

# Q-learning - umelá inteligencia na obzore?

Ing. Michal CHOVANEC  
Fakulta riadenia a informatiky

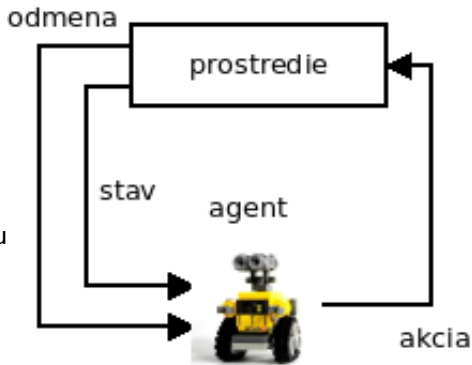
*Apríl 2016*

- Reinforcement learning
- Q-learning algoritmus
- Možnosti aproximácie



# Reinforcement learning

- Zistenie stavu
- Výber akcie
- Vykonanie akcie
- Prechod do ďalšieho stavu
- Získanie odmeny alebo trestu
- Učenie sa zo získanej skúsenosti



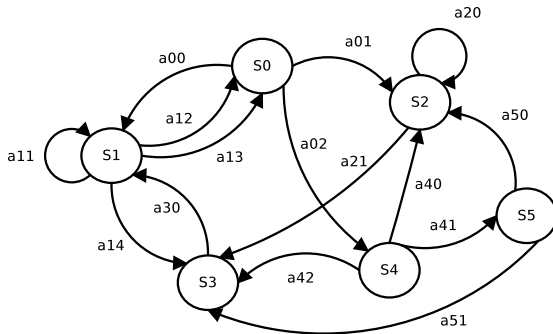
Definuje sa čo robiť, nie ako to robiť

- vďaka odmeňovacej funkcií
- agent sa môže naučiť všetky detaily problému

Lepšie konečné riešenie

- založené na skutočnej skúsenosti, nie skúsenosti programátora
- treba menej ľudského času na nájdenie dobrého riešenia

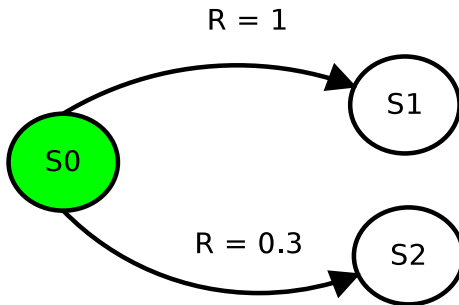
# Markovov rozhodovací proces



$$\Lambda(\pi) = \sum_{n=0}^{\infty} \gamma^n R_{\pi(n)}(s(n), s(n-1))$$

# Voľba stratégie, 2 stavy

Odmeny sú známe v každom prechode

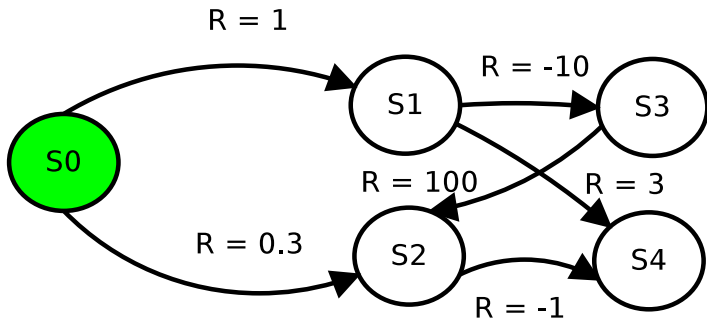


Ohodnotenie ciest :

- $Q(S_0, S_1) = 1$
- $Q(S_0, S_1) = 0.3$

Najlepšia cesta :  $S_0, S_1$

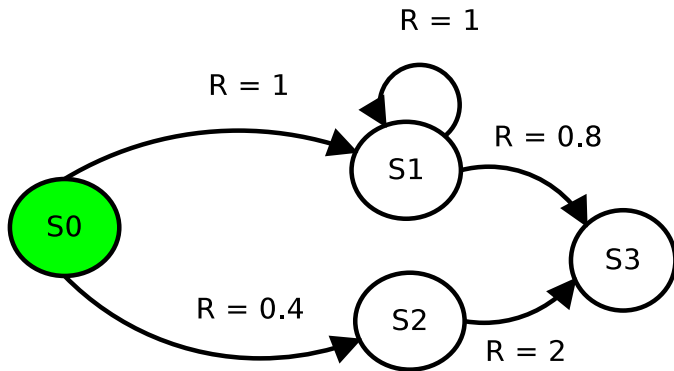
# Voľba stratégie, viac stavov



Ohodnotenie ciest :

- $Q(S0, S1, S3) = 1 + (-10) = -9$
- $Q(S0, S1, S4) = 1 + 3 = 4$
- $Q(S0, S2, S4) = 0.3 + () - 1) = -0.7$
- $Q(S0, S1, S3, S2, S4) = 1 + (-10) + 100 + (-1) = 90$
- ...

# Voľba stratégie, viac stavov

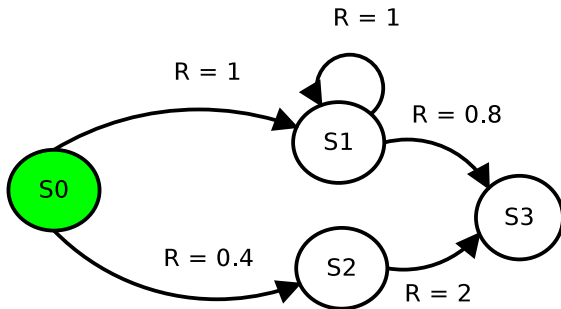


Ohodnotenie ciest :

- $Q(S0, S2, S3) = 0.4 + 2 = 2.4$
- $Q(S0, S1, S3) = 1 + 1 = 2$
- $Q(S0, S1, S1, S1) = 1 + 1 + 1 = 3$
- $Q(S0, S1, S1, S1, S1) = 1 + 1 + 1 + 1 = 4$
- $Q(S0, S1, S1, S1, S1, S1, \dots) = 1 + 1 + 1 + 1 + 1 + \dots + 1 = \infty$



# Zabúdanie $Q' = R + 0.9Q$



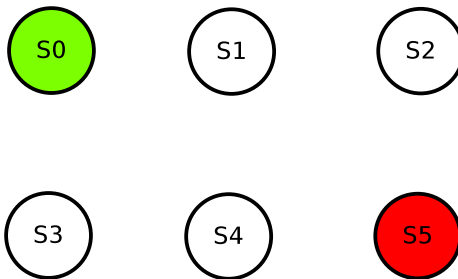
Ohodnotenie ciest :

- $Q(S0, S2, S3) = 2 + 0.9 * 0.4 = 2.36$
- $Q(S0, S1, S3) = 1 + 0.9 * 1 = 1.9$
- $Q(S0, S1, S1, S1) = 1 + 0.9 * (1 + 0.9 * 1) = 2.71$
- $Q(S0, S1, S1, S1, S1) = 1 + 0.9 * (1 + 0.9 * (1 + 0.9 * 1)) = 3.439$
- $Q(S0, S1, S1, S1, S1, S1, ...) = 10 < \text{---}$
- $Q(S0, S1, S1, S1, S1, S1, ..., S3) = 10.8 < \text{---}$

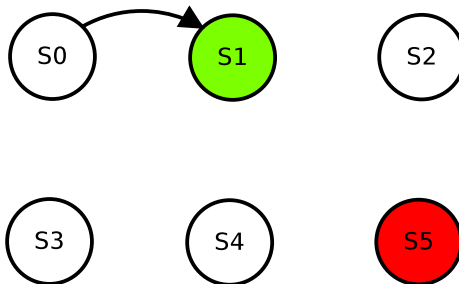
Čo ak odmeny NIE sú známe v každom prechode ?

- šachy, go, pacman
- chôdza, pohyb mechanického ramena

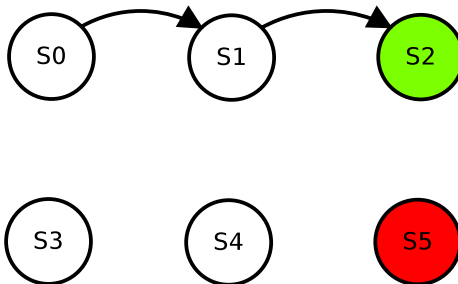
# Ilustračný príklad - inicializácia



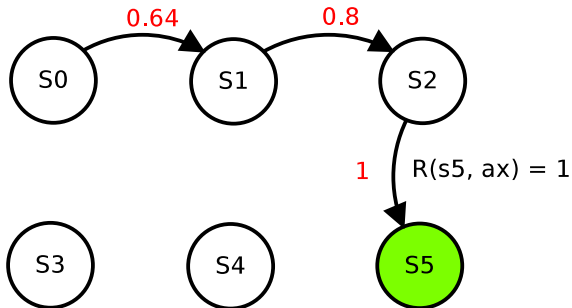
# Ilustračný príklad - prechod do ďalšieho stavu



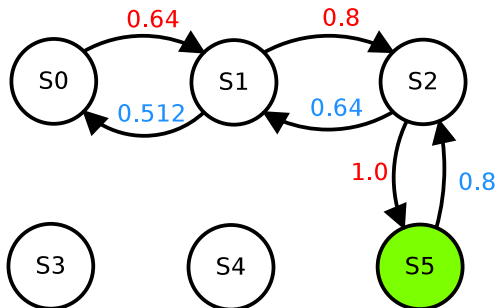
# Ilustračný príklad - prechod do ďalšieho stavu



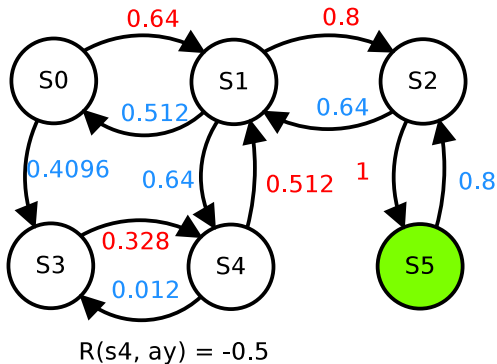
# Ilustračný príklad - prechod do cieľového stavu



# Ilustračný príklad - ďalšie prechody



# Ilustračný príklad - konečný stav algoritmu :)





# Q-learning algoritmus

Daná je množina stavov  $\mathbb{S}$  a akcií  $\mathbb{A}$ , kde  $\mathbb{S} \in \mathbb{R}^{n_s}$  a  $\mathbb{A} \in \mathbb{R}^{n_a}$ , kde  $n_s$  a  $n_a$  sú rozmery stavového vektora a vektora akcií. Je známa podmnožina východiskových stavov  $\mathbb{S}_0$ .

Existuje prechodová funkcia

$$s(n) = \lambda(s(n-1), a(n-1)) \quad (1)$$

zo stavu  $s(n)$  použitím akcie  $a(n)$  - táto funkcia je ale agentovi neznáma a odmeňovacia funkcia

$$R(s(n), a(n)) \quad (2)$$

# Odmeňovacia funkcia

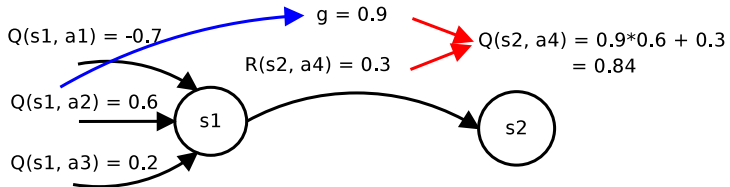
Daná je funkcia ohodnotení

$$Q(s(n), a(n)) = R(s(n), a(n)) + \gamma \max_{a(n-1) \in \mathbb{A}} Q(s(n-1), a(n-1)) \quad (3)$$

kde

- $R(s(n), a(n))$  je odmeňovacia funkcia s hodnotami
- $Q(s(n-1), a(n-1))$  je odmeňovacia funkcia v stave  $s(n-1)$  pre akciu  $a(n-1)$ ,
- $\gamma$  je odmeňovacia konštanta a platí  $\gamma \in (0, 1)$ .

# Odmeňovacia funkcia



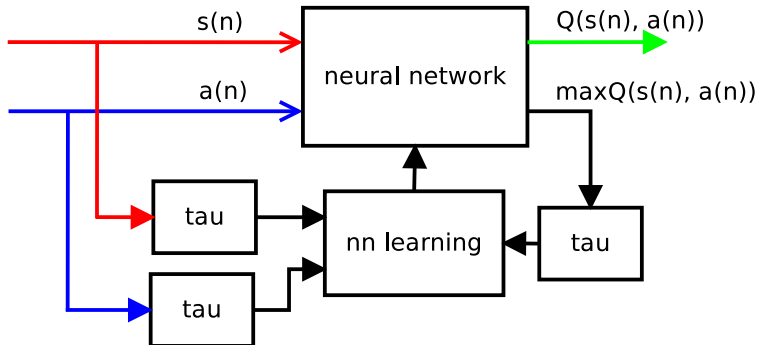
# Implementačné problémy

Problémy tabuľkovej interpretácie  $Q(s(n), a(n))$  :

- pre veľké  $n_s$  alebo  $n_a$  narastajú pamäťové nároky,
- o nevyplnených  $Q(s(n), a(n))$  nevieme povedať nič,
- pre rozsiahle stavové priestory ťažko vypočítateľné,
- ako aproximovať  $Q(s(n), a(n))$ ?

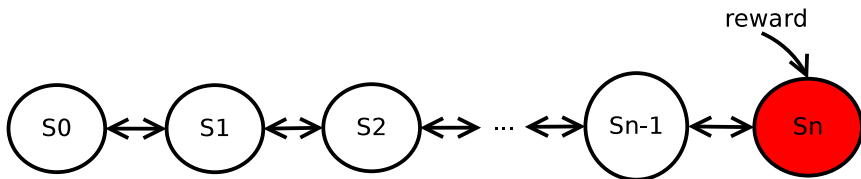
# Aproximácia neurónovou sieťou

Utopická predstava :



Prečo nedáva správne výsledky?

Na základe experimentov - Snowball problém



Pre korektné vyplnenie hodnôt v  $s_{n-1}$  sa vyžaduje korektná hodnota v  $s_n$

$$Q(s(1), a(1)) = R(s(1), a(1)) + \gamma \max_{a(0) \in \mathbb{A}} Q(s(0), a(0))$$

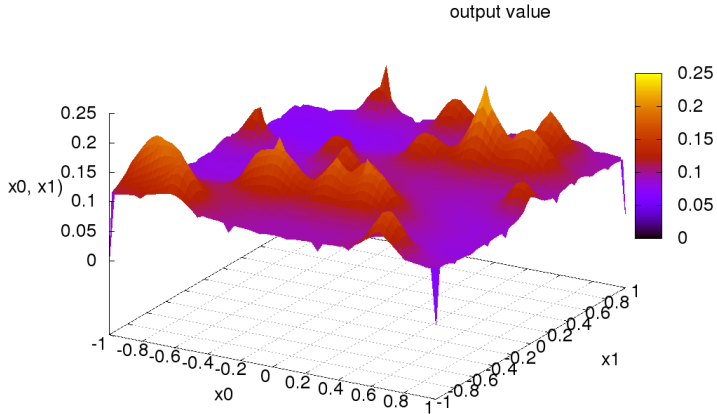
$$Q(s(2), a(2)) = R(s(2), a(2)) + \gamma \max_{a(1) \in \mathbb{A}} Q(s(1), a(1))$$

...

# Učenie doprednej siete

- Nie je homogénne!
- V priebehu učenia  $Q(s(n), a(n))$  chaoticky osciluje okolo požadovanej hodnoty.
- Ani po 10-mil. iteráciach sa hodnota neustáli na požadovanej hodnote.

# Je možné zostaviť neurónovú sieť, ktorá sa dá naučiť lokálne?





# Rozklad $Q(s(n), a(n))$ na bázické funkcie

$$f_j^1(s(n), a(n)) = e^{-\sum_{i=1}^{n_s} \beta_{aji}(n)(s_i(n) - \alpha_{aji}(n))^2}$$

$$f_j^2(s(n), a(n)) = \frac{1}{1 + \sum_{i=1}^{n_s} \beta_{aji}(n)(s_i(n) - \alpha_{aji}(n))^2}$$

$$f_j^3(s(n), a(n)) = e^{-\sum_{i=1}^{n_s} \beta_{aji}(n)|s_i(n) - \alpha_{aji}(n)|}$$

$$f_j^4(s(n), a(n)) = \sum_{k=1}^m \sum_{i=1}^{n_s} \beta_{aji}(n)(s_i(n) - \alpha_{aji}(n))^k$$

# Voľba bázičkých funkcií

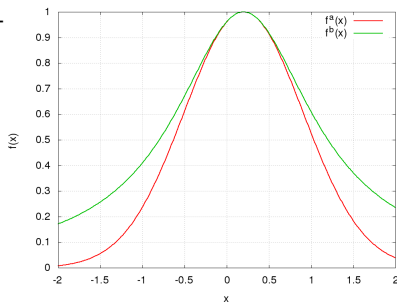
Vzhľadom na charakter učiaceho algoritmu

$$Q(s(n), a(n)) = R(s(n), a(n)) + \gamma \max_{a(n-1) \in \mathbb{A}} Q(s(n-1), a(n-1))$$

boli zvolené bázičné funkcie (parameter  $n$  pre prehľadnosť vynechaný)

$$f_j^1(s, a) = e^{-\sum_{i=1}^{n_s} \beta_{aji}(s_i - \alpha_{aji})^2}$$

$$f_j^2(s, a) = \frac{1}{1 + \sum_{i=1}^{n_s} \beta_{aji}(s_i - \alpha_{aji})^2}$$



Pre symetrické prechody medzi stavmi možno zjednodušiť na

$$f_j^1(s, a) = e^{-\beta_{aj} \sum_{i=1}^{n_s} (s_i - \alpha_{aji})^2}$$
$$f_j^2(s, a) = \frac{1}{1 + \beta_{aj} \sum_{i=1}^{n_s} (s_i - \alpha_{aji})^2}$$

a ich lineárna kombinácia

$$Q^x(s, a) = \sum_{j=1}^l w_{ja} f_j^x(s, a)$$

kde  $l$  je počet bázičných funkcií a  $x$  je voľba typu bázičkej funkcie.

# Aproximácia - nová bázičná funkcia

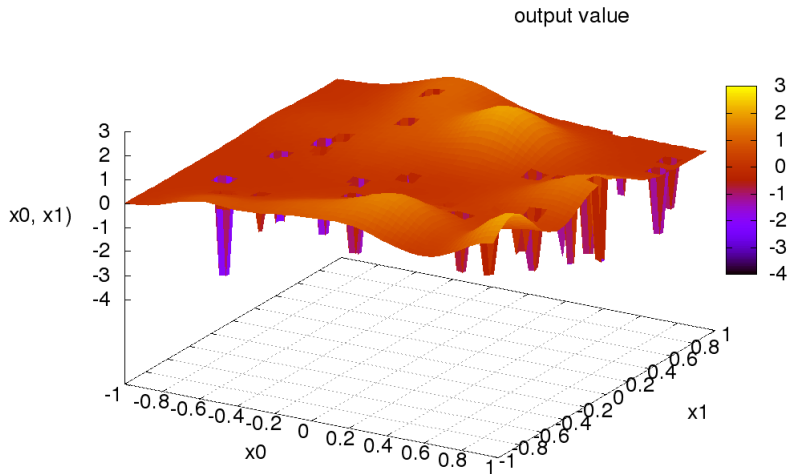
Tabuľka pre vybrané hodnoty - umožní zachytiť skokovú zmenu  
Gaussova krivka - dokáže pokryť nenulovými hodnotami celý  
definyčný obor

$$P_i(s(n), a(n)) = \begin{cases} r_{ai} & \text{if } s(n) = \alpha_i^1 \\ 0 & \text{inak} \end{cases} \quad (4)$$

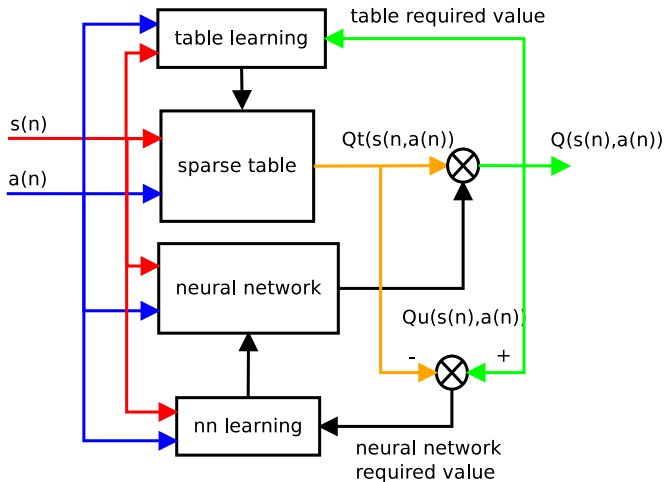
$$H_j(s(n), a(n)) = w_{aj} e^{-\beta_{aj} \sum_{i=1}^{n_s} (s_i(n) - \alpha_{aji}^2)^2} \quad (5)$$

$$Q(s(n), a(n)) = \sum_{i=1}^I P_i(s(n), a(n)) + \sum_{j=1}^J H_j(s(n), a(n)) \quad (6)$$

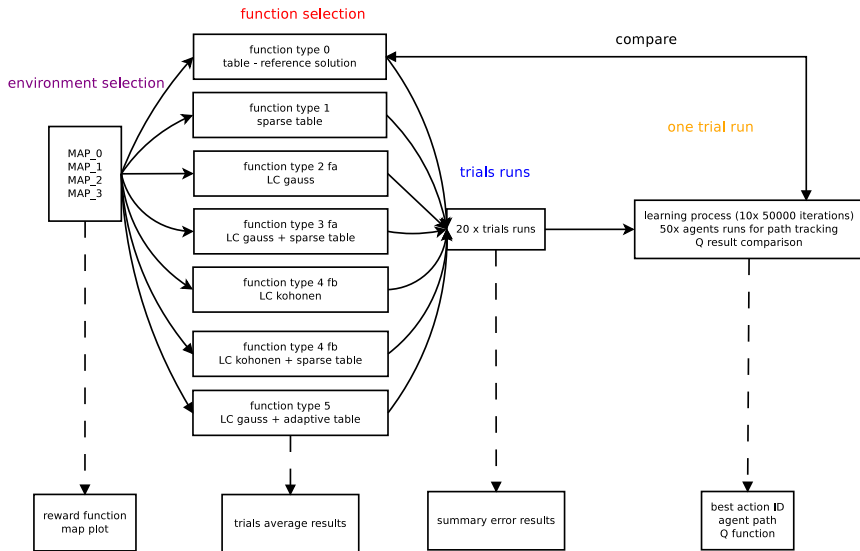
# Aproximácia - nová bážická funkcia



# Bloková schéma syntézy testovaného riešenia



# Schéma priebehu experimentov



# Návrh experimentov - podmienky

- 50000 iterácií učenia
- rozmer  $s$  je  $n_s = 2$ , rozmer  $a$  je  $n_a = 2$
- predpis funkcie ohodnotení

$$Q(s(n), a(n)) = \\ \alpha Q(s(n-1), a(n-1)) \\ (1 - \alpha)(R(s(n), a(n)) + \gamma \max_{a(n-1) \in \mathbb{A}} Q(s(n-1), a(n-1)))$$

- $R(s(n), a(n)) \in \langle -1, 1 \rangle$  náhodná mapa s 1 cieľovým stavom
- $\gamma = 0.98$  a  $\alpha = 0.7$
- hustota referenčného riešenia =  $1/32$  (4096 stavov)
- počet akcií v každom stave = 8
- hustota riedkej tabuľky =  $1/8$  (1:16 pomer)
- počet bazických funkcií  $l = 64$
- rozsah parametrov
  - $\alpha_{ja}(n) \in \langle -1, 1 \rangle$
  - $\beta_{ja}(n) \in \langle 0, 200 \rangle$
  - $w_{ja}(n) \in \langle -4, 4 \rangle$



# Návrh experimentov - podmienky

$Q_{rt}(s(n), a(n))$  referenčná funkcia  $Q$  (funkcia 0), kde  $t \in \langle 0, 19 \rangle$  je číslo trialu

$Q_{jt}(s(n), a(n))$  testované funkcie  $Q$  a  $j \in \langle 1, 5 \rangle$ .

Celková chyba behu trialu  $t$  je

$$e_{jt} = \sum_{s,a} (Q_{rt}(s, a) - Q_{jt}(s, a))^2$$

priemerná, minimálna, maximálna chyba a smerodatná odchylka

$$\bar{a}_j = \frac{1}{20} \sum_t e_{jt}$$

$$e_j^{\min} = \min_t e_{jt}$$

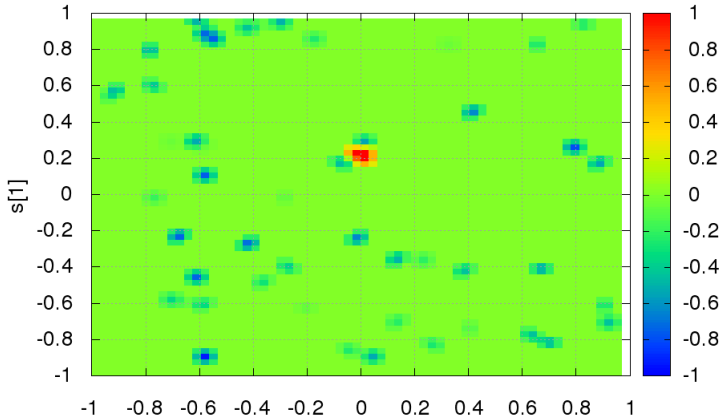
$$e_j^{\max} = \max_t e_{jt}$$

$$\sigma_j^2 = \frac{1}{20} \sum_t (\bar{a}_j - e_{jt})^2$$

# Funkcia $R(s, a)$ , mapa 1 - Výsledky experimentov

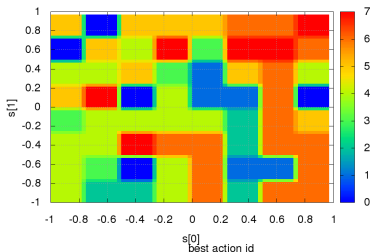
pre každý stav je zvolená rovnaká množina akcií.

Ďalej platí  $s = (s[0], s[1])$ .

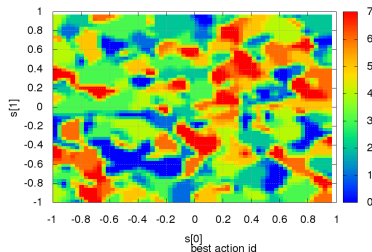


# Mapa najlepších akcií - Výsledky experimentov

Funkcia voľby najlepšej z 8 akcií v stave  $s = (s[0], s[1])$ .



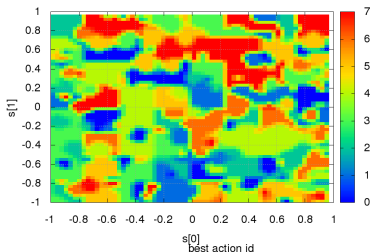
Obr. : sparse table



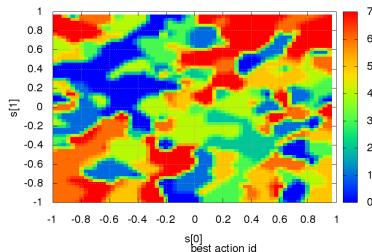
Obr. : linear combination Gauss

# Mapa najlepších akcií - Výsledky experimentov

Funkcia voľby najlepšej z 8 akcií v stave  $s = (s[0], s[1])$ .



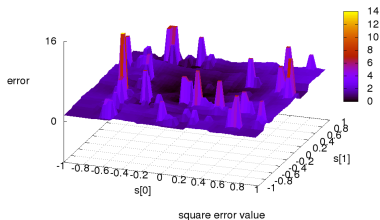
Obr. : sparse table + linear combination Gauss



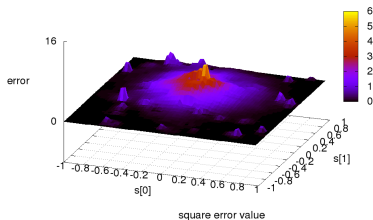
Obr. : linear combination Kohonen function

# Chybové funkcie - Výsledky experimentov

$$e_{jt}(s) = (Q_{rt}(s, a) - Q_{jt}(s, a))^2$$



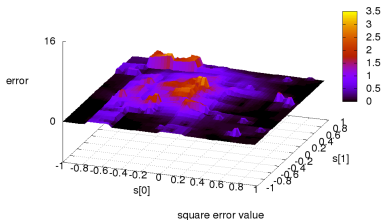
Obr. : sparse table



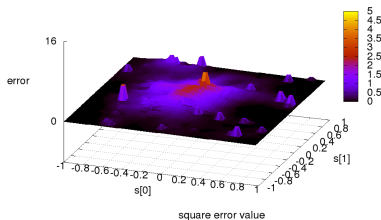
Obr. : linear combination Gauss

# Chybové funkcie - Výsledky experimentov

$$e_{jt}(s) = (Q_{rt}(s, a) - Q_{jt}(s, a))^2$$

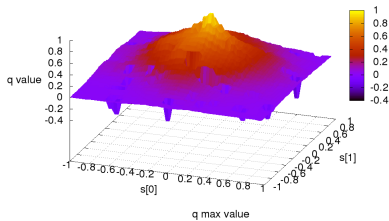


Obr. : sparse table + linear combination Gauss

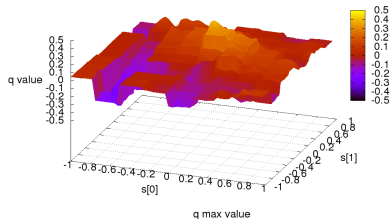


Obr. : linear combination Kohonen function

# max $Q(s, a)$ - Výsledky experimentov

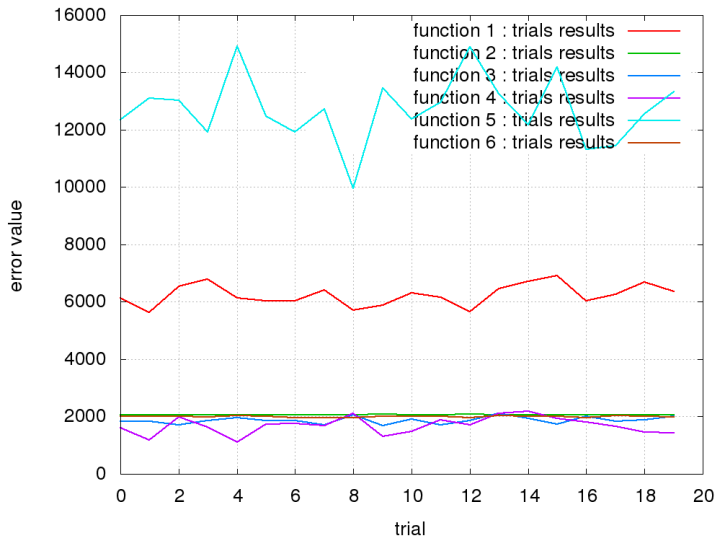


Obr. : reference table



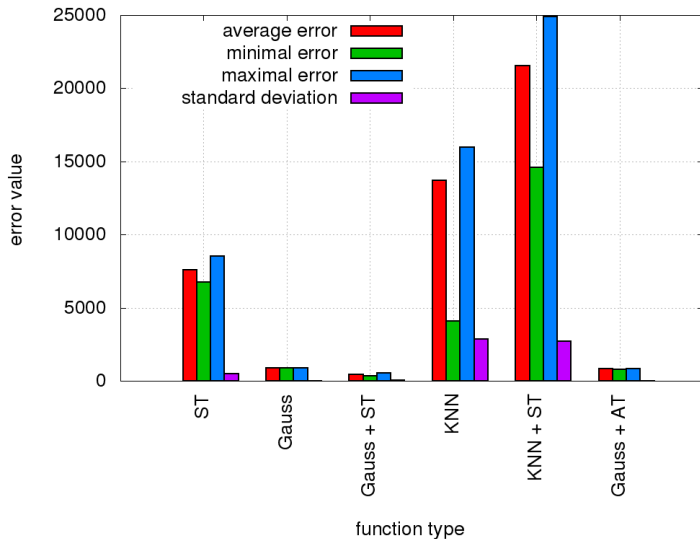
Obr. : sparse table + linear combination Gauss

# Priebeh trialov - Výsledky experimentov

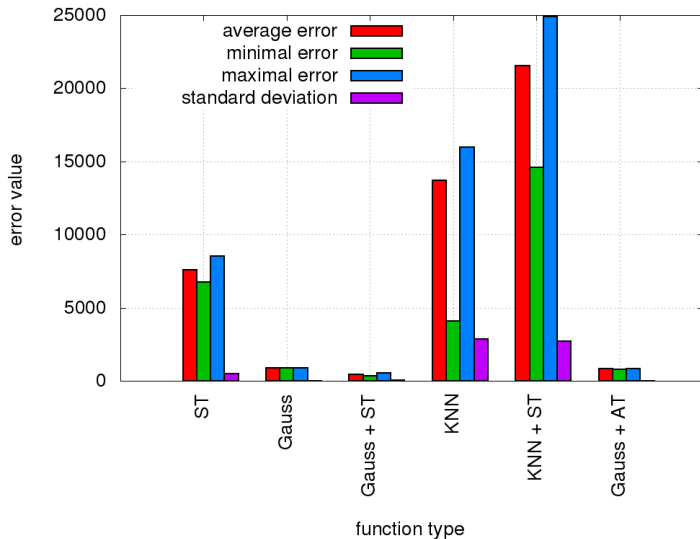




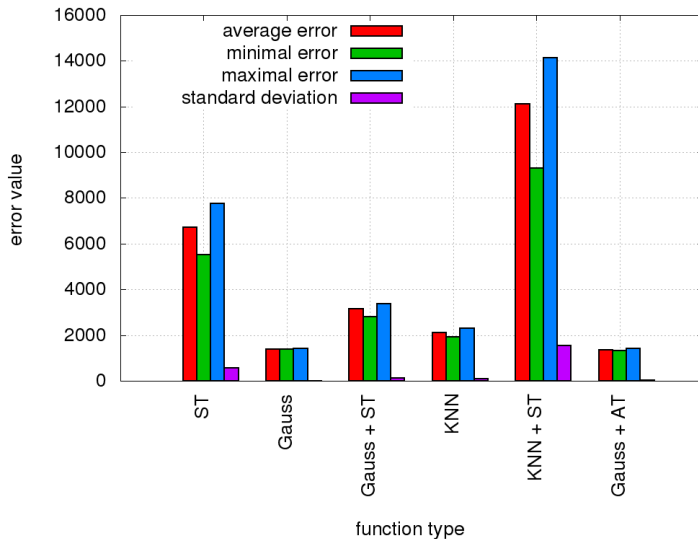
# Mapa 0 - Výsledky experimentov



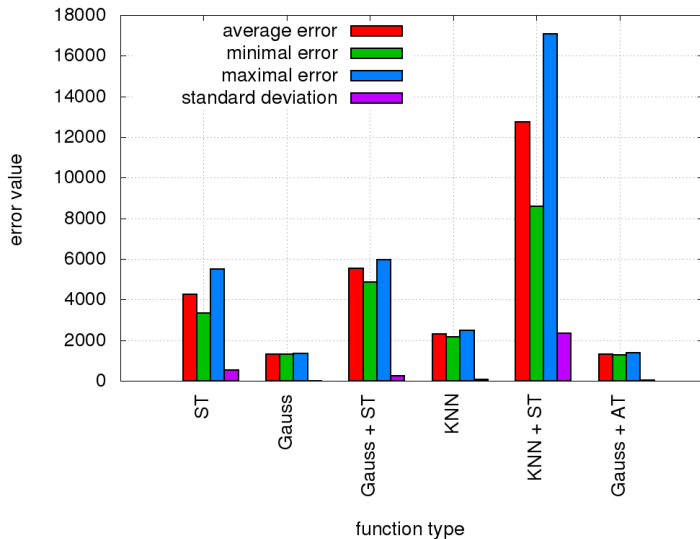
# Mapa 1 - Výsledky experimentov



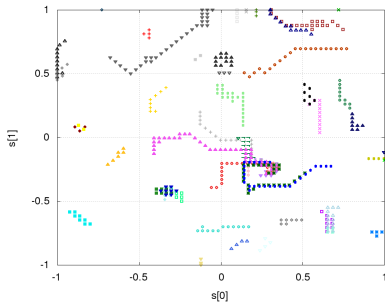
# Mapa 2 - Výsledky experimentov



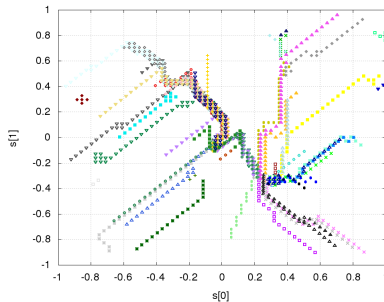
# Mapa 3 - Výsledky experimentov



# Porovnanie s ostatnými



Obr. : Dráha robotov, funkcia 2 - Gauss



Obr. : Dráha robotov, funkcia 6 - Peak and Hill

# Ďakujem za pozornosť

michal.chovanec@yandex.ru

[https://github.com/michalnand/q\\_learning](https://github.com/michalnand/q_learning)

