Task 4: Auto-Encoders Mahin

Submission Deadline Task4: Th., April 4, 11:59pm.
Last Updated: March 5, 11a.
Weight: Task weight: 30points

Learning Objectives:

1. Learn to use deep learning and generative models such as VAE
2. Learn to use classifiers
3. Learn differn tools to create different deep learning models
4. Learning how to interpret quality of models

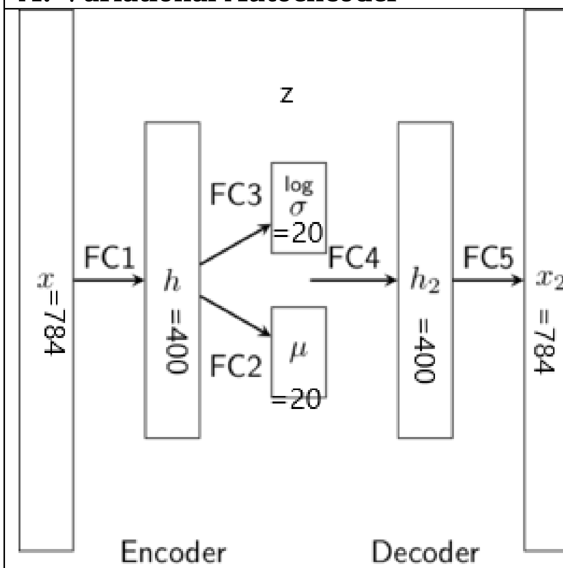
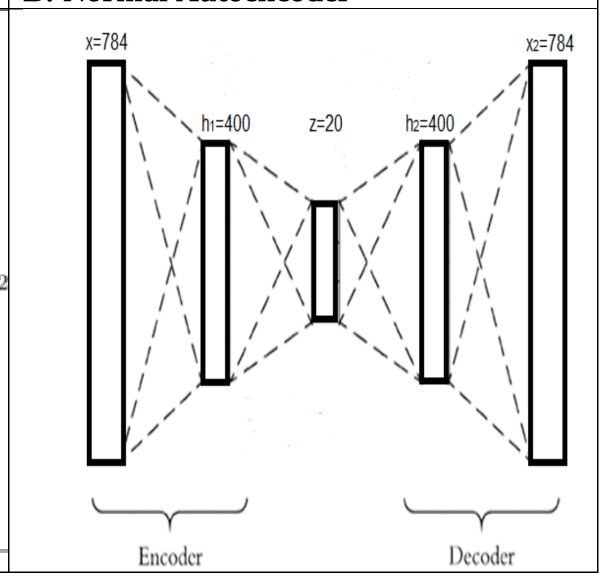
A: Variational Autoencoder	**B: Normal Autoencoder**
	

Fig. 2.: Variational and Normal Autoencoder Architecture
A: Variational Auto-encoder Architecture: 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 h_2 and ends in x_2. Decoder utilizes learned important represntation from z=(σ + μ) and tries to regenerate the image in x_2.
B: Noraml Auto-encoder Architecture: 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 Fashion MNIST computer vision digit dataset and experiment with auto-encoders such as Variational Auto-encoder(VAE) and simple autoencoder. The Jupiter notebook provided contains a VAE architecture and process of downloading the dataset. (total: 30 points)

Task 4 Subtasks:

1. **1.** Learn latent features from the Fashion MNIST dataset. Use the model given in reference [1]. Perform the following tasks: (total 10 points)
    - **a.** The given model has a three layer architecture for each encoder and decoder part. Can you modify the architecture into a four layer format. In this task, you need to convert encoder part into (x, h_1, h_2, $z=(\sigma + \mu)$)= (784*400*100*20) and decoder part into ($z=(\sigma + \mu)$, h_3, h_4, x_2)= (20, 100, 400, 784). Finally you need to compare the results based on their:
        - **i.** Optimal loss after the model is fully trained, and
        - **ii.** Visually inspecting the output they generate using the images they generate and reconstruct. You can use plot_generation() and plot_reconstruction() function from the notebook.

      Based on optimal loss and visual inspection write down your opinion which model is better and also try to give an explanation why a model is giving good performance over another.

    - **b.** Take the base three layer architecture and check the performance of the model for six different configuration, where h_dim and z_dim is changed into following patterns: [(400,50), (400, 10), (400, 30), (300, 35), (300, 5), (300,40)](Note: First one is the base architecture). Perform the same type of comparison you have done in task **a** using optimal loss of the model and visual inspection and write down your opinion which model is better and also try to give an explanation why a model is giving good performance over another. (total 10 points)

    - **c.** Take the <u>best architecture</u> from b and convert it into a normal auto-encoder (figure 1.b)[2], e.g. replace z_dim such a way that it will be single layer. As noraml autoencoder and variational auto-encoder have very different way of loss calculation, you need to modify loss function too. Now Perform the same type of comparison you have done in task **a** using optimal loss of the model and visual inspection and down your opinion which model is better and also try to give an explanation why a model is giving good performance over another. (total 10 points)

Deliverables:

1. A Jupyter notebook with your code and analysis. Your notebook should use markdown and should contains:
    a. Description of the code or changes you made to the code for every task in the markdown (before each code section, also comment properly within the code) (code + description 5 points for each task)
    b. You should describe the loss comparison using markdown after each task, e.g. try to answer the task a, b, c using markdown in the notebook after completing each task (2 points for each task)

c. Similar to b, try to explain visual comparisons using markdown after each task, e.g. try to answer the task a, b, c using markdown in the notebook after completing each task (2 points for each task)

2. A report that will be pdf generated from the markdown. But remember to do following changes (1 points for each task)
   a. Add discussion of tasks you performed but do not include code in the report
   b. All your comparison description. Remember to add the model outputs before each comparison.

References:

1. https://github.com/dataflowr/notebooks/blob/master/HW3/VAE_clustering_empty.ipynb
2. https://www.analyticsvidhya.com/blog/2021/06/complete-guide-on-how-to-use-autoencoders-in-python/

Task 5: Learning and Using Diffusion Models *Raunak*

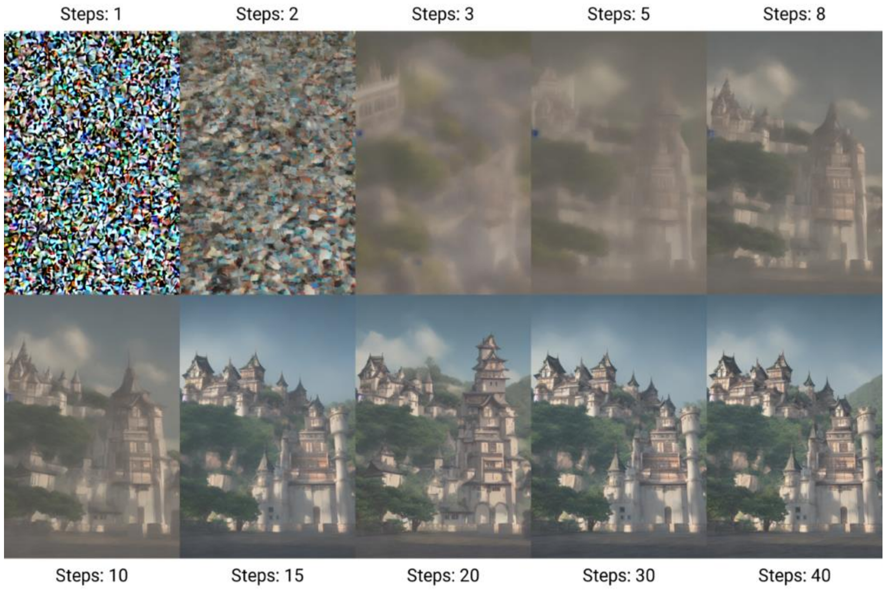


Fig. 3: The Denoising Process used by *Stable Diffusion*

Submission Deadline Task5: Thursday, April 18, 11:59p
Last Updated: March 5, 10a
Tentative Task Weight: 30-35 points

The Task5 Specification will be added by March 26, 2024 the latest.