Credit Card Fraud Detection Using Autoencoder Model in Unbalanced Datasets

Journal of Advances in Mathematics and Computer Science

33(5): 1-16, 2019; Article no.JAMCS.51106

ISSN: 2456-9968
(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

Outline

1. Introduction
2. Autoencoder
3. Experimental & Results
4. Conclusion

Introduction(1/3)

1. Fraud Detection System (FDS) should not only be effective, but should also be costeffective.
2. To minimize costs, expert rules and models based on machine learning are used to conduct the firstly examination between fraudulent and legitimate transactions and to require investigators to review high-risk cases only
3. With techniques machine learning (ML) we can detect fraudulent patterns efficiently and impact transactions that are likely to be fraudulent.

Introduction(2/3)

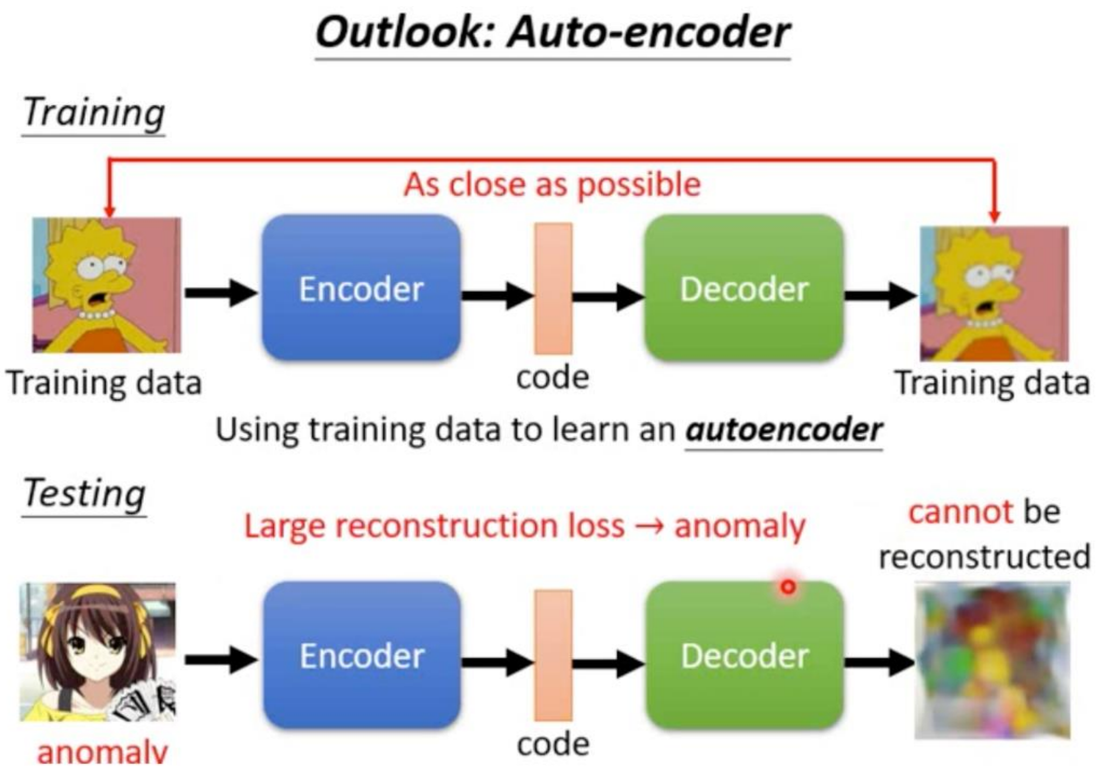
1. Adopt a new model for detecting Fraudulent credit card transaction using deep Learning Algorithm called Autoencoders
2. The propose model can achieve higher performance than the other state-of-the-art one-class methods according to Recall.

Introduction(3/3)

1. The solution relies on training for the autoencoder for the reconstruction normal data.
2. Anomalies are detected by defining a reconstruction error threshold and considering the cases with a superior threshold as anomalies.

Autoencoder(1/3)

1. 把所有訓練資料 Encode
2. 輸入一張圖片先 encode 成一個向量 code 再依據 code 過 decode 回原來的圖片
3. 訓練過程： 讓 Decoder 後的圖片 ，跟 input 端的圖片越像越好！
4. 機制： 如果是正常的測試資料輸入，圖片可以還原回來，還原度較高。異常資料輸入時， Decoder 還原出來得圖片就會根原圖差異蠻大的，可作為偵測異常用

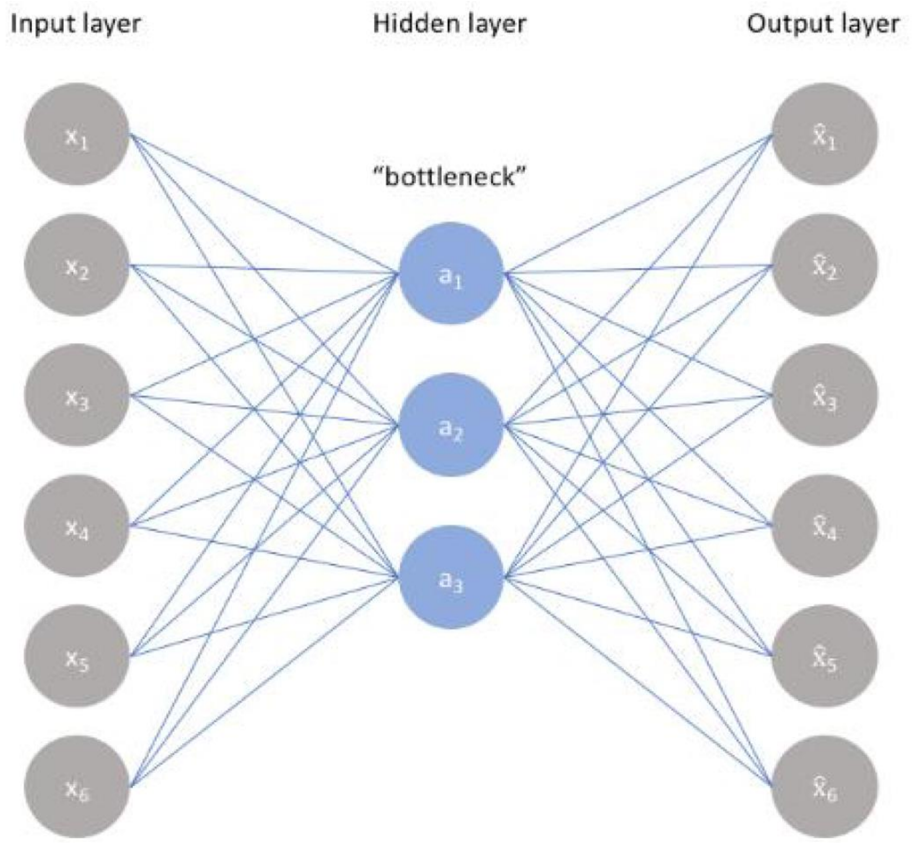


/iolin-tao.blogspot.com/2019/03/ml-anomaly-
on-machine-learning.html

Autoencoder(2/3)

Autoencoders architecture consists of four main parts

1. Encoder (number of nodes in middle layer)
2. Bottleneck
3. Decoder
4. Reconstruction Loss

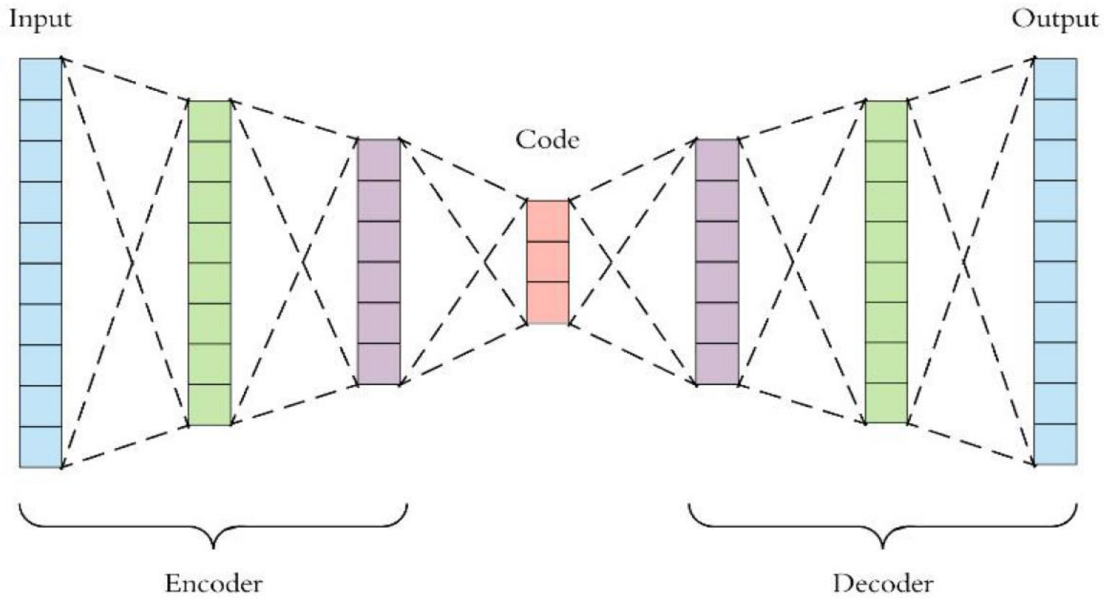


Autoencoder(3/3)

Four hyper parameters

1. Code size (number of nodes in middle layer)
2. Number of layers
3. Number of nodes per layer
4. Loss Function (Mean square error)

$$l(x,\hat{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_{i-}\hat{x_i})^2$$



Experimental & Results (1/6)

1. Two day's transactions of credit cards Number of layers
2. 284,315 normal / 492 fraudulent transactions
3. Highly unbalanced (0.172%)
4. Mean squared error is used to calculate the value of the reconstruction error
5. The high error value indicates the discovery of fraudulent transactions while the low value reveals legitimate transactions

Experimental & Results (2/6)

The decrease in the value of loss from reconstruction as the number of repeat increases

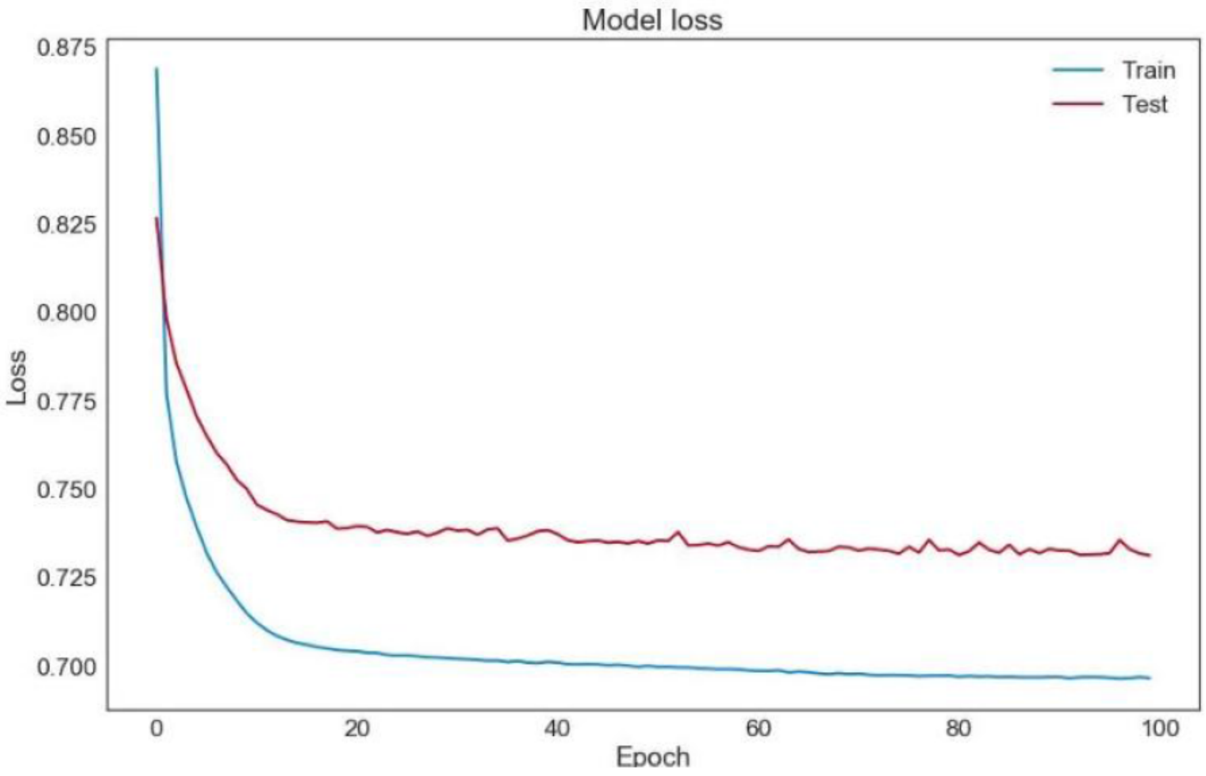


Fig. 3. Reconstruction error vs. Epochs

Experimental & Results (3/6)

The higher the value of the threshold is, the higher the precision is, while the value of the recall decreases.

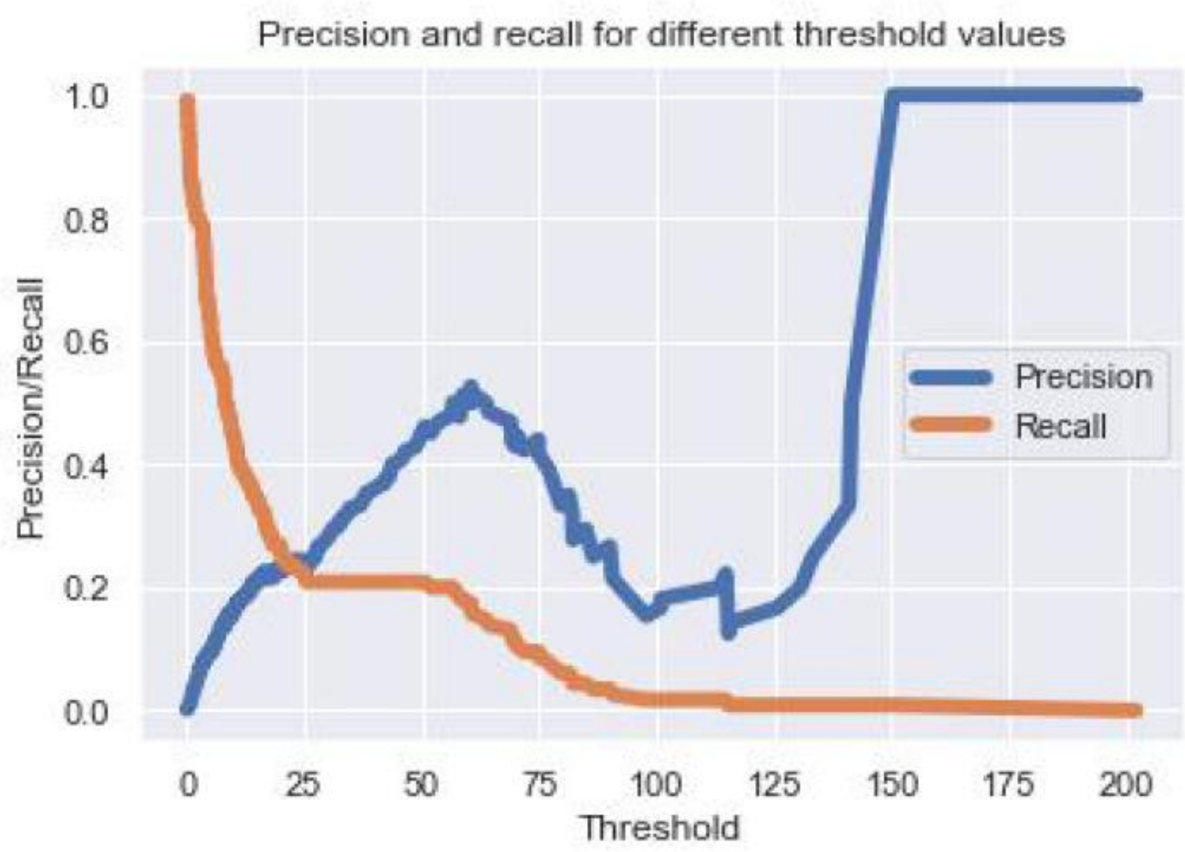
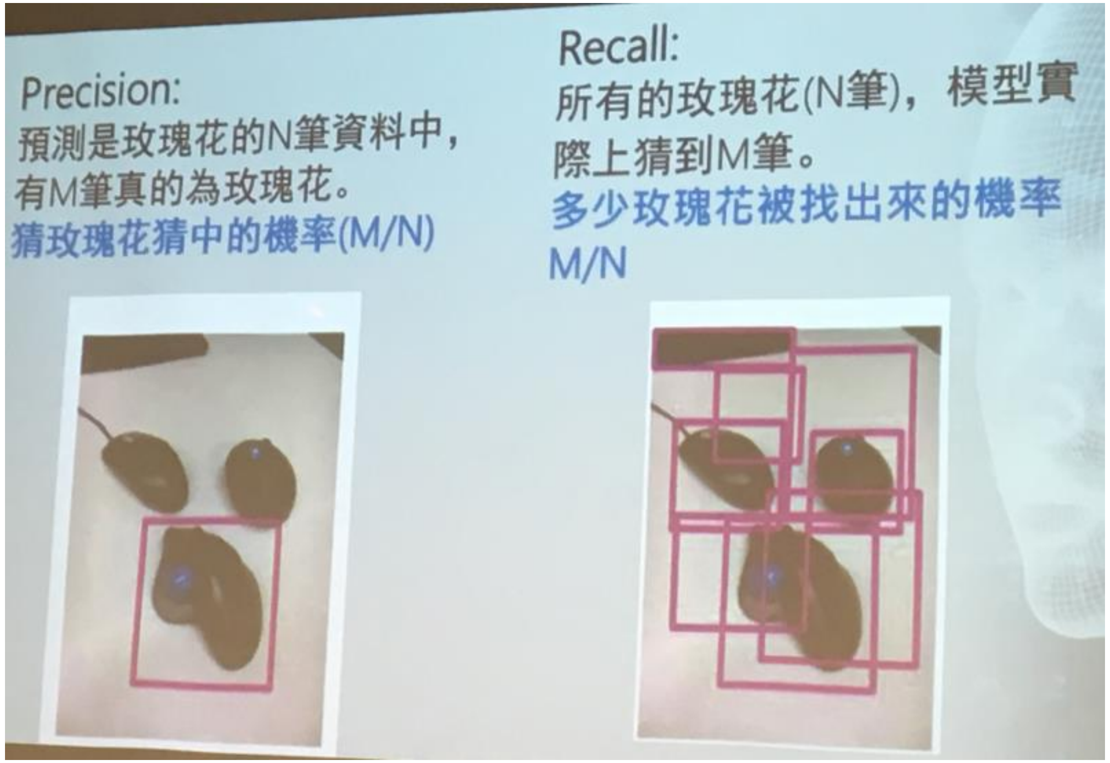


Fig. 6. Precision and recall values with various values of the threshold



Experimental & Results (4/6)

The high error value indicates the discovery of fraudulent transactions while the low value reveals legitimate transactions

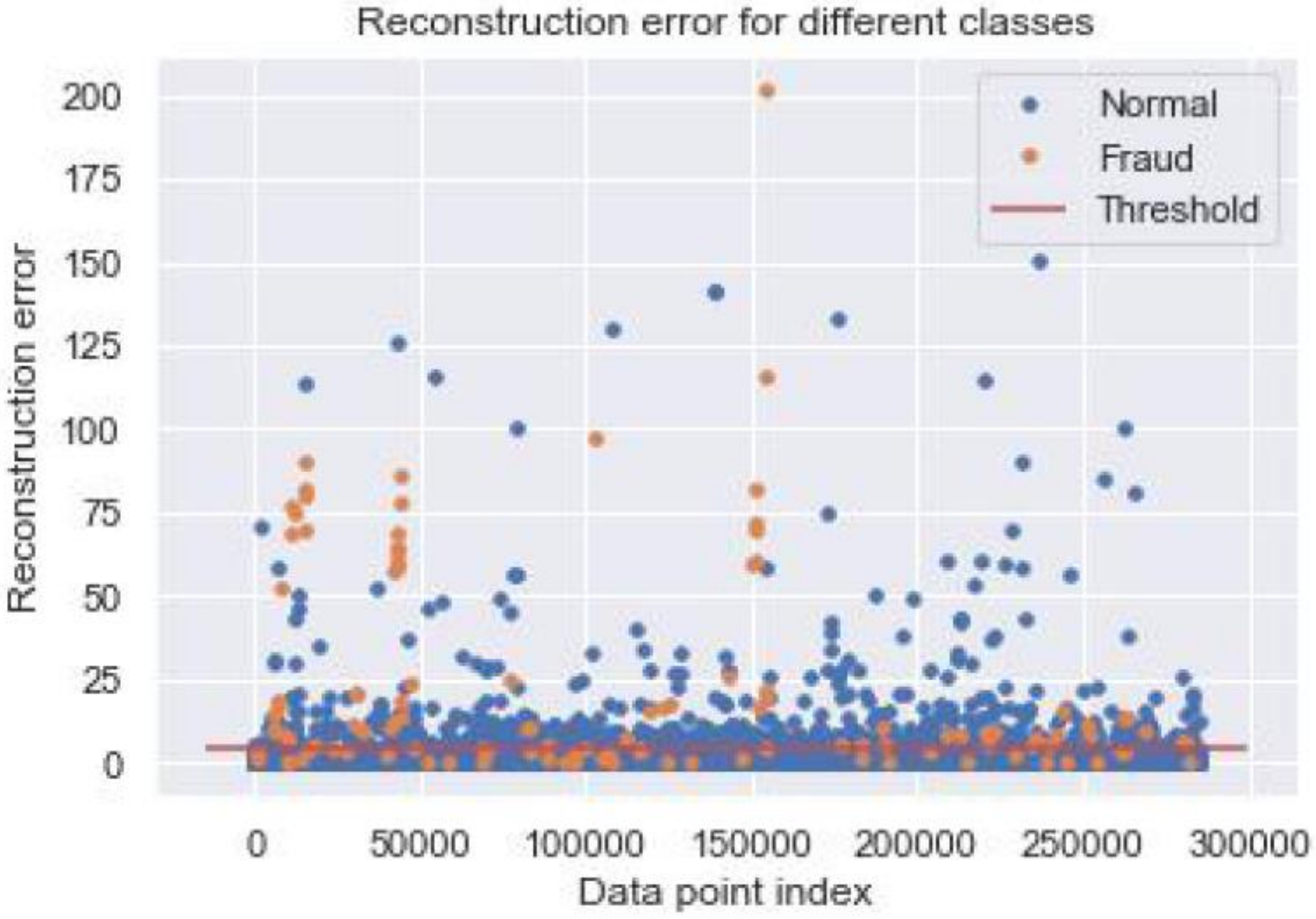


Fig. 7. Data distribution in threshold 5

Experimental & Results (5/6)

Confusion matrix is used to describe the performance of the classification model

- When Threshold = 5

Table 4. Threshold=5

		Predicted Values	
		0	**1**
Actual Values	0	56150	697
	1	41	74

It shows a matrix of confusion Table 4, that the model with threshold =5 is able to control about 60% of cases of fraud.

$$hit\ rate = \frac{TP}{TP+FN} * 100 = \frac{74}{74+41} * 100 = 64\% \tag{3}$$

While the proportion of legitimate transactions classified as fraudulent

$$False\ Positive\ rate = \frac{FP}{FP+TN} * 100 = \frac{697}{697+56150} * 100 = 1.2\%$$

Experimental & Results (6/6)

1. Autoencoder network does not need to use the data balance methods
2. Autoencoder can achieve higher performance than logistic regression according to Recall.

Table 6. Comparison of LR and autoencode

	Accuracy	Recall	F
LR (balance Data)	97.23	0.90	0
LR (unbalance Data)	99.91	0.57	0
Autoencoder (Thr=5)	98.70	0.64	0
Autoencoder (Thr=3)	97.70	0.79	0
Autoencoder (Thr=1)	90.02	0.86	0
Autoencoder (Thr=0.7)	80.00	0.91	0

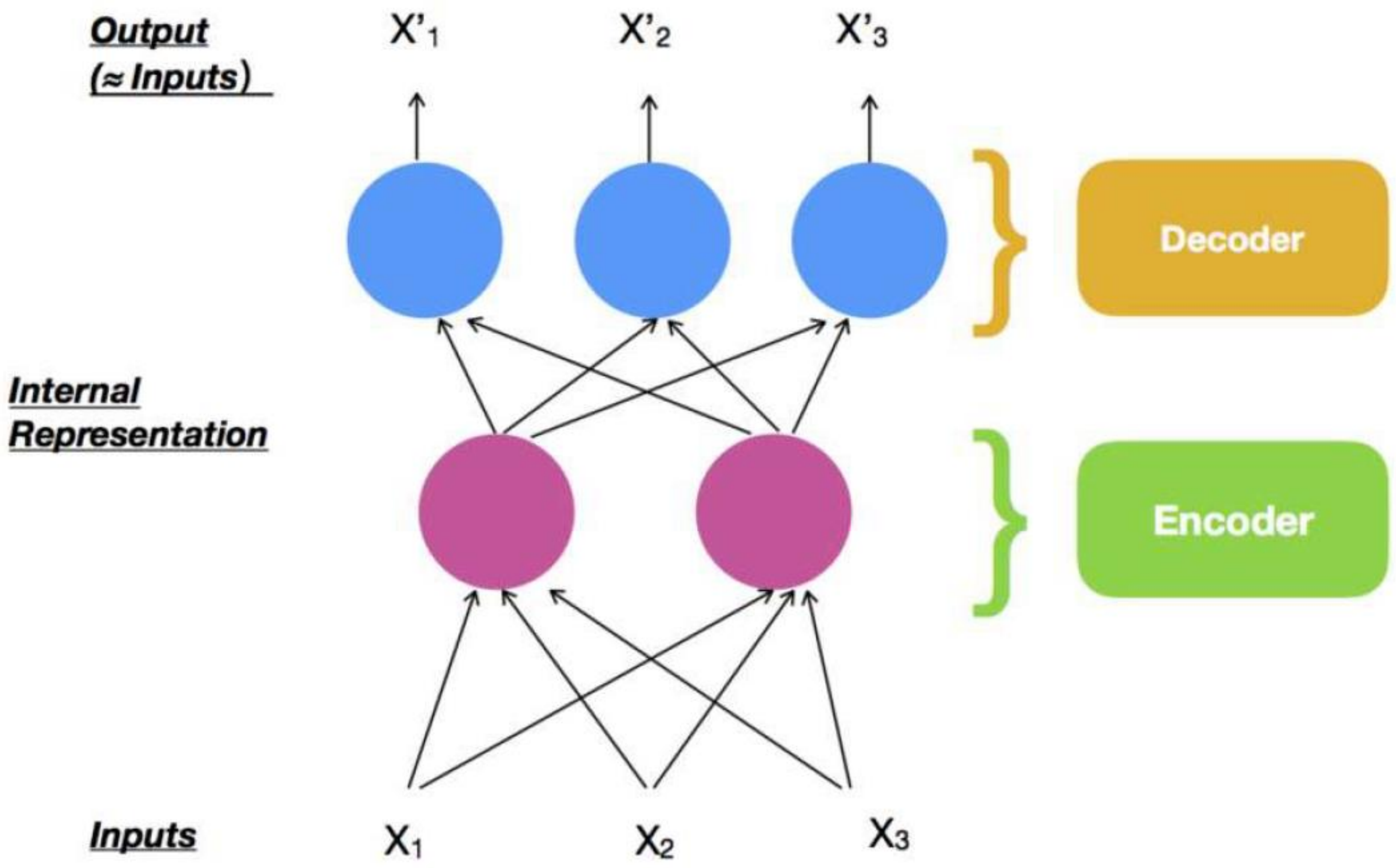
	Recall
LR (balance Data)	0.90
LR (unbalance Data)	0.57
Autoencoder (Thr=5)	0.64
Autoencoder (Thr=3)	0.79
Autoencoder (Thr=1)	0.86
Autoencoder (Thr=0.7)	0.91

Conclusion

1. Autoencoder can deal with unbalanced datasets
2. The solution relies on training for the autoencoder for the reconstruction normal data.
3. Anomalies are detected by defining a reconstruction error threshold and considering the cases with a superior threshold as anomalies.

Thank You

Autoencoder neural network



Autoencoder neural network

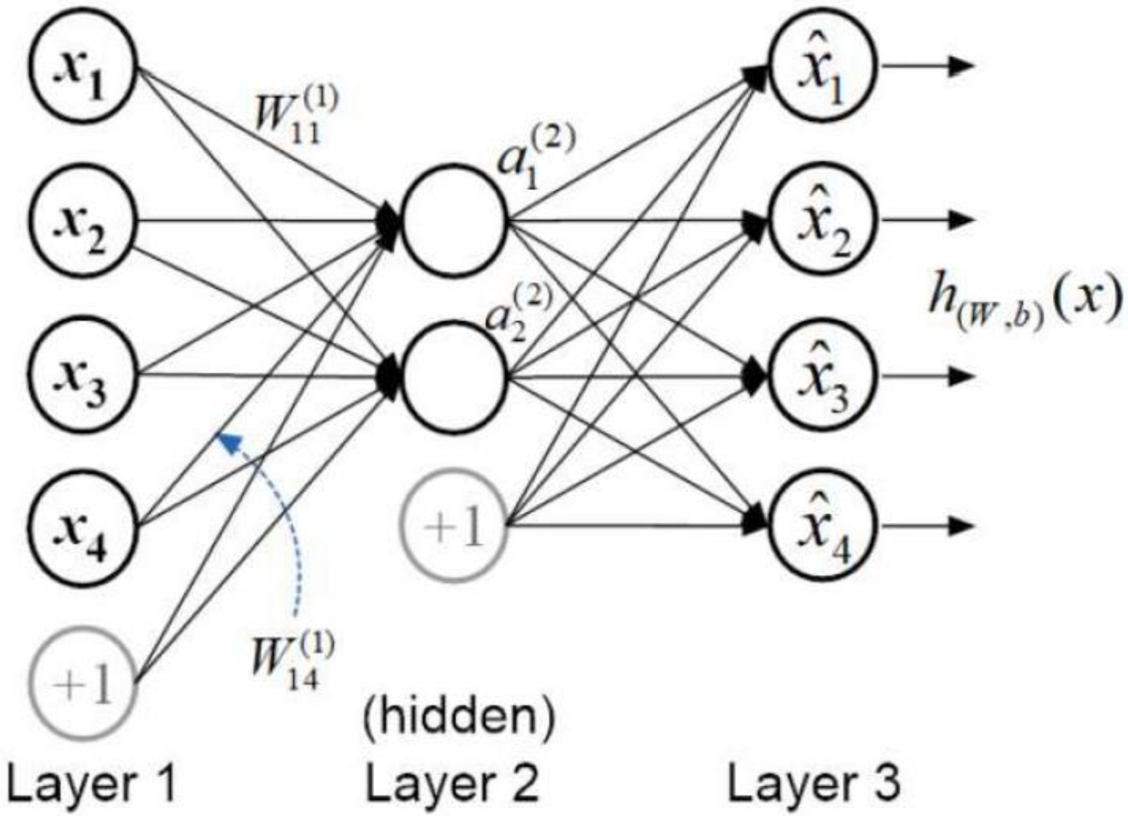


Figure 1: An autoencoder neural network.

The hyper parameters

- Neural Network=input+Weight+Bias+Activation function

$$f(z) = \frac{1}{1 - e^{-z}}.$$

$$a_1^{(2)} = f(W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + \ldots + W_{14}^{(1)} x_4 + b_1^{(1)})$$
$$a_2^{(2)} = f(W_{21}^{(1)} x_1 + W_{22}^{(1)} x_2 + \ldots + W_{24}^{(1)} x_4 + b_2^{(1)})$$

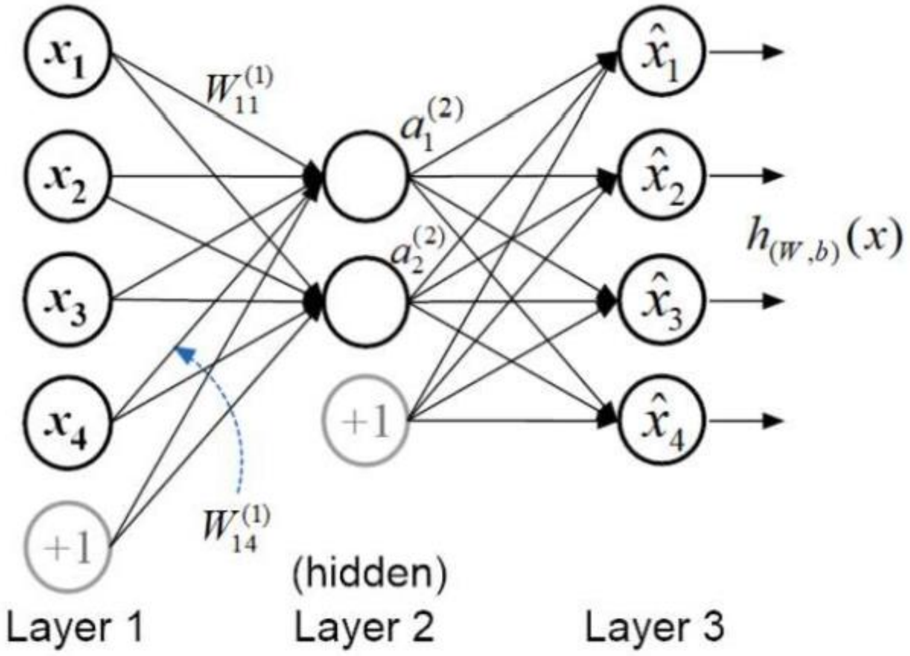


Figure 1: An autoencoder neural network.

Reference

- Basics of Autoencoders.pdf
- https://www.slideshare.net/AshokGovindarajan/autoencoders-147905176?from_action=save

- https://blog.keras.io/building-autoencoders-in-keras.html

- T. Luo and S. G. Nagarajan, "Distributed Anomaly Detection Using Autoencoder Neural Networks in WSN for IoT," 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1-6.doi: 10.1109/ICC.2018.8422402