

Bio-inspirierte Software für bio-inspirierte Hardware

Entwurf und Training von Neuromorphen Neuralen Netzen

1 Neuromorph statt GPU-morph

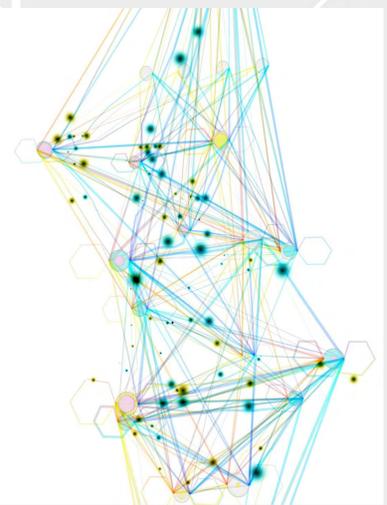
Herkömmliche Architekturen ungeeignet für neuromorphe Systeme

- Generische Netze erfordern die Kompensation **neuromorpher Hardware-Eigenschaften**, anstatt diese zu nutzen
 - Rauschen
 - Small-World-Connectivity
- Mappen auf Hardware
 - Ist komplexes Optimierungsproblem
 - Nutzt Ressourcen nicht vollständig

2 Automatisierte Suche

Schnelle **Skalierung auf verschiedene Anwendungen**

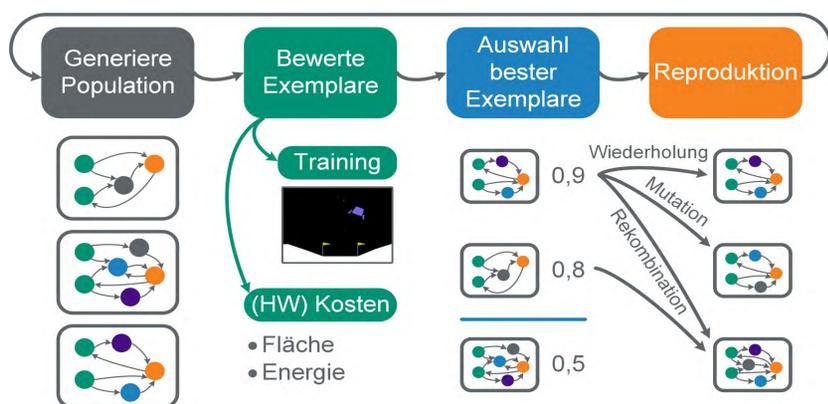
- Effektive Nutzung von Hardware Ressourcen
- Hardware Constraints schränken Suchraum ein
- Nutzt GPUs und im Machine Learning erprobte Methoden zur Vermeidung von "Handcrafting"



3 Evolutionäre Optimierung

"Überleben des am besten **Angepassten**"

- Fitness wird von KPIs und Problemstellung abgeleitet
 - Entfernen der schlechtesten Exemplare
 - Zufällige Veränderungen + Rekombination
- Auch für nicht-differenzierbare Probleme geeignet



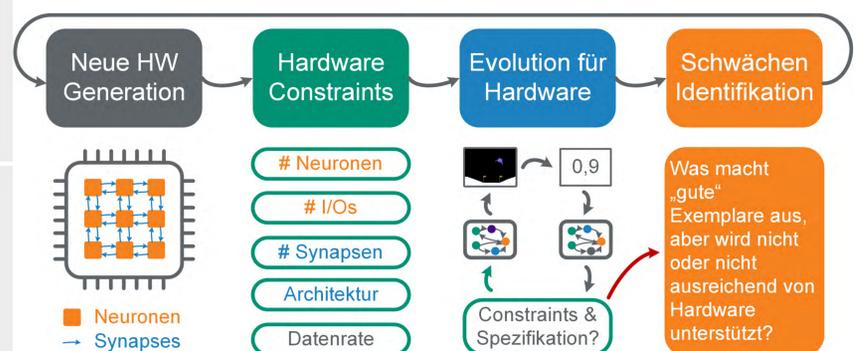
4 HW/SW Co-Design

Für **bestehende Hardware**:

- Architekturen
 - Von einzelnen Neuronen
 - Von "Layern"
- Kollektionen
- Neuronenmodelle
- Lernregeln

und die **nächste HW Generation**:

- Architektur
 - Des Systems
 - Der "Kerne"
- Verbindungsmuster
- Neuronenparameter
- Lernregeln



5 Anwendunggetrieben Forschung vorantreiben

Durch die **evolutionäre** Optimierung können Netze gefunden werden, die in Hardware **hineinwachsen** und deren Beschränkungen berücksichtigen oder sogar zur Verbesserung nutzen, um den besten **Trade-Off** für Robustheit, Genauigkeit, Latenz und Effizienz zu erzielen.

Neben der schnellen automatisierten Architektursuche für bestehende Hardware unter Berücksichtigung von **KPIs** ergeben sich Erkenntnisse für die **nächste Generation** neuromorpher Systeme und Algorithmen.