Deep Learning in Tree-Based Game Solving 4

Stephen McAleer
Outline of the next few lectures

- Deep learning in tree-based game solving 1
  - Deep learning recap
  - NFSP
  - Deep CFR
  - Policy gradient methods
- Deep learning in tree-based game solving 2
  - MCCFR
  - DREAM
  - ESCHER
  - NeuRD
- Deep learning in tree-based game solving 3
  - DeepNash for expert-level Stratego
- Deep learning in tree-based game solving 4
  - AlphaStar and OpenAI 5 for SOTA in video games
  - Double Oracle brief intro
- SOTA in double oracle algorithms
  - PSRO
  - XDO
  - SP-PSRO
A Taxonomy of Game-Theoretic RL

- Counterfactual Regret Minimization (Zinkevich et al. 2007)
  - CFR: Zinkevich et al. 2007
  - Deep CFR: Brown et al. 2019
  - DREAM: Steinberger et al. 2020
  - ESCHER: McAleer et al. 2022

- Policy Gradients
  - Regret Policy Gradient (Srinivasan et al. 2018)
  - OpenAI Five (OpenAI 2019)
  - Neural Replicator Dynamics (Hennes, Morrill, and Omidshafiei et al. 2020)
  - Actor Critic Hedge (Fu et al. 2022)
  - DeepNash for expert-level Stratego (Perolat, de Vylder, and Tuyls et al. 2022)
  - Magnetic Mirror Descent (Sokota et al. 2022)

- PSRO (McMahan et. al. 2003, Lanctot et al. 2017)
  - AlphaStar for expert-level Starcraft (Vinyals et al. 2019)
  - Pipeline PSRO (McAleer and Lanier et al. 2020)
  - α-PSRO (Muller et al. 2020)
  - XDO (McAleer et al. 2021)
  - Joint-PSRO (Marris et al. 2021)
  - Anytime PSRO (McAleer et al. 2022)
  - Self-Play PSRO (McAleer et al. 2022)

- Neural Fictitious Self Play (Heinrich and Silver 2016)
A Taxonomy of Game-Theoretic RL

- **Counterfactual Regret Minimization (Zinkevich et al. 2007)**
  - CFR: Zinkevich et al. 2007
  - **MC-CFR: Lanctot et al. 2009**
  - Deep CFR: Brown et al. 2019
  - DREAM: Steinberger et al. 2020
  - **ESCHER: McAleer et al. 2022**

- **Policy Gradients**
  - Regret Policy Gradient (Srinivasan et al. 2018)
  - OpenAI Five (OpenAI 2019)
  - **Neural Replicator Dynamics (Hennes, Morrill, and Omidshafiei et al. 2020)**
  - Actor Critic Hedge (Fu et al. 2022)
  - DeepNash for expert-level Stratego (Perolat, de Vylder, and Tuyls et al. 2022)
  - Magnetic Mirror Descent (Sokota et al. 2022)

- **PSRO (McMahan et al. 2003, Lanctot et al. 2017)**
  - AlphaStar for expert-level Starcraft (Vinyals et al. 2019)
  - Pipeline PSRO (McAuleer and Lanier et al. 2020)
  - α-PSRO (Muller et al. 2020)
  - XDO (McAuleer et al. 2021)
  - Joint-PSRO (Marris et al. 2021)
  - Anytime PSRO (McAuleer et al. 2022)
  - Self-Play PSRO (McAuleer et al. 2022)

- **Neural Fictitious Self Play (Heinrich and Silver 2016)**
A Taxonomy of Game-Theoretic RL

- Counterfactual Regret Minimization (Zinkevich et al. 2007)
  - CFR: Zinkevich et al. 2007
  - Deep CFR: Brown et al. 2019
  - DREAM: Steinberger et al. 2020
  - ESCHER: McAleer et al. 2022

- Policy Gradients
  - Regret Policy Gradient (Srinivasan et al. 2018)
  - OpenAI Five (OpenAI 2019)
  - Neural Replicator Dynamics (Hennes, Morrill, and Omidshafiei et al. 2020)
  - Actor Critic Hedge (Fu et al. 2022)
  - DeepNash for expert-level Stratego (Perolat, de Vylder, and Tuyls et al. 2022)
    - From Poincaré Recurrence to Convergence in Imperfect Information Games: Finding Equilibrium via Regularization (Perolat et al. 2021)
  - Magnetic Mirror Descent (Sokota et al. 2022)

- PSRO (McMahan et al. 2003, Lanctot et al. 2017)
  - AlphaStar for expert-level Starcraft (Vinyals et al. 2019)
  - Pipeline PSRO (McAleer and Lanier et al. 2020)
  - α-PSRO (Muller et al. 2020)
  - XDO (McAleer et al. 2021)
  - Joint-PSRO (Marris et al. 2021)
  - Anytime PSRO (McAleer et al. 2022)
  - Self-Play PSRO (McAleer et al. 2022)

- Neural Fictitious Self Play (Heinrich and Silver 2016)
A Taxonomy of Game-Theoretic RL

- Counterfactual Regret Minimization (Zinkevich et al. 2007)
  - CFR: Zinkevich et al. 2007
  - Deep CFR: Brown et al. 2019
  - DREAM: Steinberger et al. 2020
  - ESCHER: McAleer et al. 2022

- Policy Gradients
  - Regret Policy Gradient (Srinivasan et al. 2018)
  - OpenAI Five (OpenAI 2019)
  - Neural Replicator Dynamics (Hennes, Morrill, and Omidshafiei et al. 2020)
  - Actor Critic Hedge (Fu et al. 2022)
  - DeepNash for expert-level Stratego (Perolat, de Vylder, and Tuyls et al. 2022)
  - Magnetic Mirror Descent (Sokota et al. 2022)

- PSRO (McMahan et. al. 2003, Lanctot et al. 2017)
  - AlphaStar for expert-level Starcraft (Vinyals et al. 2019)
  - Pipeline PSRO (McAleer and Lanier et al. 2020)
  - α-PSRO (Muller et al. 2020)
  - XDO (McAleer et al. 2021)
  - Joint-PSRO (Marris et al. 2021)
  - Anytime PSRO (McAleer et al. 2022)
  - Self-Play PSRO (McAleer et al. 2022)

- Neural Fictitious Self Play (Heinrich and Silver 2016)
A Taxonomy of Game-Theoretic RL

- **Counterfactual Regret Minimization (Zinkevich et al. 2007)**
  - CFR: Zinkevich et al. 2007
  - Deep CFR: Brown et al. 2019
  - DREAM: Steinberger et al. 2020
  - ESCHER: McAleer et al. 2022

- **Policy Gradients**
  - Regret Policy Gradient (Srinivasan et al. 2018)
  - OpenAI Five (OpenAI 2019)
  - Neural Replicator Dynamics (Hennes, Morrill, and Omidshafiei et al. 2020)
  - Actor Critic Hedge (Fu et al. 2022)
  - DeepNash for expert-level Stratego (Perolat, de Vylder, and Tyuils et al. 2022)
  - Magnetic Mirror Descent (Sokota et al. 2022)

- **PSRO (McMahan et. al. 2003, Lanctot et al. 2017)**
  - AlphaStar for expert-level Starcraft (Vinyals et al. 2019)
  - Pipeline PSRO (McAleer and Lanier et al. 2020)
  - α-PSRO (Muller et al. 2020)
  - XDO (McAleer et al. 2021)
  - Joint-PSRO (Marris et al. 2021)
  - Anytime PSRO (McAleer et al. 2022)
  - Self-Play PSRO (McAleer et al. 2022)

- **Neural Fictitious Self Play (Heinrich and Silver 2016)**
Self Play PPO

- Just have agents play each other in self play
- Also look at a version of fictitious play where they output the latest strategy
- For these simulated robotics environments, can get emergent behaviors
- [https://openai.com/research/competitive-self-play](https://openai.com/research/competitive-self-play)

More Self Play PPO

- In hide and seek game, agents can discover complex strategies with self play
- Hiders learn how to push boxes to protect themselves
- Seekers then learn counter-strategy of pushing ramp to jump over wall

Self Play PPO Exploitability

- Since self-play doesn’t find an approximate Nash equilibrium, it is exploitable
- Best responses don’t even have to do anything sophisticated

Figure 1: Illustrative snapshots of a victim (in blue) against normal and adversarial opponents (in red). The victim wins if it crosses the finish line; otherwise, the opponent wins. Despite never standing up, the adversarial opponent wins 86% of episodes, far above the normal opponent’s 47% win rate.
Dota 2

- Multiplayer Online Battle Arena (MOBA) game
- Features two teams, the Radiant and the Dire, each with five players.
- Objective: Destroy the opposing team's "Ancient" structure.
- Over 100 unique heroes to choose from, each with distinctive abilities.
- Robust competitive scene
  - Annual tournament has multi-million-dollar prize pools
- Around 500k - 1M concurrent players
Dota 2

- Map consists of three lanes (Top, Middle, Bottom) and a jungle area
- Players select heroes during the drafting phase
- Earn gold and experience by killing enemy heroes, creeps, and buildings
- Use gold to buy items that enhance heroes' abilities
- Constant strategy and coordination required to seize objectives like Roshan, towers, and barracks
- Ultimate goal: Breach the enemy base and destroy the Ancient
Dota 2 Strategy

- **Bluffing and Deception:**
  - "Smokes of Deceit": Items that make the team invisible to wards, allowing for surprise attacks or ganks.
  - Fake Backs: Pretending to retreat and then quickly re-engaging.
  - Baiting: Luring enemies into unfavorable positions by making them think they have an advantage.

- **Mixed Strategy Play:**
  - Constantly adapting between aggressive (ganking, pushing) and passive (farming, defensive) strategies based on in-game situations.
  - Changing lanes, rotating heroes to surprise the enemy.

- **Drafting Strategy:**
  - Counter-picking enemy heroes or picking synergistic team combinations.

- **Resource Management:**
  - Balancing between farming, pushing, and fighting.
  - Ensuring that key heroes get the necessary gold and experience.

- **Map Control:**
  - Securing objectives like Roshan, runes, and outposts.

- **Team Synergy:**
  - Coordinating team abilities for maximum impact during fights.
  - Communication is vital for executing plans and adapting to changes.
OpenAI Dota Timeline

- **2017**: OpenAI introduces initial Dota 2 AI.
  - Demonstrates 1v1 gameplay against world's top players at The International.
- **2018**: Evolution of Dota AI.
  - OpenAI Five competes in more complex 5v5 matches.
  - Exhibits cooperative strategies and dynamic reactions.
- **April 2019**: OpenAI Five Finals.
  - Competes with and defeats world champion team OG.
- **June 2019**: OpenAI Five released to the public.
  - Made available for players worldwide to challenge.
  - Found to be exploitable
Model Architecture

- (Nearly) Identical observations for each team member: 2-player game not team game

Figure 1: Simplified OpenAI Five Model Architecture: The complex multi-array observation space is processed into a single vector, which is then passed through a 4096-unit LSTM. The LSTM state is projected to obtain the policy outputs (actions and value function). Each of the five heroes on the team is controlled by a replica of this network with nearly identical inputs, each with its own hidden state. The networks take different actions due to a part of the observation processing’s output indicating which of the five heroes is being controlled. The LSTM composes 84% of the model’s total parameter count. See Figure 17 and Figure 18 in Appendix H for a detailed breakdown of our model architecture.
Sampling Strategy

- 80% against latest policy
- 20% against past policies
- When sampling past policies, policies are given values
  - Sampled according to softmax of these values, values updated according to performance vs current policy

\[ q_i \leftarrow q_i - \frac{\eta}{Np_i} \]

If another agent loses to us, we down-weight that agent
Performance Over Time

![Graph showing performance over time with different lines representing benchmark, test team A, test team B, hand-scripted, and random models. The y-axis represents TrueSkill, and the x-axis represents compute (PFLOPs/s-days). The graph compares these models to the OG (world champions) and includes data points for OpenAI Five and pro matches won, as well as calibration matches.](image-url)
Starcraft II

- Real-time strategy game
- Three distinct races: Terran, Zerg, Protoss
- Objective: Gather resources, build army, conquer opponents
- Became standard for RTS competitions globally
- Popular in major tournaments like the World Championship Series (WCS)
Starcraft II Gameplay

● Economy Management:
  a. Gather two main resources: Minerals & Vespene Gas
  b. Balance between resource gathering, army production, and tech upgrades

● Scouting:
  a. Essential to anticipate opponent's moves
  b. Use early units or specialized scout units to gain intelligence

● Army Composition & Micromanagement:
  a. Different units for different strategies
  b. Units have strengths and weaknesses against certain enemy types
  c. Micromanage units during battle for optimal performance

● Positioning & Map Control:
  a. Strategic placement of buildings and units
  b. Secure key points on the map to control resources and movement pathways
  c. Prevent opponent's expansion while looking for opportunities to expand

● Tech Tree Progression:
  a. Upgrade paths unlock new abilities and units
  b. Determine the balance between investing in tech versus increasing army size

● Adaptability:
  a. No single strategy ensures victory
  b. Counter opponent's tactics and stay unpredictable
DeepMind Starcraft Timeline

- **2016**: Partnership between DeepMind and Blizzard announced
- **2017**: Introduction of the StarCraft II Learning Environment (SC2LE)
- **Early 2019**: Introduction of "AlphaStar" AI reaching Grandmaster level
- **Mid 2019**: AlphaStar competes on public 1v1 European servers anonymously
- **Late 2019**: Research paper on AlphaStar's progression published in Nature
Network Architecture
Method

- Similar to PSRO
- Prioritized Fictitious Self Play (PFSP): sample proportionate to how well opponents beat you
- Main agents
  - Trained against 35% SP, 50% PFSP, 15% exploiters
- League exploiters
  - Trained against PFSP
- Main exploiters
  - Play against main agents
- Output: meta-Nash equilibrium of the league
Method

- To compute meta-NE, have each agent play each other, compute the score
- Then, create a normal form game with the payoffs
- Finally, find a NE in this normal form game
Ablations

(a) League composition
+ League exploiters: 1,824
+ Main exploiters: 1,693
Main agents: 1,540

(b) League composition
Relative population performance (%)
+ League exploiters: 62%
+ Main exploiters: 35%
Main agents: 6%

(c) Multi-agent learning
pFSP + SP: 1,540
SP: 1,519
pFSP: 1,273
FSP: 1,143

(d) Multi-agent learning
Min win rate vs past (%)
pFSP + SP: 71%
SP: 46%
pFSP: 70%
FSP: 69%