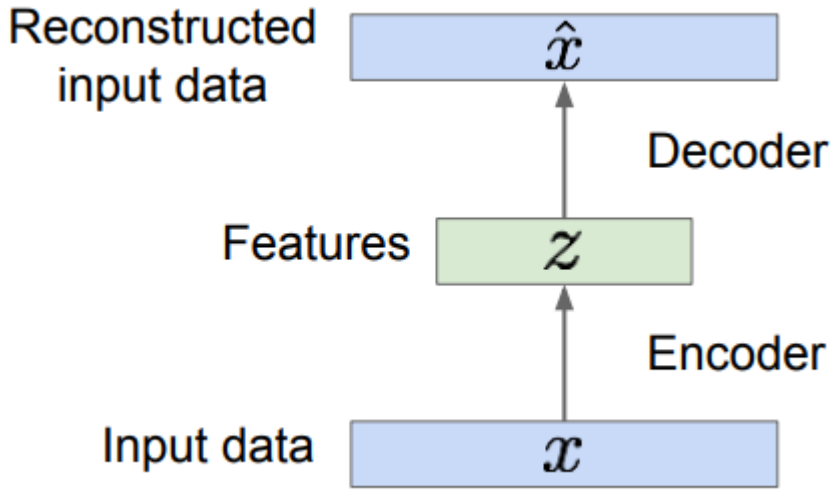
Assignment 8 – CMPUT 328

Semi-supervised learning with Autoencoder on MNIST:

Introduction: Implement a Semi-upervised learning algorithm using the feature vectors of an autoencoder on MNIST dataset by adding a classification branch to an autoencoder.

An autoencoder is a special type of neural networks architecture that comprises of two parts: an encoder and a decoder. The encoder tries to compress the original input into a latent-space representation that has lower dimension than the original input. This usually manifest as multiple consecutive fully-connected/convolutional layers that has smaller output dimension than input dimension. The decoder does the opposite of the encoder, that is to try to reconstruct the original input from the latent-space representation via multiple consecutive layers that have larger output dimension than input dimension.
The autoencoder is then trained to minimize the difference between the original input and the reconstructed input via L2/L1… loss.



The picture above shows general structure of an autoencoder. Read more about autoencoder: https://towardsdatascience.com/deep-inside-autoencoders-7e41f319999f

In this assignment, you will design an autoencoder with a classification branch. The classification branch in this assignment consists of one or a few fully connected layers similar to the last 3 layers of LeNet. The classification branch will start from the features z of the autoencoder.

The new loss will be a combination of the L2 reconstruction loss for the autoencoder and the cross entropy classification loss for the classification branch. There is one catch, though. In the semi-supervised learning problem, only a part of your data has classification labels on it. For example, in this assignment, only 20% of images have classification labels for training. In this case, the loss function will have no cross entropy classification loss for images that are unlabeled. The loss function for each image will look like this:
$L = ||\hat{x} - x||^2 + C * CrossEntropyLoss(F(z), y)$

The first term is the reconstruction loss, the second term is the classification loss. x is the input image, $\hat{x}$ is the reconstructed image of the autoencoder, z is the encoded feature vector, $F(z)$ is a shallow

neural network that classifies the input image x, y is the classification label. C is a value that depends on whether the image is labeled or not. $C = 0$ when image is unlabeled and $C = C_0$ otherwise. C_0 is a positive value that control the weight of the two losses. In this assignment, $C_0 = 1$.

Your task in this assignment is to write the code to define the autoencoder, including the classification branch. You don't have to write training code. The split of labeled/unlabeled data have already been handled also. However, you can change hyperparameters for the training inside the **run()** function of **autoencoder.py**.

Task: Complete the function **AutoEncoder()** inside the file **autoencoder.py** to return the reconstructed input data and the logits of the classification branch. See the TODO inside that file for more information. There are no architecture requirements or guidelines.

Mark:
- Coding (5 mark): Your in the coding part will scale linearly with your classification accuracy, starting from 87.5% (0 mark) and cap at 97.5% (50 mark)
- Presentation (2 mark): Present your work to the TA in your lab session.

Deadline: November 9th, 23:55 pm