UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Primož Gajski

Implementacija igralca Backgammona z nevronsko mrežo

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Branko Šter

Ljubljana, 2010

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01626/2010
Datum: 15.01.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **PRIMOŽ GAJSKI**

Naslov: **IMPLEMENTACIJA IGRALCA BACKGAMMONA Z NEVRONSKO MREŽO**

IMPLEMENTATION OF BACKGAMMON PLAYER WITH NEURAL NETWORK

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Implementirajte program za igranje igre Backgammon, ki ne uporablja nikakršnega vnaprejšnjega znanja. Uči naj se le na osnovi igranja s samim seboj. Za ocenjevanje pozicije uporabite nevronsko mrežo dvonivojski perceptron in algoritem spodbujevanega uc(enja TD(lambda). Uspešnost primerjajte z evaluacijsko funkcijo 'pubeval'. Izdelajte tudi grafični uporabniški vmesnik.

Mentor:

prof. dr. Branko Šter

Fakulteta za računalništvo in informatiko
UNIVERZA V LJUBLJANI
2

Dekan:

prof. dr. Franc Solina

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Primož Gajski,

z vpisno številko 63030121,

sem avtor diplomskega dela z naslovom:

Implementacija igralca Backgammona z nevronsko mrežo

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom
  prof. dr. Branka Štera
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 9. april 2010 Podpis avtorja:

Zahvala

Zahvala gre v prvi vrsti staršema, ki sta mi pomagala pri mojem izobraževanju in se odrekla marsičemu, da sem lahko izpolnil svoje želje. Zahvalil bi se rad tudi Simonu, ki mi je marsikdaj na takšen in drugačen način priskočil na pomoč. Velika hvala gre tudi Heleni, ki mi je ves čas stala ob strani in me vzpodbujala ter mi s svojim trudom pokazala, kaj vse je človek zmožen storiti, če si to res želi.

Nazadnje bi se še rad zahvalil mentorju Branku Šteru za ves trud in čas, ki ga je vložil v mojo diplomsko nalogo, in mi je vedno pomagal ter mi dajal prave napotke, ko se mi je kje zataknilo. Lepa hvala tudi Omarju Abdalla, ki mi je pomagal dizajnirati grafični vmesnik pri aplikaciji NBG.

Kazalo

Seznam uporabljenih kratic in simbolov

backgammon - [1. pomen] ime igre

backgammon - [2. pomen] način zmage pri igri backgammon, kjer zmagovalec dobi 3 točke (najvišja možna zmaga)

gammon - način zmage pri igri backgammon, kjer zmagovalec dobi 2 točki

pubeval - evaluacijska funkcija backgammon pozicij, uporabljena pri testiranju naučene nevronske mreže

NBG - Neural BackGammon, ime aplikacije za igranje igre backgammon

Povzetek

Diplomsko delo Implementacija igralca Backgammona z nevronsko mrežo opisuje način in vsebuje implementacijo, kako se je računalnik zmožen naučiti igre backgammon z igranjem sam s sabo, torej brez človeškega faktorja in brez kakršnega koli vnaprejšnjega znanja o sami igri. Ocenjevanje pozicij za izbor najugodnejše poteze je narejeno z evaluacijsko funkcijo, ki je realizirana z nevronsko mrežo v obliki 2-nivojskega perceptrona.

Učenje nevronske mreže poteka s spodbujevanim učenjem, ki je doseženo z algoritmom vzvratnega učenja. Mreža tako odigra veliko število iger (več milijonov), da osvoji zahtevnejši nivo igre backgammon.

Testiranje je opravljeno tudi brez človeškega faktorja, in sicer je potekalo kot igranje proti t.i. igralcu pubeval, t.j. še ena evaluacijska funkcija, ki tudi ocenjuje pozicije žetonov.

Po zaključenem učenju je nevronska mreža uspešno osvojila znanje ocenjevanja backgammon pozicije in obvlada tako osnovne kot tudi zahtevnejše elemente igre.

Za konec je bilo potrebno realizirati uporabniški grafični vmesnik, ki uporabniku omogoča igranje proti naučeni nevronski mreži.

Ključne besede:

backgammon, nevronska mreža, perceptron, spodbujevano učenje, algoritem vzvratnega učenja

Abstract

Diploma thesis Implementation of Backgammon player with neural network describes implementation of how is a computer capable to learn the game of backgammon. It is achieved with computer playing against itself, without any human help and without any prior knowledge about the game. Evaluation function of game positions, according to the thrown dice and checker positions, was implemented with neural network. Neural network is presented as a 2-layer perceptron.

The learning of neural network was achieved with reinforcement learning and backpropagation algorithm. Within backpropagation algorithm neural network plays several milions of backgammon games to achieve the advanced level of the game.

Testing was also performed without any human interaction. Neural network played backgammon game against so called pubeval player, who also uses for its logic evaluation function to play on strong intermediate level. The testing showed that the neural network successfully accomplished all basic and also many of advanced features of game playing elements.

Diploma work also involves realization of graphical user interface, which allows user to play backgammon against neural network.

Key words:

backgammon, neural network, perceptron, reinforcement learning, backpropagation algorithm

Poglavje 1

Uvod

Odkar obstajajo računalniki, se je pojavljalo vprašanje: „Kaj zmorejo?“ oz. „Kako dobri so v primerjavi s človekom?“ Še posebej nas je zanimalo, kako dobri so, kadar gre za igranje iger, kot so na primer šah, razne igre s kartami (poker, tarok ...) in še mnoge druge. Mene je zanimalo, kako dobro se je sposoben naučiti igro backgammon.

Igra backgammon je zelo zanimiva, ker s številom metov število možnih potez zelo hitro narašča. Tako hitro, da se že po nekaj korakih (metih) število možnosti oz. možnih potez tako „razcveti“, da nima smisla računati verjetnosti, ker imamo enostavno opraviti s prevelikim številom. To posledično pomeni, da je zelo težko napovedati, katera poteza je v danem trenutku najboljša.

Ponavadi se za takšne probleme uporabi metoda ali algoritem, ki ni nič drugega kot logika, ki jo uporablja človek, kadar igra to igro. Se pravi, da gre za preslikavo človeške logike oz. razmišljanja v računalniški jezik.

Za implementacijo igralca backgammona sem uporabil drugačen pristop. Definiral sem okolje, torej pravila igre, v katerem poteka odločanje, nato sem ustvaril dva računalniška igralca backgammona brez znanja, ki igrata drug proti drugemu. Več iger odigrata, večje je njuno znanje o igri, torej postajata boljša in boljša. Cilj je bil, da se igralec backgammona nauči igrati igro na zatevnejšem nivoju in ne zgolj osnovnem. Takšen igralec naj bi znal osnovne principe igre, kot je izbijanje nasprotnikovih žetonov, gradnja novih polj, ne puščanja samih žetonov. Razlago o naštetih elementih igre najdemo v poglavju Igra backgammon (2. poglavje) in v razdelku Osvojeno znanje (oštevilčeno s 4.3). Ravno tako naj bi obvladal naprednejše elemente igre, kot je postavitev blokade, statistično gledano delati „dobre poteze“, ki se na prvi pogled ne zdijo najboljša izbira, ampak se to potrdi šele čez nekaj potez.

Izdelanih je bilo že ogromno programov, ki znajo igrati backgammon. Korenine segajo vse do C. E. Shannonovega algoritma za igranje šaha iz leta 1950. Bil je zelo preprost algoritem, ki pa je vseboval zelo zanimiv koncept. Šlo je za funkcijo, ki poda oceno za vse možne poteze glede na trenutno pozicijo. Nato se izbere poteza z najmanjšo oz. največjo vrednostjo. Bistvenega pomena za uspeh oz. zmago pri igranju iger s takšnim pristopom je, kako natančna je evaluacijska funkcija [2].

To idejo je možno zaslediti tudi v moji implementaciji igralca backgammona. Torej preden se izbere določena poteza na dani poziciji, se naredi seznam vseh možnih potez in se jih oceni z evaluacijsko funkcijo. Igralec zatem odigra potezo, pri kateri bo pozicija ocenjena z največjo oz. najmanšo vrednostjo, odvisno od tega, kateri igralec je na vrsti (črni ali beli).

Zelo „elegantna“ rešitev problema izbira najprimernejše poteze, vendar se takoj pojavi naslednji problem: „Kako izdelati evaluacijsko funkcijo?“

Evaluacijska funkcija je bila narejena z nevronsko mrežo. Nevronske mreže imajo to lepo lastnost, da so se sposobne naučiti zelo kompleksnih funkcij. Kompleksnost še dodatno poveča sama narava igre backgammon, pri kateri se ustreznost ali uspešnost neke poteze izkaže šele čez določeno število potez ali celo ob samem koncu igre. Dobimo t.i. zakasnjeno oceno uspešnosti.

Nevronska mreža je izredno primerna za implementacijo igralca backgammona, ker se je sposobna naučiti funkcije, ki se zelo dobro prilega dejanskemu opisu ocenjevanja backgammon pozicij. Tukaj je potrebno poudariti, da se je sposobna naučiti, če imamo na razpolago dovolj časa. Vsako učenje, pa naj bo to človeško ali umetno, zahteva čas. Da se je nevronska mreža naučila igrati igro backgammon na določenem nivoju, je bilo potrebno odigrati približno nekaj miljonov iger, za kar je bilo potrebno tudi več kot teden dni časa.

Pri nevronskih mrežah imamo več možnih načinov učenja. Nekako najbolj primerno in tudi uporabno je bilo spodbujevano učenje z vzvratnim algoritmom (ang. „backpropagation algorithm“). Takšno učenje se izkaže še za posebej uspešno, ker imamo opravka z zakasnjeno oceno, ki skozi veliko število iger pridobi oceno uspešnosti. Zaradi ogromnega števila potrebnih odigranih iger, da igralec backgammona igra na določenem nivoju, je zelo ugodno, da med učenjem ne rabi sodelovati človek. Torej računalnik se uči „sam“! To je doseženo z implementacijo dveh igralcev backgammona, ki igrata drug proti drugemu. Več iger odigrata, večje je njuno znanje. Kajti v nasprotnem primeru bi moral človek odigrati veliko število iger proti računalniku, oziroma bi moral pripraviti veliko število različnih možnih scenarijev, kako se igra odvrti. Tako sem si prihranil ogromno časa. Tudi če bi bil sposoben odigrati toliko iger, bi igralca backgammona naučil igrati na svoj slog igre, česar pa seveda ne želim.

Pa tudi če bi imel na voljo vrhunske igralce backgammona, še vedno obstaja zmotljivi človeški faktor, ki spregleda kakšno boljšo potezo oz. rešitev problema, kar lahko privede do tega, da v kakšnih specifičnih situacijah naučeni igralec backgammona statistično gledano ne odigra najoptimalnejše poteze.

Po končanem učenju je seveda igralca potrebno testirati, kako dobra sta in kaj sta se naučila. V bistvu sem testiral vedno samo enega igralca, ker sta med učenjem imela približno razmerje zmag 1:1, tako da je bilo vseeno, katerega sem testiral. Tudi za pričakovati je bilo, da je njuno znanje približno enako, ker se ob eni igri učita oba, in ne samo tisti, ki zmaguje. Tako pri učenju kot pri testiranju sem si prihranil precej časa, saj sem tudi testiranje opravil z računalniškim igralcem, ki igra na določenem nivoju. Za testiranje sem uporabil model igralca backgammona, ki ga je definiral Gerald Tesauro [3]. V bistvu gre za ocenjevalno funkcijo „pubeval“, ki uporablja linearno funkcijo z utežmi. Igralec pubeval zna izbijati nasprotniku žetone, graditi nova polja, zgraditi blokado, zapreti svojo hišo, tako da nasprotnik čaka na prosto polje ... Skratka gre za solidnega igralca backgammona. Testiranje je potekalo tako, da je naučeni igralec proti pubeval igralcu odigral 1000 iger; pri tem me je zanimalo, koliko zmag je dosegel naučeni igralec.

Sledi kratek opis posameznih poglavij. Takoj za uvodom (1. poglavje) je podan opis same igre backgammon (2. poglavje), da bo bralec lažje razumel diplomsko nalogo in se lažje osredotočil na glavni del (3. poglavje), ki najprej na grobo opiše postopek implementacije kasneje pa natančno obdela vse probleme, ki so nastopili. Potem je opisano, kako sem se lotil testiranja (4. poglavje), kjer so zraven tudi podani in komentirani rezultati. V 5. poglavju je opisana uporaba aplikacije NBG. Temu sledijo sklepne ugotovitve, ki so zapisane v 6. poglavju. Na koncu je dodan dodatek, kako zagnati program iz urejevalnika Eclipse.

Poglavje 2

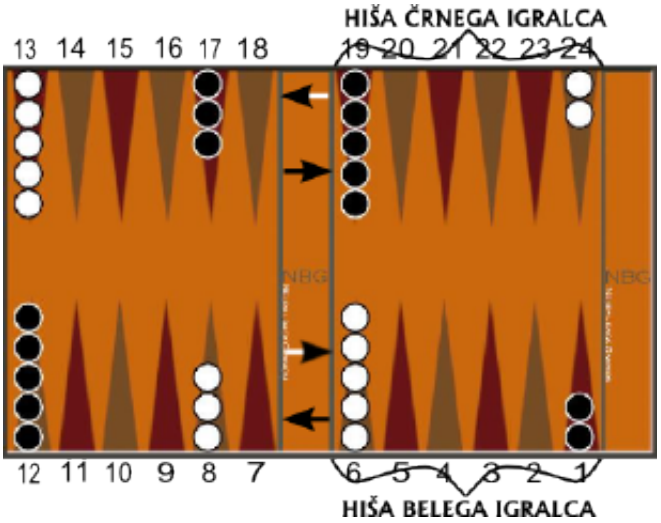
Igra backgammon

Preden se spustimo v bolj podrobno opisovanje postopka realizacije igralca backgammona, je koristno vedeti nekaj o igri in predvsem pravila igre.

Igra naj bi izvirala iz Mezopotamije. Prvotno obliko take igre, ki je zelo spominjala na backgammon, so našli pri Sumercih približno iz leta 3000 pr. n. št. Podobne verzije te igre so se razlikovale po številu polj in žetonov. Igro so kasneje poznali tudi v starem Egiptu, Perziji in Antiki. Ravno pri Rimljanih je začela najbolj spominjati na današnjo obliko, saj je imela že 2 krat po 12 polj in 2 krat po 15 žetonov. Imenovali so jo „Tabula“. V Starem Rimu je dosegla celo takšne razsežnosti, da so jo morali prepovedati zaradi hazarderjev in kockarjev. Rimljani so jo tako raznesli po celotni Evropi. Še posebej se je v srednjem veku oprijela v Britaniji, kjer so jo imenovali „Tables“ in celo prvič „Backgammon“. Beseda naj bi izhajala iz saščine, v katerem „baec“ pomeni zadaj ali nazaj in „gamen“ pomeni igra. Ena od možnih razlag pomena je igra, ki se igra nekje zadaj - po kakšnih skrivnih kotičkih. Tudi v srednjem veku je bil backgammon zaradi kockanja in stav prepovedana igra in mogoče ravno zaradi tega vedno nekako v ozadju za šahom, ki pa ni bil prepovedan. Igra je dobila popolnoma novo dimenzijo v 20-ih letih prejšnjega stoletja v New Yorku, kjer so ji kockarji dodali še kocko za podvajanje (ang. „doubling cube“). V 60-ih letih dvajsetega stoletja je Alexis Obelensky organiziral prvo uradno tekmovanje za naslov svetovnega prvaka v igri backgammon [6].

Danes je igra dobro poznana po vsem svetu. Obstajajo raznorazne knjige, ki opisujejo strategije igre, kot tudi razni backgammon strežniki, kjer je možno igrati backgammon z igralci iz vsega sveta.

Kot že rečeno, so pravila zelo preprosta. Backgammon je strateška igra za dva igralca. Vsak igralec ima 15 žetonov, ki jih mora spraviti iz začetne pozicije čez 24 polj na svojo stran - v hišo (slika 2.1), od koder lahko prične



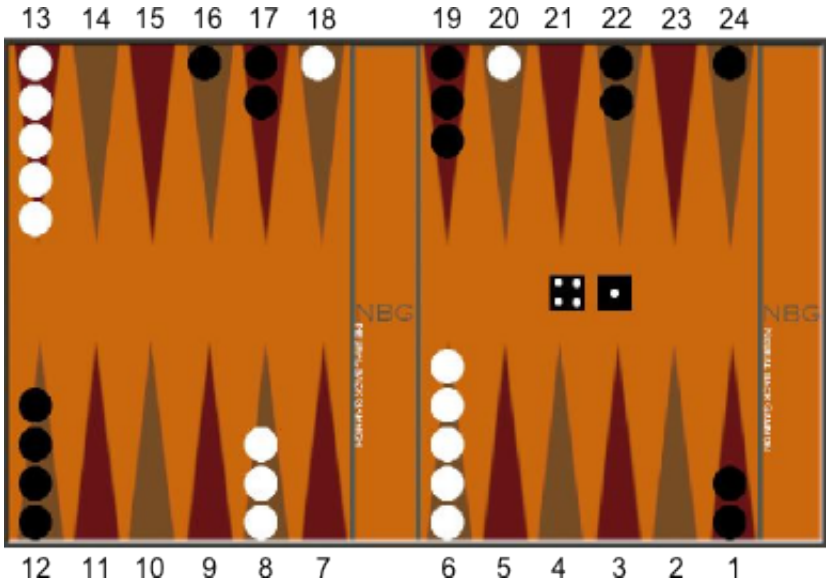
Slika 2.1: Igralna plošča s 24 polji (4*6 polj). Žetoni so razvrščeni na začetne pozicije. Puščice nakazujejo smer gibanja žetonov. Označeni sta tudi hiši igralcev.

s pospravljanjem žetonov. Vsak igralec ima 2 kocki. Začne tisti, ki vrže večjo vsoto kock, pod pogojem, da kocki nista enaki.

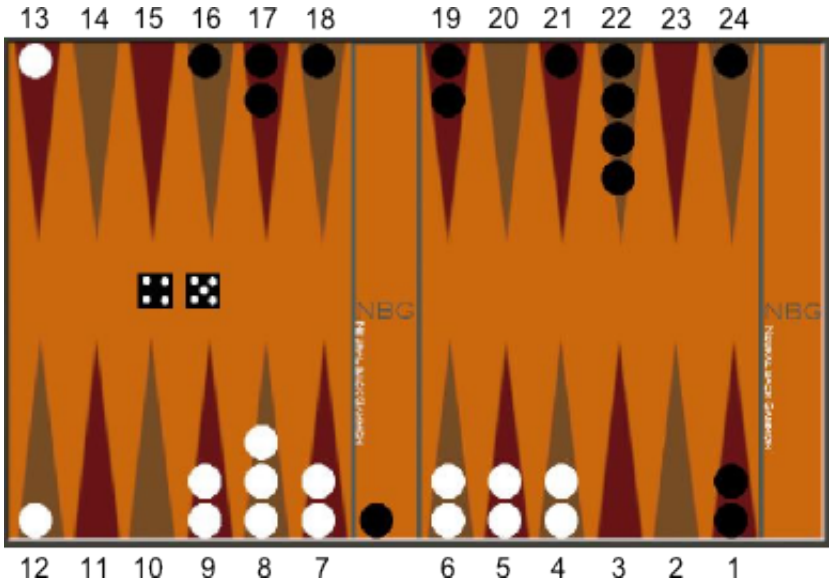
Igralca se izmenjujeta v metih. Igralec mora premakniti žeton, za število pik, kot mu jih kaže kocka. Obvezno je izkoristiti obe kocki, če je le mogoče. Torej igralec lahko premakne 2 različna žetona z vsako kocko posebej ali en žeton za skupno število pik na kockah, če je vmesno polje prosto. Če kocki kažeta enaki števili, potem se število pik podvoji. Recimo, če vržemo 6 in 6, potem imamo na voljo 4 poteze s po šestimi pikami.

Žeton je možno premakniti na polje, če na polju ni nobenega žetona, če je že kakšen igralčev žeton gori ali če je gori samo en nasprotnikov žeton (slika 2.2). V slednjem primeru se nasprotnikov žeton izbije, kar pomeni, da ga mora dati na čakanje. Če ima igralec kakšen žeton na čakanju, ne more nadaljevati z igro, dokler ga ne spravi v igro (slika 2.3). V igro ga lahko spravi, če vrže kocko, ki kaže prosto polje.

Ko ima enkrat igralec vse svoje žetone v svoji hiši (na zadnjih 6 poljih), lahko prične s pospravljanjem žetonov iz igre. Zmaga tisti, ki pospravi žetone prej iz igre. Poznamo več vrst zmage. Navadno ali običajno zmago, ki prinaša 1 točko, dosežemo, kadar premagamo nasprotnika, ki ima že pospravljen vsaj en žeton. Zmago vredno dve točki ali „gammon“ dosežemo, kadar nasprotnik ni pospravil še nobenega žetona, vendar nima nobenega žetona v nasprotnikovi hiši. Če jih ima še v nasprotnikovi hiši, smo dosegli najvišjo možno zmago



Slika 2.2: Beli igralec lahko naredi potezo 20-16 in pri tem izbije črn žeton iz 16. polja, ne more pa narediti poteze 20-19, ker so na 19. polju 3 črni žetoni. Polja so oštevilčena v smeri gibanja črnih žetonov.



Slika 2.3: Primer iz igre, kjer črni igralec ne more spraviti žetona v igro, ker sta tako 4. kot 5. polje zasedni. Igralec mora počakati na ponoven met kock.

vredno 3 točke ali „backgammon“.

V originalni igri obstaja še stavna kocka ali kocka za podvajanje, ki zmago poveča za stavni faktor. Na začetku igre je stavni faktor 1. Ob sprejemu stave se stavni faktor podvoji. Maksimalni stavni faktor je 32. Če nasprotni igralec stavo zavrne, pomeni, da avtomatsko izgubi igro [7].

Ker bi kocka za podvajanje bistveno vplivala na kompleksnost problema, sem implementiral igro backgammon brez stavne kocke.

Da je igra tako stara, niti ni nenavadno, saj jo definirajo preprosta pravila. Tudi sama igralna površina je enostavna (samo 24 polj) in figure (žetoni) so vsi enaki med sabo. Pravila so čisto preprosta, ki pa zelo hitro naredijo igro zanimivo in kompleksno hkrati, ker je težko napovedati ali se odločiti, katera poteza je najbolj primerna pri danem metu. Igro backgammon se ni težko naučiti igrati, veliko težje jo je znati dobro igrati!

Poglavje 3

Implementacija igralca backgammona z nevronsko mrežo

Načrtovanje igralca backgammona je bilo razdeljeno v dva glavna dela, in sicer v fazo učenja in v fazo testiranja. Prva faza je bila implementacija modela dveh igralcev backgammona, ki igrata drug proti drugemu, in s tem povezano učenje. Drugi del je bila faza testiranja naučenega igralca backgammona, kjer sem ugotovil, kako dobro zna igrati backgammon. Na koncu sem še realiziral uporabniški grafični vmesnik za igranje igre backgammon proti naučenemu igralcu backgammona.

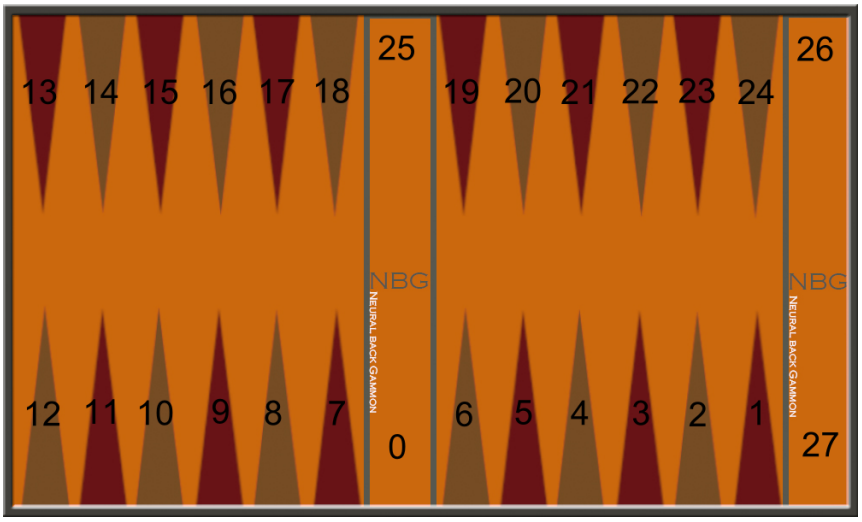
Za implementacijo modela dveh igralcev backgammona sem potreboval definirati okolje - platformo, torej igralno polje, dva nasprotna igralca, ki imata vsak svoji dve kocki, pravila igre in s tem povezan izračun možnih potez. Temu modelu igre je bilo potrebno dodati nevronsko mrežo in logiko za učenje le-te, torej algoritem vzvratnega učenja.

3.1 Implementacija platforme igre

Prva naloga je bila torej implementacija igralnega polja in definirati pravila igre. Nato je bilo potrebno postaviti v to platformo dva nasprotna igralca, ki izmenično mečeta naključni kocki in izbirata poteze iz seznama možnih potez.

Samo polje igre sem poskušal narediti čim enostavnejše in hitro dostopno. Predstavljeno je s celoštevilsko tabelo velikosti 28. Oštevilčenje prikazuje slika 3.1. Beli žetoni so predstavljeni z negativnimi števili in črni s pozitivnimi. Če ni nobenega žetona na polju, ima le-to vrednost 0.

Takšna predstavitev je zelo primerna pri premikanju žetonov. Potrebno je samo prišteti oz. odšteti število, ki ga kaže vržena kocka, k indeksu polja.



Slika 3.1: Oštevilčevanje backgammon polj od 1 do 24. Polji 0 in 25 sta čakalni polji za žetone izbite ven iz igre. Polji 26 in 27 sta končni polji za žetone, ki so zaključili igro.

Zatem sem definiral še dva igralca - črnega in belega. Igralca se razlikujeta le v premikanju žetonov. Črni jih premika od 1. polja proti 24. Beli jih premika ravno v nasprotni smeri.

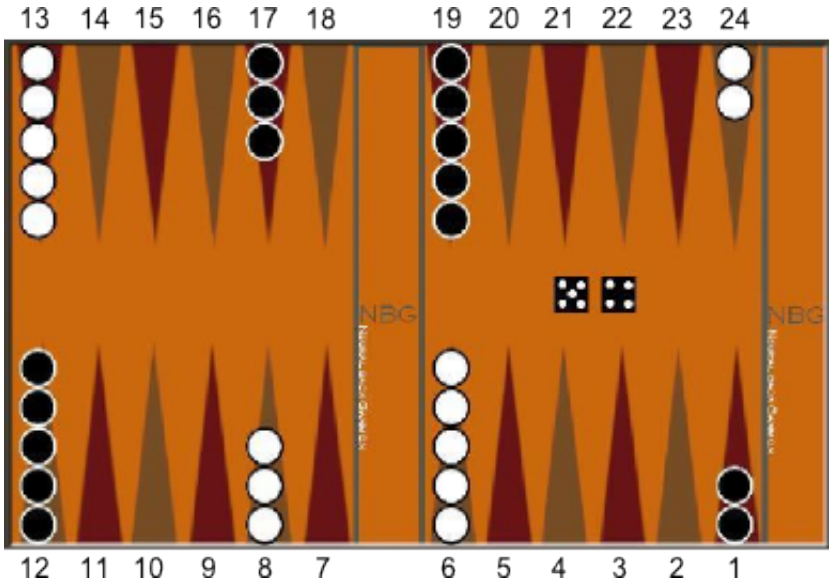
Vsak igralec ima dve neodvisni kocki. Pri vsakem metu se vrednost kocke določi na osnovi psevdo-naključnega generatorja. Seme psevdo-naključnega generatorja je določeno s sistemskim trenutnim časom, ko se ustvari kocko.

Vsak igralec vsebuje tudi seznam vseh možnih potez pri vsakem metu kock.

3.1.1 Izračun vseh možnih potez

Prvi zanimivejši problem, s katerim sem se srečal pri implementaciji igralca backgammona, je bil, kako izračunati vse možne poteze pri metu kock. Primer problema je prikazan na sliki 3.2 in 3.3.

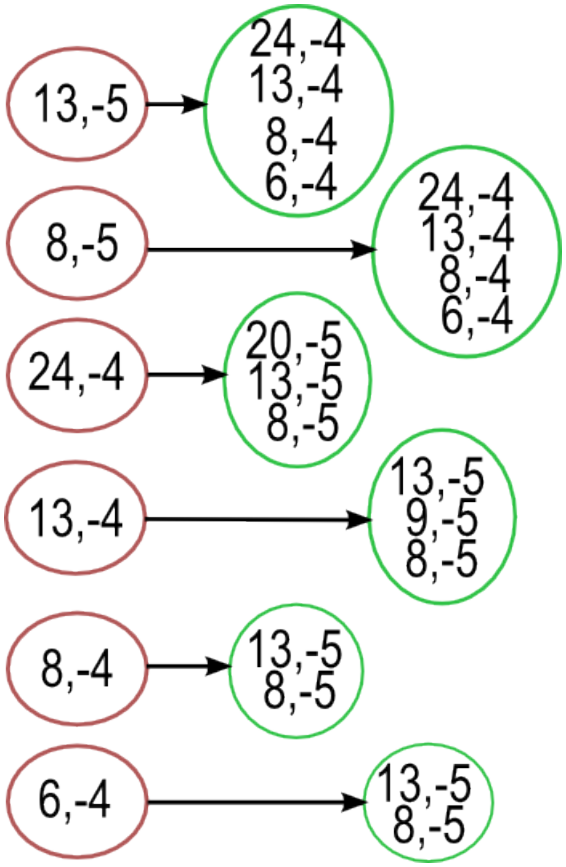
Problema sem se lotil tako, da sem najprej ugotovil, v katerem stanju ima igralec žetone. Ločil sem tri stanja. Poimenujmo jih stanje čakanja, normalno stanje in stanje pospravljanja. Najprej sem preveril, ali je v stanju čakanja. To pomeni, da sem preveril, ali ima igralec, ki je na potezi, mogoče kakšen žeton izbit ven iz igre in čaka na vstop. V tem primeru se ponavadi število možnih potez bistveno zmanjša, ker je najprej potrebno spraviti žeton(e) v igro. Torej



Slika 3.2: V začetnem stanju vrže beli igralec kocki 5 in 4. Poraja se vprašanje, koliko in katere so vse možne poteze, ki jih lahko naredi beli igralec?

igralec mora vreči kocko, ki bo kazala na prosto polje v nasprotnikovi hiši. Dokler tega ne stori, je nabor možnih potez prazen in igralec je prisiljen čakati na naslednji met. Zatem sem preveril, ali je mogoče v stanju pospravljanja. V tem stanju se nahaja, če ima igralec vse žetone v svoji hiši in mogoče že kakšnega na končnem polju. V tem primeru so poleg navadnih potez možne tudi poteze pospravljanja žetonov. Če igralčevi žetoni niso bili niti v stanju čakanja niti v stanju pospravljanja, sem predpostavil, da so v normalnem stanju.

Ko sem enkrat ugotovil, v katerem stanju so, sem začel sestavljati seznam možnih potez. Vzel sem prvo kocko in se sprehodil skozi igralno polje. Kjerkoli sem našel igralčev žeton in ga je bilo možno premakniti za število, ki ga je določala kocka, sem shranil potezo. Nato sem ponovil postopek še z drugo kocko. Tako sem dobil vse možne poteze v prvem koraku. Na primeru iz slike 3.3 bi bile to vse poteze, označene z rdečo barvo. Sedaj je bilo potrebno izračunati še vse možne poteze v drugem koraku. Sprehodil sem se skozi izgrajen seznam možnih potez iz prvega koraka, tako da sem vsako potezo iz seznama izvršil, da sem dobil novo stanje in nato sem se na novem stanju sprehodil skozi celotno polje in spet preveril žetone, ali jih je možno premakniti. Tako sem dobil vse poteze na drugem koraku (na sliki 3.3 so to poteze, označene z zeleno barvo).



Slika 3.3: Možne poteze glede na stanje, ki ga prikazuje slika 3.2. Vseh možnih potez je 18. Rdeča barva prikazuje možne poteze v prvem koraku in zelena barva možne poteze v drugem koraku. Poteza je predstavljena z dvema številkama. Prva pomeni polje žetona, druga pomeni premik ali razdaljo.

V prejšnjem odstavku je podan opis postopka, če sta vrženi 2 različni kocki. Kadar se vržeta 2 enaki kocki, imamo zelo podoben postopek, le da vedno uporabljam isto kocko, ker obe kažeta enako število, in število korakov je 4. Zato je tudi seznam ponavadi bistveno večji, kot če vržemo različni kocki.

Včasih je potrebno seznam „zredčiti“. To storim v 2 primerih. Pravila narekujejo, da je potrebno izkoristiti obe kocki. Torej vse poteze, ki so manjše od maksimalne, je potrebno izbrisati, ker so neveljavne [7]. Naj podam primer. Če obstaja neka poteza, zgrajena iz dveh korakov, je potemtakem poteza z enim korakom neveljavna in jo zato izbrišem iz seznama. Pravila narekujejo tudi, kadar imamo na voljo dve različni potezi z dolžino ena (uporabim lahko

le eno kocko), moramo izbrati potezo z večjo kocko [7]. Torej moram izbrisati potezo z manjšo kocko, če le-ta obstaja v seznamu.

Marsikateri dve različni potezi na sliki 3.3 privedeta na koncu do enakega stanja žetonov. Primer takih potez sta para (13,-5), (8,-4) in (13,-4), (9,-5). Človek bi pomislil, da potemtakem ne potrebujemo obeh potez v seznamu in gledano s stališča same igre, ne bi bilo nič narobe. Pa vendar sem se odločil pustiti obe varianti noter, kajti s tem ima igralec prosto pot izbire. Lahko najprej začne z manjšo kocko in nadaljuje z večjo ali obratno. Tukaj naj dodam še pripombo, da je včasih celo ključnega pomena, s katero kocko začnemo (manjšo ali večjo), saj končni položaj žetonov obeh variant ni enak. V tem primeru je pa nujno, da obdržimo obe možnosti v seznamu.

Seznam beležim v drevesni strukturi. Na ničnem nivoju se nahaja prazna - „korenska poteza“ (ang. „root“), ki vedno obstaja, kadar je seznam prazen ali poln. Nato sledijo poteze na 1. nivoju (koraku), iz teh izhajajo poteze na 2. nivoju. Na 3. in 4. nivoju lahko poteze obstajajo le, če igralec vrže enaki kocki. Kot zanimivost naj povem, da je globina vseh listov drevesa vedno enaka, kajti v nasprotnem primeru poteza z manjšo globino ni veljavna (zaradi takih primerov je bilo potrebno seznam „redčiti“).

Drevesno strukturo sem uporabil, ker nisem vedel vnaprejšnje dolžine seznama. Drugi bistveni razlog pa leži v priročnosti pri gradnji seznama. Namreč, ko hočemo graditi drevo na višjem koraku (poteze na 2., 3. in 4. nivoju), moramo najprej premakniti (realizirati) vse predhodne poteze, da pridemo do želenega stanja. To je pa kot nalašč ustvarjeno za rekurzijo in s tem povezano drevesno strukturo. S sprehodom po drevesu od korena proti listom, v vsakem vozlišču izvedemo potezo, ter ko pridemo na zadnje vozlišče, dodamo nove poteze (listi drevesa).

Edina slabost pri rekurziji je, da je njeno izvajanje nekoliko počasnejše zaradi uporabe sklada, kot če bi seznam realiziral s tabelo.

3.2 Implementacija nevronske mreže

Ko imamo enkrat postavljeno okolje (igralno polje, igralca) in definirana pravila igre, se lahko igra prične. Vendar takoj naletimo na najtežji problem pri igri backgammon: „Katero potezo izbrati iz seznama“?

Ta problem je rešen z nevronsko mrežo, ki nastopa kot evaluacijska funkcija in podaja ocene za možne poteze. Vendar na začetku struktura nevronske mreže nima nobenega znanja o igri backgammon, zato jo je potrebno učiti.

3.2.1 Kodiranje vhodov

Vhodni podatki nevronske mreže (na kratko kar vhodi) so pozicije žetonov na poljih obeh igralcev skupaj s številom pospravljenih in izbitih žetonov, torej trenutno stanje igralnega polja in še informacija, ki pove, kateri igralec je na vrsti.

Najenostavneje bi bilo neposredno pripeljati celo polje (28 celoštevilskih vhodov) v nevronsko mrežo in še en vhod za igralca, ki je na potezi. Izkaže se, da je takšna predstavitev skrajno neugodna za nevronsko mrežo. Problem je, ker se v enem polju vhoda (v eni številki) skriva kar nekaj informacij, za katere bi nevronska mreža porabila precej časa, da jih razvozlja. Na primer, števila 0, 1 in -1 na igralnem polju pomenijo, da na polje lahko pride vsak žeton. Če imamo številko 2 na polju, pomeni, da sta na polju 2 črna žetona in beli ne more priti gor. Torej ena številka nosi v sebi veliko informacij [5].

Črni žetoni					Beli žetoni				
Št. žetonov	4	3	2	1	Št. žetonov	4	3	2	1
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	-1	0	0	0	1
2	0	0	1	0	-2	0	0	1	0
3	0	1	1	0	-3	0	1	1	0
4	0.5	0	1	0	-4	0.5	0	1	0
5	1	0	1	0	-5	1	0	1	0
6	1.5	0	1	0	-6	1.5	0	1	0
...	...	...	...	...	...	...	...	...	...
15	6	0	1	0	-15	6	0	1	0

Tabela 3.1: Tabela kodiranja igralnih polj (1-24)

To je ugotovil že Gerald Tesauro, ko se je ukvarjal z izdelavo TD-gammona, kar je tudi implementacija igralca backgammona z nevronsko mrežo [5]. Sam sem uporabil kodiranje, ki temelji na njegovem. Vsako polje od 1. do 24. je kodirano s 4 števili (tabela 3.1). Iz same kode ni razvidno, ali je zakodiran črni ali beli žeton, saj bi s tem otežil delo nevronski mreži, kar bi se odrazilo na večjem številu potrebnih odigranih iger, da bi ugotovila, ali je kodiran črni ali beli žeton. Da sem se temu izognil, sem podvojil število vhodov in sem kodiral črne posebej in bele posebej, vendar oboje na enak način.

V tabeli 3.1 je predstavljeno kodiranje igralnih polj od 1 do 24. Po vrsti števila predstavljajo: 1. število - ali je žeton sam na polju, 2. število - ali sta 2 in več žetonov na polju, 3. število - ali so 3 žetoni na polju in 4. število - se

obnaša po linearni funkciji, ki narašča skupaj s številom žetonov na polju.

Poleg igralnih polj (1-24) je treba kodirati še 0., 25., 26. in 27. polje iz slike 3.1. Ta polja predstavljajo, koliko in kateri žetoni čakajo na vstop v igro (čakajoči žetoni) in žetoni, ki so končali igro (končani žetoni). Vsako kategorijo žetonov sem kodiral s celim številom, kajti v tem primeru število žetonov ne nosi v sebi še drugih informacij.

Igralec na potezi	2.bit	1.bit
Črni	1	0
Beli	0	1

Tabela 3.2: Kodiranje vrstnega reda igralcev

Mreža potrebuje poleg položaja žetonov tudi informacijo, ki pove, kateri igralec je na vrsti. Vsakega igralca sem kodiral z dvema bitoma, kot to prikazuje tabela 3.2.

Skupno je torej 198 vhodov, ki opisujejo trenutno stanje igre:

- igralna polja (1-24): 2(črni in beli)*4(kodirna števila)*24 = 192
- čakajoči in končani žetoni: 2(črni in beli) * 2 = 4
- vrstni red (določen z dvema vhodoma): 2

3.2.2 Matematični model perceptrona

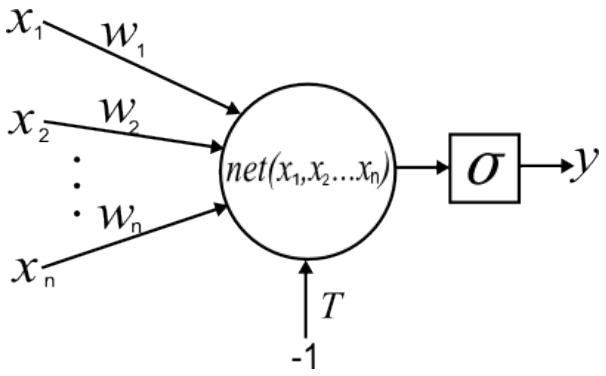
Nevronska mreža je mreža med sabo povezanih nevronov. Matematični model nevrona prikazuje slika 3.4. Na nevron, ki vsebuje pragovno funkcijo (enačba 3.1), pripeljemo vhode **x**.

$$net(x_1, \ldots, x_n) = \sum_{i=1}^{n} \omega_i * x_i - T \tag{3.1}$$

Prag **T** (ali bias **b**) označuje aktivacijsko mejo nevrona, kar pomeni, da ko vhodi skupaj z utežmi ω nevrona dosežejo ali presežejo prag, se nevron aktivira [1]. Izhod nevrona gre še preko sigmoidne funkcije (enačba 3.2).

$$y(net) = \frac{1}{1 + e^{-net}} \tag{3.2}$$

V našem primeru je sigmoidna funkcija še bolj uporabna iz razloga, ker imamo pri izhodu iz mreže dve skrajni vrednosti, in sicer 0 in 1.



Slika 3.4: Matematični model nevrona

Torej če hočemo, da nevron opisuje določeno funkcijo, ga moramo naučiti na to funkcijo. Zelo poenostavljeno rečeno, to dosežemo tako, da to funkcijo velikokrat pošljemo skozi nevron pri različnih vrednostih funkcije in pri tem primerjamo dobljeni izhod z želenim. Glede na razliko med tema vhodoma se s posebnim algoritmom (t.i. vzvratnim algoritmom) pri vsaki ponovitvi malo spremenijo uteži. To počnemo tako dolgo, dokler napaka med dobljenim in želenim izhodom ne postane za nas sprejemljiva.

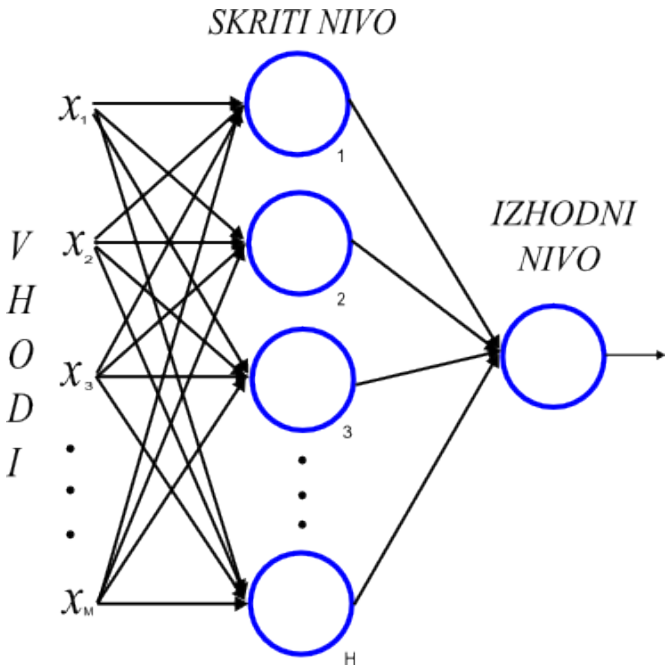
3.2.3 Zgradba nevronske mreže

Z nevronsko mrežo realiziramo evaluacijsko funkcijo, ki nam poda oceno o poziciji žetonov. Uporabil sem 2-nivojski perceptron, ki je prikazan na sliki 3.5. Mreža je sestavljena iz vhodnega nivoja (vhodi), skritega nivoja in iz izhodnega nivoja. Vhodi v nevronsko mrežo so pripeljani do nevronov v skritem nivoju. Tukaj velja opozoriti, da je vsak vhod pripeljan do vsakega nevrona na skritem nivoju.

Število nevronov v skritem nivoju vpliva na natančnost evaluacijske funkcije in posledično tudi na časovno zahtevnost učenja. Torej več nevronov imamo v skritem nivoju, večjo natančnost evaluacijske funkcije dosežemo, vendar porabimo tudi več časa za učenje, zato je potrebno najti kompromis. V mojem primeru je bila velikost skritega nivoja največ 40 nevronov.

Izhod iz nevronske mreže dobimo preko izhodnega nivoja, kjer je vedno samo, vsaj v mojem primeru, en nevron. Izhod je realno število iz intervala [0, 1]. Če je izhod enak 0, je zmagal črni igralec, in če je izhod enak 1, je zmagal beli igralec. Vsaka vmesna vrednost pomeni, da igra še ni končana. Toda evidentno je, da če je vrednost blizu 0, je pozicija žetonov zelo ugodna za

črnega igralca in velja nasprotno, da bližje kot je vrednost 1, bolj je ugodna za belega igralca. Izhodne vrednosti, ki so v območju [0, 1], dosežemo s sigmoidno funkcijo.



Slika 3.5: Nevronska mreža, predstavljena v obliki 2-nivojskega perceptrona. Na levi strani so vhodi. Vsak vhod je pripeljan do vsakega nevrona v skritem nivoju. Na sredini je prikazan skriti nivo in na desni strani je izhodni nivo (izhodni nevron).

Spodnji izrazi prikazujejo izračun evaluacijske funkcije, definirano z nevronsko mrežo s H skritimi nevroni in z M vhodi. Vhod $\mathbf{x}$ je sestavljen iz M vhodov (izraz 3.3) ter iz skalarja -1 z indeksom $M+1$, ki predstavlja vhod pri pragu T. Prag T v nevronu predstavlja utež z največjim indeksom.

$$\mathbf{x} = \{x_1, x_2, ..., x_M, -1\} \tag{3.3}$$

Izraz 3.4 podaja izračun posameznega izhoda na j-tem nevronu skritega nivoja. Indeks j poteka od 1 do H, torej gre čez vse nevrone v skritem nivoju.

$$net_{h_j} = \sum_{i=1}^{M+1} \omega_{ji} * x_i \quad , \quad 1 \leq j \leq H \tag{3.4}$$

Vrednosti net_{h_j} na skritem nivoju gredo skozi sigmoidno funkcijo (izraz 3.5).

$$h_j = \frac{1}{1 + e^{-net_{h_j}}} \quad , \quad j = 1 \dots H + 1 \tag{3.5}$$

Tudi izhodnemu vektorju **h** na konec dodamo skalar -1, ki predstavlja vhod za pragovno funkcijo na skritem nevronu (izraz 3.6).

$$\mathbf{h} = \{h_1, h_2, ..., h_H, -1\} \tag{3.6}$$

Na koncu vse izhode skritega nivoja pripeljemo kot vhode na izhodni nevron (izraz 3.7), nato gre izhod še skozi sigmoidno funkcijo (izraz 3.8).

$$net_y = \sum_{j=1}^{H+1} \omega_j * h_j \tag{3.7}$$

$$y = \frac{1}{1 + e^{-net_y}} \tag{3.8}$$

3.2.4 Spodbujevano učenje in vzvratni algoritem

Ko je celotna struktura postavljena, je treba nevronsko mrežo „pripraviti“, da čimbolj natančno ocenjuje pozicije žetonov. To se doseže s popravljanjem uteži ω v nevronih. Na začetku so vse uteži naključno nastavljene v območju od -0.1 do 0.1, zato tudi evaluacijska funkcija na začetku daje ven povprečne vrednosti. Ne glede na to, kateri igralec je dejansko v boljšem položaju, je ocena na začetku učenja vedno okrog 0.5. Ocena take pozicije pomeni, da imata oba igralca enake možnosti za zmago. Z učenjem se jih popravlja in nastavi do te mere, da so pozicije, ki so zelo blizu zmage črnega, ovrednotene blizu 0 in pozicije blizu zmage belega igralca, ovrednotene blizu 1.

Učenje poteka tako, da igralca backgammona igrata drug proti drugemu. Po vsakem metu se naredi seznam vseh možnih potez. Vsako potezo oz. pozicijo, ki sledi iz poteze, se oceni z evaluacijsko funkcijo - nevronsko mrežo. Izračun ocene podajajo enačbe za izračun evaluacijske funkcije iz prejšnjega razdelka (izrazi od 3.3 do 3.8).

Iz ovrednotenih potez se izbere najugodnejša, t.j. poteza z najboljšo oceno. Če je na vrsti črni igralec, je najboljša poteza z minimalno vrednostjo, sicer če je na vrsti beli igralec, je najboljša poteza z maksimalno vrednostjo. Oceni se tudi trenutno stanje igralnega polja, torej stanje, preden se izvede najboljša poteza. Iz razlike v oceni, ki nastane med trenutnim stanjem in stanjem najboljše izbrane poteze, se spreminjajo uteži na osnovi vzvratnega algoritma. Ta razlika je zelo majhna in s tem se tudi evaluacijska funkcija spreminja

zelo počasi. Bistven preskok v evaluacijski funkciji se zgodi ob zmagi igralca, kajti takrat zasede evaluacijska funkcija mejno vrednost (0 ali 1) in razlika med stanji je večja ter tudi popravek uteži je relativno občuten v primerjavi s popravki med samo igro. Od tukaj izhaja trditev, da je učenje odvisno od števila odigranih iger, in ne od števila odigranih potez.

Torej za učenje in s tem povezano popravljanje uteži uporablja mreža vzvratni algoritem, ki ga opisujejo spodnje enačbe. Vzvratni algoritem v vsakem koraku n popravi uteži po enačbi 3.9.

$$\omega(n+1) = \omega(n) + \Delta\omega(n) \tag{3.9}$$

$\Delta\omega(n)$ je definirana z izrazom 3.10, torej z odvodom napake po uteži.

$$\Delta\omega(n) = -\eta\frac{\partial E(n)}{\partial\omega(n)} \tag{3.10}$$

V izrazu nastopa tudi spremenljivka η, s katero povemo, kolikšen delež spremembe uteži upoštevamo. η bistveno vpliva tudi na hitrost učenja, vendar moramo biti pozorni, da ni prevelik. V tem primeru se uteži popačijo in ne kovergirajo k rešitvi. Sam sem uporabljal η ranga do 0.1. Napako izračunamo (izraz 3.11) na podlagi ocene trenutnega stanja $y(n)$ in ocene naslednjega stanja $y(n+1)$.

$$E(n) = \frac{1}{2}(y(n+1) - y(n))^2 \tag{3.11}$$

Če bi poznali želeni izhod, bi učenje igralca backgammona potekalo po principu nadzorovanega učenja. Toda v našem primeru ne poznamo želenega izhoda. Ko izberemo eno potezo, dobimo lahko le odgovor, ali je poteza dobra ali slaba. Ta odgovor se izkaže v obliki zmage ali poraza igralca. Za nameček je odgovor še zakasnjen, torej ga dobimo čez določeno število korakov - ob koncu igre. Zato je uporabljen t.i. spodbujevani način učenja - TD(λ), kjer je λ=0 (enačba 3.12) [4].

$$\Delta\omega(n) = \eta * (y(n+1) - y(n)) * \frac{\partial y(n)}{\partial\omega(n)} \tag{3.12}$$

Za želeni izhod vzamemo vrednost, ki jo predstavlja najboljše ocenjena poteza izmed vseh možnih. Ta je lahko v resnici dobra ali slaba. To informacijo dobi nevronska mreža ob koncu igre. Skozi vzvratni algoritem se mreži primerno, glede na izbrano potezo, spreminjajo uteži. Tako se mreža skozi veliko število iger nauči primerno ocenjevati poteze, glede na dano situacijo.

Popravljanje uteži se začne na izhodnem nivoju in se nadaljuje na skritem, torej poteka nazaj (od tu tudi ime vzvratno učenje), gledano iz zgradbe mreže. Izraza 3.13 in 3.14 opisujeta izračun spremembe uteži na izhodnem nivoju.

$$\delta_y = (y(n+1) - y(n)) * y(n) * (1 - y(n)) \tag{3.13}$$

$$\Delta\omega_j = \eta * \delta_y * h_j \quad , \quad 1 \leq j \leq H \tag{3.14}$$

Število uteži na izhodnem nivoju je enako številu skritih nevronov H. Le zadnja utež v nevronu (prag z indeksom $H + 1$) se računa je malo drugače (izraz 3.15), in sicer ne uporabimo $H + 1$ izhoda iz skritega nivoja, ker ga nimamo, ampak kar konstanto -1.

$$\Delta\omega_{H+1} = -\eta * \delta_y \tag{3.15}$$

Nato lahko pričnemo z izračunom sprememb uteži na skritem nivoju (enačbe 3.16, 3.17 in 3.18). Indeks j gre po vseh skritih nevronih od 1 do H. Indeks i poteka po vseh vhodih od 1 do M.

$$\delta_{h_j} = \delta_y * \omega_j * h_j * (1 - h_j) \quad , \quad 1 \leq j \leq H \tag{3.16}$$

$$\Delta\omega_{h_{ji}} = \eta * \delta_{h_j} * x_i \quad , \quad 1 \leq j \leq H \quad , \quad 1 \leq i \leq M \tag{3.17}$$

$$\Delta\omega_{h_{j,M+1}} = -\eta * \delta_{h_j} \quad , \quad 1 \leq j \leq H \tag{3.18}$$

Ob učenju sem naletel na zanimiv podatek, ki se nanaša na število potez. Dva dobra igralca backgammona porabita na igro v povprečju okrog 50 potez. Nevronska mreža igralca backgammona na začetku učenja, ko je še brez znanja, porabi za odigrano igro tudi do nekaj 1000 potez, vendar začne število potez na igro dokaj hitro upadati in dosežeta število 50 potez na igro.

Nevronska mreža s 40 skritimi nevroni se je učila, da je odigrala 4 milijone iger, kar je trajalo približno 10 dni.

Poglavje 4

Testiranje igralca backgammona

Po opravljenem učenju je potrebno nevronsko mrežo testirati. Načeloma je testiranje mogoče opraviti, tako da človek odigra določeno število iger proti naučeni nevronski mreži. Vendar bi zaradi naključnih metov kock bilo potrebno odigrati veliko število iger, da se dobi realno oceno, kako dobro se je mreža naučila igrati, kar pa seveda spet pomeni veliko časa in truda.

Dosti bolj ekonomična rešitev v smislu časa je, da se kreira preprost igralec backgammona, ki igra na določenemu nivoju. Ena varianta je bila, da se testira proti naključnemu igralcu, t.j. igralcu, ki naključno izbira potezo iz seznama. Izkaže se, da je takšen igralec zelo slab in igralec z zelo malo znanja ga je premagoval 100%. To pa pomeni, da bo rezultat testiranja pri zelo dobrem in pri povprečnem igralcu enak, torej ne bo vidna nobena razlika. Potemtakem bi potreboval igralca, ki zna igrati solidno in ga ni možno vedno premagati.

4.1 Evaluacijska funkcija pubeval

Gerald Tesauro je zasnoval zelo preprosto evaluacijsko funkcijo za ocenjevanje pozicije igre backgammona. Poimenoval jo je „pubeval“ (kratica za „public evaluation“) in je prosto dostopna [3].

V bistvu gre za linearno funkcijo. Določene so uteži, ki se jih pomnoži z vhodi in dobi se ocena za pozicijo (enačba 4.1).

$$\text{ocena}_\text{p} = \mathbf{X} * \mathbf{W}_v \tag{4.1}$$

Vhodi $\mathbf{X}$ so podobni kot pri nevronski mreži in predstavljajo trenutno pozicijo žetonov p. Loči dve vrsti uteži $\mathbf{W}$. Ene se uporabi, kadar so izbijanja možna ($v = 1$), druga ($v = 2$), kadar izbijanj več ni in se samo še tekmuje,

kdo pospravi prej svoje žetone. Tako kot pri učenju nevronske mreže, se za najboljšo potezo izbere tista z najvišjo oceno.

Igralec pubeval je srednje dober igralec. Pozna vse osnovne elemente igre, kot so izbijanje žetonov, gradnja novih polj, itd. Pozna tudi naprednejše principe, kot so postavljanje blokade, pokrivanje svoje hiše, da igralec ne more ven z žetonom, če ga ima na čakanju. Manjkajo mu mogoče statistično gledano dobre odločitve, torej poteze, ki mogoče na prvi pogled niso logične, se pa čez čas izkažejo kot odločilne. Vsekakor gre za solidnega igralca, ki ga ni tako enostavno premagati, še posebej če nimaš sreče s kockami.

4.2 Rezultati testiranja

Testiranje je potekalo hkrati z učenjem mreže. Sicer bi bilo možno opraviti testiranje po končanem učenju, vendar je zelo zanimivo opazovati sam potek učenja in celo izkaže se, da je možno najti vmesni rezultat, ki bo za odtenek boljši od končnega.

Idealno bi bilo testirati naučeno mrežo po vsaki opravljeni igri, vendar bi bilo takšno početje zelo zamudno, kajti že samo učenje traja dovolj dolgo. Tako je testiranje potekalo po intervalih, ponavadi na 10000 naučenih iger.

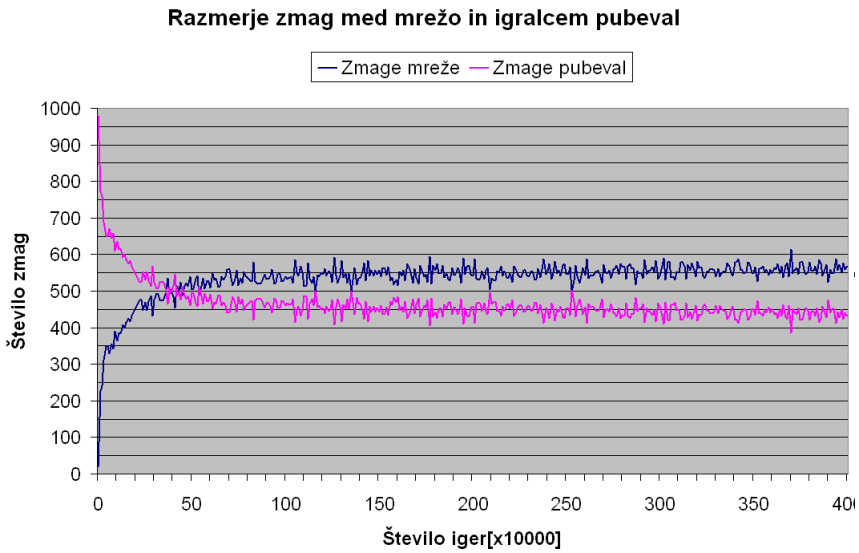
Za testiranje naučene nevronske mreže je bilo potrebno odigrati čim večje število iger, da je bil test tem bolj verodostojen. To je posledica metanja naključnih kock - enkrat so kocke bolj ugodne za enega igralca, drugič za drugega. Izkazalo se je, da je rezultat pri 1000-ih odigranih igrah že zelo zanesljiv. To pomeni, da so pri ponovljenem testiranju zelo majhna odstopanja - kvečjemu za slab procent ali nekaj promilov, kar je sprejemljivo.

Najdaljše učenje je znašalo 4 milijone odigranih iger. Mreža je imela 40 skritih nevronov. Ta številka je bila nekako optimalna glede na pogoje učenja in testiranja. Tu imam predvsem v mislih čas in računalniško opremo - torej računalnik, na katerem se je izvajalo učenje. Učenje je potekalo na računalniku tipa Intel Core 2CPU (2.66 GHz, 2GB RAM) in je trajalo približno 10 dni.

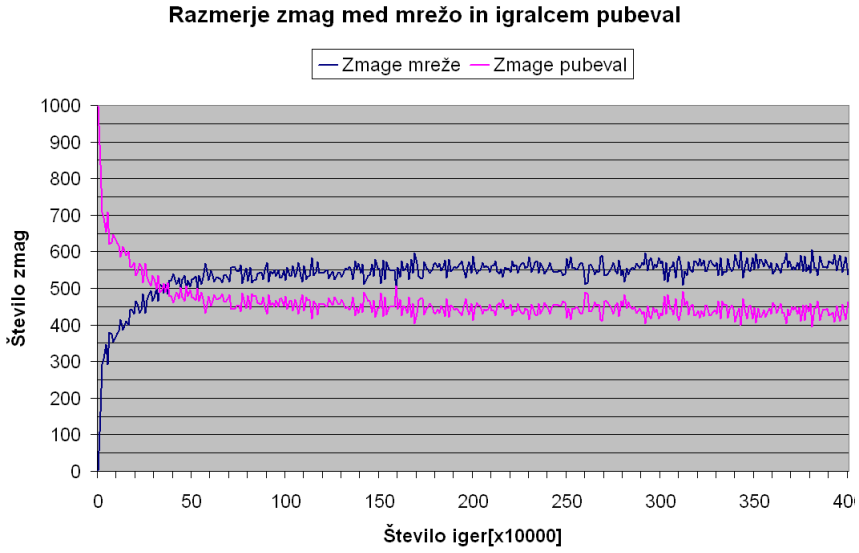
Učilni parameter η pada linearno in je na začetku učenja velikosti 0.05 (izraz 4.2), kjer je N končno število iger in n trenutni korak.

$$\eta_{n+1} = \eta_n * \frac{N-n}{N} \tag{4.2}$$

Grafa 4.1 in 4.2 predstavljata potek takšnega učenja. Gre za dva povsem ločena poteka učenja, pa vendar je rezultat praktično identičen. S tem želim poudariti, da bo ne glede na začetno stanje uteži v nevronski mreži, rezultat mreže, torej znanje, enako.



Slika 4.1: Graf učenja nevronske mreže in testiranje proti igralcu pubeval. Največje število zmag je 614 pri 3700000 odigranih igrah.



Slika 4.2: Graf učenja nevronske mreže in testiranje proti igralcu pubeval. Največje število zmag je 605 pri 3810000 odigranih igrah.

Na obeh grafih sta izrisani po dve krivulji. Krivulji namreč predstavljata število zmag določenega igralca pri 1000 testnih igrah, torej frekvenco zmag. Padajoča krivulja pripada testnemu igralcu pubeval in naraščujoča krivulja pripada naučeni mreži igralca backgammona.

Pred začetkom učenja, torej v točki 0, sta krivulji najbolj oddaljeni. To je razumljivo, saj mreža še ne vsebuje nobenega znanja. Zato tudi mreža ne premaga igralca pubeval skoraj nikoli, tako se število dobljenih iger (zmag) giblje od 0 do 10 od možnih 1000, kar znaša slab procent.

Takoj, ko nastopi učenje, se prične krivulja hitro spreminjati. Število zmag mreže začne hitro naraščati in število zmag igralca pubeval simetrično upada. V tej fazi se mreža nauči osnovnih principov igranja igre backgammon, kot so izbijanje žetonov in gradnja novih polj. To se tudi odrazi na grafu, saj se število dobljenih zmag mreže strmo povečuje.

Krivulji tako s hitrim naraščanjem in simetričnim padanjem prideta do točke, kjer sta enaki. To se zgodi približno po 400.000 do 450.000 odigranih igrah. V tem delu je znanje mreže nekako ekvivalentno testnemu igralcu pubeval, saj je razmerje v zmagah 1:1.

Potem pa mreža narašča čedalje bolj počasi in zgleda, da konvergira približno proti 600 zmagam ali 60%. Vsekakor se učenje skorajda čisto ustavi ali pa tako počasi napreduje, da je skoraj neopazno. To pomeni, da daljše učenje, kot recimo v mojem primeru 4 milijone iger, bi bilo smiselno le v primeru, če bi bilo učenje bistveno daljše, recimo 20 milijonov iger ali več. Za kaj takega pa ni bilo pogojev (predvsem časa), tako da sem ustavil učenje pri številki 4.000.000.

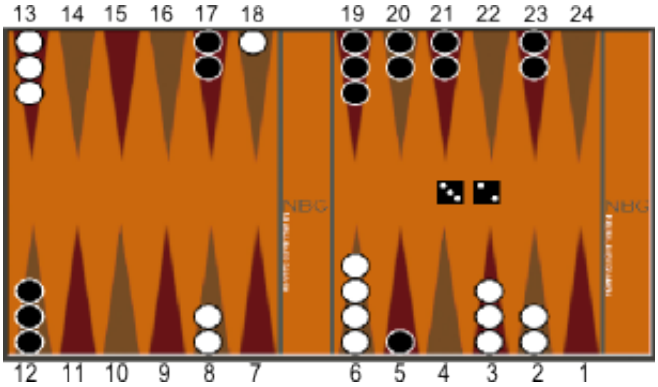
4.3 Osvojeno znanje

Nevronska mreža se je naučila igrati igro backgammon na dokaj zahtevnem nivoju. Povprečen igralec backgammona bo s težavo premagal nevronsko mrežo in bo za to potreboval dosti sreče pri metu kock.

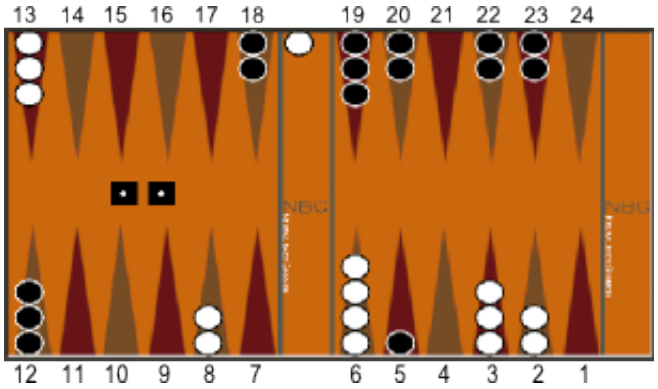
Spodnje slike (4.3, 4.4, 4.5 in 4.6) prikazujejo nekatere izmed osnovnih elementov igre backgammon, kot tudi naprednejše elemente igre (slike 4.7, 4.8, 4.9 in 4.10), ki se jih je nevronska mreža uspešno naučila. V vseh prikazanih primerih igra nevronska mreža s črnimi žetoni, nasprotnik pa z belimi.

Izbijanje žetonov spada med najvažnejše elemente igre backgammon. Če nasprotniku izbiješ žeton, pomeni, da so bile vse kocke, ki jih je vrgel, da je spravil žeton do dane pozicije, zaman. Pri tem pa mora igralec žeton še spraviti v igro, kar pomeni še dodatno oviro zanj. Torej izbiti žeton nasprotniku,

pomeni, da mu povzročiš dosti „škode“. Narejena „škoda“ je tem večja, čim bližje k svoji hiši je nasprotnik spravil žeton - je bistvena razlika ali je žeton izbit v nasprotnikovi ali naši hiši. Obstajajo tudi situacije, kjer ni pametno izbiti prostega nasprotnikovega žetona, ker lahko s tem več škodujemo sebi kot nasprotniku, vendar so to zelo specifične situacije. Vsekakor je tehnika izbijanja žetonov nujna za osnovno igranje backgammona.



Slika 4.3: Beli igralec je pustil na 18. polju en žeton, ki je izpostavljen za izbijanje.

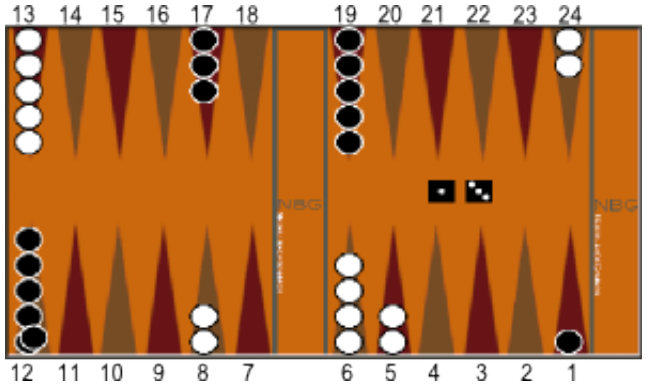


Slika 4.4: Črni izbije belemu igralcu žeton iz 18. polja s kockami 1 in 1.

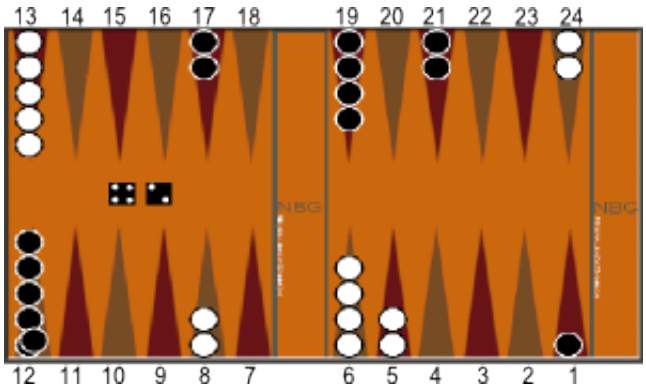
Sliki 4.3 in 4.4 prikazujeta pozicijo žetonov pred izbitjem in po izbitju žetona. Ko črni igralec izbije bel žeton, dobi črni občutno prednost, saj samo ob kockah 1 ali 4 beli pride nazaj v igro, medtem pa lahko črni še zgradi kakšno novo polje. Ob vseh možnih kombinacijah, ki jih ima na voljo črni igralec, ki se dokaj hitro povzpno na številko 1000, kadar se vržejo manjše enake kocke

(npr. 2 in 2 ali 1 in 1), se ravno odloči za potezo, s katero izbije nasprotnikov žeton. S tem nevronska mreža dokazuje, da je osvojila igranje z izbijanjem žetonov.

Ravno tako kot izbijanje, je tudi gradnja polj bistveni element za uspešno igranje backgammona. Gradnja novega polja pomeni, da igralec spravi na novo polje 2 ali več žetonov. Samo z gradnjo novih polj, lahko igralec spravlja žetone varno čez polja. Torej v tem primeru izbijanje ni možno. Seveda to vedno ni mogoče, sploh pa da bi celo igro tako odigrali. Včasih je celo škodljivo, ker nas pripelje do takšnih situacij, kjer bomo prisiljeni k razbitju dveh polj, vendar so to spet posebne situacije.



Slika 4.5: Pozicije žetonov pred izgradnjo novega polja.

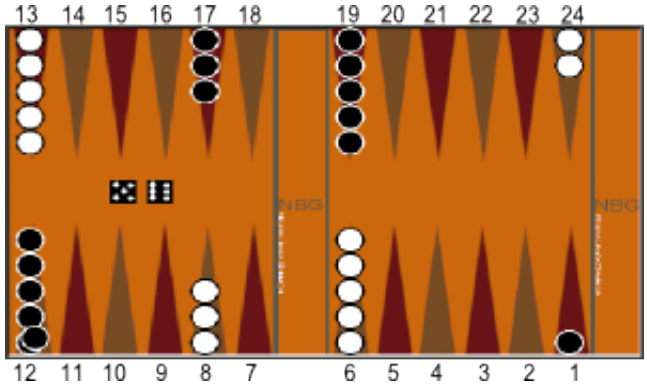


Slika 4.6: Črni igralec glede na kocki 4 in 2 izgradi novo polje na 21. polju.

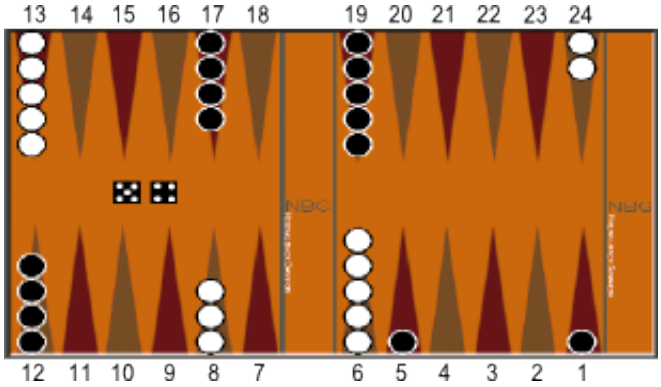
Sliki 4.5 in 4.6 prikazujeta, kako črni igralec zgradi novo polje na 21. polju.

Spet je imela nevronska mreža veliko možnosti za odigrati, pa vendar se je odločila za potezi, ki izgradita novo polje.

Sliki 4.7 in 4.8 prikazujeta otvoritve. Otvoritev je pri backgammonu izjemno pomembna, saj z njo igralec nakaže, kako bo začel igrati in katero taktiko bo ubral. Ob dobri otvorivi in malo sreče pri metu kock je zmaga že zelo blizu. Otvoritve več ne sodijo med osnovne elemente igre backgammon, ampak že med malo bolj zahtevne.



Slika 4.7: Odpiranje črnega igralca pri metu kock 5 in 6. Črni igralec (nevronska mreža) je pospravil žeton na „varno“. Naredil je zaporedni potezi 1-6 in 6-12.

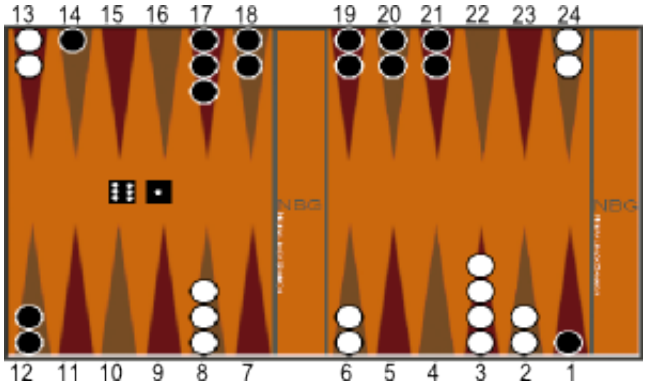


Slika 4.8: Odpiranje črnega igralca pri metu kock 5 in 4. Črni igralec (nevronska mreža) je premaknil en žeton iz ene „varne točke“ na drugo „varno točko“ (poteza 12-17) ter en žeton postavil na zelo pomembno polje, t.j. polje številka 5 v beli hiši (poteza 1-5).

Če igralec vrže kocki 5 in 6 ob otvoritvi, ima teoretično gledano možnost, da pride s potezami do 8 novih stanj. Od teh so 3 dokaj pogosto igrane, ostalih

5 pa skoraj nikoli. Situacija iz slike 4.7 prikazuje 1 od treh najbolj pogosto odigranih potez, kar dokazuje, da mreža igra pametno. Tudi situacija na sliki 4.8 prikazuje zelo inteligentno odigrano potezo, saj je ena najbolj pogostih, če ne edina logična odigrana poteza. Če mu beli igralec ne izbije žetona, bo imel črni dobre možnosti, da ob naslednji potezi pokrije 5. polje v beli hiši, ki je izrednega strateškega pomena. Če pa beli izbije črni žeton na 5. polju, to ne predstavlja velike izgube za črnega, saj izbiti žeton še ni blizu svoje hiše, pa tudi nazaj v igro ga ne bi bilo problem spraviti, ker je še več kot polovica polj v beli hiši prostih. Torej z majhnim tveganjem lahko ogromno pridobi in zelo malo izgubi.

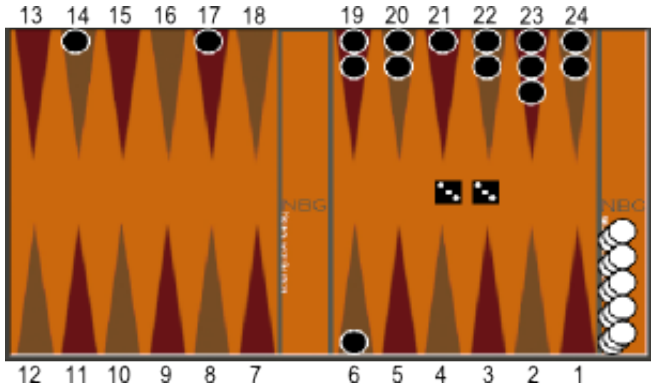
Eden od naprednejših elementov igre je tudi postavljanje blokade. Blokada pomeni, da zasedemo čim več zaporednih polj, tako da nasprotnik s težavo pride čez. Če je zasedenih 6 zaporednih polj, potem nasprotnik sploh ne more čez. Blokada je tudi tembolj koristna, čimveč polj v hiši zaseda, kajti v končni fazi, kadar zaseda celo hišo, nasprotnik sploh ne more priti na vrsto pod pogojem, da ima kakšen žeton na čakanju. Tudi tehniko blokade se je nevronska mreža naučila. Uporablja jo dokaj pogosto, sploh če ji damo možnost zanjo. Na sliki 4.9 je tako prikazana blokada dolžine 5 od 17. do 21. polja. V tem primeru ima beli igralec zelo malo možnosti za končno zmago, saj ima 2 žetona izza blokade (na polju 24).



Slika 4.9: Črni igralec je postavil blokado dolžine 5 od 17. polja naprej.

Zelo pomemben element igre je tudi pospravljanje žetonov, kjer igralec poskuša optimalno porabiti pike pri metu kock. Zelo koristno je dobro razvrščanje žetonov v domači hiši, kar pomeni, da se žetone poskuša čimbolj enakomerno razporediti po vseh poljih; nasprotno bi bilo, da se gradijo stolpci. Pri tem elementu igre sem opazil, da mreža včasih naredi kakšno očitno napako.

To ni napaka pri razporejanju žetonov, ampak pri dajanju prioritete razporejanju pred pospravljanjem. To se še izrazito pokaže, kadar mreža izgublja in ima razporejene žetone še po vseh 4 kvadrantih, kot to prikazuje slika 4.10. V tem primeru je imel črni igralec 1 žeton ujet in je bil na čakanju. V tem času je beli pospravil že večino svojih žetonov, preden je črni spravil svojega v igro. Še vedno pa je imel črni igralec dovolj časa (dovolj metov), da bi ga spravil ven iz bele hiše, kajti evidentno je, da bo igro izgubil. Vsekakor se pa da poraz ublažiti oz. zmanjšati nasprotnikovo zmago iz 3 (backgammon) na 2 (gammon) točki s tem, da spravi črn žeton (na 6. polju) iz bele hiše. Črni igralec pa je dal prioriteto razporejanju žetonov v svoji hiši, kar je sicer lepo in prav, ampak do takrat, kadar bo to uporabno - pri pospravljanju, bo igra že zdavnaj zaključena.



Slika 4.10: Črni igralec je dal prednost razvrščanju pred umikom. Ob končani igri mu ostane žeton v nasprotnikovi hiši (6. polje) - doživel je najtežji možni poraz.

Ta napaka je po svoje tudi razumljiva, saj nevronski mreži noben vhod ne pove, koliko točk dobi zmagovalec za zmago. Tukaj bi bilo še mesta za izboljšavo, ki bi preprečila hude poraze, torej backgammone.

Gledano v celoti se je mreža naučila veliko. Ne samo osnovne principe igre, temveč obvlada tudi kompleksnejše elemente igre kot so odpiranje, postavljanje blokad in razporejanje žetonov pri pospravljanju. Včasih se zgodi tudi kakšna nelogična poteza, ki je soliden igralec backgammona nikakor ne bi naredil, vendar je teh napak zelo malo. Dostikrat so to takšne poteze, ki izgledajo na prvi pogled nerazumljive, vendar bi se statistično gledano ta poteza lahko izkazala za zelo dobro. Ravno to je po drugi strani tudi glavna karakteristika nevronske mreže. Poteze, ki jih odigra, so vedno zelo optimalne gledano iz vidika statistike. Torej, če dano potezo odigramo pri recimo velikem številu

ponovitev situacije, bi se izkazalo, da je pri večini ponovitev poteza zelo v redu odigrana.

Tukaj naj omenim še eno bistveno prednost naučene mreže pred človeškim igralcem backgammona. Backgammon se dostikrat igra do 21, 40, 51 in celo več dobljenih točk. To so časovno zelo dolge partije, lahko trajajo tudi po več dni (s prekinitvami in pavzami seveda). Človeška slabost se tukaj odraža kot padec koncentracije, ki poveča verjetnost delanja napak. Tega problema nevronska mreža nima in izkaže se, da so rezultati pri dolgih igrah boljši kot pri kratkih, kadar je nasprotnik človek [5].

Poglavje 5

Grafični vmesnik za igranje backgammona - NBG

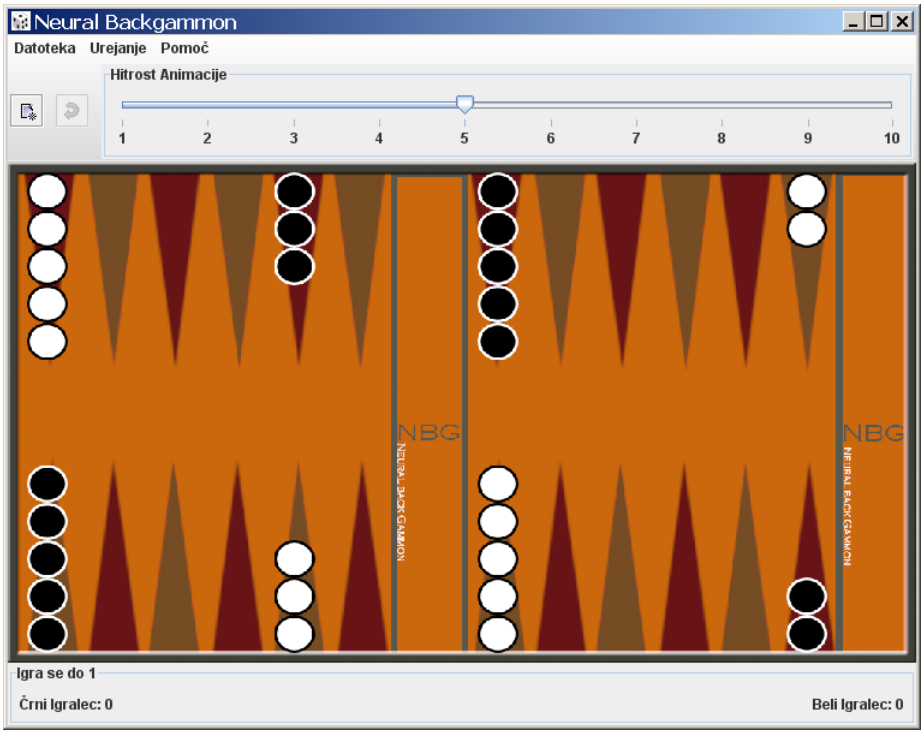
Cilj diplomske naloge je bil tudi predstaviti na nazoren način uporabniku naučeno nevronsko mrežo igralca backgammona. To je bilo doseženo z aplikacijo, ki uporabniku omogoča igranje proti naučeni nevronski mreži, da jo lahko sam preizkusi, kako dobra je. Aplikacijo sem poimenoval NBG (Neural BackGammon).

Aplikacija je narejena v programskem jeziku Java (verzija 6.16). Razvoj in testiranje aplikacije je potekalo z urejevalnikom Eclipse.

Uporabnik ima na voljo 2 načina aplikacije. Prva možnost je igranje proti naučeni mreži, druga možnost pa je učenje mreže. Privzeti način je igranje proti naučeni nevronski mreži.

Slika 5.1 prikazuje začetni prikaz aplikacije NBG. Uporabnik vidi meni, orodno vrstico, igralno površino in statusno vrstico. Meni sestavljajo 3 razdelki: „Datoteka“, „Urejanje“ in „Pomoč“. Pod razdelkom „Datoteka“ sta na voljo 2 opciji: „Nova Igra“, če želimo pričeti novo igro, in „Zapri“, če želimo zapreti aplikacijo. Pod razdelkom „Urejanje“ sta na voljo 2 opciji: “Razveljavi“, ki je zelo priročno pri igranju, kadar se zmotimo in želimo razveljaviti poteze, in „Možnosti“, kjer določimo parametre za igranje in učenje. Pod razdelkom „Pomoč“ sta na voljo 2 opciji: „Pomoč“, kjer je na kratko razloženo ravnanje z aplikacijo, in „O NBG“, kjer so zapisane osnovne informacije o aplikaciji.

Pod opcijo „Možnosti“ (slika 5.2), ki se nahaja v meniju, če izberemo „Urejanje“ in nato „Možnosti“, uporabnik lahko nastavlja različne parametre za igranje in za učenje. Najprej se seveda mora uporabnik odločiti, ali bo igral igro ali pa hoče učiti nevronsko mrežo. To določimo z izbirnim gumbom nad izbranim razdelkom: „Igra“ ali „Učenje“. V ta namen je okno



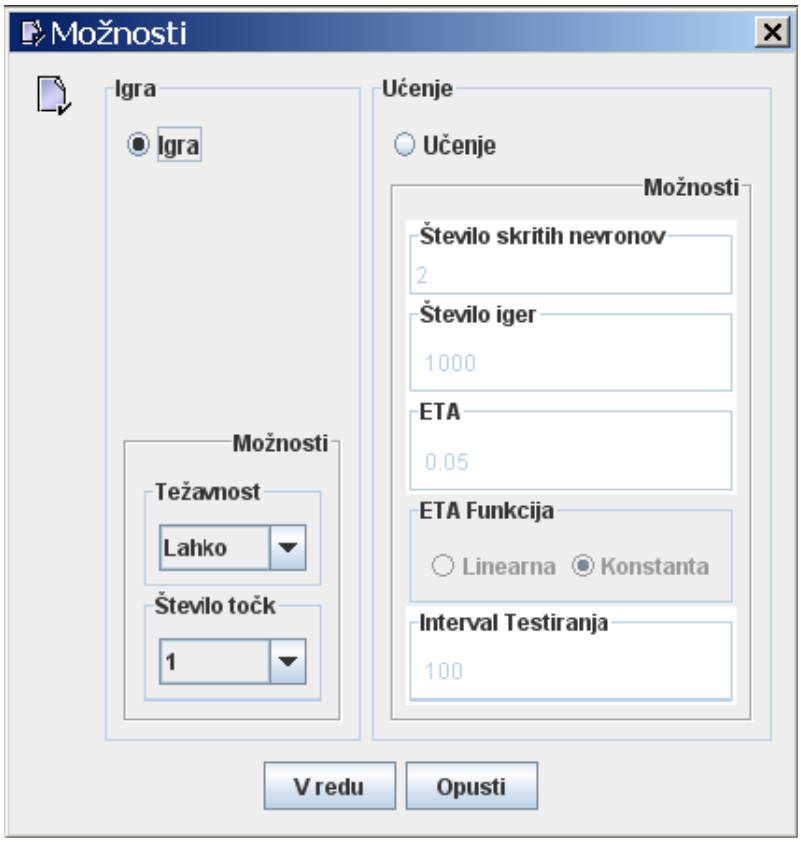
Slika 5.1: Začetni prikaz aplikacije NBG

„Možnosti“ razdeljeno na dva dela. Na levi strani se nahajajo parametri za igranje, na desni pa za učenje. Pod razdelkom „Igra“ lahko uporabnik nastavi 2 parametra: stopnjo težavnosti (level) in določi število točk, ki so potrebne za zmago. Privzeta izbira za težavnost je „Lahko“ in za število točk za zmago je potrebna 1 točka. Pod razdelkom „Učenje“ ima uporabnik na voljo nastaviti 4 oz. 5 parametrov. Določi lahko velikost mreže, t.j. število nevronov v skritem nivoju. Določi lahko število iger, ki jih mreža odigra, da se nauči igrati backgammon. Določi lahko velikost parametra η, kot tudi funkcijo parametra η skozi učenje (izbira med konstanto ali linearno funkcijo). Na koncu lahko še določi interval testiranja mreže, t.j. čez vsakih koliko iger se naučeno mrežo testira (odigra 1000 iger) proti igralcu pubeval. Privzete nastavitve so: 2 skrita nevrona, 1000 iger, η ima vrednost 0.05 in je konstanta ter interval testiranja 100.

Težavnostni nivoji so opredeljeni glede na rezultate testiranja nevronske mreže z igralcem pubeval. Level „Lahko“ je predstavljen z nevronsko mrežo, ki premaguje 40% igralca pubeval, level „Srednje“ z mrežo, ki premaguje 50%

igralca pubeval in level „Težko“ z mrežo, ki je dosegla najboljši rezultat, t.j. 61.4% premagovanje igralca pubeval.

Igro pričnemo s klikom v prazno polje. S tem vržemo kocki. Igro začne tisti, ki vrže večje število točk. Kocki črnega igralca (nevronske mreže) so na levi strani in kocke belega igralca (uporabnika) so na desni strani. Prikazane so le kocke tistega, ki je na vrsti. Ko igralec (kateri koli) zaključi s potezami, potrebuje uporabnik klikniti na prazno polje ali kocke. S tem potrdi poteze in da znak, naj se igra nadaljuje. Ob zaključku igre je igralec obveščen, kdo je zmagal.



Slika 5.2: Okno „Možnosti“, kjer določimo parametre za igranje in učenje.

Ob zagonu aplikacije NBG se v domačem uporabnikovem imeniku kreira mapa z imenom NBG. Pri učenju se uteži nevronske mreže po vsakem intervalu testiranja shranijo znotraj mape NBG, kot tudi rezultati testiranja. Znotraj mape NBG se ustvari nova mapa za vsako učenje. Mapa se poimenuje

glede na parametre učenja: H*hn*ETA*eta*C/L, kjer *hn* predstavlja število skritih nevronov, *eta* predstavlja velikost parametra η in na koncu je dodana črka C, ki pomeni, da je izbrana konstanta za parameter η, ali črka L, ki pomeni, da je izbrana linearna funkcija parametra η. Tako uporabnik loči med večimi različnimi učenji. Če je učenje izvedeno 2 krat ali večkrat z enakimi parametri, se obstoječe prepiše z zadnjim izvajanjem. Uteži so shranjene za vsako testiranje posebej v datoteki *_stevilo.weights*, kjer *stevilo* pomeni število odigranih iger, kdaj se je izvedlo testiranje. Rezultati vseh testiranj za vsako učenje so shranjeni v datoteki z imenom *Properties.txt*.

Pri učenju se v oknu aplikacije ne dogaja nič. Uporabnik samo dobi ovestilo, da poteka učenje. Se pa v urejevalniku Eclipse v oknu *Console* izpisuje trenutno stanje učenja. Izpis obsega trenutno razmerje zmag med črnim in belim igralcem. Izpis se na vsakem intervalu testiranja ustavi, ker poteka testiranje mreže. Ko je testiranje opravljeno, se ponovno nadaljuje izpis učenja.

Uporabnik ima na voljo tudi orodno vrstico, kjer lahko na hitro sproži določene akcije pri igranju. Na voljo sta 2 gumba (v vrstnem redu od leve proti desni) „Nova Igra“ ter „Razveljavi“. Desno od gumbov se nahaja še drsnik „Hitrost Animacije“, s katerim uporabnik spreminja hitrost animacije. Večja kot je vrednost na drsniku, hitreje poteka animacija. Privzeta vrednost drsnika je 8.

Uporabnik vidi tudi statusno vrstico, ki se nahaja pod igralno ploščo. Na njej je izpisano, koliko točk je potrebnih za zmago igralca, kot tudi trenutni rezultat.

Aplikacija NBG podpira tudi večjezičnost, in sicer slovenščino in angleščino. Vendar izbira jezika ni neposredno odvisna od uporabnika in je tudi ne more izbrati direktno, temveč je odvisna od jezika platforme - operacijskega sistema. Torej, če se aplikacija izvaja na platformi s slovenskim jezikom, bo izbrani jezik avtomatično slovenščina, v vsakem drugem jeziku na platformi pa bo privzeti jezik aplikacije NBG angleški jezik.

Poglavje 6

Sklepne ugotovitve

Z diplomsko nalogo sem izpolnil zadani cilj in sem tudi zadovoljen z rezultati, saj se je nevronska mreža brez kakršnega koli vnaprejšnjega znanja naučila igrati igro backgammon na dokaj visokem nivoju. Vendar sem prepričan, da se je nevronska mreža sposobna naučiti še več in s tem izboljšati igro.

To bi bilo možno narediti na več načinov. Eden od teh bi bil, da mreža že ima pred učenjem določeno znanje o igri backgammon in vedno, ko začnemo učiti mrežo, samo povečamo končno število iger in nadaljujemo tam, kjer smo ostali nazadnje. Vendar sem mnenja, da takšno učenje ne bo prineslo nekega bistvenega preskoka v igri. Sam vidim razvoj v smeri, da nevronski mreži vključimo med vhodne podatke še dodatne parametre, kot so npr. število pik, ki jih dobi zmagovalec, s tem bi mrežo naučili, da loči med tremi načini zmag (normalna, gammon in backgammon). Za popolno igro backgammona bi lahko vključili tudi kocko za podvajanje. S tem bi povečali kompleksnost igre, vendar bi šele takrat igralec lahko okusil užitek igranja backgammona v polni meri.

Za učenje in testiranje nevronske mreže bi bila potrebna skrajna optimizacija kode, saj je tako ali tako že sam algoritem vzvratnega učenje zelo časovno zahteven, čim večje je število nevronov v skritem nivoju. Priporočljivo je imeti tudi zelo zmogljivo strojno opremo, na kateri bo potekalo učenje, da bo čas učenja čim krajši.

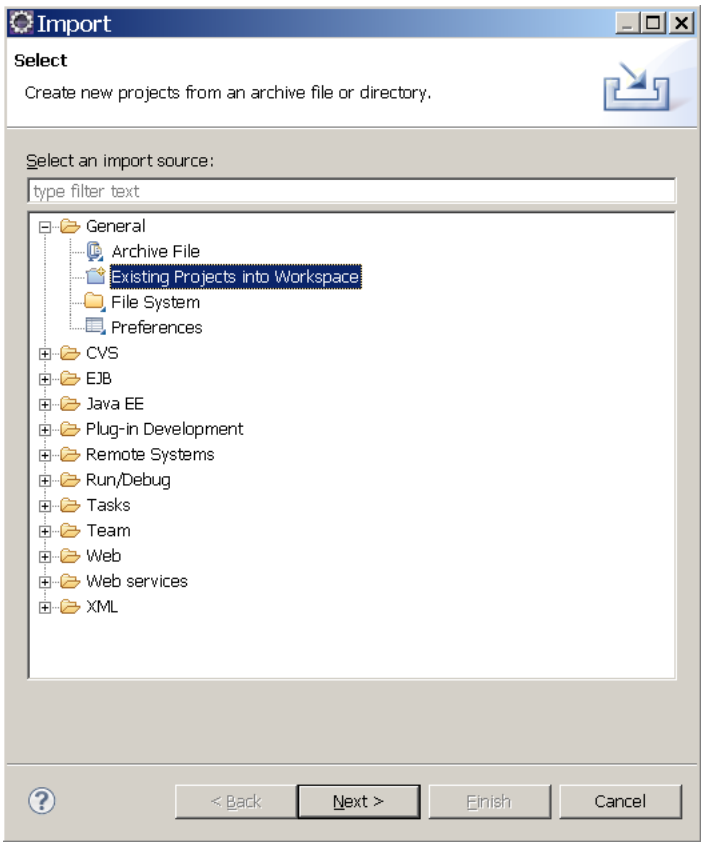
Dodatek A

Navodila za zagon aplikacije NBG iz urejevalnika Eclipse

Sledi opis, kako pognati aplikacijo NBG iz urejevalnika Eclipse.

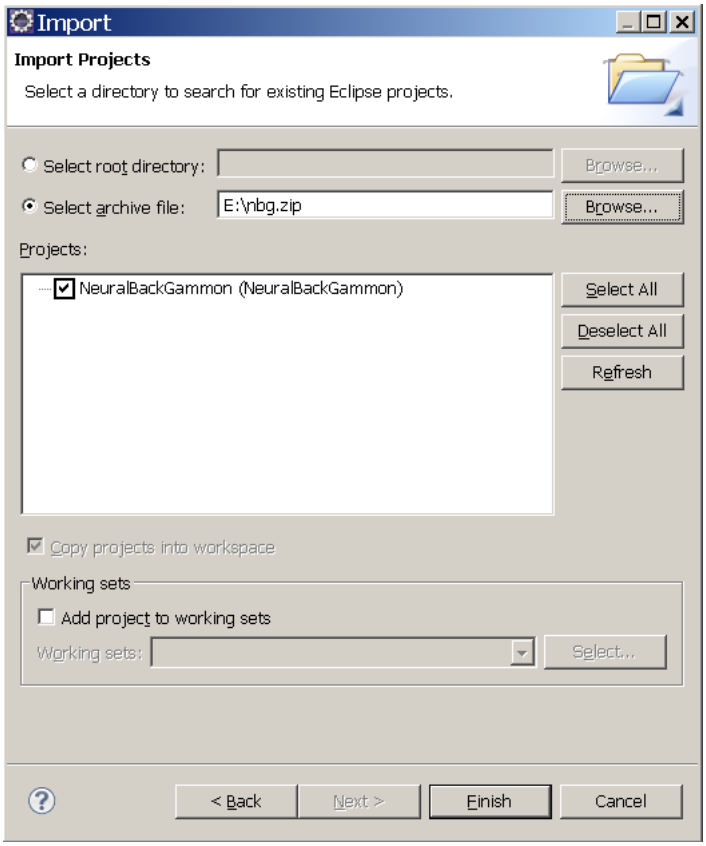
1. Namestitev programskega jezika Java
   - Predpogoj za zagon aplikacije je, da imamo nameščeno na računalniku programsko orodje Java.
   - Orodje in navodila najdemo na spletni strani: http://java.sun.com/
2. Inštalacija urejevalnika Eclipse
   - Če še nimamo nameščenega urejevalnika Eclipse, ga je potrebno namestiti.
   - Dobimo ga na naslovu: http://www.eclipse.org/downloads/
   - Imamo več možnosti, izberemo *Eclipse IDE for Java EE Developers*
   - Ko je paket prenesen, ga razpakiramo, podrobnejša navodila za inštalacijo najdemo v imeniku *readme*.
   - Za naše potrebe zdostuje, da samo poženemo datoteko *eclipse.exe*.
   - Ob prvem zagonu urejevalnika je potrebno nastaviti delovno polje (ang. „workspace“) - imenik, kjer bodo shranjeni vsi eclipse projekti.
   - Če se ob prvem zagonu pojavi zavihek *Welcome screen*, ta zavihek zapremo, ker ga ne potrebujemo.
3. Odpiranje aplikacije NBG z urejevalnikom Eclipse

- Sedaj je potrebno aplikacijo NBG uvoziti v urejevalnik Eclipse.
- V meniju izberemo *File* in nato *Import*.
- Prikaže se nam okno *Import* (slika A.1)



Slika A.1: Okno „Import“

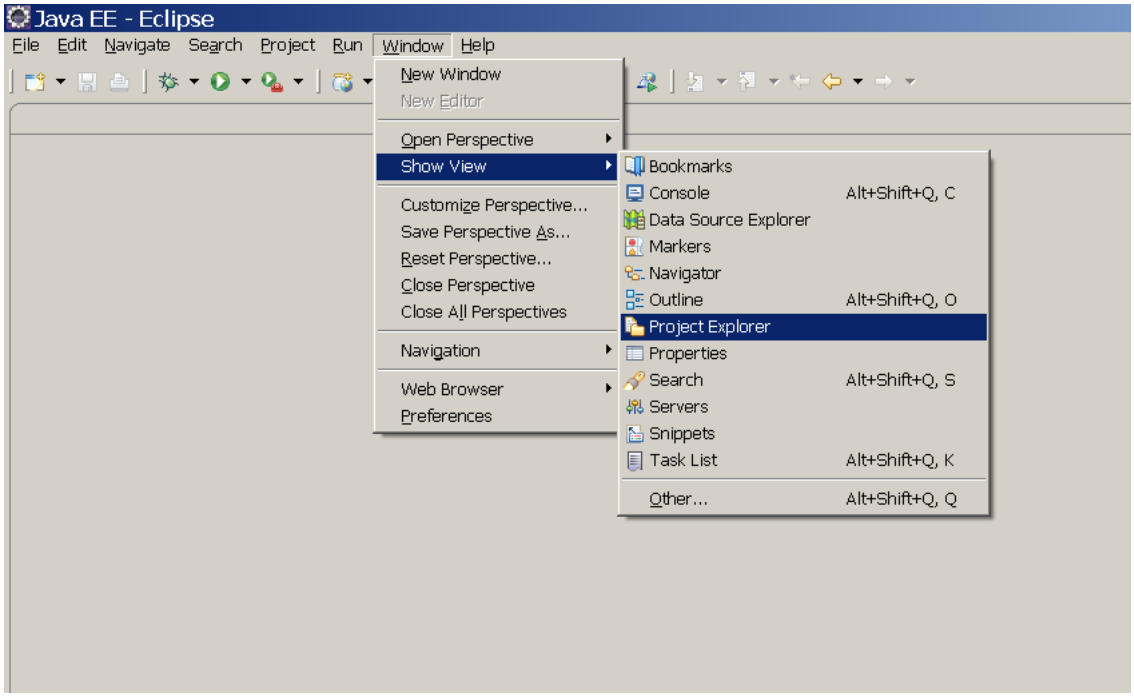
- Pod razdelkom *General* izberemo *Existing Projects Into Workspace* in kliknemo na gumb *Next*.
- Pojavi se novo okno *Import* (slika A.2)
- Izberemo možnost *Select archive file* in nato poiščemo pod *Browse* datoteko *ngb.zip*. Datoteka *ngb.zip* predstavlja zapakirano izvorno kodo, namenjeno prav za urejevalnik Eclipse, zato jo moramo uvoziti zapakirano (s končnico *.zip*).
- Če smo pravilno uvozili projekt, se ime le-tega (*NeuralBackGammon*) izpiše pod razdelkom *Projects*. Projekt mora biti obkljukan, kot je to prikazano na sliki A.2. Na koncu pritisnemo gumb *Finish*.



Slika A.2: Okno „Import“

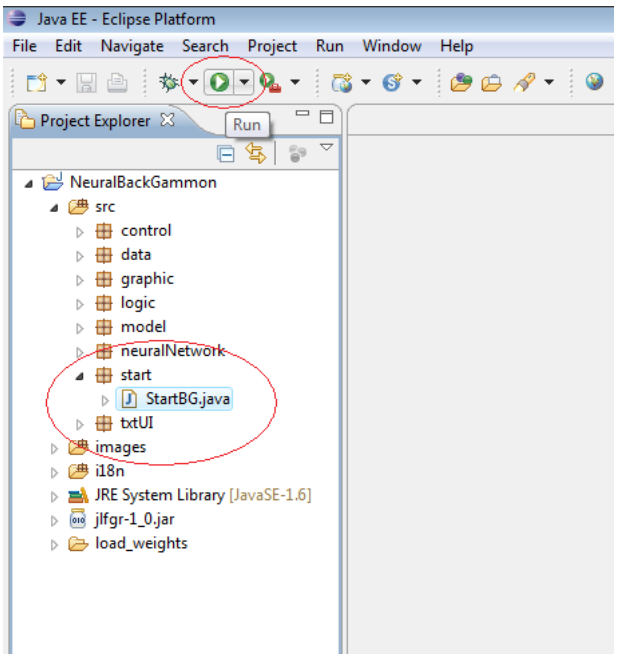
4. Zagon aplikacije NBG iz urejevalnika Eclipse

   - Sedaj je projekt *NeuralBackGammon* že uvožen v urejevalnik Eclipse.
   - Če uporabnik (na levi strani) ne vidi okna *Project Explorer*, naj v meniju izbere možnost *Window*, *Show View* in nato *Project Explorer*, kot to prikazuje slika A.3. *Project Explorer* nam pokaže celo strukturo projekta. Če želimo pogledati izvorno kodo, dvokliknemo na izbrano datoteko v projektu.

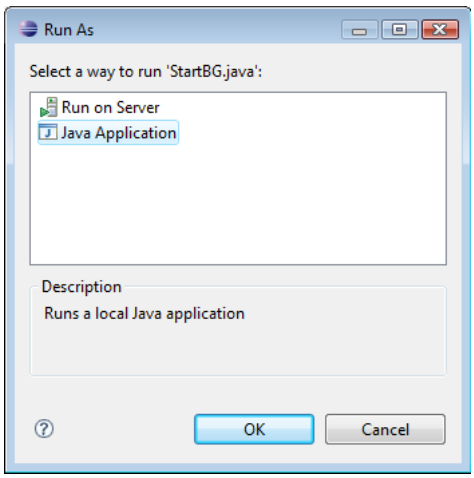


Slika A.3: Izbira „Project View“

   - Sedaj je v projektu *NeuralBackGammon* potrebno označiti datoteko, kjer se nahaja *main* metoda. *Main* metoda se nahaja v datoteki *StartBG.java*. To storimo v okno *Project Explorer*, kot to prikazuje slika A.4 in kliknemo na gumb *Run*.
   - Ob prvem zagonu aplikacije nas urejevalnik Eclipse vpraša po načinu zagona aplikacije. Izberemo lokalni način: *Java Application*, kot to prikazuje slika A.5.
   - Uporaba aplikacije je opisana v 5. poglavju Grafični vmesnik za igranje backgammona - NBG.



Slika A.4: Zagon projekta



Slika A.5: Izbira zagona

Slike

Tabele

Literatura

[1] A. Dobnikar, Adaptivni sistemi,
skripta, januar 2006

[2] C. E. Shannon, Programming a Computer for Playing Chess,
v reviji Philosopical Magazine, marec 1950

Dostopno na spletni strani: http://www.pi.infn.it/ carosi/chess/shannon.txt

[3] G. Tesauro, FTPable benchmark evaluation function,
objavljeno na spletu Backgammon Galore, februar 1993

Dostopno na spletni strani: http://www.bkgm.com/rgb/rgb.cgi?view+610

[4] Gerald Tesauro, Practical Issues in Temporal Difference Learning,
v reviji Machine Learning, maj 1992

Dostopno na spletni strani:
http://www.springerlink.com/content/n164q6921876880j/fulltext.pdf

[5] G. Tesauro, Temporal Difference Learning and TD-Gammon,
v reviji Communications of the ACM, marec 1995

Dostopno na spletni strani: http://www.research.ibm.com/massive/tdl.html

[6] (2007) Online Backgammon Inc., Backgammon History

Dostopno na spletni strani: http://www.gammoned.com/history.html

[7] (2009) Backgammon Galore, Backgammon Rules

Dostopno na spletni strani: http://www.bkgm.com/rules.html