Deep Learning for Solving Large Scale Complex Games

Bo AN
boan@ntu.edu.sg
School of Computer Science and Engineering
Nanyang Technological University

June 23, 2022
Many AI Problems are Games

- AI: study and construction of *rational* agents [Russell & Norvig, 2003]

- Many real world problems are games and game theory is needed (1940s-)
  - GT for AI: success in computer poker, security, auction…
  - The rest of the talk:
    - Solving games with algorithmic game theory
    - New trend: Deep (reinforcement) learning for solving games

- **Building a single agent** (1950s-70s)
- **Multi-agent systems (cooperative)** (1980s-)
- **Multi-agent systems (competitive)** (1995-)
Games and Computation

- Players, strategies, payoffs

<table>
<thead>
<tr>
<th></th>
<th>Player A</th>
<th>Player B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 0</td>
<td>1, -1</td>
<td>-1, 1</td>
</tr>
<tr>
<td>-1, 1</td>
<td>0, 0</td>
<td>1, -1</td>
</tr>
<tr>
<td>1, -1</td>
<td>-1, 1</td>
<td>0, 0</td>
</tr>
</tbody>
</table>

Nash Equilibrium: no agent has incentive to unilaterally deviate

- In any (finite) game, at least one Nash equilibrium (possibly mixed) exists [Nash, 50]
- In 2-player zero-sum games, a profile is an NE iff both players play minimax strategies
- Computing one (any) Nash equilibrium is PPAD-complete (even in 2-player games) [Daskalakis, Goldberg, Papadimitriou 2006; Chen, Deng 2006]
- All known algorithms require exponential time (in the worst case)
  - Lemke-Howson, support enumeration

Mechanism design
Libratus for Computer Poker

- Abstraction (offline)
  - Action abstraction
  - Card abstraction
  - Game size from $10^{161}$ to $10^{12}$

- Equilibrium Finding (offline)
  - CFR
  - CFR$^+$
  - Monte Carlo CFR

- Decomposition and Subgame Refinement (online)
  - endgame solving
  - subgame re-solving
  - max-margin subgame refinement

- Deep learning: Alberta’s DeepStack, DeepMind
Playing Games with Machines [EC’20]

- Strategies differ in their implementation complexity
- If we knew how to model complexity → Machines
  - We can focus on relevant (easy) strategies

**Corollary 4.4.** Let $\mathcal{L}$ be a size-parametric class of perfect-recall EFGs with 2 players and $M_f^S(n)$ be a small class of machine strategies of the follower in $\mathcal{L}(n)$. Then the problem of finding a strategy profile $\gamma^{SSE} = (\gamma^R, M_f)$ describing an SSE in a restriction of $\mathcal{L}(n)$ induced by $M_f^S(n)$, i.e., $M_f \in M_f^S(n)$, is polynomial.

- Lower computational & implementation

![Diagram](image-url)
Correlated equilibrium = coordination
- Signaller sends signals to players
- Assumption: everyone is rational

Real-world players are subrational!

Our contributions:
- Incorporating quantal-response behavior:
  \[ u_i(\delta_{-i}, a_i^k) \leq u_i(\delta_{-i}, a_i^l) \Rightarrow QR_i^k(\delta_{-i}) \leq QR_i^l(\delta_{-i}) \quad \forall \delta_{-i} \in \Delta_{-i}, a_i^k, a_i^l \in A_i, \]
- Analyzing quantal correlation: relations to other equilibria, topology, complexity

Theorem 1. Let \( G = (N, A, S, u) \) be a signaling game. Then (i) every quantal response equilibrium in \( G \) is a quantal correlated equilibrium with trivial signaling structure; (ii) the limit of quantal correlated equilibria in \( G \) is a correlated equilibrium in \( G \) as quantal responses approach the best response; (iii) the signaling correspondence \( \lambda \rightarrow QCE(\lambda) \) is upper hemicontinuous; and (iv) computing a quantal correlated equilibrium in \( G \) is PPAD-hard.

Formulating computational methods for
- tracing quantal correlated equilibrium using a homotopic system
- gradiently optimizing the signal structure
Global challenges for security

- Boston Marathon bombings
- French oil tanker hit by a boat
- Cyber physical attacks

Security resource allocation

- Limited security resources
- Adversary monitors defenses, exploits patterns

We pioneered the first set of applications of game theory for security resource scheduling (2007-)

- 60+ papers at premier conferences/journals, 2 best paper awards
- INFORMS Daniel H. Wagner Prize for Excellence in Operations Research Practice (2012), etc
- Operational Excellence Award from US Coast Guard (2012), etc
- United States congressional hearing (4 times)
Combining techniques from AI, Game Theory, Operations Research ...

Marry theory with practice

Approaches can be applied to other domains With proper tuning & extension

- Incremental strategy generation
- Construct (multiple) equivalent games
- Exploit compact representation
- Abstraction
- Tradeoff between optimality and efficiency
- Approximation
Converging to Team-Maxmin Equilibria

Equilibria in Multiplayer Games
- Hard to compute: PPAD-Complete
- Hard to select: NEs are not unique
- Few results:
  - Special structure: congestion games
  - No theoretical guarantee: Pluribus [Brown and Sandholm 2019]

Team-Maxmin Equilibria [von Stengel and Koller 1997]
- A team of players independently plays against an adversary
- Unique in general
- FNP-hard to compute a team-maxmin equilibrium
  - Formulated as a non-convex program
  - Solved by a global optimization solver

Converging to Team-Maxmin Equilibria
- Existing ISG for multiplayer games
  - Converge to an NE but many not to a TME
  - Difficult to extend the current ISG to converge to a TME
- ISGT: the first ISG guaranteeing to converging to a TME
  - Conditions in ISGT cannot be further relaxed
- CISGT: further improve the scalability
  - Initialize the strategy space by computing an equilibrium that is easier to be computed

Unsatisfactory scalability!
When do We Need D(R)L for Complex Games?

- When GT is better than ML
  - Requires no data
  - No assumption about players’ behavior
  - Not exploitable
  - Theoretical guarantee

- ML might be more appropriate when
  - Large scale: Millions of (even continuous) pure strategies
  - Uncertainty
  - Cannot be well modelled
  - Non-convex and cannot be approximated
  - No domain structures can be exploited

- D(R)L for games is receiving increasing attention
  - Solving games, e.g., DeepStack
  - Mechanism design

- Does not mean D(R)L can always work!

- Rest of the talk: quick overview of our two works since 2021
  - CFR-MIX: based on counterfactual regret minimization
  - NSG-NFSP: based on fictitious play
  - NSGZero: based on neural Monte Carlo tree search
Counterfactual Regret Minimization

**CFR** [Zinkevich et al. 2008]

- A popular algorithm to solve imperfect-information extensive-form games
- In every iteration, it traverses the whole game tree and computes counterfactual regret value for every information set

\[ v_i(l, \sigma) = \sum_{z \in Z_l} \pi_{-i}(z[l]) \pi_i(z[l], z) u_i(z) \]

\[ r^t(l, a) = v_i(l, \sigma_t^{l-a}) - v_i(l, \sigma_t^l) \]

- Compute the strategy of next iteration using regret matching based on the sum of counterfactual regret values

\[ R^T(l, a) = \sum_{t=1}^T r^t(l, a) \]

\[ \sigma_i^{T+1}(l)(a) = \begin{cases} \frac{R_i^{T+1}(l, a)}{\sum_{a \in A(l)} R_i^{T+1}(l, a)} & \text{if } \sum_{a \in A(l)} R_i^{T+1}(l, a) > 0 \\ \frac{1}{|A(l)|} & \text{otherwise} \end{cases} \]

- The average strategy over all iterations converges to Nash equilibrium in two-player zero-sum games

**CFR variants**

- **Sampling-based CFR**
  - Traversing the whole game tree is very time-consuming. Sampling-based CFR only traverses a subset of the game tree.
    - External Sampling CFR, Outcome Sampling CFR [Lanctot et al., 2009], Probe sampling [Gibson et al., 2012]

- **Deep-based CFR**
  - Tabular representation needs huge memory. Deep-based CFR uses neural networks to represent the regret value and strategy.
    - Deep CFR [Brown et al., 2019], Double Neural CFR [Li et al., 2019]
Team-Adversary Games

- A team of players cooperatively plays against one adversary
- Hard to solve due to the large combinatorial action space
  - The exponentially growing joint action space of the team
- Ineffectiveness of existing methods
  - Out of memory for tabular-form methods
  - Ineffective to train the strategy network over the large action space for DNNs

CFR-MIX

- Use the individual strategy representation to reduce the strategy space
  - $f_T = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ where $\sigma_i$ is the strategy for team player $i$
- Provide the consistency relationship to maintain the the NE unchanged
  - $\sigma_T(l, a) = \sigma_1(l, a_1)\sigma_2(l, a_2) \ldots \sigma_n(l, a_n)$ where $\sigma_T$ is the joint strategy for the team
- Propose a Product-form decomposition method to maintain the consistency
  - $\forall a, R_{tot}(l, a) = \prod_{i=1}^n R_i(l, a_i)$
- Implement the decomposition method using a mixing layer
- Theorem: With the probability $1 - \rho$, the total regret of player $i$ at time $T$ is bounded by
  $$R_i^T \leq \left(1 + \frac{\sqrt{2}}{\sqrt{\rho K}}\right) \Delta \sqrt{|l|\sqrt{|A|\sqrt{T}}} + 4T|l|\sqrt{|A|\Delta \epsilon_L}$$
Fictitious play is a game-theoretic algorithm for learning NE

- Players repeatedly play a game. At each iteration, each player plays with its opponent’s past average policy and best responds against it.

Full-Width Extensive-Form Fictitious Play (XFP) [Heinrich et al. 2015] extends fictitious play from normal form to extensive form.

Neural Fictitious Self-Play [Heinrich et al. 2016]

- A sampling and machine learning-based adaption of XFP
- Utilizing deep neural network function approximation
- Each agent consists of two neural networks, i.e., the average policy network and the best response (BR) policy network
- The average policy network approximates an agent’s past average policy by supervised learning
- The BR policy network best responds to other agents’ average policy by reinforcement learning

Related work

- OptGradFP [Kamra et al. 2018] firstly introduces fictitious play to continuous action spaces
- NFSP cannot solve games like NSGs whose action space is extremely large
Securing networked infrastructure with limited security resources

- Vanilla NFSP cannot solve NSGs because
  - The defender has combinational action space, and its legal action spaces change with states
  - It is impossible for the output of deep neural networks to cover the huge action space
  - The output of deep neural networks have inconsistent semantics due to the changing legal action spaces
- Sparse reward which brings difficulties in exploration
- How to represent the road network efficiently when graphs are extremely large

NSG-NFSP: approximating an NE defender policy in NSGs

- Framework: Neural Fictitious Self-Play (NFSP), which is guaranteed to converge to an NE
- Learning state and action representations when approximating BR and AVG policies
- Enabling NFSP with high-level actions for efficient exploration
- Learning efficient graph node embeddings via node2vec

Solving Large-Scale Extensive-Form Network Security Games via Neural Fictitious Self-Play [IJCAI'21]
NSGZero: Efficiently Solving Large-scale Network Security Games via Neural Monte Carlo Tree Search [AAAI’22]

- Improving **data efficiency** by performing planning with neural MCTS
  - Modeling the dynamics of NSGs
  - Predicting future state values when planning
  - Leveraging prior knowledge to do exploration

- Improving scalability by enabling neural MCTS with **decentralized execution**
  - Agents record simulation statistics separately

---

**Value network:** predict the state value of the next state from backup phase

**Prior network:** predict the prior policy of resources

**Dynamics network:** predict the next state

---

**Algorithm 2: NGSZero-EXECUTION**

Input: The current state \( s_t \), the search tree \( \Psi \).

1. \( \Psi . \text{clear}() \) \( \backslash \) clear statistics stored in the search tree;
2. for \( N \) simulations do
3. \( \Psi . \text{search}(s_t) \) \( \backslash \) perform lookahead search;
4. end
5. for resources \( i = 0, \ldots, m - 1 \) do
6. \( \pi_i(s_t, a_t) \propto O(s_t, a_t)^{1/T}; \ a_{t+1} \sim \pi(s_t) \);
7. end

Output: Joint action \( a_t = (a_{0,t}, \ldots, a_{m-1,t}) \).
Roadmap and Next Steps on DL for Solving Games

- NFSP (2016)
- Double CFR (2020)
- CFR-MIX (2021)
- α-rank PSRO (2020)
- Pipeline PSRO (2021)
- Policy Space Response Oracle (PSRO) (2017)
- Μ-PSRO (2020)

Learning Methods:
- Offline learning
- Model-based Learning
- Representation learning
- Neural & Simplex Population Learning
- Transfer/Meta/Multitask learning
- Players of Games
Many DL Scenarios can be viewed as Games
- Generative Adversarial Networks (GAN)
- Adversarial training
- Adversarial reinforcement Learning
- Even, transfer/meta/multitask learning, and self-supervised learning

GT provides theoretical and algorithmic tools
- Theoretical: regret analysis, convergence rate
- Algorithmic: CFR, NFSP, PSRO

D(R)L provides representations and training methods
- RL for general-sum games?
- The limit of representation learning [ICLR’22, ICML’22]
- Big models, e.g., transformer

There is a long way to go…