Introduction to Neural Networks 67103

Assignment 3

Due: 1/3/2018

In the practical part of this assignment you will experiment with an autoencoder, and a Generative Adversarial Network trained to generate images similar to those in MNIST. Your implementation and experiments should be done using TensorFlow.

A. Train an autoencoder that takes as input 28x28 MNIST images, and reconstructs it back after going through a low-dimensional latent space of dimension 100***.*** Visualize the distribution of the latent vectors as points in the 2D plane. You may use PCA/LDA or any other available planar embedding technique for that (LDA seems to do a better job here). Ideally the latent vectors should form 10 clusters on the plane. Try to add noise to the input and/or additional terms into the loss function to cause the autoencoder to produce latent vectors with a more apparent grouping into 10 clusters. For example, you might try adding a sparsity term (as in sparse autoencoders), or penalize the derivatives of the latent vector with respect to the input (look up Contractive Autoencoders).

B. Train a GAN that generates MNIST-like images. You may use this implementation of DCGAN **-** to do this. The training of a DCGAN can take a long time to complete without a GPU, but reasonable results begin to emerge after a small number of epochs, and that should be enough for the purposes of this assignment. Explore what happens when training the GAN with:
   a. Uniformly independently distributed random vectors z,
   b. Random vectors sampled from a Gaussian mixture distribution of 10 modes. Place the centers apart from one another (i.e., make sure their variances are large but do not create considerable overlap). Note: this distribution can be sampled directly - no need for an MCMC.
   c. Use the latent representation of the autoencoder as the source of GAN's z vectors. That is, map every MNIST image into the encoder's z vector, and use this list of vectors as if they are sampled from a probabilistic distribution when training and evaluating the GAN.

   Describe how the results change and provide an explanation.

C. Try replacing the generator from part B with the decoder portion of the autoencoder from part A. Can the training of the modified GAN somehow benefit from the availability of the trained autoencoder (from A) and/or the trained discriminator (from B), or is it necessary to train everything from scratch?

Theoretical Questions

1. **Methods:**
    a. Formulate the posterior distribution of an image given its noise-corrupted version in case of an identically independent Laplace noise with parameters (o,b).
    b. Given a uniform variable u~U[0,1], what would be the mapping needed to obtain a new variable whose distribution is p(x)=x+0.5 inside [0,1].
    c. Explain how a Gibbs sampler is used to sample uniformly from the unit 2D disc. Describe the actual steps in pseudo-code (what 1D sampling steps are done). Explain whether the resulting MCMC chain is irreducible or not.
2. **PnP Generative Networks:** In class we described a model P(I,c) where c is the image class. However, small noise perturbations over I using the MALA sampler, are often insufficient to achieve frequent changes in class (c) and hence leads to a poor mixing time. In order to accelerate this process let us use some high-level feature vector h (e.g., deep layers fc6 or fc7) under the following assumptions: the image is a deterministic function of h, the class statistically depends on the image, and h is governed by an independent distribution P(h). Use this model to write down P(I,c,h) as function of these three terms. Explain how you will model each of these terms, and how you'd sample the non-deterministic variables.
3. **Autoencoders**:
    a. Explain how you might be able to use an autoencoder to detect unusual or abnormal instances (among images of a certain class).
    b. Propose how an autoencoder may be used to perform red-eye removal from images of people's faces.
    c. Propose a simple autoencoder-based method for sharpening images.
4. **GANs:**
    a. Explain how to use the GAN framework for image restoration with a general corruption likelihood function C(I|J) (where J is the corrupted image). Explain: what would be inputted to the discriminator, what will the generator be fed with, and what would be V(D,G).
    b. A converged GAN produces samples from the example distribution it was trained on. Is this high-dimensional sampling process also correspond to an MCMC or an independent sampling process?

Submission and Grading

Submit (via moodle) a ZIP archive file named:
”ex2 _(student1name)[_student2name].zip"

The file should contain your code, and a document named "answers.pdf", which clearly describes your solutions and reports your results.

Remarks:
- The "answers.pdf" document should be a pdf, not a docx, (ONLY ONE PDF), additional documents will not be checked.
- If you are doing the assignment in pairs, only one of you should submit one zip file.
- Please write your names inside the pdf file.
- Please write some documentation explaining your code.

Grading: exercises which provide complete and correct answers and working solutions that satisfy the exercise requirements will result in a grade up to 95. A bonus of up to 5 points may be provided to creative solutions that go beyond the requirements.