Anomaly Detection using Autoencoders

PRECRIME SUMMER PROJECT EXERCISE

1 Objective

Train an autoencoder-based anomaly detector to detect anomalous classes in the the famous MNIST dataset [3].

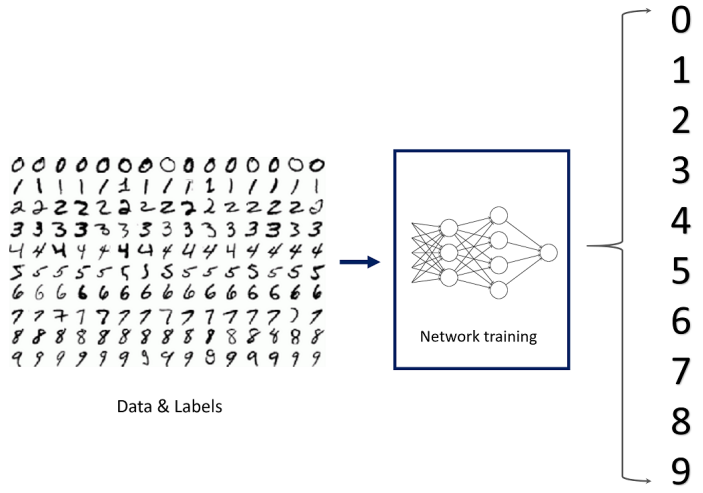

Figure 1: Exercise description: MNIST Digit Recognition.

2 Dataset

MNIST [3] (Modified National Institute of Standards and Technology database) is large database of 70,000 handwritten digits that is commonly used for training various image processing systems. MNIST has a training set of 60,000 examples, and a test set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size black-white image.

3 Exercise

Implement a Python program using the Tensorflow's Keras API [1] that performs the following task:

a. Download the MNIST dataset. Your task is to split the dataset so as to retain one of the classes (i.e., digit) unknown at training time. For example, the training set could be composed of instances of only digits "0-8", where digits representing "9" are left out as anomalies (nominal set = [0, 1, 2, 3, 4, 5, 6, 7, 8], anomaly set = [9]).

b. Define an autoencoder model in Keras, that will be used as anomaly detector. Some examples can be found online [2], but the choice of the architecture is free. To complete this task, you are also required to configure the training and evaluation process, by:

   - specifying a *loss function* that measures how closely the model's predictions match the target classes.
   - specifying a *method to optimize* such loss value during training.

c. Compile and train the model *on the training set*.

d. Evaluate the trained anomaly detector on a test set composed only of nominal instances and on an anomaly set composed only of anomalous instances. The reconstruction error of the autoencoder will be used as an anomaly score. Select an appropriate threshold.

4 Output

The program must print/plot:

- the number of detected anomalies in the anomaly set (true positive rate)
- the number of undetected anomalies in the anomaly set (false negative rate)
- the number of incorrectly anomalies detected in the test set (false positive rate)
- the number of correctly undetected nominal cases in the test set (true negative rate)

References

[1] Chollet, F. et al. (2015). Keras. https://keras.io.

[2] Chollet, F. et al. (2016). Keras. Building autoencoders in Keras.

[3] LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.