

# GAME THEORY & AI

Girraj Jayaswal (100050030)

Kumar Rahul Ranjan (100050038)

Jayanth (100050041)

# Motivation for Seminar

## Students get class-wide As by boycotting test, solving Prisoner's Dilemma

Cory Doctorow at 7:11 am Tue, Feb 19

3.9k

Like

609

Tweet

6,346

216

+1

Kindle

Johns Hopkins computer science prof Peter Fröhlich grades his students' tests on a curve -- the top-scoring student gets an A, and the rest of the students are graded relative to that brainiac. But last term, his students came up with an ingenious, cooperative solution to this system: they all boycotted the test, meaning that they all scored zero, and that zero was the top score, and so they all got As. The prof was surprisingly cool about it:

Fröhlich took a surprisingly philosophical view of his students' machinations, crediting their collaborative spirit. "The students learned that by coming together, they can achieve something that individually they could never have done," he said via e-mail. "At a school that is known (perhaps unjustly) for competitiveness I didn't expect that reaching such an agreement was possible."

The story of the boycott is a sterling example of how computer networks solve collective action problems -- the students solved a prisoner's dilemma in a mutually optimal way without having to iterate, which is impressive.

— FEATURED —



### REVIEW

Punk Rock Jesus: media-second coming/reality TV comic



### FEATURE

Where'd You Go, Bernadette funny/dark novel about the disintegration of a Microsoft family



### REVIEW

The Dude and the Zen Master

— COMICS —



### BRAIN ROT

Brain Rot: R Budd Dwyer



### TOM THE DANCING BUG

TOM THE DANCING BUG: Education of Louis - Spect Sport



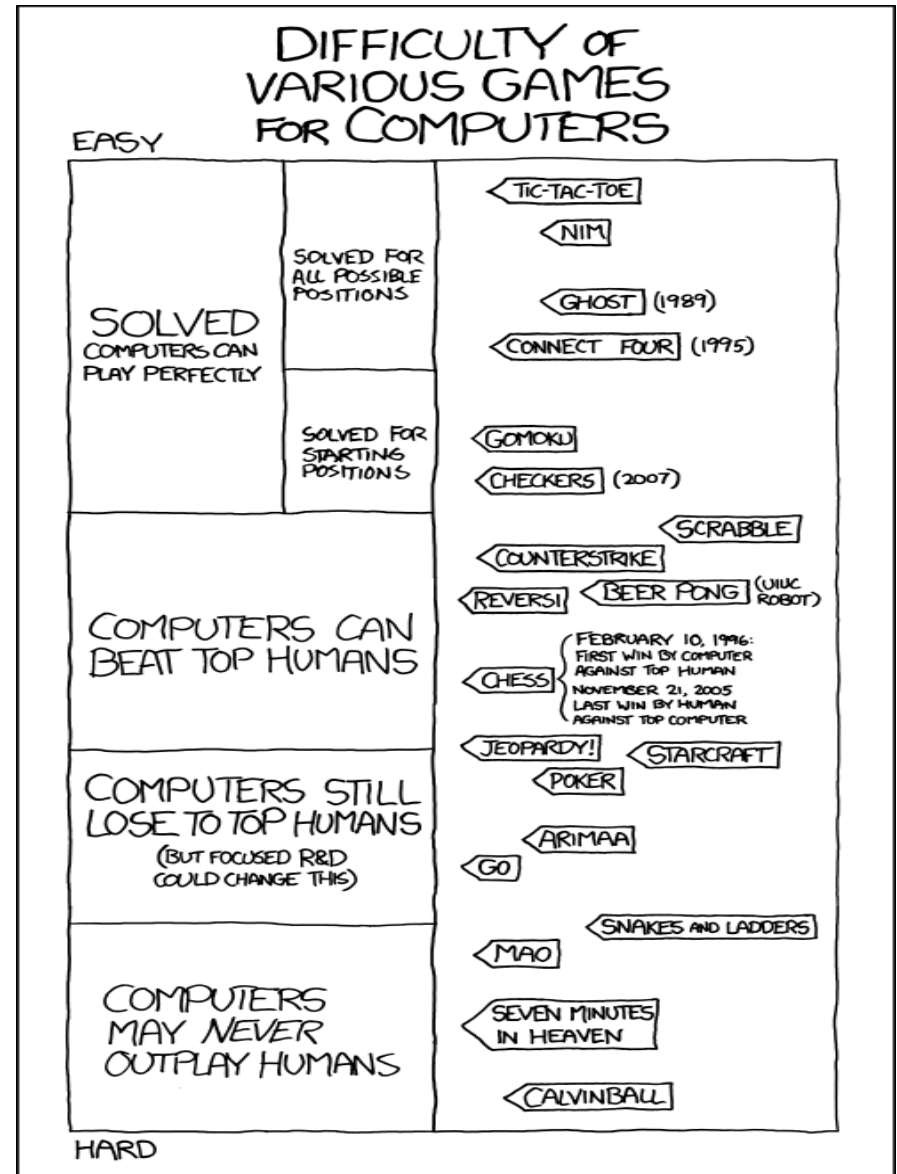
### BRAIN ROT

Brain Rot: Hip Hop Family Afrika Bambaataa Planet R

# INTRODUCTION

Game Theory is mathematical study of interaction between rational, self-interested agents.

Game Theory applies mathematical models to this interaction under the assumption that each agent's actions impact the pay-offs of all other participants in the game.



# Defining & Representing Games

- Finite,  $n$ -person game:  $\langle N, A, u \rangle$ :
  - $N$  is a finite set of  $n$  **players**, indexed by  $i$
  - $A = A_1 \times \dots \times A_n$ , where  $A_i$  is the **action set** for player  $i$ 
    - $(a_1, \dots, a_n) \in A$  is an **action profile**, and so  $A$  is the space of action profiles
  - $u = \langle u_1, \dots, u_n \rangle$ , a **utility function** for each player, where  $u_i : A \mapsto \mathbb{R}$
- Writing a 2-player game as a **matrix**:
  - row player is player 1, column player is player 2
  - rows are actions  $a \in A_1$ , columns are  $a' \in A_2$
  - cells are outcomes, written as a tuple of utility values for each player

*The normal-form representation of an  $n$ -player game specifies the players' strategy spaces  $S_1, S_2, \dots, S_n$  and their payoff functions  $u_1, u_2, \dots, u_n$ . We denote this game by*

$$G = \{S_1, S_2, \dots, S_n; u_1, u_2, \dots, u_n\}.$$

# Prisoner's Dilemma



- ▶ Two suspects are arrested and charged with a crime. The police lack sufficient evidence to convict the suspects, unless at least one confesses. The police explain the consequences that will follow from the actions they could take.
- ▶ If neither confesses then both will be sentenced to one month in jail.
- ▶ If both confess then both will be sentenced to jail for six months.
- ▶ Finally, if one confesses but the other does not, then the confessor will be released immediately but the other will be sentenced to nine months in jail.

# Battle of the Sexes

- ▶ A man and a woman are trying to decide on an evening's entertainment.
- ▶ While at separate workplaces, Pat and Chris must choose to attend either the opera or a football match.
- ▶ Both players would rather spend the evening together than apart, but Pat would rather they be together at the football match while Chris would rather they be together at the opera.



# Bi-Matrix Representation

## Prisoner's dilemma

		Prisoner 1	
		Confess	Not Confess
Prisoner 2	Confess	-6,-6	0,-9
	Not Confess	-9,0	-1,-1

## Battle of the Sexes

		Chris	
		Football	Opera
Pat	Football	2,1	0,0
	Opera	0,0	1,2

# Strategies

- Suppose the agents agent 1, agent 2, ..., agent  $n$
- For each  $i$ , let  $S_i = \{\text{all possible strategies for agent } i\}$ 
  - $s_i$  will always refer to a strategy in  $S_i$
- A **strategy profile** is an  $n$ -tuple  $S = (s_1, \dots, s_n)$ , one strategy for each agent
- **Utility**  $U_i(S) = \text{payoff for agent } i \text{ if the strategy profile is } S$
- $s_i$  **strongly dominates**  $s_i'$  if agent  $i$  always does better with  $s_i$  than  $s_i'$

$$\forall s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n,$$

$$U_i(s_1, \dots, s_{i-1}, s_i, s_{i+1}, \dots, s_n) > U_i(s_1, \dots, s_{i-1}, s_i', s_{i+1}, \dots, s_n)$$

- $s_i$  **weakly dominates**  $s_i'$  if agent  $i$  never does worse with  $s_i$  than  $s_i'$ , and there is at least one case where agent  $i$  does better with  $s_i$  than  $s_i'$ ,

$$\forall s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n, U_i(\dots, s_i, \dots) \geq U_i(\dots, s_i', \dots)$$

and

$$\exists s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n \quad U_i(\dots, s_i, \dots) > U_i(\dots, s_i', \dots)$$



# Iterated Elimination

	Left	Middle	Right
Up	1,0	1,2	0,1
Down	0,3	0,1	2,0

	Left	Middle
Up	1,0	1,2
Down	0,3	0,1

	Left	Middle
Up	1,0	1,2

	Middle
Up	1,2

# Pure and Mixed Strategies

- **Pure strategy:** select a single action and play it
  - Each row or column of a payoff matrix represents both an action and a pure strategy
- **Mixed strategy:** randomize over the set of available actions according to some probability distribution
  - Let  $A_i = \{\text{all possible actions for agent } i\}$ , and  $a_i$  be any action in  $A_i$
  - $s_i(a_j) = \text{probability that action } a_j \text{ will be played under mixed strategy } s_i$
- The **support** of  $s_i$  is
  - $\text{support}(s_i) = \{\text{actions in } A_i \text{ that have probability } > 0 \text{ under } s_i\}$
- A pure strategy is a special case of a mixed strategy
  - support consists of a single action
- **Fully mixed strategy:** every action has probability  $> 0$ 
  - i.e.,  $\text{support}(s_i) = A_i$

# Expected Utility

**Definition of Mixed Strategies:** In the normal-form game  $G = \{S_1, S_2, \dots, S_n; u_1, u_2, \dots, u_n\}$ , suppose  $S_i = \{s_{i1}, s_{i2}, \dots, s_{ik}\}$ . Then a mixed strategy for player  $i$  is a probability distribution  $p = (p_{i1}, p_{i2}, \dots, p_{ik})$ , where  $0 < p_{ik} < 1$  for  $k = 1, \dots, K$  and  $p_{i1} + p_{i2} + \dots + p_{iK} = 1$

- A payoff matrix only gives payoffs for pure-strategy profiles
- Generalization to mixed strategies uses *expected utility*
- Let  $S = (s_1, \dots, s_n)$  be a profile of mixed strategies
  - For every action profile  $(a_1, a_2, \dots, a_n)$ , multiply its probability and its utility
    - $U_i(a_1, \dots, a_n) s_1(a_1) s_2(a_2) \dots s_n(a_n)$
  - The expected utility for agent  $i$  is
$$U_i(s_1, \dots, s_n) = \sum_{(a_1, \dots, a_n) \in A} U_i(a_1, \dots, a_n) s_1(a_1) s_2(a_2) \dots s_n(a_n)$$

# Best Response

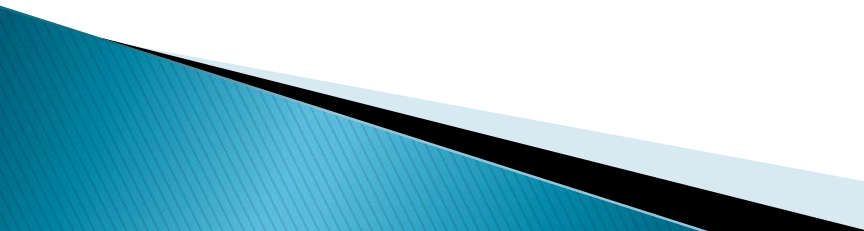
- Some notation:
  - If  $S = (s_1, \dots, s_n)$  is a strategy profile, then  $S_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$ ,
    - i.e.,  $S_{-i}$  is strategy profile  $S$  without agent  $i$ 's strategy
  - If  $s_i'$  is any strategy for agent  $i$ , then
    - $(s_i', S_{-i}) = (s_1, \dots, s_{i-1}, s_i', s_{i+1}, \dots, s_n)$
  - Hence  $(s_i, S_{-i}) = S$
- $s_i$  is a **best response** to  $S_{-i}$  if
$$U_i(s_i, S_{-i}) \geq U_i(s_i', S_{-i})$$
 for every strategy  $s_i'$  available to agent  $i$
- $s_i$  is a **unique** best response to  $S_{-i}$  if
$$U_i(s_i, S_{-i}) > U_i(s_i', S_{-i})$$
 for every  $s_i' \neq s_i$

# Nash Equilibrium

- ▶ A strategy profile  $s = (s_1, s_2, \dots, s_n)$  is a Nash equilibrium if for every  $i$ ,  $s_i$  is a best response to  $S_{-i}$ , i.e., no agent can do better by unilaterally changing his/her strategy
- ▶ **Theorem (Nash, 1951):** Every game with a finite number of agents and action profiles has at least one Nash equilibrium

	Left	Centre	Right
Top	0, <u>4</u>	<u>4</u> ,0	5,3
Middle	<u>4</u> ,0	0, <u>4</u>	5,3
Bottom	3,5	3,5	<u>6</u> , <u>6</u>

# REPRESENTATION, REASONING & LEARNING

- ▶ Both game theory and Artificial Intelligence deal with “intelligent” agents, who are embodied in a complex world.
  - ▶ These agents may interact with other agents, and try to optimize their behavior, while employing various reasoning and learning techniques.
  - ▶ The above three issues are fundamental both to Game theory/Economics and to AI/CS.
- 

# REASONING

- ▶ Protocols for agent interactions that are subject to rational constraints, i.e. agents will follow their own interests.
- ▶ Vickrey Auction – highest bidder pays the second highest bid, truth revealing equilibrium
- ▶ Protocols for distributed environments, emphasizing computational constraints and distributed systems features
- ▶ Network Routing – Pay the owner declared cost plus added value

Game Theory

Artificial Intelligence

# Learning

- ▶ Emphasizes learning as a descriptive tool, explaining the emergence of Nash equilibrium or predicting agents' behavior
- ▶ In an MDP, the agent is in one of finitely many states, and can select one of many actions, which lead to a certain payoff and to a new state
- ▶ Emphasizes a normative approach, and deals with algorithms for obtaining high payoffs in uncertain environments based on observed feedback
- ▶ In Stochastic Game, MDP is modeled by a game between two players, whose actions determine their payoffs as well as the transition probability.

Game Theory

Artificial Intelligence



# REPRESENTATION

- ▶ Modeling agents as expected utility maximizers, i.e. it assigns probabilities to the states of the environment, and utilities to various outcomes or consequences, and chooses the action, protocol, strategy or policy that maximizes its expected utility.
- ▶ Work in CS/AI has considered, in addition to that classical decision criterion, other forms of decision making. This includes, for example, competitive analysis (aka the competitive ratio decision criterion) , and the safety-level (worst case) maximization approaches.

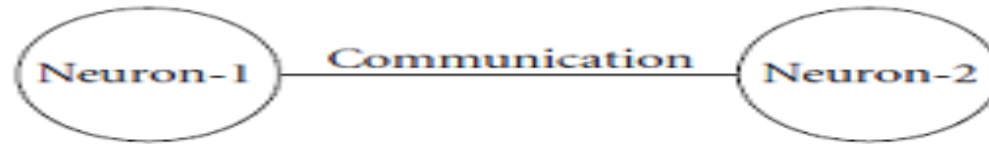
Game Theory

Artificial Intelligence

# Application of Game Theory to Neural Networks

- ▶ The model has the following global behavior: if Neuron-1 fires, then Neuron-2 shall fire, and if Neuron-1 is at rest, then Neuron-2 shall be at rest (it is possible to assume an information exchange via biochemical substances or electrical signals between two)
- ▶ Neuron-1 can either fire or be at rest, and Neuron-2 has to respond accordingly

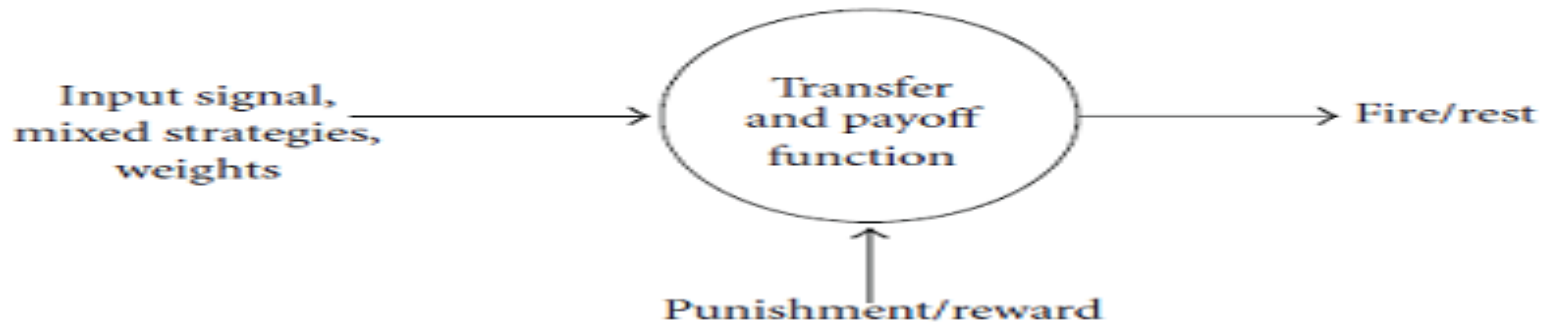
$$\text{▶ } f(x) = \begin{cases} \text{Fire} & \text{if } x > t, \\ \text{Rest} & \text{otherwise.} \end{cases}$$



(a)

		Neuron-2	
		Fire	Rest
Neuron-1	Fire	$R, R$	$P, P$
	Rest	$P, P$	$R, R$

(b)

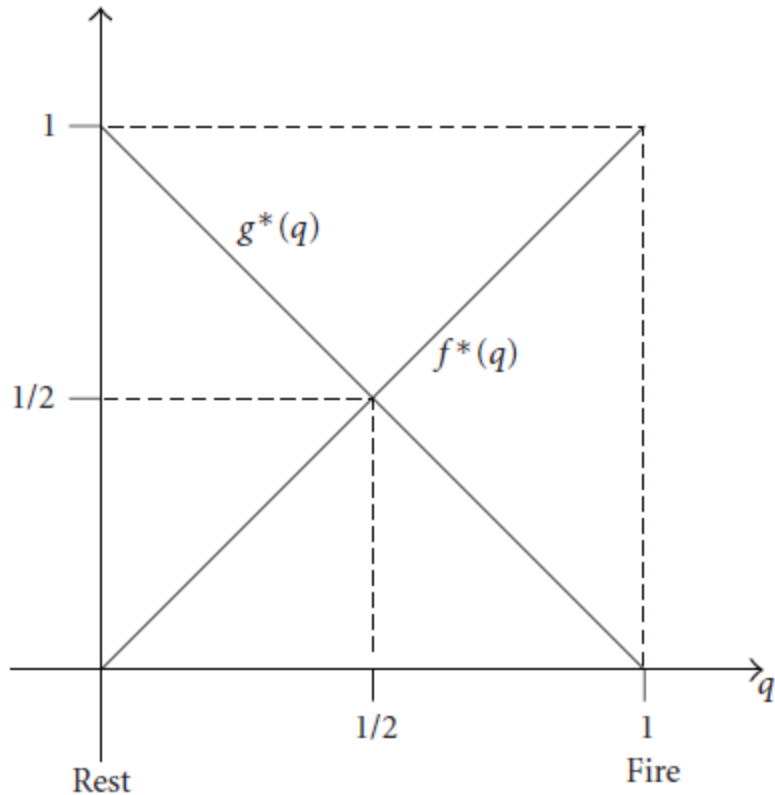


(c)

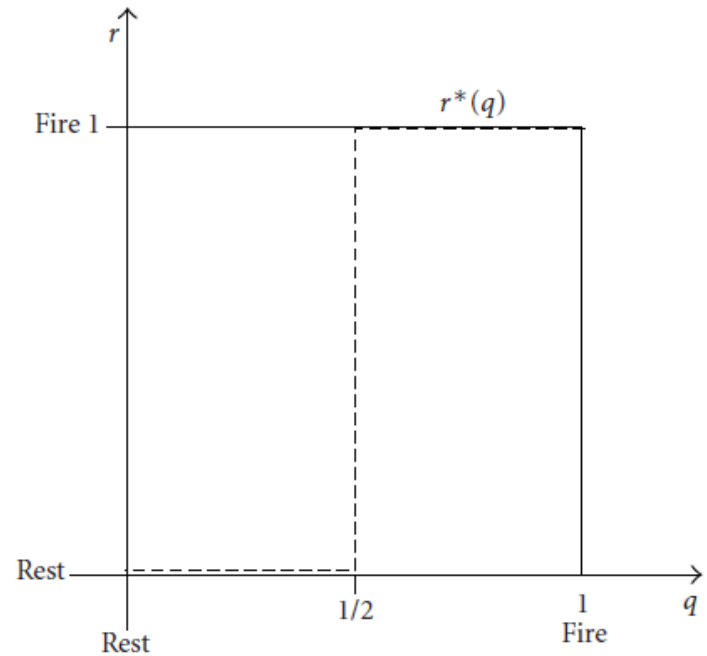
*Relationships between (a) biological neurons, (b) game theory, and (c) artificial neurons.*

# Game Theoretic Interpretations

- ▶ Player-1 believes that Player-2 will play the mixed strategy  $(q, 1 - q)$ , then the expected payoff for Player-1 for playing the pure strategy Fire is  $f^*(q) = q$  and for playing the pure strategy Rest is  $g^*(q) = 1 - q$ .
- ▶ If  $q > 1/2$ , then  $f^*(q) > g^*(q)$  in which case Player-1 should play strategy Fire else if  $q < 1/2$ , then  $g^*(q) > f^*(q)$  in which case Player-1 should adopt strategy Rest. If  $q = 1/2$ , Player-1 is indifferent about which strategy to play.

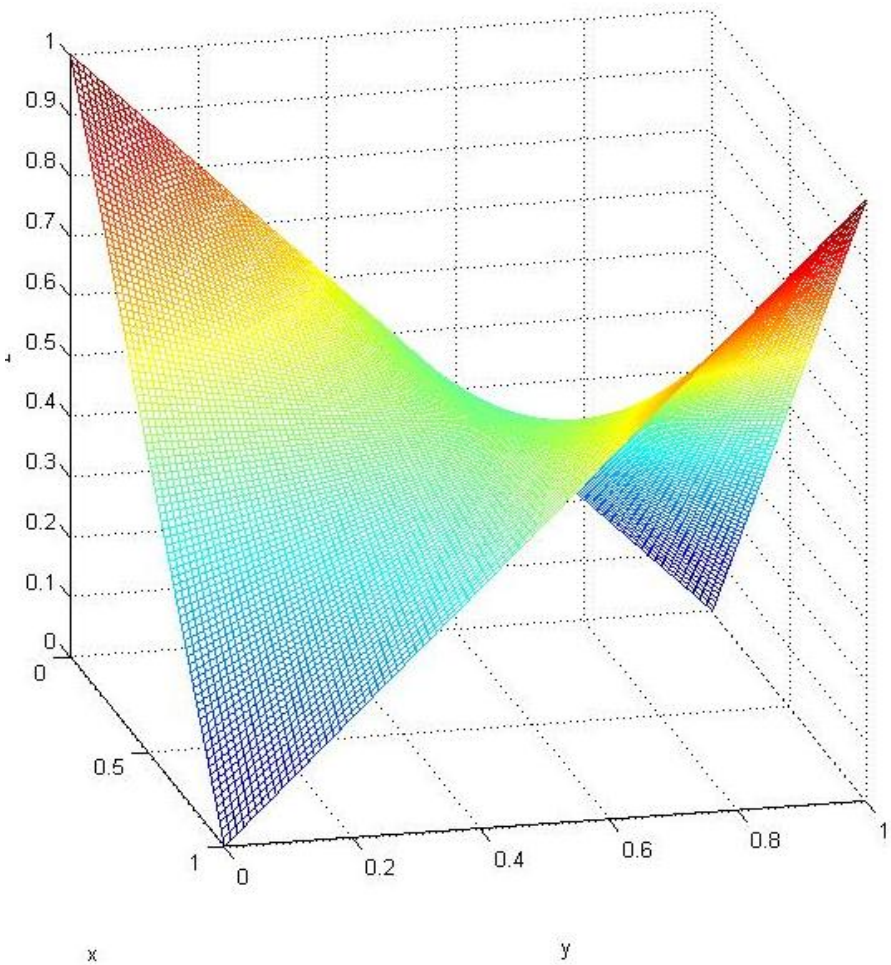


*Decision-making support for Player-1 if Player-1 believes that Player-2 plays the mixed strategy  $(q, 1 - q)$ .*

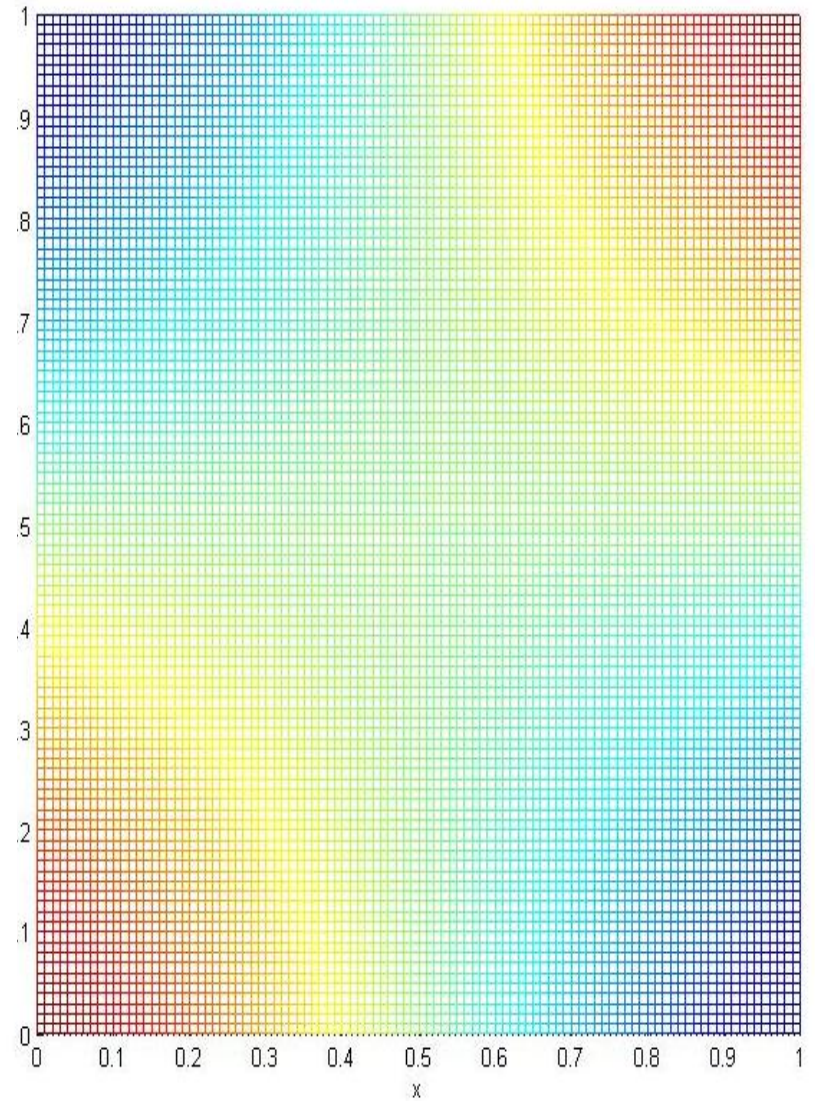


*Player-1's best response (maximizing the expected payoff  $r^*(q)$ ) from playing  $(r, 1 - r)$  when Player-2 plays  $(q, 1 - q)$ . (The additional information on the vertical axis ( $r$ , and strategies Fire, Rest) aims to support the interpretation of this figure.)*

- ▶ Player-1's expected payoff  $r^*(q)$  from playing the mixed strategy  $(r, 1 - r)$  when Player-2 plays the mixed strategy  $(q, 1 - q)$  is the weighted sum of the expected payoff for each of the pure strategies (Fire, Rest) where the weights are the probabilities  $(r, 1 - r)$ .
- ▶ 
$$r^*(q) = r \cdot q \cdot (1) + r \cdot (1 - q) \cdot (0) + (1 - r) \cdot q \cdot (0) + (1 - r) \cdot (1 - q) \cdot (1)$$
$$= r \cdot q + (1 - r) \cdot (1 - q) = 1 - q + r(2q - 1).$$
- ▶ If Player-2 plays mixed strategy  $(q, 1 - q)$ , then Player-1's best response is to play
  - strategy Fire if  $q > 1/2$ ,
  - strategy Rest if  $q < 1/2$ , and
  - any strategy if  $q = 1/2$ .



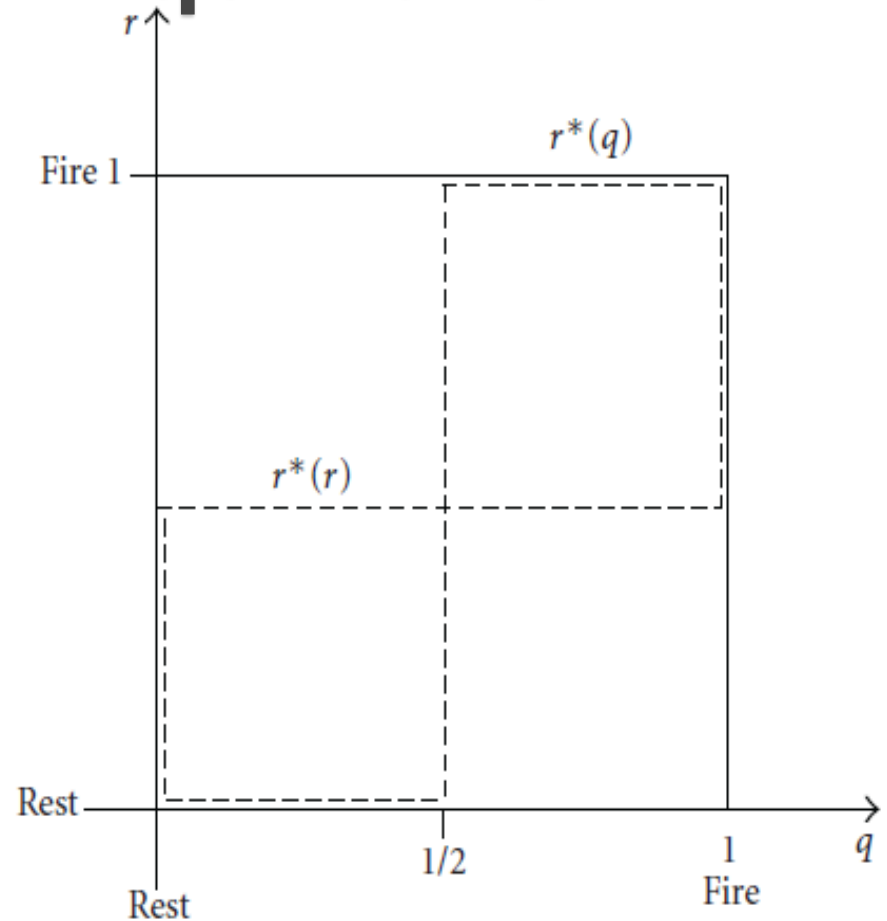
*3-D Plot of  $z=r.q + (1-r).(1-q)$*



*Mesh Plot of  $z=r.q + (1-r).(1-q)$*

# Neural Nash Equilibrium

- ▶ The interesting features in Figure 8 include those points where  $r^*(q)$  and  $r^*(r)$  intersect (i.e., points  $(0, 0)$ ,  $(1/2, 1/2)$ , and  $(1, 1)$ ).
- ▶ If Neuron-1 fires then Neuron-2's best response is to fire too.
- ▶ If Neuron-1 is at rest, then Neuron-2's best response is to be at rest too.
- ▶ An interesting situation exists for point  $(1/2, 1/2)$ . This situation may be interpreted as if
  - ▶ Neuron-2 is unaware about the state (strategy) of Neuron-1,
  - ▶ then Neuron-2 may play either strategy, and vice versa.



*Combined view of best responses for Player-1 and Player-2. The three intersections between  $r^*(q)$  and  $r^*(r)$  are the Nash equilibriums in the game.*

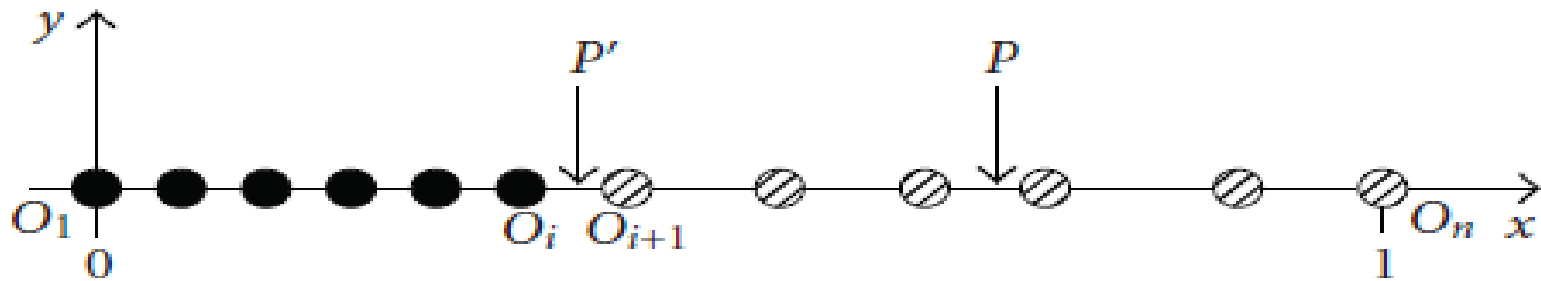


# Game Theory and Neural Network Learning

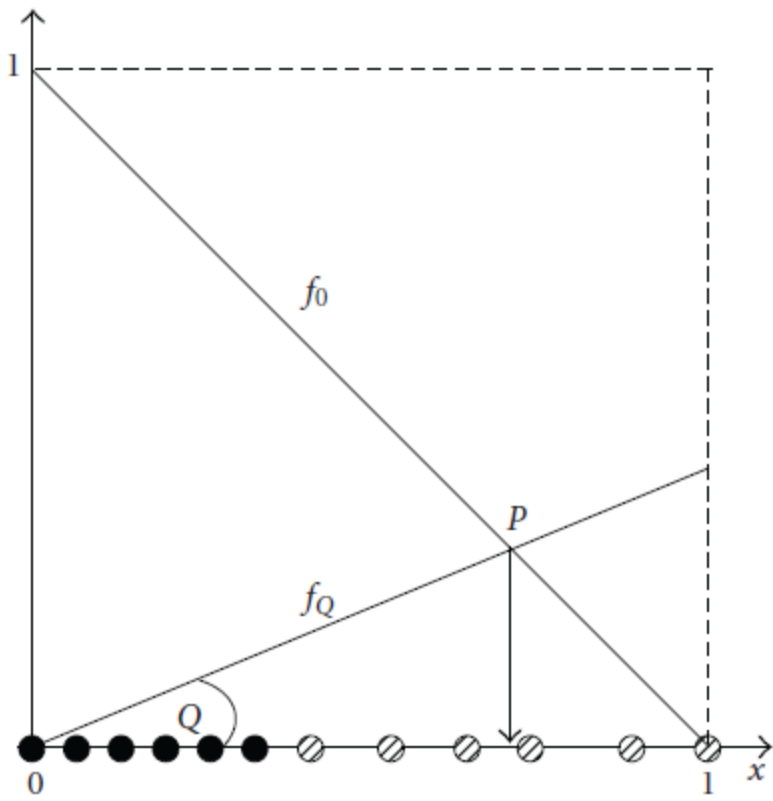
For the algorithm, imagine a one-dimensional, linearly separable, and supervised learning classification task.

The classification scenario in figure takes place in an arbitrary real-valued  $x, y$  coordinate system, involving  $n$  objects, such that for every object  $i$  yields  $x_i \in [0, 1]$ .

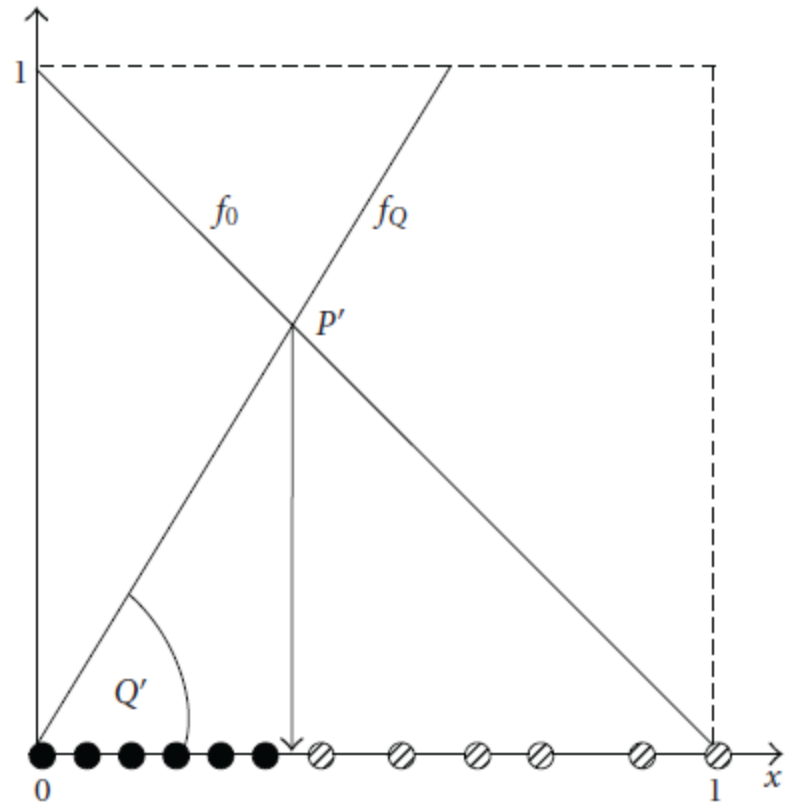
In their current positions,  $P'$  correctly separates all objects into their corresponding classes, whereas  $P$  incorrectly classifies objects. At the start of a learning scenario,  $P$  may have been positioned randomly and in successive steps the learning algorithm may have moved this starting point until it finished in location  $P'$ , which is a solution to the problem.



*A one-dimensional, linearly separable, and supervised learning classification task.*

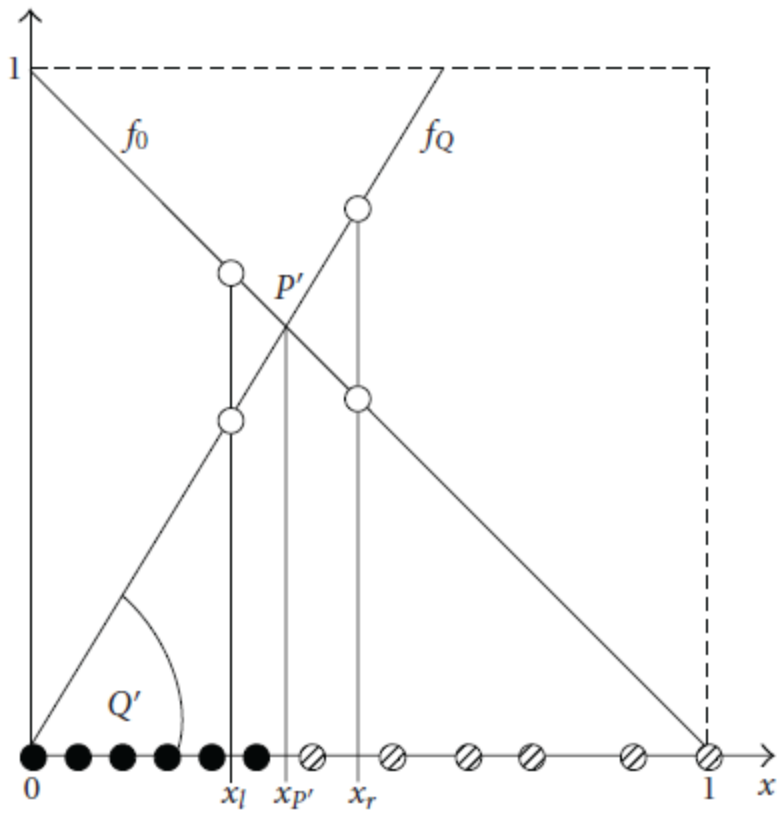


(a)

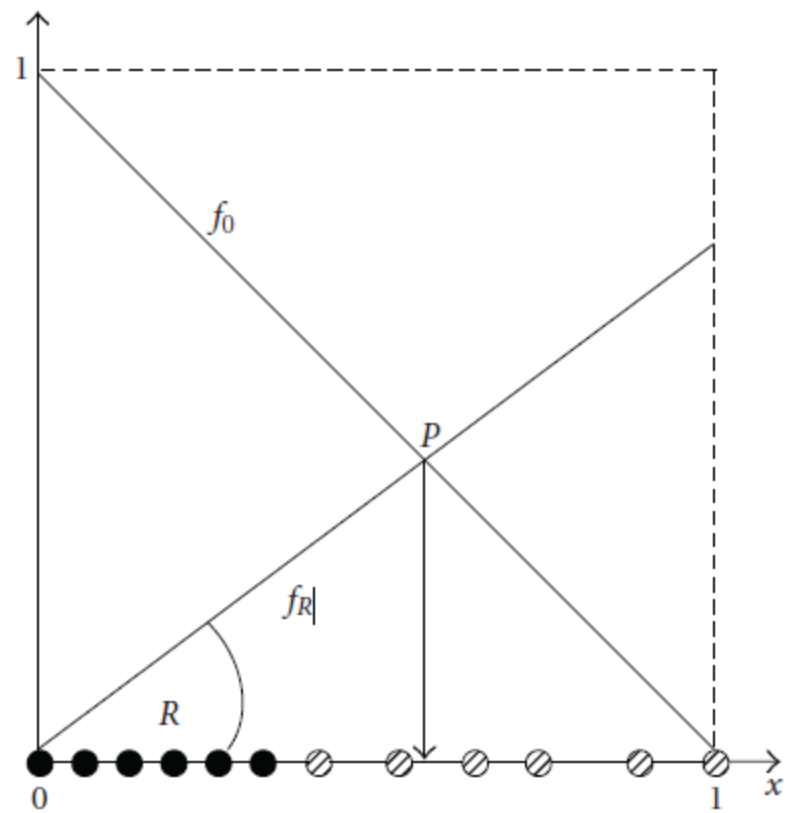


(b)

*Neuron-1's point of view: (a), (b)*



(c)



(d)

*(c) Neuron-1's point of view, (d) Neuron-2's point of view*

Every figure includes two lines  $f_0$  and either  $f_Q$  or  $f_R$ , which are all payoff functions. Line  $f_0$  is fixed and always remains unaltered during the learning process. In addition,  $f_0$  represents the payoff function for Class 1 and so, per definition, the resting state for Neuron-1. The second line  $f_Q$  is determined by the angle  $Q$ , where  $0 \leq Q \leq 90$  degree. This line represents the payoff function for Class 2 (i.e., the firing state for Neuron-1). The angle  $Q$  is derived by the uniform mapping function  $m : q = [0, 1] \rightarrow Q = [0^\circ, 90^\circ]$

The learning algorithm will find out in the training phase that this point does not separate the two classes correctly and take appropriate action. In this case, the algorithm will increase the angle  $Q$ , which moves the intersection point further to the left. There may be several such steps until the algorithm arrives at point  $P$  in Figure 13(b), which is a solution to the problem.

Any unknown object  $x_l$  to the left of point  $P$  produces two intersections, one at  $f_Q$  and one at  $f_0$ . However, any of these points yields  $f_0(x_l) > f_Q(x_l)$ . That is, the payoff for  $f_0(x_l)$  (rest) is always larger than the payoff for  $f_Q(x_l)$  (fire). Therefore, Neuron-1 chooses to stay at rest for any such value. For similar reasons, for any object  $x_r$  to the right of  $P$ , Neuron-1 chooses to fire, because for any such value, the payoff  $f_0(x_r) < f_Q(x_r)$ .

## Algorithm Game Theory Neural Learning

Start with a randomly chosen angle  $Q_0$ ;

Let  $k = 1$ ;

**While** there exist misclassified objects by  $Q_{k-1}$  **do**

Let  $o_j$  be a misclassified object;

Update the angle to  $Q_k = Q_{k-1} \pm \eta$ ;

Increment  $k$ ;

**end-While**;

$$g(x) = \begin{cases} \text{Fire} & \text{if } x > x_p, \\ \text{Rest} & \text{otherwise} \end{cases}$$

where  $x_p$  is the  $x$  coordinate of intersection point  $P'$  and in general, the separation point determined by the learning algorithm.

# STOCHASTIC GAMES

## Definition

A **stochastic game** is a tuple  $(Q, N, A, P, R)$ , where

- $Q$  is a finite set of states,
- $N$  is a finite set of  $n$  players,
- $A = A_1 \times \cdots \times A_n$ , where  $A_i$  is a finite set of actions available to player  $i$ ,
- $P : Q \times A \times Q \mapsto [0, 1]$  is the transition probability function;  $P(q, a, \hat{q})$  is the probability of transitioning from state  $q$  to state  $\hat{q}$  after joint action  $a$ , and
- $R = r_1, \dots, r_n$ , where  $r_i : Q \times A \mapsto \mathbb{R}$  is a real-valued payoff function for player  $i$ .

**Behavioral strategy:**  $s_i(h_t; a_{ij})$  returns the probability of playing action  $a_{ij}$  for history  $h_t$ .

**Markov strategy:**  $s_i$  is a behavioral strategy in which  $s_i(h_t; a_{ij}) = s_i(h'_t; a_{ij})$  if  $q_t = q'_t$ , where  $q_t$  and  $q'_t$  are the final states of  $h_t$  and  $h'_t$ , respectively.

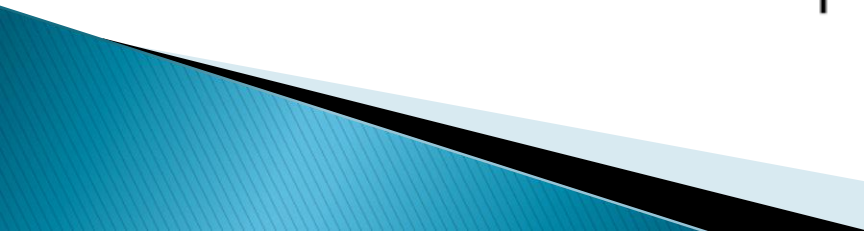
**Markov perfect equilibrium:**

- A strategy profile consisting of only Markov strategies that is a Nash equilibrium regardless of the starting state
- Analogous to subgame-perfect equilibrium

➤ **Every n-player, general sum, discounted reward stochastic game has a Markov perfect equilibrium.**

➤ **For every 2-player, general sum, average reward, irreducible stochastic game has a Nash equilibrium.**

# R-Max Algorithm

- Maintain an internal model of the stochastic game
  - Calculate an optimal policy according to model and carry it out
  - Update model based on observations
  - Calculate a new optimal policy and repeat
- 



# R-Max Algorithm Input

- Input
  - $N$ : number of games
  - $k$ : number of actions in each game
  - $\epsilon$ : the error bound
  - $\delta$ : the probability of failure
  - $R_{\max}$ : the maximum reward value
  - $T$ : the  $\epsilon$ -return mixing time of an optimal policy

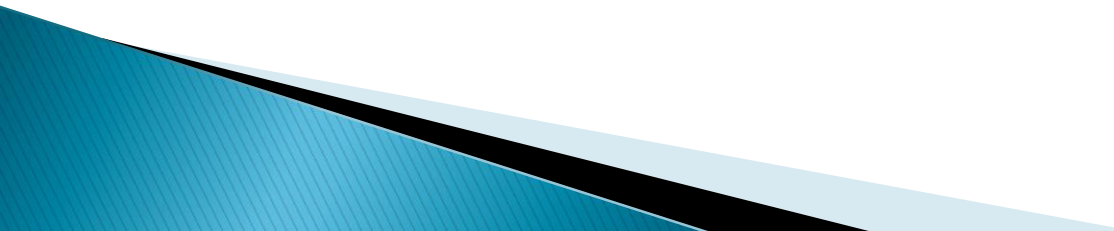
# Initialization

- Initializing the internal model
  - Create states  $\{G_1 \dots G_n\}$  to represent the stages in the stochastic game
  - Create a fictitious game  $G_0$
  - Initialize all rewards to  $(R_{\max}, 0)$
  - Set all transfer functions to point to  $G_0$
  - Associate a boolean known/unknown variable with each entry in each game, initialized to unknown
  - Associate a list of states reached with each entry, which is initially empty

# Iteration

- Repeat
  - Compute an optimal policy for  $T$  steps based on the current internal model
  - Execute that policy for  $T$  steps
  - After each step:
    - If an entry was visited for the first time, update the rewards based on observations
    - Update the list of states reached from that entry
    - If the list of states reached now contains  $c+1$  elements
      - mark that entry as known
      - update the transition function
      - compute a new policy

# CONCLUSION

- ▶ Game Theory has considered in the past CS-like representations (e.g. when players are modeled as automata), and work in AI has considered the use of game-theoretic mechanisms.
  - ▶ The connections between the AI and game theory as consists of three parts:
    1. Re-visiting economic and game-theoretic approaches, in view of their use in computational settings.
    2. Deal with computational issues in the context of game-theoretic approaches.
    3. Integrate game-theoretic approaches and CS approaches in order to yield new theories for non-cooperative multi-agent systems
- 

# REFERENCES

- ❑ **Application of Game Theory to Neuronal Networks**  
*Alfons Schuster and Yoko Yamaguchi, 2009*
- ❑ **Game Theory and Artificial Intelligence**  
*Moshe Tennenholtz, 2002*
- ❑ **A Primer in Game Theory**  
*Robert Gibbons (Published in 1992)*
- ❑ **Stochastic Games**  
*L. S. Shapley, 1953*
- ❑ **R-Max – A General Polynomial Time Algorithm for Near Optimal Reinforcement Learning**  
*Ronen I. Brafman and Moshe Tennenholtz, 2002*

# THANK YOU!

