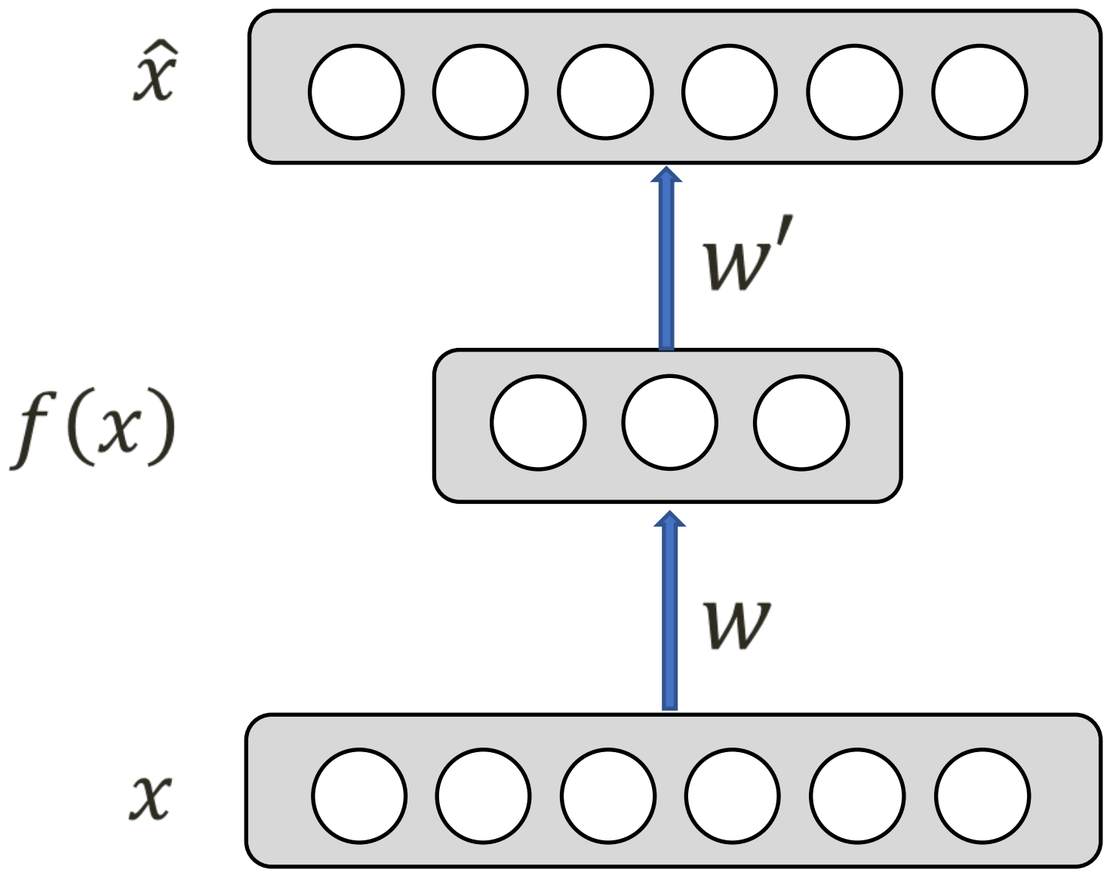
e number of hidden neurons or regularization) we can learn information about the structure of the data.

Types of Autoencoder

- Undercomplete AE
- Overcomplete AE

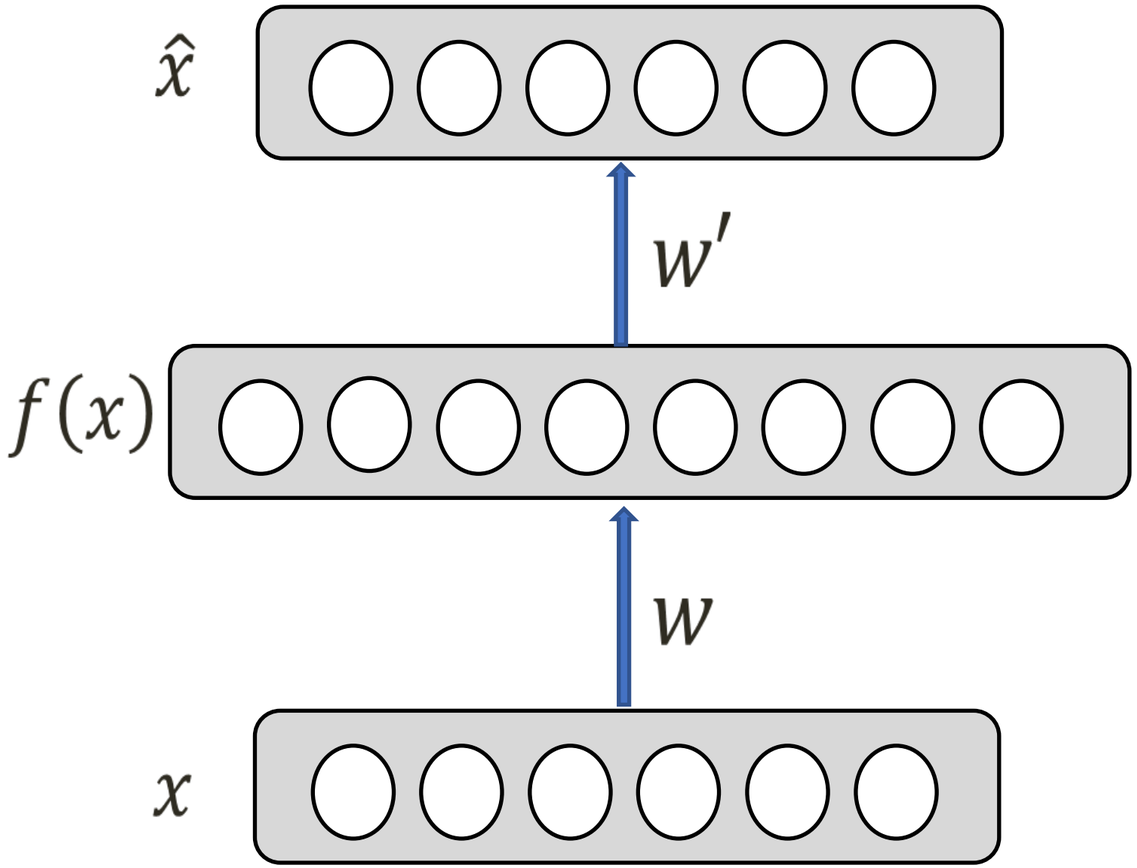
Undercomplete AE

- If hidden layer is smaller than input layer, it is undercomplete.
  - Compresses the input
  - Compresses well only for the training distribution
- Hidden nodes are:
  - Good features for the training distribution
  - Bad for other types of features on the input data



Overcomplete AE

- Hidden layer is overcomplete, if greater than the input layer
  - ❑ No compression in hidden layer
  - ❑ Each hidden unit could copy a different input component
- Adding dimensions is good for training a linear classifier
- x A higher dimension code helps model a more complex distribution



What Is Anomaly Detection?

> Anomaly detection is the identification of rare items, events, or observations which raise suspicions by differing significantly from the majority of the data.

Build an Autoencoder Model

Problem Statement: Build an autoencoder model to regenerate the objects of the given MNIST dataset.

Objective: Regenerate the objects of MNIST dataset using autoencoder model.

Access: Click the Practice Labs tab on the left panel. Now, click on the START LAB button and wait while the lab prepares itself. Then, click on the LAUNCH LAB button. A full -fledged jupyter lab opens, which you can use for your hands -on practice and projects.

ASSISTED PRACTICE

Key Takeaways

Now, you are able to:

- Explore autoencoders and their functionalities
- Create autoencoder model to detect anomalies

DATA AND
ARTIFICIAL INTELLIGENCE

Knowledge Check

Knowledge Check 1	Which of the following model categories does an autoencoder fall into?

a. Supervised

b. Unsupervised

c. Semi-Supervised

d. None of the above

Knowledge Check 1	Which of the following model categories does an autoencoder fall into?

a. Supervised

b. Unsupervised

c. Semi-Supervised

d. None of the above

The correct answer is **b**

An autoencoder falls under the category of unsupervised model.

Knowledge Check

2

Which of the following components are the parts of an autoencoder?

a. Encoder

b. Decoder

c. Coder

d. All of the above

Knowledge Check

2

Which of the following components are the parts of an autoencoder?

a. Encoder

b. Decoder

c. Coder

d. All of the above

The correct answer is **d**

Encoder, coder, and decoder all are parts of an autoencoder.

Knowledge Check 3	Which of the following part s of an autoencoder compresses input into latent sparse representation?

a. Encoder

b. Decoder

c. Coder

d. None of the above

Knowledge Check

3

Which of the following part of an autoencoder compresses input into latent sparse representation?

a. Encoder

b. Decoder

c. Coder

d. None of the above

The correct answer is **a**

Encoder compresses input into latent sparse representation.

Knowledge Check

4

Which of the following falls into generative model category?

a. PCA

b. Autoencoder

c. Vibrational Autoencoder

d. None of the above

Knowledge Check

4

Which of the following falls into generative model category?

a. PCA

b. Autoencoder

c. Vibrational Autoencoder

d. None of the above

The correct answer is **C**

Variational Autoencoder falls into the generative model category.

Anomaly Detection in Credit Card Transactions

Problem Statement: Build an autoencoder model to detect fraudulent credit card transactions for the given data set.

Objective: Use Keras deep learning framework to build the autoencoder model to detect anomalies in credit card transactions.

Access: Click the Practice Labs tab on the left panel. Now, click on the START LAB button and wait while the lab prepares itself. Then, click on the LAUNCH LAB button. A full -fledged jupyter lab opens, which you can use for your hands -on practice and projects.

LESSON-END PROJECT