Homework 5

Please upload your assignments on or before November 13, 2020.

- You are encouraged to discuss ideas with each other; but
- you **must acknowledge** your collaborator, and
- you **must compose your own** writeup and/or code independently.
- We **require** answers to theory questions to be written in LaTeX, and answers to coding questions in Python (Jupyter notebooks)
- Upload your answers in the form of a single PDF on Gradescope.

---

1. **(3 points)** *Reverse this*. In class we defined the affine coupling layer while discussing normalizing flows. In this problem, we will generalize this operation a bit. Consider a reversible block that implements the following operations:

$$x = \alpha \odot z + F_\Theta(u)$$
$$v = \beta \odot u$$

where α, β (parameter vectors) and F_Θ are all trainable and, and $\odot$ represents element-wise multiplication.

   a. Derive the inverse of this block, i.e., given x, v show how to calculate z, u.

   b. Derive a formula for the Jacobian of this block.

   c. From your answer to part b, derive the *determinant* of the Jacobian.

2. **(3 points)** *Making skip-gram training practical*. Suppose we have a dictionary containing d words, and we are trying to learn skip-gram embeddings of each word using a very large training corpus. Suppose that our embedding dimension is chosen to be h.

   a. Calculate the gradient of the cross-entropy loss for a *single* target-context training pair of words. What is the running time of calculating this gradient in terms of d and h?

   b. Here is one way to improve this running time. Argue that the regular softmax calculation as visiting all the leaves of a tree of depth-1, where each word in the dictionary corresponds a leaf. What kind of data structure would enable faster computation of the softmax, and how much speedup would you get? Intuitively explain how to construct this data structure. You don't have to be super-precise or mathematical, loose and brief intuition is enough.

   c. Here is a second approach. Argue that the gradient from part a can be written as a weighted average of some number of terms, and intuitively describe a *sampling*-based method to improve the running time of the gradient calculation.

3. **(4 points)** In this problem, the goal is to train and visualize the outputs of a simple Deep Convolutional GAN (DCGAN) to generate realistic-looking (but fake) images of clothing.

   a. Use the FashionMNIST training dataset to train the DCGAN. APIs for downloading it are available in both PyTorch and TensorFlow. Images are grayscale and size 28×28.

b. Use the following discriminator architecture (kernel size = 5×5 with stride = 2 in both directions):

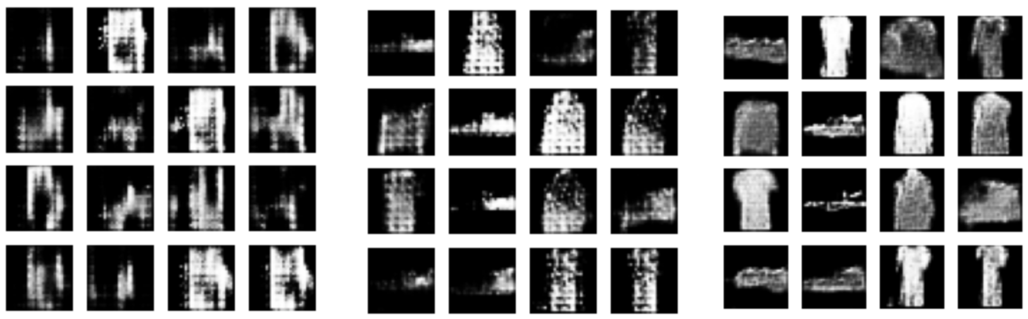
- 2D convolutions ($1 \times 28 \times 28 \rightarrow 64 \times 14 \times 14 \rightarrow 128 \times 7 \times 7$)
- each convolutional layer is equipped with a Leaky ReLU with slope 0.3, followed by Dropout with parameter 0.3.
- a dense layer that takes the flattened output of the last convolution and maps it to a scalar.

c. Use the following generator architecture (which is essentially the reverse of a standard discriminative architecture):

- a dense layer that takes a unit Gaussian noise vector of length 100 and maps it to a vector of size $7 * 7 * 256$. No bias terms.
- several transpose 2D convolutions ($256 \times 7 \times 7 \rightarrow 128 \times 7 \times 7 \rightarrow 64 \times 14 \times 14 \rightarrow 1 \times 28 \times 28$). No bias terms.
- each convolutional layer (except the last one) is equipped with Batch Normalization (BN), followed by Leaky ReLU with slope 0.3. The last (output) layer is equipped with tanh activation (no BN).

d. Use the cross-entropy loss for training both the generator and the discriminator. Use the Adam optimizer with learning rate 10^{-4}.

e. Train it for 50 epochs. Display intermediate images generated after $T = 10$, $T = 30$, and $T = 50$ epochs. If the random seeds are fixed throughout then you should get results of the following sort:



f. Report loss curves for both the discriminator and the generator loss over all epochs, and comment on their behavior.