Machine Learning (CSE 446):
Generative Adversarial Networks (GANs)

Sham M Kakade
© 2019

University of Washington
cse446-staff@cs.washington.edu
Announcements

- Weds is the final.
- One page of notes.
- List of topics posted tomorrow.
Class results with random Fourier features, HW3, Q7
These are computer generated images from the “bigGAN”.

Figure 1: Class-conditional samples generated by our model.
Classification

- SPAM detection.
- $x$ is an email.
- Suppose $Q(x)$ is the distribution over true emails.
- Suppose $G(x)$ is spammer's generative distribution over emails.
- Usual supervised learning: make a dataset of $(x, y)$ pairs, say with 50% labeled true and 50% spam.
- Suppose $P_{θ}(Y = 1|X) = D_θ(X)$ is our model's probably of that $Y = 1$, true email.

$\Pr(\theta(X) \text{ between } 0 \text{ and } 2)$. 
Building Our Classifier

- Maximum likelihood is:
\[
\max_{\theta} \mathbb{E} \log \Pr_{\theta}(Y = 1|X)
\]
- every \(\theta\) corresponds to some model \(D\).
- If our data is equally split, where all the \(Y = 1\)'s come from \(Q\) and and all the \(Y = 0\)'s come from \(G\), then:
\[
\max_{\theta} \mathbb{E} \log \Pr_{\theta}(Y = 1|X) = \max_D \left( \mathbb{E}_{x \sim Q} \left[ \frac{1}{2} \log D(x) \right] + \frac{1}{2} \mathbb{E}_{x \sim G} \left[ \log(1 - D(x)) \right] \right)
\]
- Likelihood function:
\[
L(D, G) = \left( \frac{1}{2} \mathbb{E}_{x \sim Q} \left[ \log D(x) \right] + \frac{1}{2} \mathbb{E}_{x \sim G} \left[ \log(1 - D(x)) \right] \right)
\]
The Spammer’s Job

- If the spam detector $D$ was fixed, then our spammer wants to:
  $$\min_G L(G, D)$$

- The spammer wants to make the discriminator $D$’s likelihood large.
We have a game:

- Let’s think more abstractly: $D$ is procedure to spot fake images, $G$ is a procedure to generate images.
- The game can be viewed as:

$$
\min_G \max_D L(G, D) = \\
\min_G \max_D \left( \mathbb{E}_{x \sim Q} \left[ \frac{1}{2} \log D(x) \right] + \frac{1}{2} \mathbb{E}_{x \sim G} [\log (1 - D(x))] \right)
$$

- Is this a powerful idea?
- Remember AD: if we can get gradients easily, let’s try to play game.
the Discriminator: learning

- just binary image classification “fake or not”
- we know how to do supervised learning
- for a given $G$ and samples from the truth $Q$, we can learn $D$.

\[ \text{create a labeled dataset} \]
\[ \text{and do binary classification.} \]
what is a model for generating images?
and how do we update $G$ the model?

- let say $z \sim N(0, I)$ where $I$ is a $d \times d$ matrix.
- a network:
the Generator network

input $\rightarrow W_i$

convolution $\rightarrow W_{d_{out}}$

$256 \times 256$ image

output $\rightarrow W_{d_{out}}$

$256$
Learning for $G_0$

- Fix $G$, update $D$.
  easy: just supervised learning.
- Now fix $D$. Recall:
  \[ L(G, D) = \left( \mathbb{E}_{x \sim Q} \left[ \frac{1}{2} \log D(x) \right] + \frac{1}{2} \mathbb{E}_{x \sim G} \left[ \log(1 - D(x)) \right] \right) \]
  \[ \therefore \min_{G} L(G_0, D) = \left( \mathbb{E}_{x \sim Q} \left[ \frac{1}{2} \log D(x) \right] + \frac{1}{2} \mathbb{E}_{z \sim \text{Normal}(0, I)} \left[ \log(1 - D(G(z))) \right] \right) \]

- We can estimate this loss easily. Why?
- Update $G$: how?
  \[ \text{sample } z \text{'s, with SGD, use AD to gradients easily.} \]
Convergence/Comments

- Does it converge?
  Subtle: we are not "hill climbing" on an objective.
- at best, we can get to an equilibrium.
- Comparison to EM and likelihood based approaches:
  - Computing the probability of $x$ under $G$ is difficult.
  - "mode collapse" with GANs
    sampling distribution not reflective of the truth.
- NLP generative methods do not use GANs!
  (better results with direct training approaches)
We can get creative
Style transfer. From the “cycleGAN”
Thank you!!

Thank you for the hard work!

- Good luck on the final and have a great spring break.
- You have a good toolkit.
  Please participate in the larger ML community!