ECE 157B/272B Homework 3: Variational Autoencoder

Due date: Friday, February 11th, 2022, 11:59PM

Introduction

In this homework, we will continue working with convolution neural nets, but use them on application of data encoding and decoding, and we will also have a glance at data generation.

In this homework, we will explore Autoencoder model, more specifically, Variational Autoencoder. From a very abstract view, Variational Autoencoder maps the input data into the parameters of a probability distribution, such as the mean and variance of a Gaussian distribution. This approach produces a continuous, structured latent space, which is useful for data compression and image generation. As usual, you will use tensorflow library to build a complete pipeline for data preprocessing, model building and training, and result evaluation. You will also utilize free GPU/TPU resources available on Goolge Cloud for speeding up the training.

Data Set

We will use tf.keras.datasets module to download and load our datasets. We will be using two datasets: MNIST and CIFAR-10.

MNIST Dataset

The MNIST dataset has a training set of 60000 examples, and a test set of 10000 examples. The dataset has ten classes, each associates with a digit. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image. [2].

Here's the code snippet for loading the dataset:

```
(train_images, training_labels), (test_images, test_labels) = \
        tf.keras.datasets.mnist.load_data()
```

CIFAR-10 Dataset

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The class labels are airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. [1].

Here's the code snippet for loading the dataset:

```
(train_images, training_labels), (test_images, test_labels) = \
        tf.keras.datasets.cifar10.load_data()
```

Problem Formulation

You must perform the following guiding tasks and write down answers to the following questions as prompted in the notebook:

1 MNIST VAE

1. (2 pts) Preprocess the data set. Show one image for each class from the MNIST training dataset.

2. (4 pts) Complete the code for defining the decoder net in `CVAE` class (2pts). What is the input shape to the first convolution transpose layer in the decoder? Explain how you decided this input shape. Show the calculation in terms of the `img_size`, `num_convolutions`, and `strides` used in the deconv layers (2pts).

3. (2 pts) What is the last layer of the decoder net in `CVAE` class? Explain how you decided the number of filters for this layer.

4. (2 pts) Read the function definition of `log_normal_pdf`. We know the probablity density function (PDF) of a normal distribution is:

$$f(x_i; ,^2) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

   where μ is mean, and σ is standard deviation. If we take log of the above equation, we get the log PDF of the normal distribution:

$$log(f(x_i; \mu, \sigma^2)) = -ln - \frac{1}{2} ln(2\pi) - \frac{1}{2}(\frac{x-\mu}{\sigma})^2$$

   Note that `log_normal_pdf` is given a sample x, the mean μ, and the log variance *logvar* of the distribution. Please derive the formula in `tf.reduce_sum()` in `log_normal_pdf`'s function return. (Hint: variance is the square of σ).

5. (2 pts) As described in this blog, to generate a sample for the decoder during training, you can sample from the latent distribution defined by the parameters outputted by the encoder, given an input observation, i.e, $\tilde{z} \sim q(z|x)$. However, this sampling operation creates a bottleneck because backpropagation cannot flow through a random node.
   Refer to `reparameterize` function in `CVAE` class. Explain what is the *Reparametrization Trick* and how it enables the backpropagation.

6. (2 pts) Recall from class that VAE has two optimization objectives:

   (a) Maximize the log likelihood $\log p(x|z)$, or minimize the reconstruction error.

   (b) Minimize the KL divergence of the approximate from the true posterior: $D_{KL}(\log q(z|x)||\log p(z))$.

   Describe which part in function `compute_loss` describes objective (a) and which part describes (an estimation of) objective (b)?

   Hint: We can use Monte Carlo to estimate the KL divergence of continuous distributions when there's enough data for sampling, i.e, N is large:

$$D_{KL}(\log q(z|x)||\log p(z)) \approx \frac{1}{N}\sum_{i}^{N} \log(\frac{q(z|x)}{p(z)})$$

7. (3 pts) Complete function `generate_and_display_images` (2pts). Note that we use `CVAE`'s `sample` function to obtain the decoded images from latent space. Explain why we set `apply_sigmoid == True` when decoding (1pt). (Hint: Think of why we need to apply sigmoid on the decoder's outputs.)

8. (2 pts) Plot the encoder and decoder architecture as a flow chart diagram with shape specifications below. Are the input and output shapes match your design?

9. (3 pts) Call the completed `main_train_loop` function. Show the following in the output console when training:
   (a) Time used for each epoch training.
   (b) Average ELBO (the negation of the loss value) on the test dataset for each epoch.
   (c) The reconstructed images of the sampled test data after each epoch (Use `generate_and_display_images`).
10. (3 pts) Complete the function `get_allClass_encodings` and `plot_latent_space`. Visualize the latent space. Is there a clear boundary between the classes?
11. (2 pts) Complete the function `walk_src_to_dst`. Show the morphing process of 'walking' from one class to another.

2 CIFAR-10 VAE

1. (8 pts) Repeat the above workflow on CIFAR-10 data set. Try to reuse functions already defined. (Preprocess and visualize data 1 pt, Training 5 pts, visualize latent space 1 pt, walk from two classes 1 pt)
2. (1 pt) Were you able to generate clear images with the CIFAR-10 data set? Can you think of any ways to improve the quality of the reconstructed images?

3 Grad/Extra Credits

1. (4 pts) Modify `compute_loss` to use Maximum Mean Discrepancy (MMD) as described in another blog (2 pts) and train a new model with the new loss function on CIFAR-10 (2 pts).
2. (5 pts) Repeat the visualization of latent space (1 pt) and interpolation between two classes (1 pt). Did you observe any difference before and after using the MMD loss (1 pt). Refer to the blog, explain how MMD loss differs from the original loss function (2 pt).

What to Turn In

Save your Google Colab notebook as ***.ipybn*** file and submit it to Gradescope: https://www.gradescope.com/courses/351582.

The Colab notebook should contain the following parts for grading:

- Functional code for all TODOs in the notebook.
- Answers to all the questions and diagrams/plots listed in the **Problem Formulation**. (36 pts in total for undergrad, 45 pts for grad)

Note: Do not close the cell outputs after running the code cells, so that all logging and plots will be saved in the notebook for grading. We will not run your notebook so make sure all outputs are present.

Good Luck!
ECE 157B TAs