

人工智能博弈

主讲：郭春乐、刘夏雷
南开大学计算机学院

致谢：本课件主要内容来自浙江大学吴飞教授、
南开大学程明明教授

用神经网络拟合(行动)价值函数：Deep Q-learning

使用 ϵ 贪心策略的Q学习

- 状态数量太多时，有些状态可能始终无法采样到，因此对这些状态的 q 函数进行估计是很困难的
- 状态数量无限时，不可能用一张表(数组)来记录 q 函数的值

初始化 q_π 函数

循环

初始化 s 为初始状态

循环

采样 $a \sim \epsilon\text{-greedy}_\pi(s)$

执行动作 a ，观察奖励 R 和下一个状态 s'

更新 $q_\pi(s, a) \leftarrow q_\pi(s, a) + \alpha \left[R + \gamma \max_{a'} q_\pi(s', a') - q_\pi(s, a) \right]$

$s \leftarrow s'$

直到 s 是终止状态

直到 q_π 收敛

思路：将 q 函数参数化(parametrize)，用一个非线性回归模型来拟合 q 函数，例如(深度)神经网络

- 能够用有限的参数刻画无限的状态
- 由于回归函数的连续性，没有探索过的状态也可通过周围的状态来估计

用神经网络拟合(行动)价值函数： Deep Q-learning

• 用深度神经网络拟合 Q 函数

初始化 q_π 函数的参数 θ

循环

初始化 s 为初始状态

循环

采样 $a \sim \epsilon\text{-greedy}_\pi(s; \theta)$

执行动作 a ，观察奖励 R 和下一个状态 s'

损失函数 $L(\theta) = \frac{1}{2} \left[R + \gamma \max_{a'} q_\pi(s', a'; \theta) - q_\pi(s, a; \theta) \right]^2$

根据梯度 $\partial L(\theta) / \partial \theta$ 更新参数 θ

$s \leftarrow s'$

直到 s 是终止状态

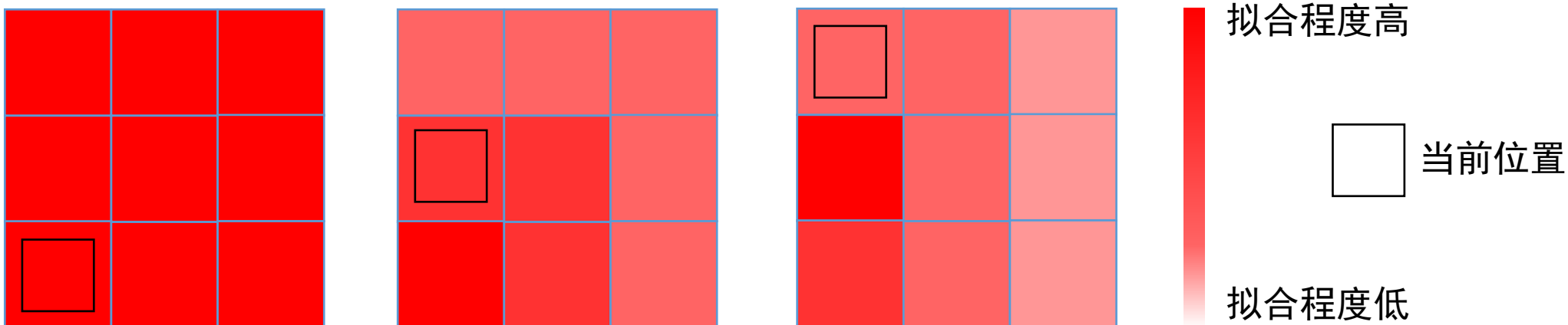
直到 q_π 收敛

- 损失函数刻画了 q 的估计值 $R + \gamma \max_{a'} q_\pi(s', a'; \theta)$ 与当前值的平方误差
- 利用梯度下降法优化参数 θ
- 如果用深度神经网络来拟合 q 函数，则算法称为深度Q学习或者深度强化学习

深度Q学习的两个不稳定因素

1. 相邻的样本来自同一条轨迹，样本之间相关性太强，集中优化相关性强的样本可能导致神经网络在其他样本上效果下降。

集中优化一条轨迹上的状态时，远离该轨迹的状态的估计值可能会发生较大偏离



2. 在损失函数中， q 函数的值既用来估计目标值，又用来计算当前值。现在这两处的 q 函数通过 θ 有所关联，可能导致优化时不稳定

$$\frac{1}{2} \left[R + \gamma \underbrace{\max_{a'} q_{\pi}(s', a'; \theta)}_{\text{预测值}} - \underbrace{q_{\pi}(s, a; \theta)}_{\text{当前值}} \right]^2$$

预测值

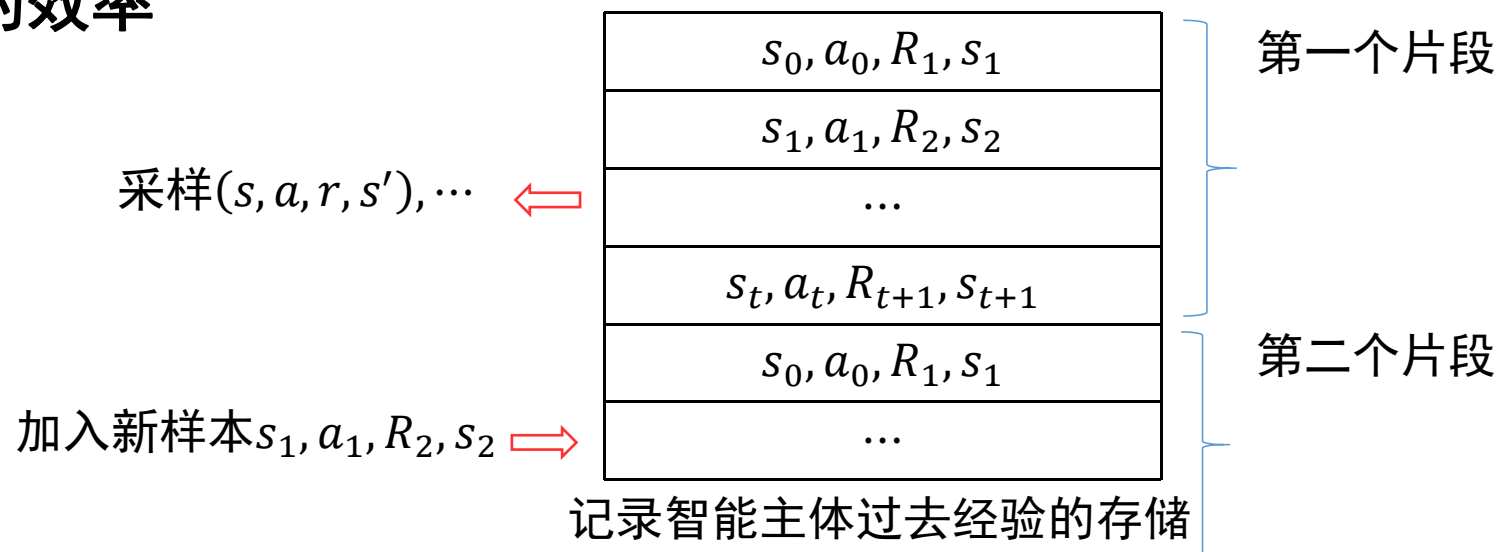
当前值

经验重现(Experience Replay)

- 相邻的样本来自同一条轨迹，样本之间相关性太强，集中优化相关性强的样本可能导致神经网络在其他样本上效果下降。

将过去的经验存储下来，每次将新的样本加入到存储中去，并从存储中采样一批样本进行优化

- 解决了样本相关性强的问题
- 重用经验，提高了信息利用的效率



目标网络(Target Network)

- 在损失函数中， q 函数的值既用来估计目标值，又用来计算当前值。现在这两处的 q 函数通过 θ 有所关联，可能导致优化时不稳定

$$\frac{1}{2} \left[R + \gamma \max_{a'} \overset{\text{目标网络}}{q_{\pi}(s', a'; \theta^-)} - q_{\pi}(s, a; \theta) \right]^2$$

损失函数的两个 q 函数使用不同的参数计算

- 用于计算估计值的 q 使用参数 θ^- 计算，这个网络叫做目标网络
- 用于计算当前值的 q 使用参数 θ 计算
- 保持 θ^- 的值相对稳定，例如 θ 每更新多次后才同步两者的值

$$\theta^- \leftarrow \theta$$

提纲

- 一、强化学习问题定义
- 二、基于价值的强化学习
- 三、基于策略的强化学习
- 四、深度强化学习的应用

基于策略的强化学习

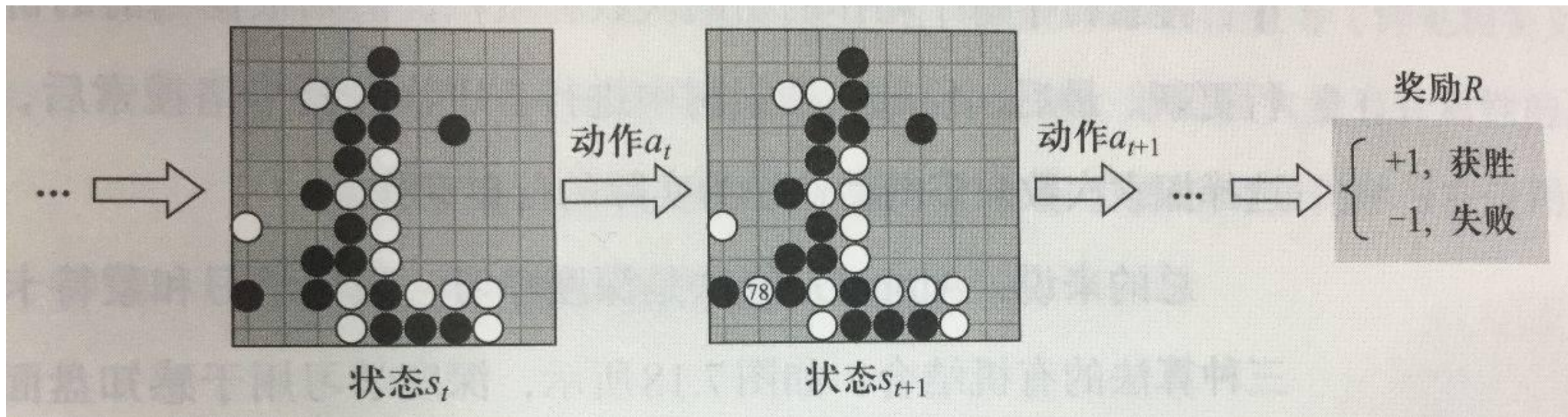
- 基于价值的强化学习:以对价值函数或动作-价值函数的建模为核心。
- 基于策略的强化学习:直接参数化策略函数，求解参数化的策略函数的梯度。
- 策略函数的参数化可以表示为 $\pi(a|s; \theta)$ (θ, a), 其中 θ 为一组参数，函数取值表示在状态 s 下选择动作 a 的概率。和Q学习的 Q 贪心策略相比，这种参数化的一个显著好处是：选择一个动作的概率是随着参数的改变而光滑变化的，实际上这种光滑性对算法收敛有更好的保证。

提纲

- 一、强化学习问题定义
- 二、基于价值的强化学习
- 三、基于策略的强化学习
- 四、深度强化学习的应用

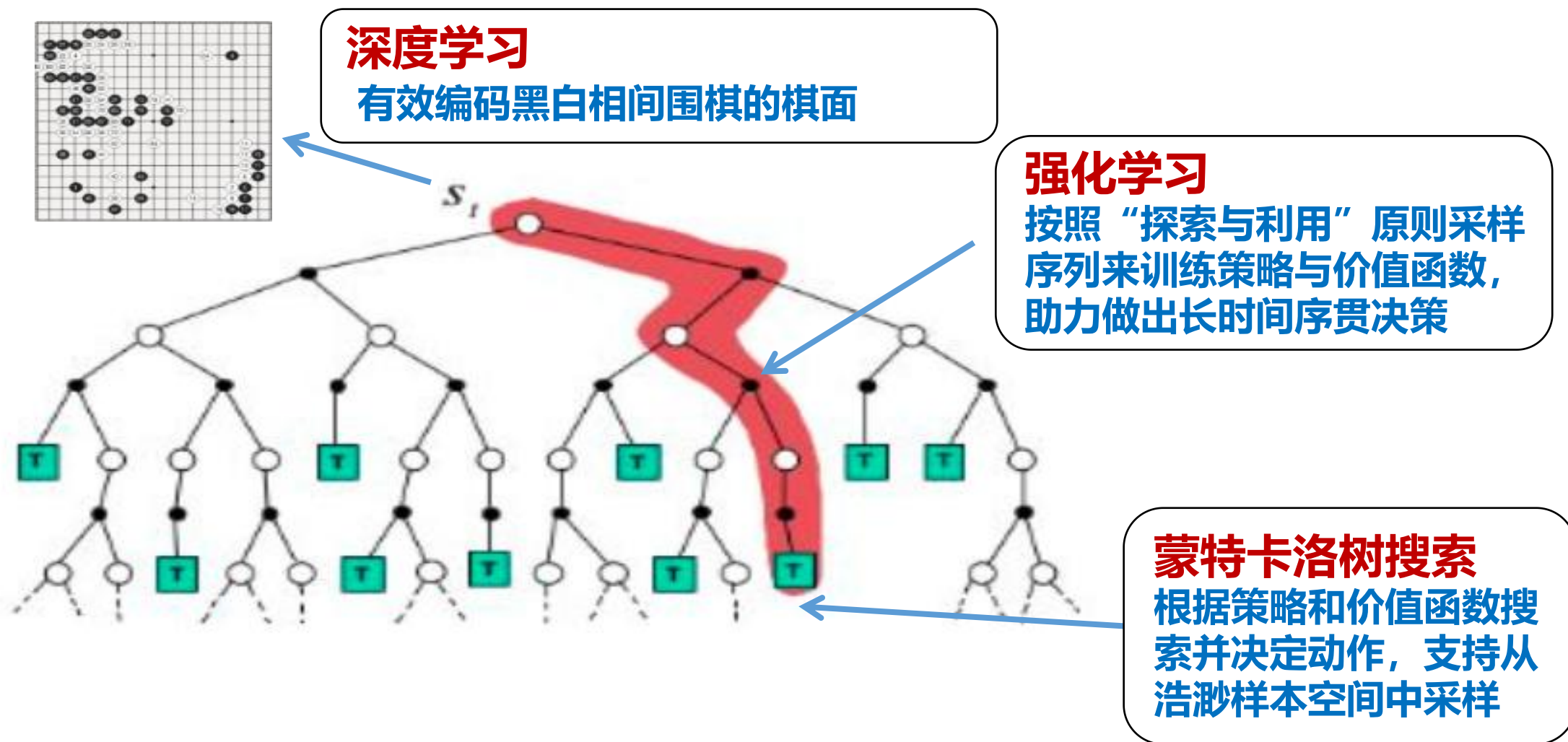
深度Q学习的应用实例：围棋博弈

- 围棋游戏一个片段的轨迹



深度Q学习的应用实例: 围棋博弈

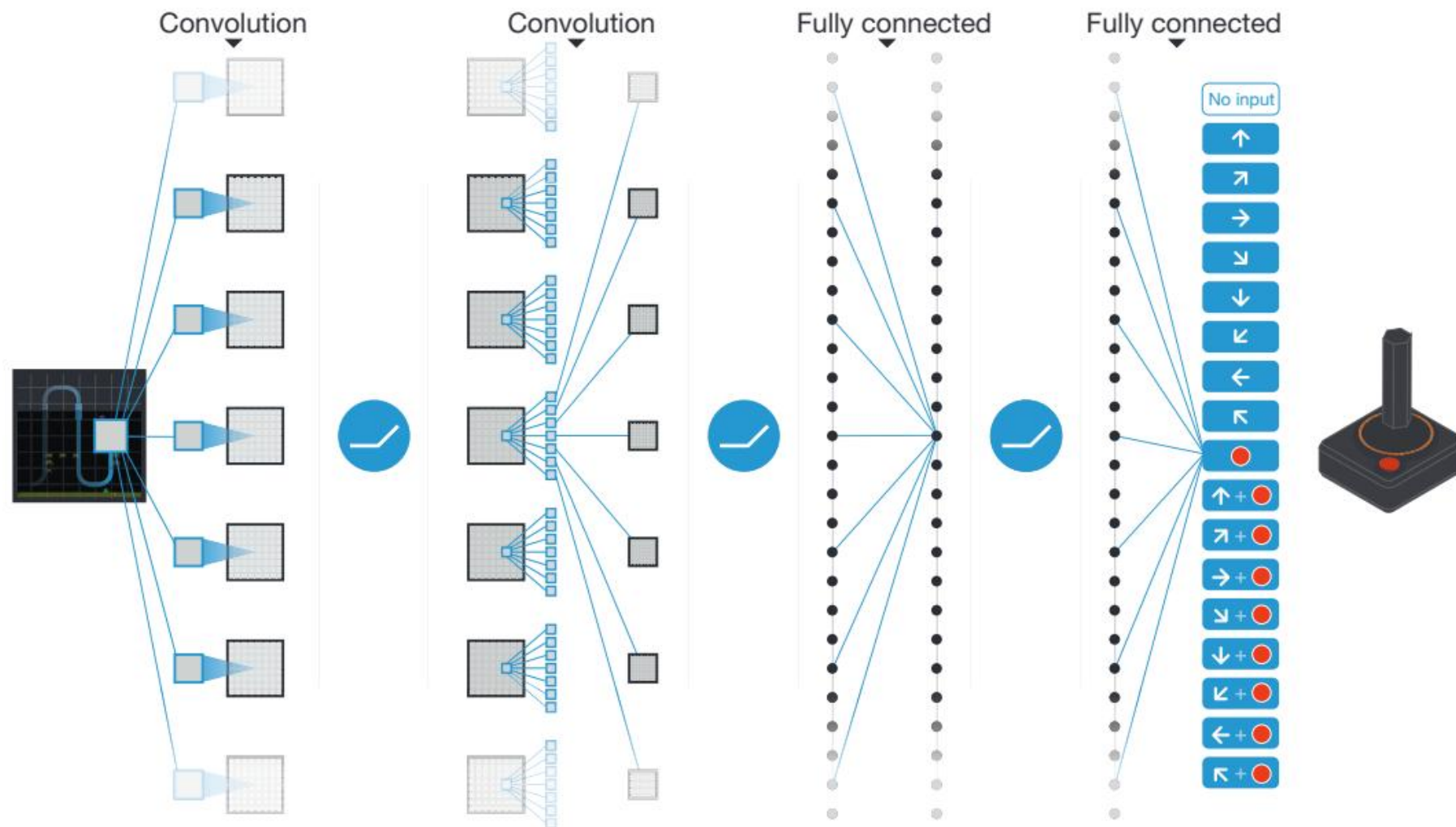
- AlphaGo算法的三个重要组成部分



深度Q学习的应用实例: 雅达利游戏

• 用于游戏的DQN动作-价值函数模型

q 函数的学习模型

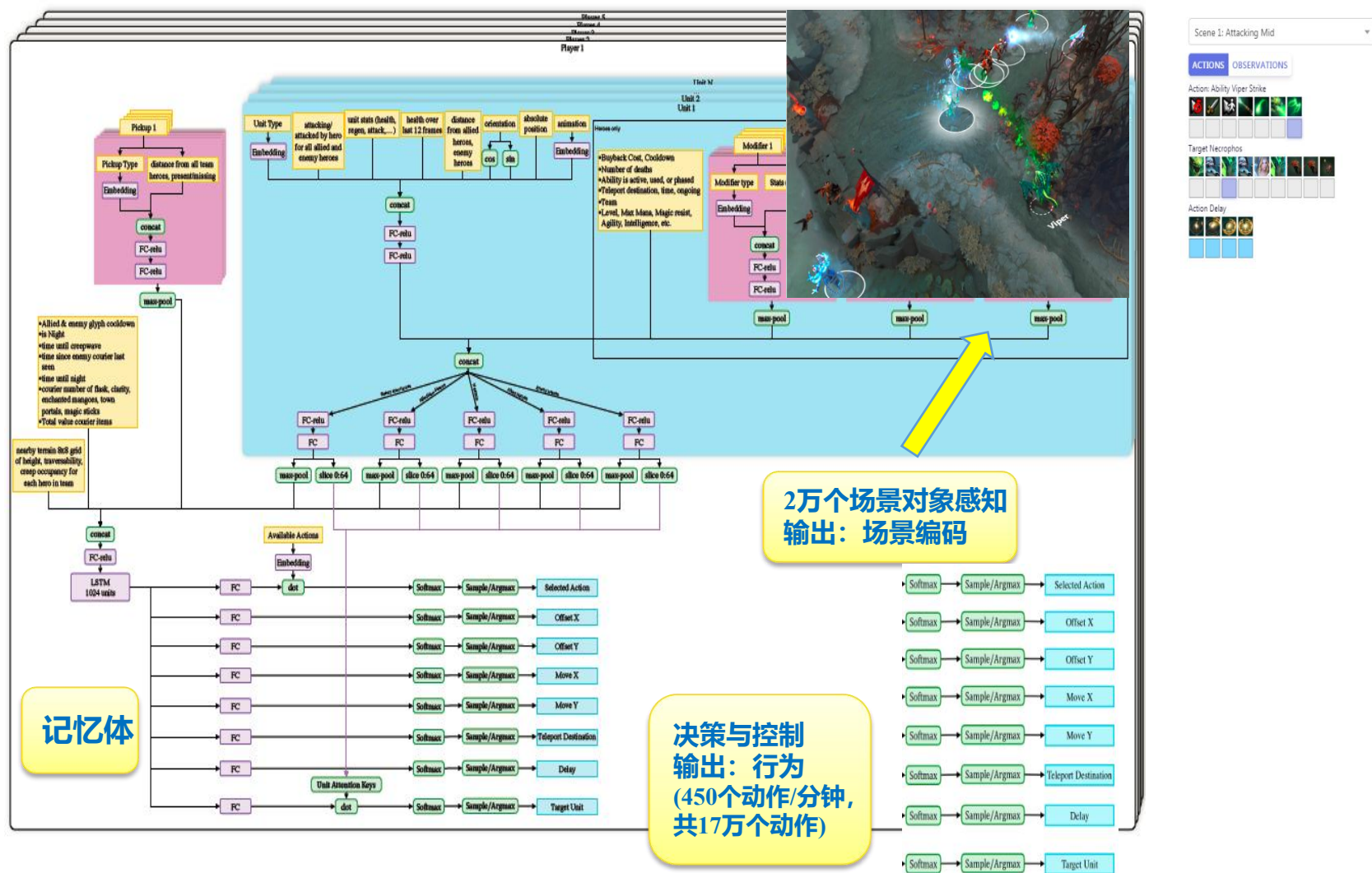


Mnih, Volodymyr, et al, Human-level control through deep reinforcement learning, Nature 518.7540 (2015)

博弈对抗的算法例子

<https://www.bilibili.com/video/BV1Eb411T77Z/>

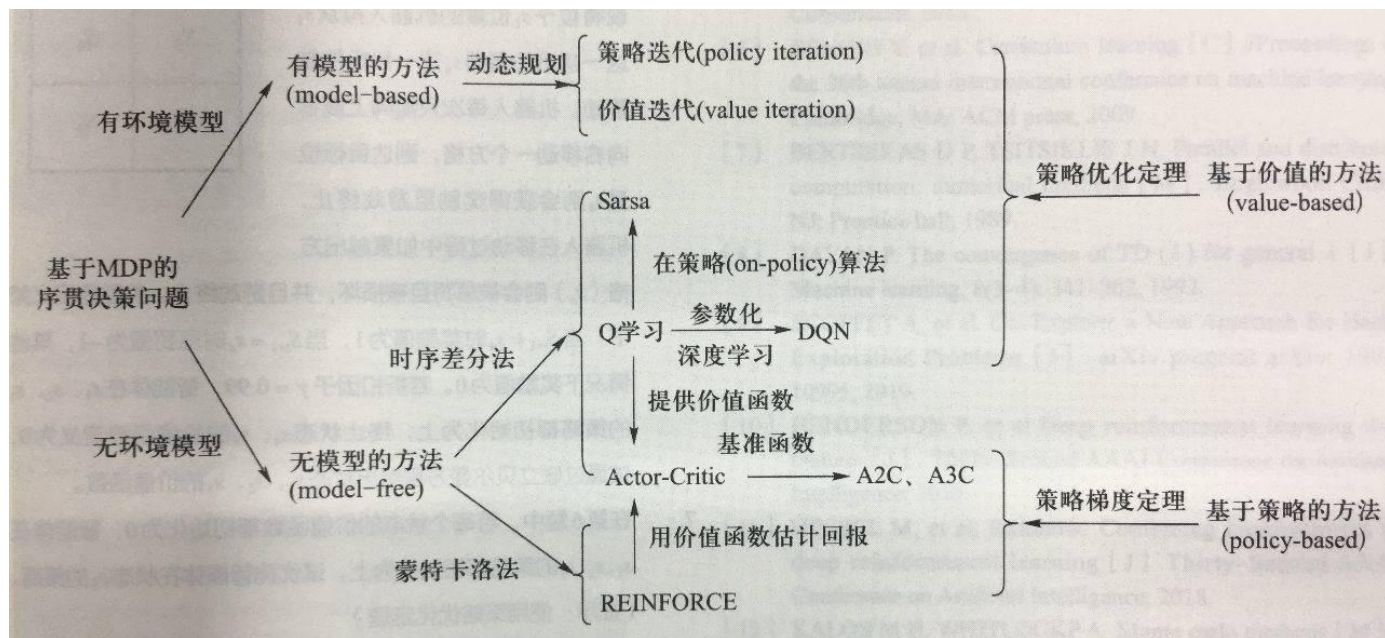
• DOTA2 (状态空间极其庞大下完全信息下博弈)



强化学习的分类

欲粟者务时
欲治者因势

所介绍方法之间相互关系示意图



图中从两个角度对强化学习算法做了分类，其中依靠对环境(即马尔可夫随机过程)的先验知识或建模的算法称为基于模型(Model-based)的方法，反之称为无模型的方法(Model-free)；只对价值函数建模并利用策略优化定理求解的方法称为基于价值(Value-based)的方法，对策略函数建模并利用策略梯度定理求解的方法称为基于策略(Policy-based)的方法。

下面对强化学习、监督学习和深度卷积神经网络学习的描述正确的是（ ）

- ☒ A 评估学习方式、有标注信息学习方式、端到端学习方式
- ☐ B 有标注信息学习方式、端到端学习方式、端到端学习方式
- ☐ C 评估学习方式、端到端学习方式、端到端学习方式
- ☐ D 无标注学习、有标注信息学习方式、端到端学习方式

提交

在强化学习中，通过哪两个步骤的迭代，来学习得到最佳策略（ ）

- ☐ A 价值函数计算与动作-价值函数计算
- ☐ B 动态规划与Q-Learning
- ☐ C 贪心策略优化与Q-learning
- ☒ D 策略优化与策略评估

提交

在强化学习中，哪个机制的引入使得强化学习具备了在利用与探索中寻求平衡的能力（ ）

- ☒ A ϵ 贪心策略
- ☐ B 蒙特卡洛采样
- ☐ C 动态规划
- ☐ D 贝尔曼方程

提交

与马尔可夫奖励过程相比，马尔可夫决策过程引入了哪一个新的元素（ ）？

- ☐ A 反馈
- ☒ B 动作
- ☐ C 终止状态
- ☐ D 概率转移矩阵

提交

在本章内容范围内，“在状态 s ，按照某个策略行动后在未来所获得回报值的期望”，这句话描述了状态 s 的（ ）

- ☐ A 策略优化
- ☒ B 价值函数
- ☐ C 动作-价值函数
- ☐ D 采样函数

提交

在本章内容范围内，“在状态 s ，按照某个策略采取动作 a 后在未来所获得回报值的期望”，这句话描述了状态 s 的（ ）

- ☐ A 策略优化
- ☐ B 价值函数
- ☒ C 动作-价值函数
- ☐ D 采样函数

提交

在题 6 中，若图 2 表示算法的初始状态，其中 a/b 表示对应状态的动作-价值函数的取值，斜线左侧的 a 表示 $q_{\pi}(s, \text{上})$ ，斜线右侧的 b 表示 $q_{\pi}(s, \text{右})$ 。若 $\alpha = 0.5$ ，试给出算法 7.6 中的 Q 学习算法的一个片段的执行过程，并给出执行完该片段后每个状态的策略。

0/0	0.1/0	0/0
	0.1/0	0.1/0

图 2 Q 学习算法的初始状态

作答

解 根据算法 7.2.6 中的 Q 学习算法, s_1 为初始状态, 根据当前策略求出智能体应该采取的动作 $a = \operatorname{argmax}_a q_\pi(s_1, a) = \text{上}$, 执行这个动作, 得到奖励 $R = 0$ 和进入下一状态 $s' = s_3$, 因此可如下更新对应的动作-价值函数:

$$q_\pi(s_1, \text{上}) \leftarrow q_\pi(s_1, \text{上}) + \alpha [R + \gamma \max_a q_\pi(s', a) - q_\pi(s, a)]$$

$$= 0.1 + 0.5 \times [0 + 0.99 \times \max\{0, 0.1\} - 0.1] = 0.0995$$

此时的 q 函数为:

0.1/0	0/0
0.0995/0	0.1/0

同时令当前状态为 s_3 , 此时智能体应该采取的动作 $a = \operatorname{argmax}_a q_\pi(s_3, a) = \text{上}$, 执行这个动作, 得到奖励 $R = -1$ 和进入下一状态 $s' = s_d$, 因此可如下更新对应的动作-价值函数:

$$q_\pi(s_3, \text{上}) \leftarrow q_\pi(s_3, \text{上}) + \alpha [R + \gamma \max_a q_\pi(s', a) - q_\pi(s, a)]$$

$$= 0.1 + 0.5 \times [-1 + 0.99 \times \max\{0, 0\} - 0.1] = -0.45$$

此时算法达到终止状态 s_d , 该片段结束。此时的 q 函数为:

-0.45/0	0/0
0.0995/0	0.1/0

此时每个状态的策略为:

→	
↑	↑

作答

提纲

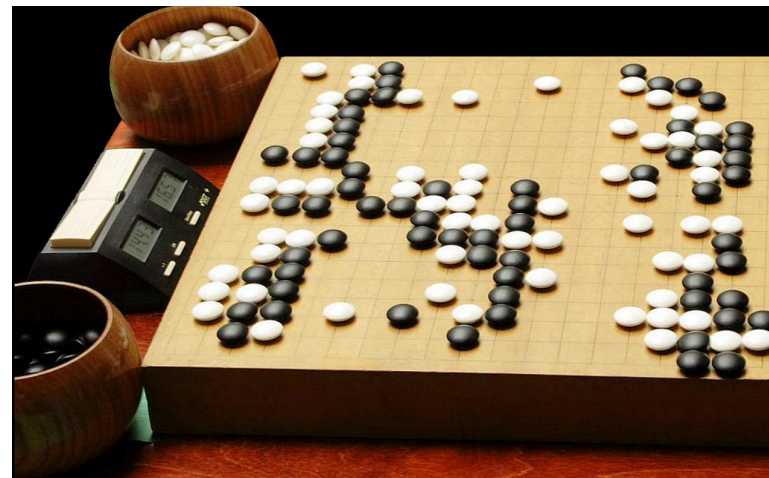
- 博弈相关概念
- 遗憾最小化算法
- 虚拟遗憾最小化算法
- 人工智能安全

博弈论的诞生：中国古代博弈思想

- 子曰：饱食终日，无所用心，难矣哉！不有**博弈**者乎？为之，犹贤乎已。

——《论语·阳货》

- 朱熹集注曰：“博，局戏；弈，围棋也”
- 颜师古注：“博，六博；弈，围碁也”
- 古语博弈所指下围棋，围棋之道蕴含古人谋划策略的智慧。
- 略观围棋，法于用兵，怯者无功，贪者先亡。——《围棋赋》
- 《孙子兵法》等古代典籍更是凸显了古人对策略的重视。



博弈论的诞生：田忌赛马

- 田忌信然之，与王及诸公子逐射千金。及临质，孙子曰：
 - “今以君之下驷与彼上驷，取君上驷与彼中驷，取君中驷与彼下驷。”既驰三辈毕，而田忌一不胜而再胜，卒得王千金。

——《史记·孙子吴起列传》

对局	齐王马	田忌马	结果
1	A+	A-	齐王胜
2	B+	B-	齐王胜
3	C+	C-	齐王胜

3:0

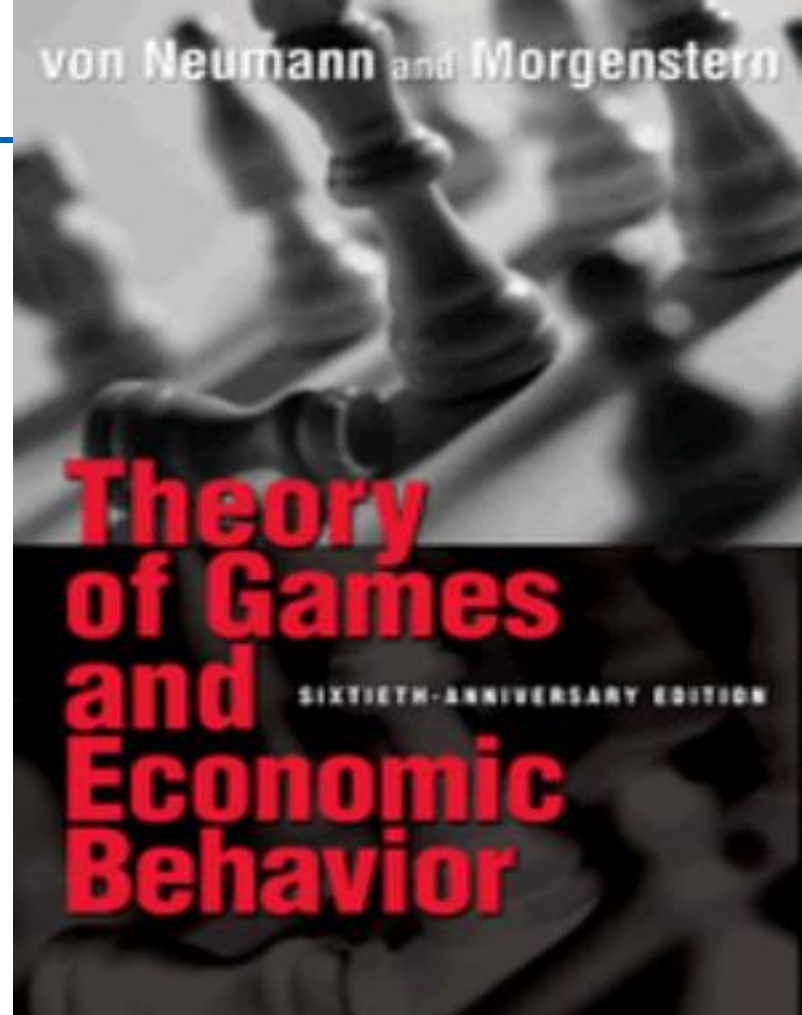
对局	齐王马	田忌马	结果
1	A+	C-	齐王胜
2	B+	A-	田忌胜
3	C+	B-	田忌胜

1:2

以己之长 攻彼之短

博弈论的诞生：现代博弈论的建立

- 博弈论 (game theory), 又称对策论。
 - 博弈行为：带有相互竞争性质的主体，为了达到各自目标和利益，采取的带有对抗性质的行为。
 - 博弈论主要研究博弈行为中最优的对抗策略及其稳定局势，协助人们在一定规则范围内寻求最合理的行为方式。
 - 《博弈论与经济行为》以数学形式来阐述博弈论及其应用。冯·诺伊曼被称为现代博弈论之父。



John von Neumann, Oskar Morgenstern, *Theory of Games and Economic Behavior*, 1944, Princeton University Press

博弈论的相关概念：博弈的要素

- **玩家 (player)**：参与博弈的决策主体
- **策略 (strategy)**：玩家可以采取的行动方案，是一整套在采取行动之前就已经准备好的完整方案。
 - 某个玩家可采纳策略的全体组合形成了**策略集** (strategy set)
 - 所有玩家各自采取行动后形成的状态被称为**局势** (outcome)
 - **混合策略** (mixed strategy): 玩家可通过一定概率来选择若干个不同的策略
 - **纯策略** (pure strategy): 玩家每次行动都选择某个确定的策略
- **收益 (payoff)**：各个玩家在不同局势下得到的利益
 - 混合策略意义下的收益应为期望收益 (expected payoff)
- **规则 (rule)**：对玩家行动的先后顺序、玩家获得信息多少等内容的规定

博弈论的相关概念：研究范式

- 建模者对**玩家**规定可采取的**策略集**和取得的收益，观察当玩家选择若干策略以最大化其收益时会产生什么结果

两害相权取其轻，两利相权取其重

博弈论的相关概念：囚徒困境 (prisoner's dilemma)

- 1950年，兰德公司的弗勒德和德雷希尔拟定了相关困境理论
 - 后来普林斯顿大学数学家阿尔伯特·塔克以“囚徒方式”阐述
 - 警方逮捕了共同犯罪的甲和乙，但没有掌握充分证据
- 分开审讯：
 - 若一人认罪并指证对方，而另一方保持沉默，则此人会被当即释放，沉默者会被判监禁10年
 - 若两人都沉默，则根据已有的犯罪证据两人各判半年
 - 若两人都认罪并相互指证，则两人各判5年

在囚徒困境中，甲乙两人最有可能被判的刑期是？

- ☐ A 甲0.5年，乙0.5年
- ☐ B 甲10年，乙0年
- ☐ C 甲0年，乙10年
- ☒ D 甲5年，乙5年

提交

博弈论的相关概念：囚徒困境 (prisoner's dilemma)

- 玩家：甲、乙
- 规则：甲、乙两人分别决策，无法得知对方的选择
- 策略集：认罪、沉默 (纯策略)

(甲,乙) 收益	乙沉默 (合作)	乙认罪 (背叛)
甲沉默 (合作)	(-0.5, -0.5)	(-10, 0)
甲认罪 (背叛)	(0, -10)	(-5, -5)

在囚徒困境中，最优解为两人同时沉默
但是两人实际倾向于选择同时认罪 (均衡解)

博弈论的相关概念：博弈的分类

- **合作(cooperative)博弈与非合作(non-cooperative)博弈**
 - **合作**：部分玩家可以组成联盟以获得更大的收益
 - **非合作**：玩家在决策中都彼此独立，不事先达成合作意向
- **静态(static)博弈与动态(dynamic)博弈**
 - **静态**：所有玩家同时决策，或玩家互相不知道对方的决策
 - **动态**：玩家所采取行为的先后顺序由规则决定，且后行动者知道先行动者所采取的行为
- **完全信息(complete information)博弈与不完全(incomplete)信息博弈**
 - **完全信息**：所有玩家均了解其他玩家的策略集、收益等信息
 - **不完全信息**：并非所有玩家均掌握了所有信息

囚徒困境是一种 [填空1]、 [填空2] 的 [填空3] 博弈

作答

正常使用填空题需3.0以上版本雨课堂

博弈论的相关概念：纳什均衡

- 博弈的稳定局势即为纳什均衡 (Nash equilibrium)
 - 玩家所作出的这样一种策略组合: 任何玩家单独改变策略都不会得到好处。
 - 即: 当所有其他人都 不改变策略时, 没有人会改变自己的策略。
- Nash定理: 若玩家有限, 每位玩家的策略集有限, 收益函数为实值函数, 则博弈必存在混合策略意义下的纳什均衡。
 - 囚徒困境中两人同时认罪就是这一问题的纳什均衡。

纳什均衡的本质：不后悔

Nash, J, Non-Cooperative Games. *The Annals of Mathematics*. 54, 2 (1951), 286.

博弈论的相关概念：混合策略下纳什均衡的例子

- 公司的雇主是否检查工作与雇员是否偷懒
 - 玩家：雇员、雇主
 - 规则：雇员与雇主两人分别决策，事先无法得知对方的选择
 - 混合策略集：
 - 雇员：偷懒、不偷懒
 - 雇主：检查、不检查
 - 局势及对应收益：雇主采取检查/不检查策略时雇员工作与偷懒对应的结果

博弈论的相关概念：混合策略下纳什均衡的例子

- 公司的雇主是否检查工作与雇员是否偷懒
 - 雇主的检查成本 C ，发现偷懒的惩罚 F
 - 雇员的贡献 V ，工资 W ，付出 H
 - 假定 $C < F < W$ ， $V > W$

		雇员	
		偷懒	不偷懒
雇主	检查	$-C + F, -F$	$V - W - C, W - H$
	不检查	$-W, W$	$V - W, W - H$

博弈论的相关概念：混合策略下纳什均衡的例子

• 若雇主检查的概率为 α ，雇员偷懒的概率为 β

• 雇主的检查成本 C ，发现偷懒的惩罚 F

• 雇员的贡献 V ，工资 W ，付出 H

• 假定 $C < V < W$ ， $F > H$

		雇员	
		偷懒 β	不偷懒
雇主	检查 α	$-C + F, -F$	$V - W - C, W - H$
	不检查	$-W, W$	$V - W, W - H$

	策略	收益
雇主	检查	$T_1 = \beta(-C + F) + (1 - \beta)(V - W - C)$
	不检查	$T_2 = -\beta W + (1 - \beta)(V - W)$
雇员	偷懒	$T_3 = -\alpha F + (1 - \alpha)W$
	不偷懒	$T_4 = (W - H) + (1 - \alpha)(W - H) = (W - H)$

博弈论的相关概念：混合策略下纳什均衡的例子

- 若雇主检查的概率为 α ，雇员偷懒的概率为 β

	策略	收益
雇主	检查	$T_1 = \beta(-C + F) + (1 - \beta)(V - W - C)$
	不检查	$T_2 = -\beta W + (1 - \beta)(V - W)$
雇员	偷懒	$T_3 = -\alpha F + (1 - \alpha)W$
	不偷懒	$T_4 = (W - H) + (1 - \alpha)(W - H) = (W - H)$

- 纳什均衡：某个玩家单独采取其他策略都不会使得收益增加
 - 无论雇主是否检查，自己收益都不增加： $T_1 = T_2$
 - 无论雇员是否偷懒，自己收益都不增加： $T_3 = T_4$

博弈论的相关概念：混合策略下纳什均衡的例子

- 纳什均衡：某个玩家单独采取其他策略都不会使得收益增加
 - 由于 $T_3 = T_4$ ，可知雇主采取检查策略的概率 $\alpha = \frac{H}{W+F}$
 - 由于 $T_1 = T_2$ ，可知雇员采取偷懒策略的概率 $\beta = \frac{C}{W+F}$
- 在检查概率为 α 之下，雇主的收益：

$$T_1 = T_2 = V - W - \frac{CV}{W + F}$$

- 对上式中 W 求导，则当 $W = \sqrt{CV} - F$ 时，雇主的收益最大，为

$$T_{max} = V - 2\sqrt{CV} + F$$

博弈论与计算机科学

- 博弈论与计算机科学的交叉领域非常多

- 理论计算机科学：算法博弈论
- 人工智能：多智能体系统、AI玩家、人机交互、广告推荐
- 互联网：互联网经济、共享经济
- 分布式系统：区块链

- 人工智能与博弈论相结合的两个主要方向

- 博弈策略的求解
- 博弈规则的设计



冯·诺依曼：现代计算机之父+博弈论之父

博弈策略求解

• 动机

- 博弈论提供了许多问题的数学模型
- 纳什定理确定了博弈过程问题存在解
- 人工智能模型可用来求解均衡局面或者最优策略

• 主要问题

- 如何高效求解博弈玩家的策略以及博弈的均衡局势？

博弈策略求解

• 应用领域

- 大规模搜索空间的问题求解： 围棋
- 非完全信息博弈问题求解： 德州扑克
- 网络对战游戏智能： Dota、 星球大战
- 动态博弈的均衡解： 厂家竞争、 信息安全

遗憾最小化算法 (Regret Minimization) : 若干定义

- 玩家 i 所采用的策略为 σ_i ，一个策略组 σ 包含所有玩家策略
$$\sigma = (\sigma_1, \sigma_2, \dots, \sigma_{|N|})$$
 - 玩家 i 的策略空间用 Σ_i 表示。
 - σ_{-i} 表示 σ 中除了 σ_i 之外的策略。
- 玩家 i 在给定策略 σ 下的期望收益为： $u_i(\sigma)$

遗憾最小化算法：最佳反应策略与纳什均衡

- 玩家 i 对于所有其他玩家的策略组 σ_{-i} 的**最佳反应策略** σ_i^* 满足：

$$u_i(\sigma_i^*, \sigma_{-i}) \geq \max_{\sigma'_i \in \Sigma_i} u_i(\sigma'_i, \sigma_{-i})$$

- 策略组 $\sigma = (\sigma_1^*, \sigma_2^*, \dots, \sigma_{|N|}^*)$ 是**纳什均衡**当且仅当对每个玩家 i ：

$$u_i(\sigma) \geq \max_{\sigma'_i \in \Sigma_i} u_i(\sigma_1^*, \sigma_2^*, \dots, \sigma'_i, \dots, \sigma_{|N|}^*)$$

- 若所有玩家都是理性的，最优反应策略就是一个纳什均衡
 - 考虑计算资源限制，难以通过遍历寻找最优反应策略
 - 需要找到一种能快速发现近似纳什均衡的方法

遗憾最小化算法：策略选择

- 根据过去博弈中的遗憾程度来决定将来动作选择的方法
- 玩家 i 在过去 T 轮中采取策略 σ_i 的累加遗憾值为：

$$Regret_i^T(\sigma_i) = \sum_{t=1}^T (\mu_i(\sigma_i, \sigma_{-i}^t) - \mu_i(\sigma^t))$$

- 在第 $T + 1$ 轮次玩家 i 选择策略 a 的概率如下 (悔值越大越选择)

$$P(a) = \frac{Regret_i^T(a)}{\sum_{b \in \Sigma_i} Regret_i^T(b)}$$

- 既能启发式提升未来收益，又能防止对手发现自己的策略

遗憾最小化算法：石头-剪刀-布的例子

- 假设玩家A和B进行石头-剪刀-布 (Rock-Paper-Scissors) 的游戏
 - 玩家收益：获胜1分，失败-1分，平局0分
- 第一局时，若玩家A出石头 (R)，玩家B出布 (P)
 - 此时玩家A的收益 $\mu_A(R, P) = -1$ ，玩家B的收益为 $\mu_B(P, R) = 1$
 - 如果玩家A选择出布或剪刀，则收益值为 $\mu_A(P, P) = 0$ 或 $\mu_A(S, P) = 1$
- 玩家A第一局没有出布的遗憾值为 $\mu_A(P, P) - \mu_A(R, P) = 1$
 - 没有出剪刀的遗憾值为 $\mu_A(S, P) - \mu_A(R, P) = 2$
- 在第二局中，A选择R, P, S这三个策略的概率分别为 $0, 1/3, 2/3$

遗憾最小化算法：石头-剪刀-布的例子

- 玩家A每一轮遗憾值及第二轮后的累加遗憾取值：

- 前提：在第一轮中玩家A选择石头和B选择布、在第二局中玩家A选择剪刀和玩家B选择石头情况下

每轮悔值\策略	石头	剪刀	布
第一轮悔值	0	2	1
第二轮悔值	1	0	2
$Regret_A^2$	1	2	3

- 在第三局时，玩家A选择R, S, P的概率分别为1/6、2/6、3/6
- 在实际使用中，可以通过多次模拟迭代累加遗憾值找到每个玩家在每一轮次的最优策略。但是当博弈状态空间呈指数增长时，对一个规模巨大的博弈树无法采用最小遗憾算法==>虚拟最小遗憾算法

博弈规则的设计

• 问题描述

- 假设博弈的玩家都是足够理性的
- 如何设计一个博弈规则能确保公正性或者达到设计者的最大利益

• 挑战

- 规则复杂
- 计算量大

• 应用领域

- 拍卖竞价：互联网广告投放、车牌竞价
- 供需匹配：污染权、学校录取
- 公正选举：选举制度、表决制度、议席分配

双边匹配算法

- 需要双向选择的情况被称为是**双边匹配问题**
 - 与资源匹配相关的决策问题 (如求职就业、 报考录取等)
 - 需要双方互相满足对方的需求才会达成匹配
- **稳定匹配**是指没有任何人能从偏离稳状态中获益
 - 如果将匹配问题看做是一种合作博弈，稳定状态对应纳什均衡
- 针对双边稳定匹配问题的算法并应用于稳定婚姻问题的求解
 - 1962年，美国数学家大卫·盖尔和博弈论学家沙普利提出

单边匹配算法

- **一类交换不可分标的物的匹配问题**

- 如远古时期以物易物、或者宿舍的床位分配
- 不可分的标的物只能属于一个所有者，且可属于任何所有者
- 沙普利和斯卡夫提出了针对单边匹配问题的稳定匹配算法

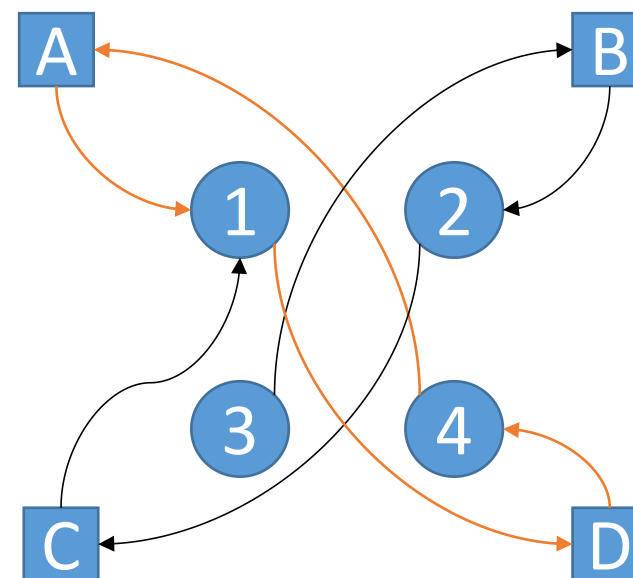
- **最大交易圈算法 (TTC) :**

- 每个交易者连接一条指向他最喜欢的标的物的边，并从每一个标的物连接到其占有者或是具有高优先权的交易者，形成一张有向图
- 如果存在交易圈，其中的交易者，将每人指向的标的物赋予其，同时交易者放弃原先的标的物，相关占有者和标的物离开市场
- 从剩余的交易者和标的物之间重复交易圈匹配，直到没有交易圈

最大交易圈算法：室友匹配问题

- 假设某寝室有A、B、C、D四位同学和1、2、3、4四个床位
 - 当前给A、B、C、D四位同学随机分配4、3、2、1四个床位
 - 已知四位同学对床位偏好如下：

同学	偏好
A	1>2>3>4
B	2>1>4>3
C	1>2>4>3
D	4>3>1>2



- 可以看出交易图中A和D之间构成一个交易圈

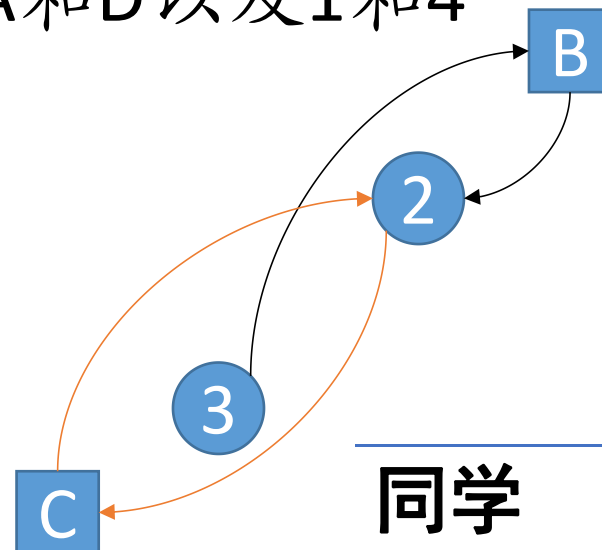
最大交易圈算法：室友匹配问题

- 假设某寝室有A、B、C、D四位同学和1、2、3、4四个床位

- 从匹配图中移除交易圈中的参与者：A和D以及1和4

- 第二轮**

- 依照算法步骤可得匹配图
- B和C都希望得到床位2
- 无法再构成交易圈
- C是床位的本身拥有者，所以仍然得到床位2
- B只能选择床位3。
- 最后交易结果A→1，B→3，C→2，D→4。



同学	偏好
A	1>2>3>4
B	2>1>4>3
C	1>2>4>3
D	4>3>1>2

基于人工智能的信息安全技术：加密协议

• 加密技术

- 将明文信息处理为难以读取的密文内容，使之不可读。
- 在网络环境中保障通信安全，保证数据的完整性
- 目前常用的加密算法有安全哈希算法 (Secure Hash Algorithm, SHA) 和高级加密标准 (Advanced Encryption Standard, AES)

• 使用神经网络的加密算法

- 2016年谷歌大脑的研究团队提出了使用对抗生成网络生成的一个加密算法，其使用了三个神经网络分别完成加密、解密和攻击的工作，以保证信息的无损传输以及第三方无法破译

learning to protect communications with adversarial neural cryptography

基于人工智能的信息安全技术：数字水印

Hiding Images in Plain Sight:
Deep Steganography

• 数字水印

- 将特定信息 嵌入在数字信号中，拷贝时水印内容会被同时拷贝
- 水印可作为版权信息的证明，避免未经授权的复制和拷贝
- 通过神经网络来添加水印和提取水印信息的成为学术研究热点。

Original
cover



secret



Reconstructed
cover



secret



人工智能的安全：数据安全与模型安全

- 人工智能很大程度是依靠数据驱动学习
- 可用性 (availability)
 - 训练数据是否充足且可靠
 - 训练数据是否有足够的标注
- 完整性 (completeness)
 - 数据是否具有代表性
- 隐私性 (privacy)
 - 数据是否涉及隐私安全问题
 - 如何保障数据不被窃取

人工智能的安全：数据安全与模型安全

- 人工智能所使用的的模型是由有限的训练数据训练得到的
- 鲁棒性 (robustness)
 - 模型是否易于受到噪声干扰或攻击
- 正确性 (correctness)
 - 模型是否正确
- 通用性 (generality)
 - 模型是否能够应用于现实场景
 - 模型对输入数据是否有过高的要求

人工智能的安全：对模型的攻击

• 对模型的攻击

- 使用特定技术对输入样本进行微小的修改就可骗过模型
- 这种经过修改，使得模型判断错误的样本被称为对抗样本

• 白盒攻击

- 攻击者熟知人工智能模型的算法和模型参数
- 生成对抗样本的过程可以与模型的每一部分进行交互

• 黑盒攻击

- 攻击者只能给定输入去获得模型输出，但并不知道被攻击模型所使用的算法和参数。可以针对任何一个人工智能模型

对抗样本的生成：白盒攻击

- 白盒攻击通常会对模型的每一部分进行逐层分解
 - 对每部分添加一定扰动，使模型结果逐步向误判目标偏移
 - 非常隐蔽，通过限制扰动的大小可使看起来与原样本差别很小



x

“panda”

57.7% confidence

+ .007 ×



$\text{sign}(\nabla_x J(\theta, x, y))$

“nematode”

8.2% confidence

=



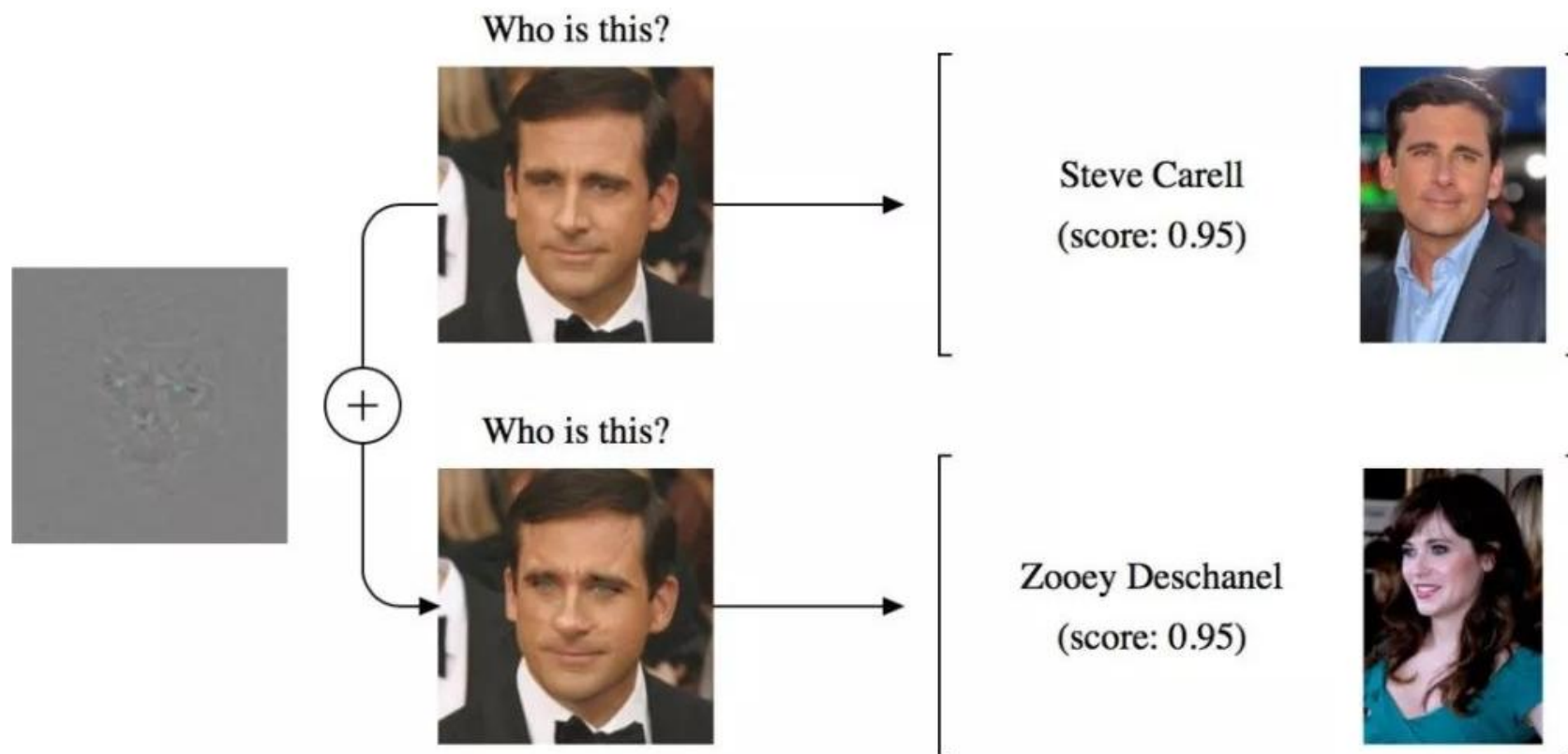
$x + \epsilon \text{sign}(\nabla_x J(\theta, x, y))$

“gibbon”

99.3 % confidence

对抗样本的生成：白盒攻击

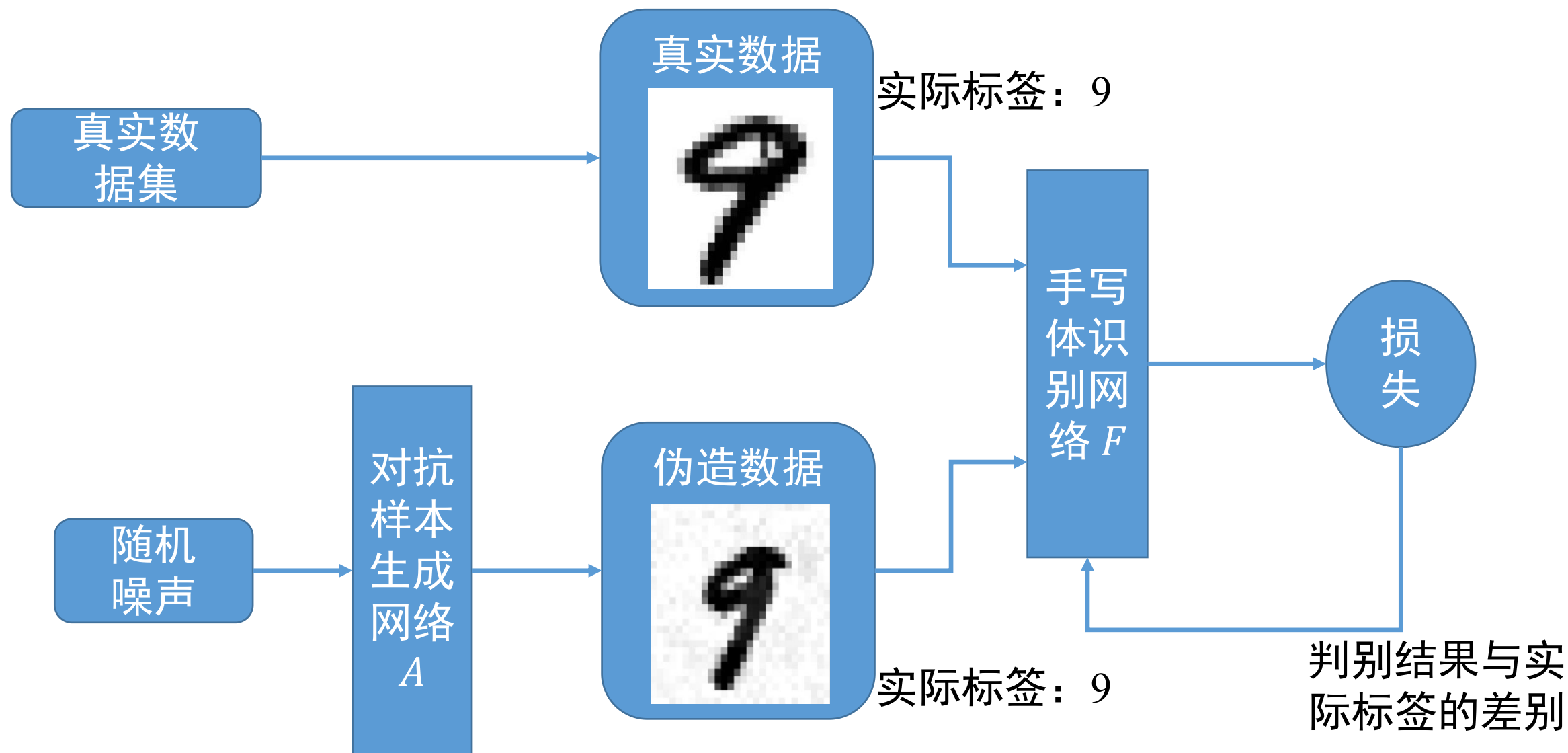
- 白盒攻击通常会对模型的每一部分进行逐层分解
 - 对每部分添加一定扰动，使模型结果逐步向误判目标偏移
 - 非常隐蔽，通过限制扰动的大小可使看起来与原样本差别很小



白盒攻击的防御策略：生成对抗网络

- 在训练时可以使用同样的方法增强模型训练的鲁棒性
 - 如生成对抗网络 (generative adversarial network, GAN) 就是一种有效的抵御白盒攻击的手段
- 生成对抗网络实际上由两个不同的网络组成：
 - 生成网络：通过神经网络将输入的一个服从简单分布的随机变量转化为能够欺骗判别网络的对抗样本
 - 判别网络：通过神经网络判断输入样本的真实类别
- 训练时两个网络交替进行优化，在对抗过程中共同提升性能

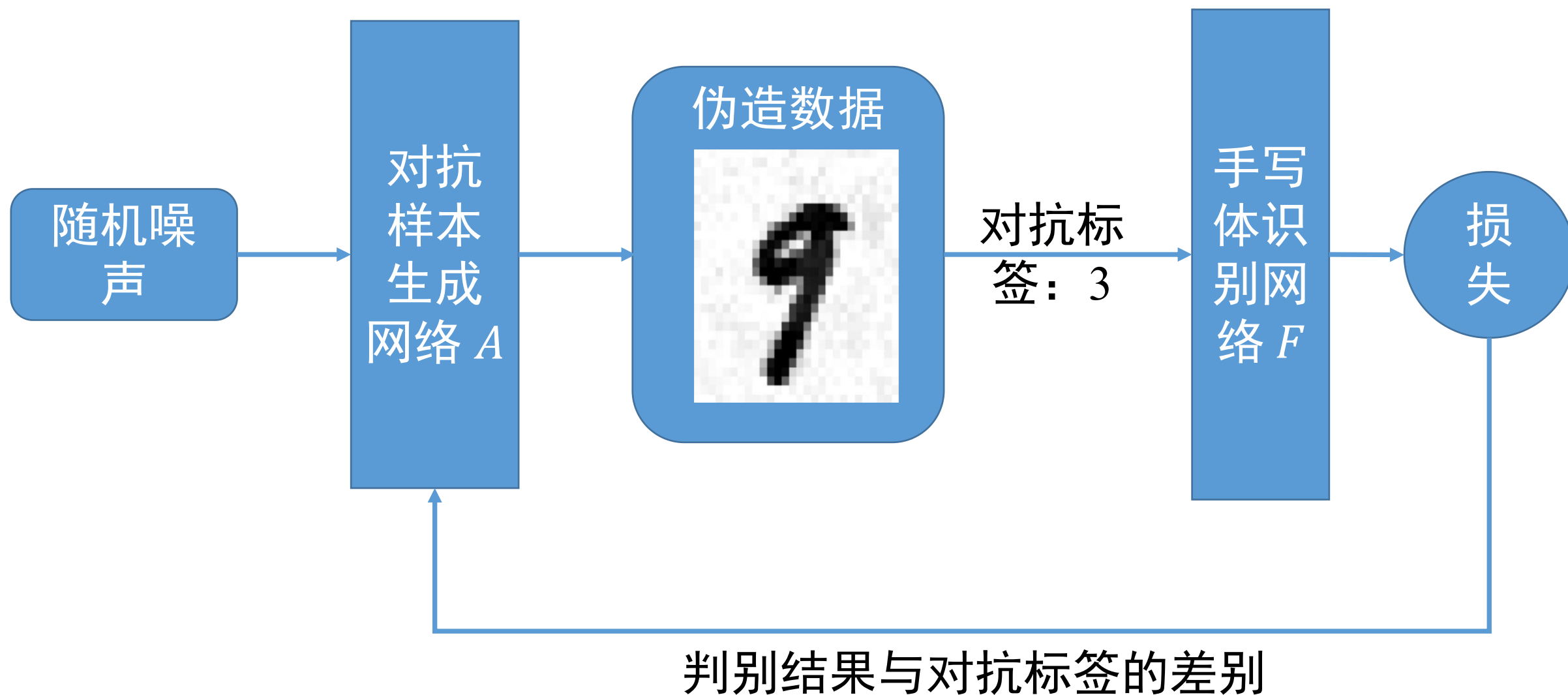
白盒攻击的防御策略：判别网络训练过程



白盒攻击的防御策略：判别网络训练过程

- 判别网络的训练过程分为两个方面
 - 根据真实数据及其真实标签来增强判别网络识别数据的能力
 - 根据成网络合成的伪造数据来增强判别网络抵抗干扰的能力
- 不论是真实数据还是对抗样本，算法都希望判别网络输出结果与图片标签一致。
 - 在上述过程中，对抗样本生成网络参数保持不变

白盒攻击的防御策略：对抗样本生成网络训练过程

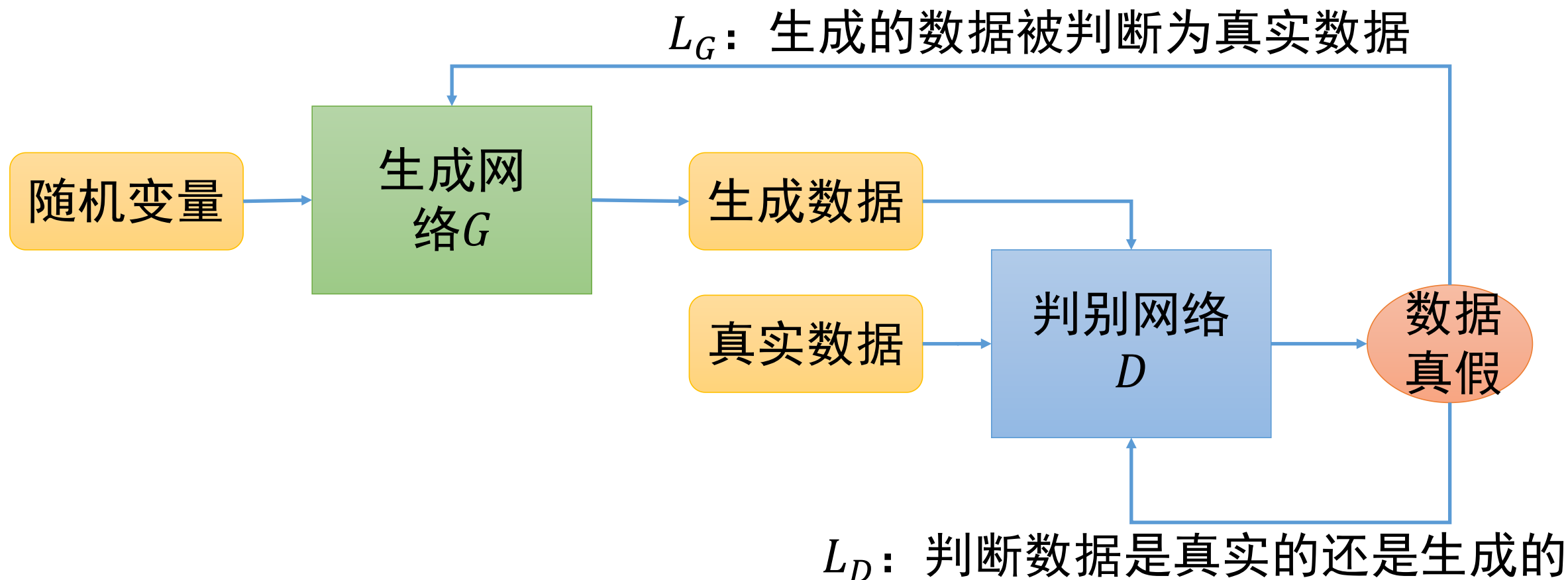


白盒攻击的防御策略：生成网络训练过程

- 生成网络的训练过程是判别网络的对抗过程
 - 根据判别网络识别的结果，不断提升对抗样本生成网络合成对抗样本的能力，从而使其能够产生更具有误导性的对抗样本
 - 此时希望合成的伪造数据被识别为伪造的对抗标签而不是合成数据所对应的实际标签
- 在该过程中，判别网络参数保持不变。

生成对抗网络

- 生成对抗网络是深度学习中常用的一种生成模型
- 生成对抗网络一般可如下表示：

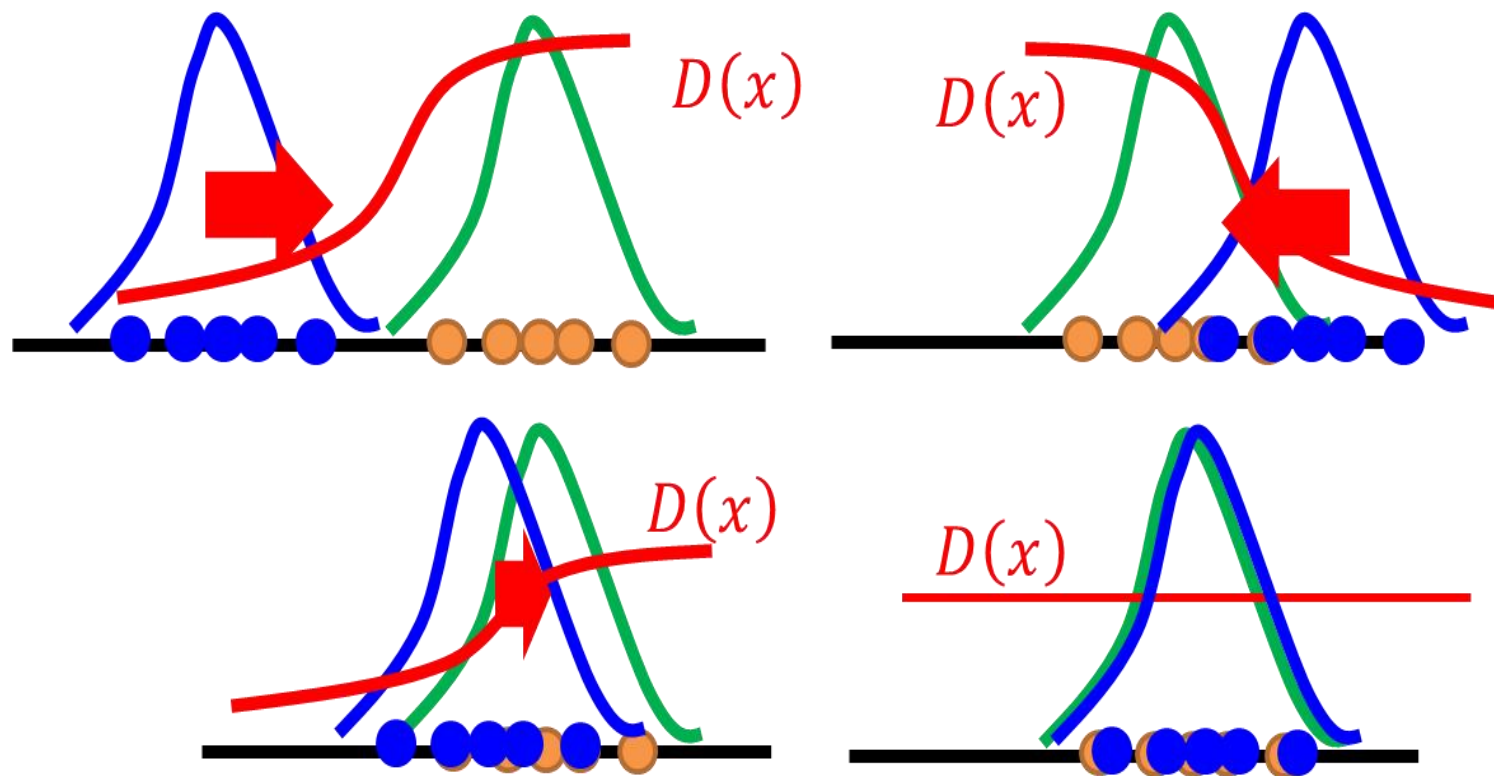


生成对抗网络

- **生成网络通过简单的分布来拟合复杂的分布**
 - 并从所拟合的分布中采样得到符合一定要求的样本
- **在训练过程中，判别网络需分辨出真实数据与生成数据的不同、生成网络会逐渐学习得到数据的真实分布情况**
 - 随着不断优化生成网络，判别网络逐渐无法分辨生成网络所合成数据的真伪
- **最后，生成网络完全模拟出了真实数据的分布情况**
 - 使得区别网络无法分辨数据的真伪，开始随机猜测结果，此时对抗网络的训练达到收敛

生成对抗网络

— 分类效果
— 真实数据分布
— 生成数据分布



生成对抗网络

生成对抗网络有两个优化目标：生成网络和判别网络。首先优化判别网络模型参数

- $G^* = \operatorname{argmin}_G \max_D V(G, D)$

- $V = E_{x \sim P_{data}} [\log D(x)] + E_{x \sim P_G} [\log(1 - D(x = G(z)))]$



- **判别网络目标最大化 $\log D(x)$**

- $P_{data}(x) \log D(x) + P_G(x) \log(1 - D(x))$

- 对 $D(x)$ 求导可得最优分类器：

- $D^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_G(x)} \longrightarrow \text{最优的分类器能准确区分真假样本}$

生成对抗网络

生成对抗网络有两个优化目标：生成网络和判别网络。接着优化生成网络参数

- $G^* = \operatorname{argmin}_G \max_D V(G, D)$

- $V = E_{x \sim P_{data}} [\log D(x)] + E_{x \sim P_G} [\log (1 - D(x = G(z)))]$



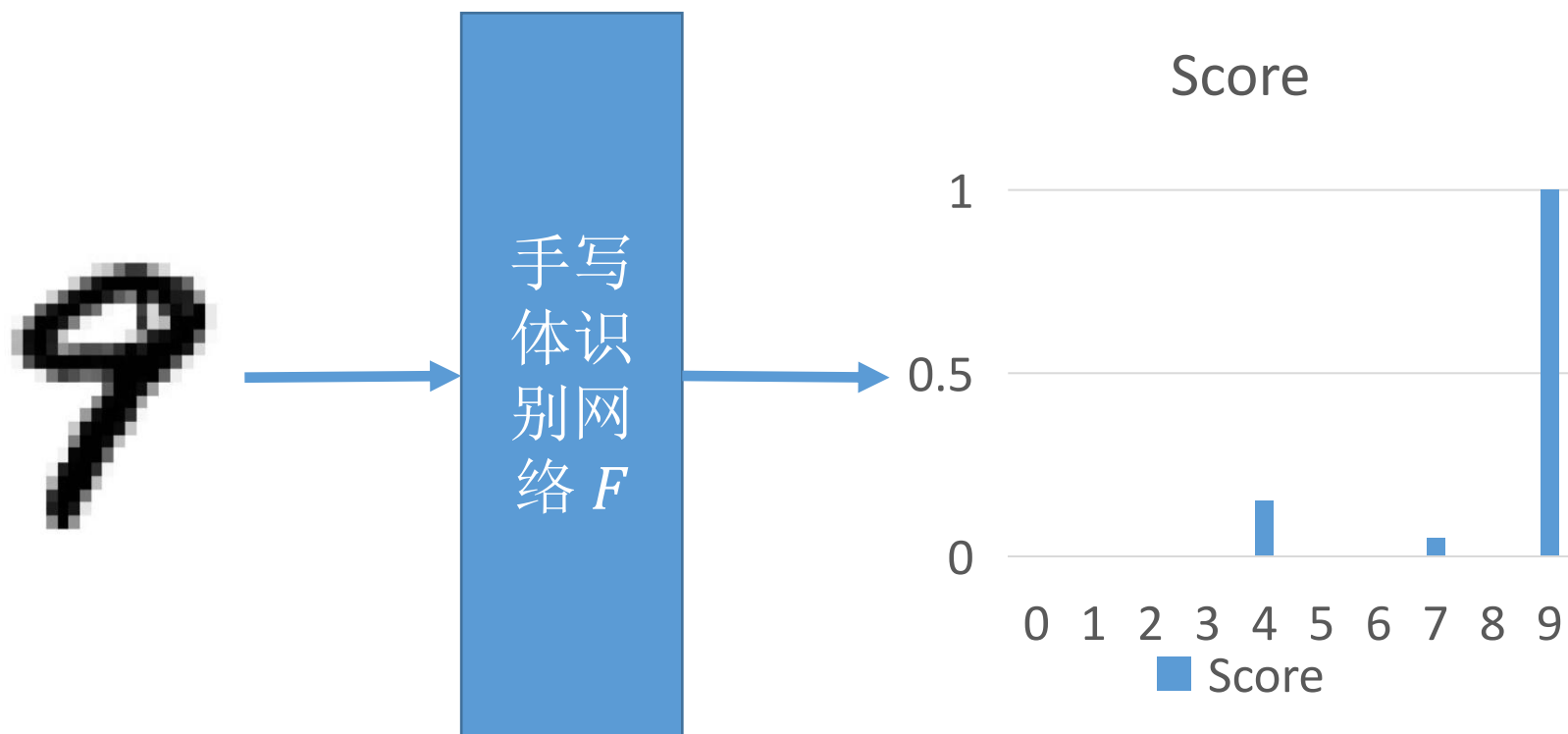
- 生成网络的优化目标是 최소화 $\log (1 - D(G(z)))$ ，将最优的判别网络代入这一优化目标

- $$\begin{aligned} V(G, D^*) &= E_{x \sim P_{data}} \left[\log \frac{P_{data}(x)}{P_{data}(x) + P_G(x)} \right] + E_{x \sim P_G} \left[\log \frac{P_G(x)}{P_{data}(x) + P_G(x)} \right] \\ &= -2 \log 2 + \text{KL} \left(P_{data}(x) \parallel \frac{P_{data}(x) + P_G(x)}{2} \right) + \text{KL} \left(P_G(x) \parallel \frac{P_{data}(x) + P_G(x)}{2} \right) \\ &= -2 \log 2 + 2JS(P_{data}(x) \parallel P_G(x)) \end{aligned}$$

- $G^* = \operatorname{argmin}_G D_f(P_{data} \parallel P_G)$  最优生成网络能生成与真实样本具有相同分布的数据

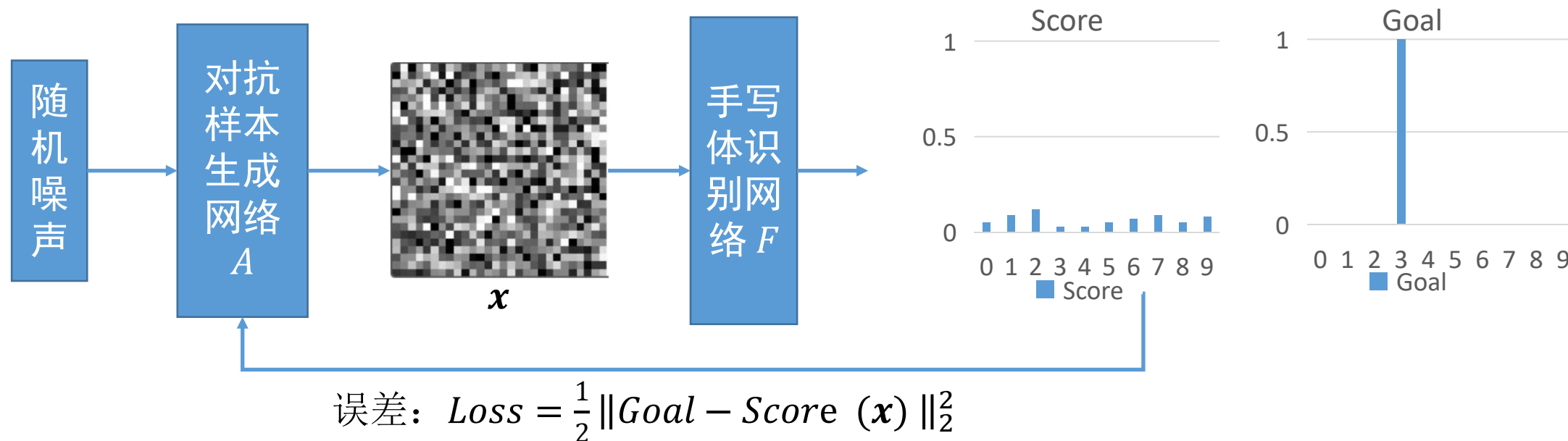
对抗样本的生成：无针对攻击 (Non-Targeted Attack)

- 无针对攻击：任意生成输入数据，使得模型输出为指定结果
- 假设已经获得一个训练好的神经网络 F ，能够识别手写数字。现在希望生成能够干扰神经网络 F 的对抗样本 i ，使得对抗样本 i 被错误识别为数字3



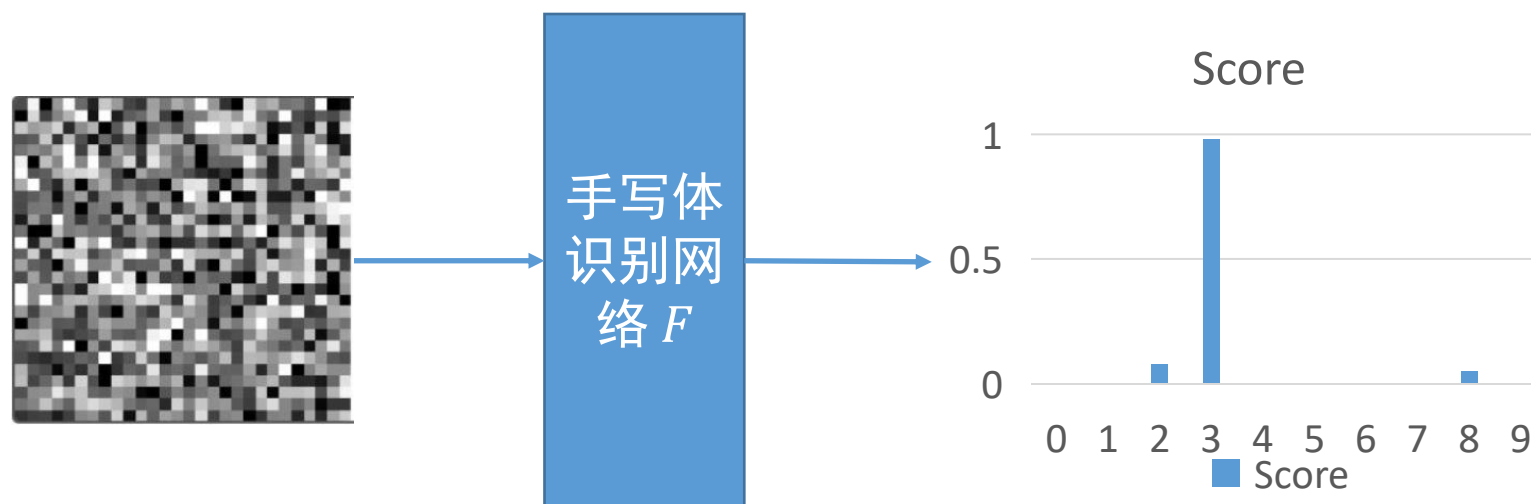
对抗样本的生成：无针对攻击 (Non-Targeted Attack)

- 训练一个能够生成对抗样本的生成网络 A ，其能够将随机噪声转化为一副对抗样本图片
- 将对抗样本输入手写体识别网络 F ，使用 F 的输出与预设目标之间的误差来优化对抗样本生成网络 A



对抗样本的生成：无针对攻击 (Non-Targeted Attack)

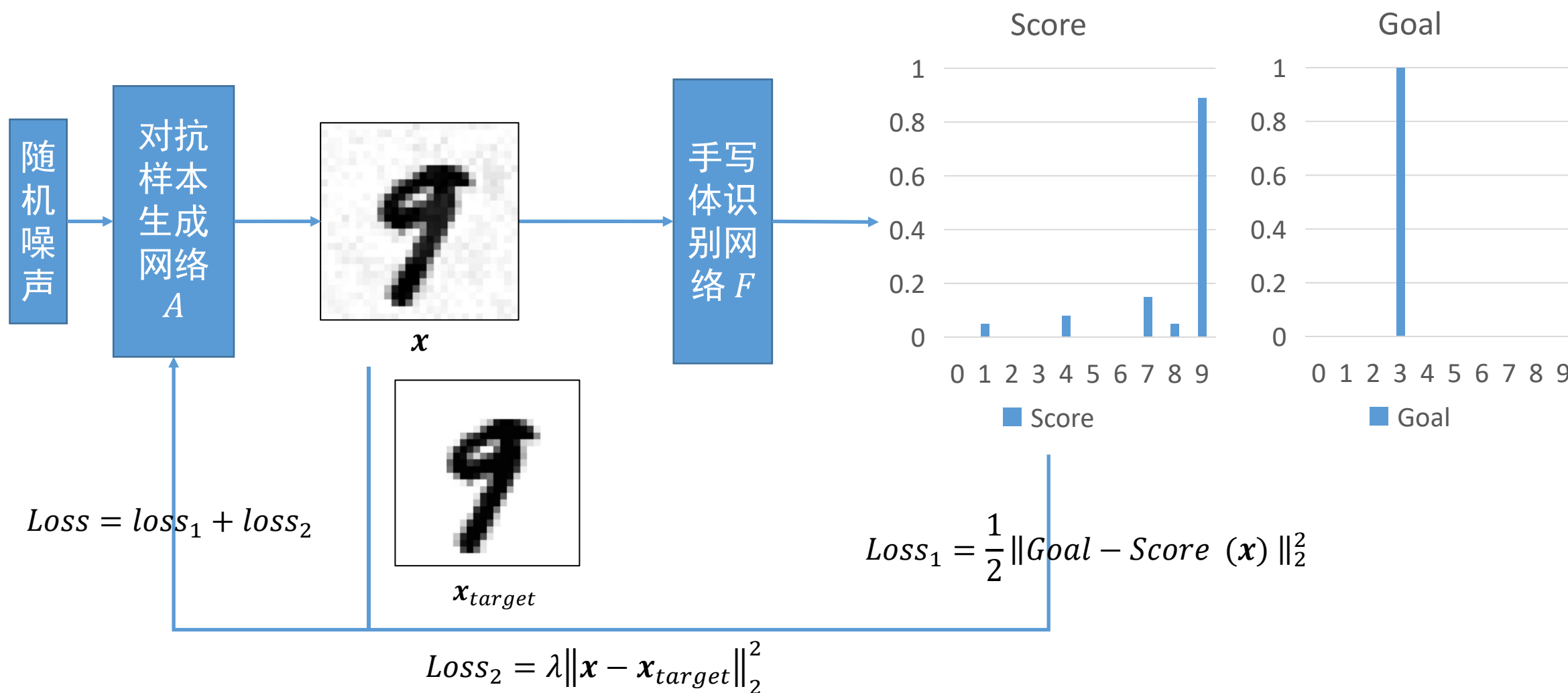
- 通过迭代训练，使得对抗样本生成网络 A 可生成众多被手写体识别网络 F 错误分类为3的对抗样本。
- 这样，手写体识别网络 F 被攻击成功。
- 在对抗样本的生成过程中，没有用到攻击模型的内部结构知识，所以这是一次黑盒攻击



对抗样本的生成：有针对攻击 (Targeted Attack)

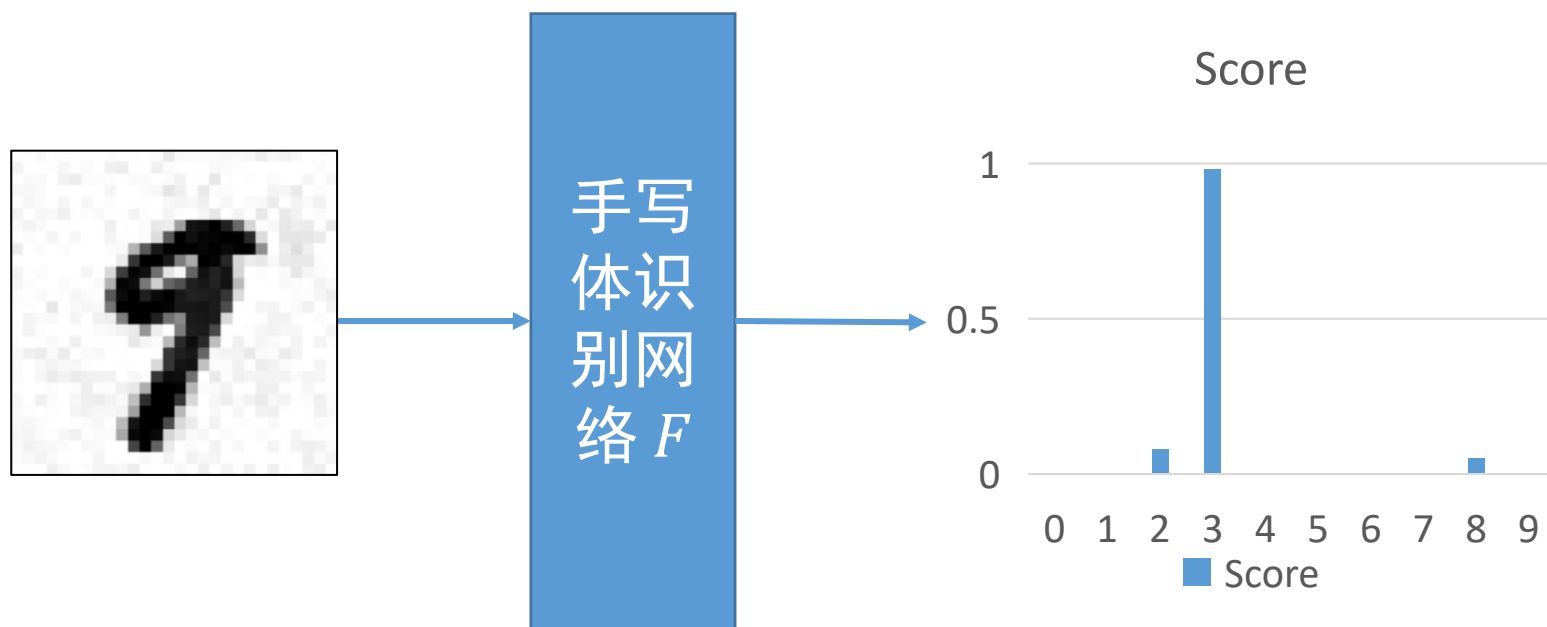
- 有针对攻击：生成人类与模型判断相互迥异的对抗样本
 - 假设已经获得一个训练好的神经网络 \hat{f} ，其能够识别手写体数字，现在想生成能够干扰神经网络 \hat{f} 的有针对的对抗样本 \hat{x} 。
 - 如：对抗样本 \hat{x} 被人识别为9，但被 \hat{f} 错误识别为3。

对抗样本的生成：有针对攻击 (Targeted Attack)



对抗样本的生成：有针对攻击 (Targeted Attack)

- 这种攻击方式同样也是黑盒攻击。可见，手写体识别网络 F 被攻击成功。



黑盒攻击的防御策略

- 常用的黑盒攻击防御策略有：

- 数据压缩：通过对输入数据进行压缩或者降维，在保证识别准确率的情况下提升模型对干扰攻击的鲁棒性
- 数据随机化：对训练数据进行随机缩放、增强等操作，提升模型的鲁棒性
- 训练额外的网络来判断训练数据是否为攻击样本

谢谢!