ECE 685D HW4

Submission Instructions:

1. Upload your Jupyter Notebook (.ipynb file).
2. Export all outputs and necessary derivations as one or multiple PDF files and upload them.

1 Problem 1: Sparse Encoding for Image Denoising (30pts)

Considering an auto-encoding with the following spec:

$$\min_{\mathbf{D}} \frac{1}{T} \sum_{t=1}^{T} \min_{\mathbf{h}^{(t)}} \frac{1}{2} ||\mathbf{x}^{(t)} - \mathbf{D}\,\mathbf{h}^{(t)}||_2^2 + \lambda ||\mathbf{h}^{(t)}||_1$$

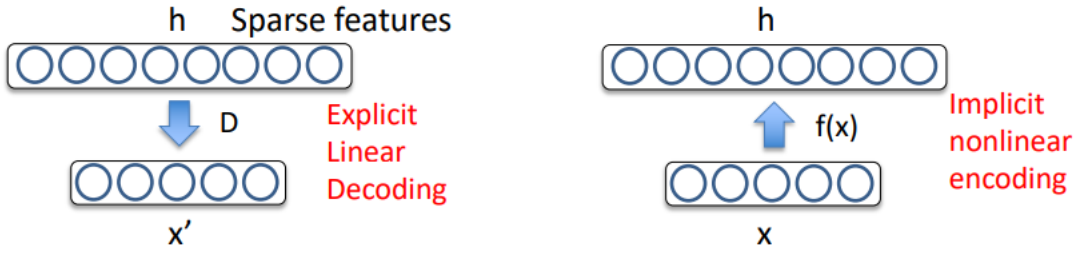

Figure 1: An over-complete auto-encoder with non-linear encoding and linear decoding modules. Lasso (L1) regularization is used to enforce meaningful feature learning

We will use MNIST in this problem. Let the dimension of $\mathbf{h}$ be 1.5 times that of your input and $f(x)$ defined by your own conjecture. Let X be a batch sampled from MNIST, taking $X + Gaussain(0, \sigma I)$ (σ of your choice) as the model input and our goal is to obtain its output $\hat{X} \approx X$. Use the default training split for your auto-encoder. You will plot

1. 5 input-output pairs $(X + Gaussain(0, \sigma I), \hat{X})$ using the data in the test set.
2. The top 5 dictionary vectors in $\mathbf{D}$ whose corresponding intensity $|\mathbf{h_i}|$ are the largest.

Hint: using MLP rather than CNN for this problem may make it easier in determining the "more important" dictionary elements.

2 Problem 2: Modified Probablistic PCA on MNIST (30pts)

In this problem, you will apply PPCA to the MNIST dataset. Your goal is to reduce the dimensionality of the images from 784 dimensions (28x28) to $n \in \{2,4,8,32\}$ dimensions using PPCA and visualize the results. In this problem, we no longer assume the variance to the the same across all dimensions (i.e., $\sigma_1^2 \neq ... = \sigma_n^2 = \sigma^2$ is not guaranteed) Instead of EM or closed-form solution, use SGD/ADAM etc. to find your own parameter. including $\sigma_i, ..., \sigma_n$.

After training your model, you will sample latent representations

$$z \sim N(0, I)$$

and check if you can also generate new samples with PPCA with the model weights you obtained through training. You may find information in this link helpful `https://www.tensorflow.org/probability/examples/Probabilistic_PCA`

3 Problem 3: Autoencoder for Pretraining (40pts)

You will use CIFAR10 in this problem. You will build a autoencoder to learn the latent representation for the CIFAR10 dataset. Please follow the steps below:

1. Define you auto-encoder architecture (consider using 3-4 CNN or Dense layers for the encoder)
2. Use your randomly initialized auto-encoder to extract feature from the CIFAR10 data, which gives a "random projection" of you data to a lower dimensional space. Save these latent features as H_0
3. Train your auto-encoder with the CIFAR10 dataset on MSE loss.
4. Use your trained auto-encoder to extract feature from the CIFAR10 data again. Save these second set of latent features as H_1
5. Train two classifiers with H_0 and H_1 being the input respectively with with the label corresponding to the original input X from CIFAR-10.

Now, compare the classifier performance on the two versions of latent representation. Discuss your results. Note: If you are encountering any constraint in computing power, you may down-sample your image to a lower resolution.