Stochastic Applications of Reinforcement Learning: An Empirical Study

Charles Rule[1], William Fallin[2], Dr. Rodion Podorozhny[2]

[1] Department of Computer Science, Rutgers University, NJ

[2] Department of Computer Science, Texas State University, TX

Introduction

Reinforcement learning is a type of machine learning rooted in behavioral psychology. Evaluating algorithms' performance in real-life scenarios is essential to choosing the best algorithm.

Like many stochastic scenarios, "Guess Who" has a large state space of 8,388,608 possible states, so we used function approximation via neural networks to reach the optimal strategy in a practical amount of time.

Stochastic Games

"Guess Who" can be modeled as a simple stochastic game, as defined by Condon[1]:

- Finite set of states, actions and players
- Defined goal
- Probability function that determines movement between states
- Quantifiable rewards
- Directed acyclic graph shape

Procedure

We tested three reinforcement learning algorithms: deep-Q (QNN), actor-critic (AC) and asynchronous actor-critic (AAC). To combat performance loss, we applied the Adamax optimizer to AC (ACMax). The algorithms played against an agent that follows the optimal "Guess Who" strategy proposed by Nica[2]:

1. If a player is ahead, make a safe move such as binary search
2. If a player is behind, make a risky guess to catch up

```
If WON:
    IF TURNS IN GAME > 5:
        RETURN 1
    ELSE:
        RETURN 2
ELSE IF LOST:
    IF TURNS IN GAME < 3:
        RETURN -1
    ELSE:
        RETURN -2
ELSE:
    IF NUMBER OF TILES FLIPPED = 0:
        RETURN -1
    ELSE
        RETURN NUMBER OF TILES FLIPPED
```

Figure 1: Reward function

- Questions that flip over no tiles are penalized, and questions that flip a lot of characters are rewarded
- Games that are won quickly are favored over longer games

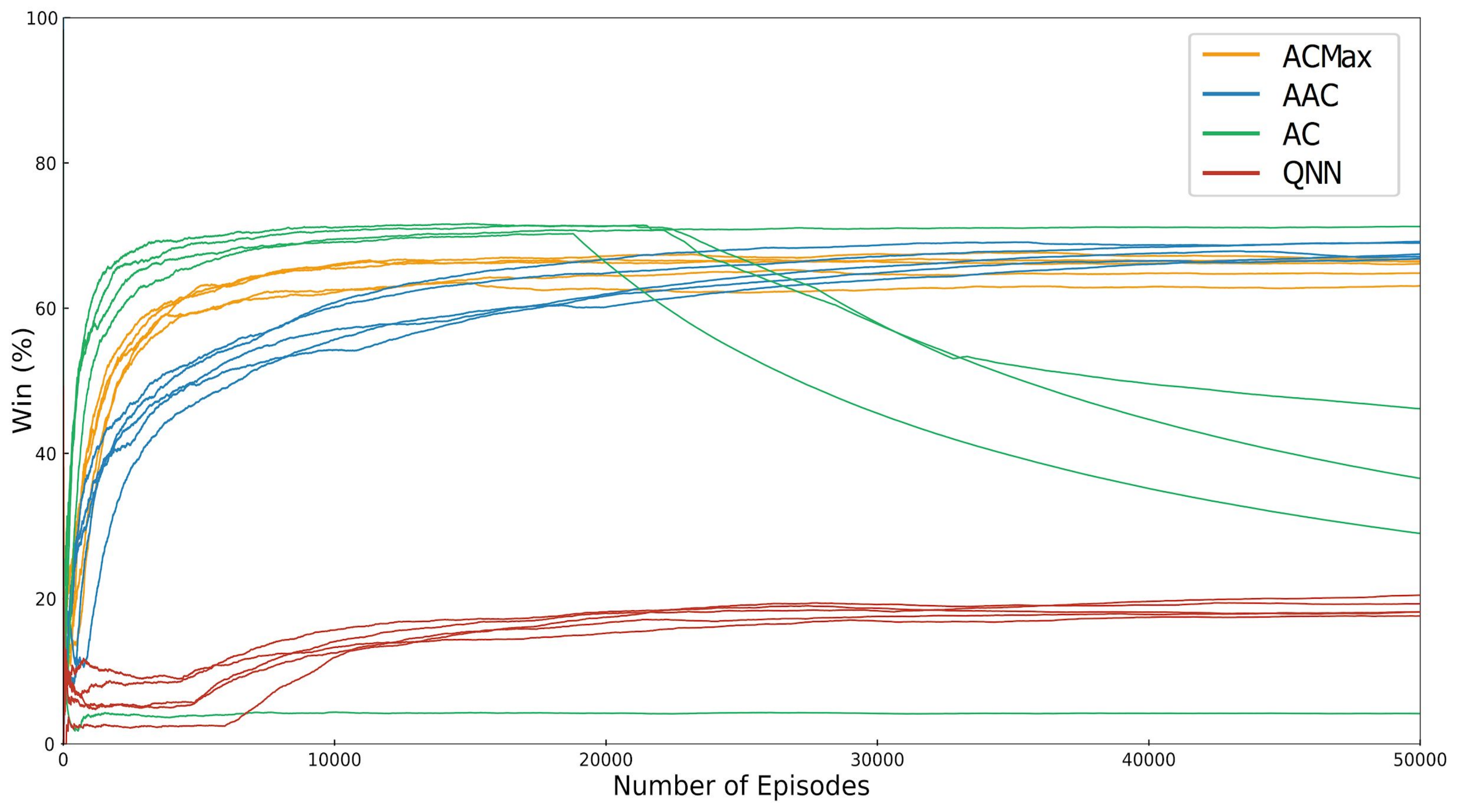

Figure 2: 50,000 games on pre-trained neural networks

Analysis

	QNN	AC	Optimal	ACMax	AAC
Mean	17.56%	44.36%	64.01%	64.26%	67.30%
σ	1.57	19.80	1.42	0.97	2.91
25th %	16.69	26.32	63.00	64.26	65.26
50th %	17.87	37.37	64.00	64.46	68.32
75th %	19.50	70.39	65.00	64.61	69.65

Figure 3: Win rates

AAC had the highest mean win rate; QNN had the lowest. ACMax had the lowest standard deviation, so it was the most stable. ACMax and AAC achieved win rates equal to or above the optimal agent.

We used the Mann–Whitney U test to confirm if the mean rank of each algorithm is different from the optimal agent's performance, and attained a p-value of <0.001.

References

1. Condon, A. (1992). The complexity of stochastic games. *Information and Computation* 96, 203–224.
2. M. Nica, "Optimal strategy in "Guess Who?": Beyond binary search," *Probability in the Engineering and Informational Sciences*, 2015.

Contributions

- Evaluated reinforcement learning algorithms that approach optimal play via approximation in zero-sum adversarial stochastic games with a large state space
- Developed an open source, reusable framework for evaluating "Guess Who" games in Python

Further Applications

- Cognitive radio
- Adversarial sports
- Resource allocation
- Bidding wars
- Non-cooperative board games

Conclusions

- The Actor-Critic models' independent weight updates are suitable for better long-term, dynamic play
- QNN is unsuitable for games with large state-action spaces, like this one
- Win rates support Nica's predicted win rate of ~62-66% for Player 1[2]

Avenues for future work:

- Player 2's perspective
- Human-agent trials
- Agents playing against each other

Acknowledgements

We thank the National Science Foundation (NSF) for funding this research under the Research Experiences for Undergraduates Program (CNS-1358939, CNS-1659807) and the infrastructure support provided by the NSF-CRI 1305302 award.