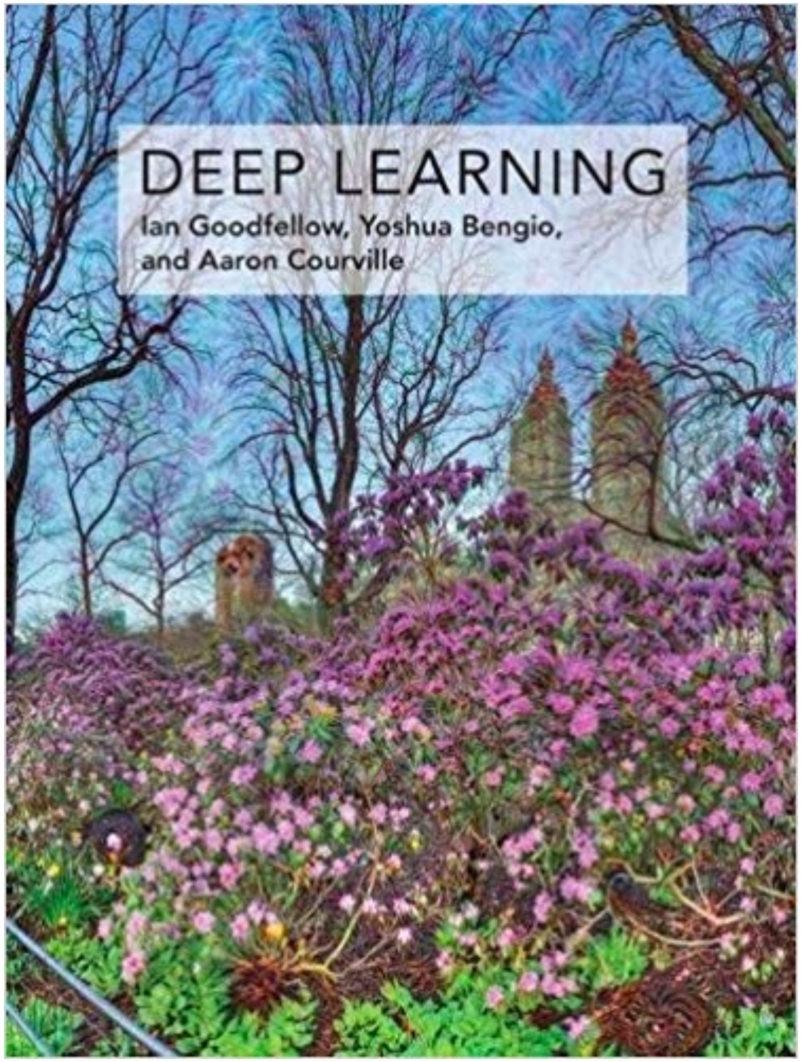
國立陽明交通大學
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

Deep Learning
深度學習
Fall 2023

Generative Adversarial Networks

Prof. Chia-Han Lee
李佳翰 教授



- Figure source: Textbook and Internet
- You are encouraged to buy the textbook.
- Please respect the copyright of the textbook. Do not distribute the materials to other people.

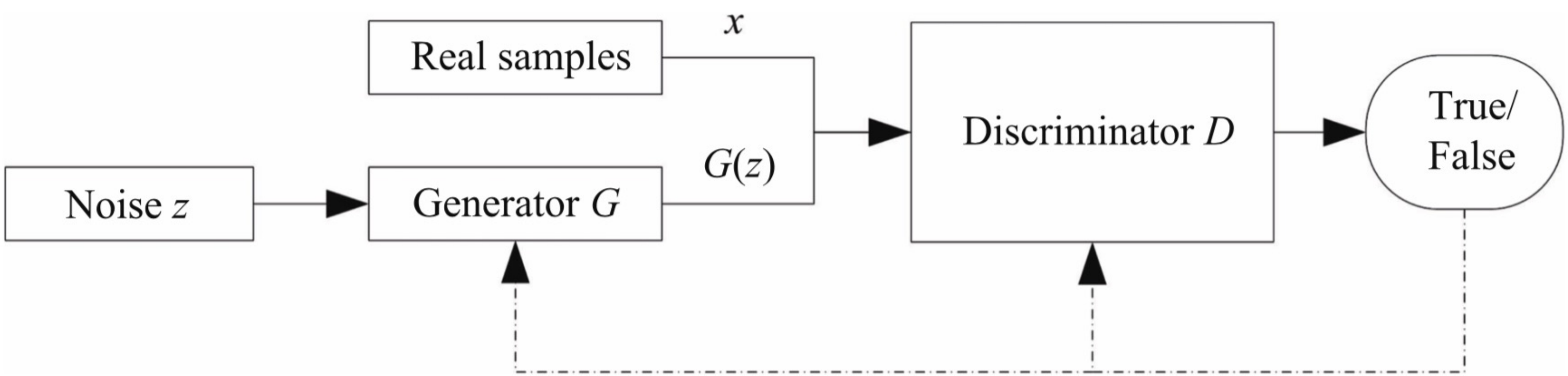
Generative adversarial networks

- Generative adversarial networks are based on a game theoretic scenario in which the generator network must compete against an adversary.

- The generator network directly produces samples $\boldsymbol{x} = g(\boldsymbol{z}; \boldsymbol{\theta}^{(g)})$. Its adversary, the discriminator network, attempts to distinguish between samples drawn from the training data and samples drawn from the generator.

- The discriminator emits a probability value given by $d(\boldsymbol{x}; \boldsymbol{\theta}^{(d)})$, indicating the probability that $\boldsymbol{x}$ is a real training example rather than a fake sample drawn from the model.

Generative adversarial networks



K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F. Y. Wang, "Generative adversarial networks: introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, Sept. 2017.

Generative adversarial networks

- The simplest way to formulate learning in generative adversarial networks is as a zero-sum game. A function $v(\boldsymbol{\theta}^{(g)}, \boldsymbol{\theta}^{(d)})$ determines the payoff of the discriminator, and $-v(\boldsymbol{\theta}^{(g)}, \boldsymbol{\theta}^{(d)})$ is the payoff of the generator.

- During learning, each player attempts to maximize its own payoff, so that at convergence

$$g^* = \arg\min_g \max_d v(g, d). \tag{20.80}$$

$$v(\boldsymbol{\theta}^{(g)}, \boldsymbol{\theta}^{(d)}) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}} \log d(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x}\sim p_{\text{model}}} \log(1 - d(\boldsymbol{x}))\,. \tag{20.81}$$

- The discriminator attempts to learn to correctly classify samples as real or fake, while the generator attempts to fool the classifier into believing its samples are real. At convergence, the generator's samples are indistinguishable from real data, and the discriminator outputs 1/2 everywhere.

Generative adversarial networks

- The main motivation for the design of GANs is that the learning process requires neither approximate inference nor approximation of a partition function gradient.

- When $\max_d v(g, d)$ is convex in $\boldsymbol{\theta}^{(g)}$, the procedure is guaranteed to converge and is asymptotically consistent.

- Unfortunately, learning in GANs can be difficult in practice when g and d are represented by neural networks and $\max_d v(g, d)$ is not convex. Nonconvergence was identified an issue that may cause GANs to underfit.

Generative adversarial networks

- In general, simultaneous gradient descent on two players' costs is not guaranteed to reach an equilibrium.

- Consider, for example, the value function $v(a, b) = ab$, where one player controls a and incurs cost ab, while the other player controls b and receives a cost $-ab$.

- If we model each player as making infinitesimally small gradient steps, each player reducing their own cost at the expense of the other player, then a and b go into a stable, circular orbit, rather than arriving at the equilibrium point at the origin.

Generative adversarial networks

- Note that the equilibria for a minimax game are not local minima of v. Instead, they are points that are simultaneously minima for both players' costs.

- This means that they are saddle points of v that are local minima with respect to the first player's parameters and local maxima with respect to the second player's parameters.

- It is possible for the two players to take turns increasing then decreasing v forever, rather than landing exactly on the saddle point, where neither player is capable of reducing its cost. It is not known to what extent this nonconvergence problem affects GANs.

Generative adversarial networks

- An alternative formulation of the payoffs was identified, in which the game is no longer zero-sum, that has the same expected gradient as maximum likelihood learning whenever the discriminator is optimal.

- Because maximum likelihood training converges, this reformulation of the GAN game should also converge, given enough samples.

- Unfortunately, this alternative formulation does not seem to improve convergence in practice, possibly because of suboptimality of the discriminator or high variance around the expected gradient.

Generative adversarial networks

- In realistic experiments, the best-performing formulation of the GAN game is a different formulation that is neither zero-sum nor equivalent to maximum likelihood.

- In this best-performing formulation, the generator aims to increase the log-probability that the discriminator makes a mistake, rather than aiming to decrease the log-probability that the discriminator makes the correct prediction.

- This reformulation is motivated solely by the observation that it causes the derivative of the generator's cost function with respect to the discriminator's logits to remain large even in the situation when the discriminator confidently rejects all generator samples.