Introduction to Deep Learn, Fall 2017

Homework on Unit 3: 100 Total Points

Due: November 6th, 11:59PM

Submissions required electronically through Sakai. Non-code is required in pdf format, and code is required in a Jupyter notebook.

The purpose of this homework is to get a deeper understanding of the generative models, autoencoders and Generative Adversarial Networks (GANs), and practical intuition about its components, architecture, and their impact on performance.

Problem 1: Variational lower bound (24 points)
For a latent variable model with observed variable x and latent variable z, we can write the log-marginal likelihood as

$$\log p(x) = D_{KL}(q(z|x)||p(z|x)) + L(q;x),$$

where

$$D_{KL}(q(z|x)||p(z|x)) = \int q(z|x) \log \frac{q(z|x)}{p(z|x)} dz.$$

Show that:

a) $D_{KL}(q(z|x)||p(z|x)) = 0$ if and only if $q(z|x) = p(z|x)$.
b) The equality above holds for $L(q;x) = \int q(z|x) \log \frac{p(x,z)}{q(z|x)} dz$.

Problem 2: Interpolation between images using GAN (24 points)
Trained GANs can generate images from a simple random distribution (*e.g.*, Gaussian or uniform). We can in principle "animate" the transition of one image into another by interpolating between them using linear interpolation in the space of the simple random distribution, rather than image space.

a) Train a GAN on MNIST using class code as starting point.
b) Generate some images, say 10, by first drawing samples, ϵ, from the simple distribution and then feeding them through the generator.
c) Visualize the images and pick any two (preferably different digit labels), identify their inputs and set them to ϵ_{start} and ϵ_{end}.
d) Generate 10 steps between ϵ_{start} and ϵ_{end} by linear interpolation.
e) Generate 10 images using $\epsilon_{\text{start}} = \epsilon_1, \ldots, \epsilon_{10} = \epsilon_{\text{end}}$ as inputs to the generator and visualize them.

Problem 3. Interpolating with variational autoencoders (24 points)
We can similarly interpolate between images using Variational Autoencoders (VAE).

a) Train a variational autoencoder on MNIST using class code as starting point.
b) Using the encoder, generate latent representations for 2 images (z_{start} and z_{end}) matching the digit labels selected in Problem 2c.
c) Generate (via decoding) and visualize images from 10 steps $z_{\text{start}} = z_1, \ldots, z_{10} = z_{\text{end}}$ using linear interpolation.

d) How does the quality of the VAE generated images compare to those from GAN?
e) Do you observe any differences in the smoothness of the interpolation from both models?

Problem 4: Conditioning on class labels (24 points)

Trained GANs can generate images from simple distributions, however, we have almost no control over the generated images. Assuming we want to generate images from specific digits, we can modify GAN to generate images conditioned on the digit label, *i.e.*, a conditional GAN.

a) Modify the code from Problem 2 to take as input a vector 10 times larger.
b) Modify the code from Problem 2 to take as input the concatenation of ϵ (random vector) and a one-hot vector encoding of the labels.
c) Train the GAN.
d) Generate and visualize 10 images from each digit by setting the labels accordingly as inputs to the generator and ϵ at random from its distribution.

Problem 5: Bookkeeping (4 points)

a) How many hours did this assignment take you? (There is No correct answer here, this is just an information gathering exercise)
b) Verify that you adhered to the Duke Community Standard in this assignment (https://studentaffairs.duke.edu/conduct/about-us/duke-community-standard), *i.e.*, write "I adhered to the Duke Community Standard in the completion of this assignment"