

Halite II un *bot* cu inteligență artificială

Andriu Ionuț-Ciprian

Rezumat

Arhitecturile de învățare profundă, cum ar fi rețelele neuronale profunde, rețelele de convingeri profunde și rețelele neuronale recurente, au fost aplicate în domenii precum recunoaștere vizuală, recunoașterea vocală, prelucrarea limbajului natural, recunoașterea audio, filtrarea rețelei sociale, traducerea automată, bioinformatică, unde au obținut rezultate comparabile și în unele cazuri superioare experților umani.

Modelele de învățare profundă sunt inspirate vag de modelele de procesare a informațiilor și de comunicare în sistemele nervoase biologice, dar au diferențe față de proprietățile structurale și funcționale ale creierului biologic, ceea ce le face incompatibile cu dovezile neuroștiinței.

Lucrarea de față își propune simularea inteligenței artificiale prin implementarea unui bot pentru competiția Halite II folosind principiile învățării profunde. Halite este o provocare de programare de inteligență artificială cu sursa deschisă(opensource), creată de firma Two Sigma, unde jucătorii construiesc boți folosind limbajul de programare ales de ei pentru a se lupta pe o tablă virtuală bidimensională. Jocul din acest an este o poveste spațială. Misiunea este să extragem sare gemă(halite). Boții controlează nave care ancorează pe planete și le minează, producând mai multe nave pentru a învinge adversarii. Pentru a obține victoria, va fi nevoie de un control optim al traseului, de deplasare între planete și a distruge navele inamice.

Pentru realizarea *botului* folosind principiile învățării profunde am creat în prealabil un *bot* cu principii simpliste și cu 3 opțiuni de mișcare și anume atac, minarea unei planete cucerită și minarea unei planete libere cu o șansă de 30% de aș schimba planul, apoi am duplicat acest bot și cu ajutorul unui script realizat în python am făcut ca cei doi boți să se lupte între ei de minim 3000 de ori și din fiecare meci extragând mișcările și diferite date ale botului câștigător, date ce le-am folosit mai departe pentru a realiza modelul de antrenare. Pentru realizarea modelului de antrenare am folosit o rețea neuronală secvențială cu două straturi ascunse de câte 256 de intrări, funcția de activare “relu” și funcția de optimizare pentru model fiind optimizatorul “adam”. Apoi cu modelul obținut, care odată încărcat și utilizat de același bot inițial, modificat doar pentru a prelua mișcările din model, am reușit să ating niște cifre promițătoare, botul final învingând împotriva celui inițial în peste 71% din cazuri.