Deep Reinforcement Learning in Continuous Action Spaces: a Case Study in the Game of Simulated Curling

Kyowoon Lee*[1], Sol-A Kim*[1], Jaesik Choi[1] and Seong-Whan Lee[2]

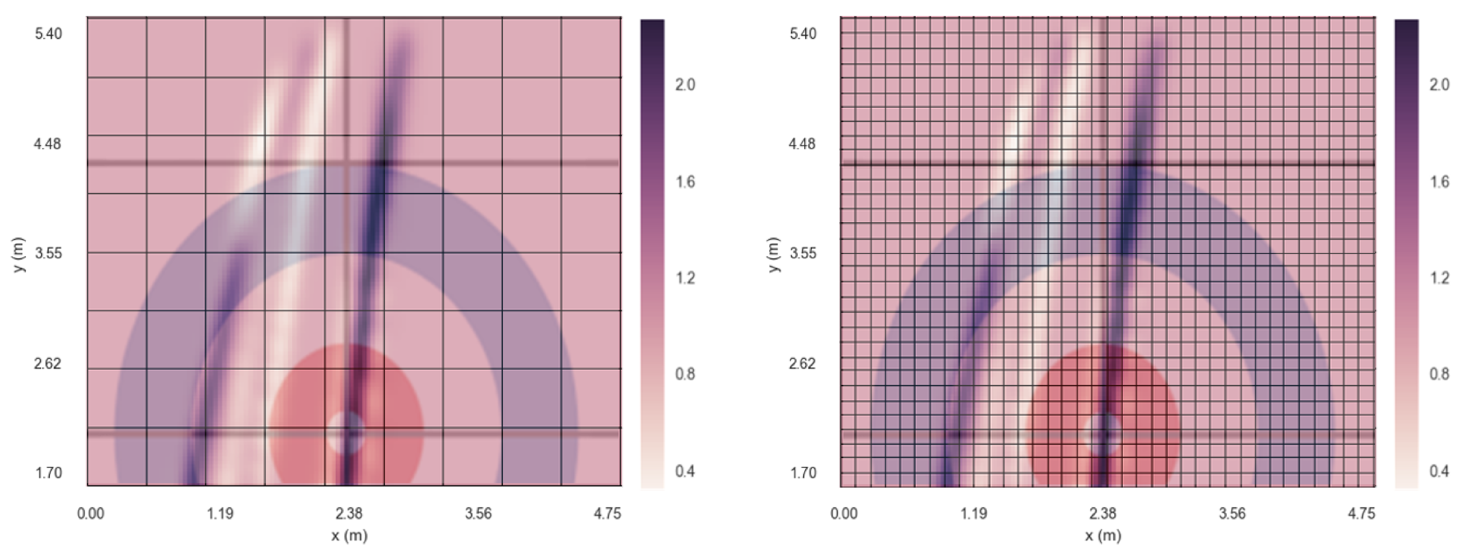
[1]Ulsan National Institute of Science and Technology, Korea. [2]Korea University, Korea.

Introduction - Deep Reinforcemnet Learning in Continuous Action Space

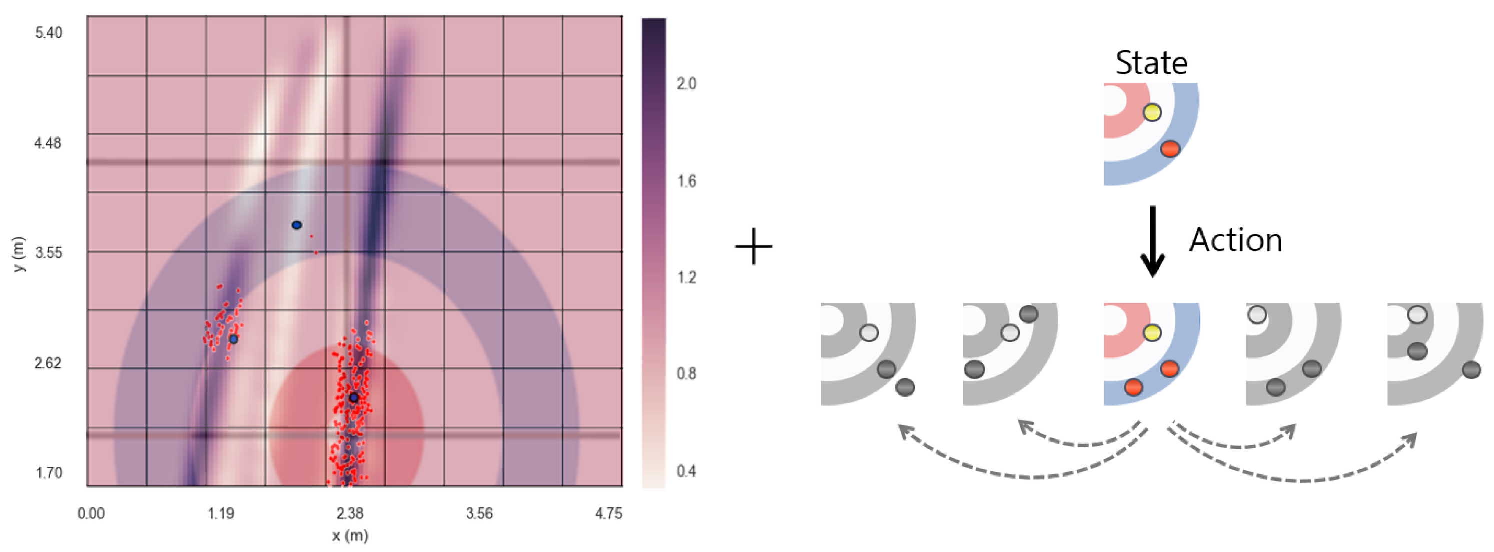
- **Goal**: Training an agent to learn a improved policy in the continuous space
- **Input**:
  - Current state form the environment
  - Reward from the environment
- **Output**: The best action given constraints
- **Case study in the game of simulated curling**:
  - Two dimensional continuous action space with two kinds of curl directions
  - Execution uncertainty modeled by asymmetric Gaussian noise.

Motivation

- Deep neural networks for the discrete actions are not suitable for devising strategies for games in which a very small change in an action can dramatically affect the outcome.
- **Challenges of learning in the discretized action**



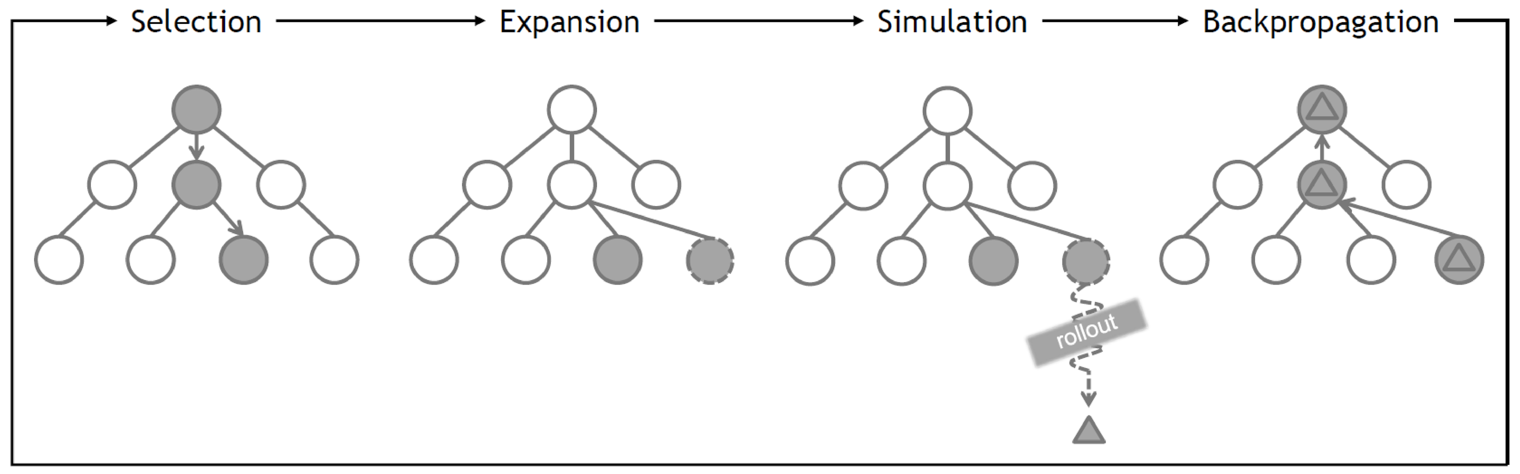
- Deterministic discretization has problems: (1) low resolution → strong bias in policy evaluation and improvement (2) high resolution → slow searching and learning speed and exponential growth in the number of actions to explore
- **Learning in the discretized action space with Kernel Meth-**



- Conducts local search with continuous action samples generated from a deep convolutional neural network (CNN).
- Generalizes the information between similar actions through kernel methods.

Monte Carlo Tree Search and Kernel Regression

- **Monte Carlo Tree Search** (MCTS) is a simulation-based search approach to planning in finite-horizon sequential decision-making settings.



- **Upper Confidence Bound applied to Trees (UCT)** is a commonly used MCTS algorithm using an Upper Confidence Bound (UCB) selection function.

$$\underset{a}{\operatorname{argmax}}\ \bar{v}_a + C\sqrt{\frac{\log \sum_b n_b}{n_a}}$$

- **Kernel Regression** is a non-parametric method which uses a kernel function as a weight for estimating the conditional expectation of a random variable.

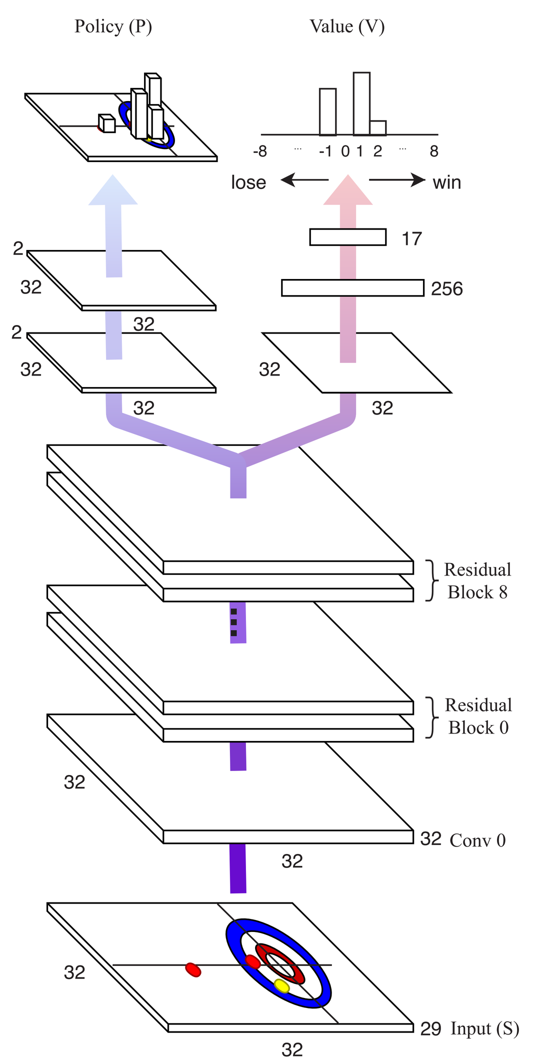
$$E[y|x] = \frac{\sum_{i=0}^{n} K(x, x_i) y_i}{\sum_{i=0}^{n} K(x, x_i)}$$

- The denominator of kernel regression is related to **Kernel Density Estimation** which is method for estimating the probability density function of random variable.

$$W(x) = \sum_{i=0}^{n} K(x, x_i)$$

Kernel Regression Deep Learning UCT

- The policy-value network



Algorithm 1 KR-DL-UCT

$\mathbf{p}_\theta \leftarrow$ the policy network
$\mathbf{v}_\theta \leftarrow$ the value network
$s_t \leftarrow$ the current state
$A_t \leftarrow$ a set of visited actions in s_t
expanded $\leftarrow$ false
if s_t is terminal **then**
  return Score(s_t), false
end if
$a_t \leftarrow arg\max_{a \in A_t} \mathbb{E}[\bar{v}_a | a] + C\sqrt{\frac{\log \sum_{b \in A_t} W(b)}{W(a)}}$ ▷Selection
if $\sqrt{\sum_{a \in A_t} n_a} < |A_t|$ **then**
  $s_{t+1} \leftarrow$ TakeAction(s_t, a_t)
  $\vec{reward}$, *expanded* $\leftarrow$ KR-DL-UCT(s_t)
end if
if not *expanded* **then**
  $a'_t \leftarrow arg\min_{K(a_t, a) > \gamma} W(a)$ ▷Expansion
  $A_t \leftarrow A_t \cup a'_t$
  $s_{t+1} \leftarrow$ TakeAction(s_t, a'_t)
  $A_{t+1} \leftarrow \cup_{i=1}^{k} \{a_{init}^{(i)}\}$ s.t. $a_{init}^{(i)} \sim \pi_{a|s_{t+1}}$ // Policy net
  $\vec{reward} \leftarrow \mathbf{v}_\theta(s_{t+1} | s_t, a'_t)$ // Value net ▷Simulation
end if
$\vec{v}_{a_t} \leftarrow \frac{1}{n_{a_t}+1}(n_{a_t}\vec{v}_{a_t} + \vec{reward})$ ▷Backpropagation
$n_{a_t} \leftarrow n_{a_t} + 1$
return $\vec{reward}$, true

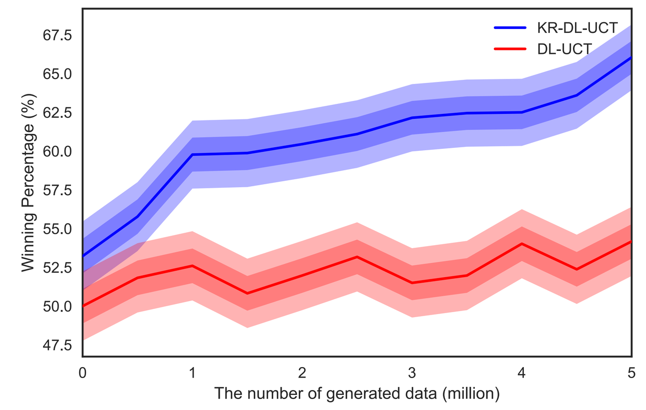
- Source codes will be available at `https://github.com/leekwoon/KR-DL-UCT`

Experimental Results

Datasets

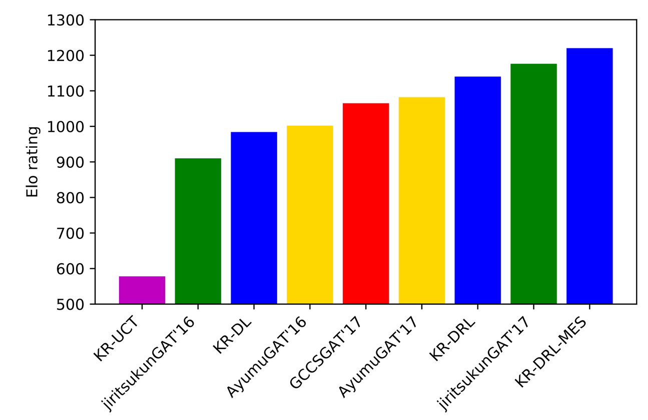
- **Supervised learning**: 0.4 million of the play data from the champion program (AyumuGAT'16) of Game AI Tournaments (GAT) digital curling championship in 2016.
- **Self-play Reinforcement Learning**: 5 million of the play data from self-play matches of KR-DL-UCT, executing 400 simulations per move.

Quantitative Results



# of data (million)	DL-UCT (1)	KR-DL-UCT (2)	(2)-(1)
0.0	50.0%	53.2%	3.2%
1.0	52.6%	60.0%	7.2%
2.0	51.9%	60.5%	8.5%
3.0	51.5%	62.2%	10.7%
4.0	54.0%	62.5%	8.5%
5.0	54.2%	66.0%	11.9%

- KR-DL-UCT (blue) outperforms (53.23%) DL-UCT (red) even without the self-play RL.
- After gathering 5 million shots from self-play, KR-DL-UCT wins 66.05% which is significantly higher than DL-UCT case.



Program	Winning Percentage
GCCSGAT'17	74.0 ± 6.22%
AyumuGAT'16	66.5 ± 6.69%
AyumuGAT'17	62.3 ± 6.87%
JiritsukunGAT'16	86.3 ± 4.88%
JiritsukunGAT'17	55.5 ± 7.04%

- KR-DL : KR-DL-UCT with supervised learning
- KR-DRL : KR-DL with self-play RL
- KR-DRL-MES: KR-DRL with winning percentage table (multi-end strategy)
- **Our program KR-DRL-MES won in the international digital curling competition, GAT-2018.**

Conclusion

- We provide a new framework which incorporates a neural network for learning strategy with a kernel based Monte Carlo tree search in the continuous action space.
- The developed method is applied to the game of Simulated Curling and achieves the state-of-the-art performance.

Acknowledgements

- This work was supported by Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-01779, A machine learning and statistical inference framework for explainable artificial intelligence, and No.2017-0-00521, AI curling robot which can establish game strategies and perform games).

References

- Yee, T., Lisý, V., and Bowling, M. Monte carlo tree search in continuous action spaces with execution uncertainty. In Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI, pp. 690697, 2016.
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. Mastering the game of go without human knowledge. Nature, pp. 354359, 2017.
- Ito, T. and Kitasei, Y. Proposal and implementation of digital curling. In Proceedings of the IEEE Conference on Computational Intelligence and Games, CIG, pp. 469-473, 2015.

*Equal contribution

Email: {leekwoon,sol-a,jaesik}@unist.ac.kr, sw.lee@korea.ac.kr