HOMEWORK 6

Nevindu M. Batagoda
9081677594

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.

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:

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

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)

Expected results are as follows.

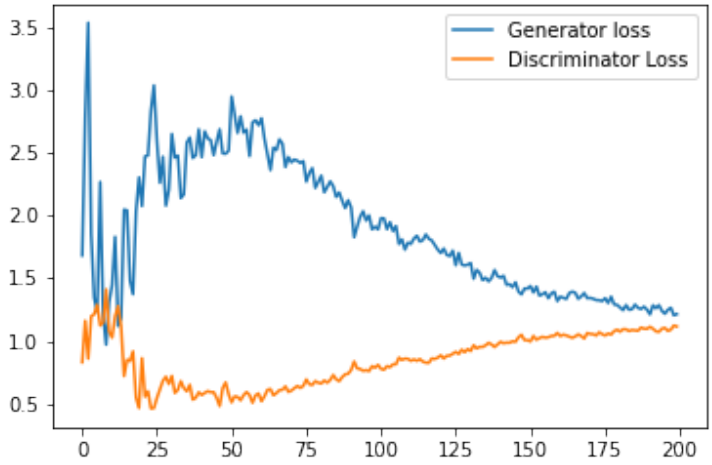
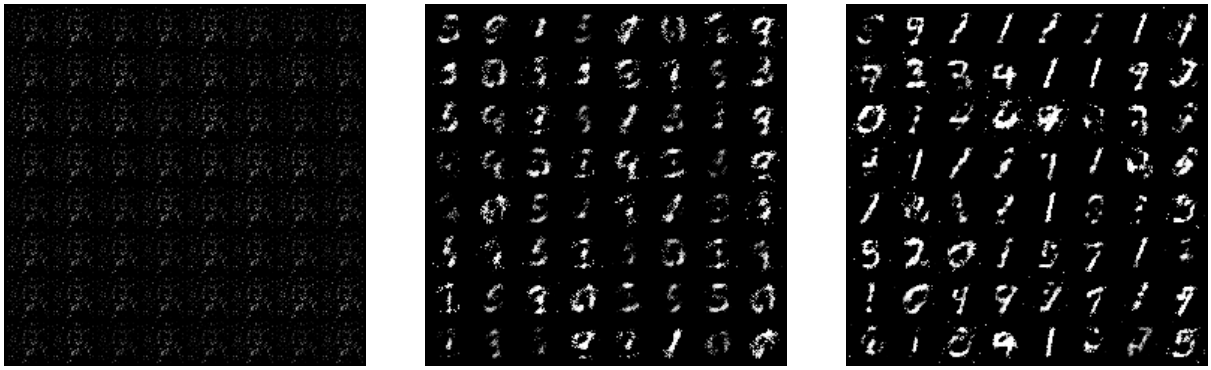


Figure 1: Learning curve



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

Figure 2: Generated images by G

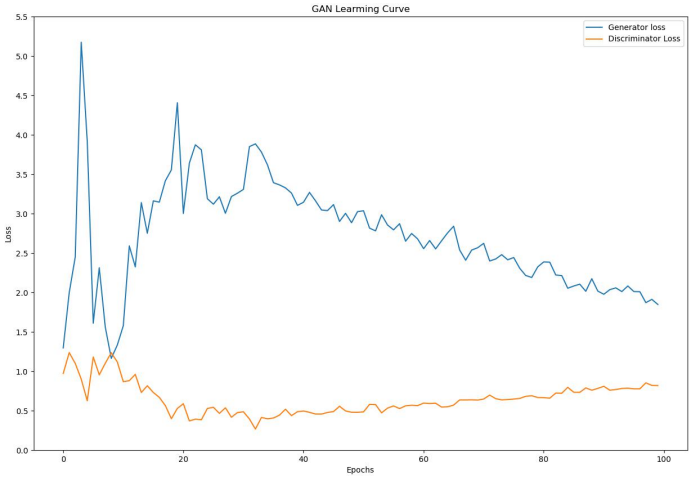
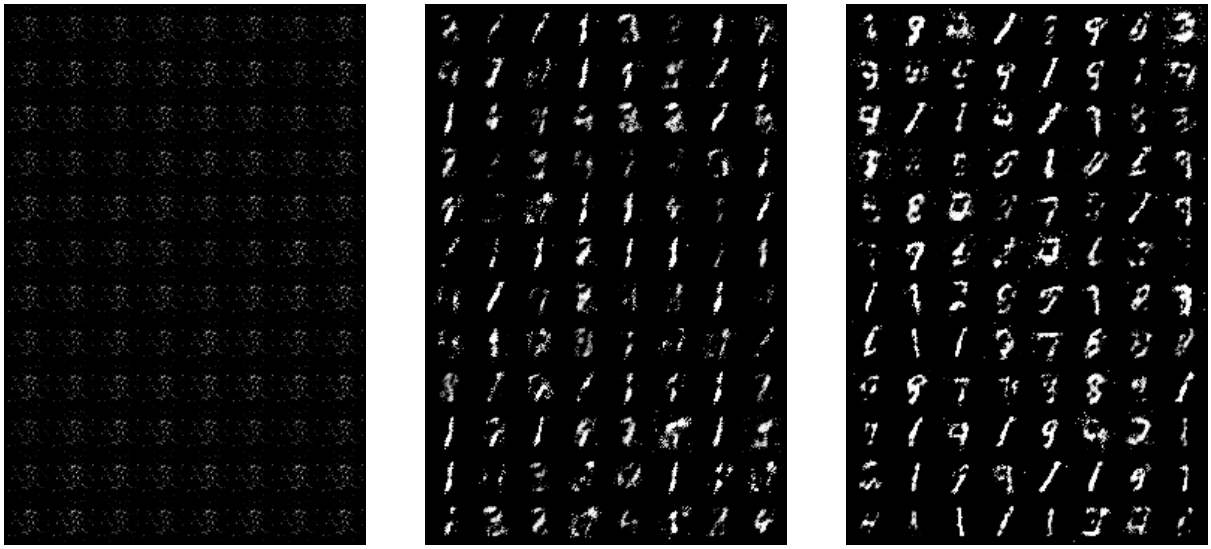


Figure 3: Learning curve (Answer)



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

Figure 4: Generated images by G (Answer)

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

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

" , 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)

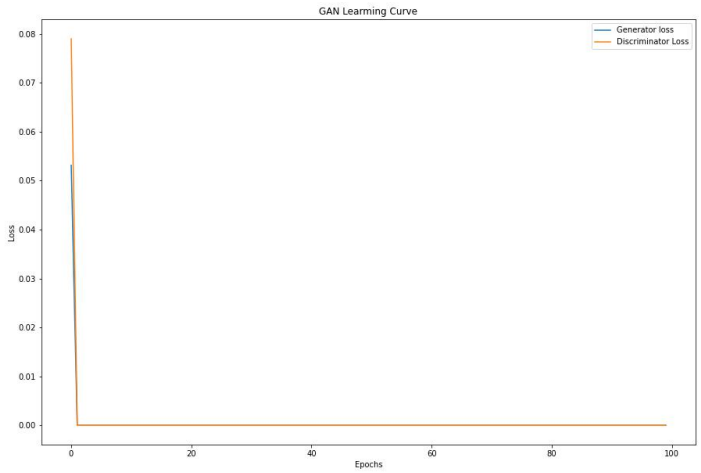
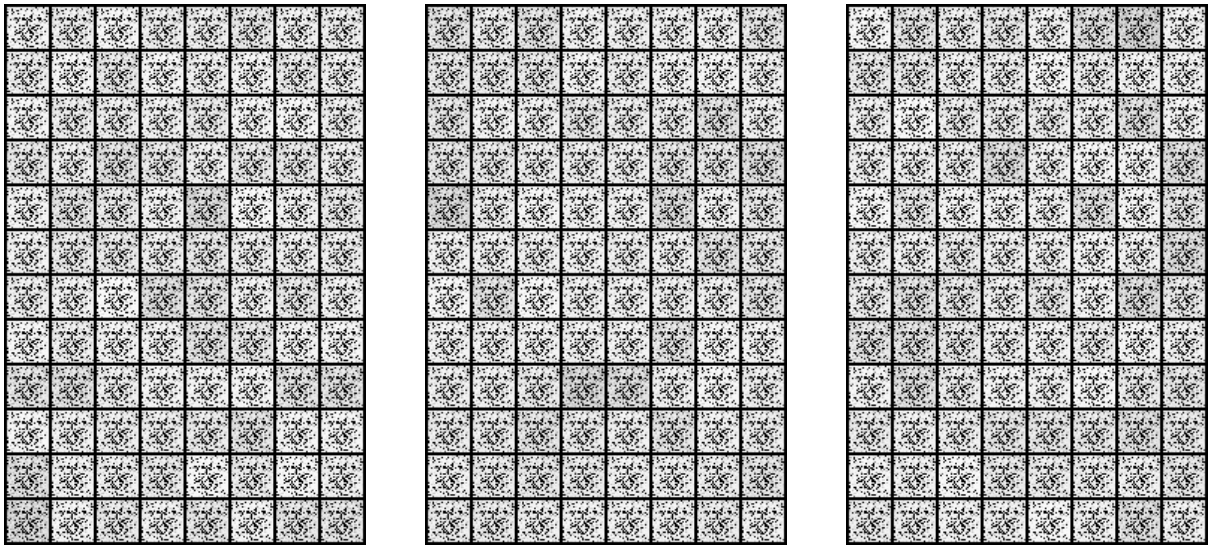


Figure 5: Learning curve for new generator update rule



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

Figure 6: Generated images by G using new generator update rule

From the above graph and images you can see that new generator update rule does not produce good results. This is because for the real and fake/generated latent distribtions we have with the our current dataset and GAN, the discriminator is quickly able to converge to an optimum with 100% accuracy (Zero loss). This causes the generator loss (new rule given above) always be 0 due to the discriminator being too strong. Hence the generator is not able to learn anything. This is called the problem of vanishing gradients.

(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)

I tried the following methods to improve the training of GAN:

(i) Adding noise to the discriminator input - Inspired by the above linked medium article I added random gaussian noise to the discriminator input. This helps with the problem of vanishing gradients or exploding gradients in the GAN loss functions.

(ii) Smoothing the labels - Instead of using 0 and 1 as labels for real and fake images respectively, I used 0.1 and 0.9 as labels for real and fake images respectively. This prevents the discriminator from being too overconfident about its predictions, improves gradient flow and subsequently helps avoid mode collapse.

(iii) I tried using the NAdam optimizer instead of Adam optimizer. NAdam is a variant of Adam optimizer which uses Nesterov acceleration. NAdam provides faster convergence than Adam and adaptive learning rates.

From the below figures you can see that the GAN trains faster with faster convergence. Furthermore the generated images are of better quality than the ones generated by the original GAN. Where it is less noisy,has less artefacts more structuraly coherent and the samples are also more varied/diverse than the original GAN from part a).

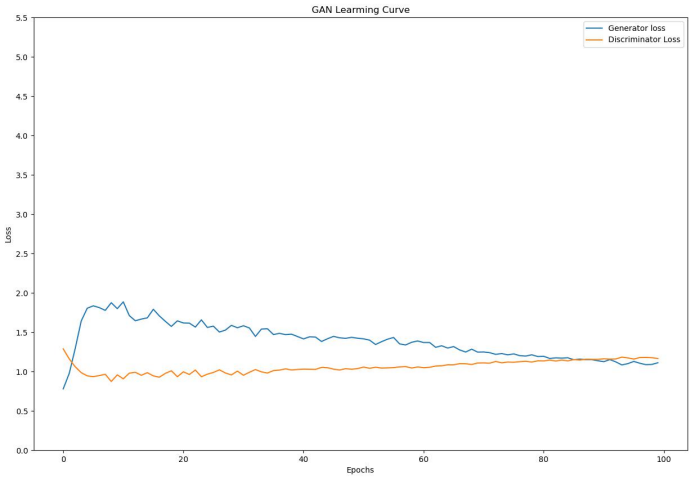


Figure 7: Learning curve for imporoved GAN training

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

Figure 8: Generated images by G from improved GAN training

2 Directed Graphical Model [25 points]

Consider the directed graphical model (aka Bayesian network) in Figure 9.

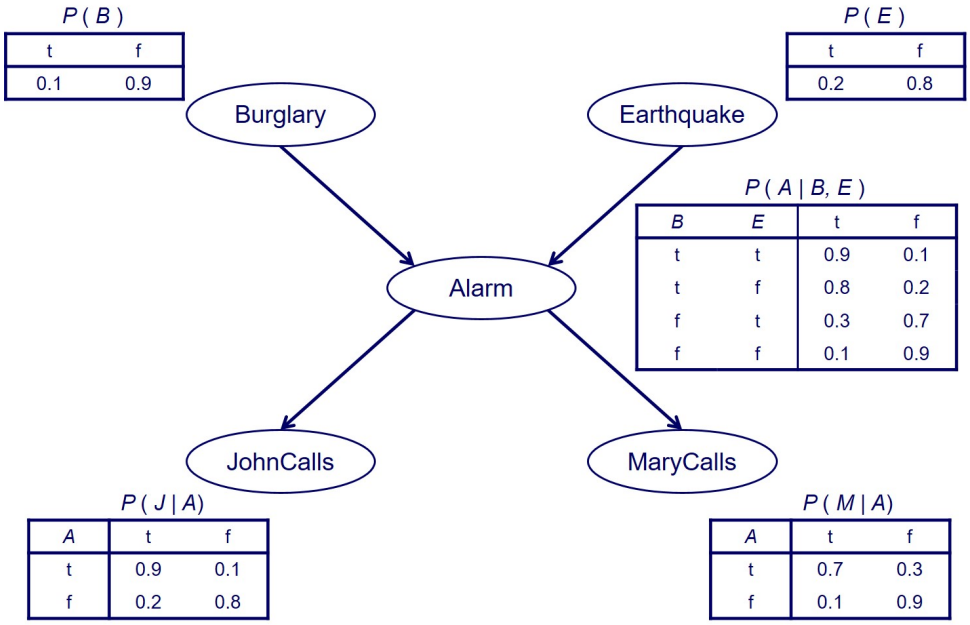


Figure 9: A Bayesian Network example.

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.

- $$P(B = t \mid E = f, J = t, M = t) = \frac{P(B = t, E = f, J = t, M = t)}{P(B = t, E = f, J = t, M = t) + P(B = f, E = f, J = t, M = t)}$$

$$\begin{aligned} P(B = t, E = f, J = t, M = t) &= P(B = t)P(E = f)P(A = t \mid B = t, E = f)P(J = t \mid A = t)P(M = t \mid A = t) \\ &\quad + P(B = t)P(E = f)P(A = f \mid B = t, E = f)P(J = t \mid A = f)P(M = t \mid A = f) \\ &= (0.1 \times 0.8 \times 0.8 \times 0.9 \times 0.7) + (0.1 \times 0.8 \times 0.2 \times 0.2 \times 0.1) \\ &= 0.04032 + 0.00032 \\ &= 0.04064 \end{aligned}$$

$$\begin{aligned} P(B = f, E = f, J = t, M = t) &= P(B = f)P(E = f)P(A = t \mid B = f, E = f)P(J = t \mid A = t)P(M = t \mid A = t) \\ &\quad + P(B = f)P(E = f)P(A = f \mid B = f, E = f)P(J = t \mid A = f)P(M = t \mid A = f) \\ &= (0.9 \times 0.8 \times 0.1 \times 0.9 \times 0.7) + (0.9 \times 0.8 \times 0.9 \times 0.2 \times 0.1) \\ &= 0.04536 + 0.01296 \\ &= 0.01138 \end{aligned}$$

$$\begin{aligned} \Rightarrow P(B = t \mid E = f, J = t, M = t) &= \frac{0.04064}{0.04064 + 0.05832} \\ &\approx 0.41067 \end{aligned}$$

- $$P(B = t \mid E = t, J = t, M = t) = \frac{P(B = t, E = t, J = t, M = t)}{P(B = t, E = t, J = t, M = t) + P(B = f, E = t, J = t, M = t)}$$

$$\begin{aligned}P(B=t,E=t,J=t,M=t) &= P(B=t)P(E=t)P(A=t \mid B=t,E=t)P(J=t \mid A=t)P(M=t \mid A=t)\\ &+ P(B=t)P(E=t)P(A=f \mid B=t,E=t)P(J=t \mid A=f)P(M=t \mid A=f)\\ &= (0.1\times 0.2\times 0.9\times 0.9\times 0.7)+(0.1\times 0.2\times 0.1\times 0.2\times 0.1)\\ &= 0.01134+0.00004\\ &= 0.01138\end{aligned}$$

$$\begin{aligned}P(B=f,E=t,J=t,M=t) &= P(B=f)P(E=t)P(A=t \mid B=f,E=t)P(J=t \mid A=t)P(M=t \mid A=t)\\ &+ P(B=f)P(E=t)P(A=f \mid B=f,E=t)P(J=t \mid A=f)P(M=t \mid A=f)\\ &= (0.9\times 0.2\times 0.3\times 0.9\times 0.7)+(0.9\times 0.2\times 0.7\times 0.2\times 0.1)\\ &= 0.03402+0.0025\\ &= 0.03654\end{aligned}$$

$$\begin{aligned}\Rightarrow P(B=t \mid E=t,J=t,M=t) &= \frac{0.01138}{0.01138+0.03654}\\ &\approx 0.23748\end{aligned}$$

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)

$$\begin{aligned}I(X,Y) &= \sum_{x\in\{T,F\}}\sum_{y\in\{T,F\}} P(X=x,Y=y)\log\frac{P(X=x,Y=y)}{P(X=x)P(Y=y)}\\ &= P(X=T,Y=T)\log\frac{P(X=T,Y=T)}{P(X=T)P(Y=T)}+P(X=T,Y=F)\log\frac{P(X=T,Y=F)}{P(X=T)P(Y=F)}\\ &+ P(X=F,Y=T)\log\frac{P(X=F,Y=T)}{P(X=F)P(Y=T)}+P(X=F,Y=F)\log\frac{P(X=F,Y=F)}{P(X=F)P(Y=F)}\\ &= 0.4\log\frac{0.4}{0.5\times 0.5}+0.1\log\frac{0.1}{0.5\times 0.5}+0.1\log\frac{0.1}{0.5\times 0.5}+0.4\log\frac{0.4}{0.5\times 0.5}\\ &= 0.2781\end{aligned}$$

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

$$
\begin{aligned}
I(X,Z) &= \sum_{x\in\{T,F\}} \sum_{z\in\{T,F\}} P(X=x, Z=z) \log \frac{P(X=x, Z=z)}{P(X=x)P(Z=z)} \\
&= P(X=T, Z=T) \log \frac{P(X=T, Z=T)}{P(X=T)P(Z=T)} + P(X=T, Z=F) \log \frac{P(X=T, Z=F)}{P(X=T)P(Z=F)} \\
&+ P(X=F, Z=T) \log \frac{P(X=F, Z=T)}{P(X=F)P(Z=T)} + P(X=F, Z=F) \log \frac{P(X=F, Z=F)}{P(X=F)P(Z=F)} \\
&= 0.38 \log \frac{0.38}{0.5 \times 0.55} + 0.12 \log \frac{0.12}{0.5 \times 0.45} + 0.17 \log \frac{0.17}{0.5 \times 0.55} + 0.33 \log \frac{0.33}{0.5 \times 0.45} \\
&= 0.1328
\end{aligned}
$$

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

$$
\begin{aligned}
I(Z,Y) &= \sum_{z\in\{T,F\}} \sum_{y\in\{T,F\}} P(Z=z, Y=y) \log \frac{P(Z=z, Y=y)}{P(Z=z)P(Y=y)} \\
&= P(Z=T, Y=T) \log \frac{P(Z=T, Y=T)}{P(Z=T)P(Y=T)} + P(Z=T, Y=F) \log \frac{P(Z=T, Y=F)}{P(Z=T)P(Y=F)} \\
&+ P(Z=F, Y=T) \log \frac{P(Z=F, Y=T)}{P(Z=F)P(Y=T)} + P(Z=F, Y=F) \log \frac{P(Z=F, Y=F)}{P(Z=F)P(Y=F)} \\
&= 0.45 \log \frac{0.45}{0.55 \times 0.5} + 0.1 \log \frac{0.1}{0.55 \times 0.5} + 0.05 \log \frac{0.05}{0.45 \times 0.5} + 0.4 \log \frac{0.4}{0.45 \times 0.5} \\
&= 0.3973
\end{aligned}
$$

4. Which undirected edges will be selected by the Chow-Liu algorithm as the maximum spanning tree? (5 pts) The set of edges in order of decreasing mutual information is $\{(Z, Y), (X, Z), (X, Y)\}$. From these edges only (Z, Y) and (X, Z) will be selected as they have the largest weights. (X, Y) will not be selected as selecting it would create a cycle. Therefore The maximum spanning tree consist of the following edges: $\{(X, Y), (Z, Y)\}$.

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

$$X \to Y \to Z$$

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*.