EEL6935 – SPRING 2020 – DEEP LEARNING
Homework Assignment 4
Denoising Autoencoders

In this final homework assignment, you will implement one of the applications we have discussed in class when we were talking about autoencoders: noise removal. You will use the popular MNIST dataset in training your autoencoder. MNIST has a separate training and testing set which consists of 60,000 and 10,000 samples respectively.

Preparation: We will use the popular MNIST dataset for this assignment. This dataset consists of 70000 28x28 greyscale images representing digits. The training and testing sets are fixed: there are 60000 training images and 10000 test images with corresponding labels. To prepare for this assignment: first you will need to download the MNIST dataset. The official website for dataset is given below:

http://yann.lecun.com/exdb/mnist/index.html

You can use the link above to download the original dataset. **The code you will prepare should assume that dataset files are located in the same folder as your notebook.** If you are using Keras, you can also import the MNIST dataset through Keras, see the documentation for details.

Objective: We want to remove noise from noisy images by passing them through an autoencoder.

Data Preparation: **Do not change the sample order of the dataset.** You need to build an autoencoder model with regards to following instructions:

- You will preprocess the data by dividing each pixel by 255 (the maximum pixel value) such that each pixel is mapped to the interval of [0-1]. You can accomplish this by using the following command in Python:

  ```
  x = x.astype('float32') / 255.0  # x is your input image
  ```

- You will add Gaussian noise to the MNIST dataset. You can accomplish this by using the following command in Python:

  ```
  x = x + np.random.randn(*x.shape) * noise_factor
  x = x.clip(0.0, 1.0)
  ```

  The first command adds random Gaussian noise scaled by a factor (noise_factor) to each pixel in the image. The second command clips the image so all pixel values are still between [0 1].

Network Definition: You need to build an autoencoder model with regards to following instructions:

- You will have a 784 x 32 x 784 autoencoder network. 784 neurons in the input layer (MNIST image size is 28x28 = 784), 32 neurons in the hidden layer (code size) and 784 neurons in the output layer (same size as the input layer as is standard for autoencoders)
- You will use rectified linear unit (RELU) activation function for the hidden layer.
- You will use the sigmoid activation function for the output layer.
- Size of the output images should be same as the input image size.
- You will then train your autoencoder with **using the noisy images as input and clean images as output.** You can use batch size of 128 with approximately 10 or so epochs

If you need a refresher on what these parameters were and how to define them, please refer to the previous lectures and homework assignments.

Goals:

1 **Prepare noisy images with a noise factor of '0.1'.** Test your trained autoencoder model with the **first 10** images of the **test set.** Display the output of the autoencoder (cleaned images) versus the noisy images **on a single figure**, similar to Figure 1 below.

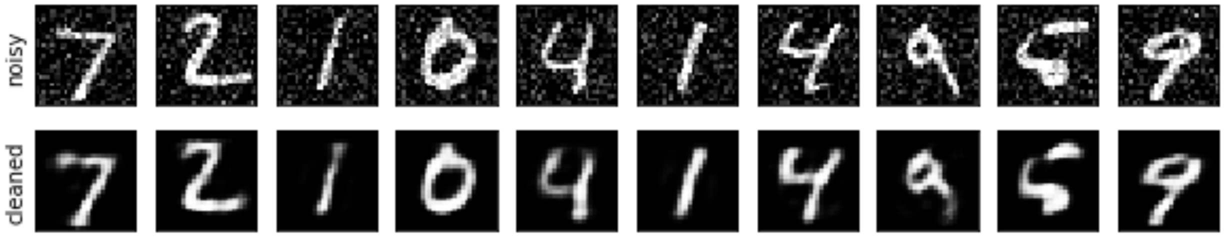

Figure 1: Example Deliverable Figure

2 Repeat goal **1** for a **noise factor** of **'0.5'**. Display results as a separate figure.
3 Repeat goal **1** for a **noise factor** of **'1.0'**. Display results as a separate figure.
4 Comment on your findings. **You can simply put your comments at the end of your script.**

Deliverables: This assignment has one deliverable:

1) Your Jupyter notebook which includes all the results, comments, and scripts. Make sure **all your results and comments are clearly visible** and **avoid printing or displaying anything that was not requested in the homework**.

IMPORTANT INFORMATION:

Remember what we discussed in class about the HW assignments:

- ⇨ **EXTREMELY IMPORTANT:** You are to submit a **single** Jupyter notebook file, which must include all the results, comments, and scripts. **All your results, plots and comments must be clearly visible and legible.**
- ⇨ Your code must run **without modification** and produce all, **and only**, the required results and figures asked for in the **Goals** section.
- ⇨ Your code should read the dataset files with proper, relative path. That means code and files can only be run if the folder structure is exactly as defined in the preparation section.
- ⇨ Your code and assignment should be saved as follows: last_name_first_initial_hwX.ipynb
  - ○ **For example, for me it would be: uysal_i_hw4.ipynb**

All questions regarding this assignment must first be sent to maktukmak@mail.usf.edu.