10-414/714 – Deep Learning Systems: Algorithms and Implementation

Generative Adversarial Networks

Fall 2022 J. Zico Kolter and Tianqi Chen (this time) Carnegie Mellon University

Outline

Generative adversarial training

Adversarial training as a module in deep learning models

Outline

Generative adversarial training

Adversarial training as a module in deep learning models

From classifier to generator



Goal: make generated the distribution of generated samples "close" to target data distribution

4

Define "distance" of distributions

Unlike supervised classification setting, the "goal" is less obvious

To build effective training mechanism, we need to define a "distance" between generated and real datasets and use that to drive the training.

What we really wanted, in text: make sure that the generated samples "looks real".

Generated samples

	٥	0	>		1	a	÷.	5	0
1	7	0	•	?	1	1	1		1
1		G.	4	Ø	7	7	9	1	3
4	3	4	2	9	0	÷	7	7	1
7	7	8	3	0	4	ŀ	3	6	7
5	8	7	9	1	8	1)	6	7
7	3	6	١	()	1	8	9	9	4
6	1	9	3	5	/	2	7	Y	9
4	8	1	1	3	9	7	3	0	E
9	1	3	.9	4	7	3	1	5	3
1	5	1	2	3	0	3	3	1	7

Target data distribution

6	7	2	ł	3	З	0	6	9	1
9	6	О	5	3	9	9	8	9	ł
3	ł	6	9	2	J	R.	3	5	4
6	4	1	2	З	7	8	4	6	0
\mathcal{D}	ما	4	6	3	3	1	R	6	3
5	ス	5	0	1	3	0	7	4	λ
3	3	2	9	7	4	0	ລ	9	4
8	7	3	7	2	9	9	6	ን	P
1	9	0	2	0	4	3	4	9	4
4	2	2	5	4	3	5	1	7	1

Learn generator through an oracle discriminator



Assume that we have an oracle discriminator that can tell the difference between real and fake data. Then we need train the generator to "fool" the oracle discriminator. We need to maximize the discriminator loss

```
Generator objective: max_G\{-E_{z \sim Noise}\log(1 - D(G(z))\}
```

Learning the discriminator



We do not have an oracle discriminator, but we can learn it using the real and generated fake data.

Discriminator objective $min_D\{-E_{x\sim Data} \log D(x) - E_{z\sim Noise} \log(1 - D(G(z)))\}$

Generative adversarial network



Putting it together, it becomes an "minimax" game between D and G

$$min_D max_G \{-E_{x\sim Data} \log D(x) - E_{z\sim Noise} \log(1 - D(G(z)))\}$$

In practice, we usually optimize *G* using $min_G\{-E_{z\sim Noise} \log(D(G(z))\}\}$, maximize the probability that discriminator predicts generated image is real

Generative adversarial training in practice



Iterative process

- Discriminator update
 - Sample minibatch of D(G(z)), get a minibatch of D(x)
 - Update D to minimize $min_D\{-E_{x\sim Data}\log D(x) E_{z\sim Noise}\log(1 D(G(z)))\}$
- Generator update
 - Sample minibatch of D(G(z))
 - Update G to minimize $min_G\{-E_{z\sim Noise} \log(D(G(z))\}\)$, this can be done by feeding label=1 to to the model

Outline

Generative adversarial training

Adversarial training as a module in deep learning models

Deep learning is modular in nature

Multi-layer Residual Net



Residual block

transpose

 h_2

 W_1



 $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$

Use GAN as a compositional module



GAN is not exactly like a loss function, as it involves an iterative update recipe. But we can compose it with other neural network modules in a similar way like loss function.

Use GAN "loss" whenever we want a collection of data to "look like" another collection

DCGAN: Deep convolutional generative adversarial networks



Convolutional units with Conv2dTranpose

Discriminator



Prediction

Fake image

CycleGAN: Image to image translation



Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ICCV 2017

CycleGAN: Structure



The goal of CycleGAN is to learn bi-directional translator between two **unpaired** collections of data

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ICCV 2017

CycleGAN: The model structure



CycleGAN: The model structure



Cycle consistency: map forward and back should map back to the original image



CycleGAN: Putting it together



Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ICCV 2017

Some results



Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ICCV 2017

Discussion

What are other ways to compose GAN module together with other deep learning components?

Outline

Generative adversarial training

Adversarial training as a module in deep learning models