

NeuGenGo

Kann unser neuronales Netz besser Go spielen als wir?

Lennart Braun, Armin Schaare, Theresa Eimer

Universität Hamburg
Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Informatik, Arbeitsbereich WR
Praktikum Parallele Programmierung SS 15

9. September 2015

- 1 Problemstellung
- 2 Lösungsansatz
- 3 Parallelisierungsschema
- 4 Laufzeitmessungen
- 5 Leistungsanalyse
- 6 Skalierbarkeit

Problemstellung

- Unser Ziel ist es, neuronale Netzwerke zu trainieren, so dass diese uns im Go schlagen können.
- Zwischenziel / Alternative: Können wir neuronale Netze so trainieren, so dass sie besser als zufällig erzeugte Netze spielen?

TODO: Ziele besser verkaufen

Go

- Asiatisches Brettspiel
- Wird auf Brettern mit 19×19 Knoten gespielt.
- Ziel: Gebiet einkreisen und gegnerische Steine schlagen
- Spielende: wenn beide Spieler passen

TODO: Graphik (möglichst unter CC / selbst erstellt)

Abbildung: Stellung eines Go Spiels

Neuronale Netzwerke

- TODO: kurze Beschreibung von neuronalen Netzwerken

TODO: Graphik (möglichst unter CC / selbst erstellt)

[Abbildung](#): Schema eines neuronalen Netzwerks

Lösungsansatz

- Beschränkung auf 9×9 Bretter
- Feedforward Netze (TODO: Layout)
- Genetische Algorithmen (TODO: Parameter)

Lösungsansatz

Algorithmus 1 sequentielle Lösung

```
1:  $N_0 \leftarrow \{n \text{ zufällig generierte neuronale Netzwerke} \}$ 
2: for  $net \in N_0$  do
3:   trainiere  $net$  auf regelgerechtes Spielen
4: end for
5: for Runde  $i = 0$  bis ... do
6:   for  $\forall net_a \neq net_b \in N_i$  do
7:     lass  $net_a, net_b$  gegeneinander spielen
8:     zähle die Anzahl der Siege
9:   end for
10:  generiere  $N_{i+1}$  mittels genetischen Algorithmus abhängig von  $N_i$  und den Spielergebnissen
11: end for
12: Speichere  $N_n$ 
```

UML

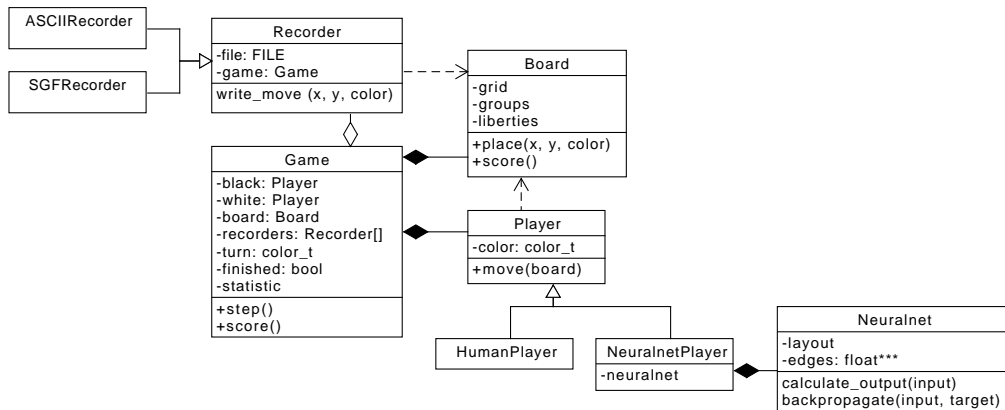


Abbildung: Klassendiagramm

