OCENJEVANO UČENJE V NEVRONSKIH MREŽAH

INFORMATICA 4/91

Andrej Dobnikar, Jelena Ficzko, Mira Trebar
Fakulteta za elektrotehniko in računalništvo

Keywords: reinforcement learning, stochastic reinforcement learning, neural networks

POVZETEK

Ena izmed kategorij učenja v umetnih nevronskih mrežah je ocenjevano (reinforcement, graded) učenje. To vrsto učenja srečamo tudi v bioloških sistemih. V članku je predstavljen stohastični algoritem z ocenjevanim učenjem. Obnašanje algoritma je podano primerjalno za eno samo stohastično enoto, za stohastično enoto v povezavi z back-propagation enoto ter za samo back-propagation enoto, kjer pa je uporabljeno nadzorovano učenje. Čeprav je nadzorovano učenje bistveno hitrejše od ocenjevanega, pa je ta vrsta učenja uporabna tudi v primerih, ko želeni izhodi za vsak vhod niso poznani.

ABSTRACT

One category of learning methods in artificial neural networks is reinforcement or graded learning. This kind of learning can be met in biological systems also. This paper presents the performance of the stochastic reinforcement learning algorithm. The performance of the algorithm with one stochastic unit and with stochastic unit and back-propagation unit is compared with the performance of the back-propagation unit that is trained using supervised learning. Although supervised learning is much faster than reinforcement learning, the latter can be used even though the desired outputs for every input are not known.

I. UVOD

Paraleleno strukturo, sestavljeno iz procesnih elementov (ti so med seboj povezani z usmerjenimi povezavami - sinapsami), ki je sposobna specifičnega porazdeljenega procesiranja informacij, imenujemo nevronska mreža. Osnovni gradniki nevronske mreže, procesni elementi, izvajajo procesiranje na osnovi svoje prenosne funkcije, trenutnih vrednosti na svojih vhodih in vrednosti v svojem lokalnem pomnilniku /1/. Pri tem igra pomembno vlogo pravilo učenja, to je pravilo, po katerem se spreminjajo vrednosti v internih pomnilnikih procesnih elementov. Nevronske mreže, ki uporabljajo eno ali več pravil učenja, morajo nujno skozi fazo učenja, ki lahko poteka v obliki :

- nadzorovanega učenja (supervised learning)
- samoorganizacije (self - organization)
- ocenjevanega učenja (graded, reinforcement learning).

Pri nadzorovanem učenju mreža potrebuje "učitelja", ki ima nalogo posredovati pravilni izhod za vsak mreži podani vhod. Mreži tako predstavimo pare (x_i, y_i), $i = 1,...,n$, pri čemer

je x_i vhod, yi pa pravilni oz. želeni odgovor na x_i.

Za samoorganizacijo je značilno, da mreža razen vhodov ne potrebuje nobene dodatne informacije, saj ji le-ti zadostujejo, da se na osnovi notranjega pravila sama ustrezno oblikuje.

V primerjavi z nadzorovanim gre pri ocenjevanem učenju za drugačno vrsto signala, ki ga mreža dobi v fazi učenja. Namesto vektorja, ki za dani vhod predstavlja želeni izhod, imamo pri ocenjevanem učenju skalarno oceno obnašanja mreže glede na neko mero. Upoštevajoč signal ocene, skuša mreža izboljšati obnašanje v smeri generiranja pravilnih izhodov. Pri ocenjevanem učenju gre tako za dvoje, za iskanje pravilnih izhodov na podane vhode in za pomnjenje pravilnih izhodov. Ker daje skalarni signal ocene pri ocenjevanem učenju manj informacije kot želeni izhod pri nadzorovanem učenju, je ocenjevano učenje običajno počasnejše od nadzorovanega, njegove prednosti pa se pokažejo v primerih, ko želeni izhodi niso vnaprej poznani, oziroma ko je kot odgovor na dani vhod možnih več alternativ (npr. pri kontroli sistemov z določeno stopnjo avtonomnosti).

II. UČENJE V BIOLOŠKIH SISTEMIH

Nevronsko računalništvo, katerega predmet zanimanja so v prvi vrsti nevronske mreže, se je razvijalo tudi pod vplivom nevrologije (znanstvene discipline, ki skuša razložiti delovanje možganov in miselnih procesov), vendar pa v zadnjem času postaja ta vpliv vzajemen. Nove arhitekture nevronskih mrež in novi koncepti ter teorije o delovanju nevronskih mrež pomagajo tudi nevrologiji na njeni poti do odkritja delovanja možganov in procesov v njih.

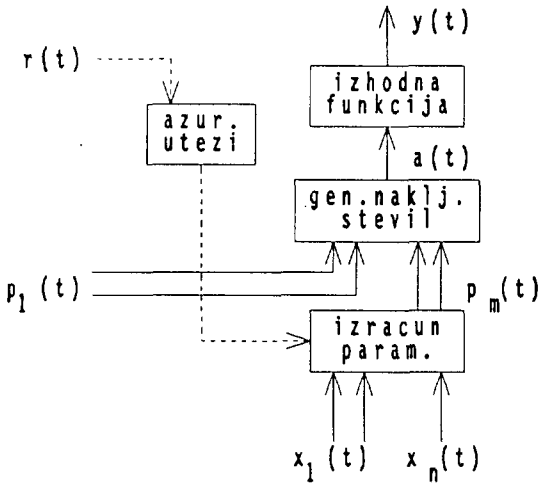
Ocenjevano učenje kot eden izmed treh možnih načinov učenja v umetnih nevronskih mrežah ima svojo "živo" vzporednico. Psihologi so v klasični teoriji učenja postavili vrsto definicij pojma "reinforcement", pri čemer pa so si te definicije v jedru enotne, da gre za operacijo okrepitve, ojačanja , utrjevanja, oziroma za sam dogodek, ki tako ojačanje oz. okrepitev povzroči. Pri tem je to, kar se ojača, običajno naučen odgovor ali pa vez med tem odgovorom in dražljajem /3/. Okrepitev je lahko pozitivna ali pa negativna. Pozitivno okrepitev predstavlja dogodek, dražljaj ali vedenje, ki, kadar je pogojen-o z odgovorom, povzroči povečanje verjetnosti za pojav tega odgovora v bodoče. Nasprotno pa negativno okrepitev predstavlja dogodek, dražljaj ali vedenje, ki, kadar je njegova ukinitev pogojena z odgovorom, poveča verjetnost tega odgovora v bodoče. Glede na to, kakšna je odvisnost okrepitve od nekega vidika odgovora (ta je lahko prostorski, časovni ali sekvenčni), so psihologi klasificirali okrepitve v tri razrede. V prvem razredu t.i. "enostavnih" okrepitev gre za samo eno vrsto pogojenosti med odgovorom in pojavom okrepitve. V drugi razred spadajo "sestavljene" okrepitve, ki so sekvenčna ali paralelna kombinacija dveh ali več "enostavnih" okrepitev. Okrepitve, ki jih ne moremo uvrstiti v nobenega od prejšnjih razredov, spadajo v razred t.i. "posebnih" okrepitev. Ocenjevano učenje (reinforcement, graded learning) morda ni najboljše poimenovanje za to vrsto učenja, vsaj v takšnem smislu ne, kot to razumejo psihologi. Glede na to, da ime dobro označi to vrsto učenja v umetnih nevronskih mrežah (ocena, kritika, ki okrepi obnašanje mreže v pravi smeri), je to poimenovanje opravičeno.

III. STOHASTIČNI ALGORITEM Z OCENJEVANIM UČENJEM

Ideja za ta algoritem, ki ga je v osnovi razvil V. Gullapalli /4/, izvira iz teorije učečih avtomatov. Stohastični učeči avtomat je

abstraktni stroj, ki akcije izbira naključno po podani verjetnostni porazdelitvi, pri tem pa iz okolja dobiva povratno informacijo, ki predstavlja ovrednotenje akcij. Upoštevajoč povratno informacijo, avtomat ažurira verjetnostno porazdelitev za akcije tako, da poveča verjetnost za pozitivno ovrednotenje akcij v bodoče.

Učeči avtomat deluje v povratni povezavi z okoljem, ki v času t pripelje na vhod avtomata vektor $\mathbf{x}(t)$. Odgovor avtomata $y(t)$ je naključno izbran glede na verjetnostno porazdelitev na intervalu $Y \subseteq R$. Signal ocene $r(t) \in R = [0,1]$, ki ga generira okolje, predstavlja oceno odgovora $y(t)$ v kontekstu vhoda $\mathbf{x}(t)$. Cilj učečega avtomata je naučiti se odgovarjati na vsak vhodni vektor $\mathbf{x}$ s takšnim odgovorom y, da bo ocena, ki jo bo za ta odgovor prejel od okolja, maksimalna. Opisani stohastični avtomat se implementira kot stohastična enota (slika 1), ki nastopa kot komponenta v mreži.



slika 1

Vhod v enoto v času t je $\mathbf{x}(t) = (x_1(t), x_2(t),...,x_n(t))$, ki se uporabi za izračun parametrov $p_1(t)$, $p_2(t)$,..., $p_m(t)$ verjetnostne porazdelitve, po kateri se naključno generira aktivnost enote. Parametre porazdelitvene funkcije lahko dobimo od zunaj ali pa so določeni kot utežena vsota vhodov, z različnim naborom uteži za vsak parameter. Aktivnost enote je tako naključna spremenljivka porazdelitve, določene s parametri $p_1(t),...,p_m(t)$. Izhod iz enote $y(t)$ je funkcija aktivnosti $y(t) = f(a(t))$, pri čemer je funkcija f (pragovna funkcija, logistična funkcija) izbrana glede na vrsto izhoda, ki ga želimo.

Za implementacijo določenega stohastičnega algoritma učenja je potrebno tako določiti :

1. porazdelitev, ki se uporabi za generiranje naključnih vrednosti aktivacije
2. funkcije, ki se uporabijo za izračun parametrov porazdelitve
3. izhodno funkcijo f
4. pravila za ažuriranje uteži

V članku je predstavljen algoritem stohastičnega ocenjevanega učenja za učenje funkcij z realnimi izhodi. Za generiranje aktivnosti enote se uporabi normalna porazdelitev $\Psi(\mu, \sigma)$. Izhod iz enote, ki je realna vrednost, je zvezna, monotona funkcija aktivnosti. Signal ocene je iz intervala $[0, 1]$. Trenutni vhodi v enoto določajo srednjo vrednost in standardno deviacijo porazdelitve, na podlagi katere se naključno generira aktivnost enote.

Učenje poteka v smislu ažuriranja parametrov normalne porazdelitve (standardne deviacije in srednje vrednosti) v smeri povečanja verjetnosti za generiranje optimalnega izhoda za vsak vhod. Pri tem je srednja vrednost μ ocena za optimalno aktivnost, standardna deviacija σ pa določa obseg iskanja okrog trenutne srednje vrednosti aktivnosti enote.

Ker naj bo srednja vrednost porazdelitve μ ocena za optimalni izhod, izračunamo μ kot uteženo vsoto vhodov :

$$\mu(t) = \sum_{i=1}^{n} w_i(t)\, x_i(t) + w_{prag}(t)$$

Za dani vhod naj bo st.deviacija σ odvisna od tega, kako blizu je trenutni pričakovani izhod optimalnemu izhodu za ta vhod. Mera za to pa je signal ocene iz okolja. Za dani vhod je tako st.deviacija σ odvisna od pričakovanega signala ocene, ki ga prav tako dobimo kot uteženo vsoto vhodov :

$$r(t) = \sum_{i=1}^{n} v_i(t)\, x_i(t) + v_{prag}(t)$$

Pričakovani signal ocene uporabimo za izračun st.deviacije :

$$\sigma(t) = s(r(t)),$$

kjer je funkcija s monotono padajoča nenegativna funkcija pričakovanega signala ocene.

Na osnovi $\sigma(t)$ in $\mu(t)$ se izračuna aktivnost a(t) kot naključna spremenljivka normalne porazdelitve:

$$a(t) \sim \Psi(\mu(t), \sigma(t)) .$$

Izhodna funkcija f preslika aktivnost enote a(t) v izhod enote y(t) :

$$y(t) = f(a(t)), \text{ pri čemer je } f(x) = \frac{1}{1 + e^{-x}} .$$

Pravila za ažuriranje uteži, ki določajo srednjo vrednost:

$$w_i(t+1) = w_i(t) + \alpha\, \Delta_w(t)\, x_i(t)$$

$$w_{prag}(t+1) = w_{prag}(t) + \alpha\, \Delta_w(t)$$

$$\Delta_w(t) = \Big(r(t) - r(t)\Big) \left(\frac{a(t) - \mu(t)}{\sigma(t)}\right)$$

Pravila za ažuriranje uteži, ki določajo st.deviacijo :

$$v_i(t+1) = v_i(t) + \beta\, \Delta_v(t)\, x_i(t)$$

$$v_{prag}(t+1) = v_{prag}(t) + \beta\, \Delta_v(t)$$

$$\Delta_v(t) = r(t) - r(t)$$

Pri tem sta α in β parametra učenja.

Osnovni cikel učenja poteka po naslednjih korakih :

1. okolje poda enoti vhodni vektor $x_i(t)$, $i = 1,...,n$

2. enota uporabi vhodni vektor za izračun srednje vrednosti $\mu(t)$ (uporabijo se uteži $w_i(t)$ in w_{prag})

3. enota izračuna pričakovani signal kritike r(t), ki ga uporabi za izračun $\sigma(t)$ (pri tem se uporabijo uteži $v_i(t)$ in v_{prag})

4. enota izračuna aktivnost $a(t) \sim \Psi(\mu(t), \sigma(t))$ in izhod $y(t) = f(a(t))$

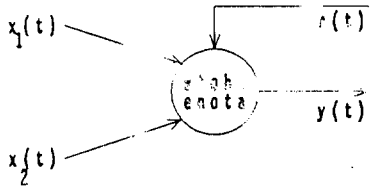
5. okolje ovrednoti izhod in odpošlje signal ocene r(t)

6. r(t) omogoči ažuriranje uteži

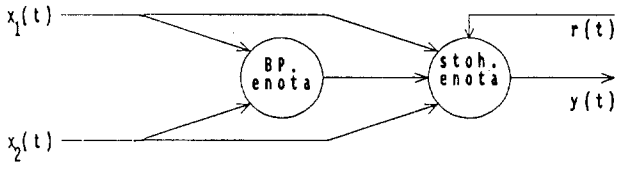
IV. OBNAŠANJE ALGORITMA

Delovanje algoritma je preizkušeno na več opravilih, ki so definirana kot množica parov "dražljaj - odgovor" iz $[0, 1]^2 \times (0, 1)$. Za vsa opravila se je uporabila najprej samo ena stohastična enota (slika 2), nato stohastična enota v povezavi z back-propagation enoto (slika 3) in primerjalno še ena sama back-propagation enota (učenje te enote je potekalo v obliki nadzorovanega učenja s

parametrom učenja 0.1). Hitrost konvergence za vse tri primere je razvidna iz grafov (slike 4, 5 in 6).



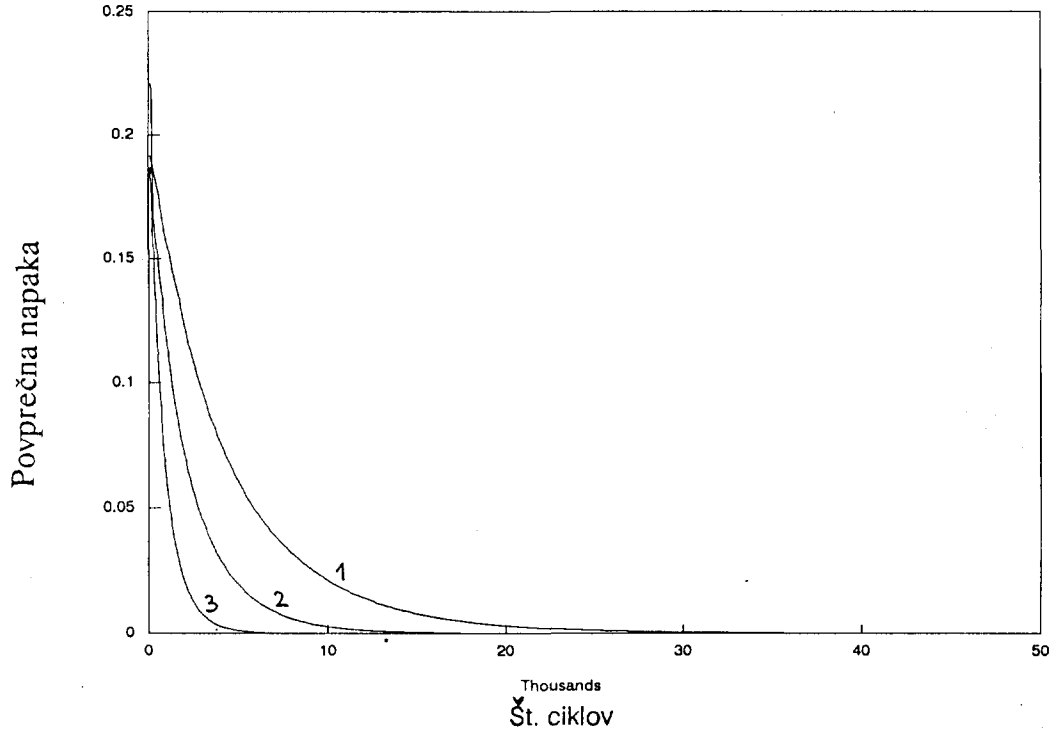
slika 2



slika 3

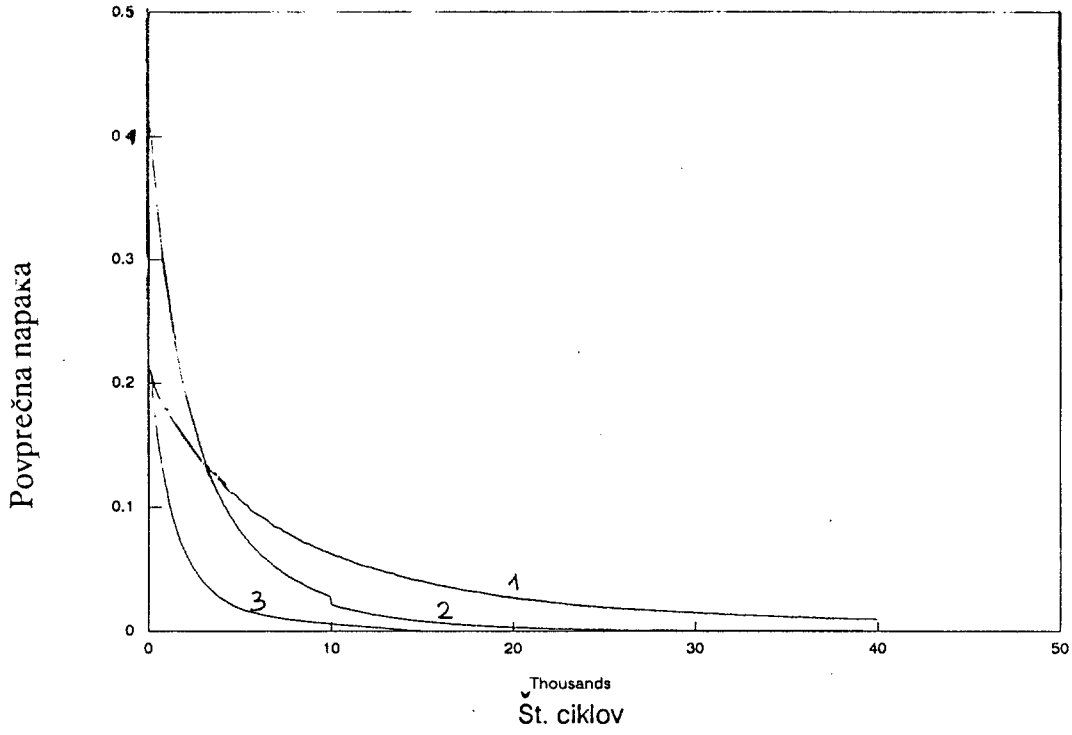
Delovanja algoritma v primerih ene stohastične enote

Ob začetku učenja so vse uteži enote postavljene na 0, vrednosti parametrov učenja pa so $\alpha = 0.5$ in $\beta = 0.7$. Na vhod enote pripeljemo v vsaki fazi učenja vse vhodne vektorje v naključnem vrstnem redu, pri tem pa se osnovni cikel učenja ponovi za vsak vhodni vektor. Ob koncu točke 4 osnovnega cikla učenja se izračuna napaka kot razlika med dejanskim in želenim izhodom, ki se uporabi za določitev signala ocene. Sledi ažuriranje uteži. Učenje poteka tako dolgo, dokler povprečna napaka ne pade pod določeno mejo, oziroma po preteku določenega števila ciklov učenja.



Krivulja št. 1 prikazuje napako za stoh. enoto. Krivulja št. 2 prikazuje napako BP in stoh. enote in krivulja 3 prikazuje napako ene same BP enote. V vseh primerih smo učili z vektorji : (0.2, 0.6) -> 0.3 in (0.8, 0.3) -> 0.7

Slika 4



Krivulja št. 1 prikazuje napako za stoh. enoto. Krivulja št. 2 prikazuje napako BP in stoh. enote in krivulja 3 prikazuje napako ene same BP enote. V vseh primerih smo učili z vektorji : (0.4, 0.7) -> 0.3, (0.1, 0.3) -> 0.6 in (0.6, 0.2) -> 0.9.

Slika 5

Signal ocene iz okolja je določen kot :

$r(t) = 1 - |\text{napaka}|$,

pri tem je $|\text{napaka}| <= 1$.

Delovanja algoritma v primerih stohastične in back-propagation enote

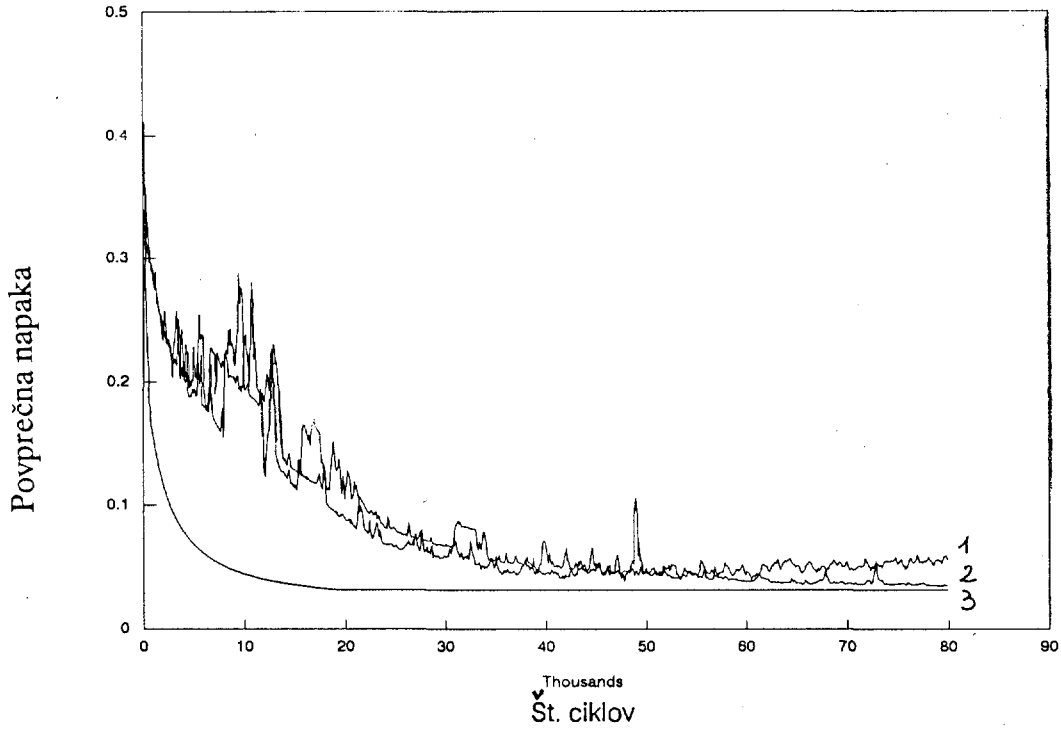
Hibridna mreža, sestavljena iz stohastične izhodne enot in enote v skritem nivoju, ki je tipa back-propagation, izkazuje boljše obnašanje kot ena sama stohastična enota.

V taki mreži se kot aproksimacija napake na izhodu uporabi vrednost $\Delta_w(t)$. Poleg tega se osnovnemu ciklu učenja doda še korak, kjer se izraz $\Delta_w(t)$ iz izhodne enote propagira nazaj kot napaka na skrito enoto in s tem omogoča ažuriranje uteži back-propagation enote.

Ob začetku učenja so uteži za back-propagation enoto postavljene naključno, za stohastično enoto pa so 0, parameter učenja za stohastično enoto $\beta = 0.7$, parameter α pa je postavljen različno za povezave na vhode $\alpha = 0.5$ in za povezave na skrito enoto $\alpha = 0.1$. Parameter učenja za back-propagation enoto je 0.1. V vsaki fazi učenja se na mrežo pripeljejo vsi vhodni vektorji, za vsakega se ponovi osnovni cikel učenja. Signal ocene je določen enako kot v primeru ene stohastične enote.

V. ZAKLJUČEK

Rezultati so pokazali, da stohastična enota, povezana z back-propagation enoto izkazuje boljše obnašanje kot ena sama stohastična enota. Ta razlika je sicer v nekaterih primerih (slika 6) majhna.



Krivulja št. 1 prikazuje napako za stoh. enoto. Krivulja št. 2 prikazuje napako BP in stoh. enotė in krivulja 3 prikazuje napako ene same BP enote. V vseh primerih smo učili z vektorji : (0.1, 0.1) -> 0.1, (0.1, 0.9) -> 0.1, (0.9, 0.1) -> 0.1 in (0.9, 0.9) -> 0.9.

Slika 6

Konvergenca učenja v primeru ene same stohastične enote in stohastične enote, povezane z back-propagation enoto, je bistveno počasnejša od konvergence učenja pri back-propagation enoti. Faktor, ki vpliva na obnašanje vseh sistemov, ki se učijo z ocenjevanim učenjem, je kvaliteta signala ocene iz okolja. Zato lahko včasih izboljšamo obnašanje mreže, če v signal ocene vgradimo več informacije, ki je specifična za neko opravilo.

Literatura:

/1/ Robert Hecht-Nielsen, Neurocomputing, Addison-Wesley,1990

/2/ Hinton, Connectionist Learning Procedures,Artificial Intelligence 40, 1989

/3/ Dictionary of psychology,

/4/ V.Gullapalli, A Stochastic Reinforcement Learning Algorithm for Learning Real-Valued Functions, Neural Networks, Vol.3,1990

/5/ R.A. Leaver, P. Mars, Stochastic Computing and reinforcement neural networks, Conference on Artificial Neural Networks, London, 1989