# Uvod

V tekmovanju GVGAI je cilj implementacija agenta, ki bo uspešno igral naključno izbrano igro izmed množice vnaprej predpisanih. V okviru tega projekta sem se lotil prepoznavanja igre, v katero je agent postavljen. Če je taka prepoznava uspešna, jo lahko izkoristimo tako, da potem agent upošteva strategijo, ki smo jo pripravili vnaprej za prav tisto določeno igro, npr. s spodbujevanim učenjem ali pa ročno.

V tem projektu sem si izbral 10 iger izmed 60 in se osredotočil na prepoznavo le-teh, in sicer gre za igre iz 2. učne množice (<http://www.gvgai.net/training_set.php?rg=2>).

# Metode strojnega učenja

## Nevronske mreže

Različne vrste nevronskih mrež so med najbolj rabljenimi metodami strojnega učenja v prepoznavanju vzorcev. Gre za približek delovanja človeških možganov – celice (nevroni) si med seboj izmenjujejo signale preko povezav (sinaps). Najpogosteje se nevronske mreže prikaže z uteženim grafom povezav, kjer vozlišča predstavljajo nevrone, povezave predstavljajo sinapse, uteži vozlišč predstavljajo signale, uteži povezav pa povejo, koliko signal enega vozlišča vpliva na drugega. Cilj nevronske mreže je najti take uteži, da se čim bolje prilegajo učnim primerom.

Najosnovnejša oblika nevronske mreže je dvodelni graf, sestavljen iz vhodne in izhodne plasti. Signal tako potuje samo v eno smer – od vhodne plasti do izhodne.

V tem projektu sem uporabil t.i. Multilayer Perceptron, ki osnovni nevronski doda še poljubno število (v mojem primeru samo 1) skritih plasti. To nam omogoča, da iz osnovnih lastnosti v vhodni plasti razberemo še t.i. vmesne koncepte, torej različne interakcije med temi lastnostmi. To pogosto omogoča veliko boljše delovanje mreže.

Implementacija, ki sem jo uporabil, je iz odprtokodne knjižnice Neuroph. Poleg tega sem uporabil še knjižnico nnet v programskem jeziku R kot preizkus.

## Naključni gozdovi

Odločitvena drevesa so ena najpreprostejših metod v strojnem učenju. Na vsakem nivoju drevesa množico učnih primerkov ločimo po neki lastnosti in to počnemo dokler ne pridemo do množic primerkov, ki vsi spadajo v isti razred, ali pa nam zmanjka lastnosti. Ko potem prepoznavamo razred novih primerkov, preprosto sledimo drevesu po lastnostih do lista, nato pa napovemo razred, ki se v listu največkrat pojavi.

Algoritem naključnega gozda zgradi veliko število odločitvenih dreves (v mojem primeru 100), vsakega na podlagi neke naključne podmnožice lastnosti in izbora učnih primerov. Za končno prepoznavo nato vsako od dreves odda svojo napoved in napovemo razred, ki se med napovedmi pojavi največkrat.

Naključni gozd sem implementiral sam. Za učno množico sem poskusil tako neuravnoteženo kot tudi uravnoteženo (pri nevronski mreži uravnoteženost ni imela vidnega pozitivnega vpliva).

## Atributi

Metode strojnega učenja za prepoznavanje potrebujejo kompakten opis lastnosti primerkov. V ta namen uporabljamo t.i. atribute (značilke).

Za igre sem definiral preproste atribute: katere akcije so agentu na voljo, kakšno začetno hitrost ima, koliko življenskih točk (health points) ima, kako je na začetku obrnjen. Vsi ti atributi so deterministično določeni glede na igro in jih agent zelo preprosto lahko ugotovi v začetnem stanju.

Poleg tega sem uporabil dodatne, bolj kompleksne atribute iz dela Nejca Novaka (Novak, 2016). Ker nisem imel dostopa do njegovega klasifikatorja, s katerim je te atribute prepoznaval, sem jih v svojih podatkih simuliral ročno. Pri tem sem upošteval napake, ki jih opisuje, in sicer sem pravilne prepoznave pokvaril z verjetnostma

in .

# Meritve

Ker se osredotočamo na prepoznavi izbranih 10 iger, algoritme testiramo na bolj uravnoteženi množici podatkov, kjer se vsaka od izbranih 10 iger pojavi 10-krat, preostale pa le enkrat. To moramo potem upoštevati v sami analizi, saj v samem tekmovanju temu ne bo tako.

## Nevronska mreža

Če predpostavimo, da lahko zgoraj opisane Novakove atribute napovemo s 100% točnostjo, nevronska mreža deluje že izredno dobro – doseže kar 95% točnost napovedi. Podobno točnost doseže tudi, če uporabimo neuravnotežno testno množico, kar pomeni, da bi zelo dobro delovala tudi v samem tekmovanju. Težave ji delata le 2 izmed 10 iger (firestorms in whackamole), preostale prepozna zanesljivo.

Jasno pa je taka predpostavka premočna in moramo upoštevati napake v napovedni atributov, kot opisano v prejšnjem poglavju. To jasno precej vpliva na točnost napovedi, saj imamo sedaj nedeterministične atribute. V tem primeru mreža doseže 45% točnost napovedi. Vseeno pa 3 izmed 10 iger (seaquest, eggomania in nekoliko slabše pacman) prepozna zanesljivo, poleg tega pa zelo dobro prepozna, kadar igra ne spada med izbranih 10.

## Naključni gozd

Ob predpostavki 100% točnosti napovedi atributov naključni gozd deluje nekoliko slabše od nevronske mreže – doseže 85% točnost. Nekoliko slabše namreč prepozna, kdaj igra ne spada med izbranih 10, sicer pa 9 od 10 iger prepozna pravilno. Le overload napačno prepozna kot digdug.

Če uporabimo neuravnoteženo učno množico, naključni gozd igro seaquest zanesljivo razpozna, vse preostale primerke pa označi, da ne spadajo med izbranih 10 in tako doseže 40% točnost.

Če uporabimo uravnoteženo učno množico, naključni gozd 6 od 10 iger prepozna vedno. Pri 2 (camelRace, whackamole) ima 70% točnost, preostalih 2 (overload, digdug) pa ne prepozna. Slabo pa prepoznava, kdaj igra ne spada med izbranih 10 in tako doseže skupno točnost 55%.

# Zaključek

Dejstvo, da nevronska mreža skoraj vedno pravilno prepozna igre ob predpostavki 100% točnosti pri napovedi atributov, nam pove, da je izbira atributov (predvsem Novakovih) dobra, saj nam očitno povejo dovolj o igri, ki jo agent igra. Žal pa jih ne znamo napovedovati popolno.

Nevronska mreža se je na izbranih 10 igrah obnesla precej slabše kot naključni gozd, je pa boljše razpoznala, kdaj igra ni spadala med njih. Kot predlog za nadaljnje delo bi torej lahko nevronsko mrežo uporabili za prepoznavanje ali igra spada med teh 10 iger ali ne, nato pa v primeru, da je ena od 10, uporabili naključni gozd za prepoznavo same igre.

Poleg tega bi potrebovali podobne klasifikatorje še za preostale igre, ne le za teh 10 (trenutno jih je na voljo 80).

Nazadnje pa bi seveda lahko precej izboljšali točnost, če bi nadaljevali Novakovo delo in izboljšali napovedi njegovih atributov.

# Reference

Novak, N. (2016). *General Video Game AI Competition.*