

Wuziqi

**wei-cui**

Last modified just a moment ago

An application of deep reinforce learning on Gomoku - an [abstract strategy board game](#). It is implemented with tensorflow.

Github link: <https://github.com/WellsCui/neural-network/tree/master/wuziqi>

- [Introduction of Gomoku](#)
- [Introduction of deep reinforce learning](#)
 - [Reinforcement learning \(RL\)](#)
 - [Dynamic programming \(also known as dynamic optimization\)](#)
 - [Markov decision processes \(MDPs\)](#)
 - [Temporal difference \(TD\) learning](#)
 - [Bellman equation](#)
 - [Reinforce Learning Online Course](#)
- [Design of reinforce learning Agent in Wuziqi](#)
 - [Math Equations for value network](#)
 - [Network Models](#)
 - [Rollout / Rehearsal](#)
- [Training](#)

Introduction of Gomoku

Gomoku is an [abstract strategy board game](#). Also called **Gobang** or **Five in a Row**, it is traditionally played with [Go](#) pieces (black and white stones) on a Go board, using 15×15 of the 19×19 grid intersections.^[1]

Players alternate turns placing a stone of their colour on an empty intersection. The winner is the first player to form an unbroken chain of five stones horizontally, vertically, or diagonally.

wiki: <https://en.wikipedia.org/wiki/Gomoku>

[The Gomoku AI Tournament](#)

Introduction of deep reinforce learning

https://en.wikipedia.org/wiki/Reinforcement_learning

Reinforcement learning (RL)

An area of [machine learning](#) inspired by [behaviorist psychology](#), concerned with how [software agents](#) ought to take [actions](#) in an *environment* so as to maximize some notion of cumulative *reward*.

Dynamic programming (also known as dynamic optimization)

A method for solving a complex problem by breaking it down into a collection of simpler subproblems, solving each of those subproblems just once, and storing their solutions.

Markov decision processes (MDPs)

provide a mathematical framework for modeling [decision making](#) in situations where outcomes are partly [random](#) and partly under the control of a decision maker. MDPs are useful for studying a wide range of [optimization problems](#) solved via [dynamic programming](#) and [reinforcement learning](#).

Temporal difference (TD) learning

It is a prediction-based [machine learning](#) method. It has primarily been used for the [reinforcement learning](#) problem, and is said to be "a combination of [Monte Carlo](#) ideas and [dynamic programming](#) (DP) ideas."^[1]

Bellman equation

It is named after its discoverer, [Richard Bellman](#), also known as a *dynamic programming equation*, is a [necessary condition](#) for optimality associated with the mathematical [optimization](#) method known as [dynamic programming](#).

Reinforce Learning Online Course

<http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html>

Design of reinforce learning Agent in Wuziqi

Like the design of Alphago, Reinforce learning Agent is consist of an action value network and a policy network. There is also a variation which is consist of a value network and a policy network.

the design of Alphago <https://gogameguru.com/i/2016/03/deepmind-mastering-go.pdf>

Math Equations for value network

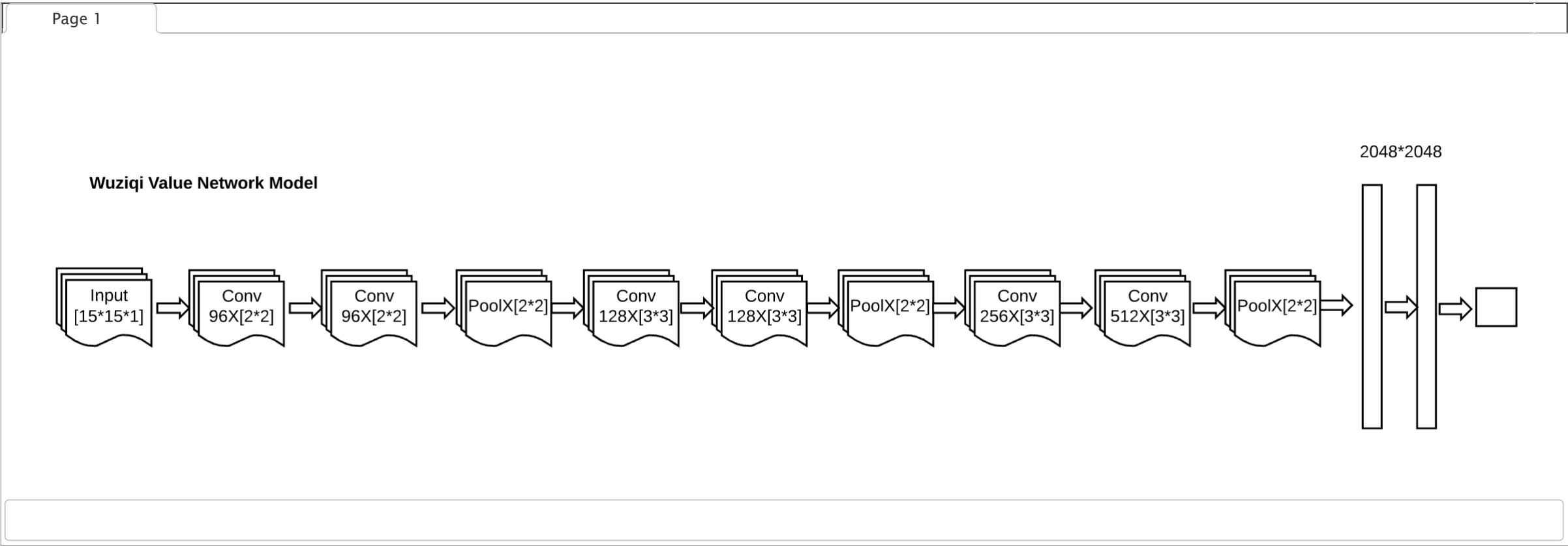
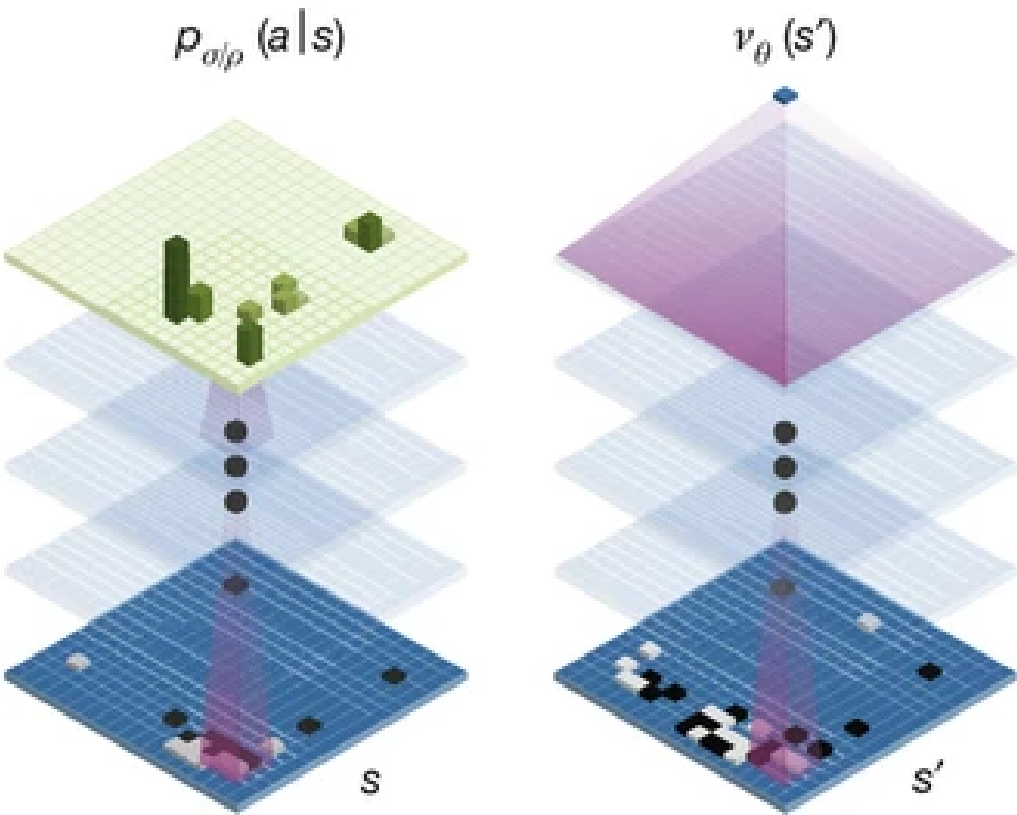
$$V^*(s) = \max_a \{R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^*(s')\}.$$

$$V^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} P(s'|s, \pi(s)) V^\pi(s').$$

Network Models

Policy network

Value network

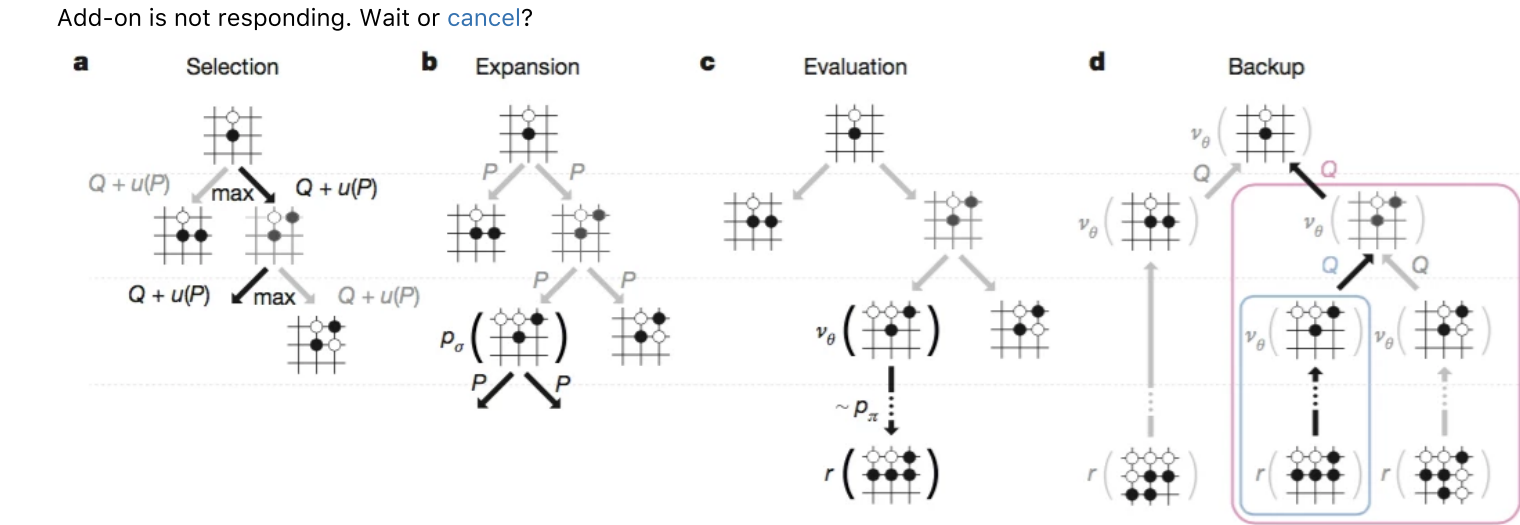


Rollout / Rehearsal

Agent do rehearsals before each move. It chooses the first action of the best rehearsal. Before each move, it select 20 candidate actions which partly from the policy network and partly from the direct neighbors of last actions of agent and its opponent. The agent rehearsals with opponent policy. And it also rehearsal with reversed state which means it thinks what it should do if it was the opponent in that state. So there are 40 rehearsals with 20 candidate actions on state and reversed state. During a rehearsal, it select the best candidate action for each move out of 20 candidate actions by using value network to evaluate those candidate actions. Each rehearsal is limited to 5 steps on each side.

502 Bad Gateway

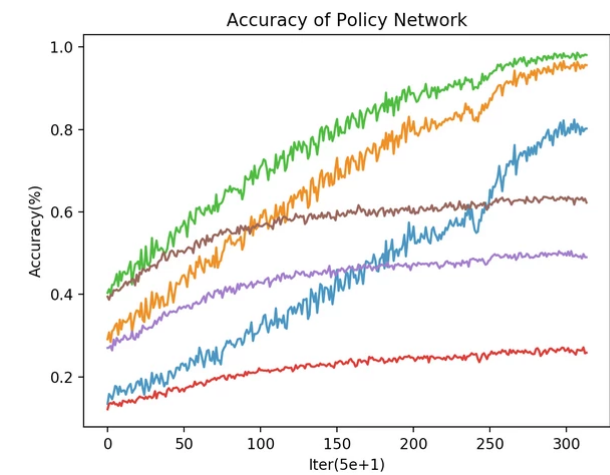
The server returned an invalid or incomplete response.




Training

Training data is from the 9000 games from 2009 to 2017 in [The Gomoku AI Tournament](#) . The following figures are the train results after the networks were trained in google cloud compute vm with one Nvidia K80 gpu for 8 hours.

A reinforcement learning (RL) policy network pp is initialised to the SL policy network, and is then improved by policy gradient learning to maximize the outcome (i.e. winning more games) against previous versions of the policy network. A new data-set is generated by playing games of self-play with the RL policy network. Finally, a value network v_θ is trained by regression to predict the expected outcome (i.e. whether the current player wins) in positions from the selfplay data-set.



 [Like](#) Be the first to like this

No labels 