Artificial Intelligence, Spring 2017

Homework 1 – Thoughts about AI

Xinglu Wang *3140102282* *ISEE 1403, ZJU*

1 AlphaGo

I try to understand the mechanism of AlphaGo[1]. My conclusion/thought is that AlphaGo is powerful but is not mysterious and there is a long way to general AI (Current AI just act rationally on specific and *isolated* task). Let me give some shallow introductions to AlphaGo.

1.1 MCTS

Computer is good at computation, so I think search is *still* the basic key component, while the prediction methods(CNN, SL, RL) are just to improve.

Def. for Go Game	Symbol
state space	$\mathcal{S}$
action space for $s \in \mathcal{S}$	$\mathcal{A}_s$
state transitions at step t	$s_{t+1} = f(s_t, a_t)$
reward function at step t	$r(s_t)$
outcome of game at terminal time T	$z_t = \pm r(s_T)$

The traditional method (*minmax* search) define an optimal value function recursively and need to expand the tree to the terminal time-step when calculating.

$$v^*(s) = \begin{cases} z_T & \text{if } s = s_T, \\ \max\limits_a \left(-v^*(f(s,a))\right) & \text{otherwise} \end{cases} \tag{1}$$

To calculate optimal value, we observe that *minmax* search involve two steps:

- If the policy for us and opponent is fixed, we need to calculate expected outcome.
- The policy is taking minmax optimal actions in each step.

An alternative approach to minimax search is Monte-Carlo tree search (MCTS). Similar to *minmax* search, MCTS make double approximation to get $V^n(s) \approx v^*(s)$:

- Use n Monte-Carlo simulation to estimate expectation.
- Use a simulation policy to take actions a_t controlled by function:

$$a_t = \arg\max_a (Q^n(s,a) + u(s,a)) \tag{2}$$

In UCT, $Q^n(s,a) = -V^n(f(s,a))$ and $u(s,a)$ is responsible for encouraging explorations. (I try to understand "encourage explorations" in AlphaGo's method.)

AlphaGo is based on MCTS integrated with CNN. There are policy function/network, p_σ (accurate) and p_π (fast); value function/network v_θ. The eq. (2) is split into two part shown below.

The overall evaluation is evaluated by mixed value network v_θ and rollout evaluations with weight λ:

$$Q(s_t, a_t) = (1-\lambda)v_\theta(f(s_t, a_t)) + \lambda z_t \tag{3}$$

The outcome z_t is random rollout outcome follow the fast policy function p_π (i.e. $a_t \sim p_\pi(\cdot|s_t)$).

$$u(s,a) \propto \frac{P(s,a)}{1+N(s,a)} \tag{4}$$

Here $P(s,a) = p_\sigma(a|s)$ is the prior probabilities for state/node s, calculated by accurate policy network p_σ. We can explain why $u(s,a)$ "encourage explorations" now: Initially, this search control strategy prefer actions with prior probability and low visit count. But after iterating and updating $Q(s,a)$, it will prefer to actions with high value.

To conclude, $u(s,a)$ (concerning $p_\sigma(a|s)$) choose the *local* optimal action while $Q(s_t,a)$ (concerning $v_\theta(f(s_t,a))$ choose the *global* optimal action.

Meanwhile, we can further explain why use(and train) p_σ, p_π, and v_θ rather than others afterwards. Ref. to sec. 1.2.

1.2 Policy & Value Network

Illustrated in fig. 1, the architecture of p_σ, p_π, p_ρ, and v_θ is easy to understand. We just need to be careful with the notation $p_{\sigma/\rho}(a|s)$, which means $p_\sigma(a|s)$ *or* $p_\rho(a|s)$, rather than $P_{Y|X}(y|x)$. (As to p_ρ, I need more time to figure out how to apply *police gradient descent* in Reinforcement Learning).

p_σ, p_π are trained by SL(classification), p_ρ is trained by RL and v_θ is trained by SL(regression).

- p_π composed with linear units and hand-crafted feature, with less accuracy and fast speed, is proper for rollout search.

- p_σ trained on Human expert datasets, learn rules of Go and have more diverse choose than p_ρ, thus used in initialize prior probabilities.

- v_θ trained on huge generated datasets, have a good sense of global optimal intuition.

This two modules trained on such huge datasets, and only achieve Amateur dan if not combined with MCTS. Meanwhile they cannot generalize to other tasks. Thus, these modules are not so clever as human and there is a long way to go.

2 My Opinions

Conclude from sec. 1 and [2], my opinions are:

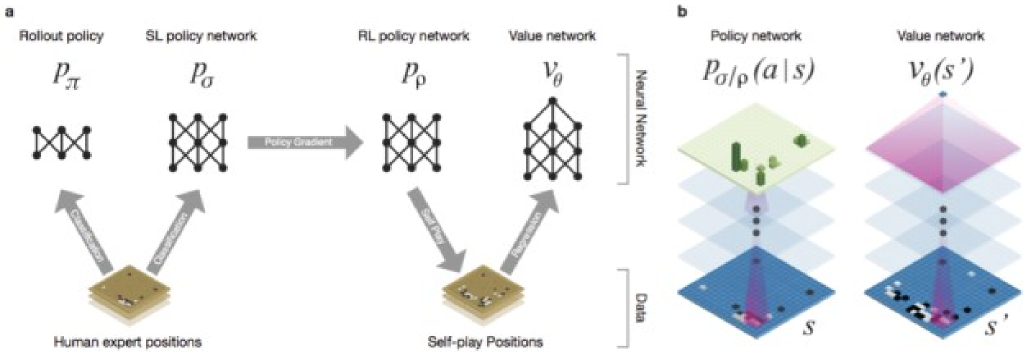

Figure 1: **Left**: Neural network training pipeline; **Right**: Network architecture; **Note:** $p_{\sigma/\rho}(a|s)$ means $p_\sigma(a|s)$ or $p_\rho(a|s)$

- The mathematics principle behind Search and Reinforcement Learning are quite interesting. Some ideas behind the algorithms in AI are delicate. I am looking forward to learn AI systematically!

- As stated in AlphaGO (sec. 1), CNN help AI form some intuition about global optimum. But I do not think it is the intuition human have, because it needs so much training data, still limited in isolated task and cannot transfer to other domain. There is a long run to go, and as researchers, we just try to improve it.

- People are crazy about AI, but we should keep calm, try to figure out why some algorithms in AI perform so good and try to form more delicate idea.

References

[1] Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 529.7587 (2016): 484-489. Publishing Company , 1984-1986.

[2] https://www.quora.com/Are-neural-networks-the-future-of-AI

[3] https://youtu.be/yCALyQRN3hw?list=PLqYmG7hTraZA7v9Hpbps0QNmJC4L1NE3S&t=11434