IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models

Thesis introduction

https://arxiv.org/abs/1705.10513

SIGIR 2017

Takahiro Ishikawa

takahish.nil@gmail.com

2017

Agenda

1. Introduction
2. What is GAN?
3. IRGAN: GAN for IR
4. Additional experiment
5. Appendix

Agenda

1. **Introduction**
2. What is GAN?
3. IRGAN: GAN for IR
4. Additional experiment
5. Appendix

Introduction

- **Information retrieval (IR)** is to provide a rank list of documents given a query.
  - Web search
  - Recommender systems
  - Question answering etc.
- How a document is generated from a given information.
  - q → d, where q is the query
    (e.g., keywords, user profiles, questions, depending on the application)

Introduction

- The method of IR shifts to a **discriminative** (classification) learned from labelled relevant judgements.
  - q + d → s, where s denotes relevance score (and + denotes the combining of features)
  - They lack a principled way of obtaining useful features from massive unlabelled data available
- While the **generative** model of IR are very successful in modeling features
  - They suffer from the difficulty in relevancy signals from links, clicks etc.
  - This signals in Internet based applications.

Introduction

- **IRGAN** is considering the generative and discriminative retrieval models as two sides of the same coin.

  - Inspired by **Generative Adversarial Nets (GANs)**.

  - Objective functions including aspects of IR are defined for both model.

  - “IRGAN” means “GAN for IR”.

Agenda

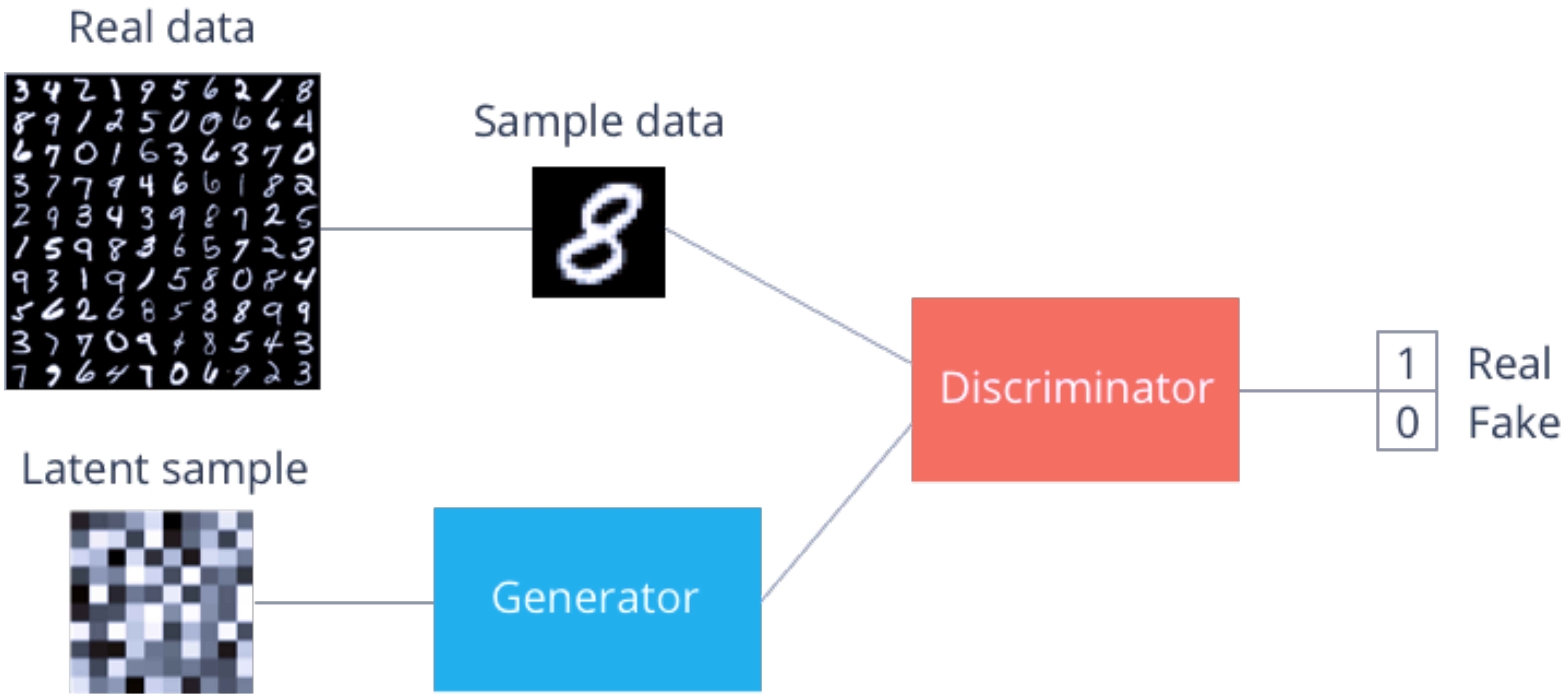
1. Introduction
2. **What is GAN?**
3. IRGAN: GAN for IR
4. Additional experiment
5. Appendix

What is GAN?

- GAN is “Generative Adversarial Network”.
- The original paper is presented by Ian Goodfellow et al, https://arxiv.org/abs/1406.2661.
- He has published a famous textbook of deep learning, http://www.deeplearningbook.org/.

Source: Deep Learning Nanodegree Foundation, Udacity

GAN for Image generation

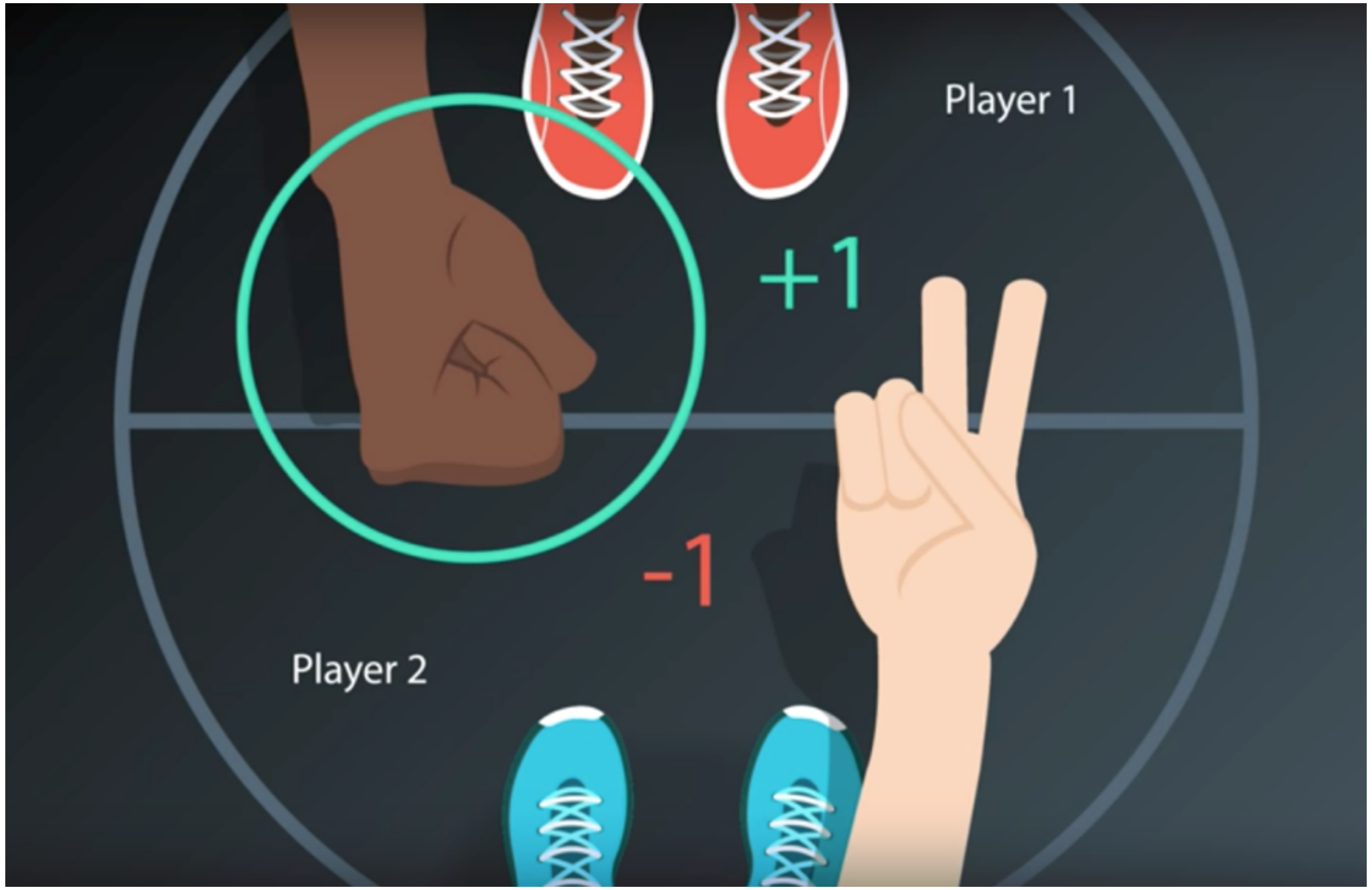


Source: Deep Learning Nanodegree Foundation, Udacity

This diagram shows GAN for MINIST datasets.

Generator is a generative model, Discriminator is a binary classifier.

Minimax game

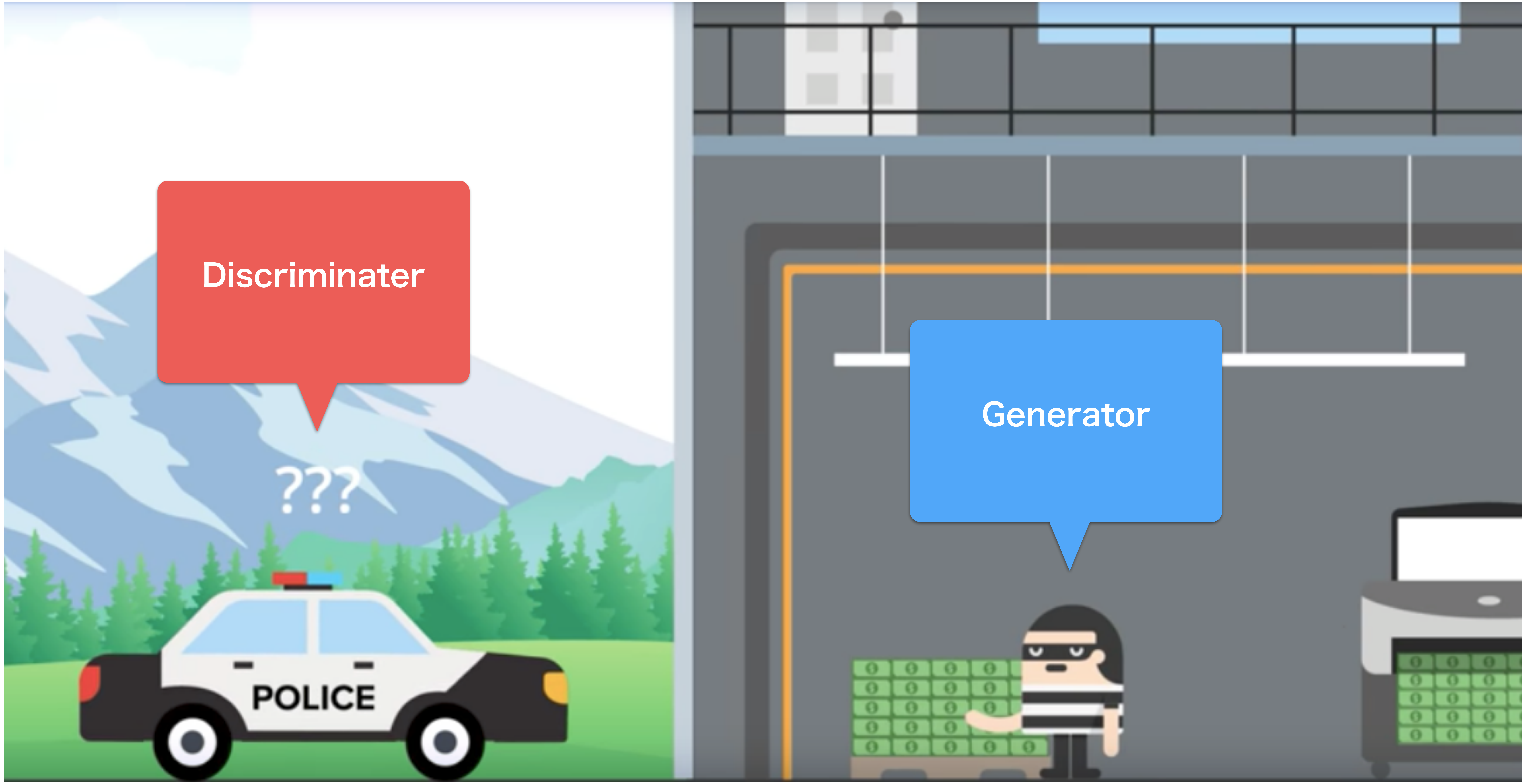


Source: Deep Learning Nanodegree Foundation, Udacity

Minimax game is zero-sum game.

The minimax solution is the same as the Nash equilibrium.

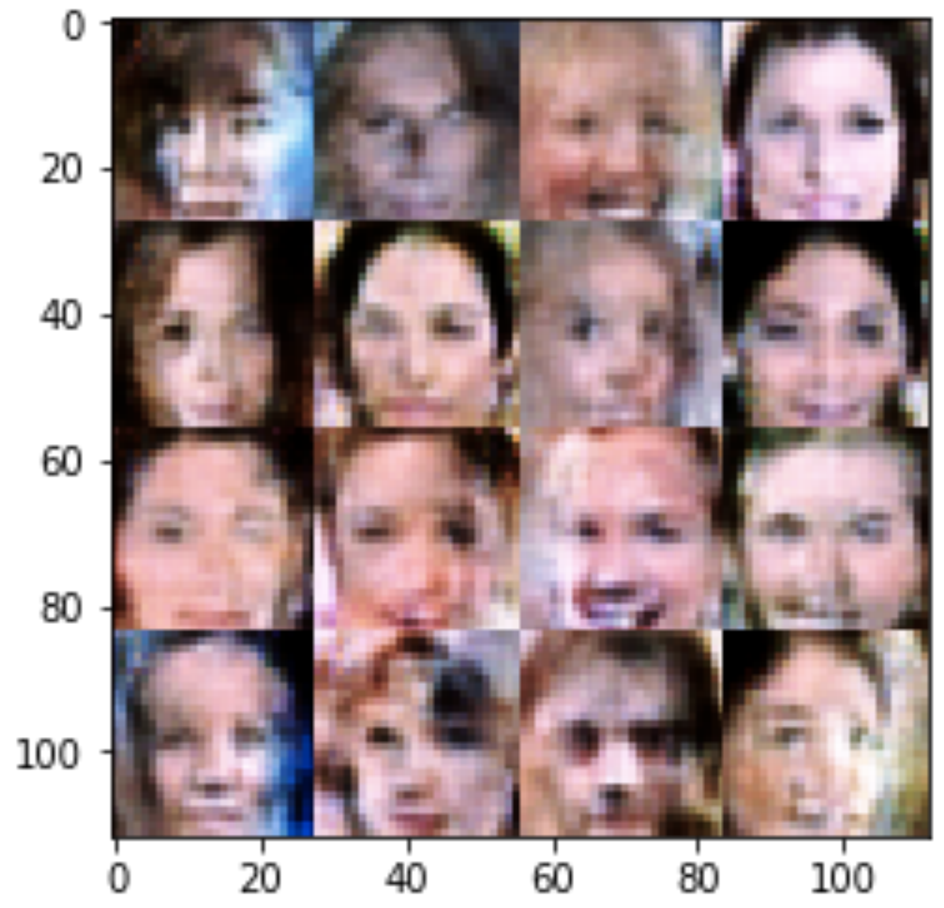
What is GAN?



Source: Deep Learning Nanodegree Foundation, Udacity

GAN explode in popularity

- DCGAN https://arxiv.org/pdf/1511.06434.pdf
- Pix2Pix https://affinelayer.com/pixsrv/
- CycleGAN https://github.com/junyanz/CycleGAN



Source: DCGAN face generation


Source: CycleGAN

Agenda

1. Introduction

2. What is GAN?

3. IRGAN: GAN for IR

4. Additional experiment

5. Appendix

A minimax retrieval framework

- ## Generative retrieval model
  - $p_\theta(d|q, r)$, which tries to select relevant documents, from candidate pool for the given query q.
  - “r” is relevancy between a query and a document.
- ## Discriminative retrieval model
  - $f_\phi(q, d)$, which tries to discriminate well-matched query-document tuples (q, d) from ill-matched ones.
  - It is in fact simply a binary classifier (positive or negative).

Overall objective function

A expected value to classify a true data to positive.

$$J^{G^*,D^*} = \min_{\theta} \max_{\phi} \sum_{n=1}^{N} \Big(\underline{\mathbb{E}_{d \sim p_{\text{true}}(d|q_n,r)} \left[\log D(d|q_n)\right]} + \underline{\mathbb{E}_{d \sim p_{\theta}(d|q_n,r)} \left[\log(1 - D(d|q_n))\right]} \Big)$$

A expected value to classify a fake data of generator to negative.

D(d|q) is the probability which is given by the sigmoid function.

Optimizing Discriminative Retrieval

$$\phi^* = \arg\max_{\phi} \sum_{n=1}^{N} \Big(\mathbb{E}_{d \sim p_{\text{true}}(d|q_n, r)} \left[\log(\sigma(f_\phi(d, q_n))) \right] + \mathbb{E}_{d \sim p_{\theta^*}(d|q_n, r)} \left[\log(1 - \sigma(f_\phi(d, q_n))) \right] \Big)$$

θ^*: currently generator params

$\sigma(f_\phi(d, q_n))$: $D(d|q_n)$ in the previous slide

The objective for the discriminator is to maximize the log-likelihood of currently distinguishing the true and generated documents.

Optimizing Generative Retrieval

$$\theta^* = \arg\min_\theta \sum_{n=1}^{N} \Big(\mathbb{E}_{d\sim p_{\text{true}}(d|q_n,r)} \left[\log \sigma(f_\phi(d,q_n))\right] + \mathbb{E}_{d\sim p_\theta(d|q_n,r)} \left[\log(1-\sigma(f_\phi(d,q_n)))\right] \Big)$$

$$= \arg\max_\theta \sum_{n=1}^{N} \underbrace{\mathbb{E}_{d\sim p_\theta(d|q_n,r)} \left[\log(1+\exp(f_\phi(d,q_n)))\right]}_{\text{denoted as } J^G(q_n)}$$

$f_\phi(q, d)$ is fixed after its maximisation

The generative retrieval model intends to minimize the objective, which fits the underlying relevance distribution over document.

Extension to Pairwise Case

$$D(\langle d_u, d_v \rangle | q) = \sigma(f_\phi(d_u, q) - f_\phi(d_v, q))$$
$$= \frac{\exp(f_\phi(d_u, q) - f_\phi(d_v, q))}{1 + \exp(f_\phi(d_u, q) - f_\phi(d_v, q))} = \frac{1}{1 + \exp(-z)}$$

where $z = f_\phi(d_u, q) - f_\phi(d_v, q)$

$$J^{G^*, D^*} = \min_\theta \max_\phi \sum_{n=1}^{N} \Big(\mathbb{E}_{\mathbf{o} \sim p_{\text{true}}(\mathbf{o}|q_n)} \left[\log D(\mathbf{o}|q_n)\right] + \mathbb{E}_{\mathbf{o}' \sim p_\theta(\mathbf{o}'|q_n)} \left[\log(1 - D(\mathbf{o}'|q_n))\right] \Big)$$

where

$o = \langle d_u, d_v \rangle$ is true documents pair

$o' = \langle d'_u, d'_v \rangle$ is generated documents pair

Application

- The implementation two scoring functions
  - g_θ is used in generator (e.g., $p_\theta(d|q, r) = \sigma(g_\theta(d, q))$).
  - f_θ is used in discriminator.
  - $g_\theta(d, q)$ and $f_\theta(q, d)$ are task-specific scoring functions.
  - g_θ and f_θ is implemented as a three-layer neural net etc.
  - $g_\theta(q, d) = s_\theta(q, d)$ and $f_\phi(q, d) = s_\phi(q, d)$

Scoring functions

- Web Search

$$s(q, d) = \boldsymbol{w}_2^\top \tanh(\boldsymbol{W}_1 \boldsymbol{x}_{q,d} + \boldsymbol{b}_1) + w_0$$

- Item Recommendation

$$s(u, i) = b_i + \boldsymbol{v}_u^\top \boldsymbol{v}_i$$

- Question Answering

$$s(q, a) = \cos(\boldsymbol{v}_q, \boldsymbol{v}_a) = \frac{\boldsymbol{v}_q^\top \boldsymbol{v}_a}{|\boldsymbol{v}_q| \cdot |\boldsymbol{v}_a|}$$

Experiment: Web Search

	P@3	P@5	P@10	MAP
MLE	0.1556	0.1295	0.1029	0.1604
RankNet [3]	0.1619	0.1219	0.1010	0.1517
LambdaRank [5]	0.1651	0.1352	0.1076	0.1658
LambdaMART [4]	0.1368	0.1026	0.0846	0.1288
IRGAN-pointwise	0.1714	0.1657	**0.1257**	**0.1915**
IRGAN-pairwise	**0.2000**	**0.1676**	0.1248	0.1816
Impv-pointwise	3.82%	22.56%*	16.82%*	15.50%*
Impv-pairwise	21.14%*	23.96%*	15.98%	9.53%
	NDCG@3	**NDCG@5**	**NDCG@10**	**MRR**
MLE	0.1893	0.1854	0.2054	0.3194
RankNet [3]	0.1801	0.1709	0.1943	0.3062
LambdaRank [5]	0.1926	0.1920	0.2093	0.3242
LambdaMART [4]	0.1573	0.1456	0.1627	0.2696
IRGAN-pointwise	0.2065	**0.2225**	**0.2483**	**0.3508**
IRGAN-pairwise	**0.2148**	0.2154	0.2380	0.3322
Impv-pointwise	7.22%	15.89%	18.63%	8.20%
Impv-pairwise	11.53%	12.19%	13.71%	2.47%

Ranking performance comparison on the MQ2008-semi dataset, where * means a significant improvement (Wilcoxon signed-rank test)

Experiment: Item Recommendation

	P@3	P@5	P@10	MAP
MLE	0.3369	0.3013	0.2559	0.2005
BPR [34]	0.3289	0.3044	0.2656	0.2009
LambdaFM [46]	0.3845	0.3474	0.2967	0.2222
IRGAN-pointwise	0.4072	0.3750	0.3140	0.2418
Impv-pointwise	5.90%*	7.94%*	5.83%*	8.82%*
	NDCG@3	**NDCG@5**	**NDCG@10**	**MRR**
MLE	0.3461	0.3236	0.3017	0.5264
BPR [34]	0.3410	0.3245	0.3076	0.5290
LambdaFM [46]	0.3986	0.3749	0.3518	0.5797
IRGAN-pointwise	0.4222	0.4009	0.3723	0.6082
Impv-pointwise	5.92%*	6.94%*	5.83%*	4.92%*

Performance comparison on the Movielens dataset,
4-star and 5-star ratings is regarded as positive feedback.

Experiment: Item Recommendation

	P@3	P@5	P@10	MAP
MLE	0.2941	0.2945	0.2777	0.0957
BPR [34]	0.3040	0.2933	0.2774	0.0935
LambdaFM [46]	0.3901	0.3790	0.3489	0.1672
IRGAN-pointwise	0.4456	0.4335	0.3923	0.1720
Impv-pointwise	14.23%*	14.38%*	12.44%*	2.87%*
	NDCG@3	**NDCG@5**	**NDCG@10**	**MRR**
MLE	0.3032	0.3011	0.2878	0.5085
BPR [34]	0.3077	0.2993	0.2866	0.5040
LambdaFM [46]	0.3942	0.3854	0.3624	0.5857
IRGAN-pointwise	0.4498	0.4404	0.4097	0.6371
Impv-pointwise	14.10%*	14.27%*	13.05%*	8.78%*

Performance comparison on the Netflix dataset,
5-star ratings is regarded as positive feedback.

Experiment: Question Answering

	test-1	test-2
QA-CNN [9]	0.6133	0.5689
LambdaCNN [9, 51]	0.6294	0.6006
IRGAN-pairwise	0.6444	0.6111
Impv-pairwise	2.38%*	1.75%

Performance comparison on the InsuranceQA dataset,
It is expected to find one and only real answer
from 500 candidate answers under the Precision@1.

Agenda

1. Introduction

2. What is GAN?

3. IRGAN: GAN for IR

4. Additional experiment

5. Appendix

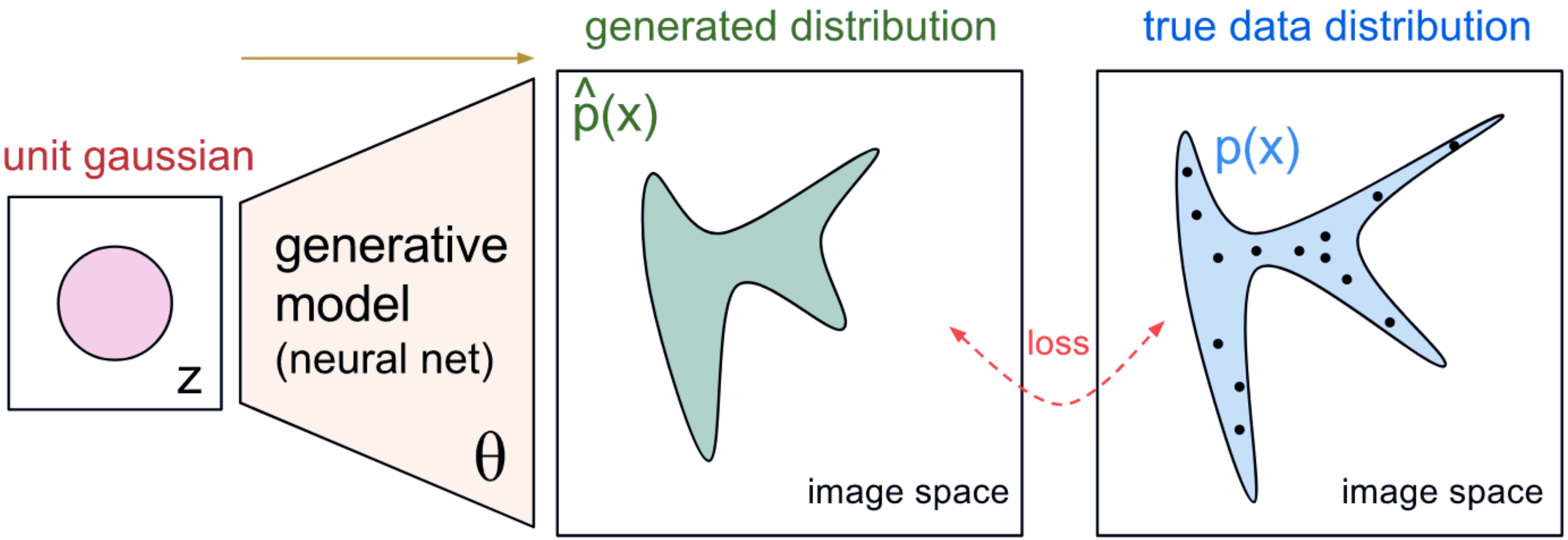
Additional experiment

- Implementation using TensorFlow
    - IRGAN Learning to Rank Pairwise

Agenda

1. Introduction
2. What is GAN?
3. IRGAN: GAN for IR
4. Additional experiment
5. **Appendix**

What is a generative model?

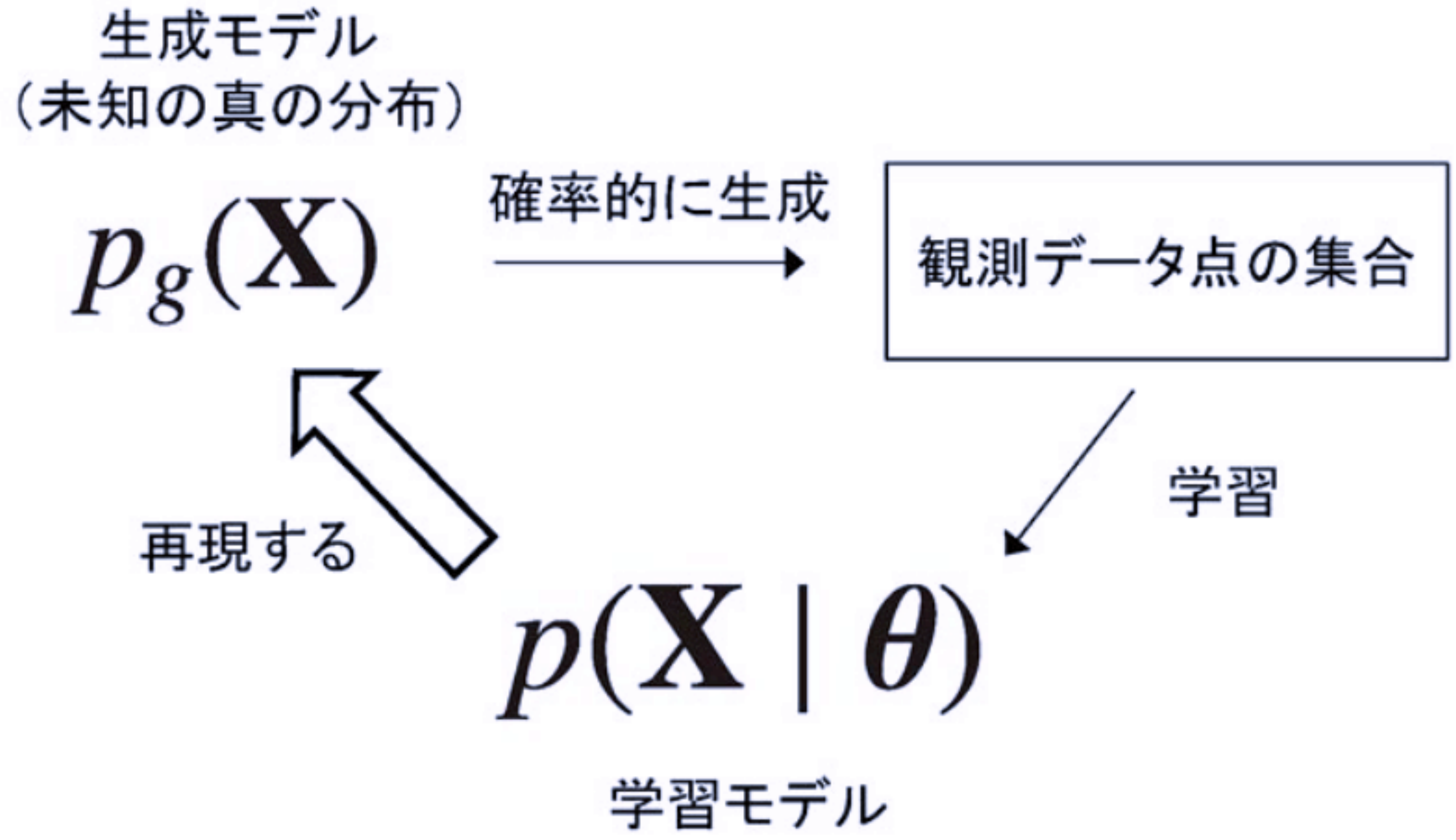


Source: https://blog.openai.com/generative-models/

"A generative model is a model for generating all values for a phenomenon that can be observed in the world."

Source: https://en.wikipedia.org/wiki/Machine_learning

What is a generative model?



Source:『深層学習 Deep Learning (監修:人工知能学会)』

GAN's objective function

A expected value to classify a true data to positive.

$$\min_G \max_D V(D,G) = \underbrace{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]} + \underbrace{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}.$$

A expected value to classify a fake data of generator to negative.

D and G play the two-player minimax game.

A approximation of generator gradient

$$
\begin{aligned}
&\nabla_\theta J^G(q_n) \\
&= \nabla_\theta \mathbb{E}_{d\sim p_\theta(d|q_n,r)}\left[\log(1+\exp(f_\phi(d,q_n)))\right] \\
&= \sum_{i=1}^{M} \nabla_\theta p_\theta(d_i|q_n,r)\log(1+\exp(f_\phi(d_i,q_n))) \\
&= \sum_{i=1}^{M} p_\theta(d_i|q_n,r)\nabla_\theta \log p_\theta(d_i|q_n,r)\log(1+\exp(f_\phi(d_i,q_n))) \\
&= \mathbb{E}_{d\sim p_\theta(d|q_n,r)}\left[\nabla_\theta \log p_\theta(d|q_n,r)\log(1+\exp(f_\phi(d,q_n)))\right] \\
&\simeq \frac{1}{K}\sum_{k=1}^{K} \nabla_\theta \log p_\theta(d_k|q_n,r)\log(1+\exp(f_\phi(d_k,q_n)))\,,
\end{aligned}
$$

d_k is the k-th document sampled from the current version of generator $p(d|q_n, r)$.