Raunak Sarbajna and Christoph F. Eick

COSC 3337 *“Data Science I”* Fall 2023

**TASK 5**: Using Neural Networks and Autoencoders for Outlier Detection

Individual Task

Task 5 Due: Saturday, Dec 2, 11:59p (electronic Submission)

Last update: November 29, 2023

**Learning Objectives**:

1. Getting some exposure to neural network learning and neural network architectures
2. Learn what autoencoders are and how they can be used for outlier detection
3. Learn how to use deep learning and generative models, such as VAE
4. Learn how to assess the quality of various models

|  |  |
| --- | --- |
| **A: Variational Autoencoder** | **B: Normal Autoencoder** |
| C:\Users\mdmah\Google Drive\UH\TA\AI\Spring 2023\Picktures\VAE.png | C:\Users\mdmah\Google Drive\UH\TA\AI\Spring 2023\Picktures\autoencoder_header.png |

**Figure 1:**

**A**: A sample Variational Auto-encoder. The VAE contains one encoder and one decoder part. Encoder starts from x,h and ends in z=(σ + μ). [(σ + μ) learns latent representation or key features of the images]. Decoder starts from z=(σ + μ) to h2 and ends in x2. Decoder utilizes learned important represntation from z=(σ + μ) and tries to regenerate the image in x2.

**B**: A sample normal auto-encoder. A normal autoenoder contains only a fully connected layer z instead of a pair of layers (σ + μ) to learn the hidden representation.

In this project we will use the Wikipedia Web Traffic dataset [3] and experiment with auto-encoders and variational autoencoders to detect outliers. The Two Jupiter notebooks provided contains the architecture and process of using them, and instructions on how to work.

**Task 5 Subtasks:**

1. Preprocessing:
   1. Visualize all the languages the provided wikipedia time-series dataset.
   2. Select a language by performing the operation

*<Your UH ID> mod 5*. ([google link](https://www.google.com/search?q=1111111+mod+5))

* 1. Perform scaling
  2. Split the data into training, validation, and testing sets.

1. Use a basic autoencoder to find anomalies within a single language of the wikipedia traffic dataset. Use the Notebook given in reference [1]. Perform the following tasks:
   1. (\*\*\*) The given model has a three layer architecture for both the encoder and decoder. Can you modify the architecture into a four (or more) layer format?
   2. (\*\*)Compare the results based on their:
      1. Optimal loss after the model is fully trained, and
      2. Visually inspecting the anomalies detected.

Is your model performing better than what was provided in the notebook? Give reasons why or why not.

* 1. (\*\*) Three different systems for finding the anomaly threshold are provided in the notebook. One uses quantiles, one usese percentiles, and one uses the training dataset instead of the testing.
     1. Give reasons for which is best.
     2. Create your own threshold that’s three standard deviations from the mean.
  2. (\*) Three different plots are provided for each of the thresholding methods. Explain what you see. Figure out a method to visualize the *anomalies* variable.

1. Use a Variational Autoencoder to find anomalies within the entire wikipedia traffic dataset. Use the Notebook given in reference [2]. Perform the following tasks:
   1. (\*\*\*) Recall that in a VAE, the encoder first learns the parameters of the distributions that make up the latent vector, and then generates latent vector **z** by sampling from this distribution. The given model uses the reparameterization trick. What happens if:
      1. You add in an additional random term into the sampling function?
      2. You add in more layers before the sampling function?
   2. Compile the model. Modify the following parameters: learning rate, epochs, batch size. Try to obtain the best performace. Is your model performing better than what was provided in the notebook? Give reasons why or why not.
   3. (\*\*) Explain the threshold system used in the notebook. Can you come up with a better one?
   4. (\*) Examine the Principle Component Decomposition figures in the notebook. Why are anomalies close to the center of the figure? What can be done to improve the performance of the model

**SUBMISSION GUIDELINES**

Write your answers *within the Jupyter Notebooks*. Write your Name+UH ID at the top. *Save/print the notebooks as pdfs*. Submit them directly.

**References:**

1. [**https://github.com/RaunakDune/UH-COSC-3337/blob/ed8e1c66662355f9da3299a8d3c8e61e3c61ed38/Wiki%20Anomaly%20Det.ipynb**](https://github.com/RaunakDune/UH-COSC-3337/blob/ed8e1c66662355f9da3299a8d3c8e61e3c61ed38/Wiki%20Anomaly%20Det.ipynb)
2. [**https://github.com/RaunakDune/UH-COSC-3337/blob/ed8e1c66662355f9da3299a8d3c8e61e3c61ed38/Wiki%20Anomaly%20Det%20VAE.ipynb**](https://github.com/RaunakDune/UH-COSC-3337/blob/ed8e1c66662355f9da3299a8d3c8e61e3c61ed38/Wiki%20Anomaly%20Det%20VAE.ipynb)
3. **https://github.com/RaunakDune/UH-COSC-3337/blob/1885fdc6c1bfc12b0e9cf64460ffe2bbc815469d/wiki\_lang\_summ.csv**
4. <https://towardsdatascience.com/hands-on-anomaly-detection-with-variational-autoencoders-d4044672acd5>
5. <https://keras.io/examples/generative/vae/>