ENGR-E 533 “Deep Learning Systems” Lecture 13: Generative Adversarial Networks

Minje Kim

Department of Intelligent Systems Engineering

Email: minje@indiana.edu

Website: http://minjekim.com

Research Group: http://saige.sice.indiana.edu

Meeting Request: http://doodle.com/minje

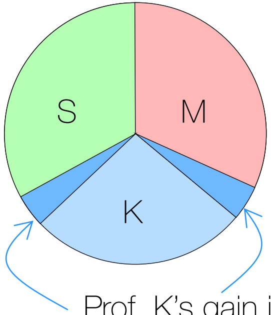
INDIANA UNIVERSITY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Game Theory

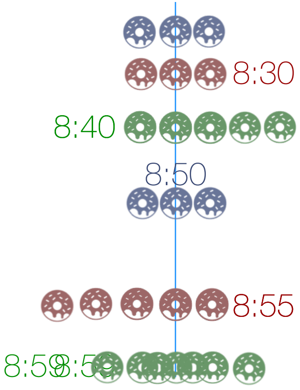
- Zero-sum game

- In a colloquium talk there are 11 donuts
  - Prof. K likes sweets so he took more than the allowed amount
  - Stud. S and Stud. M will lose what Prof. K gained
  - Since the number of donuts is constant, the sum of the total loss and gain is zero



Prof. K's gain is the lost of the others

- Rules for the “Stupid Free Donut” game
  - The students compete: each student wants to eat more than the other one
  - Each student has two choices due to the bus schedule
  - Prof. K is always the earliest one
  - The first two get 3 donuts each and the third one gets the rest
    - But Prof. K eats another 3 at 8:50 if Stud. S as the third one doesn't arrive by then
      - Because he thinks Stud. S won't come and he is always hungry
    - Prof. K doesn't eat any more donuts if Stud. M is the third person
      - Because he knows Stud. M will come
    - If none of the students comes by 8:50
      - Prof. K doesn't take any more donuts, but waits for one to arrive



SW \ MM	8:30 AM	8:55 AM
8:40 AM	-2 +2	+2 -2
8:59 AM	+1 -1	-2 +2

- This creates a complicated payoff matrix

Game Theory

- Zero-sum game

- For Stud. S 8:59 is always the better choice
  - The potential loss is smaller
- If Stud. M knows Stud. S prefers 8:59, Stud. M will choose 8:30
  - Stud. M gets +1
- To beat this Stud. S would like to choose 8:40 instead
- And so on..
- Who wins this game?
- If both repeat this game for 100 times during their PhD study
  - There must be an expected gain depending on their strategy (strategy?)
  - Strategy means their probability of choosing the two options
  - **Minimax optimization: minimize the maximum expected point-loss independent of the opponent's strategy**
- Linear programming can solve this problem
  - Nash equilibrium

MM / SW	8:30 AM	8:55 AM
8:40 AM	-2 / +2	+2 / -2
8:59 AM	+1 / -1	-2 / +2

Drawback of VAE

- Quality of the approximate posterior distribution matters

$$
\begin{aligned}
KL\big(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})\big) &= \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x},\mathbf{z})] + \log p(\mathbf{x}) \\
&= -ELBO + \text{evidence} \\
\leftrightarrow \text{evidence} &= \underbrace{KL\big(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})\big)}_{\text{If this KL-div is too large, ELBO is not tight}} + ELBO
\end{aligned}
$$

  - You maximize ELBO w.r.t. the network parameters instead of directly working on the evidence
  - If the ELBO is not tight, you're solving a wrong problem

- Quality of the prior matters, too

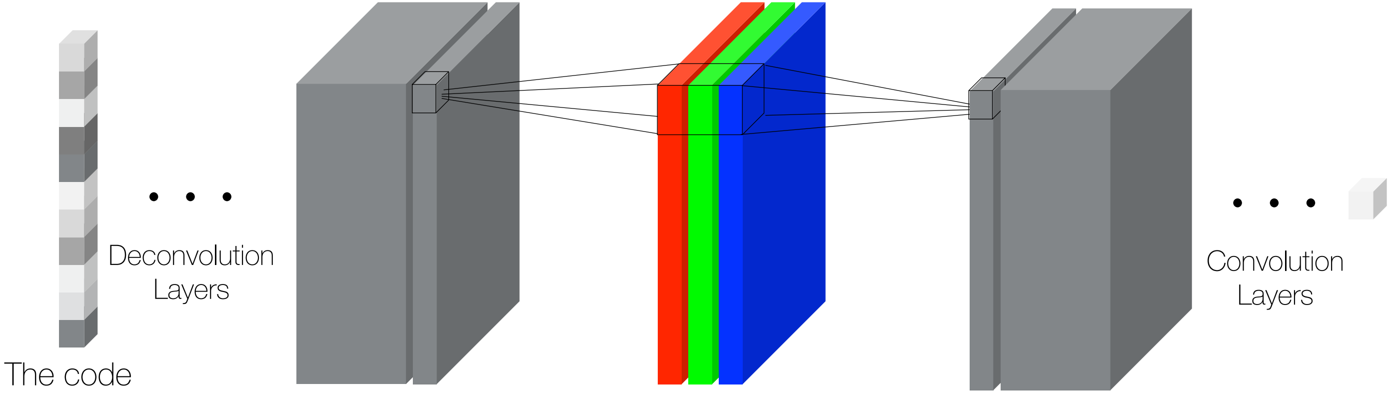
$$
\begin{aligned}
-ELBO(q) &= -\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] + KL\big(q(\mathbf{z})||p(\mathbf{z})\big) \\
-ELBO_i(\boldsymbol{\theta},\boldsymbol{\phi}) &= -\mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{z}_i|\mathbf{x}_i)}[\log p_{\boldsymbol{\phi}}(\mathbf{x}_i|\mathbf{z}_i)] + KL\big(q_{\boldsymbol{\theta}}(\mathbf{z}_i|\mathbf{x}_i)||p(\mathbf{z})\big)
\end{aligned}
$$

  - If the prior distribution doesn't make sense (or too strong), maximizing ELBO doesn't guarantee good reconstruction

- GANs are considered to produce better results
  - Better in the sense of the subjective quality

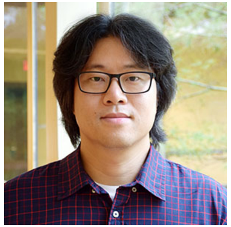
Why the Game Theory?

- One of the main goals of generative models
  - Create an example that makes sense
  - But, makes sense to whom?
    - Maybe to human perception
    - Any other ideas?
- Another network for discrimination
  - Any problem with this?
  - Well, you need to train this discriminator network
  - From what?



Why the Game Theory?

- e.g. A discriminator that identifies ImageNet examples
  - It’s a binary classification task: {ImageNet looking images} versus {some other images}

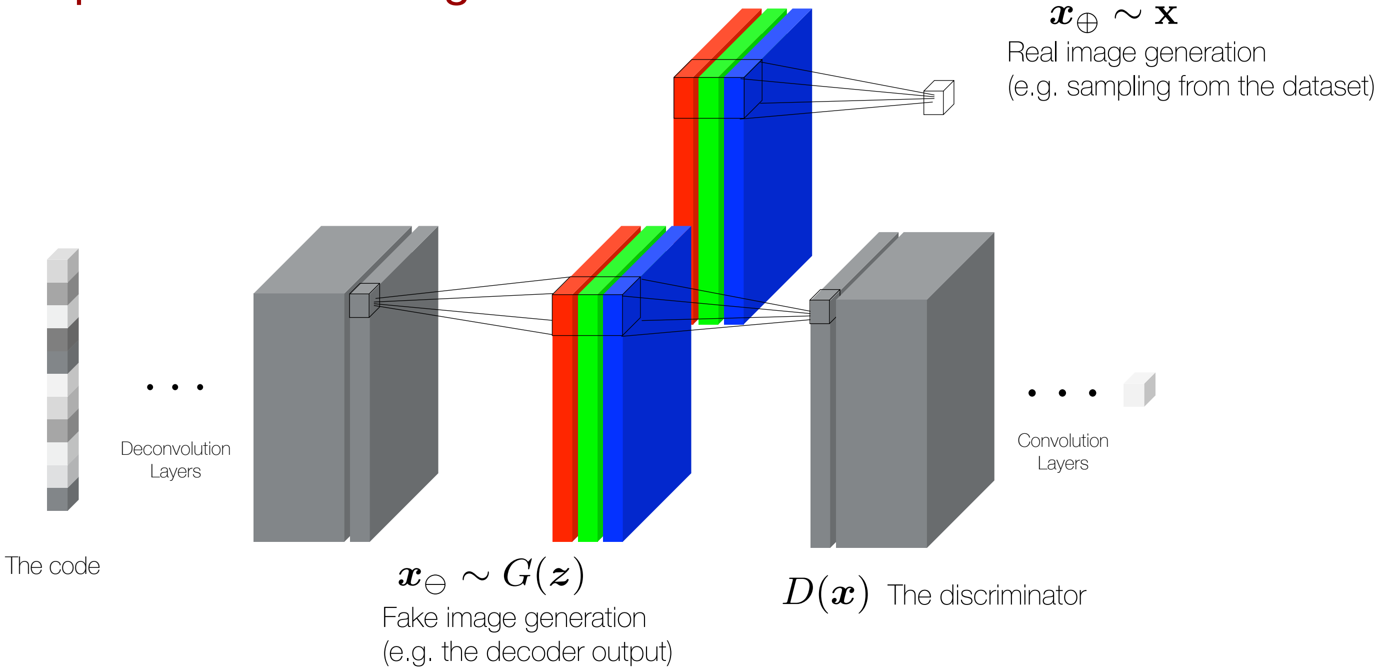


ImageNet?

- What kind of counter examples do I need?
- The goal of your generative model is to fool the discriminator. So, the discriminator had better be?
  - Strong! So that the generator produces high quality examples
- Building a strong discriminator is difficult
  - Difficult to come up with good counter examples

Generative Adversarial Networks

- Competition between generator and discriminator



- The expected behavior of $D(\boldsymbol{x})$

  $D(\boldsymbol{x}_{\oplus}) = 1 \quad D(\boldsymbol{x}_{\ominus}) = 0$

- The expected behavior of $G(\boldsymbol{z})$

  $D(\boldsymbol{x}_{\ominus}) = 1$

Generative Adversarial Networks

- Competition between generator and discriminator

- Competition?
- If the generator does a good job
  - It creates fake examples that look like real
  - Therefore, $D(\boldsymbol{x}_\ominus) \uparrow$
- If the discriminator does a good job
  - It becomes good at distinguishing $\boldsymbol{x}_\oplus$ and $\boldsymbol{x}_\ominus$
  - Therefore, $D(\boldsymbol{x}_\oplus) \uparrow$ and $D(\boldsymbol{x}_\ominus) \downarrow$
- The dilemma
  - To perform better, the discriminator needs the generator to be good
    - So that it's learned from confusing fake examples
  - But, by definition the generator promotes $D(\boldsymbol{x}_\ominus) \uparrow$
    - Harms the performance of the discriminator
- Nash equilibrium

$$D(\boldsymbol{x}_\oplus) = D(\boldsymbol{x}_\ominus) = 0.5$$ (long story short)

Generative Adversarial Networks

- Objectives

- Let's start from the usual cross entropy for the discriminator objective

$$\mathcal{J}^{(D)}(\boldsymbol{\theta}^{(D)}, \boldsymbol{\theta}^{(G)}) = \sum_i -y_i \log D(\boldsymbol{x}_i) - (1 - y_i) \log \left(1 - D(\boldsymbol{x}_i)\right) = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}_\oplus} \log D(\boldsymbol{x}_\oplus) - \frac{1}{2}\mathbb{E}_{\boldsymbol{x}_\ominus} \log \left(1 - D(\boldsymbol{x}_\ominus)\right)$$

1 for real examples
0 for fake examples

Sigmoid output

  - Solution (w.r.t. $\boldsymbol{\theta}^{(D)}$)

$$D^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_{model}(\boldsymbol{x})}$$

- Optimization
  - Minimax $\quad \boldsymbol{\theta}^{(G)*} = \arg\min_{\boldsymbol{\theta}^{(G)}} \max_{\boldsymbol{\theta}^{(D)}} -\mathcal{J}\left(\boldsymbol{\theta}^{(D)}, \boldsymbol{\theta}^{(G)}\right)$
  - A difficult optimization

Generative Adversarial Networks

- Remaining questions
  - How to formulate the latent space
    - Control?
- Optimization is difficult
- Evaluation of generated samples
- Generator cannot produce discrete data

Reading

- Ian Goodfellow, “NIPS 2016 Tutorial: Generative Adversarial Networks”
  - https://arxiv.org/abs/1701.00160
  - https://channel9.msdn.com/Events/Neural-Information-Processing-Systems-Conference/Neural-Information-Processing-Systems-Conference-NIPS-2016/Generative-Adversarial-Networks
- Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.
- https://www.oreilly.com/learning/generative-adversarial-networks-for-beginners
- https://www.youtube.com/watch?v=9c4z6YsBGQ0

Thank You!