ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY

DIZERTAČNÁ PRÁCA

Študijný odbor: **Aplikovaná informatika**

Ing. Michal Chovanec

Aproximácia funkcie ohodnotení v algoritmoch Q-learning neurónovou sieťou

Vedúci: **prof. Ing. Juraj Miček, PhD**

Reg.č. xxx/2008 Máj 2012

Abstrakt

PRIEZVISKO MENO: *Názov diplomovej práce* [Diplomová práca]

Žilinská Univerzita v Žiline, Fakulta riadenia a informatiky, Katedra matematických metód.

Vedúci: doc. RNDr. Štefan Peško, CSc.

Stupeň odbornej kvalifikácie: Inžinier v odbore .... Žilina.

FRI ŽU v Žiline, 2012 — ?? s.

Obsahom práce je...

Abstract

PRIEZVISKO MENO: *Name of the Diploma thesis* [Diploma thesis]
University of Žilina, Faculty of Management Science and Informatics, Department of mathematical methods.
Tutor: Assoc. Prof. RNDr. Štefan Peško, CSc.
Qualification level: Engineer in field ..... Žilina:
FRI ŽU v Žiline, 2009 — ?? p.

The main idea of this ...

Prehlásenie

Prehlasujem, že som túto prácu napísal samostatne a že som uviedol všetky použité pramene a literatúru, z ktorých som čerpal.

V Žiline, dňa 15.5.2012 Meno Priezvisko

Obsah

Kapitola 1

Súčasný stav problematiky

Kapitola 2

Q-larning algoritmus

Q-learning algoritmus je definovaný pre časovo diskrétne systémy. Agent ktorý prechádza stavový priestor vykonaním niektorej z vopred daných akcií získava za tieto prechody odmeny. Cieľom algoritmu je ohodnotiť všetky akcie v jednotlivých stavoch, tak aby bol dosiahnutý ustálený stav a v každom stave bolo možno vybrať akciu prinášajúcu najväčšiu odmenu, v globálnom zmysle.

2.1 Definícia algoritmu

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 počty prvkov stavového vektora a vektora akcií.

Existuje prechodová funkcia

$$s(n+1) = \lambda(s(n), a(n)) \tag{2.1}$$

zo stavu $s(n) \in \mathbb{S}$ použitím akcie $a(n) \in \mathbb{A}$, táto funkcia je ale algoritmu neznáma.

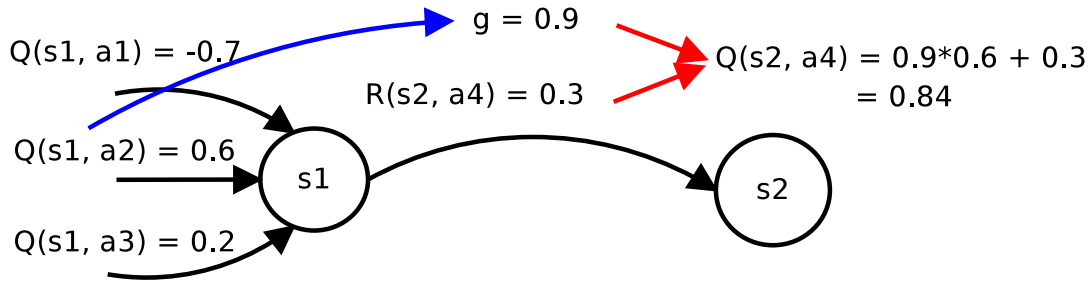
Ďalej je daná odmeňovacia funkcia $R(s(n), a(n))$, ktorá vyjadruje okamžité ohodnotenie konania agenta v $s(n)$ a $a(n)$. V reálnych aplikáciach táto funkcia nadobúda takmer v každom $s(n)$ a $a(n)$ hodnotu 0. Pre správnu funkciu algoritmu, musí byť aspoň jedna hodnota nenulová - napr. ohodnotenie dosiahnutia cieľového stavu (samotná existencia cieľového stavu však pre algoritmus nie je potrebná).

Funkcia ohodnotení je definovaná ako

$$Q_n(s(n),a(n)) = R(s(n),a(n)) + \gamma \max_{a(n-1)\in\mathbb{A}} Q_{n-1}(s(n-1),a(n-1)) \tag{2.2}$$

- $R(s(n),a(n))$ je odmeňovacia funkcia
- $Q_{n-1}(s(n-1),a(n-1))$ je funkcia ohodnotení v stave $s(n-1)$ pre akciu $a(n-1)$
- γ je odmeňovacia konštanta a platí $\gamma \in (0,1)$.

Funkcia 3.1 definuje ohodnotenie akcií vo všetkých stavoch t.j. agent ktorý sa dostal do stavu $s(n)$ vykonaním akcie $a(n)$ zo stavu $s(n-1)$ získal odmenu $R(s(n),a(n))$ a zlomok najväčšieho možného ohodnotenia ktoré mohol získať dostaním sa do stavu $s(n-1)$, situáciu ilustruje obrázok 2.1.

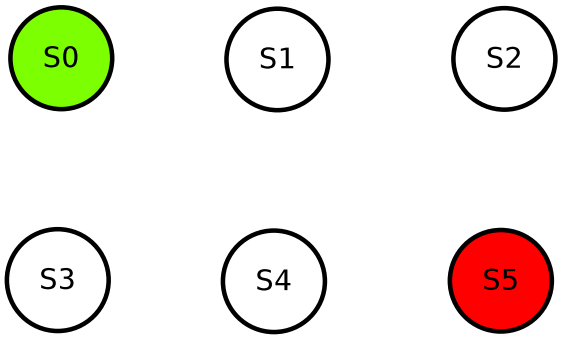


Obr. 2.1: Ilustrácia funkcie ohodnotení, pre $\gamma = 0.9$

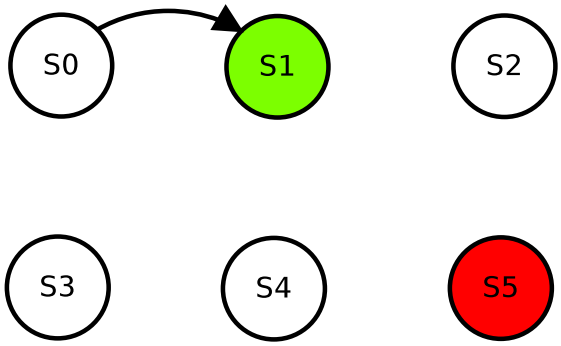
Nasledujúce obrázky ilustrujú beh algoritmu pre systém so 6 stavmi pre $\gamma = 0.8$. Na začiatku nie sú známe ani samotné prechody medzi stavmi (Obr. 2.2), bol definovaný 1 cieľový stav $S5$, agent začína v stave $S0$ (môže však v ľubovolnom inom). Ďalej sa pre jednoduchosť predpokladá že

$$R(s(n),a(n)) = \begin{cases} 1 & ak\ s(n) = \text{S5} \\ -0.5 & ak\ s(n) = \text{S4} \wedge \text{a(n)} = \text{Ay} \\ 0 & inak \end{cases} \tag{2.3}$$

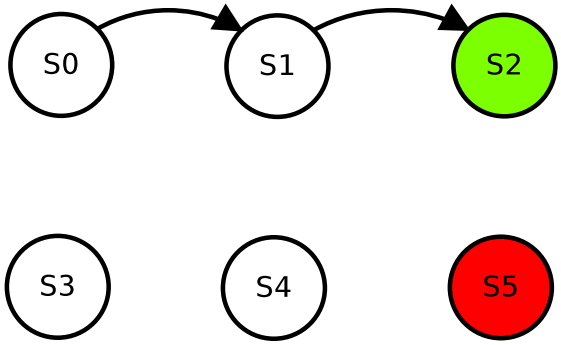
t.j. odmeňovacia funkcia nadobúda hodnotu 1 len ak sa agent dostal do stavu $S5$ a pre ilustráciu je definovaná aj jedna záporná odmena pri prechode z $S4$ do $S3$ akciou Ay.



Obr. 2.2: Inicializácia



Obr. 2.3: Prechod do stavu S1

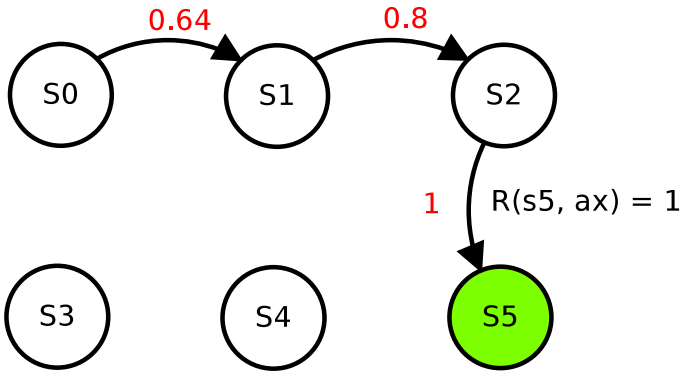


Obr. 2.4: Prechod do stavu S2

Agent v každom stave náhodne vyberá akcie (na výbere nezáleží, dôležité je aby každá akcia mala nenulovú pravdepodobnoť výberu, a rovnako bola nenulová prevdepodobnosť dosiahnutia ľubovolného stavu). Obrázky Obr. 2.3 a Obr. 2.4 ilustrujú jednu z možných ciest.

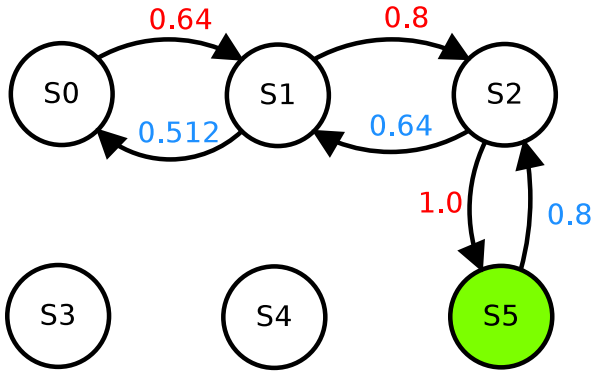
Po dosiahnutí cieľového stavu Obr. 2.5 je na základe 2.3 možné spočítať podľa 3.1 ohodnotenia doteraz vykonaných akcií - agent získal nenulovú odmenu $R(s_5, a_x) = 1$ (kde a_x značí ľubovolnú akciu), ktorú rekurentne spočíta pre všetky doteraz vykonané akcie.

Pre zjednodušenú variantu 3.1 by bolo možné pamätať si len jeden predošlý stav a nepostupovať v ohodnocovaní rekuretne. V praktickej aplikácií je potrebné obmedziť hĺbku rekurzie, a pamätať si len posledných P stavov a vnich urobených rozhodnutiach. V tomto jedńoduchom príklade však nie sú nutné tieto obmedzenia, je teda možné pamätať si celú cestu.



Obr. 2.5: Prechod do stavu S3

Agent môže pokračovať v ceste ďalej, napr. späť 2.6 a pribežne počítať ohodnotenia. Prechod z s_5 do s_2 je ohodnotený ako 0.8 - vybralo sa najlepšie možné ohodnotenie ako sa dostať do s_5 (1) násobené γ, $R(s_2, a_x) = 0$ (podľa 2.3).

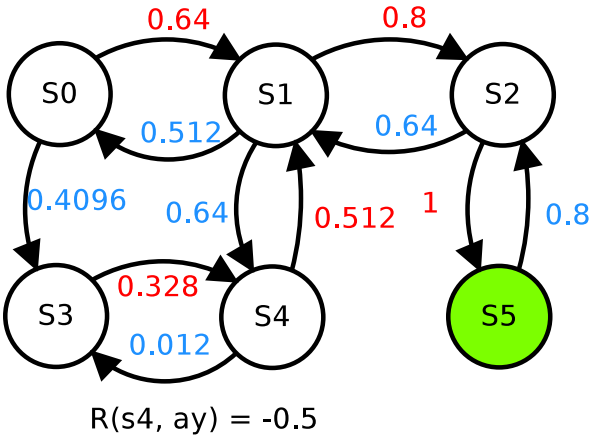


Obr. 2.6: Ďalšie prechody agenta

Po prejdení celého grafu, kedy agent vykonal všetky možné akcie dosiahne funkcia $Q(s(n), a(n))$ konečný, ustálený stav Obr. 2.7, teda

$$\forall s(n),\ \forall a(n), \forall \varepsilon > 0\ \exists Q_n :|\ Q_n(s(n), a(n)) - Q_{n-1}(s(n), a(n))\ | < \varepsilon \tag{2.4}$$

hodnoty Q-funkcie sa teda pri pevne danom $R(s(a),a(n))$ už nemenia.



Obr. 2.7: Konečný stav

2.2 Výber akcie

Pre nájdenie konečných hodnôt funkcie ohodnotení podľa 2.4 stačí aby každý prechod mal nenulovú pravdepodobnosť vykonania. Pre ďalšie vyšetrovanie konania agenta ja daná pravdepodobnosť výberu akcie ako

$$P(s(n),a(s)) = \frac{e^{kQ(s(n),a(s))}}{\sum_{i=1}^{C_a} e^{kQ(s(n),a(i))}} \tag{2.5}$$

kde

s je zvolená akcia

C_a je počet akcií

k je konštanta a platí $k \geq 0$

agent ktorý vyberá všetky akcie s rovnakou pravdepodobnosťou má teda $k = 0$. Pre vysoké hodnoty k : $\lim_{k\to\infty}$ bude agent vyberať len najlepšie dostupné akcie. Pre učenie agenta je teda vhodné zvoliť malé k.

Je možné definovať ľubovolné iné možnosti výberu akcie, napr. uprednostňovať menej často vykonané akcie, prípadne podľa zmeny $\mid Q_n(s(n),a(n)) - Q_{n-1}(s(n),a(n)) \mid$ uprednostňovať prechody s veľkou hodnotou zmeny.

2.3 Problémy výpočtu $Q(s,a)$

Algoritmus je definovaný pre diskrétnu množinu stavov. Ďalej sa predpokladá, že $s(n) \in \langle -1,1 \rangle$ a podobne $a(n) \in \langle -1,1 \rangle$

Pre počty prvkov stavového vektora a vektora akcií (n_s, n_a) je možné definovať delenie ich hodnôt na diskrétny počet d_s a d_a, potom je možné vyjadriť celkový počet hodnôt $Q(s(n),a(n))$ ako

$$C = {d_s}^{n_s} {d_a}^{n_a} \tag{2.6}$$

Samotný počet hodnôt ktoré treba spočítať teda exponenciálne narastná z rastom počtu prvkov stvového a vektora akcií.

Pre úlohy kde do systému vstupuje mnoho nezávislých vstupov sa stáva implentácia Q(s(n), a(n)) problémom najmä z dôvodov :

- veľké pamäťové nároky
- o nenavštívených prechodoch nevie agent povedať nič

Vhodným riešením sa ukazuje aproximácia Q-funkcie. Nech je aproximovaná funkcia označená $Q'_n(s(n),a(n))$ a presné riešenie ako $Q_n(s(n),a(n))$.

Dané sú postuláty o tejto aproximácií

Postulát 1 *Neobmedzená prenosť aproximácie : Pre všetky stavy $s(n)$ a akcie $a(n)$ musí platiť' $\mid Q_n(s(n),a(n)) - Q'_n(s(n),a(n)) \mid < \varepsilon$. Kde $\varepsilon > 0$ a určuje kvalitu aproximácie. Zlepšením vlastností $Q'_n(s(n),a(n))$ je možné l'ubovolne zmenšovať ε.*

Postulát 2 *Lokálna zmena : Lokálna zmena hodnoty $\delta = \mid Q'_n(s(n),a(n)) - Q'_{n-1}(s(n),a(n)) \mid$ neovplyvní hodnotu funkcie v inom bode o viac ako $\forall\, s(n') \; \forall a(n'),\; n \neq n' : \delta < \kappa$. Zmenšovaním hdonoty κ sa funkcia stáva menej závislá na okolí bodu $[s(n),a(n)]$.*

Funkciu $Q(s(n),a(n))$ je možné aproximovať niekoľkými spôsobmi. Tie najbežnejšie sú

- tabuľka

- neurónová sieť
  - dopredná neurónová sieť
  - kohonenova mapa
  - neurónová sieť bázickych funkcií

2.4 Tabuľka

Dané sú celočíselné indexy

$$I_s(n) = \lceil \sum_{i=1}^{n_s} \left(d_s \frac{s_i(n)+1}{2} \right)^i \rceil \tag{2.7}$$

$$I_a(n) = \lceil \sum_{i=1}^{n_a} \left(d_a \frac{a_i(n)+1}{2} \right)^i \rceil \tag{2.8}$$

kde

$I_s(n)$ je index stavu

$I_a(n)$ je index akcie

$s_i(n)$ je i-ty prvok vektora stavu $s(n)$

$a_i(n)$ je i-ty prvok vektora akcií $a(n)$

Pre diskrétny počet stavov a akcií je možné definovať tabuľkovú interpretáciu ako $Q^t(I_s(n), I_a(n))$. Pre $\lim_{d_s \to \infty}$ a $\lim_{d_a \to \infty}$ je možné považovať tabuľku za presné riešenie pretože spĺňa postuláty 1 aj 2.

2.5 Dopredná neurónová sieť

Pre aproximáciu funkcie ohdnotení je možné použiť dobrednú neurónovú sieť ako univerzálny aproximátor.

Je daný vstupný vektor

$$I(n) = (s(n), a(n)) \tag{2.9}$$

výstupná hodnota neurónovej siete ako $y_{nn}(n)$ a požadovaná hodnota ako $y_r(n)$.

Ďalej je definovaná chyba ako

$$e(n) = y_r(n) - y_{nn}(n) \tag{2.10}$$

Vrstva l doprednej siete je definovaná ako

$$\begin{aligned} y^l(n) &= f^l\left(W^l(n)I^l(n)\right) \\ &= f^l\left(\begin{pmatrix} w^l_{1,1}(n) & w^l_{1,2}(n) & \cdots & w^l_{1,n'}(n) \\ w^l_{2,1}(n) & w^l_{2,2}(n) & \cdots & w^l_{2,n'}(n) \\ \vdots & \vdots & \ddots & \vdots \\ w^l_{m',1}(n) & w^l_{m',2}(n) & \cdots & w^l_{m',n'}(n) \end{pmatrix} \begin{pmatrix} i^l_{1,1}(n) & i^l_{1,2}(n) & \cdots & i^l_{1,n'}(n) \end{pmatrix}\right) \end{aligned} \tag{2.11}$$

kde

n' je počet prvkov vstupného vektora

m' je počet prvkov výstupného vektora

$f(X)$ je aktivačná funkcia

$W^l(n)$ je matica váh.

Najčastejšie používané aktivačné funkcie sú sigmoida, hyberbolický tangens , lineárna, usmerňovač a skoková funkcia. Ich predpisy sú

$$y_1(x) = \frac{1}{1+e^{-x}} \tag{2.12}$$

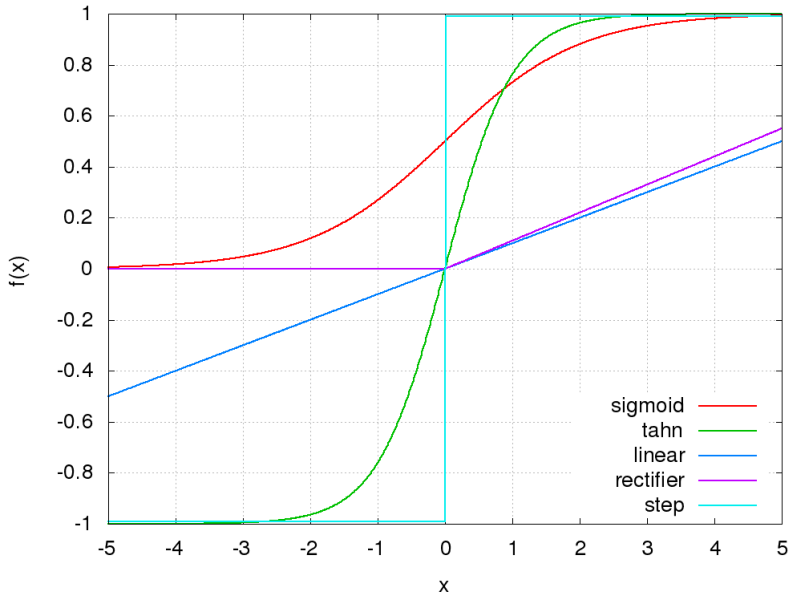
$$y_2(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.13}$$

$$y_3(x) = x \tag{2.14}$$

$$y_3(x) = \begin{cases} x & ak\ x > 0 \\ 0 & inak \end{cases} \tag{2.15}$$

$$y_4(x) = \begin{cases} 1 & ak\ x > 0 \\ -1 & inak \end{cases} \tag{2.16}$$

Ich priebehy sú znázornené na obrázku Obr. 2.8.



Obr. 2.8: Grafické znázornenie priebehov aktivačných funkcií

Zoradením niekoľkých vrstiev za sebou, tak že výstup predošlej je vstupom do aktuálnej vrstvy je možné získať doprednú neurónovu sieť. Takáto sieť je vhodná na riešenie klasifikačných aj aproximačných problémov.

Najväčším nedostatkom je mechanizmus učenia.

Existuje niekoľko spôsobov ako túto sieť učiť, najčastejšie sú

- gradientová metóda minimalizácie chyby

- simulované žíhanie

Gradientová metóda nemá zaručené dosiahnutie globálneho minima pre všeobecné prípady, simulované žíhanie je časovo náročné pre veľký počet váhových parametrov.

Dopredná sieť je známa nelokálnosťou učenia : pri trénovaní na podmnožinu množiny požadovaných výstupov sa mení hodnota aj mimo túto podnmožinu. Sieť teda veľmi problematicky spĺňa postutlát 2. Dobre však spĺňa postulát 1, vhodnou voľbou počtu vrstiev a počtu neurónov je možné dosiahnúť ľubovolnú presnosť aproximácie.

2.6 Kohonenová neurónová sieť

Pre aproximáciu môže byť použitá Kohonenová neurónová sieť. Každému neurónu tejto siete je priradená jedna hodnota.

Najpr sa spočítajú vzdialenosti od vstupného vektora

$$d_j(n) = \sum_{i=1}^{N} (I_i(n) - w_{ji}(n))^2 \tag{2.17}$$

kde $w(n) \in \mathbb{R}$ je matica váh, na začiatku sa volí náhodná. Víťazný neurón v je definovaný ako

$$v : \forall j : d_v(n) \leq d_j(n) \tag{2.18}$$

A pre každý neurón existuje priradená výstupná hodnota $y_j(n) \in \mathbb{R}$, výstupom siete je hodnota priradená výťaznému neurónu $y_{nn}(n) = y_v(n)$

Učenie siete prebieha v dvoch krokoch

1) zmena váh $w(n)$ - zmenia sa váhy výťazného neurónu, pretože najlepšie zopovedajú požadovaným váham

$$w_{ji}(n+1) = (1 - \eta_1(j))w_{ji}(n) + \eta_1 I_i(n) \tag{2.19}$$

kde $\eta_1(j) \in (0,1)$ je krok učenia a závisí od polohy neurónu v sieti. V najjednoduhšom prípade

$$\eta_1(j) = \begin{cases} \eta & ak\ j = v \\ 0 & inak \end{cases} \tag{2.20}$$

k zmene váh teda dôjde len pri víťaznom neuróne. Ďalší často používaný tvar funkcie postupne zmenšuje hodnotu $\eta_1(j)$ podľa $d_j(n)$ a to ako

$$\eta_1(j,n) = \eta e^{-kd_j(n)} \tag{2.21}$$

kde $k \in (0,\infty)$. Krok učenia $\eta_1(j,n)$ je teda premenný a záleží aj od predloženého vzoru podľa n.

Po dostatočnom počte iterácií sa hodnoty váh ustália na hodnotách tak aby rozdelili množinu vstupných vektorov na lokálne oblasti. Tento stav je znázornený na Obr. 2.9. Vstupný proces generoval 8 zhlukov dát ktoré sieť klasifikovala použitím 16 neurónov. Každému neurónu je možne priradiť požadovanú hodnotu výstupu.

2) upraví sa výstupná hodnota $y_v(n)$

$$y_v(n+1) = y_v(n) + \eta_2 e(n) \tag{2.22}$$

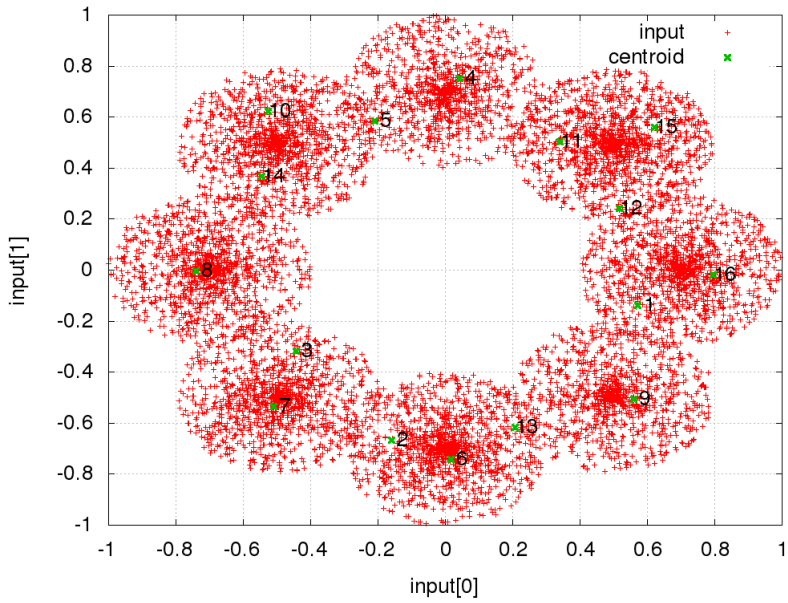
kde $e(n)$ je chyba podľa 2.10.

2.7 Neurónová sieť bázickych funkcií

Dané sú bázické funkcie $f_j^x(s(n),a(n))$, kde x je typ bázickej funkcie. Požadovaná hodnota $Q^x(s(n),a(n))$ je potom lineárnou kombináciou týchto funkcií typu x.

Z charakteru Q-learning algoritmu 3.1 je možné určiť požiadavky na tieto funkcie :

1. predpis 3.1 je tvorený klesajúcou exponenciálou - podobný charakter by mala mať aj bázická funkcia



Obr. 2.9: Znázornenie váhových parametrov w pre dvojrozmerný priestor

2. existencia jedného globálneho maxima a zmenou parametrov určovať polohu tohto bodu

3. možnosť ľubovolne meniť strmosť funkcie v okolí maxima

4. funkcia by mala byť zhora aj z dola ohraničená

Cieľom je mať možnosť nezávisle nastaviť maximá funkcií do oblastí, ktoré zodpovedajú nenulovím hodnotám $R(s(n),a(n))$ - bod 2. Ak ohodnotenie spĺňa podmienku najlepšej možnej akcie v danom stave, dá sa očakávať že bude mať menšiu strmosť, naopak, ak funkcia popisuje bod kde $R(s(n),a(n))$ dosahuje malé hodnoty (obvykle záporné), bude požadovaná vysoká strmosť tejto funkcie - obe požiadavky sú zhrnuté v bode 3. Bod 4 umožňuje rozumne ohraničiť rozsah funkcie.

Niektoré tvary bázických funkcií

$$f_j^1(s(n),a(n)) = e^{-\sum_{i=1}^{n_s} \beta_{aji}(n)(s_i(n)-\alpha_{aji}(n))^2} \tag{2.23}$$

$$f_j^2(s(n),a(n)) = \frac{1}{1+\sum_{i=1}^{n_s} \beta_{aji}(n)(s_i(n)-\alpha_{aji}(n))^2} \tag{2.24}$$

$$f_j^3(s(n),a(n)) = e^{-\sum_{i=1}^{n_s} \beta_{aji}(n)|s_i(n)-\alpha_{aji}(n)|} \tag{2.25}$$

kde

$\alpha_{aji}(n) \in \langle -1,1 \rangle$ určuje polohu maxima funkcie

$\beta_{aji}(n) \in (0,\infty)$ určuje strmosť funkcie.

Pre symetrické prechody medzi stavmi ich možno zjednodušiť na

$$f_j^1(s(n),a(n)) = e^{-\beta_{aj}\sum_{i=1}^{n_s} (s_i(n)-\alpha_{aji})^2} \tag{2.26}$$

$$f_j^2(s(n),a(n)) = \frac{1}{1+\beta_{aj}\sum_{i=1}^{n_s} (s_i(n)-\alpha_{aji})^2} \tag{2.27}$$

$$f_j^3((n)s,a(n)) = e^{-\beta_{aj}\sum_{i=1}^{n_s} |s_i(n)-\alpha_{aji}(n)|} \tag{2.28}$$

Aproximovaná funkcia ohodnotení pre l bázických funkcií je potom

$$Q^x(s(n),a(n)) = \sum_{j=1}^{l} w(n)_j^x f_j^x(s(n),a(n)) \tag{2.29}$$

kde $w(n)_j^x$ sú váhy bázických funkcií.

Je teda potrebné stanoviť celkovo 3 sady parametrov : α β w.

2.7.1 Určenie parametrov α

Parameter α určuje posunutie maxima funkcie a postupuje sa podobne ako v prípade 2.19. Treba zohľadniť fakt, že pre konečný výsledok je dôležité pokryť všetky oblasti s nenulovým $R(s(n),a(n))$, vrchol krivky bude ležať nad nad bodom $[s(n),a(n)]$.

Zmena parametrov α prebieha v piatich krokoch.

- na začiatku sa zvolia $\alpha_{jia}(n)$ náhodne, ze $\langle -1, 1 \rangle$
- spočítajú sa vzdialenosti od predloženého vstupu $d_{ja}(n) = \mid s(n) - \alpha_{ja}(n) \mid$
- nájde sa také ka kde pre $\forall j : d_{ka}(n) \leq d_{ja}(n)$
- spočíta sa krok učenia $\eta'_a(n) = \eta_1 \mid Q_r(s(n), a(n)) \mid$
- upravia sa parametre $\alpha_{aki}(n+1) = (1-\eta')\alpha_{aki}(n) + \eta' s_i(n)$

kde
$Q_r(s(n), a(n))$ je požadovaný výstup
η_1 je konštanta učenia

Krok učenia teda závisí od veľkosti požadovanej hodnoty, tým sa zabezpečí aby maximum krivky naozaj ležalo nad bodom $[s(n), a(n)]$.

2.7.2 Určenie parametrov β

Parameter β určuje strmosť krivky. Ak boli k dizpozicií naraz všetky požadované výstupy, bolo by možné spočítat tento parameter z rozptylu. Požadované hodnoty však prichádzajú postupne, strmosť krivky sa preto upravuje preibežne, podľa toho či požadovaná hodnota leží nad, alebo pod krivkou.

- stanoví sa chyba $e(n) = Q_r(s(n), a(n)) - Q(s(n), a(n))$
- pre každú bázickú funkciu $\beta_{ja}(n+1) = \beta_{ja}(n) + \eta_2 e(n) w_{ja}(n)$
- skontroluje sa $\beta_{ja}(n) \in (0, \infty)$

kde
$Q_r(s(n), a(n))$ je požadovaný výstup
η_2 je konštanta učenia

2.7.3 Určenie váhových parametrov w

Nakoniec sa gradientovou metódou určia váhové paramete. Pre presné riešenie by bolo možné použiť metódu nejmenších štvorcov, tá je však pre veľký počet bázcikých funkcií ťažko vypočítateľná. Zmena parametrov je potom daná nasledujúcim postupom

- stanoví sa chyba $e(n) = Q_r(s(n), a(n)) - Q(s(n), a(n))$
- pre každé $w_{ja} : w_{ja}(n+1) = w_{ja}(n) + \eta_3 e(n) y_j(n)$
- skontroluje sa $w_{ja}(n) \in (-r, r)$

kde

η_3 je konštanta učenia

r je maximálny rozsah váh

$$H_j(s(n)) = \begin{cases} r_j & if\ s(n) = \alpha_j \\ 0 & inak \end{cases} \tag{2.30}$$

$$f_j(s(n)) = H_j(s(n)) + w_j e^{-\sum_{i=1}^{n_s} \beta_{ji}(s_i(n)-\alpha_{ji})^2} \tag{2.31}$$

$$Q(s(n)) = \sum_{j=1}^{J} f_j(s(n)) \tag{2.32}$$

kde

α_j je stav pre ktorý sa počíta funkcia

r_j je hodnota okamžitej odmeny $R(s(n))$ v tomto stave

β_j je strmosť, a platí $\beta > 0$

w_j je váha

Kapitola 3

Experimentálna časť

3.1 Ciele práce

V oblasti Q-learning algoritmoch je možné pozorovať dva hlavné smery výskumu

- aproximácia funkcie ohodnotení
- spôsob výberu akcie

Obe majú široké pole diskusií v snahe vyriešit niekoľko hlavných problém Q-learning algoritmu a to najmä

- veľký počet prechodov medzi stavmi
- malá zmena vo výpočte $Q(s(n), a(n))$ môže spôsobiť veľké zmeny v stratégií.

Cieľom práce je na danej množine odmeňovacích funkcií $R(s(n), a(n))$ overiť možnosti aproximácie $Q(s(n), a(n))$. V niekoľkých bodoch je možné postup určiť ako

- výber funkcií $R(s(n), a(n))$
- určenie presného riešenia, použitím tabuľky s veľkým počtom prvkov
- voľba aproximačnej metódy
- pre každú $R(s(n), a(n))$ spočítať niekoľko nezávislych behov

- výsledky porovnať s presným riešením, overiť a zosumarizovať

Funkcie $R(s(n),a(n))$ budu vybrané tak aby boli riedke a plne sa využil Q-learing - okamžité ohodnotenie je známe len v malom počte prípadov. Postupne sa obmenia pre rôzne počty nenulových prvkov.

Presné riešenie, aby bolo možné spočítať bude mať niekoľko tisíc diskrétnych stavov. Pre jednoduchosť, bude v každom stave rovnaká a presne definovaná množina akcií.

Vyberie sa niekoľko aproximačných metód, ktoré sa použijí na spočítanie $Q(s(n),a(n))$. Tu je nevyhnutné upozorniť na častú metodickú chybu : aj keď je možné $Q(s(n),a(n))$ spočítať presne, nesmie byť toto presne riešenie použité na stanovenie približného riešenia. Príkladom je dopredná neurónová sieť, ktorá sa dá veľmi ľahko natrénovať ak je množina požadovaných výstupov vopred známa. V prípade Q-learning algoritmu sa ale požadované hodnoty spočítavajú rekuretne, až počas behu.

Kedže voľba niektorých počiatočných parametrov aproximačných metód je náhodná, je nevyhnutné spočítať niekoľko nezávislých behov a overť tak rozptyl, minimálnu, maximálnu a priemernu chybu.

3.2 Návrh experimentu

Aby sa dalo kvalitatívne ohodnotiť použité riešenie, je nutné urobiť veľký počet experimentov. Aby bolo možné ľahko graficky znázorniť výsledok, bude stavový priestor dvojrozmerný a platí $s(n) \in \langle -1,1 \rangle$. Agent si bude vyberať z pevne danej množiny akcií a bude sa tak v tomto priestore môcť pohybovať a to :

$\mathbb{A} = [[0,1],[0,-1],[1,0],[-1,0],[1,-1],[1,1],[-1,-1],[-1,1]]$

prostredie umožní zmenu stavu vykonaním akcie $a(n) \in \mathbb{A}$, a to podľa

$$s(n+1) = s(n) + a(n)dt \tag{3.1}$$

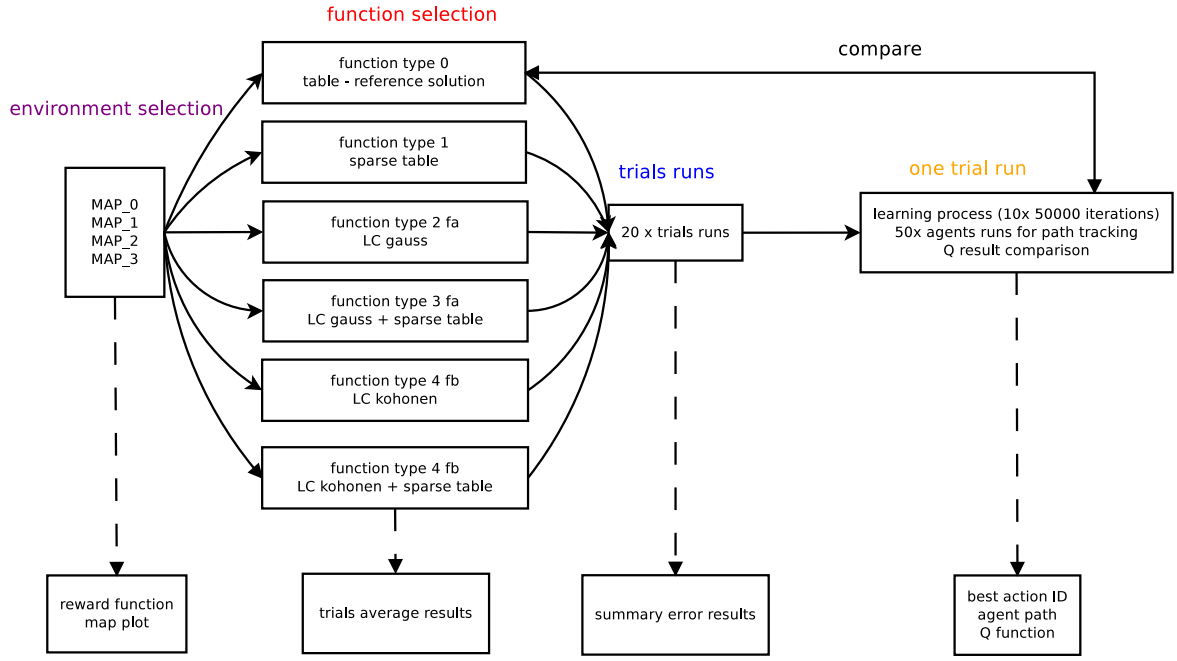
Jednotlivé funkcie $R^k(s(n),a(n))$ predstavujú mapy odmien v ktorých sa agent pohybuje. Pre zjednodušenie bude platiť, že nezáleží ktorou akciou sa agent dostal do daného stavu -

funkcia bude mať teda tvar $R^k(s(n))$ a predstavuje teda odmenu za to, že sa agent dostal na nejaké miesto.

Ako metódy aprximácie je zvolených 5 rôznych funkcií.

1. riedka tabuľka
2. Gaussova krivka $f_j^1(s(n),a(n))$ 2.28
3. Gaussova krivka $f_j^1(s(n),a(n))$ kombinovaná s riedkou tabuľkou
4. Modifikácia Kohonenovej neurónovej siete $f_j^2(s(n),a(n))$
5. Modifikácia Kohonenovej neurónovej siete $f_j^2(s(n),a(n))$ s riedkou tabuľkou

Pre každú z nich prebehne 20 trialov aby bolo možné urobiť štatistické vyhodnotenie. V každom trialy prebehne 10*50000 učiacich interácií aby bolo možné v 10 tich krokoch sledovať priebeh učenia. Na konci prebehne 50 behov agentov z náhodných východzich stavov aby bolo možné sledovať ich cestu stavovým priestorom.



Obr. 3.1: Schéma experimentu

Súhrnná schéma behu experimentov je na obrázku 3.1. Plné šípky predstavujú prepojenie úrovni metodológie. Čiarkované šípky znázorňujú výstupy v jednotlivých úrovniach. Presné riešenie je použité na porovnanie výslednej chyby.

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

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

- $R(s(n),a(n)) \in \langle -1,1\rangle$ 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 bázických funkcií $l = 64$
- rozsah parametrov
  - $\alpha_{ja}(n) \in \langle -1,1\rangle$
  - $\beta_{ja}(n) \in \langle 0,200\rangle$
  - $w_{ja}(n) \in \langle -4,4\rangle$

$Q_{rt}(s(n),a(n))$ 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

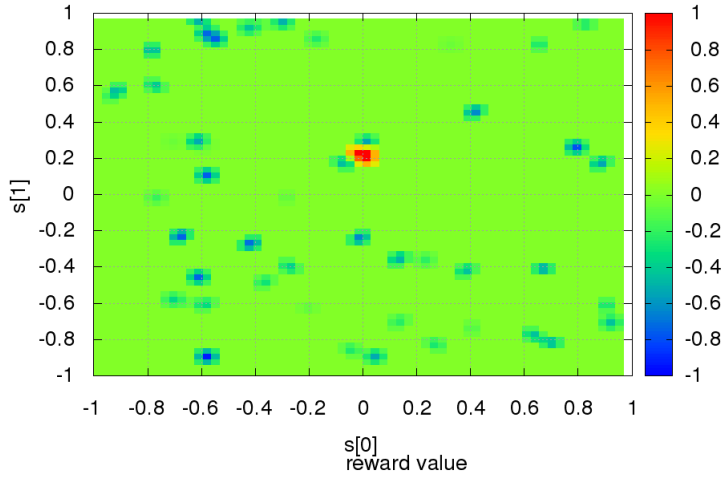
$$\bar{a_j} = \frac{1}{20} \sum_t e_{jt}$$

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

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

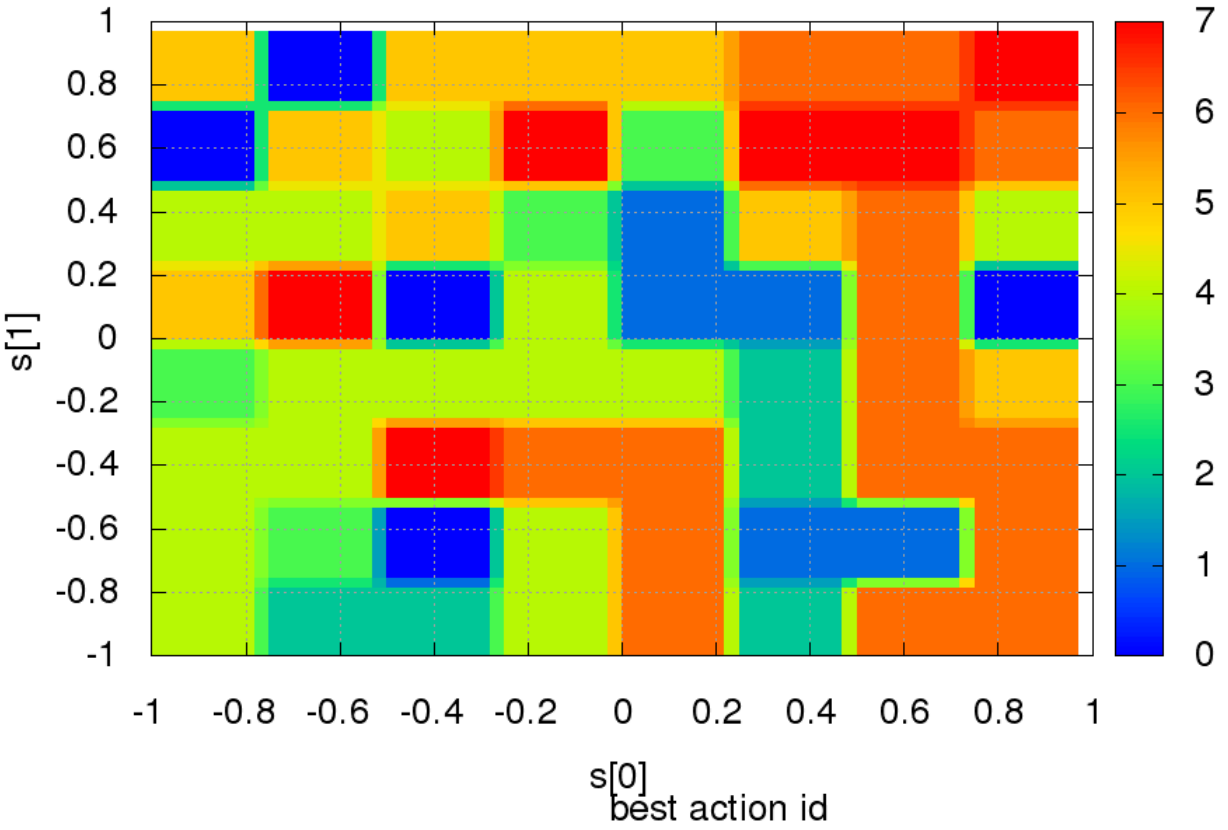
$$\sigma_j{}^2 = \frac{1}{20} \sum_t (\bar{a_j} - e_{jt})^2$$

3.3 Výsledky experimentu



Obr. 3.2: odmeňovacia funkcia

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



Obr. 3.3: fig:sparse table

Chybové funkcie - Výsledky experimentov

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

max Q(s, a) - Výsledky experimentov

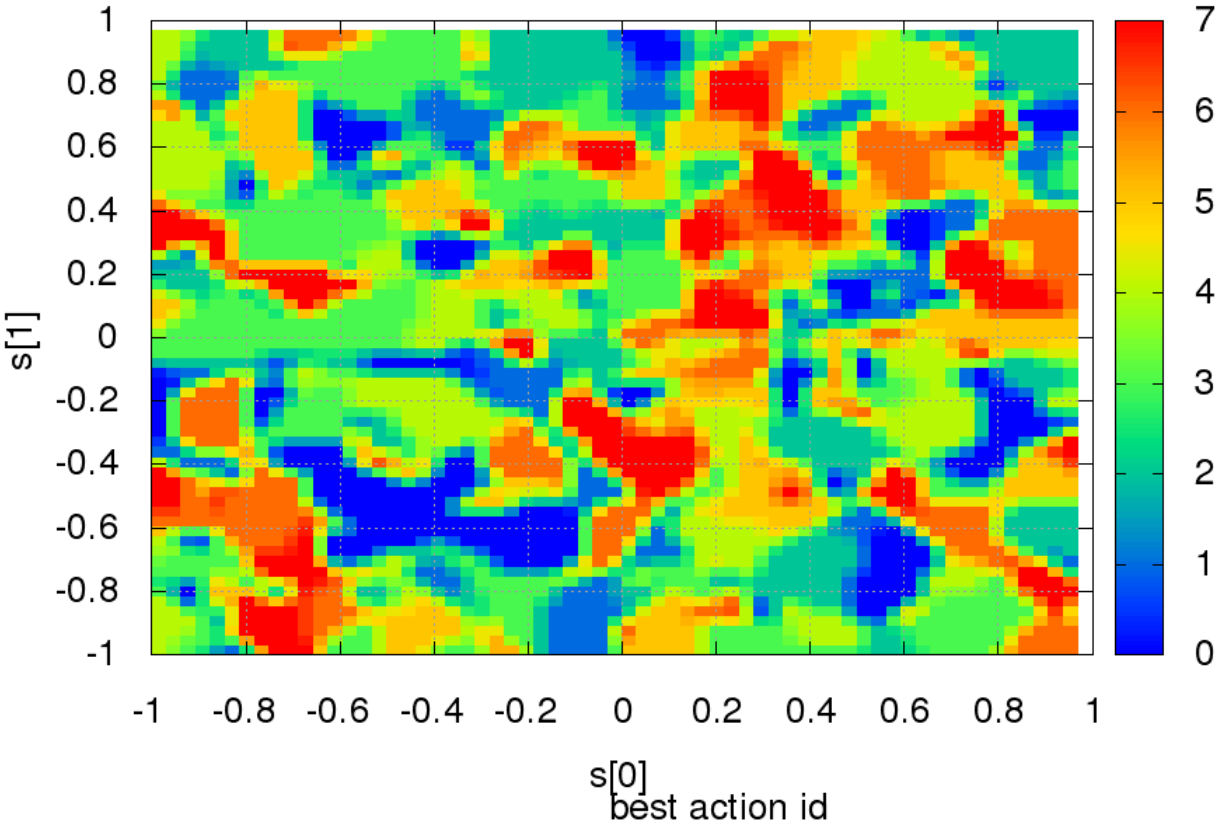
Priebeh trialov - Výsledky experimentov

Mapa 1 - Výsledky experimentov

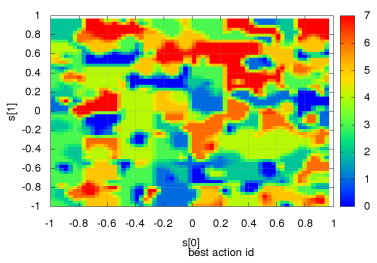
Mapa 0 - Výsledky experimentov

Mapa 2 - Výsledky experimentov

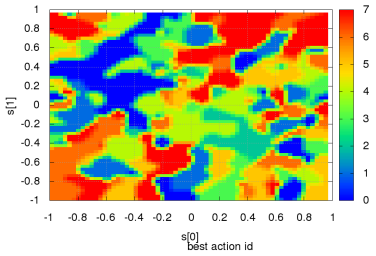
Mapa 3 - Výsledky experimentov



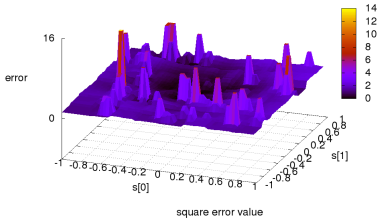
Obr. 3.4: fig:linear combination Gauss



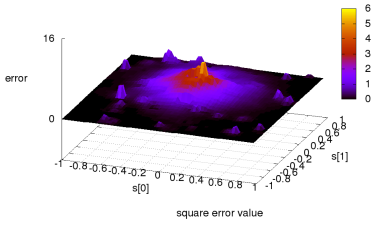
Obr. 3.5: sparse table + linear combination Gauss



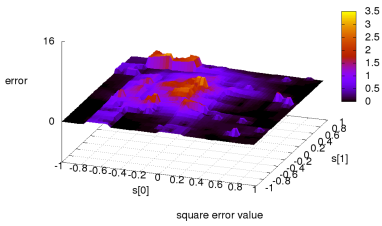
Obr. 3.6: linear combination Kohonen function



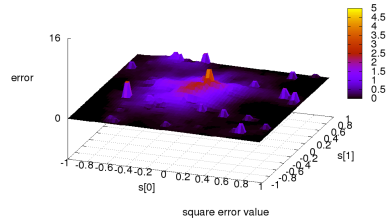
Obr. 3.7: sparse table



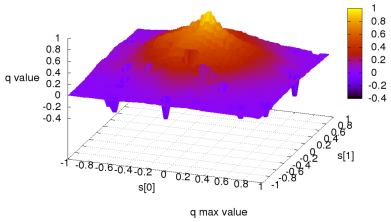
Obr. 3.8: linear combination Gauss



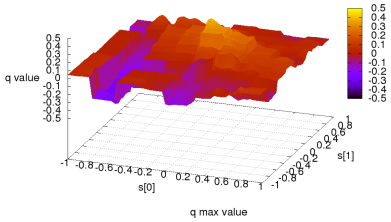
Obr. 3.9: sparse table + linear combination Gauss



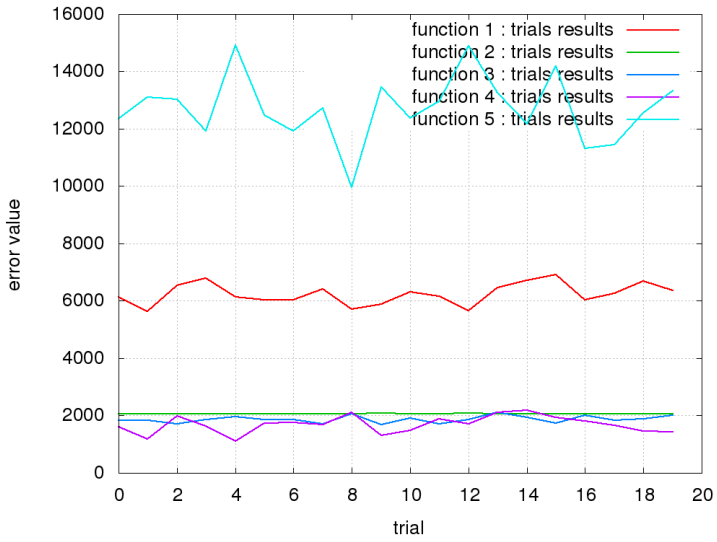
Obr. 3.10: linear combination Kohonen function

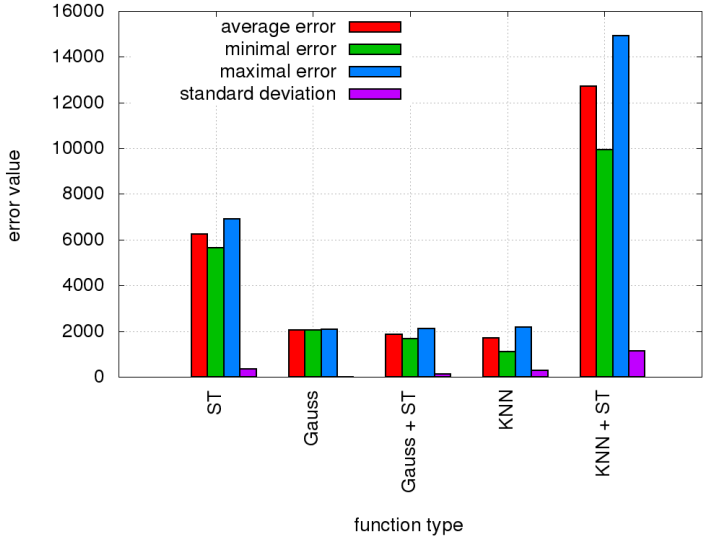


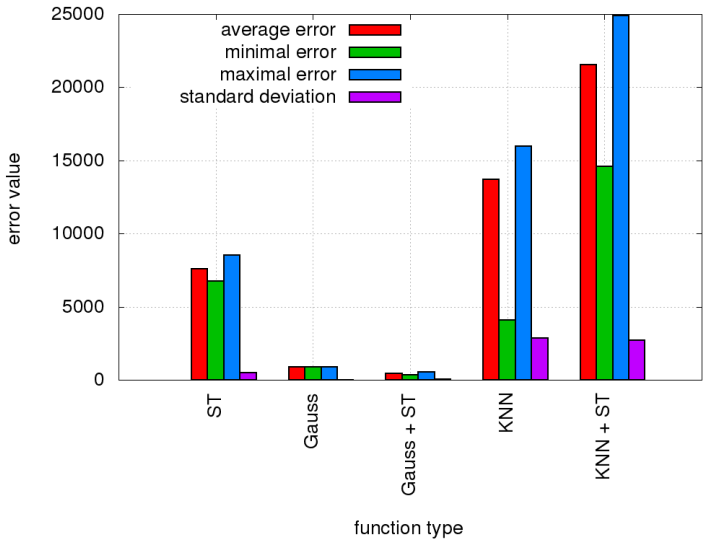
Obr. 3.11: reference table

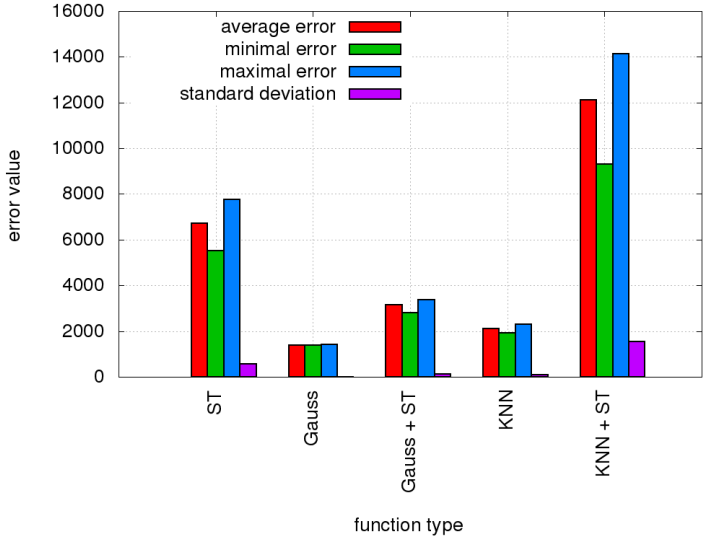


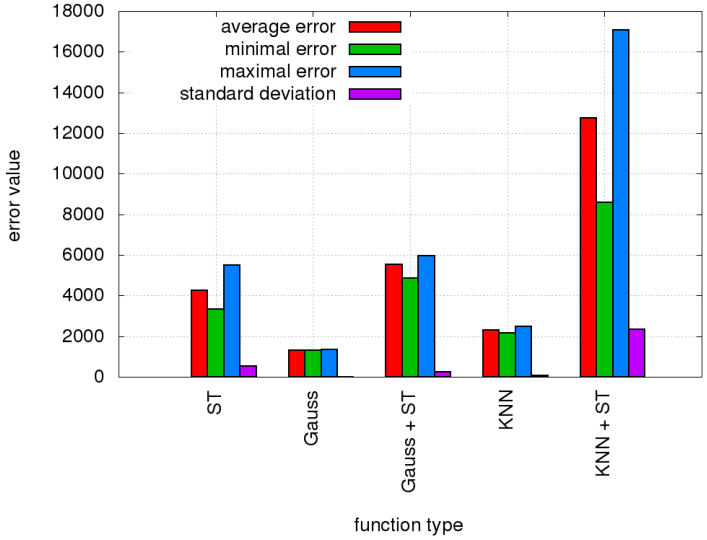
Obr. 3.12: sparse table + linear combination Gauss


function 1 : trials results
function 2 : trials results
function 3 : trials results
function 4 : trials results
function 5 : trials results
error value
trial
0
2000
4000
6000
8000
10000
12000
14000
16000
0
2
4
6
8
10
12
14
16
18
20


average error
minimal error
maximal error
standard deviation
error value
function type
ST
Gauss
Gauss + ST
KNN
KNN + ST
0
2000
4000
6000
8000
10000
12000
14000
16000


average error
minimal error
maximal error
standard deviation
error value
function type
ST
Gauss
Gauss + ST
KNN
KNN + ST
0
5000
10000
15000
20000
25000


16000
14000
12000
10000
8000
6000
4000
2000
0
error value
average error
minimal error
maximal error
standard deviation
ST
Gauss
Gauss + ST
KNN
KNN + ST
function type


18000
16000
14000
12000
10000
8000
6000
4000
2000
0
error value
average error
minimal error
maximal error
standard deviation
ST
Gauss
Gauss + ST
KNN
KNN + ST
function type

Literatúra

[1] Bartsch H. J., *Matematické vzorce*, 3. revidované vydání, Praha, Mladá fronta 2000, ISBN 80-204-0607-7.

[2] Berman G. N., *Zbierka úloh z matematickej analýzy*, Bratislava, ŠNTL 1955.

[3] Peško, Š., *Operační systémi*, Knižnice výpočetní techniky, Nakladatelství technické literatury (1992), SNTL, ISBN 80-03-00269-9.

[4] World of mathematics, A Wolfram Web Resource, `http://mathworld.wolfram.com/`, WolframAlpha – computational knowledge engine, `http://www.wolframalpha.com/`.