

1

bezpieczne pola

a) jak je wykorzystać do heury?

większa waga na bezpiecznym polu

b) zdefiniuj precyzyjnie warunek bycia bezpiecznym polem

na każdym kierunku (pion, poziom, dwa skosy) sąsiaduje z co najmniej jednym swoim bezpiecznym polem (poza mapą same bezpieczne dla obu)

c) jak szybko je liczyć?

bezpieczne pola pączkują od narożników, zajmując jedno bezpieczne pole, nowe mogą pojawić się obok
można dla graczy pamiętać już zajęte bezpieczne pola i potencjalnie nowe dla każdego z nich

2

W jaki sposób można wykorzystać czas poświęcony na obliczenia najlepszego poprzedniego ruchu do obliczenia najlepszego bieżącego ruchu (zakładamy, że rozgrywamy tylko jedną partię).

- **Alpha-Beta-Search:**
 - leniwe drzewo gry, nie liczymy następników wiele razy
- **MCTS:**
 - leniwe drzewo gry, nie liczymy następników wiele razy
 - po wykonaniu ruchu zostawiamy obliczenia w poddrzewie do którego przechodzimy

W jaki sposób wykorzystać możliwość równoległego wykonywania kodu w celu poprawy jakości gry?

- **Alpha-Beta-Search:**
 - generowanie następników
 - liczenie heurystyki (gdy monoid)
- **MCTS:**
 - generowanie następników
 - przeprowadzanie losowych rozgrywek równoległe

3

ox z buczeniem, zaproponuj agenta

- wagi: środek > rogi > reszta
- strzelasz losowo (przez wagi bardziej prawdopodobne), tam gdzie samemu dążysz do wyniku i możesz zablokować przeciwnika

6

Dla jakich zbiorów danych i jakich cech wystarcza 1 neuron?

- dwie grupki, x_1 i x_2 , prosta
- kula, kwadraty, koło
- szachownica, $x_1 \times x_2$, układ współrzędnych, oba pozytywne lub oba negatywne

Co dzieje się, gdy dla bardziej złożonych sieci damy zbyt duży Learning rate?

miota nim jak szatan

W którym zadaniu przydają się \sin i \cos ?

spirala

Dla każdego zbioru danych (oprócz spirali) powiedz jaka najprostsza sieć neuronowa (tylko x_1 , x_2) poprawnie klasyfikuje

- dwie grupki, 1 neuron, prosta
- kula, 3 neurony, prawie trójkąt by zaznaczyć kulę
- szachownica, 4 neurony, ale długo się uczy (4+4 się nawet szybko uczy jakies)

7

$x \vee y$

$x + y$

$x \wedge y$

$x + y - 1$

$\neg x$

$\neg x + 1$

$x \text{ xor } y$

$(x \wedge \neg y) \vee (\neg x \wedge y)$

xor z jednym neuronem niemożliwy

Wyobraźmy sobie przestrzeń 3D. Zaznaczmy punkty (0, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 0). To są wartości tego xora (x , y , $x \text{ xor } y$). Funkcja jednego neuronu wygląda tak: $ax + by + c$. Można zwizualizować to jak płaszczyznę. Nie można poprowadzić płaszczyzny przez te 4 punkty.

8

każdą funkcję boolowską można, bo można sprowadzić do postaci CNF/DNF a tamte operacje potrafimy reprezentować

możliwie mało warstw: CNF/DNF

możliwie mało neuronów: CNF lub DNF wybierz ten krótszy wariant; jak nie to można się starać grupować w duże koniunkcje/dysjunkcje bo wtedy jeden neuron załatwi

9

if $x > T$ then a else c

$\max(a, b)$

$a + b$