Assignment No. 4

Problem Statement:

Use Autoencoder to implement anomaly detection.
Build the model by using

a. Import required libraries

b. Upload/access the dataset

c. Encoder converts it into latent representation

d. Decoder networks convert it back to the original input

e. Compile the models with Optimizer, Loss, and Evaluation

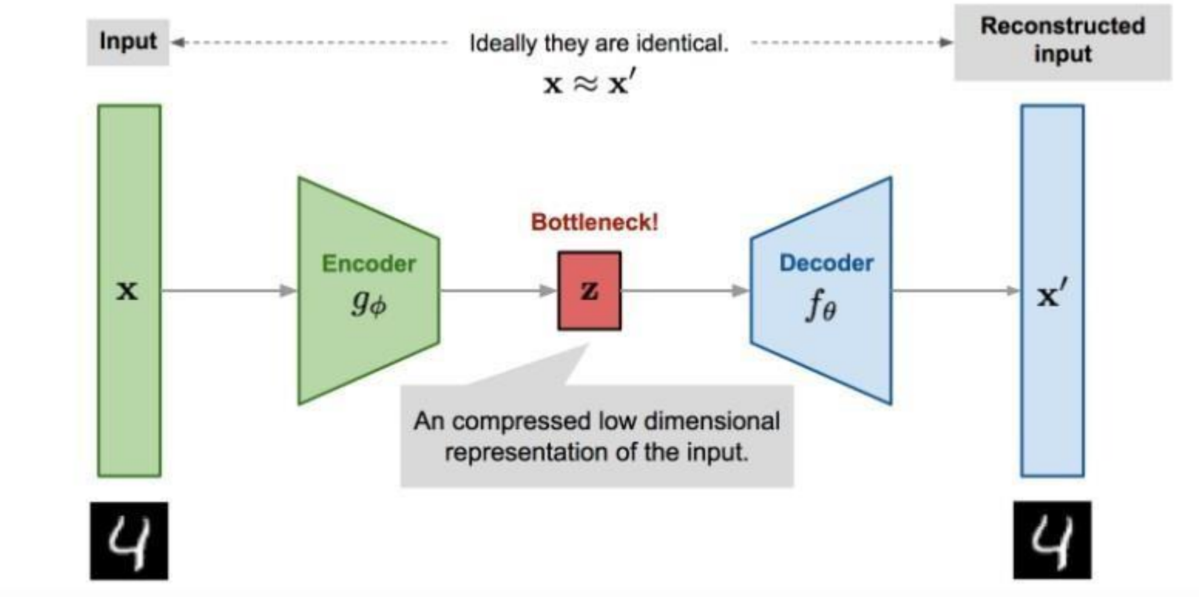
Solution Expected:

AutoEncoders are widely used in anomaly detection. The reconstruction errors are used as theanomaly scores. Let us look at how we can use AutoEncoder for anomaly detection using TensorFlow.

Import the required libraries and load the data. Here we are using the ECG data which consists oflabels 0 and 1. Label 0 denotes the observation as an anomaly and label 1 denotes the observationas normal.

Methodology to be used:

AutoEncoder is a generative unsupervised deep learning algorithm used for reconstructing high- dimensional input data using a neural network with a narrow bottleneck layer in the middle whichcontains the latent representation of the input data.


Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
x
Encoder
g_ϕ
Bottleneck!
z
Decoder
f_θ
$\mathbf{x}'$
An compressed low dimensional representation of the input.

Time and Amount are the columns that are not scaled, so applying StandardScaler to only Amountand Time columns. Normalizing the values between 0 and 1 did not work great for the dataset.

The last column in the dataset is our target variable.

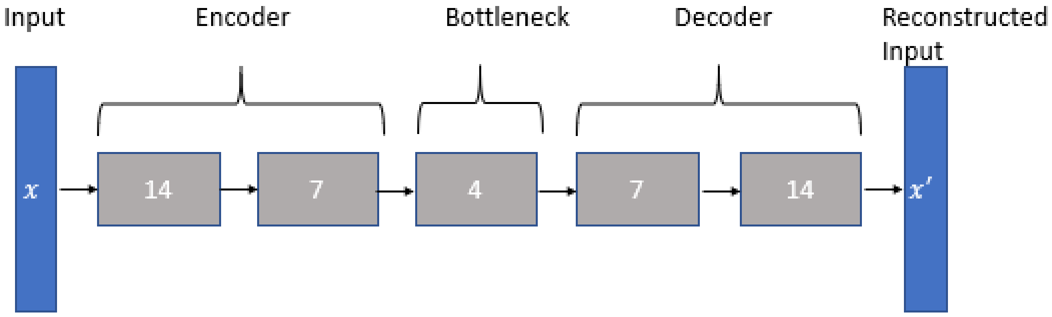
Normalize the data to have a value between 0 and 1Use only normal transactions to train the Autoencoder.

Normal data has a value of 0 in the target variable. Using the target variable to create a normaland fraud dataset.

Set the training parameter values

Create the Autoencoder

The architecture of the autoencoder is shown below.



Plot training and test loss

Plot training and test loss

Detect Anomalies on test data

Anomalies are data points where the reconstruction loss is higher.

To calculate the reconstruction loss on test data, predict the test data and calculate the meansquare error between the test data and the reconstructed test data.

Plotting the test data points and their respective reconstruction error sets a threshold value tovisualize if the threshold value needs to be adjusted.

detect anomalies as points where the reconstruction loss is greater than a fixed threshold. Herewe see that a value of 52 for the threshold will be good.

As our dataset is highly imbalanced, we see a high accuracy but a low recall and precision. Thingsto further improve precision and recall would add more relevant features, different architecture for autoencoder, different hyperparameters, or a different algorithm.

Conclusion:

Autoencoders can be used as an anomaly detection algorithm when we have an unbalanceddataset where we have a lot of good examples and only a few anomalies. Autoencoders are trained to minimize reconstruction error. When we train the autoencoders on normal data or good data, we can hypothesize that the anomalies will have higher reconstruction errors than the goodor normal data.