

Analyse der Optimierungsverfahren mechanischer neuronaler Netzwerke

Projektskizze

Alexander Reimer

Matteo Friedrich

3. Dezember 2023

Inhaltsverzeichnis

1	Einleitung	2
2	Forschfrage	3
3	Zeitplan	3
4	Materialien	4
5	Erwartete Schwierigkeiten	4
6	Bisherige Ergebnisse	4
7	Gruppe	5
8	Externe Kooperationspartner	5
9	Literatur	5

Zusammenfassung

Wir wollen uns mit dem neuen, vergleichsweise wenig erforschten Bereich der Mechanical Neural Networks, kurz MNNs, beschäftigen. Während die bisherige Forschung sich auf die technische, physische Implementation dieser Netzwerke fokussiert hat, wollen wir das Trainingsverfahren optimieren. Dazu wollen wir die bisher verwendeten Algorithmen (evolutionäres Lernen und Pattern Search) selbst implementieren und genauer sowie mit neuen Parametern ausprobieren und vergleichen. Außerdem wollen wir versuchen, eine Methode für Backpropagation bei MNNs zu entwickeln und implementieren, da eine solche unserer Recherche nach noch nicht ausprobiert wurde.

Dafür werden wir uns jedoch auf die Anwendung dieser Algorithmen in einer Simulation beschränken. Die Ergebnisse sollten dennoch einen guten Startpunkt für reale MNNs bieten.

Wir haben bereits eine Simulation sowie einen rudimentären evolutionären Algorithmus umgesetzt.

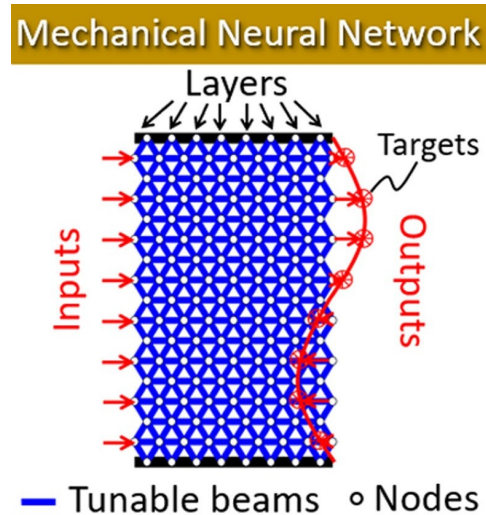


Abbildung 1: Repräsentation eines MNN (von [2])

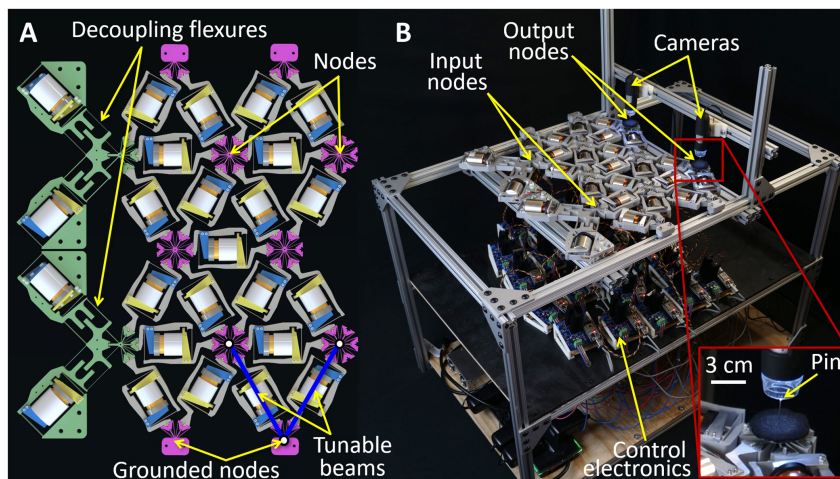


Abbildung 2: Mechanische Umsetzung eines MNN (von [2])

1 Einleitung

Inspiration für dieses Projekt ist ein Artikel von 2022 mit dem Titel „Mechanical neural networks: Architected materials that learn behaviors“ [2].

Es ist der erste veröffentlichte Artikel, der sogenannte *mechanical neural networks*, kurz MNNs, beschreibt, ein neuer Forschungsbereich, in dem neuronale Netzwerke in der physischen Welt umgesetzt werden. Dazu werden Knotenpunkte gebaut, welche mit Federn mit variabler Steifigkeit verbunden werden.

MNNs können aufgrund der variablen Steifheit der Federn als lernfähiges Material verwendet werden, welches seinen Bedingungen oder gewünschten Verwendungszwecken angepasst werden kann.

Dazu kann das Material sowohl in einer Simulation als auch physisch, durch Sensoren in den Federn, trainiert werden.

2 Forscherfrage

Ziel des Projektes ist es, ein neues Optimierungsverfahren für das Trainieren von MNNs zu entwickeln, sowie dieses mit den beiden im Artikel verwendeten Verfahren – ein evolutionärer Algorithmus und Pattern Search – zu vergleichen.

Ziel des Projektes ist es nicht, ein MNN selbst mechanisch umzusetzen. Es wird davon ausgegangen, dass

1. ein Vergleich der Verfahren in einer Simulation auch für das physische Trainieren der mechanischen Netzwerke aussagekräftig sein könnte und
2. ein „Vortrainieren“ eines physischen MNN mit einer Simulation die Gesamtzeit zum Trainieren verkürzt und somit Optimierungsverfahren, auch wenn sie nur in einer Simulation verwendet werden, von Nutzen sind.

3 Zeitplan

	Zeit	Thema	Person
<input checked="" type="checkbox"/>		Proof-of-Concept für MNNs	Matteo
<input checked="" type="checkbox"/>		Refactor des ersten Proof-of-Concept	Alex
<input checked="" type="checkbox"/>		Proof-of-Concept für analytische Umsetzung der Simulation	Matteo
<input checked="" type="checkbox"/>		Umsetzung des Trainierens mehrerer Verhaltensweisen gleichzeitig	Alex
<input checked="" type="checkbox"/>	bis 30.11.2023	JuFo-Anmeldung	
<input type="checkbox"/>		Projektskizze anfertigen	Alex
<input type="checkbox"/>	bis 3.12.2023	weitere Literatur lesen ([1, 5, 3])	Alex
<input type="checkbox"/>	bis 3.12.2023	Implementierung analytischer Definition für Simulation (Proof-of-Concept) im Hauptprojekt	Matteo
<input type="checkbox"/>	bis 10.12.2023	Proof-of-Concept für Gradient Descent	
<input type="checkbox"/>	bis 10.12.2023	Oberflächliche Recherche zu Pattern Search: Leicht umzusetzen? Wenn nein, dann erstmal außen vor lassen ([2] hat bereits gezeigt, dass schlechter als evolutionär, auch wenn mit stark unterschiedlicher Trainingsdauer)	
<input type="checkbox"/>	bis 17.12.2023	Verbesserung evolutionärer Algorithmus	
		Projektskizze aktuell halten	Alex
<input type="checkbox"/>		UML-Diagramm des Codes	Alex
<input type="checkbox"/>		Recherche zu Resonanzfrequenzen: wirklich ein Problem? Wenn ja, Resonanzfrequenzen in Simulation analysieren (FFT?)	
<input type="checkbox"/>	15.01.2023	Abgabe Bericht	

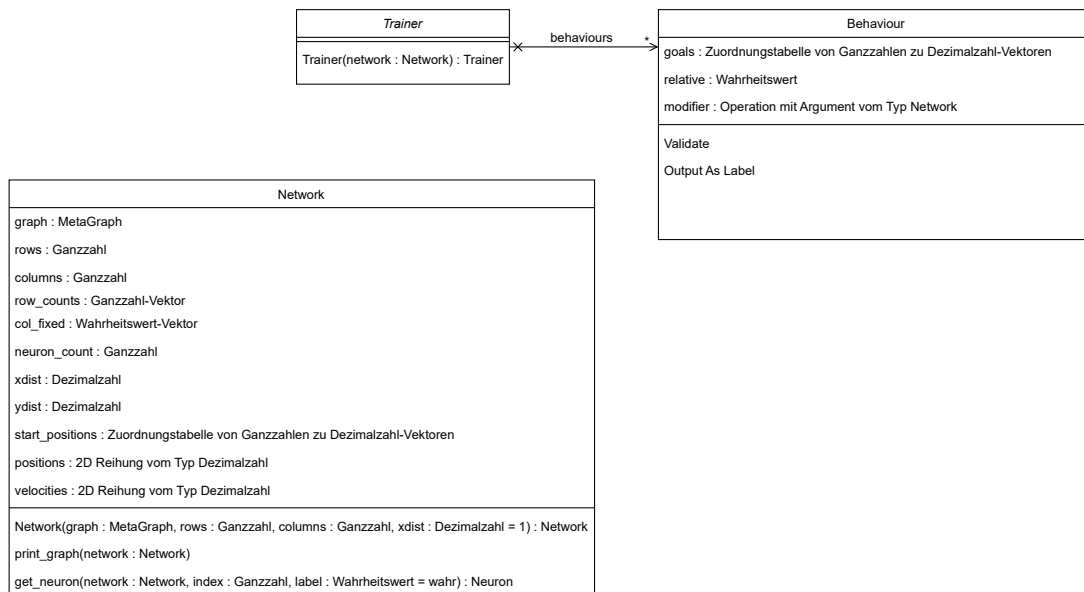


Abbildung 3: UML-Diagramm des Programm-Aufbaus.

4 Materialien

- Laptop (i7-11800H, 16 GB, RTX 3060)
- Julia
 - DifferentialEquations.jl und LinearAlgebra.jl für analytische Definition der Simulation des Federsystems
 - Graphs und MetaGraphsNext für Modellierung des Netzwerks als Graph
 - GLMakie und Observables für Visualisierung

5 Erwartete Schwierigkeiten

- Training von mehr als einer Verhaltensweise nicht erfolgreich.
- Ungenügende Leistung für evolutionäres Training.

6 Bisherige Ergebnisse

Quellcode: Siehe GitHub-Repository [4]. Siehe Abb. 3 für ein UML-Diagramm des Projekts.

Wie teilweise auch aus dem Zeitplan ersichtlich wurden bereits die Simulation, eine Optimierung dieser durch eine analytische Funktion, die Visualisierung und Animation des Netzwerks sowie ein simpler evolutionärer Algorithmus implementiert.

Trainieren mit diesem evolutionären Algorithmus hat bereits bei einzelnen, simplen Verhaltensweisen Erfolg gezeigt.

Bereits probierte und erfolgreich antrainierte Verhaltensweisen:

- Links eine Kraft nach rechts ausüben; Bewegung der Ausgangsneuronen nach rechts
- Links eine Kraft nach rechts ausüben; Bewegung der Ausgangsneuronen nach links
- Links eine Kraft nach links ausüben; Bewegung der Ausgangsneuronen nach rechts

7 Gruppe

Da es sich um Grundlagenforschung handelt, ist viel Recherche sowie Trial-and-Error nötig; es gibt keine fertige Anleitung. Deshalb arbeiten wir zusammen daran.

Aufgaben werden aufgeteilt, jedoch gibt es keine übergreifende Zuständigkeitsbereiche. Wir arbeiten beide an Programmierung, Recherche und Dokumentation.

8 Externe Kooperationspartner

Es gibt keine externe Kooperationspartner.

9 Literatur

Literatur

- [1] Jonathan B Hopkins, Ryan H Lee und Pietro Sainaghi. „Using binary-stiffness beams within mechanical neural-network metamaterials to learn“. In: *Smart Materials and Structures* 32.3 (Feb. 2023), S. 035015. DOI: 10.1088/1361-665X/acb519. URL: <https://dx.doi.org/10.1088/1361-665X/acb519>.
- [2] Ryan H. Lee, Erwin A. B. Mulder und Jonathan B. Hopkins. „Mechanical neural networks: Architected materials that learn behaviors“. In: *Science Robotics* 7.71 (2022), eabq7278. DOI: 10.1126/scirobotics.abq7278. eprint: <https://www.science.org/doi/pdf/10.1126/scirobotics.abq7278>. URL: <https://www.science.org/doi/abs/10.1126/scirobotics.abq7278>.
- [3] Anna Napolitano. „Intelligent Materials: Science Fiction to Science Fact“. In: *Physics* 15 (Nov. 2022). ISSN: 1943-2879. DOI: 10.1103/physics.15.184. URL: <http://dx.doi.org/10.1103/Physics.15.184>.
- [4] Alexander Reimer und Matteo Friedrich. *Simulation of MNNs*. GitHub Repository. 2023. URL: <https://github.com/Alexander-Reimer/Simulation-of-MNNs> (besucht am 27.11.2023).
- [5] Amir A. Zadpoor u. a. „Design, material, function, and fabrication of metamaterials“. In: *APL Materials* 11.2 (Feb. 2023), S. 020401. ISSN: 2166-532X. DOI: 10.1063/5.0144454. eprint: https://pubs.aip.org/aip/apm/article-pdf/doi/10.1063/5.0144454/16696272/020401_1_online.pdf. URL: <https://doi.org/10.1063/5.0144454>.