MCTS as Regularized Policy Optimization

Jean-Bastien Grill, Florent Altché, Yunhao Tang, Thomas Hubert, Michal Valko, Ioannis Antonoglou, Rémi Munos
Motivation

- Planning is thinking before acting.

- Imagining all future possible trajectories is too costly.

- Instead Monte Carlo Tree Search (MCTS) selects few of them.
Motivation

Huge success of MCTS in board games (UCT, AlphaZero), with extensions to richer environments (MuZero)

AlphaZero uses learned **policy** and **value networks** to **select** which path to search during planning.
MCTS in AlphaZero

AlphaZero action selection formula:

$$\arg \max_a q(a) + c\sqrt{N} \cdot \frac{\pi_\theta(a)}{1+n(a)}$$

- exploitation
- exploration

Handcrafted selection formula

Yet, generalised well to other games than Go.
MCTS in AlphaZero

**AlphaZero action selection formula:**

$$\arg \max_a q(a) + c \sqrt{N} \cdot \frac{\pi_\theta(a)}{1 + n(a)}$$

(exploitation  exploration)

**Handcrafted selection formula**

Yet, generalised well to other games than Go.

**AlphaZero**

approximates

**Regularized policy optimization**
AlphaZero search procedure

Simulation budget = 4

Action selection formula:

$$\arg \max_a q(a) + c\sqrt{N} \cdot \frac{\pi_\theta(a)}{n(a)}$$
AlphaZero search procedure

Simulation budget = 4

Action selection formula:

$$\arg \max_a q(a) + c\sqrt{N} \cdot \frac{\pi_\theta(a)}{n(a)}$$
**AlphaZero search procedure**

Simulation budget = 4

Action selection formula:

$$\arg \max_a q(a) + c\sqrt{N} \cdot \frac{\pi_\theta(a)}{n(a)}$$
AlphaZero search procedure

Simulation budget = 4

Action selection formula:

$$\arg \max_a q(a) + c\sqrt{N} \cdot \frac{\pi_\theta(a)}{n(a)}$$
AlphaZero search procedure

Action selection formula:

$$\arg \max_a q(a) + c\sqrt{N} \cdot \frac{\pi_\theta(a)}{n(a)}$$
AlphaZero search procedure

Simulation budget = 4

Action selection formula:

$$\arg \max_a q(a) + c \sqrt{N} \cdot \frac{\pi_\theta(a)}{n(a)}$$

Visit distribution:

$$\hat{\pi}(a) = \frac{n(a)}{\sum_{a'} n(a')}$$
AlphaZero search procedure

Simulation budget = 4

Action selection formula:
\[
\arg \max_a q(a) + c\sqrt{N} \cdot \frac{\pi_\theta(a)}{n(a)}
\]

Visit distribution:
\[
\hat{\pi}(a) = \frac{n(a)}{\sum_{a'} n(a')}
\]

Policy network is trained with visit distribution
\[
\pi_\theta \rightarrow \hat{\pi}
\]

The action is sampled according to the visit distribution
\[
\text{action} \sim \hat{\pi}
\]
Main result: AlphaZero as policy optimization

\[
\hat{\pi}(a) = \frac{n(a)}{\sum_{a'} n(a')}
\]

Regularized policy optimization:

\[
\bar{\pi} = \arg\max_\pi (q^T \pi - \lambda_N KL [\pi_\theta, \pi])
\]
Main result: AlphaZero as policy optimization

\[ \hat{\pi}(a) = \frac{n(a)}{\sum_{a'} n(a')} \]

Regularized policy optimization:

\[ \bar{\pi} = \arg \max_{\pi} \left( q^T \pi - \lambda_N \text{KL} \left[ \pi_{\theta}, \pi \right] \right) \]
Main result: AlphaZero as policy optimization

Regularized policy optimization:

\[ \hat{\pi}(a) = \frac{n(a)}{\sum_{a'} n(a')} \]

\[ \bar{\pi} = \arg \max_\pi \left( q^T \pi - \lambda N \text{KL} [\pi_\theta, \pi] \right) \]
Main result: AlphaZero as policy optimization

Regularized policy optimization:

$$\bar{\pi} = \arg \max_\pi \left( q^T \pi - \lambda_N KL [\pi_\theta, \pi] \right)$$

Gradient ascent step from $\hat{\pi}$:

$$\hat{\pi} \leftarrow \hat{\pi} + \eta \cdot \frac{\partial}{\partial n} \left( q^T \hat{\pi} - \lambda_N KL [\pi_\theta, \hat{\pi}] \right)$$
Main result: AlphaZero as policy optimization

Regularized policy optimization:

\[ \bar{\pi} = \arg \max_{\pi} \left( q^T \pi - \lambda_N KL [\pi_\theta, \pi] \right) \]

Gradient ascent step from \( \hat{\pi} \):

\[ \hat{\pi} \leftarrow \hat{\pi} + \eta \cdot \frac{\partial}{\partial n} \left( q^T \hat{\pi} - \lambda_N KL [\pi_\theta, \hat{\pi}] \right) \]

Corresponding action selection:

\[ a^* = \arg \max_{a} \left[ \frac{\partial}{\partial n} \left( q^T \hat{\pi} - \lambda_N KL [\pi_\theta, \hat{\pi}] \right) \right] \]
How to use $\bar{\pi}$ in AlphaZero

AlphaZero can be broken down into three main components:

- **Act:** sample $\hat{\pi}$
- **Learn:** train towards $\hat{\pi}$
- **Search:** action selection
How to use $\bar{\pi}$ in AlphaZero

AlphaZero can be broken down into three main components:

- **Act:** sample $\hat{\pi}$ → sample $\bar{\pi}$
- **Learn:** train towards $\hat{\pi}$ → train towards $\bar{\pi}$
- **Search:** action selection → sample $\bar{\pi}$

Use $\bar{\pi}$ instead
How to use $\overline{\pi}$ in AlphaZero

AlphaZero can be broken down into three main components:

- **Act:** sample $\hat{\pi}$ → sample $\overline{\pi}$

- **Learn:** train towards $\hat{\pi}$ → train towards $\overline{\pi}$

- **Search:** action selection → sample $\overline{\pi}$

Use $\overline{\pi}$ instead

The learning becomes regularized policy optimization using search Q–values for its Q–values estimates.
How to use $\tilde{\pi}$ in AlphaZero

AlphaZero can be broken down into three main components:

- **Act:** sample $\hat{\pi}$ → sample $\tilde{\pi}$
- **Learn:** train towards $\hat{\pi}$ → train towards $\tilde{\pi}$
- **Search:** action selection → sample $\tilde{\pi}$

The learning becomes regularized policy optimization using search Q-values for its Q-values estimates.

Search becomes regularized policy optimization algorithm on imaginary trajectories.

Use $\tilde{\pi}$ instead.
Results on Ms Pacman (Atari)
Results on Ms Pacman (Atari)

5 simulations

50 simulations

MuZero

Policy Optimization

Ms Pacman, 8 seeds -- 8 GPU, 4096 CPU actors
Ablation study on Ms Pacman (Atari)

- 5 simulations
- 50 simulations

Episode return

- MuZero
- PO
- Act
- Learn
- Search

Ms Pacman, 8 seeds -- 8 GPU, 4096 CPU actors
Application: DM Control Suite
Application: DM Control Suite

level_name=cheetah_run

level_name=walker_run

level_name=walker_stand

level_name=walker_walk
Summary

What we showed:

- AlphaZero approximates the solution to a regularized policy optimization problem.
Summary

What we showed:

- **AlphaZero** approximates the solution to a regularized policy optimization problem.

- **Experimentally**, using the exact solution provides improved performance without requiring additional parameter tuning.
Summary

What we showed:

- **AlphaZero** approximates the solution to a regularized policy optimization problem.
- Experimentally, using the exact solution provides improved performance without requiring additional parameter tuning.