Md. Mahin and Christoph F. Eick

**Task 4: Deep Learning Centering on Generative Models and Auto-Encoders**

Individual, not Peer-reviewed Task

First Draft

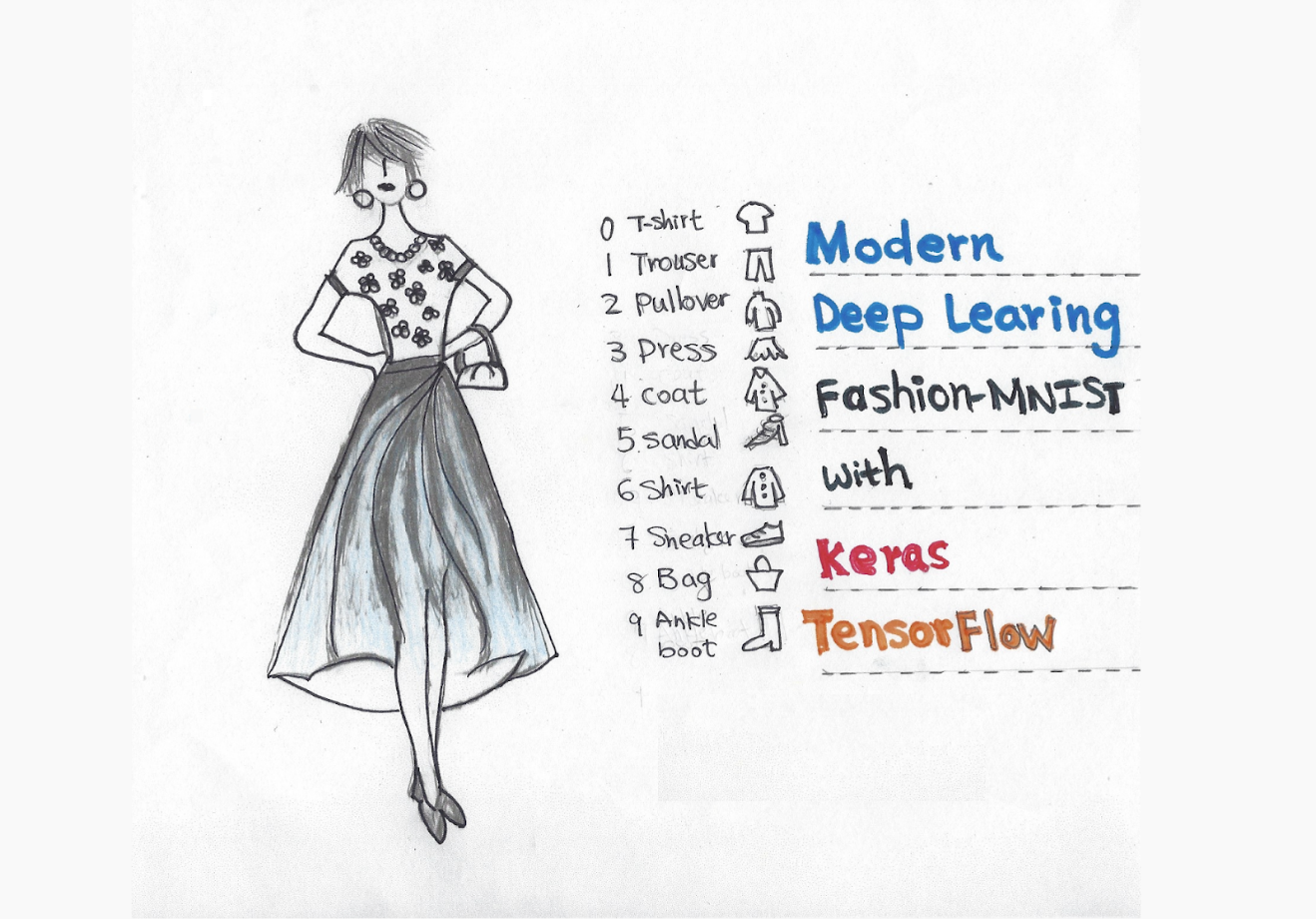


Fig. 1: Training a model to classify images is the archetypal neural network task

Submission Deadline Task4: Mo., November 11:59p in MS Teams

Last Updated: October 11, 3p

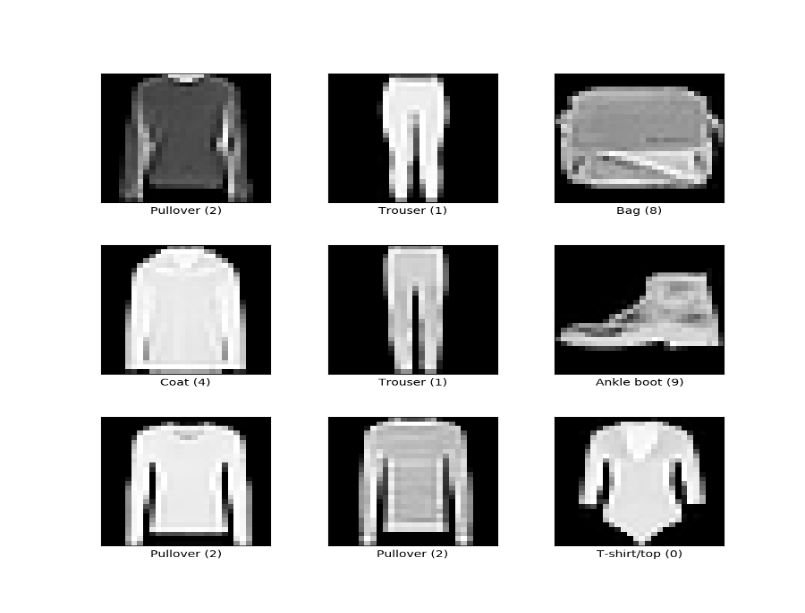
Weight: Task weight: T4=20 points

**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

**Dataset:**

Fashion MNIST is a popular image classification dataset containing 70,000 grayscale images of 10 different clothing items (T-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, ankle boot). We will use this dataset to train our neural network to identify these clothing items. Below some samples are given from Fashion MNIST dataset.:



**Tasks:**

|  |  |
| --- | --- |
| **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 3: AutoEncoder Architectures

**Figure3:**

**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 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 code toof download the dataset.

**Task 4 Subtasks:**

1. Learn latent features from the Fashion MNIST dataset. Use the model given in reference [1]. Perform the following tasks: (total 7 points)
   1. The given model has a three layer architecture for each encoder and decoder part. Can you modify the architecture into a five layer format. In this task, you need to convert encoder part into (x, h1, h2, z=(σ + μ) )= (784,400,200,100,20) and decoder part into (z=(σ + μ), h3, h4, x2 )= (20, 100,200, 400, 784). Stop the model training before the model starts overfitting. Finally you need to compare the results based on their:
      1. Optimal loss after the model is fully trained (you need to show model trained untill it reached overfitting point), and
      2. 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.

* 1. 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: [(400,40), (200,50), (400, 70), (300, 30), (250, 80), (300, 5)](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 6 points)
  2. Take the best architecture from a 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 (careful with KL divergence). 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 7 points)

## **Deliverables:**

1. A Jupyter notebook with your code and analysis. Your notebook should use markdown and should contains:
   1. 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)
   2. 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)
   3. 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)

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

**References/ Resources:**

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/>
3. Fashion MNIST dataset: <http://pytorch.org/vision/stable/generated/torchvision.datasets.FashionMNIST.html>

* TensorFlow tutorials: <https://www.tensorflow.org/tutorials>
* PyTorch tutorials: <https://pytorch.org/tutorials/>