HOMEWORK 6

William Powell
908 343 3244

Instructions: Use this latex file as a template to develop your homework. Submit your homework on time as a single pdf file. Please wrap your code and upload to a public GitHub repo, then attach the link below the instructions so that we can access it. Answers to the questions that are not within the pdf are not accepted. This includes external links or answers attached to the code implementation. Late submissions may not be accepted. You can choose any programming language (i.e. python, R, or MATLAB). Please check Piazza for updates about the homework. It is ok to share the results of the experiments and compare them with each other.

Source code for this assignment is available at https://github.com/wgraysonp/CS760/tree/main/HW_6.

1 Implementation: GAN (50 pts)

In this part, you are expected to implement GAN with MNIST dataset. We have provided a base jupyter notebook (gan-base.ipynb) for you to start with, which provides a model setup and training configurations to train GAN with MNIST dataset.

(a) Implement training loop and report learning curves and generated images in epoch 1, 50, 100. Note that drawing learning curves and visualization of images are already implemented in provided jupyter notebook. (20 pts)

Procedure 1 Training GAN, modified from Goodfellow et al. (2014)

```
Input: $m$: real data batch size, $n_z$: fake data batch size
Output: Discriminator $D$, Generator $G$
for number of training iterations do
    # Training discriminator
    Sample minibatch of $n_z$ noise samples $\{z^{(1)}, z^{(2)}, \cdots, z^{(n_z)}\}$ from noise prior $p_g(z)$
    Sample minibatch of $\{x^{(1)}, x^{(2)}, \cdots, x^{(m)}\}$
    Update the discriminator by ascending its stochastic gradient:
        $$\nabla_{\theta_d}\Big(\frac{1}{m}\sum_{i=1}^{m}\log D(x^{(i)}) + \frac{1}{n_z}\sum_{i=1}^{n_z}\log(1 - D(G(z^{(i)})))\Big)$$
    # Training generator
    Sample minibatch of $n_z$ noise samples $\{z^{(1)}, z^{(2)}, \cdots, z^{(n_z)}\}$ from noise prior $p_g(z)$
    Update the generator by ascending its stochastic gradient:
        $$\nabla_{\theta_g}\frac{1}{n_z}\sum_{i=1}^{n_z}\log D(G(z^{(i)})))$$
end for
# The gradient-based updates can use any standard gradient-based learning rule. In the base code, we are using Adam optimizer (Kingma and Ba, 2014)
```

The Learning curve is displayed below in Figure 1. Generated images from epochs 1, 50, and 100 are shown in Figure 2.

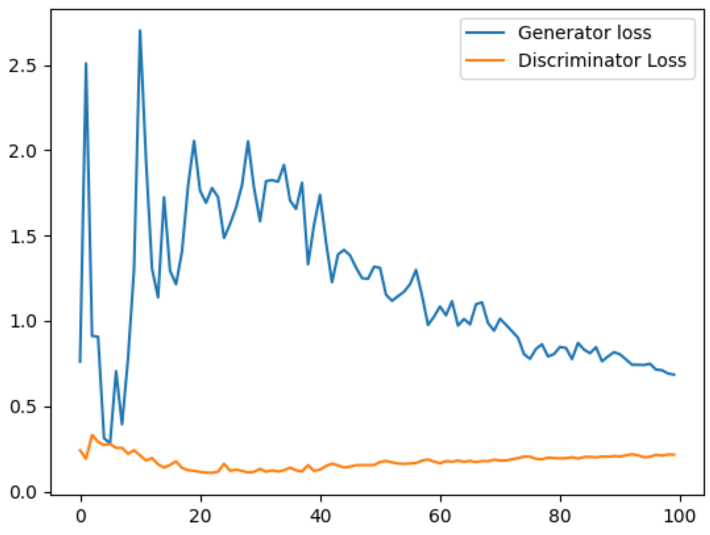
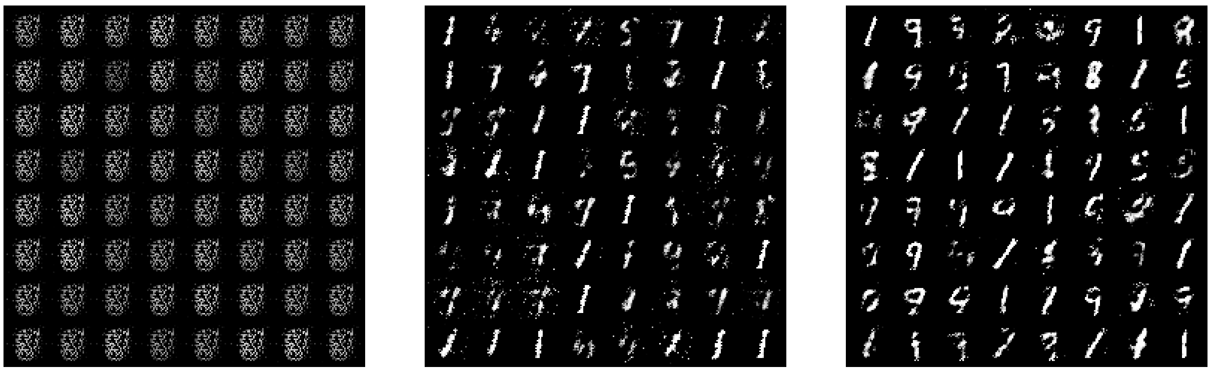


Figure 1: Learning curve



(a) epoch 1 (b) epoch 50 (c) epoch 100

Figure 2: Generated images by G

(b) Replace the generator update rule as the original one in the slide,
"Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{n_z} \sum_{i=1}^{n_z} \log(1 - D(G(z^{(i)})))$$

" , and report learning curves and generated images in epoch 1, 50, 100. Compare the result with (a). Note that it may not work. If training does not work, explain why it doesn't work.
You may find this helpful: https://jonathan-hui.medium.com/gan-what-is-wrong-with-the-gan-cost-function-6f594162ce01 (10 pts)

The Learning curve is displayed below in Figure 3. Generated images from epochs 1, 50, and 100 are shown in Figure 4.

Training using the originally proposed generator cost function failed. Let

$$F(\theta) = \mathbb{E}_{z \sim p} \log(1 - D(G_\theta(z))) \tag{1}$$

According to the provided article, we have $\|\nabla F(\theta)\| \to 0$ as the accuracy of the discriminator improves. Then improving the discriminator during the training process causes the parameters of the generator to approach a sub-optimal stationary point. With small gradients, gradient based optimization methods cannot progress meaning the training of the generator is stalled.

Although not encountered in this example, another problem with the loss function F is that it is unbounded from below so it is generally intractable to solve $\min_\theta F(\theta)$. Also, as $y \to 1$ we have $\frac{d}{dy}\log(1-y) \to -\infty$. So, using this loss function introduces the potential of exploding gradients.

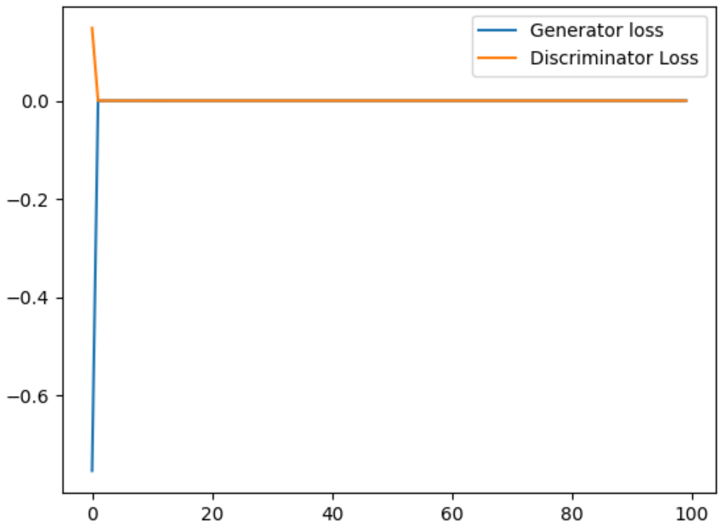
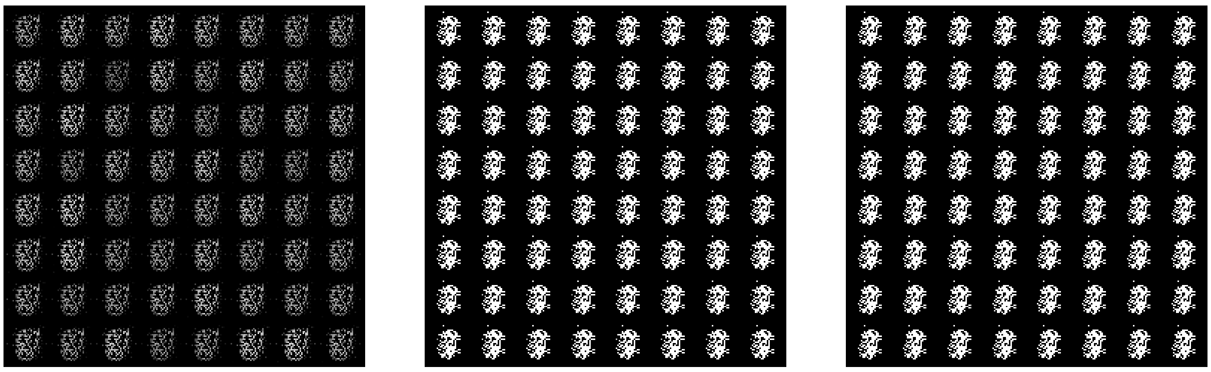


Figure 3: Learning curve



(a) epoch 1 (b) epoch 50 (c) epoch 100

Figure 4: Generated images by G

(c) Except the method that we used in (a), how can we improve training for GAN? Implement that and report your setup, learning curves, and generated images in epoch 1, 50, 100. This question is an open-ended question and you can choose whichever method you want. (20 pts)

One option is to give the generator more data. In Procedure 1, the batch size used to train the generator and the number of fake images used to train the discriminator are the same. In this experiment, the number of noise samples used to train the generator was increased to $4n_z$ with the goal of adding more stability to the Generator's stochastic gradient. The Generator loss does decrease slightly more quickly than in part (a), but there is little difference in final image quality. Results are shown in Figures 5 and 6

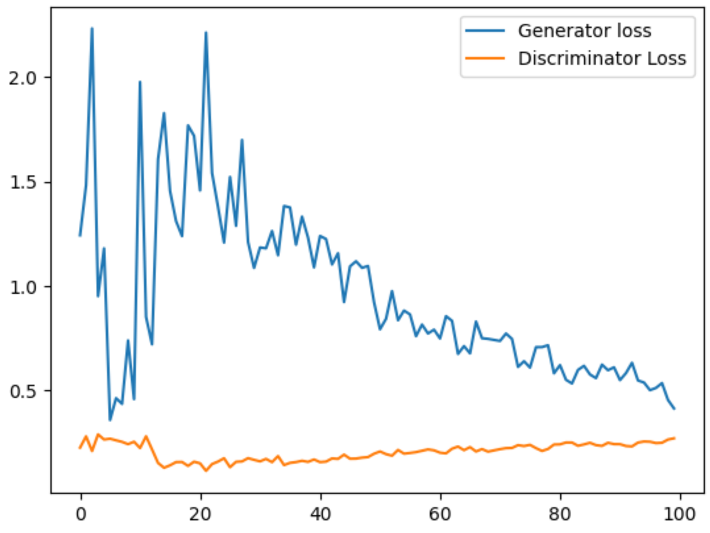
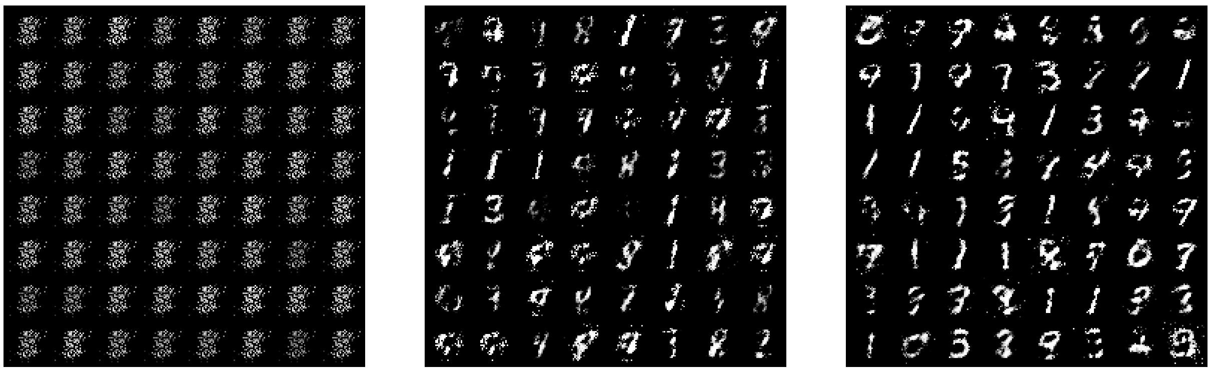


Figure 5: Learning curve



(a) epoch 1 (b) epoch 50 (c) epoch 100

Figure 6: Generated images by G with larger batch size

2 Directed Graphical Model [25 points]

Consider the directed graphical model (aka Bayesian network) in Figure 7.

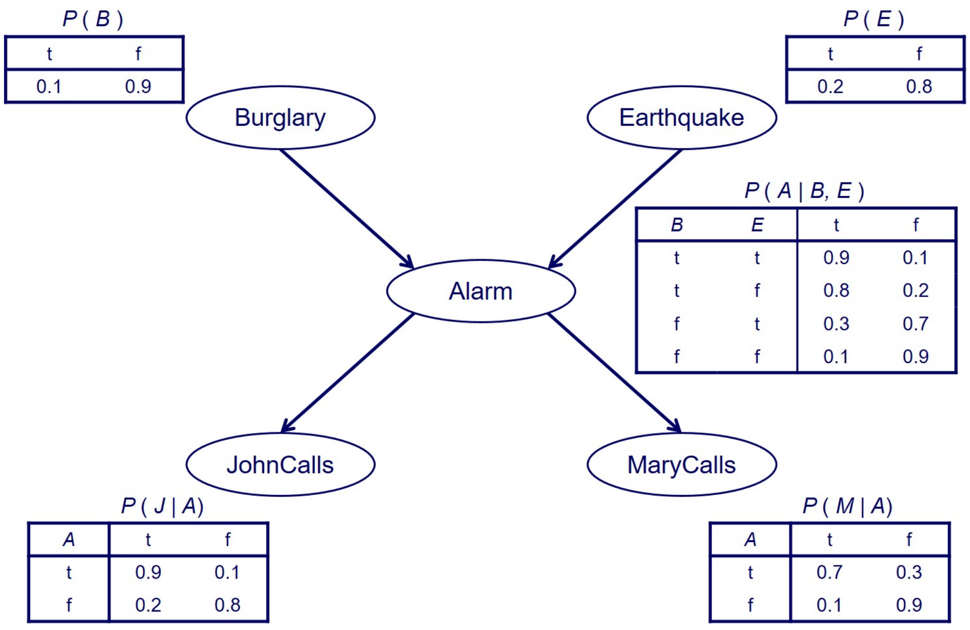


Figure 7: A Bayesian Network example.

Compute $P(B = t \mid E = f, J = t, M = t)$ and $P(B = t \mid E = t, J = t, M = t)$. (10 points for each) These are the conditional probabilities of a burglar in your house (yikes!) when both of your neighbors John and Mary call you and say they hear an alarm in your house, but without or with an earthquake also going on in that area (what a busy day), respectively.

To compute $P(B = t|E = f, J = t, M = t)$ we first compute

$$P(B = t, E = f, J = t, M = t) = \underbrace{P(B = t, E = f, J = t, M = t, A = t)}_{\text{call this } I} + \underbrace{P(B = t, E = f, J = t, M = t, A = f)}_{\text{call this } II}.$$

We have

$$\begin{aligned} I &= P(B = t)P(E = f)P(A = t|B = t, E = f)P(J = t|A = t)P(M = t|A = t) \\ &= 0.1 * 0.8 * 0.8 * 0.9 * 0.7 \\ &= 0.04032 \end{aligned}$$

and

$$\begin{aligned} II &= P(B = t)P(E = f)P(A = f|B = t, E = f)P(J = t|A = f)P(M = t|A = f) \\ &= 0.1 * 0.8 * 0.2 * 0.2 * 0.1 \\ &= 0.00032 \end{aligned}$$

Adding the two we get

$$P(B = t, E = f, J = t, M = t) = I + II = 0.04064$$

Next we need to compute $P(B = f, E = f, J = t, M = t)$. Using a similar strategy:

$$P(B = f, E = f, J = t, M = t) = \underbrace{P(B = f, E = f, J = t, M = t, A = t)}_{I} + \underbrace{P(B = f, E = f, J = t, M = t, A = f)}_{II}.$$

Computing the two quantities on the right:

$$\begin{aligned} I &= P(B = f)P(E = f)P(A = t|B = t, E = f)P(J = t|A = t)P(M = t|A = t) \\ &= 0.9 * 0.8 * 0.8 * 0.9 * 0.7 \\ &= 0.36288 \end{aligned}$$

and

$$\begin{aligned}II &= P(B=f)P(E=f)P(A=f|B=t,E=f)P(J=t|A=f)P(M=t|A=f)\\ &= 0.9*0.8*0.2*0.2*0.1\\ &= 0.00288.\end{aligned}$$

So $P(B=f,E=f,J=t,M=t)=0.36576$. Finally,

$$\begin{aligned}P(B=t|E=f,J=t,M=t) &= \frac{P(B=t,E=f,J=t,M=t)}{P(E=f,J=t,M=t)}\\ &= \frac{P(B=t,E=f,J=t,M=t)}{P(B=t,E=f,J=t,M=t)+P(B=f,E=f,J=t,M=t)}\\ &= \frac{0.04064}{0.04064+0.36576}\\ &= 0.526.\end{aligned}$$

We repeat this procedure to compute $P(B=t|E=t,J=t,M=t)$. We have

$$P(B=t,E=t,J=t,M=t)=P(B=t,E=t,J=t,M=t,A=t)+P(B=t,E=t,J=t,M=t,A=f).$$

The first quantity on the right is

$$\begin{aligned}&P(B=t)P(E=t)P(A=t|B=t,E=t)P(J=t|A=t)P(M=t|A=t)\\ &= 0.1*0.2*0.9*0.9*0.7\\ &= 0.01134\end{aligned}$$

and the second is

$$\begin{aligned}&P(B=t)P(E=t)P(A=f|B=t,E=t)P(J=t|A=f)P(M=t|A=f)\\ &= 0.1*0.2*0.1*0.2*0.1\\ &= 0.00004.\end{aligned}$$

So $P(B=t,E=t,J=t,M=t)=0.01138$. Now

$$P(B=f,E=t,J=t,M=t)=P(B=f,E=t,J=t,M=t,A=t)+P(B=f,E=t,J=t,M=t,A=f).$$

The first term in the sum is

$$\begin{aligned}&P(B=f)P(E=t)P(A=t|B=f,E=t)P(J=t|A=t)P(M=t|A=t)\\ &= 0.9*0.2*0.3*0.9*0.7\\ &= 0.03402\end{aligned}$$

and the second is

$$\begin{aligned}&P(B=f)P(E=t)P(A=f|B=f,E=t)P(J=t|A=f)P(M=t|A=f)\\ &= 0.9*0.2*0.7*0.2*0.1\\ &= 0.00252.\end{aligned}$$

So $P(B=f,E=t,J=t,M=t)=0.03654$. Finally we have

$$\begin{aligned}P(B=t|E=t,J=t,M=t) &= \frac{P(B=t,E=t,J=t,M=t)}{P(B=t,E=t,J=t,M=t)+P(B=f,E=t,J=t,M=t)}\\ &= \frac{0.01138}{0.01138+0.03654}\\ &= 0.237.\end{aligned}$$

3 Chow-Liu Algorithm [25 pts]

Suppose we wish to construct a directed graphical model for 3 features X, Y, and Z using the Chow-Liu algorithm. We are given data from 100 independent experiments where each feature is binary and takes value T or F. Below is a table summarizing the observations of the experiment:

X	Y	Z	Count
T	T	T	36
T	T	F	4
T	F	T	2
T	F	F	8
F	T	T	9
F	T	F	1
F	F	T	8
F	F	F	32

1. Compute the mutual information $I(X, Y)$ based on the frequencies observed in the data. (5 pts)

   Using the tables we can compute the empirical marginal distributions of X, Y, and (X, Y). For the marginals of X and Y we have $P(X = T) = P(X = F) = 1/2$ and $P(Y = T) = P(Y = F) = 1/2$. And for the vector (X, Y) we can compute

$$\begin{aligned} P(X = T, Y = T) &= 2/5 \\ P(X = T, Y = F) &= 1/10 \\ P(X = F, Y = T) &= 1/10 \\ P(X = F, Y = F) &= 2/5. \end{aligned}$$

   Then

$$\begin{aligned} I(X, Y) &= \sum_{(i,j)\in\{T,F\}^2} P(X = i, Y = j) \log_2 \frac{P(X = i, Y = j)}{P(X = i)P(Y = j)} \\ &= 2 * 2/5 \log_2 \frac{2/5}{1/4} + 2 * 1/10 \log_2 \frac{1/10}{1/4} \\ &= 4/5 * (3 - \log_2 5) + 1/5 * (1 - \log_2 5) \\ &= 13/5 - \log_2 5 \\ &= 0.2781. \end{aligned}$$

2. Compute the mutual information $I(X, Z)$ based on the frequencies observed in the data. (5 pts)
   The marginal distribution of Z is given by $P(Z = T) = \frac{55}{100}$ and $P(Z = F) = \frac{45}{100}$ and (X, Z) has distribution

$$\begin{aligned} P(X = T, Z = T) &= 38/100 \\ P(X = T, Z = F) &= 12/100 \\ P(X = F, Z = T) &= 17/100 \\ P(X = F, Z = F) &= 33/100. \end{aligned}$$

   Then

$$\begin{aligned} I(X, Z) &= 38/100 \log_2 \frac{38/100}{55/200} + 12/100 \log_2 \frac{12/100}{45/200} + \frac{17}{100} \log_2 \frac{17/100}{55/200} + 33/100 \log_2 \frac{33/100}{45/200} \\ &= 0.1328 \end{aligned}$$

3. Compute the mutual information $I(Z, Y)$ based on the frequencies observed in the data. (5 pts)

   We can use the marginal distributions of Z and Y from the last two problems. We then see that (Y, Z) is distributed as

$$\begin{aligned} P(Y = T, Z = T) &= 45/100 \\ P(Y = T, Z = F) &= 5/100 \\ P(Y = F, Z = T) &= 1/10 \\ P(Y = F, Z = F) &= 2/5. \end{aligned}$$

   Then

$$\begin{aligned} I(Y, Z) &= 45/100 * \log_2 \frac{45/100}{55/200} + 5/100 * \log_2 \frac{5/100}{45/200} + 1/10 * \log_2 \frac{1/10}{55/200} + 2/5 * \log_2 \frac{2/5}{45/200} \\ &= 0.397. \end{aligned}$$

4. Which undirected edges will be selected by the Chow-Liu algorithm as the maximum spanning tree? (5 pts)

   The Chow-Liu algorithm will select the edges (X, Y) and (Y, Z) since we need two edges to span the graph and they have the highest mutual information.

5. Root your tree at node X, assign directions to the selected edges. (5 pts)

   To assign directions, we start with X as the root and make all edges directed from the root. In this case we will have edges $X \to Y$ and $Y \to X$.

References

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.