SNE 9 (AlphaGo)

1. Podstawowe Teorie

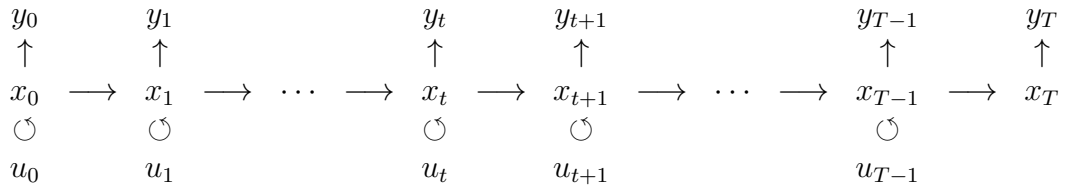
Automat (*ang.* automaton, automata)

u_t wejście (*ang.* input) w chwili t $(0 \le t \le T-1)$

x_t stan (*ang.* state) w chwili t $(0 \le t \le T)$

y_t wyjście (*ang.* output) w chwili t $(0 \le t \le T)$

$$\begin{cases} x_{t+1} = f(x_t, u_t) \\ y_t = g(x_t, u_t) \end{cases}$$



Dywergencja Kullbacka-Leiblera, relatywna entropia, entropia względna
(Kullback–Leibler divergence, relative entropy)
Solomon Kullback (1907–1994) amerykański matematyk, statystyk
Richard Leibler (1914–2003) amerykański matematyk, statystyk

$$D = \sum_{a \in \mathcal{A}} p(a|s) \log \frac{p(a|s)}{p_\sigma(a|s)}$$

$D \ge 0$ ("$=$" $\Leftrightarrow \forall a \in \mathcal{A}\colon p_\sigma(a|s) = p(a|s)$)

Stochastic gradient descent (or ascent) Shun'ichi Amari, *IEEE Trans.* **EC** (1967)
$D(\sigma) = \sum_{a \in \mathcal{A}} D_a(\sigma) \longrightarrow$ minimum (or maximum)

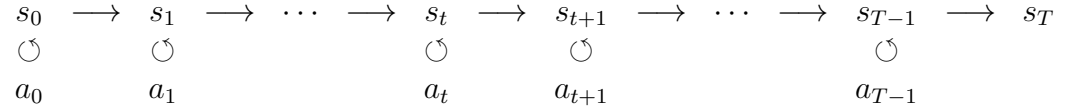
Pseudocode
Randomly shuffle examples in the training set.
Do for each $a \in \mathcal{A}$: $\sigma = \sigma - c\frac{\partial D_a}{\partial \sigma}$.

$D_a(\sigma) = p(a|s) \log \frac{p(a|s)}{p_\sigma(a|s)} = p(a|s) \log p(a|s) - p(a|s) \log p_\sigma(a|s)$

$$\Delta\sigma \propto \frac{\partial \log p_\sigma(a|s)}{\partial \sigma} \quad \text{(stochastic gradient ascent)}$$

Reinforcement Learning (RL) for Markov Decision Process (MDP)
(Uczenie przez Wzmacnianie)

- $s_t \in \mathcal{S}$ stan (*ang.* state), *board position w Go* w kroku t $(0 \le t \le T)$
- $a_t \in \mathcal{A}$ akcja (*ang.* action), *move w Go* w kroku t $(0 \le t \le T-1)$



- Prob(s_0) warunek początkowy
- Prob$(s_{t+1}|s_t, a_t)$ prawdopodobieństwo przejścia (Markov process)

r_t nagroda (*ang.* reward)
R return, dywidenda (*ang.* dividend), defined as the sum of future discounted rewards

$$R = \sum_{t=0}^{T} \gamma^t r_t \qquad (R = r_T \text{ dla AlphaGo})$$

$\gamma \in (0, 1)$ discount-rate, stopa dyskontowa, stałe

$p(a|s)$ **policy** (polityka $p(a|s)$)

Cel RL Znaleźć optimal policy p^*

$$p^* = \underset{p}{\operatorname{argmax}}\, \mathbb{E}^p[R]$$

$v^p(s)$ state value function

$$v^p(s_t) = \mathbb{E}^p\left[\sum_{k=t}^{T} \gamma^{k-t} r_k \middle| s_t\right]$$

$Q^p(s, a)$ action value function, Q-function

$$Q^p(s_t, a_t) = \mathbb{E}^p\left[\sum_{k=t}^{T} \gamma^{k-t} r_k \middle| s_t, a_t\right]$$

Bellman equation
Richard Bellman (1920–1984)
amerykański matematyk, twórca programowania dynamicznego (Dynamic Programming 1957)
(Ph. D 1947 Lefschetz)

$$v^{p^*}(s_t) = \mathbb{E}[r_t + \gamma v^{p^*}(s_{t+1})|s_t]$$
$$Q^{p^*}(s_t, a_t) = \mathbb{E}[r_t + \gamma \max_b Q^{p^*}(s_{t+1}, b)|s_t, a_t]$$

Wtedy możemy obliczyć p^* w następujący sposób.

$$\begin{aligned} p^*(a_t|s_t) &= \underset{a_t}{\operatorname{argmax}}\, \mathbb{E}[v^{p^*}(s_{t+1})|s_t, a_t] \\ &= \underset{a_t}{\operatorname{argmax}}\, Q^{p^*}(s_t, a_t) \end{aligned}$$

MCTS (Monte Carlo tree search)
Metoda Monte Carlo
N. Metropolis and S. Ulam, The Monte Carlo method, *Journal of the American Statistical Association* **44** (1949), 335–341.
Stanisław Ulam (ur. 1909 we **Lwowie**, zm. 1984 w Santa Fe w stanie Nowy Meksyk)
polski i amerykański matematyk

Historia MCTS (cytat z *Wikipedea*)
Metoda Monte Carlo, oparta na koncepcji losowego próbkowania, powstała w latach 1940. W roku 1992 B. Brügmann jako pierwszy zastosował ją w programie do gry w go, ale jego pomysłu nie potraktowano wówczas poważnie. W roku 2006, zwanym rokiem rewolucji Monte Carlo w go, R. Coulom opisał zastosowanie metody Monte Carlo do przeszukiwania drzew wariantów i wprowadził nazwę Monte-Carlo Tree Search, L. Kocsis i Cs. Szepesvári opracowali algorytm

UCT, a S. Gelly i inni zastosowali UCT w swoim programie MoGo.

Każda tura tej metody składa się z czterech kroków (cytat z *Wikipedea*):
Wybór (ang. selection): zaczynając od korzenia drzewa R, wybieraj kolejne węzły potomne, aż dotrzesz do liścia drzewa L. Poniżej więcej o takim sposobie wyboru węzłów potomnych, dzięki któremu drzewo wariantów rozrasta się w kierunku najbardziej obiecujących ruchów, co stanowi clou metody MCTS.
Rozrost (ang. expansion): o ile L nie kończy gry, utwórz w nim jeden lub więcej węzłów potomnych i wybierz z nich jeden węzeł C.
Symulacja (ang. playout): rozegraj losową symulację z węzła C
Propagacja wstecz (ang. backpropagation): na podstawie wyniku rozegranej symulacji uaktualnij informacje w węzłach na ścieżce prowadzącej od C do R.

2. AlphaGo

Training pipeline in AlphaGo follows $\mathbf{A} \to \mathbf{B} \to \mathbf{C} \to \mathbf{D}$.
(Uczenie głębokie jest używane w **A**, **B** i **C**.)

CNN (Convolutional Neural Network)

A. SL Policy network $p_\sigma(a|s)$ (SL=Supervised Learning)
Sieć polityki z uczeniem nadzorowanym

$$s \longrightarrow \underset{\text{z parametrami } \sigma}{\text{CNN}} \longrightarrow \Big(p_\sigma(a|s)\Big)_{a\in\mathcal{A}}$$

$$\Delta\sigma \propto \frac{\partial \log p_\sigma(a|s)}{\partial\sigma} \quad \text{(stochastic gradient ascent)}$$

B. RL Policy network $p_\rho(a|s)$ (RL=Reinforcement Learning)
Sieć polityki z uczeniem przez wzmacnianie

$$s \longrightarrow \underset{\text{z parametrami } \rho}{\text{CNN}} \longrightarrow \Big(p_\rho(a|s)\Big)_{a\in\mathcal{A}}$$

$r(s)$ reward function

$$z_t = \begin{cases} +r(s_T) = +1 & \text{for winning} \\ -r(s_T) = -1 & \text{for losing} \end{cases}$$

$$\Delta\rho \propto \frac{\partial \log p_\rho(a_t|s_t)}{\partial\rho} z_t \quad \text{(stochastic gradient ascent)}$$

C. RL Value network $v_\theta(s)$ (RL=Reinforcement Learning)
Sieć wartości z uczeniem przez wzmacnianie

$$s \longrightarrow \underset{\text{z parametrami } \theta}{\text{CNN}} \longrightarrow v_\theta(s)$$

$$v^p(s) = \mathbb{E}[z_t|s_t = s, a_{t\ldots T} \sim p] \quad (\textbf{(state) value function})$$

$v_\theta \to v^{p_\rho} \approx v^*$ (Use stochastic gradient descent to minimize the least squared error.)

$$\frac{1}{2}\sum_s (z - v_\theta(s))^2 \longrightarrow \text{minumum}$$

$$\Delta\theta \propto \frac{\partial v_\theta(s)}{\partial\theta}(z - v_\theta(s)) \quad \text{(stochastic gradient descent)}$$

Combining policy networks and value networks in an MCTS algorithm

$$a_t = \underset{a}{\operatorname{argmax}}(Q(s_t, a) + u(s_t, a))$$

$P(s, a) = p_\sigma(a|s)$ prior probability
$1(s, a, i)$ indicates whether an edge (s, a) was traversed during the ith iteration

$$N(s, a) = \sum_{i=1}^{n} 1(s, a, i) \quad \text{(visit count)}$$

$$u(s, a) \propto \frac{P(s, a)}{1 + N(s, a)} \quad \text{(bonus)}$$

(proportional to the prior probability but decays with repeated visits to encourage exploration)
s_L the leaf node
z_L the outcome of a random rollout played out until terminal step T using the fast **rollout policy** $p_\pi(a|s)$
Rollout policy $p_\pi(a|s)$ a faster but less acurate rollout policy
$p_\pi(a|s) = \frac{e^{\pi_a \cdot s}}{\sum_a e^{\pi_a \cdot s}} \to \max$ (linear softmax)
($\pi_a \cdot s$ iloczyn skalarny, π_a weight, wektor wag)

$$V(s_L) = (1 - \lambda)v_\theta(s_L) + \lambda z_L$$

s_L^i the leaf node from the ith iteration

$$Q(s, a) = \frac{1}{N(s, a)} \sum_{i=1}^{n} 1(s, a, i)V(s_L^i) \quad \textbf{(action value (function), } Q\textbf{-function)}$$

D. MCTS in AlphaGo
a Selection: Traverse the tree by selecting the edge with maximun $Q + u(P)$.
b Expansion: Expand the leaf node by the policy network p_σ. p_σ is stored as $P(s, a)$.
c Evaluation: Evaluate the leaf node by computing the winner with reward function r.
d Backup: Update action values Q.

3. Dodatek (dla zainteresowanego Czytelnika)
Ramsey games (Gry Ramseya), clique games, positional games
Frank Ramsey (1903–1930) angielski matematyk, ekonomista; Senior Wrangler* w 1923

Ramsey był dobrym znajomym ekonomisty Johna Keynesa, którego pionierskie prace z rachunku prawdopodobieństwa zainspirowały Ramseya do prac nad tak zwanym prawdopodobieństwem subiektywnym. Na odwrót, analizy Ramseya miały wpływ na poglądy Keynesa na temat prawdopodobieństwa. Ramsey twierdził, że aksjomaty rachunku prawdopodobieństwa powinny odzwierciedlać stopnie naszego subiektywnego przekonania. Jego prace z tej dziedziny stały się szerzej znane dopiero w latach 50. XX wieku. (cytat z *Wikipedea*)

Twierdzenie. (Ramsey) *Dla każdej dodatniej liczby całkowitej k istnieje dodatnia liczba całkowita N taka, że jeśli wszystkie krawędzie grafu pełnego N wierzchołków są kolorowane na niebiesko lub czerwono, wtedy musi być k wierzchołki takie, że wszystkie te krawędzie mają ten sam kolor.*

∗ 1930: Jacob Bronowski, 1973: Lee Hsien Loong (premier Singapuru od 2004!)

Najmniejsza taka liczba całkowita N jest znana jako *liczba Ramseya* $R(k)$.

Fakty. Wiemy, że $R(3) > 5$ i $R(3) = 6$.

Twierdzenie. (Erdős) $R(k) \leq 4^k$
P. Erdős, Some remarks on the theory of graphs, *Bull. A.M.S.* **53** (1947), 292–4.
Dowód. W rewolacyjnym dowodzie Erdősa używany jest probabilistyczny argument. ■
Paul Erdős (ur. 26 marca 1913 w Budapeszcie, zm. 20 września 1996 w Warszawie) – węgierski matematyk.

Problem (Hipoteza). $\exists\ a > \sqrt{2}, b < 4\ \forall\ k \gg 1\ \ (a^k \leq R(k) \leq b^k)$?

Zadanie dla zainteresowanego Czytelnika
(1) Zdefiniować "ciekawą" grę Ramseya.
(2) Implementować grę Ramseya (grafiki itd.).
(3) Napisać kod uczenia głębokiego dla gry Ramseya.

J. Beck, Ramsey games, *Discrete Mathematics* **249** (2002) 3–30