Announcements

1. HW 2 now due Monday, 5/10

Student holiday on Fri!

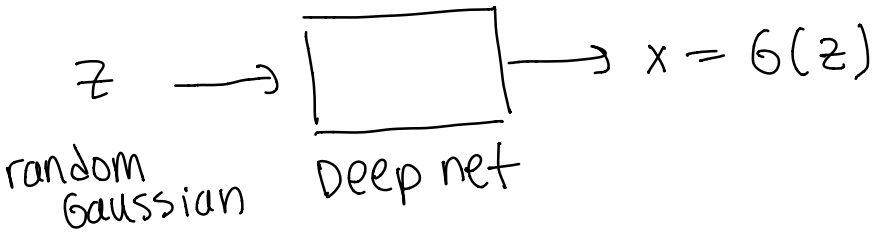
2. still doing project meetings...

Deep Generative Models

Deep learning is useful for more than just supervised learning

Main Question: Can we learn to generate realistic new data?

The usual setup is:



But what does realistic mean?

Goal: No deep network can discriminate between outputs of G and real samples

Goodfellow et al. defined a min-max objective function

$$\min_{u} \max_{v} \mathbb{E}_{x \sim D_{real}}[\log D_v(x)] + \mathbb{E}_{x \sim G_u}[\log 1 - D_v(x)] \quad (*)$$

where:

(1) D_{real} is the distribution on real data

(2) D_{G_u} the distribution on outputs of generator G_u and u are its params

(3) D_v is the discriminator, that tries to output 1 on real data, 0 o.w

Then we try to find an equilibrium, i.e.

"A generator G_u so that no discrim can beat random guessing."

Some things to worry about

(1) How do we solve this optimization problem?

As usual, we use SGD or some other heuristic

(2) what if we only have a finite (polynomial) sized set of samples?

(3) what if instead of converging, we cycling?

Let's come back to these later, and first study the properties at equilibrium

Lemma 1 If D_r is allowed to be any function from $\mathbb{R}^d$ to $[0,1]$ then the optimal choice is:

$$D_V(x) = \frac{P_{real}(x)}{P_{real}(x) + P_{Gu}(x)}$$

where P_{real} and P_{Gu} are the density fnctns of D_{real} and D_{Gu} respectively

In particular, if the supports of D_{real} and D_{Gu} are disjoint, then the optimal D_V achieves zero loss

The intuition behind the proof is simple. Let

H_0: x came from D_{real}

H_1: x came from D_{Gu}

Then the optimal D_V computes the optimal hypothesis test

Our actual proof will use the KL-divergence

def: The KL-divergence between discrete distributions p and q is

$$D_{KL}(p||q) \triangleq \sum_x p(x) \log \frac{p(x)}{q(x)}$$

Or if p and q are continuous

$$D_{KL}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} \, dx$$

we will use the following elementary fact

Fact 1: For distributions p and q, $D_{KL}(p||q) \geq 0$ and equality is achieved iff $p = q$ almost surely

Rearranging things, we get

Corollary 1: For distributions p and q, $\sum_x p(x) \log q(x)$ is maximized for $p = q$

Now we can prove the lemma

Proof: We can rewrite the objective function in (*) as

$$\int \left(P_{real}(x) \log D_v(x) + P_{Gu}(x) \log(1 - D_v(x))\right) dx$$

and we can maximize this (over D_v) on an x-by-x basis, and from Corollary 1 the optimal choice is

$$D_v(x) = \frac{P_{real}(x)}{P_{real}(x) + P_{Gu}(x)}$$

as desired. ▨

Now let's plug in the optimal D_v and see what we get for the optimal choice of the generator D_{Gu}

$$\int p_{real}(x) \log \frac{p_{real}(x)}{p_{real}(x)+p_{Gen}(x)} dx + \int p_{Gen}(x) \log \frac{p_{Gen}(x)}{p_{real}(x)+p_{Gen}(x)} dx$$

$$= D_{KL}\left(p_{real} \,||\, \frac{p_{real}+p_{Gen}}{2}\right) + D_{KL}\left(p_{Gen} \,||\, \frac{p_{real}+p_{Gen}}{2}\right)$$

def: For two distributions p and q, the Jensen-Shannon divergence is

$$D_{JS}(p||q) \triangleq \frac{1}{2} D_{KL}\left(p \,||\, \frac{p+q}{2}\right) + \frac{1}{2} D_{KL}\left(q \,||\, \frac{p+q}{2}\right)$$

Theorem [Goodfellow et al] If D_V is chosen among all functions $\mathbb{R}^d \to [0,1]$, then at equilibrium the optimal generator minimizes $D_{JS}(D_{real} \,||\, D_{Gen})$

Training GANs following (*) is tricky

Main Question: what happens if we change the game? Characterize equilibrium?

Arjovsky et al. first considered:

$$\min_{u} \max_{v} \mathbb{E}_{x \sim D_{real}}[D_v(x)] + \mathbb{E}_{x \sim D_{G_u}}[1 - D_v(x)] \quad (\square)$$

To understand its equilibrium, we will need yet another distance on distributions

def: The Wasserstein distance between p and q is

$$W_1(p, q) = \sup_{D \text{ is 1-lipschitz}} \left| \mathbb{E}_{x \sim p}[D(x)] - \mathbb{E}_{x \sim q}[D(x)] \right|$$

Lets see some examples to get some intuition

Suppose p and q are discrete distributions

the Wasserstein distance measures

$$\text{cost} \triangleq \text{mass} \times \text{distance}$$

of moving p into q

Thus it is a smoother distance, e.g. it distinguishes between

What can we say about equilibrium?

Theorem 2 [Arjovsky et al.] If D_v is chosen optimally all 1-Lipschitz functions, then optimal generator minimizes

$$W_1(D_{real}, D_{Gu})$$

Now let's dig into some of the theoretical issues

Issue #1: We can only estimate

$$\mathbb{E}_{x \sim D_{real}}[\log(D_v(x))], \; D_{JS}(D_{Gu} \| D_{real}), \; W_1(P, Q)$$

from a polynomial # of samples

i.e. does $W_1(P, Q)$ being small imply that $W_1(\hat{P}, \hat{Q})$ is too

($\hat{P}, \hat{Q}$: empirical distributions)

Definitely not!

Observation 1 [Arora et al.] Consider $P = N(0, \frac{1}{d} I)$. Then whp for poly. samples

$$d_{JS}(P \| \hat{P}) = \log 2 \quad \text{(not prob.)}$$

$$W_1(P \| \hat{P}) \geq 1.1$$

Various works have proposed using the sliced Wasserstein distance instead

Arora, Ge, Liang, Ma, Zhang define a distance based on neural networks:

def: Let $\mathcal{F}$ be a class of neural nets with n parameters. Let ϕ be concave

$$D_{\mathcal{F},\phi}(P, Q) \triangleq \sup_{D \in \mathcal{F}} \mathbb{E}_{x \sim P}[\phi(D(x))] + \mathbb{E}_{x \sim Q}[\phi(1 - D(x))] - 2\phi\left(\frac{1}{2}\right)$$

They study generalization and (approximate) equilibria

Theorem 3 [Arora et al]. Suppose that

(1) ϕ is concave, L_ϕ-Lipschitz and takes values in $[-\Delta, \Delta]$

(2) The class of discriminators is L-Lipschitz w.r.t. the parameters v

If we take m samples from p and q with

$$m \geq \frac{C n \Delta^2 \log(L L_\phi n/\varepsilon)}{\varepsilon^2}$$

whp $|d_{F,\phi}(\hat{p}, \hat{q}) - d_{F,\phi}(p, q)| \leq \varepsilon$

Intuition: Both W_1 and $d_{\mathcal{F},\phi}$ were defined in terms of test functions, and for the latter we can bound the compl. using the # of parameters

Main Question: what can we say about the equilibrium?

Theorem [Arora et al., informal] If the generator has $\tilde{\Omega}(n^2)$ parameters, then the generator can win

In particular, we will construct an approximate equilibrium where the discriminator can't do better than $2\phi(\frac{1}{2})$

First we will need to introduce some tools from game theory

def: A zero sum game is defined by

(1) a set U of strategies for Alice

(2) a set V of strategies for Bob

(3) A payoff function $F(u,v)$ that for any $u \in U, v \in V$ describes how much Alice wins / Bob loses

Notice the sum of payoffs is always equal to zero

Von Neumann's minimax theorem is a fundamental result in the area

Theorem [minimax for finite games]
If U and V are finite, there is a pair of distributions p and q on U and V respectively and a param. n called the game value s.t.

(1) $\forall_{v \in V} \; \mathbb{E}_{u \sim p}[F(u,v)] \geq n$ = payoff of p and q

(2) $\forall_{u \in U} \; \mathbb{E}_{v \sim q}[F(u,v)] \leq n$

Informally, Alice and Bob can each guarantee themselves $\pm n$

Finally we call (p, q) an equilibrium

It is the solution to a convex/concave minimax problem, like we had for training GANs

$$\min_{q} \max_{p} \mathbb{E}_{\substack{u \sim p \\ v \sim q}} [F(u, v)]$$ (or other way around)

Moreover if (1) and (2) hold only up to $\pm \varepsilon$, we call it an ε-approx. equilibrium

Note: There are versions of the minimax theorem that work with ∞ strategy spaces, you need additional conditions

An important result of Lipton, Markakis and Mehta is:

Theorem 4 [Lipton et al.]: For any zero sum game with $|u|=m, |v|=n$ $\exists \hat{p}, \hat{q}$ satisfying

(1) $\hat{p}, \hat{q}$ is an ε-approx. equilibrium

(2) $\hat{p}$ has $O\left(\frac{\log n}{\varepsilon^2}\right)$ strategies in its support

$\hat{q}$ has $O\left(\frac{\log m}{\varepsilon^2}\right)$ "

The proof is by subsampling

Proof: Let (p, q) be an equilibrium.

Then let

$\hat{p}$ = empirical distribution of $s = O(\frac{\log n}{\varepsilon^2})$ i.i.d draws from p

and similarly for $\hat{q}$.

Then by standard concentration bds, no $v \in V$ will do more than ε-better on avg. playing against $\hat{p}$ compared to p

The same holds for any $u \in U$ and $\hat{q}$ and q. ▨

Now we will explain the main ingredients that go into analyzing GANs

First, you need the assumption

> "The generator can approximate any point mass."

i.e. for any x, $\exists$ generator G_u with

$$\underset{y \sim D_{G_u}}{\mathbb{E}} [\|x - y\|] \leq \varepsilon$$

Now the reasoning goes

(1) There is a mixed strategy for the generator that no discrim. can succeed against

This is true because you can represent D_{real}

(2) There is a set of $\tilde{O}(n)$ generators where the uniform distribution fools any discriminator

This relies on discretizing the strategy space of discriminators and subsampling mixed strategy as before

(3) Can fold the generators into a larger deep net that uses the randomness in z to select uniformly from them

Applications

Deep generative models can be used in downstream applications as more realistic models for real world inputs

Let's study compressed sensing

Setup: Unknown $x \in \mathbb{R}^d$

We get linear measurements $\underset{m \times d}{A}x = b$, where both A and b are known

How many rows (i.e. measurements) do we need to recover x?

Claim: If x is arbitrary, then we need A to have full column rank

But what happens if x is structured (e.g. sparse)?

Suppose x has at most k nonzeros

Theorem [Donoho; Candes, Romberg, Tao]

For a random A ($m \times d$, standard normal) with

$$m \geq C k \log \frac{d}{k}$$

then whp can recover a k-sparse x exactly

In fact, the following algorithm works:

$$\min \|z\|_1 \text{ s.t. } Az = b$$

and it is stable in the presence of noise/model misspecification

i.e. it'll approximately recover the k largest coordinates in x

Implications: Can cutting radiation you're exposed to in an MRI by 90%

Food for Thought: Is sparsity a reasonable assumption for natural images?

It is a good starting point, but it turns out we can do much better!

In an influential work, Bora, Jalal, Price, Dimakis studied compressed sensing w/ generative models

Setup: Unknown $x = G(z)$, where $G: \mathbb{R}^k \to \mathbb{R}^d$ is an L-layer network with ReLU activations

We observe $Ax + \eta = b$

Main Question: How few measurements do we need to accurately recover x?

They proved:

Theorem [Bora et al]: Suppose A is a random $m \times d$ matrix with

standard normal entries. Further suppose

$$m \geq C L k \log d$$

Then whp the estimator

$$\hat{z} = \arg\min_{z} \| b - A G(z) \|_2$$

satisfies $\| G(\hat{z}) - x \|_2 \leq 6 \min_{z^*} \| G(z^*) - x \|_2 + 3 \| \eta \|_2$

Note: x can be anything — need not be a valid output of G. Thus we can approximate the best fit from few noisy linear measurements

Many of the usual notions from vanilla compressed sensing carry over with some twists

Vanilla Compressed Sensing

Main Question: why can't there be two sparse solutions x and x' that both fit the data?

The key is that random matrices of the right dimensions satisfy the following conditions whp

def: we say that A satisfies the $(2k, \delta)$-restricted isometry property if for all z with at most $2k$ nonzeros

$$(1-\delta)\|z\|_2^2 \leq \|Az\|_2^2 \leq (1+\delta)\|z\|_2^2$$

So if we have two sparse candidate solutions $x \neq x'$, they can't both fit our observations (even approximately)

For the sake of contradiction, suppose they do, i.e.

$$\|Ax - Ax'\|_2 \ll 1 - \delta$$

Then we can choose $z = x - x'$ and

$$\|Az\|_2 \text{ is small}$$

which would violate the $(2k, \delta)$-RIP

Fact 2: A random $m \times d$ matrix A with constant x standard normal entries

$$m \geq ck \log \frac{d}{k} \Rightarrow (2k, 1/3)\text{-RIP whp}$$

Taking a step back:

RIP condition $\Longleftrightarrow$ compatibility btwn models for x's and measurements

It turns out for more complex models, we can tweak the notion of RIP/REC

def: A matrix A satisfies the set restricted eigenvalue condition for a set $S \subseteq \mathbb{R}^d$ with parameters γ and δ if $\forall x, x' \in S$ we have

$$\|A(x-x')\|_2 \geq \gamma \|x-x'\|_2 - \delta$$

They set $S = G(B^k(R))$

$B^k(R)$: ball of radius R in k-dimensions

Now the main ideas are

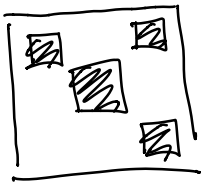
(1) A random matrix of appropriate size satisfies S-REC

This is the analogue of Fact 2

(2) S-REC guarantees that $\hat{z}$ achieves the desired accuracy

But how do you find $\hat{z}$? SGD

Applications: In image inpainting, you are missing many of the pixels



You can model this as observing

$$A\underbrace{G(z)}_{\text{natural image}}$$

where the rows of A correspond to observed pixels

By finding $\hat{z}$, can compute

$$\hat{x} = \text{inpainting} = G(\hat{z})$$

This leads to state-of-the-art results and some amazing pictures

Moral: Deep generative models can be a powerful replacement for simpler assumptions about the structure of realistic data