

brmiversity: Umělá inteligence a teoretická informatika

Přednáška č. 8

Petr Baudiš <pasky@ucw.cz>

brmlab 2011



Outline

① Adaptivní agenti

② Neuronové sítě

③ Vyčísitelnost

Metody pro řízení agentů

- Animat: Sensory — umělá mysl — efekторы
- Action selection problem

Metody pro řízení agentů

- Animat: Sensory — umělá mysl — efekty
- Action selection problem

- Reaktivní if-then pravidla
- Konečné stavové automaty
- Free-flow architektura
- Neuronové sítě (příště)
- SOAR (později)

Reaktivní plánování

- **Reaktivní plánování** vybírá pouze přímo následující akci na základě aktuálního kontextu

Reaktivní plánování

- **Reaktivní plánování** vybírá pouze přímo následující akci na základě aktuálního kontextu
- If-then pravidla: **If p then A** , prioritizovaná (“produkční pravidla”)
- A může být akce nebo sekvence akcí s pamětí (reflexy)
- Rychlá evaluace: RETE a lazy vyhodnocování (později)

(Brooks, 1986; Wooldridge, 2002)

```
# When starts: not at home && be in picking state
if see_obstacle then change_direction
if basketful_of_m. and picking then stop_picking
if see_mush. and picking then pick_up_the_mush.
if midday and picking then stop_picking
if home then END
if picking then move_random
if not_picking then move_home
```

Simple Hierarchical Reactive Planning

- Dekompozice? SHRP!
- Hierarchické reaktivní plánování: A může být sada dalších úřů — podcíle, úřový strom s implicitním *and* po cestě
- Modelový cyklus:
Appetitive → Taxis → Consumatory → Clean
- Jak vybírat top-level cíle? Jízdní řád, pudy, dlouhodobé plány
- Subsumpční architektura

Parallel-rooted Ordered Slip-stack Hierarchy

- POSH: Hierarchická if-then pravidla s pudy (drives), postupy (competencies) a skripty (action patterns)
- Drive: Základní reaktivní *patterny* okamžitě spouštějící reakce
- Competency: Reaktivní *plány* postupně spouštějící reakce
- Action pattern: Slepá posloupnost akcí

(Bryson, 2002)

```
(RDC LIFE (GOAL (fail))
 (DRIVES
  ((grooming (TRIGGER ((want-to-groom))) groom-comp))
  ((receive-grooming (TRIGGER ((being-groomed))) tolerate-grooming))
  ((exploring (TRIGGER ()) explore-comp)))
 (C groom-comp (GOAL (fail)) (ELEMENTS
  ((groom-gp (TRIGGER ((partner-chosen) (aligned-w-target))) groom-note))
  ((new-partner-gp (TRIGGER ((being-groomed))) choose-groomer-as-partner))
  ((align-w-gp (TRIGGER ((partner-chosen) (touching-target))) engage))
  ((touch-gp (TRIGGER ((partner-chosen))) approach))
  ((choose-gp (TRIGGER ()) choose-grooming-partner))))
 (C explore-comp (GOAL (fail)) (ELEMENTS ...)))
```


Belief — Desire — Intention

- Practical reasoning: Deliberative vs. means-ends
- Obecná šablona výběru top-level cílů: BDI
- Máme nějakou *představu*, na základě nich *toužíme* po různých věcech a z toho se dopracujeme k *úmyslu* něco konkrétního provést
- Beliefs mohou být mylné
- Desires jsou budoucí zajímavé možnosti agenta
- Intentions jsou akce, ke kterým se agent rozhodl
- Cyklus: Vygeneruj možnosti, rozhodni se pro jednu, na základě vstupů profiltruj možnosti
- Provádí se intentional stack, POSH-like

FSM a Fuzzy logika

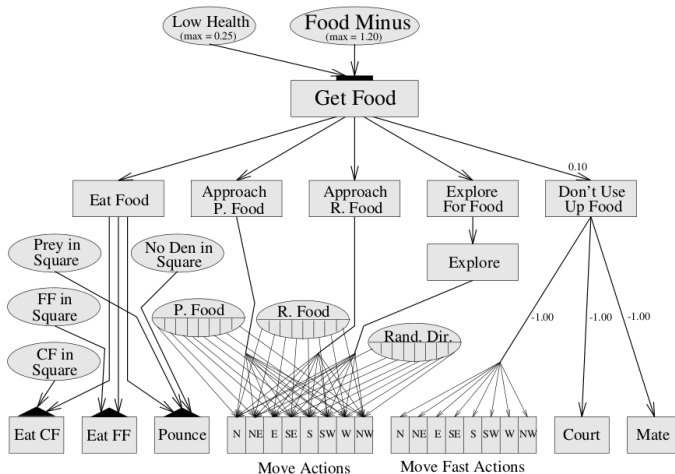
- Konečný automat; stav je akce (aktivuj efektor), přechod je vstup (vstup ze senzorů)
- Hierarchický FSM — některé stavy jsou samy FSM; umožňuje dekompozici
- Probabilistický FSM — přechody jsou asociovány i s pravděpodobnostmi

FSM a Fuzzy logika

- Konečný automat; stav je akce (aktivuj efektor), přechod je vstup (vstup ze senzorů)
- Hierarchický FSM — některé stavy jsou samy FSM; umožňuje dekompozici
- Probabilistický FSM — přechody jsou asociovány i s pravděpodobnostmi

- Fuzzy množina X je (X, m) , $m: X \rightarrow [0, 1]$
- Defuzzifikace — weighted MOM
- Fuzzy FSM — fuzzy stavy i přechody, MIN-MAX metoda a prahování stavů
- Modelování emocí

Free-flow architektura



(Tyrrell, 1993)

Alternativy: Afordance; viz později reprezentace znalostí.

Otázky?

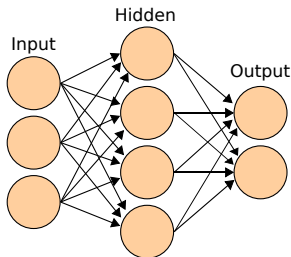
Příště: Metody pro učení agentů.

Outline

- 1 Adaptivní agenti
- 2 Neuronové sítě
- 3 Vyčísitelnost

ANN Revisited

- Umělé neurony (“výpočetní krabičky”) dostávají vstupy (čísla) a na jejich základě generují výstup (číslo)
- Obvykle: Vrstvy striktně oddělené, vstupní vrstva se vstupy zvnějšku, výstupní vrstva s výstupem pro uživatele, skryté vrstvy vyhodnocují různé charakteristiky vstupů
- Dnes: Více vrstev neuronů, jak je učit?



Backpropagation Revisited

- Myšlenka: Závislosti mezi vstupy a výstupy dokážeme přiměřeně matematicky popsat
- Chceme upravit váhy podle *chyby*, kterou propagovaly; větší váha nese větší chybu
- Iterujeme učení podle vstupních množin:
 - Zjistíme chybu výstupu
 - Spočítáme *gradient* chyby podle vah jednotlivých spojů
 - Chybu se pokusíme zredukovat posunutím vah proti gradientu
 - Chybu “zpětně šíříme” do předchozí vrstvy a opakujeme

Vylepšení učení ANN

- Inicializace vah: rozumně malé, náhodné, vyvážené
- Chytřejší sestup podle gradientu
- Rozšíření gradientu derivacemi druhého řádu
- Relaxační metody: perturbace vah
- Modulární sítě, úprava parametrů (prahy, neurony)
- Genetické algoritmy

BP s momentem

Setrvačnost (moment): Zamez oscilacím v úzkých údolích
(zejména jsou-li osy různě dlouhé)

$$\Delta_E w_{i,j}(t+1) = -\alpha \frac{\partial E}{\partial w_{i,j}(t)} + \gamma(w_{i,j}(t) - w_{i,j}(t-1))$$

Najít α a γ je magie
 α musí být kompromis mezi lokálními minimy a oscilacemi

Dynamický parametr učení

$$\Delta_E w_{ij}(t+1) = -\alpha_i \frac{\partial E}{\partial w_{ij}(t)}$$

V nelineárním případě je třeba α_i volit dynamicky

Silva-Almeida

- **Urychluj**, pokud se nezměnilo znaménko; $\alpha'_i = u\alpha_i$, $u > 1$
- **Zpomaluj**, pokud se znaménko změnilo; $\alpha'_i = d\alpha_i$, $d < 1$
- Exponenciální růst/pokles může být příliš velký

Delta-bar-delta

- Pokud se nezměnilo znaménko; $\alpha'_i = u + \alpha_i$,
- Pokud se znaménko změnilo; $\alpha'_i = d\alpha_i$, $d < 1$
- Změna znaménka oproti $\delta'_i = (1 - \Phi)\Delta_i E' + \Phi\delta_i$

Super SAB

- Kombinace dynamického α a momentu
- Zpětné šíření s momentem; pokud se nezměnilo znaménko derivace, $\alpha'_i = \alpha^+ \alpha_i$
- Pokud se změnilo znaménko derivace, undo, zmenši parametr učení $\alpha'_i = \alpha^- \alpha_i$ a zkus znovu

Interní reprezentace znalostí

- Jak posoudit efektivitu interní reprezentace (vah skrytých neuronů)?

Interní reprezentace znalostí

- Jak posoudit efektivitu interní reprezentace (vah skrytých neuronů)?
- Chceme neurony, které jsou aktivní (výstup 1), pasivní (výstup 0) nebo tiché (0.5)
“Něco mezi” tolik nepřispívá k jednoznačné klasifikaci
- Jak byste udělali vzoreček?

Interní reprezentace znalostí

- Jak posoudit efektivitu interní reprezentace (vah skrytých neuronů)?
- Chceme neurony, které jsou aktivní (výstup 1), pasivní (výstup 0) nebo tiché (0.5)
“Něco mezi” tolik nepřispívá k jednoznačné klasifikaci
- Jak byste udělali vzoreček?
- TODO

Interní reprezentace znalostí

- Jak posoudit efektivitu interní reprezentace (vah skrytých neuronů)?
- Chceme neurony, které jsou aktivní (výstup 1), pasivní (výstup 0) nebo tiché (0.5)
“Něco mezi” tolik nepřispívá k jednoznačné klasifikaci
- Jak byste udělali vzoreček?
- TODO
- Ještě lepší je mít neurony, které jsou buď aktivní nebo pasivní (nejsou nikdy zbytečné)

Interní reprezentace znalostí

- Jak posoudit efektivitu interní reprezentace (vah skrytých neuronů)?
- Chceme neurony, které jsou aktivní (výstup 1), pasivní (výstup 0) nebo tiché (0.5)
“Něco mezi” tolik nepřispívá k jednoznačné klasifikaci
- Jak byste udělali vzoreček?
- TODO
- Ještě lepší je mít neurony, které jsou buď aktivní nebo pasivní (nejsou nikdy zbytečné)
- Interní reprezentace by měla být jednoznačná (hodně odlišná pro hodně odlišné výstupy)

Otázky?

Příště: ANN úplně jinak — asociativní paměti.

Outline

- 1 Adaptivní agenti
- 2 Neuronové sítě
- 3 Vyčísitelnost

Věta o rekurzi

- Mějme ČRF f . Pak $\exists a$ (tzv. pevný bod),
 $\forall x : \Psi(f(a), x) \simeq \Psi(a, x)$
- Důkaz: Mějme $a = s_1(z, z)$ (z si vymyslíme).
 $\Psi(s_1(z, z), x)$ má index e a lze zapsat jako
 $\Psi(s_1(z, z), x) \simeq \Psi(e, z, x) \simeq \Psi(s_1(e, z), x)$.
Můžeme dosadit $z = e$ a dostaneme $\Psi(f(a), x) \simeq$
 $\Psi(f(s_1(e, e)), x) \simeq \Psi(e, e, x) \simeq \Psi(s_1(e, e), x) \simeq \Psi(a, x)$.
- Varianty: Můžeme vyrobit místo a celou funkci g a udělat to pro více proměnných.

Riceova věta

- Mějme netriviální třídu ČRF A . Pak množina $B = \{x : \Psi(x) \in A\}$ není rekurzivní

Riceova věta

- Mějme netriviální třídu ČRF A . Pak množina $B = \{x : \Psi(x) \in A\}$ není rekurzivní
- Nelze určit, zda dva programy dělají to samé!

Riceova věta

- Mějme netriviální třídu ČRF A . Pak množina $B = \{x : \Psi(x) \in A\}$ není rekurzivní
- Nelze určit, zda dva programy dělají to samé!
- Důkaz sporem: Vezměme $b \in B$ a $c \notin B$.
Sestrojme $f(x)$, $f(B) = c$, $f(\bar{B}) = b$.
Nalezněme pevný bod e . Co když $e \in B$ resp. $e \notin B$?

Otázky?

Příště: Algoritmicky nerozhodnutelné problémy.

