Cornell University
CS 5788: Introduction to Generative Models
Spring 2026 Instructor: Andrew Owens

Problem Set 2: Latent variable models

Posted: Friday, February 13, 2026 **Due:** Tuesday, March 10, 2026

Please submit your written solution to Gradescope as a `.pdf` file. Please convert your Colab notebooks to PDF. For your convenience, we have included the PDF conversion script at the end of the notebook. Last updated: 2/17/26.

We recommend editing and running your code in Google Colab, although you are welcome to use your local machine instead. Please note that problems marked optional will not be graded.

Problem 2.1 *Variational Autoencoder (VAE)*

We will implement a variational autoencoder and manipulate images in latent space. The notebook `VAE.ipynb` will walk you through the implementation using the MNIST digit dataset.

Reconstruction examples:

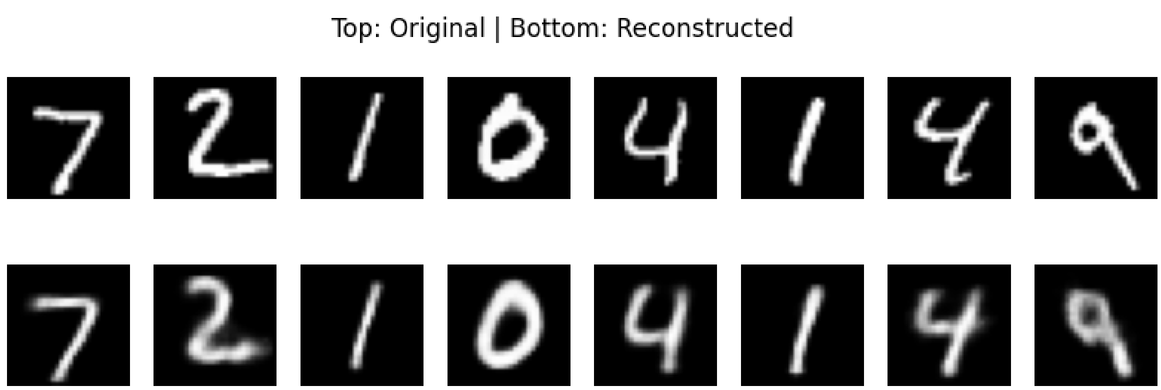

Figure 1: VAE-reconstructed samples.

Generation examples:

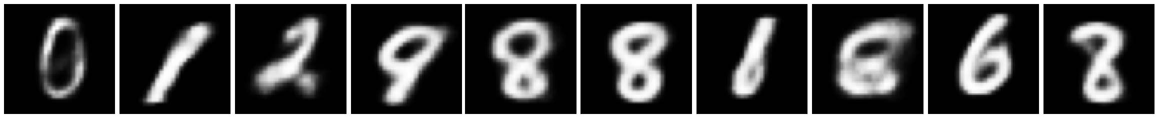

Figure 2: VAE-generated samples.

Problem 2.2 *Cycle-Consistent Adversarial Networks (CycleGAN)*

We will use CycleGAN to perform unpaired image-to-image translation, translating apples to oranges and vice versa. The notebook `CycleGAN.ipynb` will walk you through the implementation of a CycleGAN on the "apple2orange" dataset.

`Warning`:

- This takes a long time to train! In our impelmentation, it took 6-8 hours on a T4 GPU, and less than an hour on an A100 GPU (available on Google Colab Pro, which is currently free for Cornell students).
- Since we changed the model to speed up training (versus the original paper), the results won't be perfect. We show some examples in Fig. 3 and Fig. 4 for reference.

Translation examples:

Figure 3: CycleGAN translation example 1.

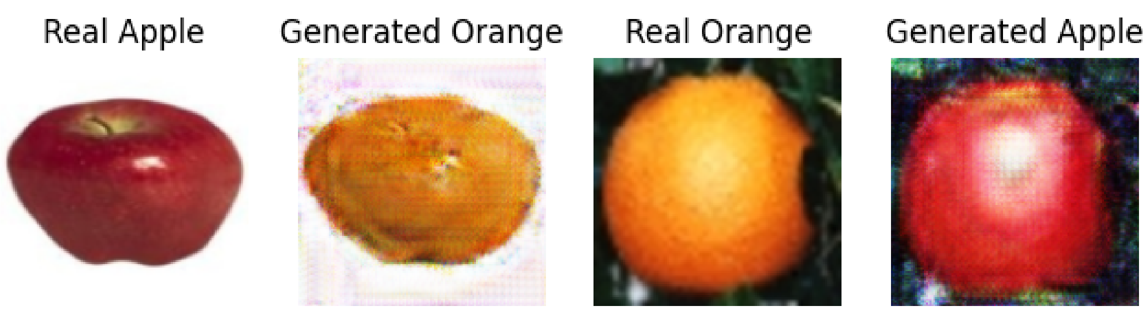

Figure 4: CycleGAN translation example 2.

Problem 2.3 *Written problems*

Please turn in your answers (either written or typeset) as a separate PDF.

(a) In class, we derived the *evidence lower bound* (ELBO), which we used to train a variational autoencoder:

$$\log p_\theta(\mathbf{x}) \geq \mathcal{L}_{ELBO} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - D_{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})), \tag{1}$$

where $\log p_\theta(\mathbf{x})$ is the log likelihood of the model and $q_\phi(\mathbf{z} \mid \mathbf{x})$ is defined by the encoder.

(i) Explain why maximizing $\log p_\theta(\mathbf{x})$ is similar to training an autoencoder.

(ii) Suppose that we eliminated the second term (the D_{KL} term) and optimized only the first term. How would this differ from training a standard autoencoder?

(b) In class, we used Jensen's inequality to obtain a lower bound for the log likelihood (namely, Equation 1).

(i) Instead of obtaining a lower bound, show that the following *equality* holds:

$$\log p_\theta(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})}\right]\right] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{p_\theta(\mathbf{z} \mid \mathbf{x})}\right]\right]. \tag{2}$$

Hint: This will be brtuam similar to the derivation that we did in class, but there is no need for applying Jensen's inequality. Instead, perform another simple algebraic manipulation.

(ii) Show that first term in Equation 2 is equivalent to the ELBO (Equation 1).

(iii) When is $\mathcal{L}_{ELBO}$ (Eq. 1) a tight bound for $\log p_\theta(\mathbf{x})$?

(c) In the standard GAN formulation, the discriminator maximizes the following objective function:

$$V_L(D) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}\left[\log D(\mathbf{x})\right] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}\left[\log(1 - D(G(\mathbf{z})))\right] \tag{3}$$

Suppose that we have the discriminator maximize a *squared loss* instead:

$$V_S(D) = \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}\left[(D(\mathbf{x}) - b)^2\right] + \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}\left[(D(G(\mathbf{z})) - a)^2\right] \tag{4}$$

for some constants $a, b \in [0, 1]$. We'll use this formulation in the CycleGAN problem (Problem 2.2). The generator is simultaneously trained to minimize a closely related quantity:

$$V_G(G) = \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}\left[(D(G(\mathbf{z})) - c)^2\right] \tag{5}$$

for another constant $c \in [0, 1]$. For example, we might set $a = 0$, $b = 1$, and $c = 1$.

(i) Give a plain language description of this objective function, and what the "goals" of the generator and discriminator are when using it.

(ii) In class, we derived the optimal discriminator when it is trained on V_L. Derive the optimal discriminator D^* for V_S.

Credits. This problem set was written by Yen-Yu Chang, Yiming Dou, and Andrew Owens. The variational autoencoder problem is based on a problem set by Justin Johnson.