

Praktikum Ingenieursmäßige Softwareentwicklung

Implementierung von neuroevolutionären KI-Verfahren für das Erlernen von Atari-Spielen – Nico Peter

SOFTWARE-ENTWURF UND -QUALITÄT,
INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION, KIT-FAKULTÄT FÜR INFORMATIK



Inhalt

- Motivation
- Grundlagen
 - Evolutionäre Algorithmen
 - Neuroevolution of Augmenting Topologies (NEAT)
 - OpenAI Gym
- Implementierungen von NEAT
- Ergebnisse
- Zusammenfassung

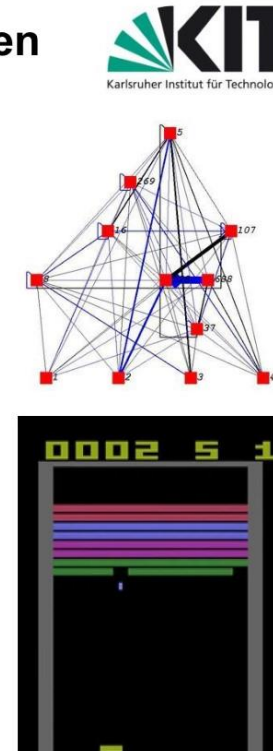
Motivation – Aufgabenstellung

1) Implementierung von neuroevolutionären KI-Verfahren für das Erlernen von Atari-Spielen (1-2 Studierende)

- Neuroevolutionäre Verfahren wie HyperNEAT können (im Gegensatz zu vielen anderen Deep Learning Algorithmen) neben den Parametergewichten auch die Netztopologie evolvieren
- Atari-Spiele haben sich für die Evaluation der Leistungsfähigkeit von ML-Verfahren etabliert

Aufgabe: Entwickeln Sie in Python eine KI für das Erlernen von Atari-Spielen unter Nutzung von HyperNEAT

- Verwenden Sie das OpenAI Gym Framework, welches eine Vielzahl von Atari-Spielen enthält
- Nutzung eines bestehenden HyperNEAT Frameworks



11 17.10.2019

Prof. Ralf H. Reussner: Vorbesprechung Praktikum „Ingenieurmäßige Software-Entwicklung“ Wintersemester 2019/20

Software-Entwurf und -Qualität
Institut für Programmstrukturen und Datenorganisation

Motivation – Schritte und Schwierigkeiten

■ Schritte:

- Einarbeitung in NEAT, OpenAI Gym und Tools
- Installation der Entwicklungsumgebung für Tools
- Trainieren der Netze via Robotik-Umgebungen
- Hyperparameter konfigurieren

■ Schwierigkeiten:

- Relativ wenig Codebeispiele

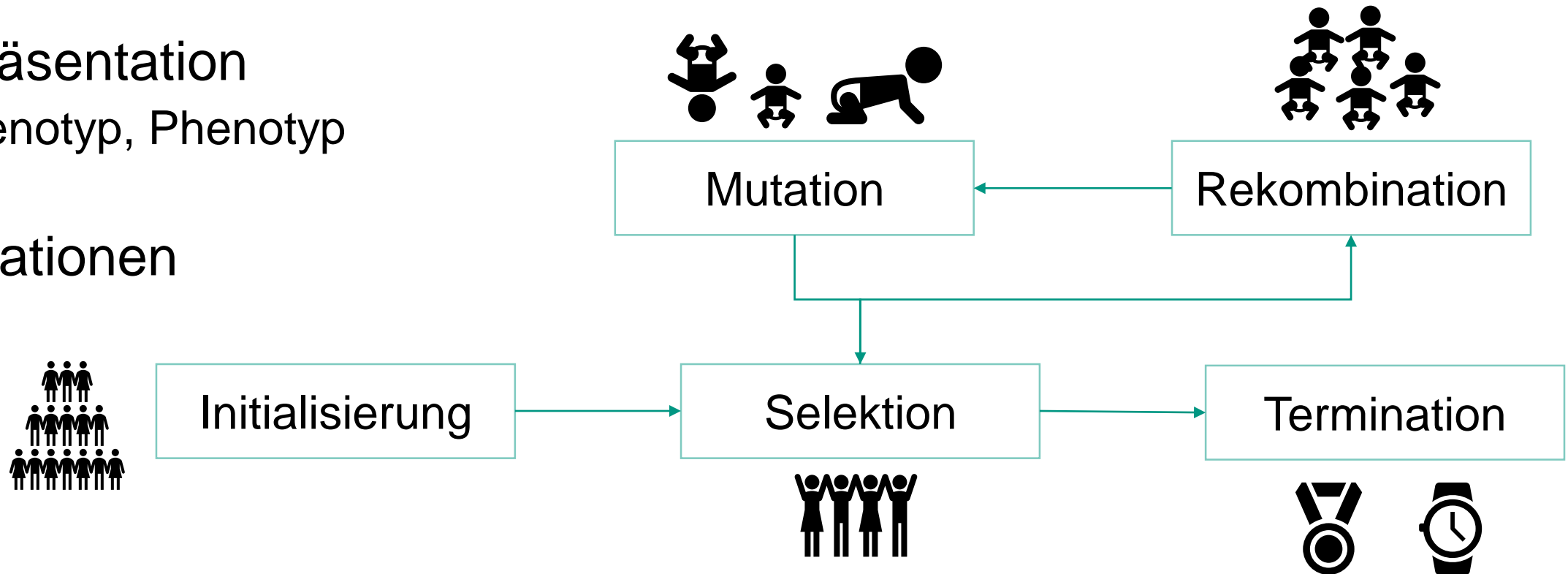
Grundlagen – Evolutionäre Algorithmen

■ Orientierung an Natur

■ Repräsentation

■ Genotyp, Phenotyp

■ Operationen



Grundlagen – NEAT

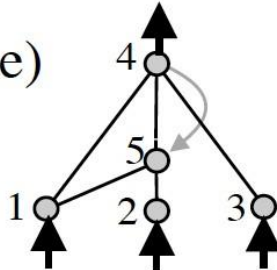
- Initialisierung: Feed Forward Netze
- Selektion: Die besten n Netze gemäß Fitness-Werten
- Rekombination: Homologie durch Innovations-Marker
- Mutation: Neuron/Verbindung wird hinzugefügt/gelöscht
- Artbildung: Splittung in Gruppen ähnlicher Topologie/Verbindungen

Grundlagen – NEAT

■ Repräsentation:

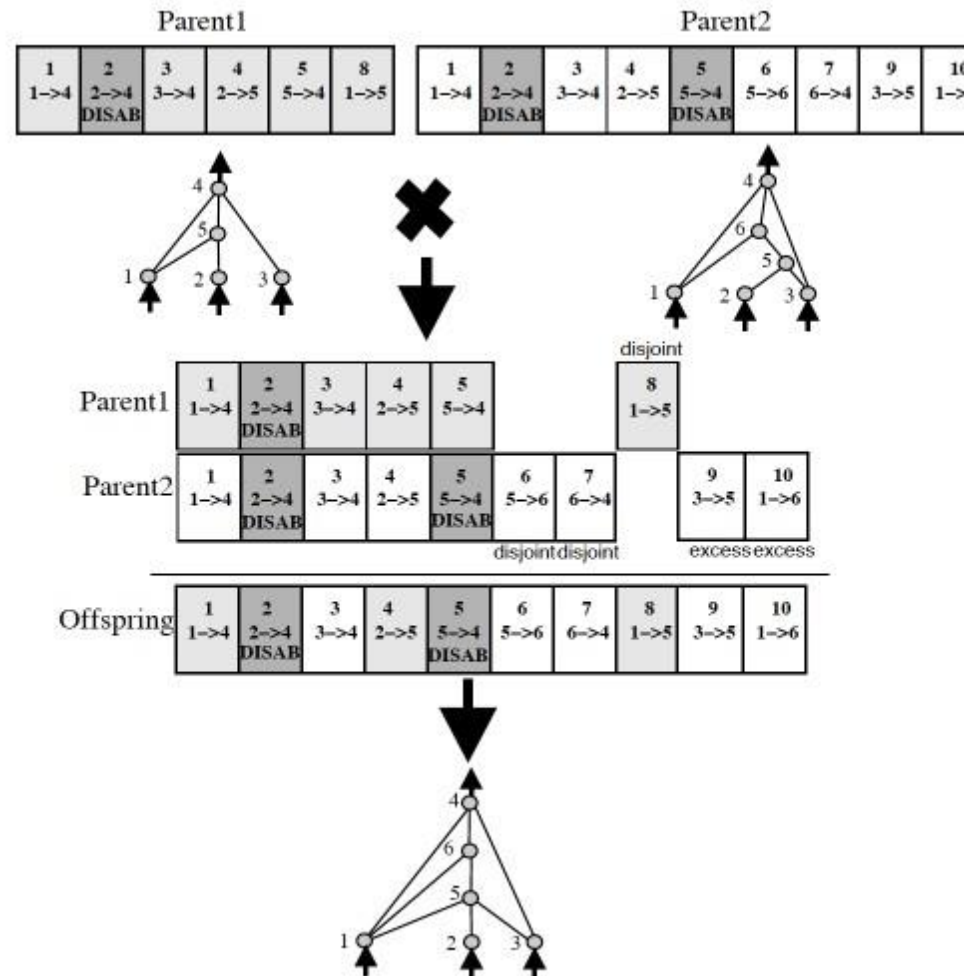
Genome (Genotype)							
Node	Node 1	Node 2	Node 3	Node 4	Node 5		
Genes	Sensor	Sensor	Sensor	Output	Hidden		
Connect.	In 1	In 2	In 3	In 2	In 5	In 1	In 4
Genes	Out 4	Out 4	Out 4	Out 5	Out 4	Out 5	Out 5
	Weight 0.7	Weight -0.5	Weight 0.5	Weight 0.2	Weight 0.4	Weight 0.6	Weight 0.6
	Enabled	DISABLED	Enabled	Enabled	Enabled	Enabled	Enabled
	Innov 1	Innov 2	Innov 3	Innov 4	Innov 5	Innov 6	Innov 11

Network (Phenotype)



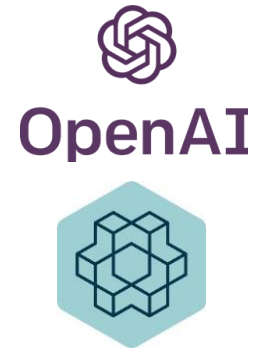
Grundlagen – NEAT

■ Rekombination:



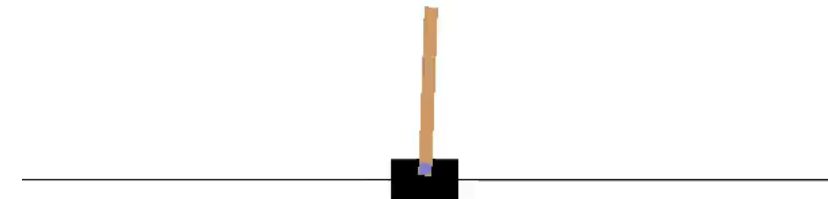
Grundlagen – OpenAI Gym

- OpenAI
 - Forschungslabor, 2015
- OpenAI Gym
 - Python-Bibliothek, 2016
- Gym-Umgebung besitzt
 - Aktionsraum,
 - Beobachtungsraum,
 - Debugging Infos,
 - Boolesche Variable „Fertig“



```
import gym
env = gym.make("CartPole-v1")
observation = env.reset()
for _ in range(1000):
    env.render()
    action = env.action_space.sample() # your agent here
    observation, reward, done, info = env.step(action)

    if done:
        observation = env.reset()
env.close()
```



Grundlagen – OpenAI Gym

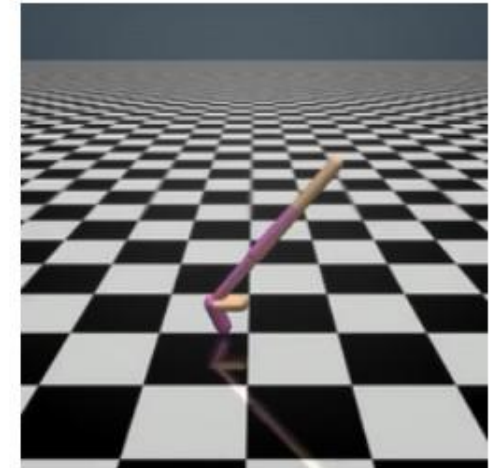
■ Bestenliste

- Wer (Github-Account)
- Punktzahl
- Link zu Repository/Paper
- Video des trainierten Modells

Walker2d-v1 and Walker 2d-v2

Make a two-dimensional bipedal robot walk forward as fast as possible.

- *Walker2d-v1 is an unsolved environment, which means it does not have a specified reward threshold at which it's considered solved.*
- *The robot model is based on work by Erez, Tassa, and Todorov [Erez11].*



User	100-Episode Average Score	Write-up	Video
zlw21gxy	7197.15	writeup	
pat-coady	7167.24	link	video

Implementierungen – Tools

■ NEAT-Python

- 2008, BSD-3-Clause, Installation des Pakets über Pip

■ MultiNEAT

- 2012, LGPL, einfache Installation mit Anaconda

■ PyTorch NEAT

- 2018, Apache License 2.0, benötigt neat-python und torch

Implementierungen – Konfiguration

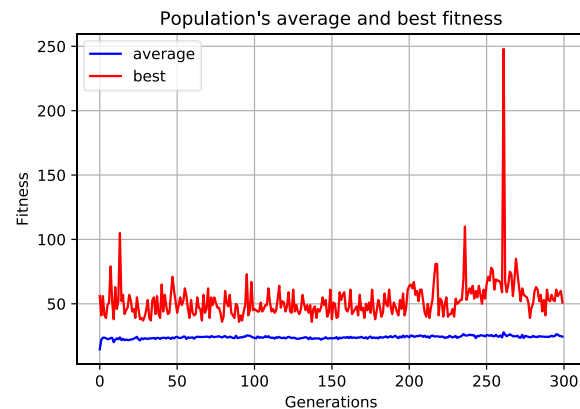
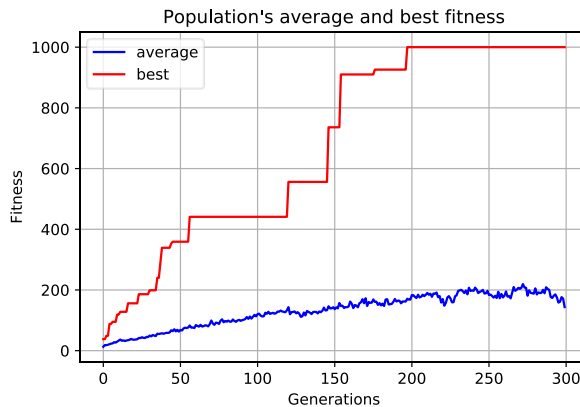
- Konfigurationsmöglichkeiten (Beispiele)
 - Schwellwert für Erfolg
 - Anzahl Genome pro Population
 - Aufhören nach Aussterben aller Spezies?
 - Gibt es Elite?
 - Wahrscheinlichkeit für neue Knoten bei Mutation
 - Aktivierungsfunktion
 - Anteil direkt verbundener Neuronen zu Beginn

Ergebnisse - Ansatz

- Laden der Konfiguration
- Erzeugen der Population (+ Reporter)
- Für jeden Arbeiter:
 - Erstelle Umgebung
 - Erstelle Netz
 - Trainingsschleife bis „done“
 - Gebe erreichte Fitness zurück
- Speichere Gewinner
- Statistik anzeigen und speichern

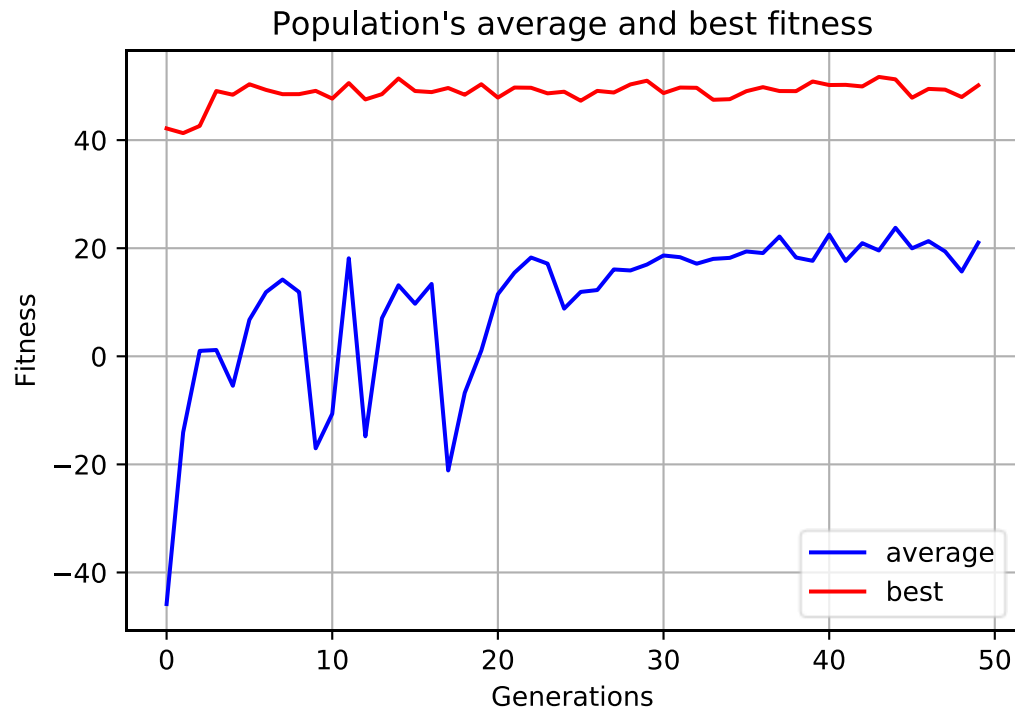
Ergebnisse - Umgebungen

■ InvertedPendulum-v2 (Film)



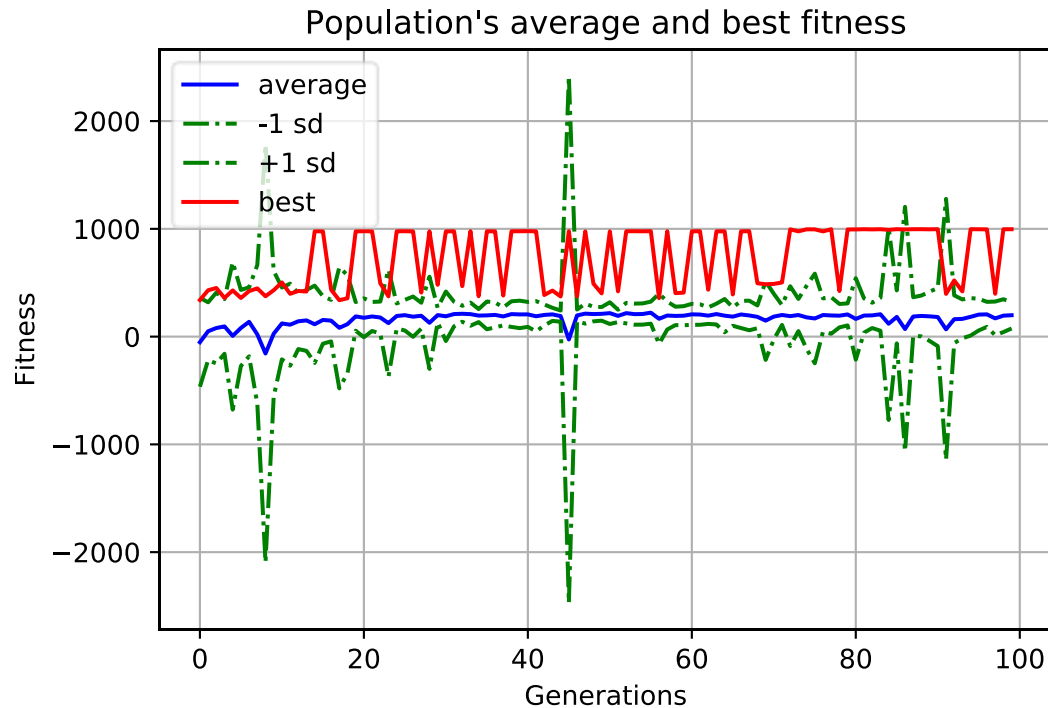
Ergebnisse - Umgebungen

■ Swimmer-v2



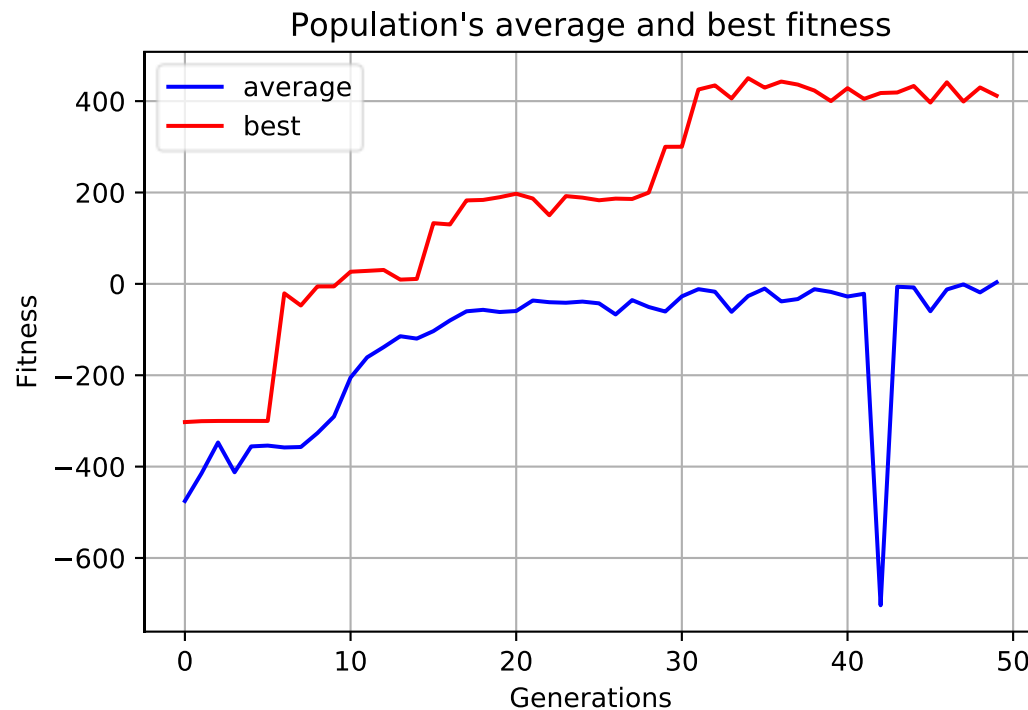
Ergebnisse - Umgebungen

■ Hopper-v2 (Film)



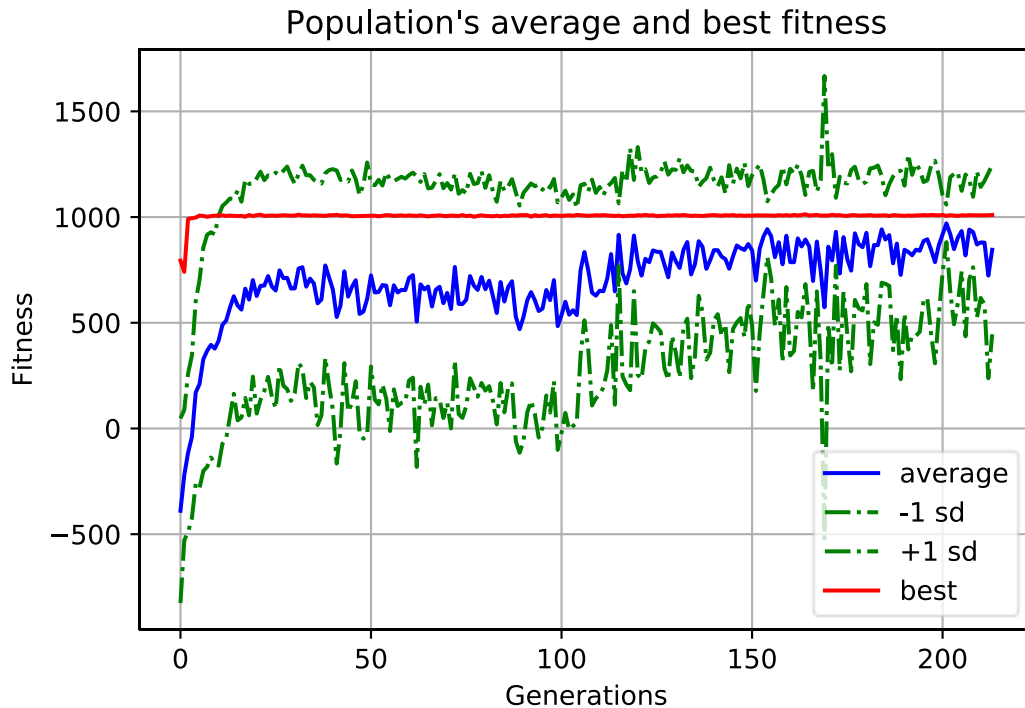
Ergebnisse - Umgebungen

■ HalfCheetah-v2 (Film)



Ergebnisse - Umgebungen

■ Ant-v2 (Film)



Zusammenfassung

- NEAT-Python
- Robotik-Umgebungen aus Gym
 - Liefert Netzeingabe, Vereinfacht Vergleiche
- Ergebnisse
 - NEAT eher ungeeignet für Robotik-Probleme
- Empfehlung / Ausblick
 - Verwendung anderer Algorithmen wie HyperNEAT

Quellen

■ Evolutionäre Algorithmen:

- <https://towardsdatascience.com/introduction-to-evolutionary-algorithms-a8594b484ac>

■ NEAT

- Stanley, Kenneth O., and Risto Miikkulainen. "Evolving neural networks through augmenting topologies." *Evolutionary computation* 10.2 (2002): 99-127.

■ Open AI Gym

- <https://towardsdatascience.com/reinforcement-learning-with-openai-d445c2c687d2>
- <https://openai.com/about/>

Danke für Ihre Aufmerksamkeit!