

Шах управљан говором (српски)

- Ђорђе Његић SW-12-2018
- Његош Благојевић SW-18-2018
- Лука Курељушић SW-23-2018

Увод

Шах представља стратешку игру на табли за два играча. Циљ игре је заробити противничког краља на такав начин да се не може извући.

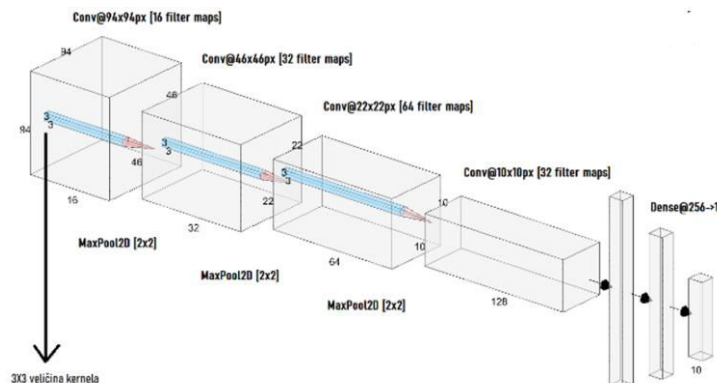
Већ деценијама шах представља плодно тло за тестирање и развој различитих алгоритама вештачке интелигенције.

Опис проблема

Циљ овог пројекта јесте омогућавање играња шаха говорним командама. У сврху тога, развијена су два различита алгорита за играње шаха. Први представља класични минимакс са алфа бета одсецањем. За други је иницијална замисао била да се имплементира *Monte Carlo tree search* уз употребу неуронских мрежа али се тај алгоритам показао као превише хардверски захтеван за комплексне игре као што је шах. Из тог разлога, неуронска мрежа је искориштена у комбинацији са минимаксом.

Препознавање говора

Основни проблем пројекта представља препознавање говора (цифара и слова) помоћу којих ће се вршити селекција поља на шаховској табли. Проблем је решен коришћењем две неуронске мреже. Подаци за неуронску мрежу састоје се од ручно прикупљених аудио снимака. Подаци су балансирани, те свака класа за слова садржи по 105, а класа за бројеве по 151 оригиналан снимак. Сви подаци обеју класа су додатно аугментовани на различите начине (повећавањем и смањивањем брзине и интензитета, додавањем шума и сл) након чега су бројале 1155 и 1661 снимака респективно. Аудио фајлови су затим трансформисани у мел спектограме и као такви се прослеђивали конволуционој неуронској мрежи.



Једна конволутивна мрежа коришћена је за класификацију слова а друга за класификацију цифара (иста архитектура). Као оптимизатор кориштен је *adam* са *default learning rate*-ом и након 3 епохе остварена је тачност на валидационом скупу од око 97%. Сама архитектура мреже је приказана на слици изнад.

Алгоритми за шах

Што се тиче алгоритама за шах, пошто је минимакс опште познат алгоритам који се веома дуго користи у игри шах, неће се улазити у његово објашњавање.

Приликом имплементације другог алгоритма, коришћена је неуронска мрежа која се обучавала на подацима из 20 хиљада партија, са укупним бројем стања партије од милион и по.

Архитектура се састојала од 4 *dense* слоја са по 512, 256 и 50 неурона, док је последњи имао 3 неурона што је уједно и био излаз из мреже (3 класе, победа, пораз и нерешено). Активациона функција је *relu* а између сваког слоја је додат *dropout* слој са коефицијентом 0.2 да би се спречио оверфитинг.

Други алгоритам представља покушај оптимизације минимакс алгоритма у коме ће се иницијални потези бирати на основу резултата добијених из неуронске мреже.

Резултати

Резултати детекције звука су добијени коришћењем *accuracy* методе. Над тестним подацима, неуронска мрежа за детекцију цифара је остварила резултат од 97% док је *snr* за слова остварила резултат од 95%.

Што се тиче неуронске мреже за процену табле, након 20 епоха постигнуту су резултати од око 68%.

Алгоритам који користи искључиво минимакс тестиран је у игри са ботом са 1200 ELO ранком и остварио је победу (на тешком левелу). Други алгоритам је показао слабије резултате јер због временске ограничености ради на мањој дубини.

Закључак

Иако су неуронске мреже дале релативно добре резултате, и даље постоји велики простор за оптимизацију. Због ограничености хардвера, приликом имплементације *MCTS* алгоритма били смо принуђени на одређене уступке:

- Сет података није могао бити већи од 20000 партија
- Због комплексности шаха, стабло се није могло развијати до краја што је суштина *MCTS* алгоритма већ се заустављало на одређеној дубини

Због поменутих уступака алгоритам није имао очекивану перформансу.

Референце

1. https://en.wikipedia.org/wiki/Monte_Carlo_algorithm
2. https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning

