

BACHELOR THESIS  
Majid Moussa Adoyi

# Entwicklung einer Java-Anwendung zur Erklärbarkeit (XAI) der neuroevolutionären Algorithmen: NEAT und HyperNEAT in der Bildklassifizierung

---

FAKULTÄT TECHNIK UND INFORMATIK  
Department Informatik

Faculty of Engineering and Computer Science  
Department Computer Science

Majid Moussa Adoyi

# Entwicklung einer Java-Anwendung zur Erklärbarkeit (XAI) der neuroevolutionären Algorithmen: NEAT und HyperNEAT in der Bildklassifizierung

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung  
im Studiengang *Bachelor of Science Angewandte Informatik*  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing. Christian Lins  
Zweitgutachter: Prof. Dr. Stefan Sarstedt

Eingereicht am: 30. August 2024

**Majid Moussa Adoyi**

**Thema der Arbeit**

Entwicklung einer Java-Anwendung zur Erklärbarkeit (XAI) der neuroevolutionären Algorithmen: NEAT und HyperNEAT in der Bildklassifizierung

**Stichworte**

Erklärbare KI, Künstliche Intelligenz, NEAT, HyperNEAT, Anwendungsentwicklung

**Kurzzusammenfassung**

Diese Arbeit widmet sich der Verbesserung der Erklärbarkeit von NEAT und HyperNEAT. Ziel ist die Entwicklung eines Prototyps für eine XAI-Anwendung. Die Anwendung zielt darauf ab, durch algorithmische Transparenz die internen Prozesse der Algorithmen verständlich zu machen. Die Wirksamkeit der Anwendung wird anhand eines Experiments getestet, bei dem die Erreichung spezifischer Erklärbarkeitsziele evaluiert wird.  
...

**Majid Moussa Adoyi**

**Title of Thesis**

Development of a Java application for the explainability (XAI) of neuroevolutionary algorithms: NEAT and HyperNEAT in image classification

**Keywords**

Explainable AI, Artificial Intelligence, NEAT, HyperNEAT, Application development

**Abstract**

This thesis is dedicated to improving the explainability of NEAT and HyperNEAT. The aim is to develop a prototype for an XAI application. The application aims to make the internal processes of the algorithms transparent through algorithmic transparency. The effectiveness of the application is tested by means of an experiment, in which the achievement of specific explainability goals is evaluated. ...

# Inhaltsverzeichnis

Abbildungsverzeichnis	vi
<b>1 Einleitung</b>	<b>1</b>
1.1 Zielsetzung	2
<b>2 Theoretischer Hintergrund</b>	<b>3</b>
2.1 Künstliche Intelligenz (KI)	3
2.2 Erklärbare künstliche Intelligenz (erklärbare-KI)	3
2.2.1 Nutzerzentrierte Entwicklung von erklärbarer KI	5
2.2.2 Einblick in Modell-agnostische, Modell-spezifische und weitere Strategien	6
2.3 Bildklassifizierung mit Feedforward Neural Networks (FNNs)	7
<b>3 Neuroevolution of augmenting topologies (NEAT)</b>	<b>9</b>
3.1 Hypercube-based NEAT (HyperNEAT)	12
3.2 Weiterentwicklungen und Optimierungen von NEAT	15
<b>4 Algorithmen Implementierung</b>	<b>16</b>
4.1 Genotyp Implementierung	18
4.2 Phänotyp Implementierung	19
<b>5 Entwicklung der XAI-Anwendung (Prototyp)</b>	<b>20</b>
5.1 Vorgehen/Methodik	20
5.2 Problemidentifikation und Motivation	22
5.3 Definieren von Zielen	22
5.4 Design & Entwicklung	22
5.4.1 Startseite	25
5.4.2 Registerkarten	26
5.4.3 Simulation	27
5.5 Demonstration	30

5.6	Evaluation . . . . .	31
5.6.1	Umfragedesign . . . . .	31
5.6.2	Analyse der Umfrageergebnisse . . . . .	34
5.7	Kommunikation . . . . .	37
<b>6</b>	<b>Zusammenfassung / Fazit</b>	<b>38</b>
	<b>Literaturverzeichnis</b>	<b>40</b>
<b>A</b>	<b>Anhang</b>	<b>42</b>
A.1	Einführung . . . . .	42
A.2	HyperNEAT . . . . .	46
A.3	Mutation . . . . .	50
A.4	NEAT . . . . .	54
A.5	Parameterübersicht . . . . .	57
A.6	Reproduktion . . . . .	61
A.7	Speziation . . . . .	65
A.8	Evaluation . . . . .	68
A.9	Selektion . . . . .	71
A.10	Fragebogen - Antworten . . . . .	73
	<b>Selbstständigkeitserklärung</b>	<b>90</b>

# Abbildungsverzeichnis

2.1	Vorläufige Taxonomie für XAI-Anwendungen. . . . .	5
2.2	Beispielhafte Zerlegung eines Eingabebildes in Teilbereiche (nicht Pixel) und deren Verarbeitung. . . . .	7
3.1	Die Zuordnung von Genotyp zu Phänotyp. [17] . . . . .	9
3.2	Abgleich von Genomen für verschiedene Netzwerktopologien unter Verwendung von Innovationszahlen, um "Nachkommen" zu generieren. [17] . . . . .	11
3.3	Alternative Substratkonfigurationen. (a) ursprüngliche Gitterkonfiguration, (b) eine dreidimensionale Konfiguration von Knoten, die auf (0, 0, 0) zentriert sind, (c) eine Zustandsraum-Sandwich-Konfiguration, bei der ein Quellblatt von Neuronen direkt mit einem Zielblatt verbunden ist und (d) eine kreisförmige Konfiguration. [16] . . . . .	13
3.4	Grafischer Zusammenhang von Substrat und CPPN . . . . .	14
4.1	Architekturdiagramm zu den Algorithmen (NEAT und HyperNEAT) . . . . .	16
4.2	Architektur des Genotypen . . . . .	18
4.3	Architektur von Phänotypen . . . . .	19
5.1	Aufteilung der Generationen in fünf Prozesse . . . . .	24
5.2	Bild der Startseite . . . . .	25
5.3	Aufteilung der Generationen in fünf Prozesse . . . . .	26
5.4	Übersicht der konfigurierbaren Hyperparameter . . . . .	27
5.5	Übersicht der Simulation. In fünf Bereiche gegliedert. . . . .	28
5.6	Tabelle mit allen Individuen der Population. . . . .	28
5.7	Anzeige der Netzwerkstruktur des CPPNs und des verbundenen Substarts. . . . .	29

# 1 Einleitung

Eine vom Bundesministerium für Wirtschaft und Energie in Auftrag gegebene Studie mit dem Titel „ERKLÄRBARE KI - Anforderungen, Anwendungsfälle und Lösungen“ kam im Jahr 2021 zu der Einschätzung, dass Dienstleistungen und Produkte, die auf dem Einsatz von Künstlicher Intelligenz (KI) basieren, 2025 rund 13 Prozent (488 Milliarden Euro) des Bruttoinlandsprodukts ausmachen werden – die Auswirkungen des Erfolgs von ChatGPT-3, 3.5 und 4 noch nicht mit inbegriffen. Dieselbe Studie beobachtete, dass die Kenntnisse über Methoden zur Erklärbarkeit von neuronalen Netzen unter den KI-Entwicklern nur partiell vorhanden sind und ein Bedarf an wissenschaftlich-methodischem Diskurs zwischen den Disziplinen der Informatik und der Mathematik in Grundlagenforschung und Ausbildung besteht. [5]

Die Fähigkeit, komplexe KI-Modelle und deren Ergebnisse auf verständliche Weise zu erklären, ist entscheidend, um effektiv mit Interessengruppen zu kommunizieren, die möglicherweise keine tiefen technischen Kenntnisse haben. Studierende des Fachbereichs Informatik und verwandter Studiengänge sollten bereits im Verlauf ihres akademischen Werdegangs die Kompetenz zur Entwicklung von erklärbaren KI-Modellen kultivieren. Ein fundiertes Wissen über die zugrunde liegenden Algorithmen und deren Funktionsweise ermöglicht es zukünftigen Fachkräften, die Mechanismen und Entscheidungen von KI-Systemen nachvollziehbar zu machen. Dies ist besonders wichtig, da die zunehmende Verbreitung von KI-Anwendungen in verschiedenen Lebensbereichen ein Grundverständnis bei durchschnittlichen Nutzern erfordert, um Vertrauen in die Technologie zu schaffen und Missverständnisse zu vermeiden.

Ein Vertreter der schwer erklärbaren Algorithmen stammt aus der Familie der evolutionären Algorithmen und modifiziert neuronale Netze: NeuroEvolution of Augmenting Topologies (NEAT). Der Algorithmus zeigt ein breites Anwendungsspektrum bei verschiedenen Problemstellungen und gilt daher als ein interessanter Ansatz im Bereich der Künstlichen Intelligenz. Im Rahmen meiner Bachelorarbeit soll das Problem der Erklärbarkeit von NEAT und HyperNEAT behandelt werden.

## 1.1 Zielsetzung

Gegenwärtig konnte im Zuge der umfassenden Recherche keine Implementierung der NEAT-Algorithmen identifiziert werden, die NEAT und HyperNEAT in einer Anwendung integriert und dabei den spezifischen Schwerpunkt auf Erklärbarkeit legt. Die beiden genannten Ausführungen des NEAT-Algorithmus stellen das heutige Fundament für sämtliche nachfolgende NEAT-Optimierungsansätze dar und eignen sich daher als Gegenstand der Forschung. Vor diesem Hintergrund bietet sich die Entwicklung einer Anwendung im Bereich der erklärbaren künstlichen Intelligenz (XAI) an, die sowohl NEAT als auch HyperNEAT integriert. Eine solche XAI-Anwendung könnte dazu beitragen, die Entscheidungsprozesse dieser Algorithmen transparenter und nachvollziehbarer zu machen. Dies ist besonders relevant, da die Komplexität von evolutionären Algorithmen wie NEAT und HyperNEAT häufig zu einem „Black-Box“-Charakter führt, bei dem die interne Funktionsweise für Anwender schwer zu durchschauen ist. Auf Basis dieser Problematik resultiert die folgende Forschungsfrage:

*"Kann eine speziell für die NEAT-Algorithmen entwickelte XAI-Anwendung die Verständlichkeit und das Wissen über diese Algorithmen für die Anwender verbessern?"*

Um die Umsetzbarkeit zu demonstrieren, soll eine prototypische Anwendung entwickelt werden. Die Wahl des Problems ist dabei mit Bedacht zu treffen, da ein zu komplexes Problem die Wirksamkeit der Anwendung beeinträchtigen könnte. Das Problem der Bildklassifizierung scheint aufgrund seiner großen Bekanntheit – auch über die Forschung hinaus – eine gute Wahl zu sein. Die Wirksamkeit der Anwendung soll anhand einer kleinen Gruppe ausgewählter Anwender gemessen werden. Eine entsprechende Umfrage soll mindestens vier der neun von Chromik und Schluessler definierten Erklärbarkeitsziele abhandeln: Transparenz, Vertrauen, Wirksamkeit und Effizienz [3]. Die Probanden sollen durch die Nutzung der Anwendung dazu befähigt werden, die Wirkungsmechanismen zu identifizieren (Transparenz), Vertrauen im Umgang mit den Algorithmen zu erlangen (Vertrauen), sicherer und schneller im Umgang mit den Algorithmen zu werden (Wirksamkeit, Effizienz). In nachfolgenden Forschungen könnte der Prototyp weiterentwickelt und optimiert werden, um seine Anwendungsmöglichkeiten und die Erreichung der Erklärbarkeitsziele weiter zu verbessern.

## 2 Theoretischer Hintergrund

### 2.1 Künstliche Intelligenz (KI)

Kurzweil definierte Künstliche Intelligenz (KI) bereits 1990 als „Die Kunst, Maschinen zu erschaffen, die Funktionen erfüllen, die, werden sie von Menschen ausgeführt, der Intelligenz bedürfen“ [6]. Diese Definition ist eine von vielen, aber eine greifbare. Heutige Techniken und Verfahren zur Erstellung von Künstliche Intelligenz beziehen sich in der Regel auf das Maschinelle Lernen (ML) welche es KI-Systemen ermöglicht aus Daten zu lernen, ohne explizit programmiert worden zu sein. Anstatt von Menschen definierten Anweisungen (nach dem WENN-DANN Muster) zu folgen, analysieren die Modelle große Mengen von Daten, erkennen Muster darin und lernen, Vorhersagen oder Entscheidungen, basierend auf diesen Mustern zu treffen. Aber auch auf Fakten und Regeln basierende Expertensysteme finden Anwendung in der KI.

### 2.2 Erklärbare künstliche Intelligenz (erklärbare-KI)

Die menschliche Fähigkeit, Kausalitäten und Korrelationen in umfangreichen Datensätzen zu identifizieren, ist begrenzt. Eine holistische Perspektive, in der ein Mensch jeden Schritt in einem KI-System nachvollziehen kann ist unwirklich. Das Vertrauen in KI-Systeme kann aber nur bestehen, wenn resultierende Entscheidungen für den Anwender nachvollziehbar sind. Es erfordert alternative (technische) Ansätze, um einen umfassenden Überblick zu gewähren, ohne zwangsläufig alle Komponenten des Prozesses vollständig zu erfassen.

Das Problem der Erklärbarkeit von künstlicher Intelligenz (KI) ist so alt wie die künstliche Intelligenz selbst. Die Begriffe „Explainable Artificial Intelligence“ (XAI), „Interpretable Machine Learning“ (IML) und „Algorithmic transparency“ sind historisch gewachsen

und lassen sich nicht eindeutig auf einen einzelnen Ursprung zurückführen. Im Allgemeinen beschreiben diese Termini die Fähigkeit von Modellen und Algorithmen, ihre Entscheidungen und Wirkungsmechanismen verständlich für Menschen zu erklären.[10] [11] [5]

Erklärbarkeit und Interpretierbarkeit haben sich in der Forschung von XAI als Modelleigenschaften etabliert, wobei weiterhin ein fortlaufender Diskurs bezüglich dieser Aspekte besteht. Modelle, welche für den Anwender per se transparent und verständlich - also interpretierbar - sind, gelten als “White-Box” Modelle. Alle anderen Modelle, für die keines der genannten Attribute zutrifft, gelten als “Erklärbar” und zählen zu der Rubrik der “Black-Box” Modelle. Es bedarf diesen Modellen an Strategien und Werkzeugen, um sie für Anwender verständlich zu machen. Neuronale Netze sind die bekanntesten Vertreter dieser Rubrik. [10] [5]

Michael Chromik und Martin Schuessler haben in ihrer Arbeit spezifische Ziele der Erklärbarkeit identifiziert, die in den Entwicklungsprozess von XAI-Anwendungen integriert werden sollten [3]. Diese Ziele umfassen:

- Transparenz
- Prüfbarkeit
- Vertrauen
- Überzeugungskraft
- Zufriedenheit
- Effektivität
- Effizienz
- Bildung
- Debugging

*Transparenz* bedeutet, dass die Funktionsweise eines Modells offen und nachvollziehbar ist. *Prüfbarkeit* ermöglicht es, die Entscheidungen eines Modells systematisch zu überprüfen. *Vertrauen* entsteht, indem Benutzer die Zuverlässigkeit eines Modells nachvollziehen können. *Überzeugungskraft* sorgt dafür, dass die Erklärungen eines Modells den Benutzer von der Richtigkeit der Entscheidungen überzeugen. *Zufriedenheit* beschreibt, wie gut die Erklärungen den Erwartungen und Bedürfnissen der Benutzer entsprechen. *Effektivität* misst, wie gut Erklärungen den Benutzer dabei unterstützen, eine Aufgabe erfolgreich abzuschließen. *Effizienz* bezieht sich darauf, dass Erklärungen schnell und ohne unnötige Komplexität bereitgestellt werden. *Bildung* zielt darauf ab, die Nutzer zu befähigen, zu verallgemeinern und zu lernen. *Debugging* ermöglicht es, durch Erklärungen Fehler im Modell zu identifizieren und zu beheben. [3]

### 2.2.1 Nutzerzentrierte Entwicklung von erklärbarer KI

Wojciech Samek et al. haben vier mögliche Ansätze zum Erreichen von selektiver Erklärbarkeit eingeführt, die wiederum ein breites Sammelsurium an verschiedensten Techniken beinhalten: 1. Erklärung individueller Ausgaben eines Modells, 2. Erklären des Modellverhaltens, 3. Erklärung anhand repräsentativer Beispiele (verwendet konkrete Fälle oder Beispiele, um die Funktionsweise des Modells zu illustrieren) und 4. Erklärung der gelernten Repräsentation (bezieht sich auf die Interpretation der internen Repräsentationen, die das Modell während des Trainings gelernt hat). Es ist unerlässlich zu erwähnen, dass die Wahl unter den verschiedenen Ansätzen in starker Abhängigkeit von ihrem Zweck und der Zielgruppe liegt. [15]

Zu der gleichen Auffassung kommen auch Michael Chromik und Martin Schluessler, welche den Aspekt der unterschiedlichen Zielgruppen aufgreifen. Die beiden Wissenschaftler forschen jenseits der technischen Möglichkeiten, und legen den Fokus auf die Anwender der XI-Anwendungen. Sie präsentieren in ihrem Werk: "A Taxonomy for Human Subject Evaluation of Black-Box Explanations in XAI" einen interessanten ersten Entwurf zum evaluieren von XAI-Anwendungen durch ihre Anwender (siehe 2.1).

Task Dimensions			Study Design Dimensions																																				
<b>Intended Explanation Goal</b> [24, 30, 32] Transparency      Persuasiveness      Satisfaction Scrutability      Effectiveness      Efficiency Trust      Education      Debugging			<b>Study Approach</b> Qualitative Quantitative Mixed	<b>Treat. Assignment</b> Within-subjects Between-subjects	<b>Treat. Combination</b> [24] Single Explanation With and Without Explanation Altern. Explanation Altern. Explanation Interface																																		
<b>Human Involvement</b> [18] Feedback Feedforward	<table border="1"> <thead> <tr> <th rowspan="2">Task Type [4, 18, 33, 14]</th> <th colspan="3">Information given to Participant</th> </tr> <tr> <th>Input</th> <th>Explanation</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>Verification</td> <td>✓</td> <td>✓</td> <td>✓</td> </tr> <tr> <td>Forced Choice</td> <td>✓</td> <td>✓, ..., ✓</td> <td>✓</td> </tr> <tr> <td>Forward Simulation</td> <td>✓</td> <td>✓</td> <td>?</td> </tr> <tr> <td>Counterfactual Simulation</td> <td>✓, ?</td> <td>✓</td> <td>✓, ✓</td> </tr> <tr> <td>"Clever Hans" Detection</td> <td>✓</td> <td>✓</td> <td>✓</td> </tr> <tr> <td>System Usage</td> <td>✓</td> <td>✓</td> <td>✓</td> </tr> <tr> <td>Annotation</td> <td>✓</td> <td>?</td> <td>✓</td> </tr> </tbody> </table> ✓ = information provided to participant ? = information inquired of participant			Task Type [4, 18, 33, 14]	Information given to Participant			Input	Explanation	Output	Verification	✓	✓	✓	Forced Choice	✓	✓, ..., ✓	✓	Forward Simulation	✓	✓	?	Counterfactual Simulation	✓, ?	✓	✓, ✓	"Clever Hans" Detection	✓	✓	✓	System Usage	✓	✓	✓	Annotation	✓	?	✓	<b>Participant Incentivation</b> [28, 29, 25] Monetary Non-Monetary
Task Type [4, 18, 33, 14]	Information given to Participant																																						
	Input	Explanation	Output																																				
Verification	✓	✓	✓																																				
Forced Choice	✓	✓, ..., ✓	✓																																				
Forward Simulation	✓	✓	?																																				
Counterfactual Simulation	✓, ?	✓	✓, ✓																																				
"Clever Hans" Detection	✓	✓	✓																																				
System Usage	✓	✓	✓																																				
Annotation	✓	?	✓																																				
<b>Evaluation Level</b> [11] Test of Satisfaction Test of Comprehension Test of Performance				<b>Number of Participants</b> Low High																																			
<b>Abstraction Level</b> [4] Human-grounded Application-grounded	<b>Participant Foresight</b> [21] Intrinsic Extrinsic	<table border="1"> <thead> <tr> <th rowspan="2">Participant Type [19]</th> <th colspan="2">Level of Expertise</th> </tr> <tr> <th>AI</th> <th>Domain</th> </tr> </thead> <tbody> <tr> <td>(AI) Novice User</td> <td>low</td> <td>low</td> </tr> <tr> <td>Domain Expert</td> <td>low</td> <td>high</td> </tr> <tr> <td>AI Expert</td> <td>high</td> <td>low</td> </tr> </tbody> </table>		Participant Type [19]	Level of Expertise		AI	Domain	(AI) Novice User	low	low	Domain Expert	low	high	AI Expert	high	low	<b>Participant Recruiting</b> Field Study Lab Study Online Study Crowd-sourcing																					
Participant Type [19]	Level of Expertise																																						
	AI	Domain																																					
(AI) Novice User	low	low																																					
Domain Expert	low	high																																					
AI Expert	high	low																																					

Participant Dimensions

Abbildung 2.1: Vorläufige Taxonomie für XAI-Anwendungen.

Chromik und Schluessler entwickeln in diesem Entwurf eine systematische Methodik für die Gestaltung von XAI-Anwendungen, die primär die Bedürfnisse und Erwartungen der Endnutzer berücksichtigt. Sie betonen, dass die Entwicklung solcher Anwendungen einer strukturierten Vorgehensweise bedarf, die den vielfältigen Anforderungen unterschiedlicher Anwendungsbereiche und Zielgruppen gerecht wird. Ihre Forschung zeigt, dass es keine universelle Methode zur Schaffung von Erklärbarkeit für KI-Modelle gibt; vielmehr erfordert die Heterogenität der Anwendungsfälle flexible und kontextspezifische Erklärungsstrategien. [3]

### 2.2.2 Einblick in Modell-agnostische, Modell-spezifische und weitere Strategien

Die gängigsten Ansätze zum Erreichen der Erklärbarkeit von KI-Anwendungen fallen unter die Rubriken Modell-agnostische und Modell-spezifische Methoden. Modell-spezifische Methoden sind darauf ausgelegt, Erklärungen für eine bestimmte Art von Modell zu liefern, indem sie die spezifische Struktur und Mechanismen des Modells nutzen, wie z.B. Grad-CAM für neuronale Netze. Diese Methoden sind eng an die Modellarchitektur gebunden und können daher detaillierte, aber weniger universelle Erklärungen bieten. Modell-agnostische Methoden hingegen sind unabhängig von der Modellarchitektur und können auf verschiedene Arten von Modellen angewendet werden. Sie bieten allgemeine Erklärungen, indem sie das Modell als Black-Box behandeln und beispielsweise mit Methoden wie LIME oder SHAP die Vorhersagen eines Modells erklären, ohne dessen innere Funktionsweise zu kennen. [5] Beide Ansätze verfolgen das gleiche Ziel: Sie sollen Anwender befähigen, die Ergebnisse der Modelle nachvollziehbar zu machen, insbesondere das "Warum" hinter den Entscheidungen des Modells zu verstehen.

Darüber hinaus befassen sich weitere Ansätze mit der Frage des "Wie: Sie zielen darauf ab, interne Abläufe und Mechanismen, die zu den Entscheidungen führen, transparent zu machen. Beispiele dafür sind Algorithm Tracing and Debugging Tools (Algorithmusverfolgungs- und Debugging-Werkzeuge). Diese geben den Anwendern Haltepunkte vor und ermöglichen die Komplexität von KI-Anwendungen, durch saubere logische Schnitte auf mehrere Teilaspekte zu gliedern [7]. Model Internals Visualizations (Modell interne Visualisierungen) stellen Strukturen und Prozesse grafisch da. Durch die Anwendung solcher Visualisierungen können beispielsweise neuronale Netzwerke in ihrer Tiefe untersucht werden.

## 2.3 Bildklassifizierung mit Feedforward Neural Networks (FNNs)

Im weiteren Verlauf dieser Arbeit werden die Algorithmen NEAT und HyperNEAT erläutert und ihre Anwendung im Kontext der Bildklassifizierung in einer zu entwickelnden XAI-Anwendung demonstriert.

In der Domäne der Bildklassifizierung ist die Pixel-basierte Klassifizierung von Bildern unter Einsatz künstlichen neuronaler Netze eine effektive Prognosemethodik. Die künstlichen neuronalen Netze (KNN) sind in ihren Netztopologien weitgehend vereinheitlicht. Die Farbinformationen oder Graustufen von Bildpixeln werden den Eingangsneuronen als Aktivierungswerte zugeführt, während die verschiedenen Vorhersageklassen durch die Ausgangsneuronen repräsentiert werden (siehe 2.2). Topologieunterschiede, welche die Art des neuronalen Netzes bestimmen, ergeben sich erst in den versteckten Schichten. Diese Schichten sind für die Extraktion und Transformation von Merkmalen verantwortlich, die für die letztendliche Klassifikation relevant sind.

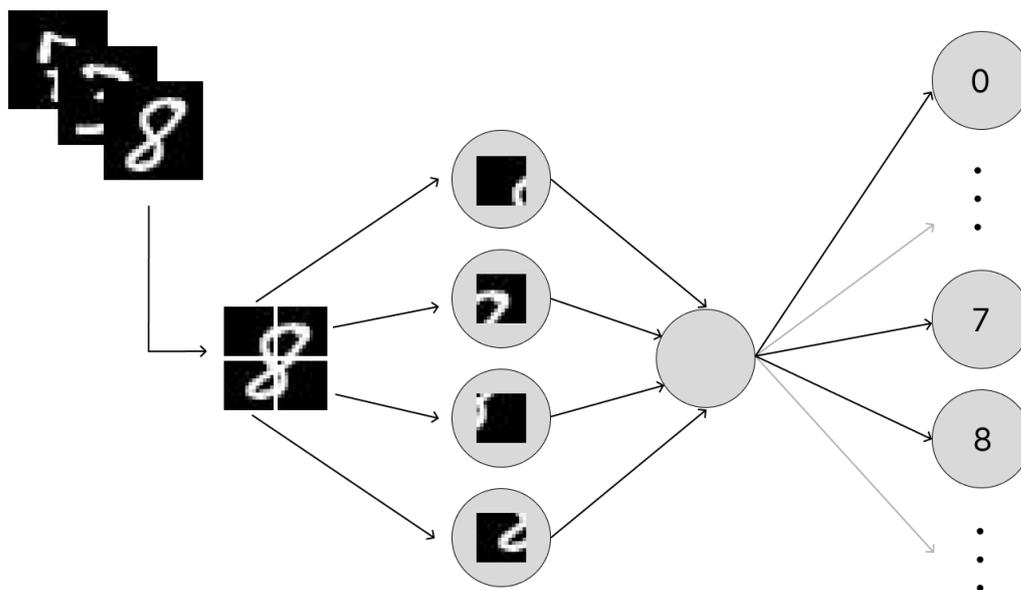


Abbildung 2.2: Beispielhafte Zerlegung eines Eingabebildes in Teilbereiche (nicht Pixel) und deren Verarbeitung.

Das einfachste und grundlegendste Modell eines neuronalen Netzes, bei dem die Informationen in eine Richtung - von den Eingabeknoten zu den Ausgabeknoten - fließen, ist das Feedforward Neural Networks (FNN). Es enthält keine Schleifen oder Rückkopplungen. Jede versteckte Schicht nimmt die Ausgabe der vorhergehenden Schicht als Eingabe und verarbeitet sie weiter. Die Aktivierungsfunktion, die in den Neuronen eines neuronalen Netzes verwendet wird, ist von zentraler Bedeutung, da sie nichtlineare Transformationen ermöglicht. Zunächst wird im Neuron eine lineare Kombination der Eingaben berechnet, also eine gewichtete Summe. Durch die anschließende Anwendung der Aktivierungsfunktion auf diese gewichtete Summe erhält das neuronale Netz die Fähigkeit, komplexe und nichtlineare Muster in den Daten zu erkennen und zu verarbeiten. Zu den gängigen Aktivierungsfunktionen gehören die Sigmoid-Funktion, die ReLU (Rectified Linear Unit) und die Tanh-Funktion. Schließlich erreicht die Information die Ausgabeschicht, wo die Neuronen die Wahrscheinlichkeiten der verschiedenen Klassen (z. B. „Hund“ oder „Katze“) berechnen. Die Klasse mit der höchsten Wahrscheinlichkeit wird dann als das Modellvorhersageergebnis ausgewählt.[14]

FNNs sind konzeptionell einfach und eignen sich hervorragend für die Einführung in die Thematik der neuronalen Netze. Darüber hinaus sind sie im Kontext von XAI besonders wertvoll, da es nicht nur auf die Leistungsfähigkeit der Modelle ankommt, sondern auch darauf, dass diese für Anwender verständlich und zugänglich bleiben.

Obwohl FNNs in ihrer Einfachheit oft als Grundmodell dienen, haben sie in der Praxis Einschränkungen. Das gilt insbesondere bei der Verarbeitung von komplexeren, nicht-linearen Daten. Diese kommen allerdings häufig in der Bildklassifizierung vor und somit werden in modernen Anwendungen auch komplexere Netztopologien verwendet - beispielsweise die Convolutional neural networks. Diese sind speziell darauf ausgelegt, räumliche Hierarchien in den Bilddaten zu erfassen und zu verarbeiten.

### 3 Neuroevolution of augmenting topologies (NEAT)

Der NEAT-Algorithmus wurde im Jahr 2002 von Kenneth O. Stanley und Risto Miikkulainen erstmals vorgestellt. Er gehört zu der Gattung der evolutionären Algorithmen und diese stellen wiederum eine Teilmenge des maschinellen Lernens dar. Der NEAT-Algorithmus ist zur Entwicklung und Optimierung von neuronalen Netzen konzipiert. Durch das Evolvieren von neuronalen Netzen, werden für ein gegebenes Problem geeignete Lösungskandidaten in einem Lösungsraum gesucht. Hierbei repräsentiert jeder Lösungskandidat ein Individuum, das zunächst ein Genom für ein neuronales Netz bildet. Ein Genome besteht aus Node-Genes und Connection-Genes. Das Genom fungiert als Bauplan, aus dem das neuronale Netz (der Phenotyp) generiert wird (siehe Abbildung 1) [17].

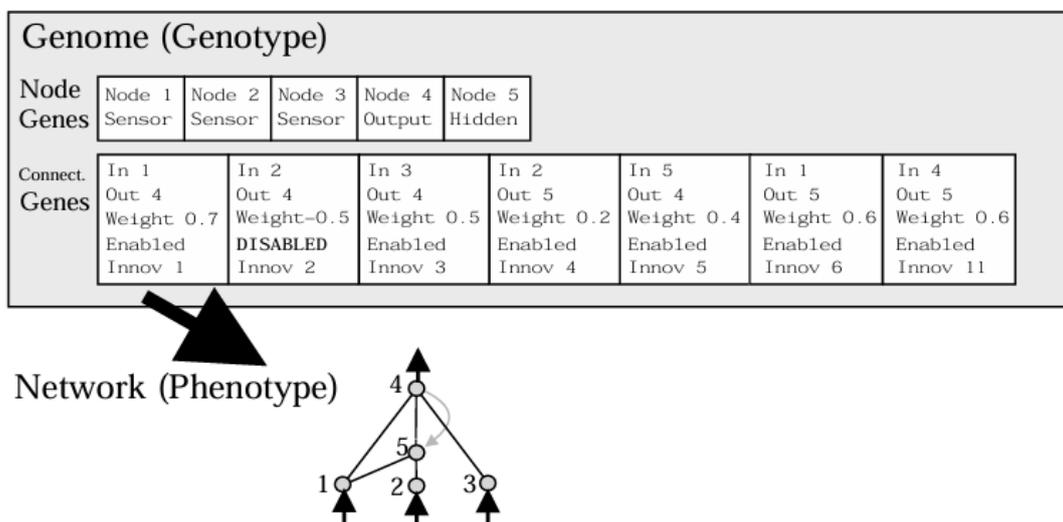


Abbildung 3.1: Die Zuordnung von Genotyp zu Phänotyp. [17]

Die durch NEAT evolvierten neuronalen Netze folgen, abgesehen von der Aufteilung in die standardmäßige Schichten-Architektur (Eingangsschicht, versteckte Schicht und Ausgangsschicht), keinen weiteren topologischen Restriktionen. Dies macht NEAT für eine Vielzahl von Optimierungsproblemen anwendbar. Das Modell ist somit in der Lage, flexible und unkonventionelle Netzwerktopologien zu entwickeln, die an die spezifischen Anforderungen des Problems angepasst sind.

Im Detail initialisiert und verwaltet der Algorithmus eine Population von Individuen, welche in Generationsschritten (oder auch Evolutionszyklen) evolviert werden. Diese Entwicklung erfolgt angelehnt an die Biologie: Mit der Vermischung von Genen durch die Kreuzung von Individuen (siehe Abbildung 2) und dem Auftreten zufälliger Mutationen. NEAT stellt sicher, dass neue genetische Mutationen eindeutig identifiziert und verfolgt werden können. Jedes Mal, wenn eine neue Verbindung oder ein neuer Knoten durch Mutation entsteht, wird dieser mit einer einzigartigen Innovationsnummer versehen. Diese Nummerierung ermöglicht es, die genetische Historie von Netzwerken nachzuvollziehen und erleichtert das Zuordnen von gleichen Gensequenzen während der Kreuzung.

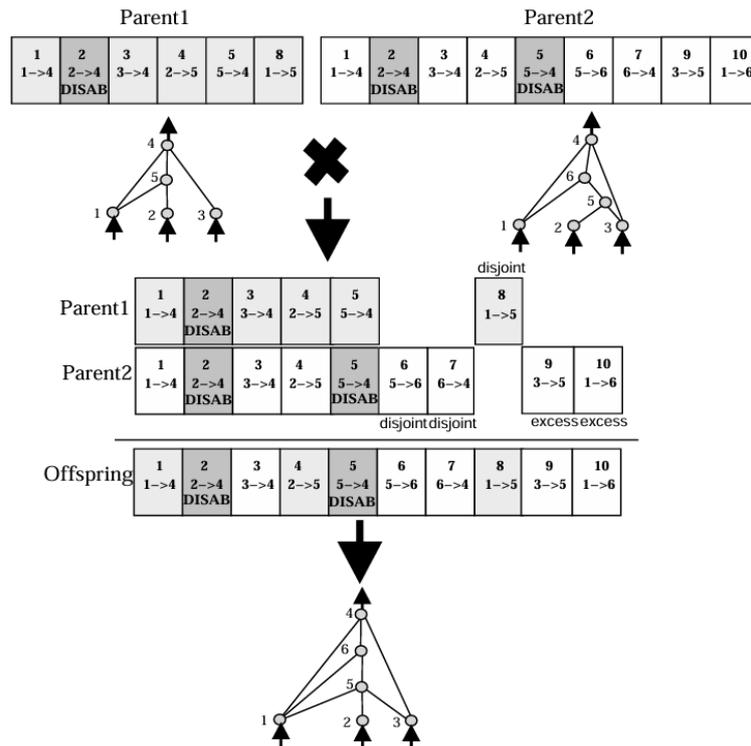


Abbildung 3.2: Abgleich von Genomen für verschiedene Netzwerktopologien unter Verwendung von Innovationszahlen, um "Nachkommen" zu generieren. [17]

Die Population wird in jedem der Generationsschritte in disjunkte Spezies unterteilt. Mit der Aufteilung in die Spezies, soll ein isolierter Schutzraum für evolvierte Individuen geschaffen werden, die ihr volles Potenzial noch nicht entfalten konnten. Dieser Schritt schafft evolutionäre Vielfalt. Statt mit allen Individuen der Gesamtpopulation zu konkurrieren, muss ein Individuum dies nur innerhalb der eigenen Spezies tun.

$$\delta = \frac{c_1 \cdot E}{N} + \frac{c_2 \cdot D}{N} + c_3 \cdot \bar{W}$$

Die Ermittlung der Zugehörigkeit eines Individuums zu einer Spezies erfolgt durch die Distanzberechnung (*delta*). Dabei ist *E* die Anzahl der überschüssigen Gene (engl. Excess Genes), die in einem Genom vorhanden sind, aber außerhalb des Gültigkeitsbereichs des anderen Genoms liegen. Die disjunkten Gene (*D*) befinden sich in einem Genom, haben jedoch keine entsprechende Übereinstimmung im anderen Genom innerhalb des gleichen

Bereichs.  $\overline{W}$  (engl. "Weight") ist der durchschnittliche Gewichtsunterschied zwischen den übereinstimmenden Verbindungs-Genen der beiden Genome (siehe 3.2). [17]

Die Bewertung von Individuen erfolgt in jeder Generation basierend auf einer Fitness-Funktion. Der Algorithmus selektiert nur die am besten angepassten Kandidaten. Damit stellt der Selektionsprozess sicher, dass nur diejenigen Individuen weiterentwickelt werden, die am besten auf die aktuelle Umgebung angepasst sind, wodurch die Qualität der Population von Generation zu Generation verbessert wird. Nicht-selektierte Individuen werden aus der Population entfernt und durch Reproduktion auf Basis von Kreuzungen oder durch zufällig generierte Individuen ersetzt.

Der Anteil, den jede Spezies zur Reproduktion beitragen darf, wird anhand einer geteilten Fitness bemessen. Dabei wird die Gesamtfitness der Spezies durch die Anzahl ihrer Mitglieder geteilt, um eine durchschnittliche Fitness zu berechnen. Diese Methode stellt sicher, dass nicht nur große Spezies bevorzugt werden, sondern auch kleinere Spezies mit höherer Innovationskraft eine Chance erhalten, zur nächsten Generation beizutragen.

## 3.1 Hypercube-based NEAT (HyperNEAT)

HyperNEAT (Hypercube-based NeuroEvolution of Augmenting Topologies) ergänzt NEAT um das Konzept der Mustererkennung, indem es wiederkehrende Muster identifiziert, um die Komplexität der Netztopologie zu reduzieren. Anstatt für jeden Anwendungsfall einen geeigneten Lösungskandidaten zu entwickeln, welcher in Abhängigkeit zur Größe der Eingangsdaten steht, ermöglicht HyperNEAT die Anwendung seiner Individuen auf skalierte Probleme. Dies geschieht unter Zuhilfenahme des Substrats, das als einer Art Leinwand interpretiert werden kann. Das Substrat ordnet die Neuronen in einem mehrdimensionalen Raum an und kann unterschiedlich konfiguriert sein (siehe 3.3). Die Wahl der Substratkonfiguration hängt dabei von der spezifischen Problemstellung ab.

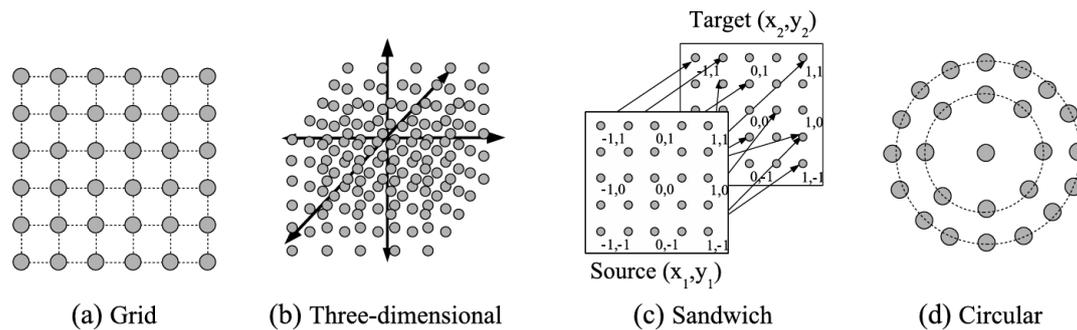


Abbildung 3.3: Alternative Substratkonfigurationen. (a) ursprüngliche Gitterkonfiguration, (b) eine dreidimensionale Konfiguration von Knoten, die auf  $(0, 0, 0)$  zentriert sind, (c) eine Zustandsraum-Sandwich-Konfiguration, bei der ein Quellblatt von Neuronen direkt mit einem Zielblatt verbunden ist und (d) eine kreisförmige Konfiguration. [16]

Für Bildklassifizierungsaufgaben wird das Substrat so organisiert, dass es die räumliche Struktur der Eingabebilder und Vorhersageklassen widerspiegelt. Die Neuronen im Substrat sind in einer Weise angeordnet, dass ihre Positionen den Pixelpositionen im Bild entsprechen. Diese Anordnung ermöglicht es, die topologischen Merkmale und lokalen Muster der Bilder zu nutzen.

Ein zentrales Merkmal von HyperNEAT ist die indirekte Kodierung, die es ermöglicht, komplexe Netztopologien effizient zu erzeugen. Anstatt jede Verbindung und jedes Gewicht direkt zu kodieren, evolviert HyperNEAT ein Connectivity Compositional Pattern-Producing Network (Connectivity-CPPN), welches sich einzig auf das Erstellen von Verbindungen im Substrat und deren Gewichtsrechnung beschränkt. Ein Connectivity-CPPN ist ein Netzwerk, das aus Eingabeknoten (Positionsdaten), versteckten Knoten (mathematische Aktivierungsfunktionen) und einem Ausgabeknoten (Verbindungsgewicht) besteht. Durch die Kombination dieser Elemente erzeugt das Netzwerk komplexe, häufig symmetrische oder wiederkehrende Muster, welche die Verbindungen innerhalb eines neuronalen Netzes gezielt und effizient steuern. Für diese Berechnung benötigt das CPPN, die Positionsinformationen  $(x_1, x_2, y_1, y_2)$  von je zwei Neuronen des Substrats. Die beiden Neuronen müssen dabei in unterschiedlichen Dimensionen des Substrats liegen (siehe 3.4).

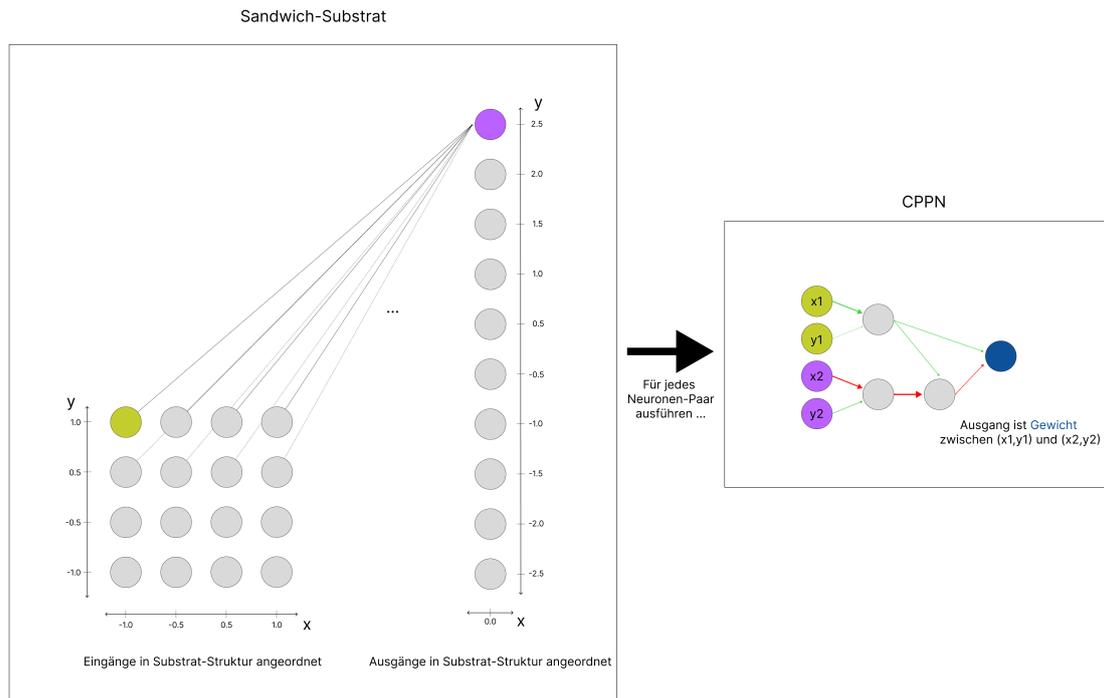


Abbildung 3.4: Grafischer Zusammenhang von Substrat und CPPN

Das resultierende Gewicht des CPPN wird einer Schwellenwertüberprüfung unterzogen. Falls das Gewicht diesen Schwellenwert unterschreitet, ist es nicht bedeutend genug für die weiteren Berechnungen des neuronalen Netzwerks und die Verbindung wird nicht implementiert. Dieser Schritt dient der Komplexitätsreduktion und erzeugt als Endresultat ein partiell verbundenes Substrat, das als Phänotyp des evolvierten Genotyps (Connectivity-CPPN) fungiert. Das Connectivity-CPPN in HyperNEAT folgt denselben Prinzipien der Evaluation, Selektion, Reproduktion, Speziation und Mutationen, die bereits in NEAT etabliert wurden (siehe 3).

Während NEAT darauf abzielt, die Topologie des Phänotyps direkt zu evolvieren, ermöglicht die indirekte Kodierung bei HyperNEAT, Netzwerke (Connectivity-CPPNs) zu evolvieren, die räumliche und strukturelle Merkmale des Problems berücksichtigen und verarbeiten. Dieser Ansatz führt zu Phänotypen, die nicht nur an die spezifischen Anforderungen des Problems angepasst sind, sondern auch eine höhere Effizienz und Generalisierungsfähigkeit aufweisen.

## 3.2 Weiterentwicklungen und Optimierungen von NEAT

Die beiden Algorithmen NEAT (3) und HyperNEAT (3.1) weisen jeweils spezifische Nachteile auf. Ein Nachteil des NEAT-Algorithmus besteht in seiner Skalierbarkeit. Die Größe der Netztopologie ist abhängig von dem zugrundeliegenden Problem: Für Probleme, die sich mit kleinen Netztopologien lösen lassen, ist NEAT geeignet. Bei immer größer werdenden Netztopologien, verliert der Algorithmus an Effizienz [9]. Der Nachteil bei HyperNEAT besteht hingegen darin, dass zur effizienten Lösung von Problemen, eine für das Problem geeignete Substratstruktur benötigt wird [18]. Die Konfiguration des Substrates liegt aber in Entwicklerhand und bietet somit viel Spielraum für suboptimale Lösungen.

Auch über das Jahr 2009 hinaus entstanden weitere nennenswerte Erweiterungen, die darauf abzielen, die Leistungsfähigkeit und Flexibilität von HyperNEAT zu verbessern. Insbesondere in Bezug auf die Optimierung des Substrates. Zu diesen Erweiterungen gehören Evolvable-substrate HyperNEAT (ES-HyperNEAT - 2012), Evolvablemulti-spatial-substrate HyperNEAT (EMSS HyperNEAT - 2019) und Deep Evolvable-Substrate HyperNEAT (DES HyperNEAT - 2020). [18] [19] [20]

## 4 Algorithmen Implementierung

Im folgenden Abschnitt wird eine Analyse des strukturellen Aufbaus der Algorithmen präsentiert. Darüber hinaus werden spezifische Besonderheiten der Implementierung erläutert. Weiterführende Details können im Quellcode, der im Anhang referenziert ist (siehe [2]), eingesehen werden.

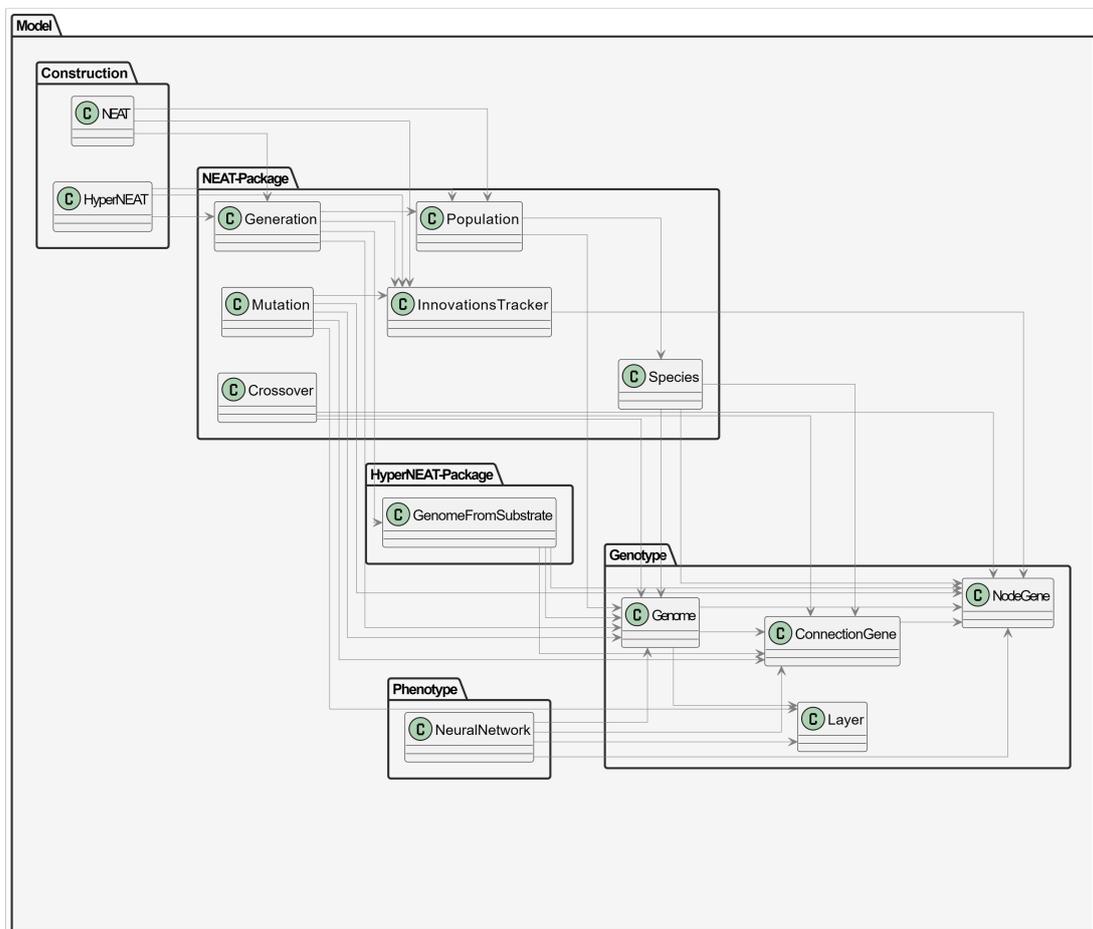


Abbildung 4.1: Architekturdiagramm zu den Algorithmen (NEAT und HyperNEAT)

Das Architekturdiagramm (siehe 4.1) umfasst alle Klassen, die in direktem Zusammenhang mit den Algorithmen stehen. Klassen und Pakete, die der Bildbeladung, der Orchestrierung der Anwendung und ähnlichen Aufgaben dienen, sind in dieser Darstellung nicht enthalten. Für die gesamte Implementierung wurde ein nativer Ansatz verfolgt, was bedeutet, dass weitgehend auf die Einbindung externer Quellen verzichtet wurde. Eine Ausnahme bildet der Einsatz der Open-Source-Library *GraphStream* zur Visualisierung des Phänotyps im Kontext der Algorithmen. Die Verwendung dieser Bibliothek wurde gewählt, um die Komplexität der Visualisierungsaufgabe effizient zu bewältigen [13].

Eine zentrale Herausforderung bei der Implementierung von NEAT war die dynamisch variierende Topologie der neuronalen Netze zur Laufzeit. Diese Veränderungen erfordern eine kontinuierliche Nachverfolgung der Node- und Connection-Genes, um sicherzustellen, dass gleiche Veränderungen über alle Individuen der Population hinweg mit denselben Innovationsnummern versehen werden. Darüber hinaus muss der Algorithmus in der Lage sein, während der Mutationsphase sowohl neue Neuronen als auch neue Verbindungen zwischen den Neuronen zu erzeugen. Diese dynamischen Veränderungen stellen konventionelle Array-Strukturen aufgrund ihres statischen Aufbaus vor erhebliche Schwierigkeiten. Um diese topologischen Anpassungen effektiv zu handhaben, wurde eine speziell entwickelte, flexible Datenstruktur implementiert (siehe 4.2 ).

Aus dem Architekturdiagramm (siehe 4.1) wird ersichtlich, dass für die Implementierung von HyperNEAT lediglich eine zusätzliche Klasse erforderlich war, die für die Erstellung des Substrats verantwortlich ist. Um die Komplexität und den zeitlichen Rahmen dieser Arbeit nicht zu überschreiten, wurde das Substrat auf eine minimale Struktur beschränkt, was bedeutet, dass es keine versteckten Schichten enthält. Zudem enthält das ursprüngliche HyperNEAT-Paper keine spezifischen Handlungsempfehlungen zur Topologie des Substrats [16]. Aufgrund dieser Limitierung kann das neuronale Netz keine effektive Tiefe in seiner Topologie annehmen, welche für die Vorhersage von Bildklassifizierungsaufgaben meist nötig ist. Da der Schwerpunkt dieser Arbeit jedoch auf der Erklärbarkeit der Anwendung liegt, wurde dieser Aspekt als von untergeordneter Bedeutung angesehen.

## 4.1 Genotyp Implementierung

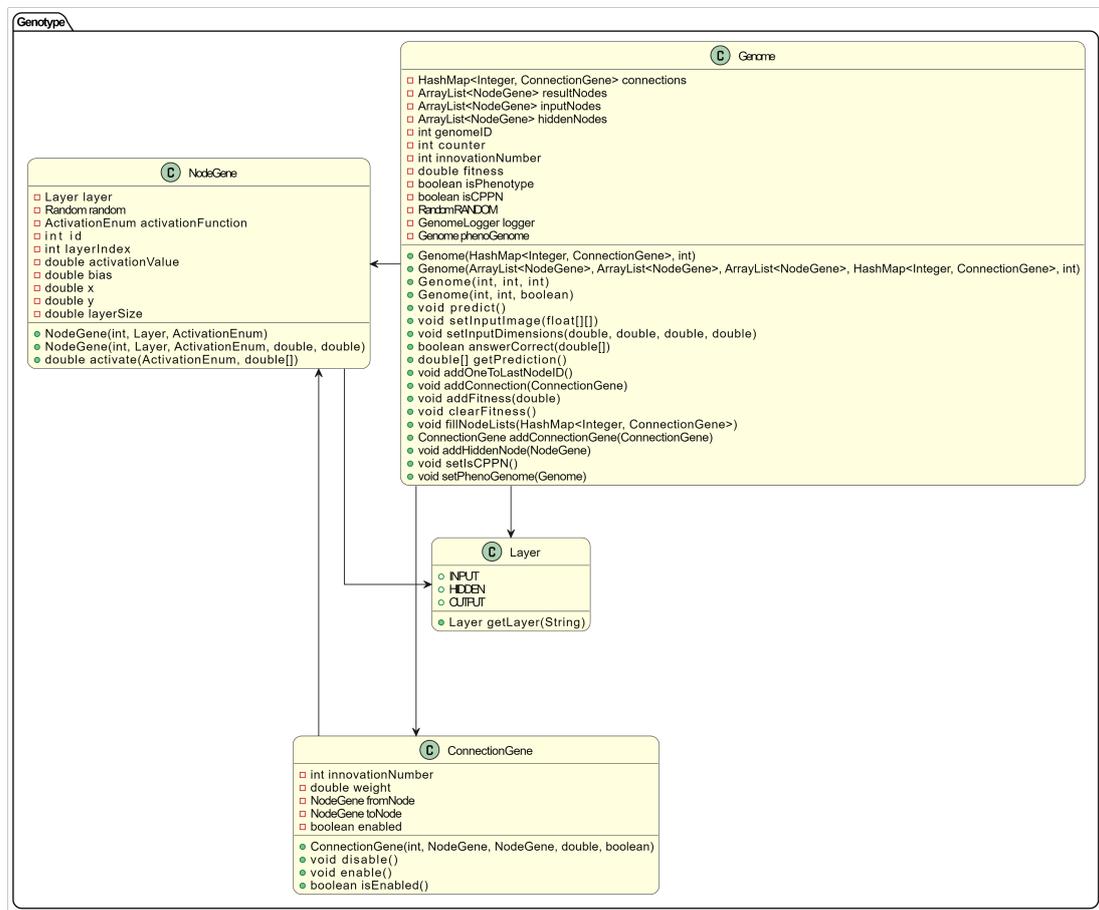


Abbildung 4.2: Architektur des Genotypen

Ein Genom besteht, wie in 3 beschrieben, aus einer Vielzahl von Node-Genes. Diese Node-Genes können, wie in 4.2 dargestellt, den Layern Input, Hidden oder Output zugeordnet werden. Die Verbindung zwischen den Node-Genes erfolgt über Connection-Genes. Zu Beginn sind alle Input-Nodes mit allen Output-Nodes verbunden.

Eine besondere Eigenschaft des Genoms ist seine flexible Verwendbarkeit. Es bildet nicht nur das Standardgenom ab, das im NEAT-Algorithmus verwendet wird, sondern auch das Compositional Pattern-Producing Network (CPPN) sowie das Substrat, welche für HyperNEAT benötigt werden.

## 4.2 Phänotyp Implementierung

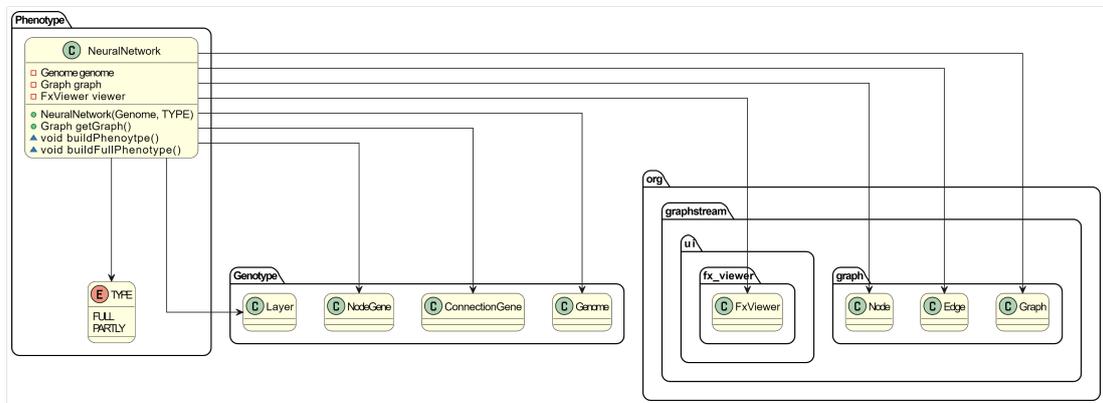


Abbildung 4.3: Architektur von Phänotypen

Der Phänotyp wird, wie zu Beginn des Kapitels erwähnt, durch ein Graphstream-Objekt dargestellt (siehe 4.3). Dabei werden alle Node-Genes und Connection-Genes des Genoms in entsprechende Node- und Edge-Objekte des Graphen aus Graphstream überführt. Um die visuelle Darstellung zu unterstützen, werden den Graphstream-Elementen visuelle Attribute hinzugefügt. Beispielsweise wird die Farbe der Nodes abhängig vom zugehörigen Layer festgelegt, und jedes Node-Gene erhält eine eindeutige ID, um eine klare Identifikation zu gewährleisten.

# 5 Entwicklung der XAI-Anwendung (Prototyp)

## 5.1 Vorgehen/Methodik

Die Gestaltung des Prototypen erfordert einen Ansatz, der sowohl die praktischen Anforderungen an die Anwendung als auch eine wissenschaftlich fundierte Grundlage zum Erstellen der Anwendung umfasst. An dieser Schnittstelle setzt die Design Science (DS) an, ein theoretischer Rahmen, der sich auf die Entwicklung und Evaluation von Artefakten konzentriert, um konkrete Probleme in der Praxis zu lösen. Einen systematischen Rahmen dafür liefert die von Ken Peffers et al. entworfene Methodologie: Design Science Research Methodology (DSRM), welche den Forschungsprozess in sechs aufeinanderfolgende Aktivitäten gliedert und so den Weg von der Problemidentifikation bis hin zur Kommunikation der Ergebnisse abdeckt. [12]

DSRM schafft somit einen evaluativen Forschungsansatz, welcher optimal zur Entwicklung des Prototypen der XAI-Anwendung zur Erklärbarkeit von NEAT und HyperNEAT geeignet ist. Im Folgenden werden die Prozessschritte zusammengefasst und anschließend das Vorgehen in der Anwendungsentwicklung anhand von DSRM beschrieben:

### 1. *Problemidentifikation und Motivation*

Die erste Aktivität soll dabei unterstützen, ein Problem zu spezifizieren. Aus meist rein oberflächlichen Beobachtungen, funktionalen oder auch nicht-funktionalen Anforderungen, soll somit eine starke Motivation gebildet werden, welche die Notwendigkeit einer Lösung hervorhebt. Die Motivation dient als erste Plausibilitätsüberprüfung. Somit kann gewährleistet werden, dass Anforderungen in einer Verhältnismäßigkeit zur Notwendigkeit der Problemlösung stehen.

Der Aufbau der nachfolgenden Artefakte basiert auf der Problemidentifikation, weshalb das Problem und die Motivation präzise formuliert sein müssen.

### 2. Definieren von Zielen

Nachdem das Problem klar identifiziert wurde, besteht der nächste Schritt darin, die Ziele für die zu entwickelnde Lösung festzulegen. Diese Ziele dienen als Leitlinien für die gesamte Entwicklung des Artefakts und helfen, den Forschungsprozess zu fokussieren. Die Ziele müssen realistisch sein und erfordern Kenntnisse bezüglich des Problemstandes und gegebenenfalls aktueller Lösungen sowie deren Wirksamkeit. Ziele dürfen qualitativer und quantitativer Natur sein. [12]

### 3. Design & Entwicklung

"Diese Aktivität umfasst das Festlegen der gewünschten Funktionalität des Artefakts und seiner Architektur und anschließend die tatsächliche Erstellung des Artefakts. Zu den erforderlichen Ressourcen gehört der Übergang von den Zielen zum Entwurf und zur Entwicklung." [12]

### 4. Demonstration

Nach der Entwicklung erfolgt die Demonstration des entwickelten Artefakts. Diese Demonstration muss nicht direkt alle Aspekte des Problems umfassen; sie kann bei großen Problemen iterativ für Teilprobleme angewandt werden. Sie kann in Form von Experimenten, Simulationen, Fallstudien, Beweisen oder anderen geeigneten Aktivitäten erfolgen. [12]

### 5. Evaluation

Bei der Evaluation gilt es zu überprüfen, wie gut das Artefakt zur Lösung des Problems beiträgt. Dafür werden geeignete wissenschaftliche Evaluationsmethoden benötigt. Bei einem zufriedenstellenden Ergebnis kann der Forscher in die nächste Aktivität *Kommunikation* übergehen. Sollte das Ergebnis nicht der Erwartung entsprechen, kann die Optimierung des Artefakts über die Aktivität *Design & Entwicklung* vorgenommen werden. Diese Aktivität ist somit iterativ und theoretisch in ihrer Anzahl der Durchläufe nicht limitiert. [12]

### 6. Kommunikation

Die Kommunikation dient dazu, die gewonnenen Erkenntnisse und das entwickelte Artefakt mit der wissenschaftlichen Gemeinschaft zu teilen. [12]

## 5.2 Problemidentifikation und Motivation

Das Problem ergibt sich aus dem Mangel an Erklärbarkeit für die Algorithmen NEAT und HyperNEAT, welcher in 1.1 erwähnt wurde. Beide Algorithmen modifizieren künstliche neuronale Netze, die inhärent als Black-Box-Modelle gelten. Aufgrund ihrer weiten Verbreitung existiert für künstliche neuronale Netze jedoch eine Vielzahl sowohl modellagnostischer als auch modell-spezifischer Methoden zur Erklärbarkeit. Im Gegensatz dazu kann NEAT als ein Nischenmodell betrachtet werden, für das noch spezifische Strategien zur Erklärbarkeit entwickelt werden müssen.

## 5.3 Definieren von Zielen

Um das Ziel präzise zu definieren, muss eine klare Verknüpfung zur Forschungsfrage hergestellt werden:

*"Kann eine speziell für die NEAT-Algorithmen entwickelte XAI-Anwendung die Verständlichkeit und das Wissen über diese Algorithmen für die Anwender verbessern?"*

**Ziel:** Aufgrund des begrenzten Umfangs einer Bachelorarbeit besteht das Ziel darin, den Prototypen einer XAI-Anwendung für NEAT und HyperNEAT zu entwickeln, welcher die Erklärbarkeitsziele von Chromik und Schuessler integriert (siehe 2.2). Diese Ausarbeitung soll als Basis für weiterführende Forschungs- und Entwicklungsarbeiten dienen.

**Randbedingung:** Die Algorithmen sollen zur Lösung eines Problems der Bildklassifizierung eingesetzt werden. Hierfür wurde der MNIST-Datensatz ausgewählt [8]. Dieser Datensatz bietet durch seine vergleichsweise kleinen Bilder den Vorteil einer reduzierten Komplexität, was die Berechnungen beschleunigt und die Umsetzung der Erklärbarkeit erleichtert.

## 5.4 Design & Entwicklung

Wie bereits definiert, umfasst das Ziel der XAI-Anwendung die Verbesserung der Erklärbarkeit von Algorithmen. Dies soll durch die gezielte Schaffung algorithmischer Transparenz erreicht werden. Im Rahmen der durchgeführten Recherche wurde festgestellt, dass für NEAT und HyperNEAT keine etablierten Modelle existieren, die eine ausreichende

algorithmische Transparenz gewährleisten. Daher ist es notwendig, auf native Ansätze zurückzugreifen, die speziell auf die Eigenheiten der beiden zu untersuchenden Algorithmen zugeschnitten sind. Die Entwicklung konzentriert sich dabei nicht auf die Erklärung einzelner Vorhersagen, sondern auf die Transparenz der internen Funktionsweise der Algorithmen. Als geeignete Strategien zur Erreichung dieses Ziels wurden folgende Ansätze identifiziert:

- Algorithm Tracing (Algorithmische Verfolgung) [7]
- Analyse und Dekonstruktion des Algorithmus
- Entwicklung spezifischer Erklärungsansätze
- Model Internals Visualizations (Modellinterne Visualisierungen)

Diese Ansätze ermöglichen es, die inneren Prozesse der Algorithmen detailliert zu durchleuchten und damit eine fundierte Basis für die Entwicklung von Erklärbarkeitsstrategien zu schaffen.

Für eine detaillierte Analyse ist es notwendig, die Algorithmen in ihre einzelnen Prozessschritte zu dekonstruieren. Der Aufbau des NEAT-Algorithmus bietet dabei eine natürliche Gliederung in verschiedene Prozesse, die eine systematische Erklärung ermöglichen (siehe Abbildung 5.1).

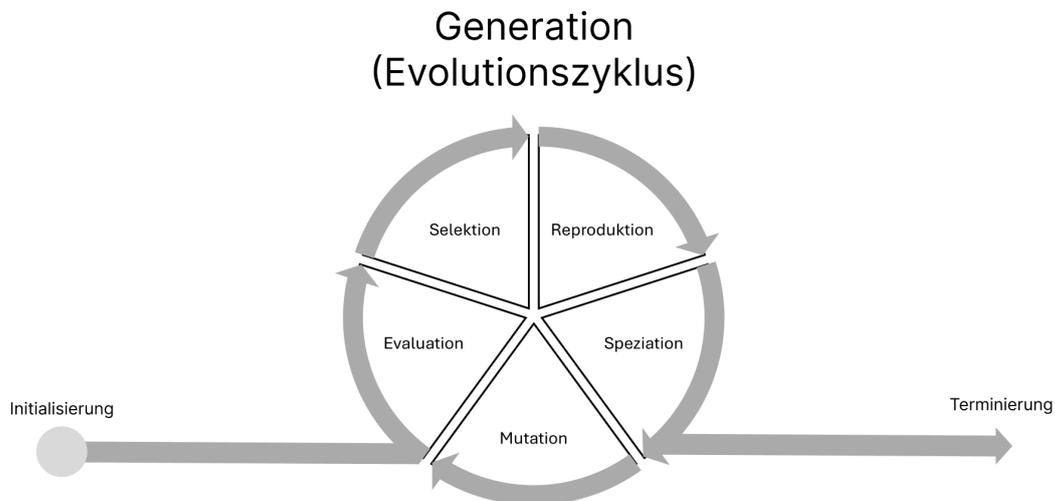


Abbildung 5.1: Aufteilung der Generationen in fünf Prozesse

**Design:** In dieser Arbeit wird die Erklärbarkeit der Algorithmen durch eine Simulation angestrebt, die alle identifizierten Prozessschritte integriert. Jeder dieser Prozessschritte wird durch eine prägnante und leicht verständliche Erklärung, sei es visuell oder textuell, begleitet. Im Mittelpunkt der Simulation steht die Generationsphase, die als zentraler Haltepunkt für das Algorithm Tracing dient.

**Entwicklung:** An der Fakultät *Technik und Informatik* der HAW Hamburg wird Java als grundlegende Programmiersprache in den Studiengängen *Angewandte Informatik*, *Wirtschaftsinformatik* und *Technische Informatik* verwendet. Im Hinblick auf die Aktivität *Kommunikation* fördert dies bei der Veröffentlichung des Quellcodes das Verständnis der Implementierung seitens der Anwender der HAW Hamburg. Technische Details der Implementierung können dem begleitenden Git-Repository entnommen werden [2]. Der gesamte Quellcode wurde unter Verwendung von Java und JavaFX entwickelt. Zur Sicherstellung einer klaren Trennung zwischen Benutzeroberfläche und Logikschicht wurde das Model-View-Controller (MVC)-Muster angewendet [4]. In den folgenden Abschnitten wird eine detaillierte Analyse des Designs der Benutzeroberfläche vorgenommen.

### 5.4.1 Startseite

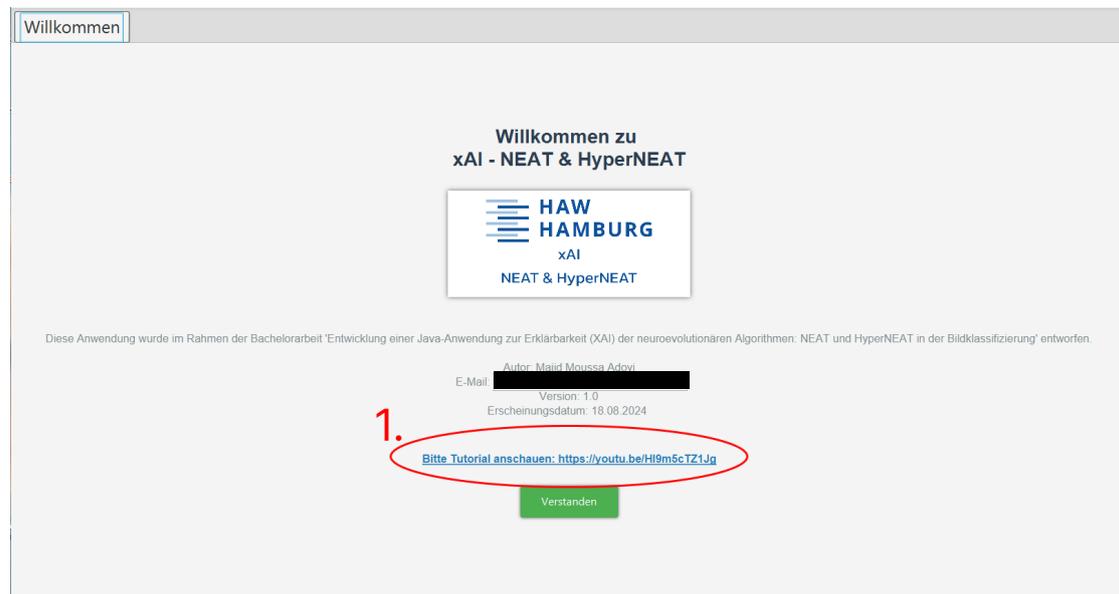


Abbildung 5.2: Bild der Startseite

Die Startseite bietet neben Angaben zum Veröffentlichungsdatum und Informationen zum Autor auch einen einfachen Einstieg in das Projekt durch ein Tutorial-Video (1). Dieses Video ist über einen Hyperlink eingebettet und liefert eine anschauliche Einführung in die ersten Schritte der Anwendung sowie weiterführende Hinweise zur Nutzung [1].

## 5.4.2 Registerkarten

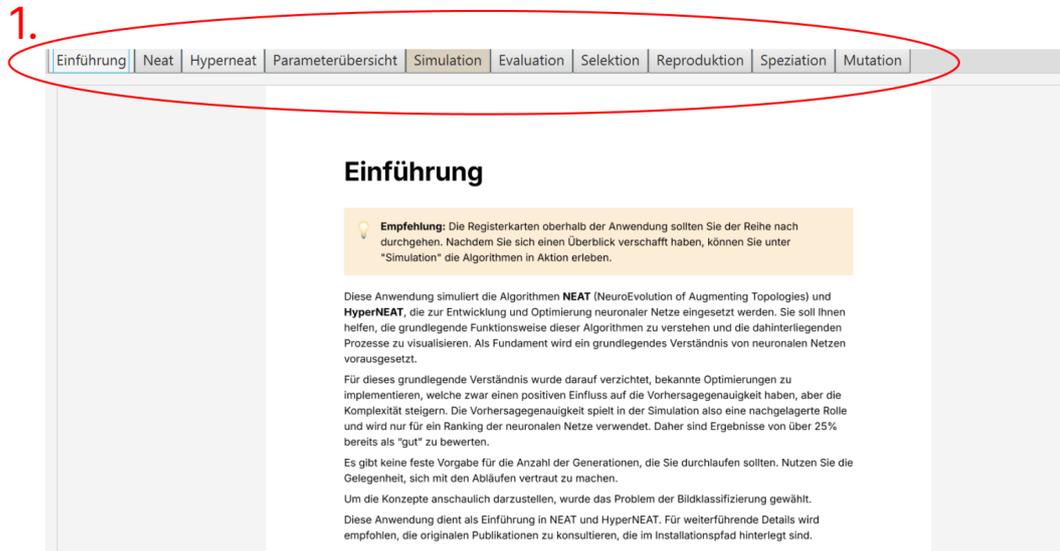


Abbildung 5.3: Aufteilung der Generationen in fünf Prozesse

Während der gesamten Ausführung der Anwendung sind am oberen Rand Registerkarten angeheftet, die als textuelle Erklärungen dienen und teilweise mit visuellen Darstellungen ergänzt werden. Die Registerkarten, die sich links von der Simulation (1.) befinden, bieten grundlegende Informationen und eine Einführung in die Thematik (siehe A.1, A.4, A.2, A.5). Die Registerkarten rechts der Simulation (1.) erläutern die in den Logs dokumentierten Schritte, auf welche im nächsten Unterabschnitt genauer eingegangen wird (siehe A.8, A.9, A.6, A.7, A.3).

### 5.4.3 Simulation

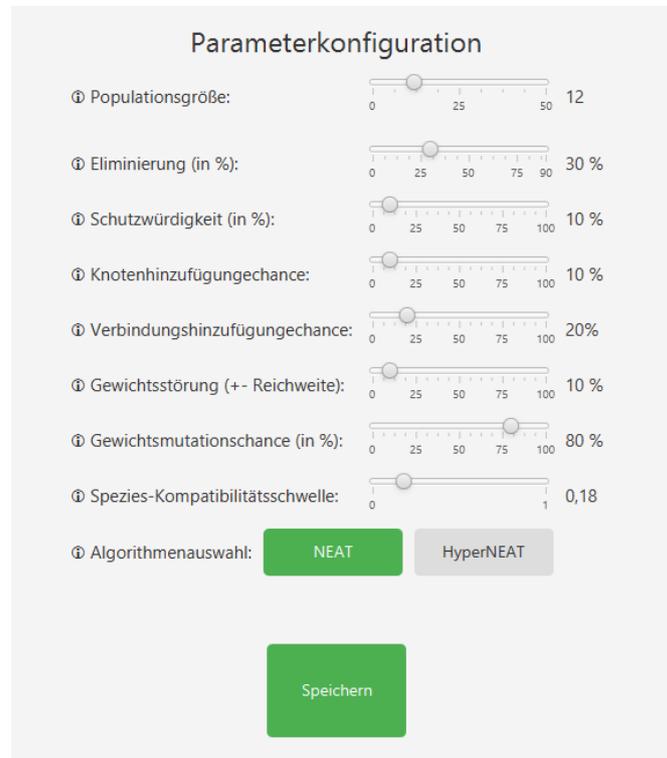


Abbildung 5.4: Übersicht der konfigurierbaren Hyperparameter

Die Parameterkonfiguration, wie in 5.4 dargestellt, bietet die Konfigurationsmöglichkeit für diverse Hyperparameter. Die *Populationsgröße* legt fest, wie viele Individuen in jeder Generation vorhanden sind. Der Parameter *Eliminierung* gibt den Prozentsatz der Individuen an, die nach jeder Generation aussortiert werden. Diese Auswahl basiert auf ihrer Fitnessbewertung und ermöglicht, dass nur die leistungsfähigeren Individuen in den nächsten Evolutionszyklus einfließen. *Schutzwürdigkeit* bezeichnet den Anteil der besten Individuen einer Spezies, die unverändert in die nächste Generation übernommen werden. Diese "geschützten" Individuen gewährleisten, dass wertvolle genetische Informationen erhalten bleiben und nicht durch zufällige Mutationen verloren gehen. Die *Knotenhinzufigeance* und *Verbindungshinzufigeance* bestimmen, wie wahrscheinlich es ist, dass in einem neuronalen Netz neue Knoten oder Verbindungen entstehen. Die *Gewichtsstörung* definiert die Spanne, in der die Verbindungsgewichte in einem Netz zufällig verändert werden können. Die *Gewichtsmutationschance* gibt die Wahrscheinlichkeit an, ob es zu *Gewichtsstörungen* kommen kann. Die *Spezies-Kompatibilitätsschwelle* legt fest, wie un-

## 5 Entwicklung der XAI-Anwendung (Prototyp)

terschiedlich zwei Netze sein müssen, um in verschiedene Spezies eingeteilt zu werden. Dieser Parameter ist entscheidend, um die genetische Vielfalt in der Population zu erhalten und zu verhindern, dass eine einzige Spezies dominiert. Zuletzt ermöglicht die Algorithmenwahl dem Benutzer eine Auswahl zwischen den NEAT- und HyperNEAT-Algorithmen, was den grundlegenden Ansatz der Netzwerkevolution bestimmt.

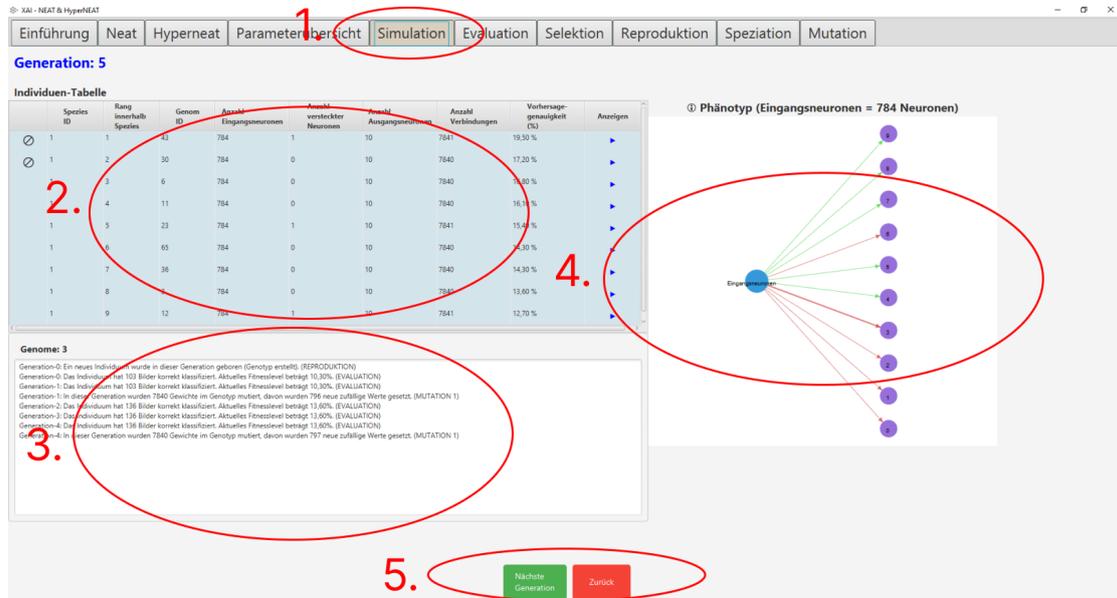


Abbildung 5.5: Übersicht der Simulation. In fünf Bereiche gegliedert.

Individuen-Tabelle									
Spezies ID	Rang innerhalb Spezies	Genom ID	Anzahl Eingangsgangneuronen	Anzahl versteckter Neuronen	Anzahl Ausgangsgangneuronen	Anzahl Verbindungen	Vorhersagegenauigkeit (%)	Anzeigen	
1	1	25	784	1	10	7841	19,50 %		
1	2	30	784	0	10	7840	17,20 %		
1	3	6	784	0	10	7840	17,80 %		
1	4	11	784	0	10	7840	16,19 %		
1	5	23	784	1	10	7841	15,44 %		
1	6	65	784	0	10	7840	14,50 %		
1	7	36	784	0	10	7840	14,30 %		
1	8	7	784	0	10	7840	13,90 %		
1	9	12	784	1	10	7841	12,70 %		

Abbildung 5.6: Tabelle mit allen Individuen der Population.

Die *Simulation* (1.) bildet das zentrale Element der XAI-Anwendung. Ziel ist es, den Anwendern durch *Algorithmic Tracing* die Möglichkeit zu bieten, nach jedem Simulationsschritt, der gleichzeitig einen Generationsschritt darstellt, die Veränderungen in der Ausführung des Algorithmus nachvollziehen zu können.

*Individuen-Tabelle* (2.): Die Individuen-Tabelle (siehe 5.5 und 5.6) stellt eine umfassende Übersicht über alle Individuen bereit, die den Generationsschritt überstanden haben. In der ersten Spalte der Tabelle werden besondere Individuen durch spezifische Symbole

hervorgehoben. Ein Totenkopf-Symbol markiert Individuen, die aufgrund eines unzureichenden Fitness-Wertes aus der Population ausgeschlossen werden und somit nicht in die nächste Generation übergehen. Das Symbol eines durchgestrichenen Kreises hingegen kennzeichnet Individuen, die aufgrund ihres hohen Fitness-Wertes von Mutationen geschützt sind. Die Spalte *Spezies-ID* ordnet jedem Individuum eine eindeutige Spezies zu, wobei jede Spezies-ID eine charakteristische Hintergrundfarbe erhält, um die visuelle Differenzierung zu erleichtern. Die *Genome-ID* identifiziert jedes Individuum innerhalb der Population eindeutig.

Die folgenden vier Spalten, welche die *Anzahl der Eingangsneuronen*, der *versteckten Neuronen* und der *Ausgangsneuronen* sowie die *Anzahl der Verbindungen* spezifizieren, bieten eine detaillierte Übersicht über die Netzwerkstruktur der Individuen. Die Spalte *Vorhersagegenauigkeit (%)* gibt den evaluierten Fitness-Wert als Prozentsatz an, welcher auf dem Anteil der korrekt klassifizierten Bilder im Verhältnis zur Gesamtanzahl basiert. Schließlich bietet die letzte Spalte, *Anzeige*, eine interaktive Funktionalität: Durch das Anklicken eines Elements wird der Phänotyp des ausgewählten Individuums dargestellt. Bei der Simulation von HyperNEAT wird zusätzlich der Phänotyp des Compositional Pattern Producing Network (CPPN) angezeigt (siehe 5.7).

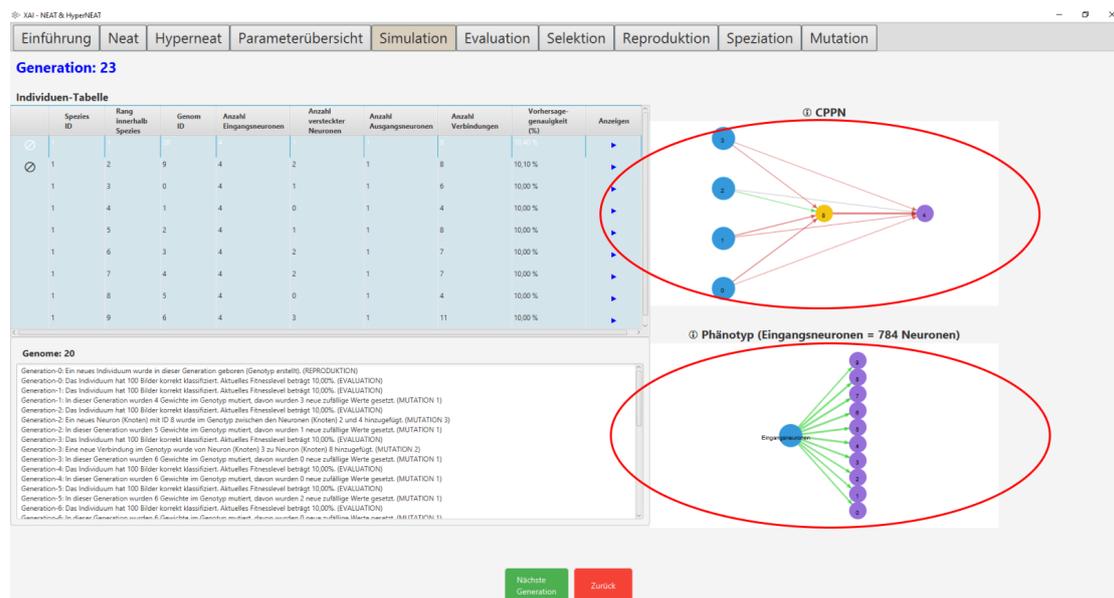


Abbildung 5.7: Anzeige der Netzwerkstruktur des CPPNs und des verbundenen Substarts.

*Logs (3.):* In jeder Generation werden für jedes Individuum Veränderungen am Genom protokolliert. Diese Protokolle erfassen sämtliche in 5.1 dargestellten Prozesse. Für bestimmte Prozesse, wie beispielsweise die Mutation, existieren verschiedene Typen, die in den Protokollen mit Bezeichnungen wie "MUTATION 1" gekennzeichnet sind. Anhand dieser Kennzeichnung lässt sich unter der Registerkarte *Mutation* detailliert nachvollziehen, welche spezifischen Veränderungen durch die Mutation bewirkt wurden.

*Modellinterne Visualisierungen (4.):* Die Visualisierung der neuronalen Netze (Phänotypen) unterscheidet sich in der Simulation von NEAT und HyperNEAT. Während bei NEAT ausschließlich der Phänotyp dargestellt wird, visualisiert HyperNEAT sowohl das durch Verbindungen definierte Substrat als auch den eigentlichen Phänotyp - das CPPN.

Die Darstellung des verbundenen Substrats und des Phänotyps umfasst dabei lediglich einen Eingangsknoten. Dies liegt daran, dass die Gesamtdarstellung aller 784 Eingangsknoten die Visualisierung unübersichtlich gemacht hätte. Stattdessen werden die 784 Eingangsknoten durch einen einzelnen Knoten repräsentiert. Die Stärke der Verbindungsgewichte wird durch unterschiedliche Pfeildicken visualisiert, wobei negative Gewichte in rot, positive in grün und deaktivierte Verbindungen in grau dargestellt werden.

*Knöpfe (5.):* Die grüne Schaltfläche mit der Aufschrift *Nächste Generation* startet die Simulation bei jeder Aktivierung ausgehend von dem aktuellen Haltepunkt. Die Simulation pausiert erneut, wenn der in 5.1 gezeigte Generationsschritt durchlaufen wurde und die Simulation sich wieder am Anfang der neuen Generation befindet. Die rote Schaltfläche mit der Aufschrift *Zurück* setzt die Simulation zurück und ermöglicht das Einstellen einer neuen Parameterkonfiguration für einen weiteren Durchlauf.

## 5.5 Demonstration

Im Rahmen der Demonstration galt es zu evaluieren, inwiefern der entwickelte Prototyp zum Wissensaufbau bezüglich der beiden Algorithmen NEAT und HyperNEAT dient. Um dies herauszufinden, wurde ein quantitatives Forschungsdesign gewählt, bei dem Studienteilnehmer die entwickelte XAI-Anwendung testen und diverse Fragen beantworten sollten. Die Fragen haben sich überwiegend auf die von Chromik und Schuessler definierten Erklärbarkeitsziele gestützt.

Der Fragebogen wurde mit Hilfe der Software *Google Forms* erstellt. Die geschlossenen Eingangsfragen zielen auf die nähere Beschreibung der Probanden ab. Die Antworten auf die anwendungsspezifischen Fragen wurden mit Likert-Skalen erfasst, um ein konkretes Feedback zu erhalten. Da es sich um die Entwicklung eines Prototyps handelt, wurde die abschließende Frage bewusst offen gestaltet, um qualitative Rückmeldungen zu ermöglichen und so wertvolle Meinungen in den weiteren Entwicklungsprozess einzubinden.

Den Probanden wurde ein Link per *Google Drive* zur Verfügung gestellt. So konnten sie einen Ordner herunterladen, der die Anwendung als ausführbare .exe-Datei sowie den Link zu der webbasierten Umfrage enthielt. Die Probanden wurden aufgefordert, die bereitgestellten Materialien eigenständig zu erkunden. Abgesehen von einer kurzen Ausführungsreihenfolge wurden keine weiteren Handlungsempfehlungen zur Nutzung der Anwendung gegeben.

Bei der Auswahl der Personen, welche die Anwendung testen sollten, handelt es sich um eine gezielte Stichprobe: Die Probanden verfügen über grundlegende Kenntnisse und Interesse im Bereich der künstlichen Intelligenz. Insgesamt haben 12 Personen den Prototypen getestet und bewertet, wodurch wertvolle Erkenntnisse für die Weiterentwicklung sowie für das definierte Forschungsziel gewonnen werden konnten.

## 5.6 Evaluation

### 5.6.1 Umfragedesign

Hier folgend werden die Fragen der Umfrage nach den Erklärbarkeitszielen von Chromik und Schuessler eingeordnet:

#### **Transparenz**

- Die Erklärungen der XAI-Anwendung waren durchgängig nachvollziehbar.
- Ich konnte jede Erklärung in der Simulation klar identifizieren.
- Die Darstellung der Erklärungen half mir, die Entscheidungsprozesse der Algorithmen besser zu verstehen.

#### **Prüfbarkeit**

- Die Ergebnisse der XAI-Anwendung waren für mich nachvollziehbar und überprüfbar.

### **Vertrauen**

- Ich habe zu keinem Zeitpunkt während der Simulation an der Vorhersagegenauigkeit gezweifelt.
- Ich bin davon überzeugt, dass ich durch die Nutzung der XAI-Anwendung ein gutes Verständnis von NEAT entwickelt habe.
- Ich bin davon überzeugt, dass ich durch die Nutzung der XAI-Anwendung ein gutes Verständnis von HyperNEAT entwickelt habe.

### **Überzeugungskraft**

- Die Anwendung hat mich davon überzeugt, dass die Algorithmen in der Praxis nützlich und zuverlässig sind.

### **Zufriedenheit**

- Die Erklärungen waren ausgewogen, indem sie Text, Bilder, Videos und Simulationen angemessen kombinierten.
- Ich würde bei zukünftigen Projekten mit NEAT oder HyperNEAT auf die XAI-Anwendung zurückgreifen.
- Ich bin insgesamt zufrieden mit den bereitgestellten Erklärungen.
- Ich würde die XAI-Anwendung weiterempfehlen.

### **Effektivität**

- Die Möglichkeit, die Simulation mehrfach durchzuführen, erleichterte mir die Überprüfung der Ergebnisse.
- Die Möglichkeit, Parameter der Simulation anzupassen, hat mir bei der Überprüfung der Algorithmen geholfen.
- Die XAI-Anwendung hat mein Wissen über evolutionäre Algorithmen erweitert.

- Die XAI-Anwendung hat mir geholfen, schneller ein tieferes Verständnis der Algorithmen zu entwickeln.
- Der Umfang der bereitgestellten Informationen war genau richtig, um ein umfassendes Verständnis zu ermöglichen.

### **Effizienz**

- Die XAI-Anwendung ist strukturiert aufgebaut.
- Die XAI-Anwendung ist intuitiv zu bedienen.
- Die XAI-Anwendung lief während der gesamten Nutzung stabil und ohne Abstürze.
- Die Ladezeiten und Reaktionszeiten der XAI-Anwendung waren angemessen.
- Die Parameterkonfiguration war übersichtlich und leicht einzustellen.
- Das Design der XAI-Anwendung entspricht meinen Erwartungen an eine solche Anwendung.

### **Bildung**

- Nach der Nutzung der XAI-Anwendung fühle ich mich in der Lage, die Grundzüge von NEAT zu erklären.
- Nach der Nutzung der XAI-Anwendung fühle ich mich in der Lage, die Grundzüge von HyperNEAT zu erklären.

### **Debugging**

- Ich fühle mich dazu befähigt, Fehler in der Simulation zu identifizieren.

### **Sonstige**

- Hast Du bereits Vorkenntnisse zur Thematik Neuronale Netze erlangt?
- Hast Du bereits vom NEAT Algorithmus gehört?
- Hast Du bereits vom HyperNEAT Algorithmus gehört?
- Ich sehe weiteres Entwicklungspotential in der Anwendung, um die Algorithmen noch besser verständlich zu machen.

- Die Anwendung bot ausreichend Flexibilität in der Konfiguration und Nutzung.
- Das Problem der Bildklassifizierung war ein leicht verständliches Problem.
- Ich würde grundsätzlich eher auf eine XAI-Anwendung zurückgreifen, statt mich mit wissenschaftlicher Lektüre auseinanderzusetzen.
- Es traten keine technischen Probleme auf, welche meine Arbeit behinderten.

### 5.6.2 Analyse der Umfrageergebnisse

Den auf die Erklärbarkeitsziele gerichteten Fragen gingen drei Fragen zu den Vorkenntnissen der Teilnehmer voraus. Dabei wurde erfragt, ob die Algorithmen bereits bekannt sind, was bei NEAT 33,3 % und bei HyperNEAT 25 % der Befragten bejahten. Die überwiegende Mehrheit von 75 % verfügte über Vorkenntnisse zu neuronalen Netzwerken. Eine Analyse der Antworten ergab, dass außer einem Ausreißer kein signifikanter Unterschied in den Bewertungen zwischen Nutzern mit und ohne Vorkenntnisse festzustellen war. In einem folgenden Iterationsdurchlauf der Aktivitäten Design & Entwicklung, Demonstration und Evaluation sollte der Kenntnisstand der Nutzer detaillierter erfasst werden, um gezielte, auf die jeweilige Zielgruppe abgestimmte Verbesserungen implementieren zu können.

Die Skala des Fragebogens diente dazu, die Zustimmung der Probanden zu den gestellten Aussagen zu quantifizieren und wurde so angeordnet, dass 1 für vollständige Zustimmung und 5 für völlige Ablehnung stand. Der Fragebogen inklusive der Ergebnisse befindet sich im Anhang A.10.

Die folgenden Abschnitte fassen die Umfrageergebnisse basierend auf den Erklärbarkeitszielen zusammen:

**Transparenz** Die Transparenz der XAI-Anwendung wurde insgesamt positiv bewertet. Die Mehrheit der Teilnehmer fand die Erklärungen nachvollziehbar und verständlich. Diese positive Ansicht hatten zusammengefasst 95,8 % (zustimmend und stark zustimmend) der Befragten. Allerdings gab es einige Teilnehmer (41,6 %), die neutral auf die Frage der klaren Identifikation jeder Erklärung in der Simulation antworteten. Dies deutet darauf hin, dass zwar eine hohe Transparenz erreicht wurde, aber noch Potenzial für Verbesserungen in der Klarheit und Strukturierung der Erklärungen besteht.

**Prüfbarkeit** Die Prüfbarkeit der Ergebnisse wurde von den Teilnehmern weitgehend als positiv bewertet. 33,3 % der Befragten stimmten stark zu, dass die Ergebnisse der XAI-Anwendung nachvollziehbar und überprüfbar waren, während 50 % zustimmten. Dies zeigt, dass die Teilnehmer das System als verlässlich und überprüfbar empfanden, was ein zentrales Ziel für eine erklärbare KI ist. Dennoch könnte eine weitere Verbesserung der Zugänglichkeit und Überprüfbarkeit der Ergebnisse die Benutzererfahrung weiter stärken.

**Vertrauen** Das Vertrauen in die XAI-Anwendung und ihre Ergebnisse war insgesamt hoch. Die Hälfte der Teilnehmer (50 %) äußerte während der Simulation keinerlei Zweifel an der Vorhersagegenauigkeit der Algorithmen. Darüber hinaus gaben 66,7 % der Befragten an, durch die Nutzung der XAI-Anwendung ein gutes Verständnis sowohl von NEAT als auch von HyperNEAT entwickelt zu haben. Allerdings bleibt festzuhalten, dass eine signifikante Anzahl der Teilnehmer nicht stark zustimmend geantwortet hat, was darauf hindeutet, dass noch gewisse Aspekte fehlen, um das Vertrauen vollständig zu festigen. Dies unterstreicht, dass die Anwendung zwar erfolgreich das Vertrauen der Nutzer in die Funktionsweise und Genauigkeit der Algorithmen gestärkt hat, jedoch weiteres Potenzial besteht, um dieses Vertrauen noch weiter zu erhöhen.

**Überzeugungskraft** Die Überzeugungskraft der XAI-Anwendung bezüglich der Nützlichkeit und Zuverlässigkeit der Algorithmen in der Praxis wurde von den Teilnehmern ebenfalls als hoch eingeschätzt. 50 % der Teilnehmer stimmten stark zu, dass die Anwendung sie davon überzeugt hat, dass die Algorithmen sowohl praktisch nutzbar als auch zuverlässig sind. Dies verdeutlicht, dass die Anwendung nicht nur theoretisches Wissen vermittelt, sondern auch das Vertrauen in die Anwendbarkeit der Algorithmen in realen Szenarien stärkt. Darüber hinaus lässt sich daraus ableiten, dass die Wahl des Bildklassifizierungsproblems als Anwendungsfall gut getroffen war, da es ein für die Teilnehmer nachvollziehbares und praxisnahes Problem darstellt.

**Zufriedenheit** Die Auswertung zeigt, dass die XAI-Anwendung die Erwartungen der meisten Nutzer erfüllt oder sogar übertroffen hat. Es besteht jedoch weiterhin Raum für Verbesserungen, um die Zufriedenheit noch weiter zu steigern, insbesondere hinsichtlich der Struktur und Präsentation der Informationen.

**Effektivität** Die Effektivität der XAI-Anwendung in Bezug auf das Erreichen von Lernzielen und die Unterstützung bei der Überprüfung von Algorithmen wurde durchweg positiv bewertet. 50 % der Teilnehmer stimmten stark zu, dass die Möglichkeit, die Simulation mehrfach durchzuführen, die Überprüfung der Ergebnisse erleichterte. Zudem

gaben 58,3 % an, dass die Anwendung ihr Wissen über evolutionäre Algorithmen erweitert hat, was auf eine erfolgreiche Vermittlung komplexer Konzepte hinweist. Dies bestätigt die Nützlichkeit der Anwendung als Bildungsinstrument.

**Effizienz** Die Effizienz der Anwendung wurde ebenfalls positiv wahrgenommen. 58,3 % der Teilnehmer stimmten stark zu, dass die XAI-Anwendung strukturiert aufgebaut ist, und 41,7 % empfanden die Bedienung als intuitiv. Die Ergebnisse zeigen, dass die Benutzerführung und die Struktur der Anwendung den Erwartungen der Nutzer weitgehend entsprechen. Dennoch könnten geringfügige Verbesserungen, insbesondere in der Bedienfreundlichkeit, die Effizienz der Anwendung weiter optimieren.

**Bildung** Im Hinblick auf das Bildungsziel zeigte die Umfrage, dass die XAI-Anwendung erfolgreich dazu beigetragen hat, das Verständnis der Nutzer für die Algorithmen NEAT und HyperNEAT zu verbessern. 50 % der Teilnehmer fühlten sich nach der Nutzung der Anwendung in der Lage, die Grundzüge dieser Algorithmen zu erklären, was die Wirksamkeit der Anwendung als Lehrmittel unterstreicht.

**Debugging** Die Befähigung, Fehler in der Simulation zu identifizieren, wurde von den Teilnehmern überwiegend zurückhaltend bewertet. Nur ein Teilnehmer stimmte dieser Aussage stark zu, während die Mehrheit der anderen Teilnehmer neutral oder nicht zustimmend reagierte. Dies deutet darauf hin, dass die derzeitige Struktur der Anwendung in Bezug auf Debugging und Algorithm Tracing noch nicht ausreichend ist, um den Nutzern die notwendige Unterstützung zu bieten, um die Entstehung der Ergebnisse effektiv zu bewerten. Eine tiefere und detailliertere Debugging- oder Algorithm Tracing-Struktur könnte hier hilfreich sein, um den Nutzern mehr Einblicke in die inneren Abläufe der Algorithmen zu geben und somit die Fähigkeit zur Fehleridentifikation zu verbessern.

**Zusammenfassung der offenen Frage** Die Verbesserungsvorschläge der Teilnehmer konzentrierten sich hauptsächlich auf die Benutzeroberfläche, die als rudimentär, aber verständlich beschrieben wurde. Es wurde angeregt, das Design zu überarbeiten, um die Übersichtlichkeit zu erhöhen und die Lesbarkeit der Parameter in der Simulation zu verbessern. Zudem wurde der Wunsch nach mehr interaktiven Elementen, wie einem Quiz oder visuellen Beispielen, geäußert, um das Verständnis für die Algorithmen zu vertiefen. Die Teilnehmer schlugen außerdem vor, ein Glossar sowie detailliertere Erklärungen zu den festen Parametern und Schwellenwerten in die Anwendung zu integrieren. Technische Verbesserungen, wie die Möglichkeit, mehrere Generationen gleichzeitig durchlaufen zu lassen, wurden ebenfalls empfohlen. Insgesamt zeigen die Rückmeldungen, dass die Anwendung als nützlich empfunden wird, aber in mehreren Bereichen noch Potenzial für

Verbesserungen besteht. Dieser Eindruck wird durch die Antworten auf die Frage: *Ich sehe weiteres Entwicklungspotential in der Anwendung, um die Algorithmen noch besser verständlich zu machen*, weiter untermauert. Hier gaben 66,7 % der Befragten an, dass sie Potenzial für weitere Entwicklungen und Verbesserungen in der Anwendung erkennen, um die Verständlichkeit der Algorithmen zu erhöhen.

### 5.7 Kommunikation

Die Ergebnisse dieser Bachelorarbeit werden im Rahmen der Abgabe des Thesis-Dokuments veröffentlicht. Für Praktiker und Entwickler stellt die Veröffentlichung des Prototyps über ein öffentlich zugängliches Git-Repository (siehe [2]) eine wichtige Ressource dar. Der Quellcode der Anwendung ist in Java implementiert und nach dem Model-View-Controller (MVC)-Muster strukturiert, was eine klare Trennung zwischen Benutzeroberfläche und Logikschicht ermöglicht. Durch die Bereitstellung des Quellcodes im öffentlichen Repository können Anwender die Anwendung selbstständig nutzen, anpassen und weiterentwickeln. Dieses Repository wird für jedermann zugänglich sein, um die Nachnutzung und den offenen Austausch innerhalb der Entwickler-Community zu fördern.

Zusätzlich zur Abgabe der Thesis wird eine Kopie des Thesis-Dokuments und des begleitenden Quellcodes an die Bibliothek der HAW Hamburg übergeben, um sicherzustellen, dass die Ergebnisse auch langfristig verfügbar bleiben und zukünftigen Studierenden als Referenz dienen können.

## 6 Zusammenfassung / Fazit

Unter Verwendung des Design Science Research Methodology (DSRM) wurde ein strukturierter Rahmen für die Entwicklung, Demonstration und Evaluation des Prototyps geschaffen, der speziell für die Erklärbarkeit der Algorithmen NEAT und HyperNEAT konzipiert wurde [12]. Die Evaluation der Umfrage führte zu dem klaren Ergebnis, dass die entwickelte XAI-Anwendung das Verständnis dieser Algorithmen erheblich stärkt. Durch den Einbezug der Erklärbarkeitsziele nach Chromik und Schuessler konnten zudem wichtige Eigenschaften der Anwendung getestet werden. Dies ermöglichte wertvolle Rückschlüsse, beispielsweise in den Bereichen der Benutzerfreundlichkeit, Transparenz und Effektivität. Diese Erkenntnisse unterstreichen nicht nur die Stärken der Anwendung in der Wissensvermittlung, sondern auch ihr Potenzial, das Vertrauen der Nutzer in die praktische Anwendbarkeit der Algorithmen zu fördern.

Gleichzeitig wurden spezifische Bereiche identifiziert, in denen weitere Optimierungen sinnvoll wären. Dazu zählen insbesondere die Benutzeroberfläche und zusätzliche interaktive Elemente, die das Lern- und Nutzungserlebnis weiter verbessern könnten. In diesem Zusammenhang könnte es lohnenswert sein, sich intensiver mit Aspekten des UX-Designs auseinanderzusetzen, um die Anwendung noch benutzerfreundlicher zu gestalten.

Ein weiterer zentraler Aspekt war die Entwicklung der Algorithmen. Durch die native Implementierung und den Verzicht auf externe Bibliotheken konnten die zugrundeliegenden neuronalen Netze optimal verwaltet und modifiziert werden, was zu einer höheren Anpassungsfähigkeit und Kontrolle führte. Dieser Ansatz brachte jedoch auch einen deutlich höheren Entwicklungsaufwand mit sich, was zu erheblichen Zeitverzögerungen führte. Diese Entscheidung zeigt einerseits die Vorteile einer maßgeschneiderten Lösung, verdeutlicht aber auch die Notwendigkeit, den Aufwand und die verfügbaren Ressourcen in zukünftigen Projekten sorgfältig abzuwägen.

Für künftige Forschungsarbeiten wäre es sinnvoll, die Komplexität der Algorithmen zu erhöhen, um den Anwendern ein noch umfassenderes Bild ihrer Funktionsweise zu ver-

mitteln. Derzeit bestehen Einschränkungen in der Anzahl der verwendeten Parameter, und HyperNEAT baut mit seinem Substrat keine tiefen Netzstrukturen auf 4.

Darüber hinaus hat sich das gewählte Problem der Bildklassifizierung im gegebenen Rahmen als sehr geeignet erwiesen, was auch durch das Feedback der Anwender bestätigt wurde. Dennoch wäre es vorteilhaft, die Anwendung auf eine breitere Palette von Problemen auszudehnen, um die Vielseitigkeit und Stärke der Algorithmen in unterschiedlichen Kontexten zu demonstrieren.

Abschließend lässt sich festhalten, dass der entwickelte Prototyp eine solide Grundlage für weiterführende Forschungs- und Entwicklungsarbeiten im Bereich der Erklärbarkeit von NEAT und HyperNEAT bildet. Die gewonnenen Erkenntnisse bieten klare Ansätze für zukünftige Verbesserungen und zeigen das Potenzial der Anwendung, einen wichtigen Beitrag zur Verständlichkeit und Transparenz komplexer Algorithmen zu leisten.

# Literaturverzeichnis

- [1] ADOYI, Majid M.: *Tutorial XAI NEAT & HyperNEAT*. YouTube Video. 2024. – URL <https://www.youtube.com/watch?v=H19m5cTZ1Jg>. – Veröffentlicht am: 18. August 2024, Zugriff am: 24. August 2024
- [2] ADOYI, Majid M.: *xai-neat-hyperneat*. 2024. – URL <https://github.com/mousmaj/xai-neat-hyperneat>. – Zugriff am 27. August 2024
- [3] CHROMIK, Michael ; SCHUESSLER, Martin: A Taxonomy for Human Subject Evaluation of Black-BoxExplanations in XAI. (2020)
- [4] GEEKSFORGEEKS: *MVC Design Pattern*. 2024. – URL <https://www.geeksforgeeks.org/mvc-design-pattern/>. – Zugriff am 27. August 2024
- [5] KRAUS, Tom ; GANSCHOW, Lene ; EISENTRÄGER, Marlene ; WISCHMANN, Steffen: ERKLÄRBARE KI - Anforderungen, Anwendungsfälle und Lösungen. (2021)
- [6] KURZWEIL, Ray: *The Age of Intelligent Machines*. Cambridge, MA : MIT Press, 1990
- [7] LABS, IB: *Algorithm Tracing*. <https://medium.com/@IBLabs/algorithm-tracing-6c72a4b00ea3>. 2020. – Accessed: 2024-08-21
- [8] LECUN, Yann ; CORTES, Corinna ; BURGESS, Christopher J.: *The MNIST Database of Handwritten Digits*. <https://yann.lecun.com/exdb/mnist/>. 1998. – Accessed: 2024-08-22
- [9] LOWELL, Jessica ; BIRGER, Kir ; GRABKOVSKY, Sergey: Comparison of NEAT and HyperNEAT on a Strategic Decision-Making Problem. (2010)
- [10] MITTELSTADT, Brent ; RUSSELL, Chris ; WACHTER, Sandra: Explaining Explanations in AI. (2019)
- [11] MOLNAR, Christoph: *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*. 2022

- [12] PEFFERS, Ken ; TUUNANEN, Tuure ; ROTHENBERGER, Marcus A.: A Design Science Research Methodology for Information Systems Research. (2007)
- [13] PROJECT, GraphStream: *GraphStream: A Dynamic Graph Library*. <http://graphstream-project.org/>. 2023. – Zugriff am 28. August 2024
- [14] RUSSELL, Stuart ; NORVIG, Peter: *Künstliche Intelligenz: Ein moderner Ansatz*. 3. Pearson Deutschland GmbH, 2012. – ISBN 978-3-86894-098-5
- [15] SAMEK, Wojciech (Hrsg.) ; MONTAVON, Grégoire (Hrsg.) ; VEDALDI, Andrea (Hrsg.) ; HANSEN, Lars K. (Hrsg.) ; MÜLLER, Klaus-Robert (Hrsg.): *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer Nature, 2019. – ISBN 978-3-030-28953-9
- [16] STANLEY, Kenneth O. ; D’AMBROSIO, David B. ; GAUCI, Jason: A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks. (2009)
- [17] STANLEY, Kenneth O. ; MIIKKULAINEN, Risto: Evolving Neural Networks through Augmenting Topologies. (2002)
- [18] STANLEY, Kenneth O. ; RISI, Sebastian: A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks. (2012)
- [19] STENSBYE ; RØYSET, Andreas: Evolving Multi-Spatial-Substrates to extend HyperNEAT in order to increase Functional Modularity. (2019)
- [20] TENSTAD ; AMUND: Deep Evolvable-Substrate HyperNEAT - Extending ES-HyperNEAT with Multiple Substrates in an Evolving Topology. (2020)

# A Anhang

## A.1 Einführung

# Einführung



**Empfehlung:** Die Registerkarten oberhalb der Anwendung sollten Sie der Reihe nach durchgehen. Nachdem Sie sich einen Überblick verschafft haben, können Sie unter "Simulation" die Algorithmen in Aktion erleben.

Diese Anwendung simuliert die Algorithmen **NEAT** (NeuroEvolution of Augmenting Topologies) und **HyperNEAT** (Hypercube-based NeuroEvolution of Augmenting Topologies), die zur Entwicklung und Optimierung neuronaler Netze eingesetzt werden. Sie soll Ihnen helfen, die grundlegende Funktionsweise dieser Algorithmen zu verstehen und die dahinterliegenden Prozesse zu visualisieren. Als Fundament wird ein grundlegendes Verständnis von neuronalen Netzen vorausgesetzt.

Für dieses grundlegende Verständnis wurde darauf verzichtet, bekannte Optimierungen zu implementieren, welche zwar einen positiven Einfluss auf die Vorhersagegenauigkeit haben, aber die Komplexität steigern. Die Vorhersagegenauigkeit spielt in der Simulation also eine nachgelagerte Rolle und wird nur für ein Ranking der neuronalen Netze verwendet. Daher sind Ergebnisse von über 25% bereits als "gut" zu bewerten.

Es gibt keine feste Vorgabe für die Anzahl der Generationen, die Sie durchlaufen sollten. Nutzen Sie die Gelegenheit, sich mit den Abläufen vertraut zu machen.

Um die Konzepte anschaulich darzustellen, wurde das Problem der Bildklassifizierung gewählt.

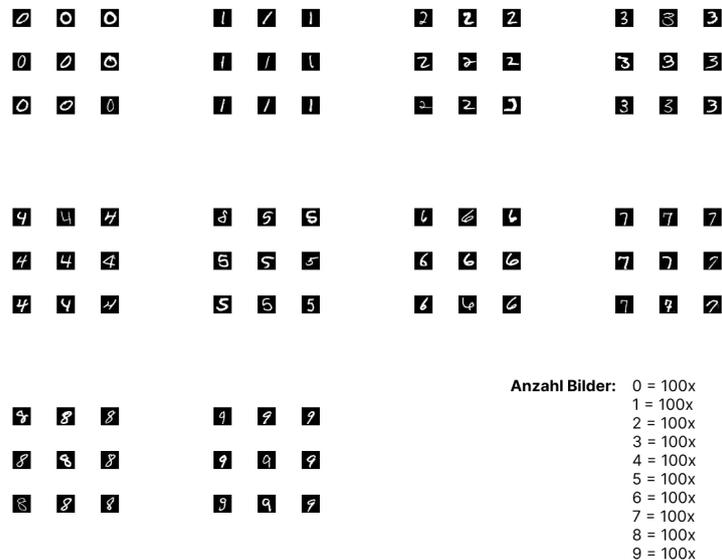
Diese Anwendung dient als Einführung in NEAT und HyperNEAT. Für weiterführende Details wird empfohlen, die originalen Publikationen zu konsultieren, die im Installationspfad hinterlegt sind.

## Das "Problem" der Bildklassifizierung

Als Menschen fällt es uns gekonnt einfach geschriebene Ziffern zu klassifizieren. Die Zuordnung um welche Ziffer es sich handelt, erfolgt meist unterbewusst. Der Prozess der Zuordnung wird uns erst bewusst, wenn wir nicht genau entscheiden können, welche Ziffer uns vorliegt. Die Frage: "Ist das eine eins oder eine sieben?" hat sich wahrscheinlich jeder schon einmal gestellt. In der Schreibweise von Ziffern und Buchstaben gibt es viele Einflussfaktoren, welche für Variation sorgen. Das kann beispielsweise die individuelle Handschrift eines Menschen sein, oder auch das Schreibgerät welches verwendet wurde.

In der Simulation befassen wir uns genau mit diesem Problem. Anstatt eines Menschen, legen wir die Ziffern aber neuronalen Netzen vor und bewerten diese anhand ihrer Vorhersagegenauigkeit.

Die neuronalen Netze bekommen in jeder Generation die gleichen 1000 Bilder vorgesetzt und sollen diese den Klassen: [0,1,2,3,4,5,6,7,8,9] zuordnen. Als Fitness-Wert dient die Anzahl der korrekt klassifizierten Bilder. Es handelt sich bei den Bildern um 28×28 Pixel Bilder aus dem MNIST-Datensatz. Die Bilder sind bereits in der Anwendung hinterlegt.



Die 1000 Bilder sind gleichmäßig auf ihre Klassen verteilt.

## Feedforward-Neuronales Netz

Ein Feedforward-Neuronales Netz ist eine Art von künstlichem neuronalen Netz, bei dem die Informationen nur in eine Richtung fließen: von den Eingabeneuronen über die versteckten Schichten (Hidden Layers) bis zu den Ausgabeneuronen. Es gibt keine Rückkopplungen oder Zyklen, wodurch das Netz einfach strukturiert ist und sich gut für viele grundlegende Aufgaben eignet.

Für Anwendungen im Bereich Explainable AI (XAI) ist es aufgrund seiner leicht nachvollziehbaren Struktur besonders geeignet, allerdings geht die Vereinfachung der Komplexität oft mit einer geringeren Vorhersagegenauigkeit einher.

In der Bildklassifizierung arbeitet ein Feedforward-Netz, indem es ein Bild als Eingabe in Form von Pixelwerten erhält. Diese Pixelwerte werden durch die Neuronen in der ersten Schicht geleitet, die einfache Merkmale wie Kanten oder Ecken erkennen. Diese Informationen werden dann an die nächste Schicht weitergegeben, die komplexere Merkmale identifiziert, wie Muster oder Formen. Schließlich erreicht die Information die Ausgabeschicht, die eine Klassifikation des Bildes vornimmt, z. B. in welche Kategorie das Bild gehört (z. B. eine Ziffer von 0 bis 9).

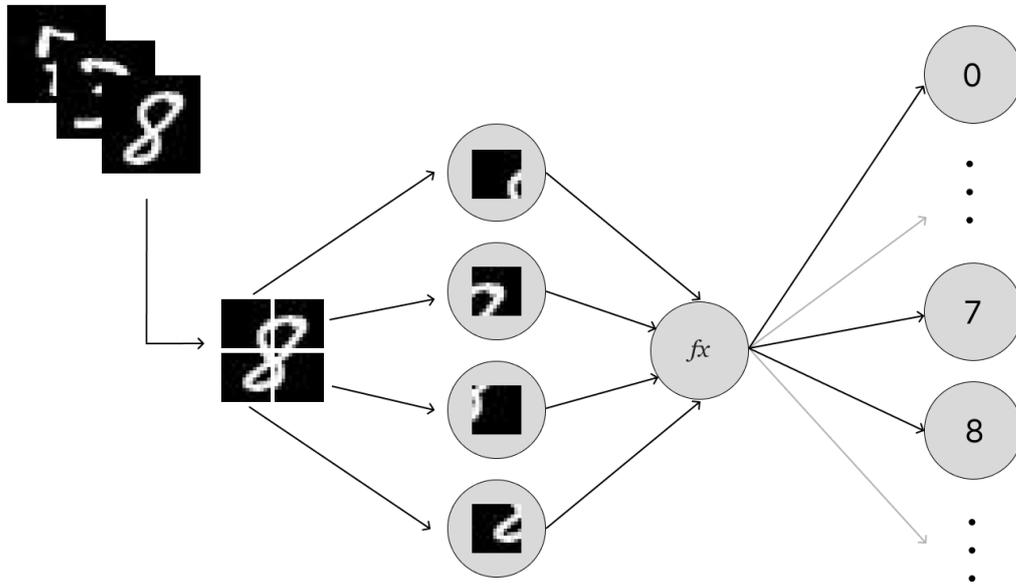


Abbildung eines vereinfachten Beispiels. Das Bild wird hierbei nicht in alle Pixel aufgeteilt, sondern in vier Elemente, welche einem Neuron in der Eingangsschicht zugewiesen werden.

## **A.2 HyperNEAT**

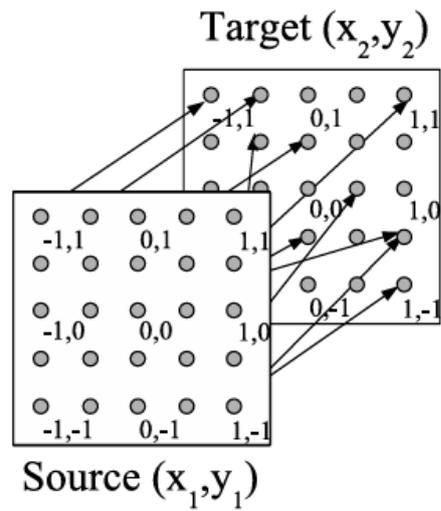
# HyperNEAT

**HyperNEAT** (Hypercube-based NeuroEvolution of Augmenting Topologies) ist eine Erweiterung des NEAT-Algorithmus, die speziell dafür entwickelt wurde, größere und komplexere neuronale Netze effizient zu erzeugen. Während NEAT sich auf die Evolution der Netzstruktur konzentriert, geht HyperNEAT einen Schritt weiter und ermöglicht es, Muster in der Struktur des Netzes auszunutzen, um effizienter zu lernen.

## Wie funktioniert HyperNEAT?

1. **Grundidee:** HyperNEAT basiert auf der Idee, dass neuronale Netze oft wiederkehrende Muster oder Symmetrien in ihren Verbindungen haben. Ein einfaches Beispiel wäre ein neuronales Netz, das Bilddaten verarbeitet, wo benachbarte Pixel oft ähnliche Merkmale aufweisen. HyperNEAT nutzt diese Symmetrien, um Netzwerke zu entwickeln, die solche Muster von vornherein berücksichtigen.
2. **CPPN (Compositional Pattern Producing Network):** Im Zentrum von HyperNEAT steht das CPPN, ein spezielles neuronales Netz, das als „Generator“ für die Gewichte des zu entwickelnden neuronalen Netzes dient. Anstatt jedes Gewicht einzeln zu optimieren, wie es NEAT macht, berechnet das CPPN die Gewichte basierend auf der relativen Position der Neuronen im Netz. Dies ermöglicht die Erzeugung von Netzwerken mit komplexen, aber geordneten Verbindungsstrukturen.
3. **Substrat:** Das Substrat in HyperNEAT bezeichnet den physischen Raum oder die geometrische Anordnung, in dem das neuronale Netz eingebettet ist. Es definiert die Positionen der Neuronen im Netzwerk, auf denen das CPPN operiert. Das Substrat kann als eine Art Raster oder Gitter betrachtet werden, auf dem die Neuronen angeordnet sind. Diese Anordnung ist entscheidend, weil das CPPN die Gewichte zwischen den Neuronen basierend auf deren Positionen im Substrat berechnet. So können räumliche Muster und Symmetrien in den Verbindungen natürlich entstehen, was besonders nützlich für Aufgaben wie Bildverarbeitung oder Robotik ist, bei denen die räumliche Beziehung zwischen den Eingaben eine zentrale Rolle spielt.

In der Implementierung dieser Anwendung wurde sich für die einfache Sandwich-Architektur entschieden. Das Sandwich-Substrat besteht aus mehreren Schichten von Neuronen, die wie die Schichten eines Sandwichs übereinander angeordnet sind. Jede Schicht kann eine unterschiedliche Funktion im neuronalen Netz haben, beispielsweise als Eingabe-, Verarbeitungs- oder Ausgabeschicht. Diese Anordnung ermöglicht es dem Netzwerk, Informationen in mehreren Verarbeitungsschritten zu verarbeiten, ähnlich wie in einem traditionellen mehrschichtigen neuronalen Netzwerk.

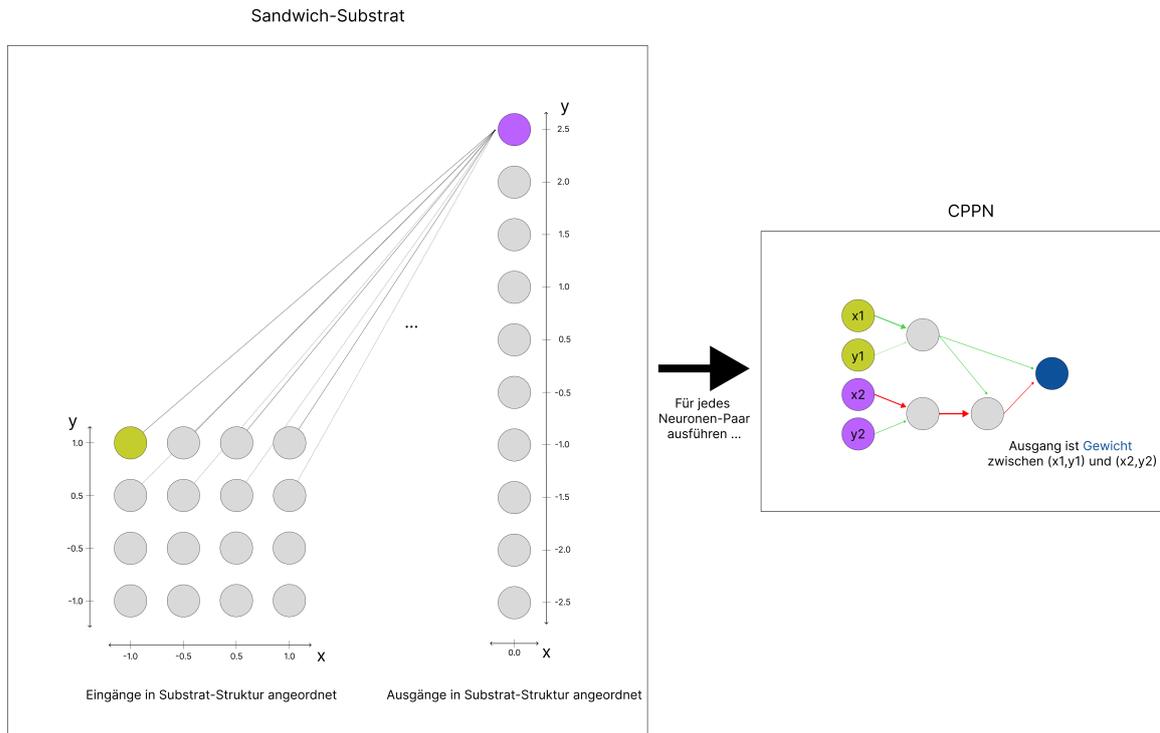


### (c) Sandwich

Abbildung einer einfachen Sandwich-Struktur.

Quelle: A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks (2009). Kenneth O. Stanley

4. **Entwicklung des Netzwerks:** HyperNEAT verwendet das CPPN, um die Gewichte für die Verbindungen zwischen den Neuronen zu bestimmen. Das CPPN kann dabei Muster und Symmetrien in den Gewichten erzeugen, die in vielen realen Anwendungen nützlich sind. Zum Beispiel könnte ein CPPN ein symmetrisches Muster erzeugen, das ein neuronales Netz effizienter macht, wenn es mit symmetrischen Daten arbeitet, wie etwa bei der Bilderkennung.



Beispiel für Substrat für ein 4×4 Pixel Bild. Alle Neuronen der Eingangsschicht werden mit allen Neuronen der Ausgangsschicht verbunden. Die Koordinaten werden alle nacheinander dem CPPN übergeben, welches die Verbindungsgewichte berechnet.

- Rolle von NEAT:** NEAT kommt in HyperNEAT ins Spiel, **indem es das CPPN optimiert**. Es hilft dabei, das beste CPPN zu finden, das die passenden Gewichte für das zu entwickelnde neuronale Netz erzeugt. Durch Evolution verbessert NEAT die Struktur und die Gewichtungen des CPPN, sodass das resultierende neuronale Netz immer besser an die zu lösende Aufgabe angepasst wird.

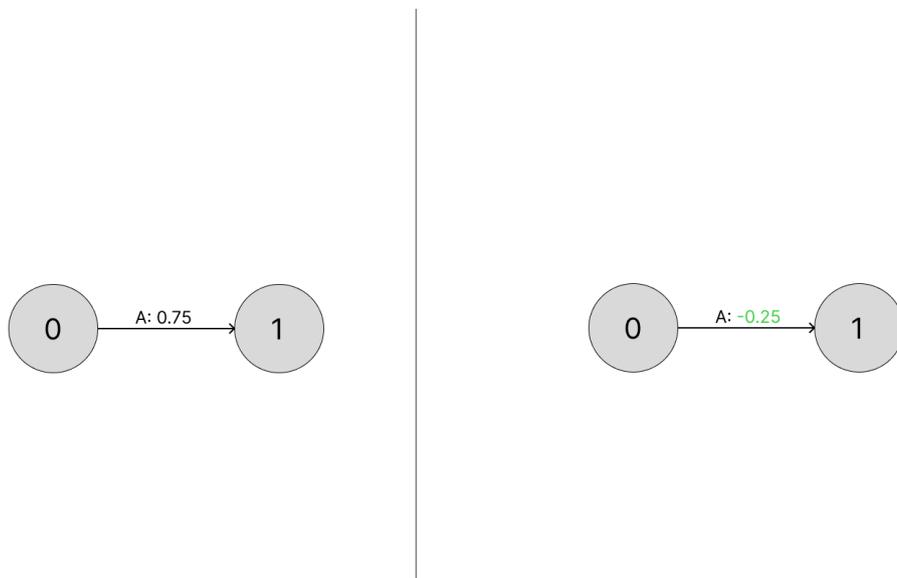
### **A.3 Mutation**

# Mutation

Eine Mutation in NEAT und HyperNEAT ist eine zufällige Veränderung in der Struktur eines neuronalen Netzwerks. Dieser Prozess ist vergleichbar mit der biologischen Evolution, bei der kleine Veränderungen in den Genen von Lebewesen auftreten. Diese Veränderungen können auf verschiedene Weise auftreten und beeinflussen, wie das Netzwerk Probleme löst oder Entscheidungen trifft. Durch Mutation können Netzwerke neue Fähigkeiten erlangen oder bestehende verbessern, ähnlich wie Lebewesen durch genetische Veränderungen an ihre Umwelt angepasst werden.

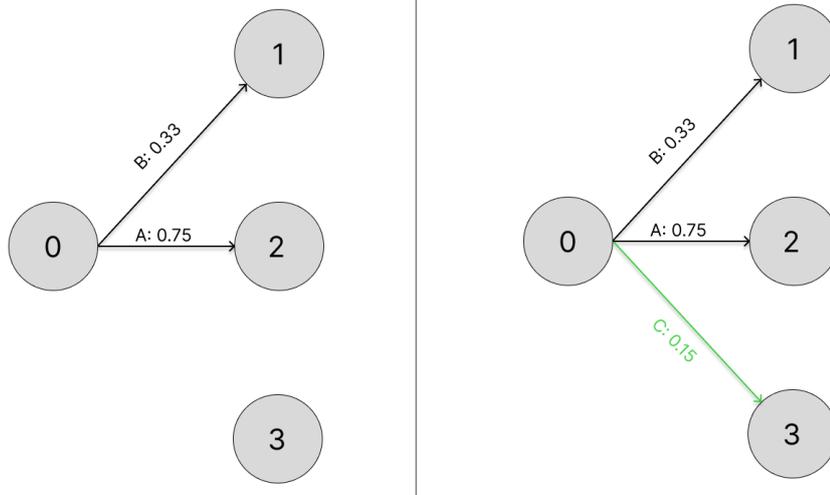
## Arten von Mutationen

1. **Verbindungsgewicht-Mutation:** Die Gewichte, welche die Stärke einer Verbindung zwischen zwei Neuronen bestimmen, werden verändert. Diese Änderung kann die Art und Weise beeinflussen, wie Informationen durch das Netzwerk fließen.



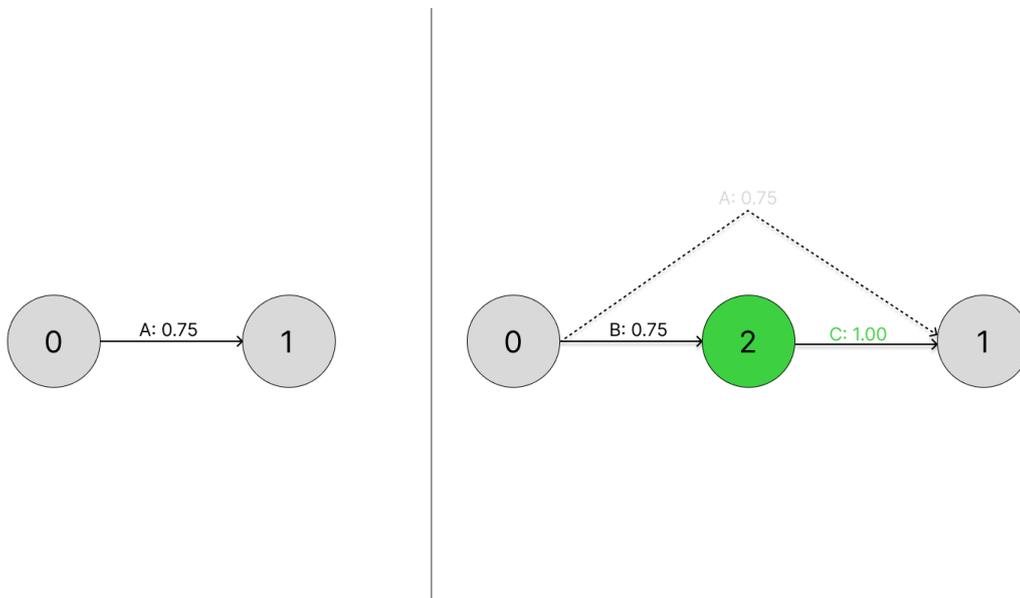
Selbst kleine Veränderungen in den Verbindungsgewichten können die Leistung des Netzwerks verbessern oder verschlechtern. Diese Mutation hilft dabei, feine Anpassungen vorzunehmen.

2. **Neue Verbindung:** Eine neue Verbindung zwischen zwei bisher nicht verbundenen Neuronen wird hinzugefügt.



Diese Mutation ermöglicht es dem Netzwerk, komplexere Verbindungen zu schaffen und dadurch möglicherweise besser auf verschiedene Eingaben zu reagieren.

3. **Neues Neuron:** Ein neues Neuron wird in eine bestehende Verbindung eingefügt, wodurch die Struktur des Netzwerks komplexer wird.



Durch das Hinzufügen eines Neurons kann das Netzwerk neue Wege finden, um Informationen zu verarbeiten, was die Fähigkeit zur Problemlösung verbessert.

### Warum sind Mutationen wichtig?

Mutationen sind entscheidend, weil sie die Vielfalt innerhalb einer Population von neuronalen Netzwerken erhöhen. Diese Vielfalt ist notwendig, damit die evolutionäre Suche effektiv nach besseren Lösungen suchen kann. Ohne Mutationen würden alle Netzwerke schnell sehr ähnlich werden, was die Chancen, wirklich innovative Lösungen zu finden, verringern würde.



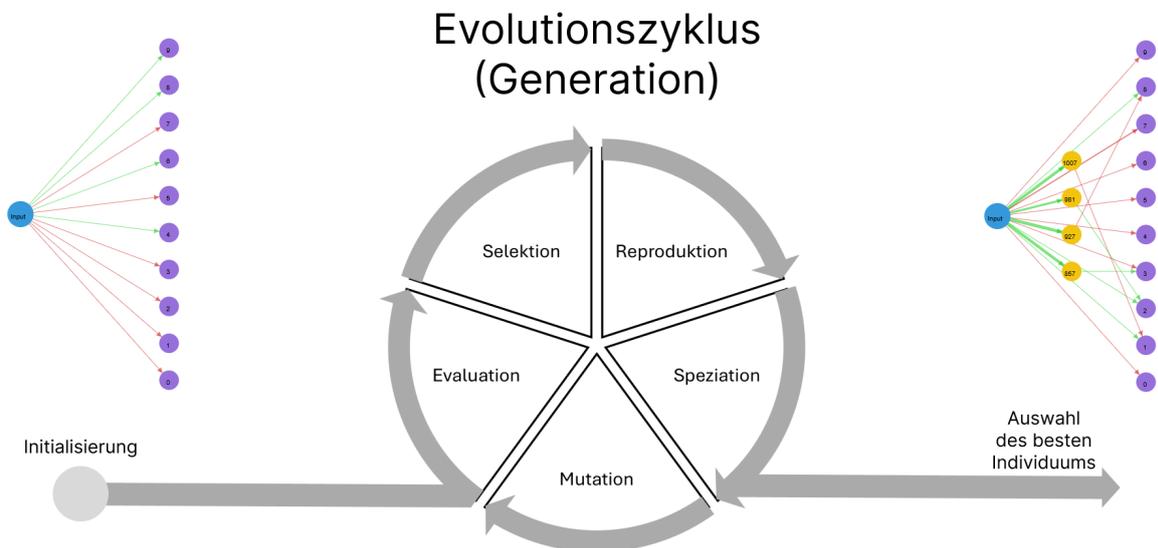
## **A.4 NEAT**

# NEAT

**NEAT (NeuroEvolution of Augmenting Topologies)** ist ein evolutionärer Suchalgorithmus zur Entwicklung und Optimierung von neuronalen Netzen. Im Gegensatz zu traditionellen Methoden, bei denen die Struktur eines neuronalen Netzes meist festgelegt ist und nur die Gewichte der Verbindungen angepasst werden, kann NEAT sowohl die Struktur als auch die Gewichte eines Netzes dynamisch verändern. Das bedeutet, dass NEAT in der Lage ist, Netze zu entwickeln, die im Laufe der Zeit komplexer werden und sich besser an die zu lösende Aufgabe anpassen.

Als evolutionärer Algorithmus nutzt NEAT Prinzipien der natürlichen Selektion und genetischen Variation, um Netzwerke zu optimieren. Netzwerke werden wie Individuen in einer Population behandelt, die durch Prozesse wie Kreuzung, Mutation und Selektion weiterentwickelt werden. Diese evolutionären Prozesse ermöglichen es NEAT, eine Vielzahl von Netzwerktopologien und -konfigurationen zu erkunden, um die bestmögliche Lösung für ein gegebenes Problem zu finden.

Neben der Kreuzung, bei denen neue Netzwerke basierend auf bestehenden Elternnetzen erzeugt werden, erlaubt NEAT auch die zufällige Generierung von Netzwerken. Diese zufällig erzeugten Netzwerke dienen dazu, den Suchraum zu erweitern und neue, möglicherweise innovative Strukturen in die Population einzuführen. Durch diese Methode wird sichergestellt, dass der Algorithmus nicht in lokalen Optima stecken bleibt, sondern weiterhin neue Ansätze erkundet, um optimale Lösungen zu finden.

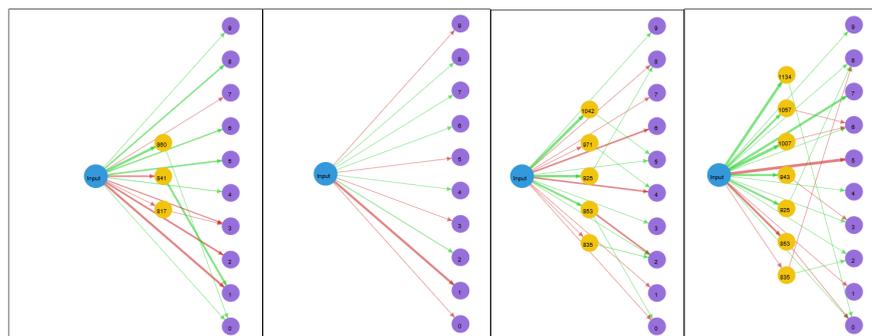


Visualisierung der Prozessschritte, welche in jeder Generation durchlaufen werden.

## Wie funktioniert NEAT?

1. **Initialisierung:** Der Prozess beginnt mit einer Population von einfachen neuronalen Netzen, die zufällig erzeugt werden. Diese Netze bestehen nur aus Eingabe- und Ausgabeschichten, ohne versteckten Schichten.
2. **Evaluation:** Jedes Netz in der Population wird auf eine bestimmte Aufgabe getestet, beispielsweise die Klassifizierung von Bildern. Die Leistung wird durch eine Fitnessfunktion bewertet, die misst, wie gut das Netz die Aufgabe löst.
3. **Selektion:** Die Netze mit der besten Leistung (höchste Fitness) werden ausgewählt, um in die nächste Generation überzugehen. Dabei werden weniger erfolgreiche Netze aussortiert.
4. **Reproduktion:** Die leeren Plätze in der Population, welche durch den Selektionsprozess entstanden sind, werden wieder aufgefüllt.
5. **Speziation:** NEAT gruppiert Netze in sogenannte Spezies, um die Vielfalt innerhalb der Population zu erhalten. Netze, die ähnliche Eigenschaften haben, werden in dieselbe Spezies eingeordnet. Diese Mechanik verhindert, dass neuartige und potenziell wertvolle Strukturen in den frühen Phasen der Evolution verloren gehen, bevor sie sich bewährt haben.
6. **Mutation:** Bei der Mutation können verschiedene Änderungen am Netz vorgenommen werden, wie das Hinzufügen neuer Neuronen oder Verbindungen, das Ändern der Gewichte oder das Entfernen bestehender Verbindungen. Diese Veränderungen ermöglichen es dem Netz, neue Strukturen auszuprobieren und potenziell leistungsfähigere Architekturen zu entwickeln.

**Iterative Evolution:** Dieser Zyklus aus Bewertung, Selektion, Reproduktion, Speziation und Mutation wird wiederholt, wobei die Netze von Generation zu Generation immer leistungsfähiger werden. Mit der Zeit entstehen immer komplexere und effektivere Netzstrukturen, die besser in der Lage sind, die zu lösende Aufgabe zu meistern.



Verschieden Netzwerktopologien, welcher sich durch Anwendung von NEAT über Generationen evolviert haben.

## **A.5 Parameterübersicht**

# Parameterübersicht

## Legende:

- Nicht implementierte oder nicht konfigurierbare Parameter

Kategorie	Parameter	Englischer Name	Erklärung	In der Anwendung
<b>Evaluation</b>	Fitness-Funktion	Fitness Function	Bewertet die Leistung eines neuronalen Netzes anhand einer bestimmten Aufgabe. Ein höherer Fitnesswert bedeutet bessere Leistung.	Die Anzahl der Korrekt vorhergesagten Bilder / Gesamtzahl der Bilder.
	Generationen	Evaluation Limit	Bestimmt, wie viele Generationen oder Epochen ein Netz evaluiert wird, bevor es weiterentwickelt oder verworfen wird.	Wurde auf 1000 Generationen festgesetzt.
<b>Speziation</b>	Kompatibilitätsschwelle	Species compability Threshold	Bestimmt, wie unterschiedlich zwei Netze sein müssen, um in verschiedene Spezies eingeteilt zu werden.	Zwischen 0.00 und 1.00. Bei Netzen mit vielen Neuronen sollte der Schwellwert unter 0.5 sein.
	Speziationsdruck	Speciation Pressure	Regelt, wie stark die Population in verschiedene Spezies aufgeteilt wird, um die genetische Vielfalt zu erhalten.	Die Maximale Speziesanzahl steht in der Anwendung in Abhängigkeit zu der Populationsgröße (x 0,2).
	Spezies-Stagnation Schwellenwert	Species Stagnation Threshold	Definiert die maximale Anzahl von Generationen, die eine Spezies ohne Fortschritt haben darf, bis sie von der Reproduktion ausgeschlossen wird.	Durch die Umsetzung des Speziationsdrucks nicht notwendig.
<b>Mutation</b>	Mutationschance (für Gewichte)	Weight Mutation Rate	Gibt an, wie wahrscheinlich es ist, dass die Gewichte der Verbindungen in einem Netz mutiert werden.	Kann in der Simulation eingestellt werden.

Kategorie	Parameter	Englischer Name	Erklärung	In der Anwendung
	Verbindunghinzufügechance	Link Mutation Rate	Bestimmt die Wahrscheinlichkeit, dass neue Verbindungen zwischen Neuronen im Netz hinzugefügt werden.	Kann in der Simulation eingestellt werden.
	Knotenhinzufügechance	Node Mutation Rate	Gibt an, wie oft neue Neuronen durch die Aufteilung bestehender Verbindungen eingefügt werden.	Kann in der Simulation eingestellt werden.
	Mutationsrate für Deaktivierung von Genen	Gene Deactivation Rate	Regelt, wie häufig Gene (z. B. Verbindungen oder Neuronen) deaktiviert werden, was die Topologie des Netzes vereinfacht.	Nicht implementiert
	Gewichtsstörung	Weight Perturbation Rate	Gibt an, wie stark die Störung der Gewichte sein darf.	Die Spanne reicht vom positive bis in den negativen Bereich.
	Schutzwürdigkeit	Protected Individuals	Gibt an, die wievielten besten Prozent an Individuen nicht verändert werden dürfen.	Betroffene Individuen werden durch folgendes Icon markiert:  (schwarz-weiß)
<b>Reproduktion</b>	Rekombinationsrate	Crossover Rate	Bestimmt, wie häufig genetisches Material zwischen zwei Elternnetzen ausgetauscht wird, um Nachkommen zu erzeugen.	Wurde intern auf 0.75 festgelegt. Bedeutet drei Viertel der Reproduzierten Individuen stammen aus der Kreuzung zwei Individuen.
	Kreuzungsrate	mutation with crossover	Die Wahrscheinlichkeit wie viele Individuen durch Kreuzung entstehen	Intern auf 75% gesetzt.
<b>Selektion</b>	Eliminierung	Selection Threshold	Bestimmt, welche Prozentzahl der schlechtesten Netze eliminiert werden. Die übrigen werden für die Mutation und	Betroffene Individuen werden durch folgendes Icon markiert:  (schwarz-weiß)

Kategorie	Parameter	Englischer Name	Erklärung	In der Anwendung
			Reproduktion ausgewählt.	
	Elitismusrate	Elitism Rate	Legt fest, wie viele der besten Netze einer Generation direkt in die nächste Generation übergehen, ohne verändert zu werden.	Nicht implementiert.

## **A.6 Reproduktion**

# Reproduktion

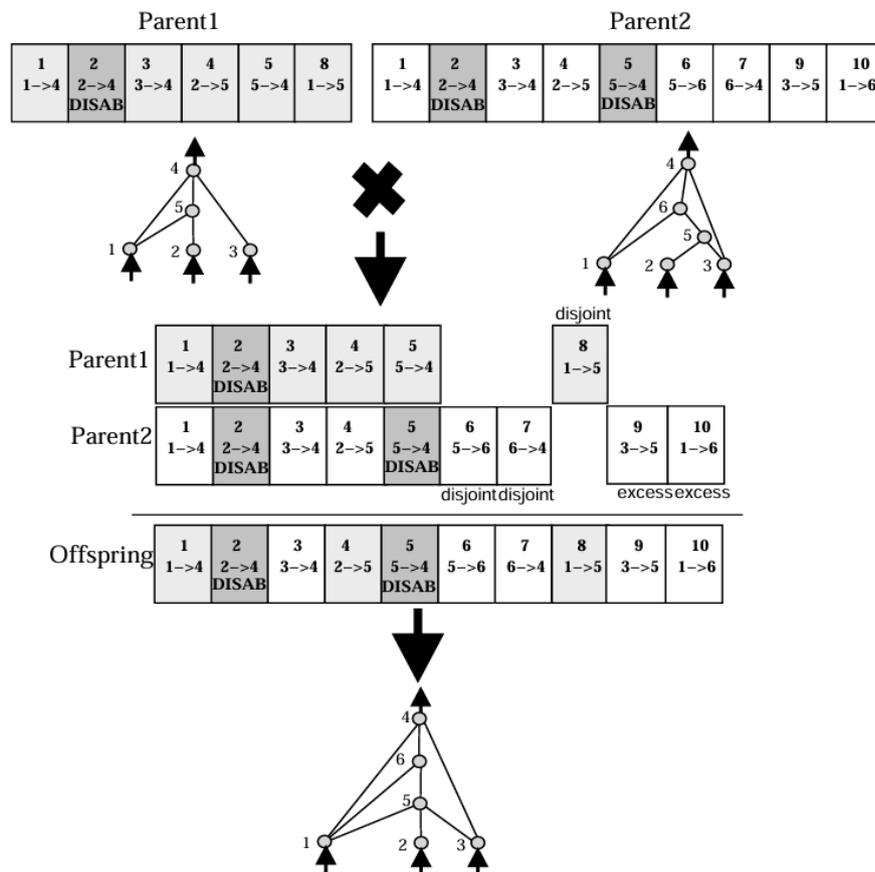
Die Reproduktion in NEAT kann über zwei Arten erfolgen. Entweder über das Erzeugen eines Individuums mit zufälligen Verbindungsgewichten und der minimal Menge an Neuronen oder über die Kreuzung (auch als "Crossover" bekannt) von Individuen. Die Kreuzung ermöglicht, Eigenschaften von zwei "Eltern"-Netzwerken zu kombinieren, um ein neues "Kind"-Netzwerk zu erzeugen. Dieser Prozess ist vergleichbar mit der biologischen Fortpflanzung, bei der das Erbgut beider Elternteile vermischt wird, um Nachkommen zu schaffen, die Merkmale beider Eltern erben.

## Arten der Reproduktion?

1. **Kreuzung:** Bei der Kreuzung (Crossover) werden zwei disjunkte Individuen, also Netzwerke aus unterschiedlichen Spezies oder mit deutlich verschiedenen Topologien, ausgewählt. Alle gemeinsamen Erbgutinformationen, d.h. Verbindungen und Neuronen, die in beiden Elternteilen vorhanden sind, werden direkt an das Kind weitergegeben. Für die Verbindungsgewichte dieser gemeinsamen Merkmale wird zufällig entschieden, von welchem Elternteil sie übernommen werden.

Das Elternteil, das ursprünglich den besseren Fitnesswert hatte, erhält einen zusätzlichen Vorteil: Es darf alle Neuronen und Verbindungen, die nicht in beiden Elternteilen vorhanden sind (also die disjunkten und überflüssigen Gene), ebenfalls an das Kind weitergeben. Diese Neuronen und Verbindungen können wichtige Innovationen darstellen, die das Kind befähigen, bessere Leistungen als die Eltern zu erbringen.

Dieser Prozess der Kreuzung ermöglicht es, vorteilhafte Merkmale aus beiden Elternteilen zu kombinieren und gleichzeitig Raum für neue, innovative Strukturen im Kind zu schaffen. Dadurch trägt die Kreuzung in NEAT dazu bei, die Evolution von Netzwerken voranzutreiben und die genetische Vielfalt innerhalb der Population zu bewahren.



Quelle: Kenneth O. Stanley (2001). Evolving Neural Networks through Augmenting Topologies

2. **Parthenogenese (eingeschlechtliche Fortpflanzung):** Bei der Parthenogenese in NEAT handelt es sich um eine Form der Fortpflanzung, bei der ein Individuum nur einen Elternteil besitzt. Das bedeutet, dass das neue Netzwerk direkt von einem einzelnen Elternnetzwerk abstammt. Dieses Elternteil übergibt alle seine Erbgutinformationen, einschließlich der vollständigen Netzwerktopologie und der Verbindungsgewichte, an das neue Individuum. **Zur Anwendung kommt diese Art, wenn eine Spezies aus nur einem Individuum besteht und somit keine geeigneten Partner für eine Kreuzung existieren.**

Um sicherzustellen, dass das neue Individuum nicht eine exakte Kopie seines Elternteils ist und weiterhin genetische Vielfalt in der Population gefördert wird, werden während dieses Prozesses die **Verbindungsgewichte leicht gestört (mutiert)**. Diese kleinen Mutationen sorgen dafür, dass das neue Netzwerk sich in seiner Struktur und Funktionsweise leicht von dem erzeugenden Netzwerk unterscheidet. Diese Methode ermöglicht eine Form der genetischen Variation ohne die Notwendigkeit eines zweiten Elternteils und trägt dazu bei, das Potenzial zum Entdecken neuer, besserer Lösungen in der Population zu erhöhen.

3. **Reproduktion von Netzen durch zufällige Generierung:** Bei der zufälligen Generierung werden Netzwerke ohne direkte Eltern erstellt. Das bedeutet, dass ihre Topologie und die initialen Verbindungsgewichte völlig neu und unabhängig von bestehenden Netzwerken in der Population zufällig festgelegt werden. Diese Netzwerke können in ihrer Struktur und Komplexität stark variieren, je nach den Parametern, die bei ihrer Erzeugung definiert sind.

Der Hauptzweck dieser Methode besteht darin, den Suchraum für mögliche Lösungen zu erweitern. Während Kreuzung und Parthenogenese hauptsächlich auf bestehenden, bewährten Strukturen aufbauen, kann die zufällige Generierung völlig neue und möglicherweise innovative Netzwerke in die Population einbringen. Obwohl viele dieser zufällig erzeugten Netzwerke möglicherweise nicht sofort erfolgreich sind, bietet ihre Einführung dennoch eine wertvolle Möglichkeit, unerforschte Bereiche des Lösungsspektrums zu erkunden.

## **A.7 Speziation**

# Speziation

Speziation bedeutet, dass die Population in verschiedene Gruppen, sogenannte Spezies, unterteilt wird. Netzwerke innerhalb derselben Spezies teilen eine gewisse strukturelle Ähnlichkeit, während Netzwerke aus unterschiedlichen Spezies sich deutlich voneinander unterscheiden können.

Der Zweck der Speziation besteht darin, Netzwerke mit ähnlichen Eigenschaften miteinander konkurrieren zu lassen, anstatt sie direkt gegen sehr unterschiedliche Netzwerke antreten zu lassen. Dies verhindert, dass neu entstandene oder ungewöhnliche Netzwerke von etablierten Netzwerken, die bereits eine hohe Fitness haben, sofort verdrängt werden. Stattdessen haben diese neuen Netzwerke Zeit, sich innerhalb ihrer eigenen Spezies weiterzuentwickeln und zu verbessern, bevor sie in direkter Konkurrenz zu anderen Spezies stehen.

In NEAT spielt die Distanzberechnung eine zentrale Rolle bei der Zuordnung von Netzwerken zu Spezies. Diese Berechnung bestimmt, wie ähnlich oder unterschiedlich zwei Netzwerke sind, und ist entscheidend für die Speziation, also die Einteilung der Population in verschiedene Gruppen oder Spezies.

## Wie funktioniert die Speziesbildung?

### 1. Distanzberechnung:

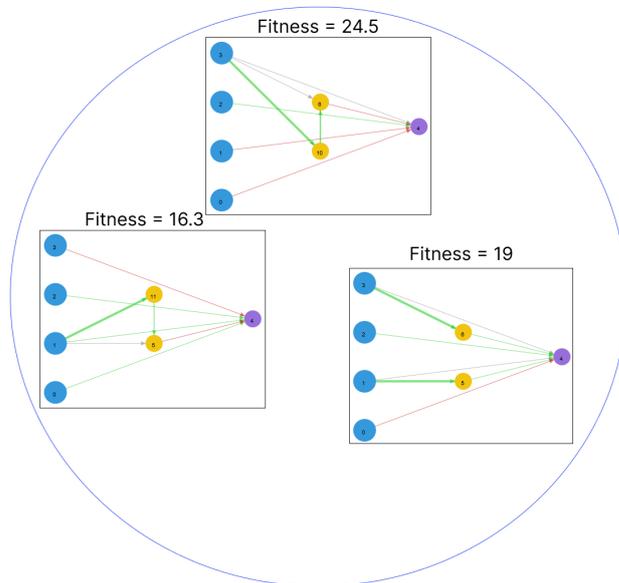
- **Was passiert?:** Für jedes Netzwerk wird eine genetische Distanz zu einem zufälligen anderen Netzwerk einer jeden Spezies berechnet. Diese Distanz misst, wie unterschiedlich zwei Netzwerke in Bezug auf ihre Struktur (Anzahl und Art der Verbindungen und Neuronen) sind. Netzwerke, die eine geringe genetische Distanz zueinander aufweisen, werden in dieselbe Spezies gruppiert. Netzwerke, die sich stark unterscheiden, werden in separate Spezies eingeordnet.

2. **Geteilte Fitness:** Die geteilte Fitness sorgt dafür, dass Netzwerke nicht isoliert, sondern im Kontext ihrer Spezies bewertet werden. Diese Bewertung ist essentiell für die Reproduktion. Die Individuen einer Spezies dürfen sich anteilig ihrer geteilten Fitness reproduzieren, was bedeutet, dass Netzwerke in überfüllten Spezies, deren Fitness geteilt und dadurch reduziert wird, weniger Chancen auf Reproduktion haben. Umgekehrt erhalten Netzwerke in kleineren Spezies, wo die geteilte Fitness höher ist, eine größere Chance, ihre Gene in die nächste Generation weiterzugeben. Dies fördert eine ausgewogene Evolution und erhält die genetische Vielfalt innerhalb der Population.

Die geteilte Fitness berechnet sich aus der Summe der Fitness innerhalb einer Spezies geteilt durch die Anzahl der Mitglieder.

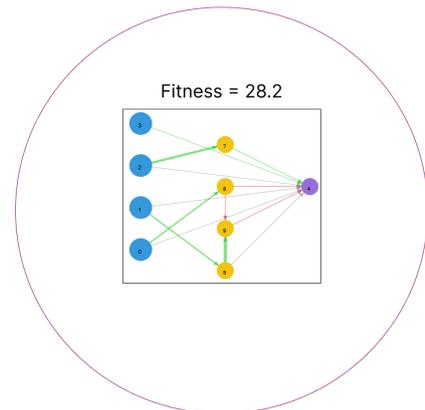
## Spezies 1

geteilte Fitness = 19.93



## Spezies 2

geteilte Fitness = 28.2



Darstellung der geteilten Fitness.

### Warum ist die Speziesbildung wichtig?

Ohne Speziesbildung könnten dominante Netzwerke die Population schnell homogenisieren, wodurch vielversprechende, aber weniger verbreitete Ansätze verdrängt würden. Speziesbildung stellt sicher, dass verschiedene Ideen überleben und weiterentwickelt werden können, was die Evolution effektiver macht.

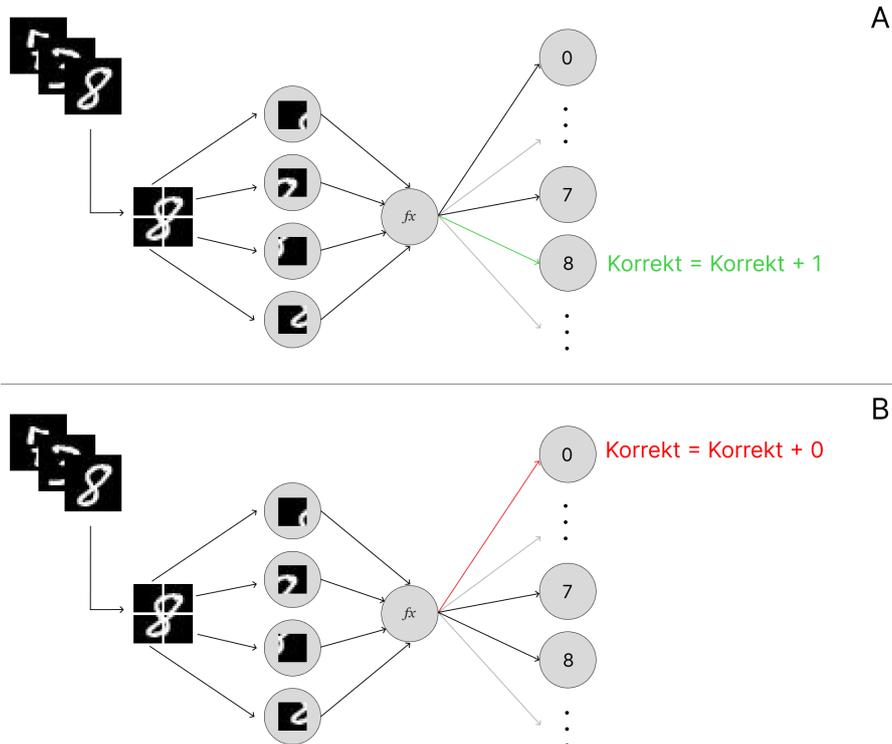
## **A.8 Evaluation**

# Evaluation

Die Evaluation in NEAT (NeuroEvolution of Augmenting Topologies) ist ein zentraler Prozess, bei dem die Leistung der entwickelten neuronalen Netze bewertet wird. Die Evaluation dient dazu, die Qualität der verschiedenen Netzwerke innerhalb einer Population zu bestimmen, damit die besten Netzwerke in die nächste Generation überführt werden können.

Der Evaluationsprozess basiert auf einer sogenannten **Fitness-Funktion**, die den Erfolg eines Netzes in Bezug auf eine bestimmte Aufgabe misst. Diese Fitness kann unterschiedlich definiert werden, je nach Problemstellung. Netzwerke mit höherer Fitness haben eine größere Chance, selektiert zu werden, um durch genetische Operationen wie Crossover und Mutation neue Netzwerke zu erzeugen.

Die wohl einsteigerfreundlichste Fitness-Funktion in der Bildklassifizierung ist **die Genauigkeit, mit der Bilder korrekt klassifiziert werden**. Jedes neuronale Netz in der Population erhält die gleichen 1000 Bildern als Eingabe und muss diese in die korrekten Kategorien einordnen.



In Abbildung A ist die Vorhersage für ein Eingabebild korrekt, weshalb der interne "Korrekt"-Zähler hochgezählt wird. In Abbildung B ist dies nicht der Fall.

**Die Leistung eines Netzes wird dann anhand des Prozentsatzes der korrekt klassifizierten Bilder gemessen. Wenn ein Netz beispielsweise 100 Bilder klassifiziert und davon 75 korrekt einordnet, beträgt die Genauigkeit 75%.** Dieser Wert repräsentiert die Fitness des Netzwerks für diese spezielle Aufgabe.

Dieser Evaluationsprozess wird wiederholt, bis die Netze eine ausreichende Genauigkeit erreichen oder eine vorgegebene Anzahl an Generationen durchlaufen wurde. Ziel ist es, im Laufe der Evolution Netzwerke zu erzeugen, die eine immer höhere Klassifizierungsgenauigkeit erzielen.

## **A.9 Selektion**

# Selektion

Die Selektion bestimmt, welche Individuen in die nächste Generation überführt werden und somit zur Evolution beitragen. In NEAT wird die Selektion so gestaltet, dass sowohl leistungsstarke als auch vielfältige Netzwerke bevorzugt werden, um die Optimierung des neuronalen Netzwerks zu unterstützen und gleichzeitig die genetische Vielfalt innerhalb der Population zu erhalten. Im Umkehrschluss sind **nicht-selektierte Individuen zur Eliminierung freigegeben**.

Die Selektion erfolgt in Abhängigkeit zu den Selektiven-Parametern für jede Spezies einzeln.

Individuen-Tabelle									
	Spezies ID	Ranking innerhalb Spezies	Genom ID	Anzahl Eingangneuronen	Anzahl versteckter Neuronen	Anzahl Ausgangsneuronen	Anzahl Verbindungen	Vorhersagegenauigkeit (%)	Anzeigen
☉	1	1	3	784	0	10	7840	15,80 %	▶
☉	1	2	9	784	0	10	7840	14,60 %	▶
	1	3	5	784	1	10	7841	14,00 %	▶
	1	4	11	784	0	10	7840	14,00 %	▶
	1	5	2	784	2	10	7842	13,00 %	▶
☠	1	6	0	784	0	10	7840	8,80 %	▶
☠	1	7	14	784	0	10	7840	8,60 %	▶
☉	2	1	13	784	1	10	7841	11,40 %	▶
	2	2	10	784	0	10	7840	9,40 %	▶

Der Totenkopf dient als Kennzeichnung für Individuen, die nicht mit übernommen werden. Alle anderen Genome werden mit übernommen. Individuen mit einem durchgestrichenen Kreis werden ohne Veränderung übernommen.

Der Selektionsprozess in NEAT sorgt dafür, dass leistungsstarke Netzwerke gefördert werden, während gleichzeitig Innovationen geschützt und die Vielfalt innerhalb der Population aufrechterhalten werden. Dies ermöglicht eine robuste Evolution, die im Laufe der Zeit zu immer besser angepassten neuronalen Netzwerken führt.

## **A.10 Fragebogen - Antworten**

# Fragebogen - BA

12 Antworten

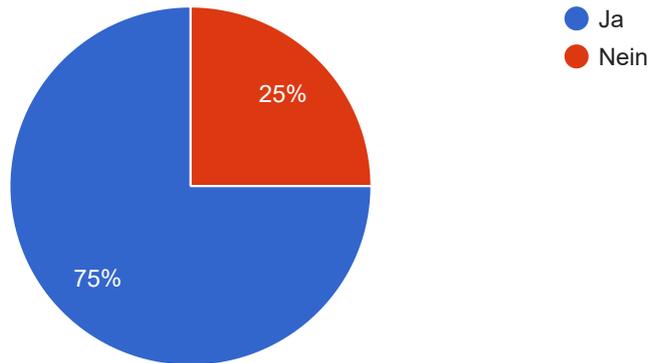
[Analytics veröffentlichen](#)

## Fragen I

Hast Du bereits Vorkenntnisse zur Thematik *Neuronale Netze* erlangt?

[Kopieren](#)

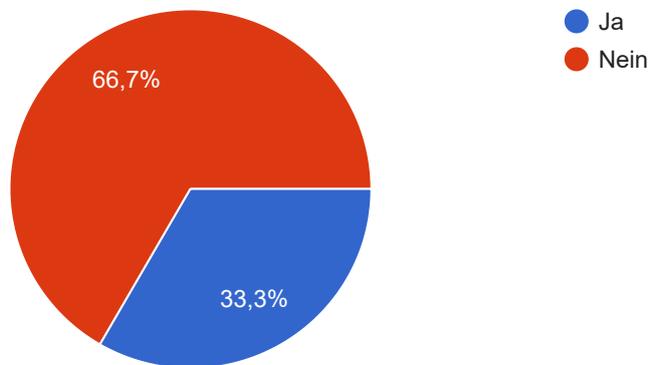
12 Antworten



Hast Du bereits vom NEAT Algorithmus gehört?

[Kopieren](#)

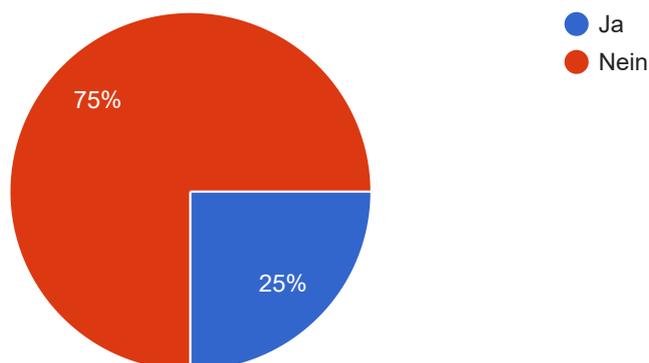
12 Antworten



Hast Du bereits vom HyperNEAT Algorithmus gehört?

[Kopieren](#)

12 Antworten

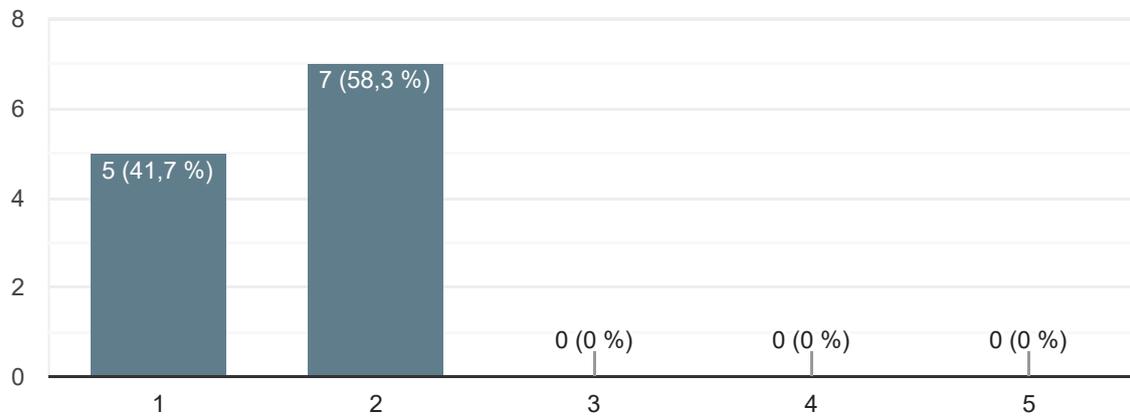


## Fragen II

Die XAI-Anwendung ist strukturiert aufgebaut.

 Kopieren

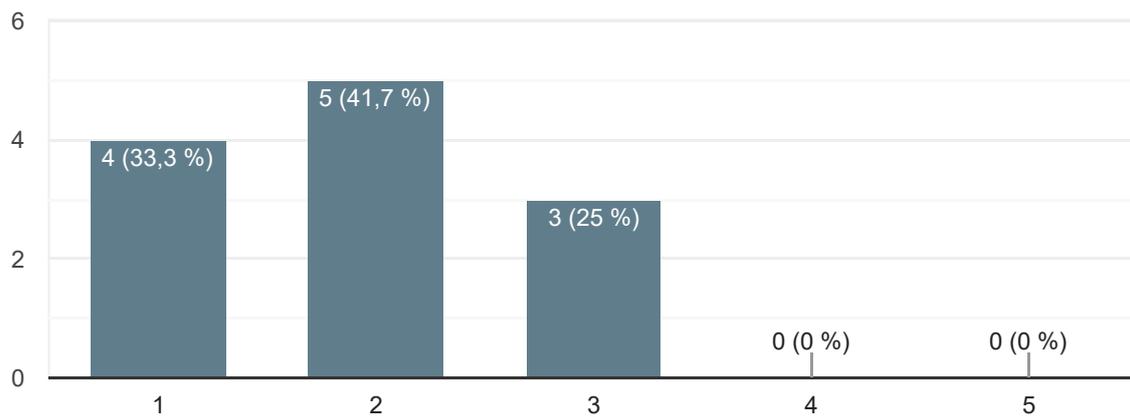
12 Antworten



Die XAI-Anwendung ist intuitiv zu bedienen.

 Kopieren

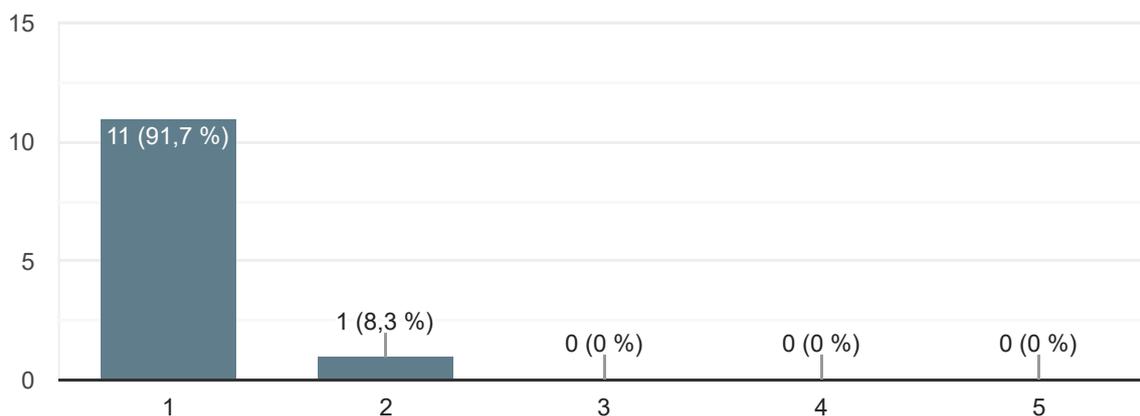
12 Antworten



Die XAI-Anwendung lief während der gesamten Nutzung stabil und ohne Abstürze.

 Kopieren

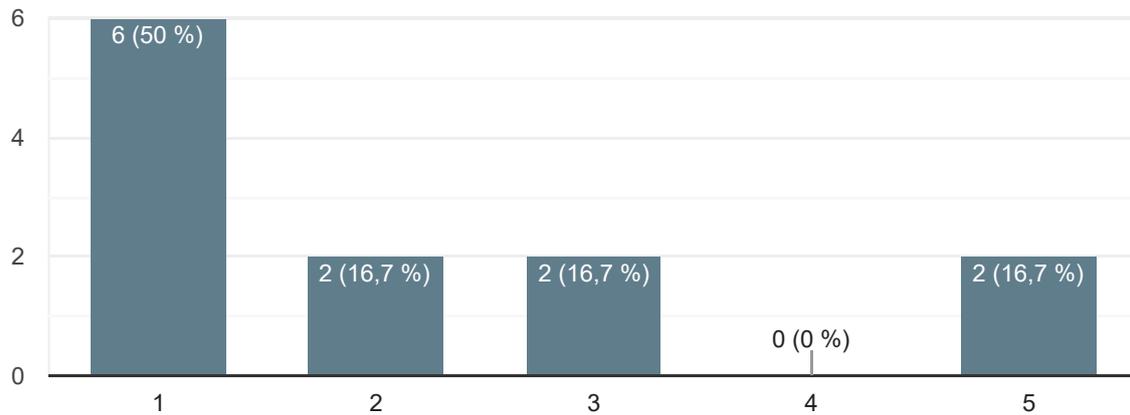
12 Antworten



Es traten keine technischen Probleme auf, die meine Arbeit behinderten.

 Kopieren

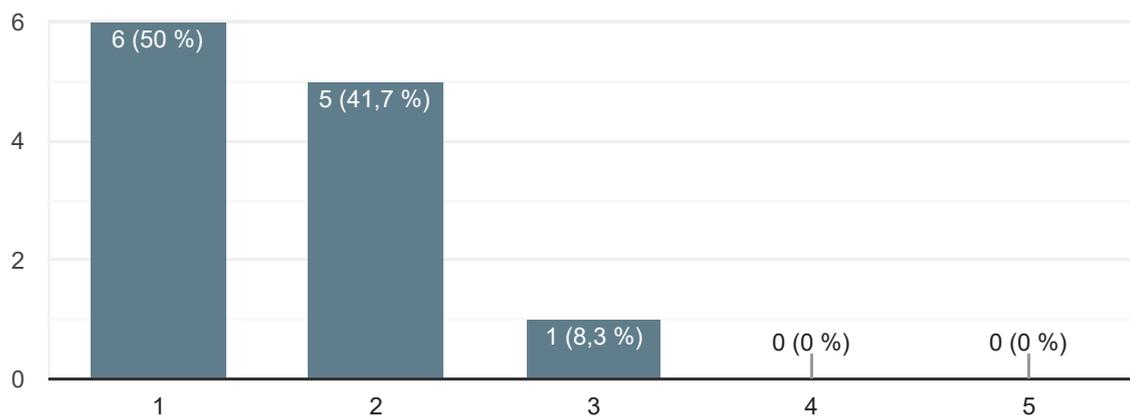
12 Antworten



Die Parameterkonfiguraion war übersichtlich und leicht einzustellen.

 Kopieren

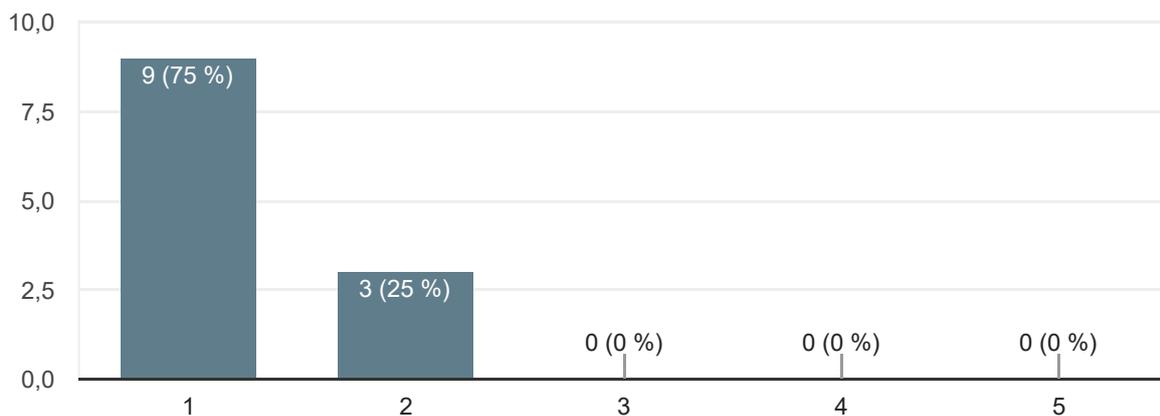
12 Antworten



Die Ladezeiten und Reaktionszeiten der XAI-Anwendung waren angemessen.

 Kopieren

12 Antworten

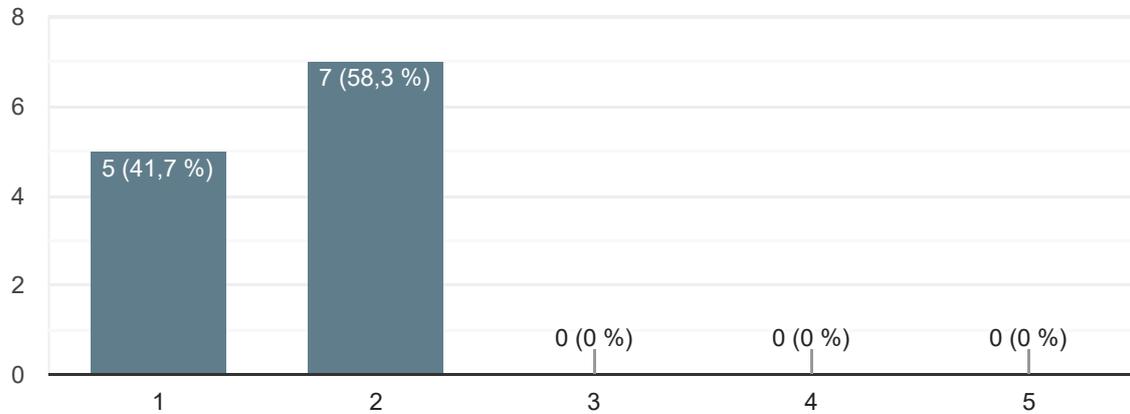


### Fragen III

Die Erklärungen der XAI-Anwendung waren durchgängig nachvollziehbar.

 Kopieren

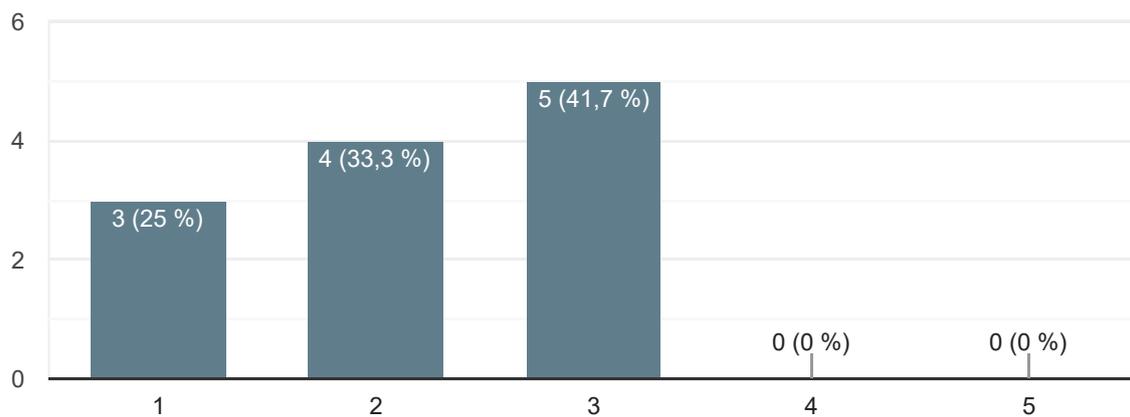
12 Antworten



Ich konnte jede Erklärung in der Simulation klar identifizieren.

 Kopieren

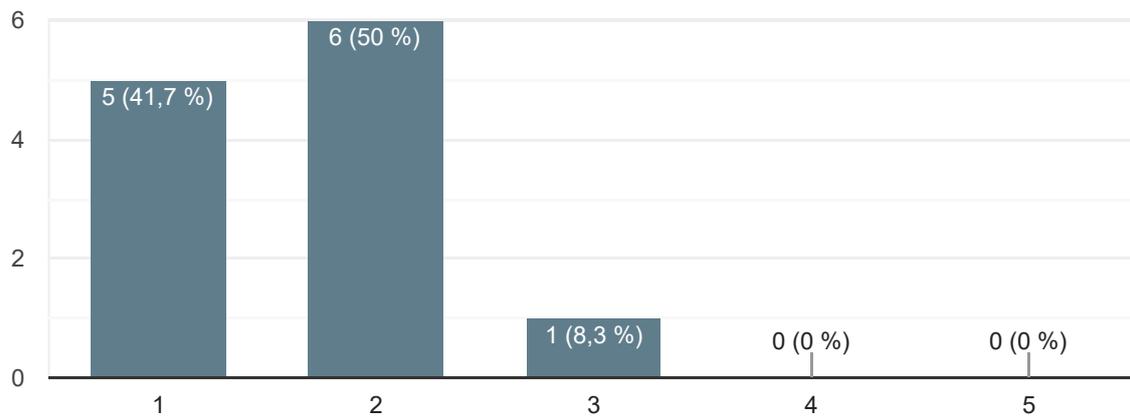
12 Antworten



Die Darstellung der Erklärungen half mir, die Entscheidungsprozesse der Algorithmen besser zu verstehen.

 Kopieren

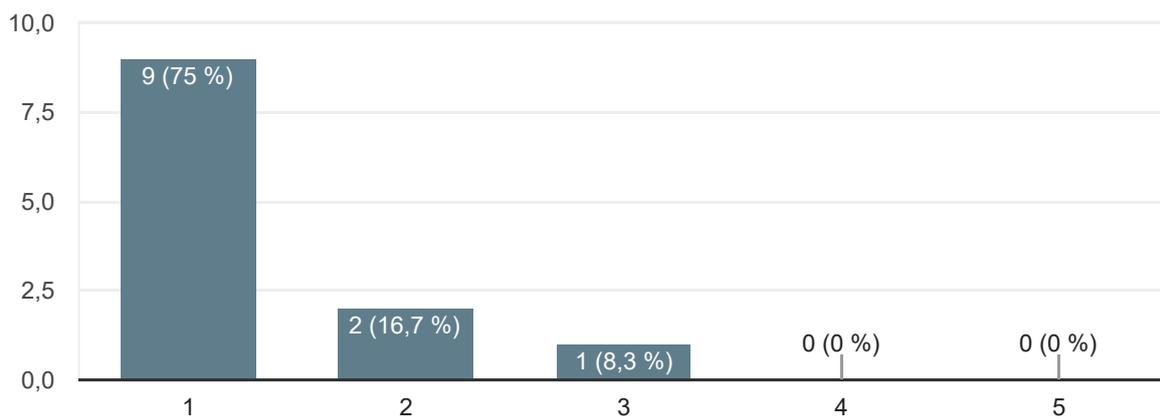
12 Antworten



Die Auswahl an konfigurierbaren Parametern in der Simulation empfand ich als ausreichend.

 Kopieren

12 Antworten



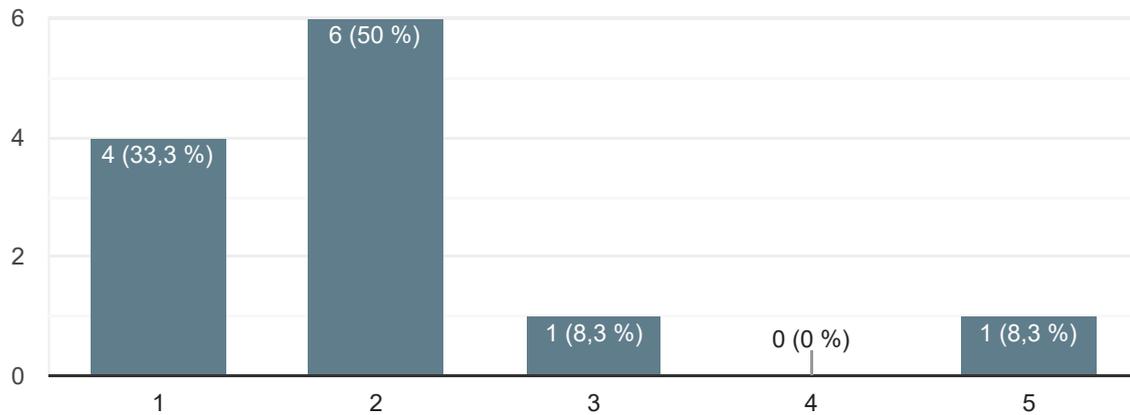
Fragen IV



Die Ergebnisse der XAI-Anwendung waren für mich nachvollziehbar und überprüfbar.

 Kopieren

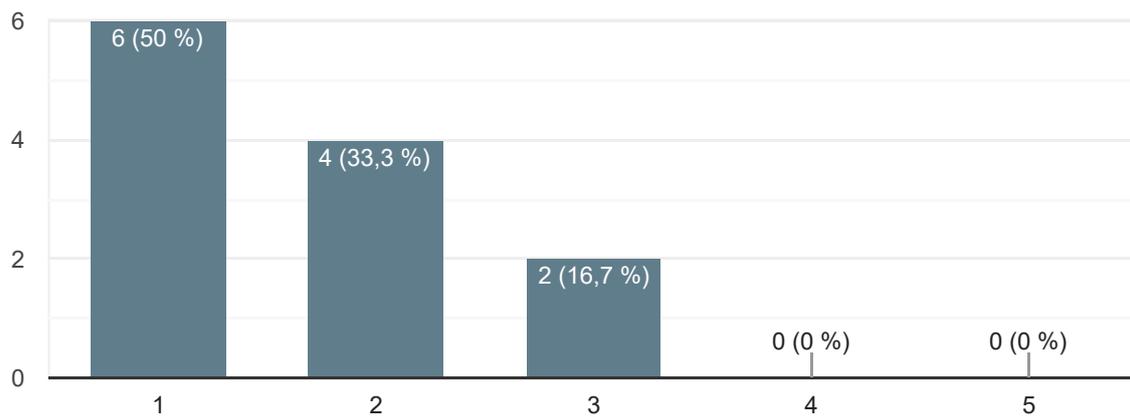
12 Antworten



Die Möglichkeit, die Simulation mehrfach durchzuführen, erleichterte mir die Überprüfung der Ergebnisse.

 Kopieren

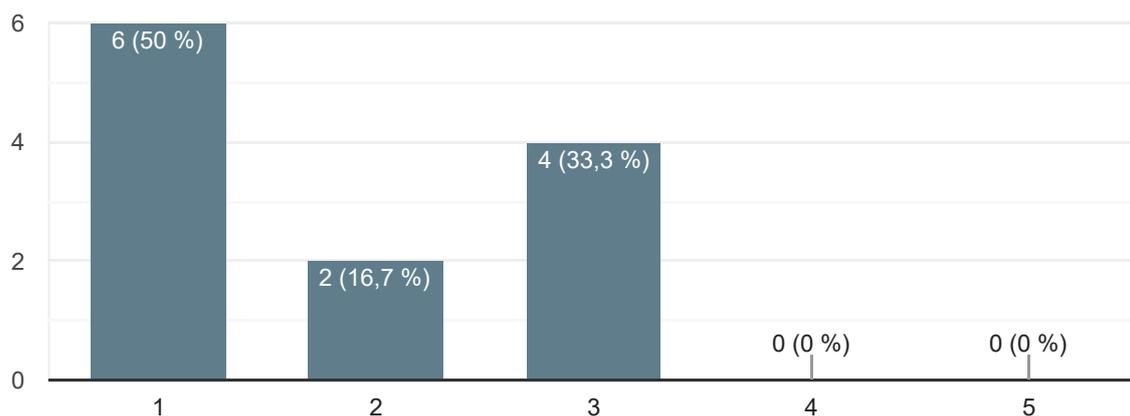
12 Antworten



Die Möglichkeit, Parameter der Simulation anzupassen, hat mir bei der Überprüfung der Algorithmen geholfen.

 Kopieren

12 Antworten

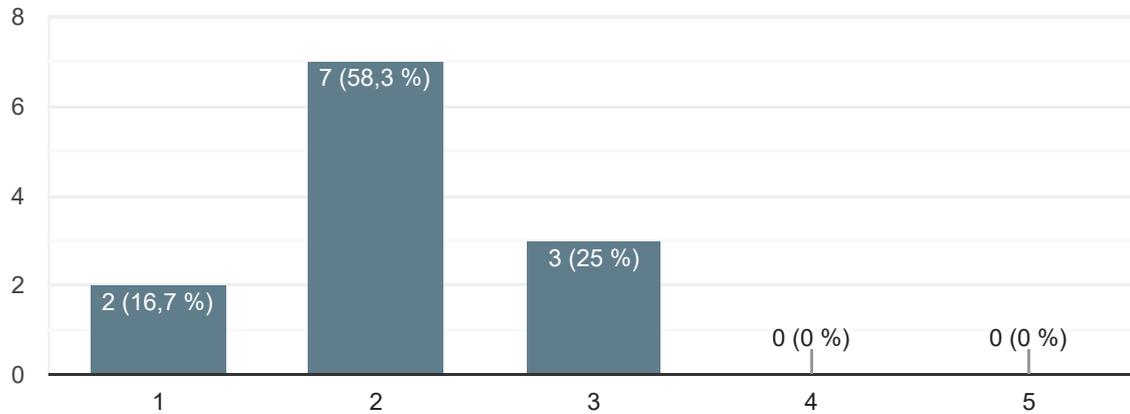


## Fragen V

Ich würde bei zukünftigen Projekten mit NEAT oder HyperNEAT auf die XAI-Anwendung zurückgreifen.

 Kopieren

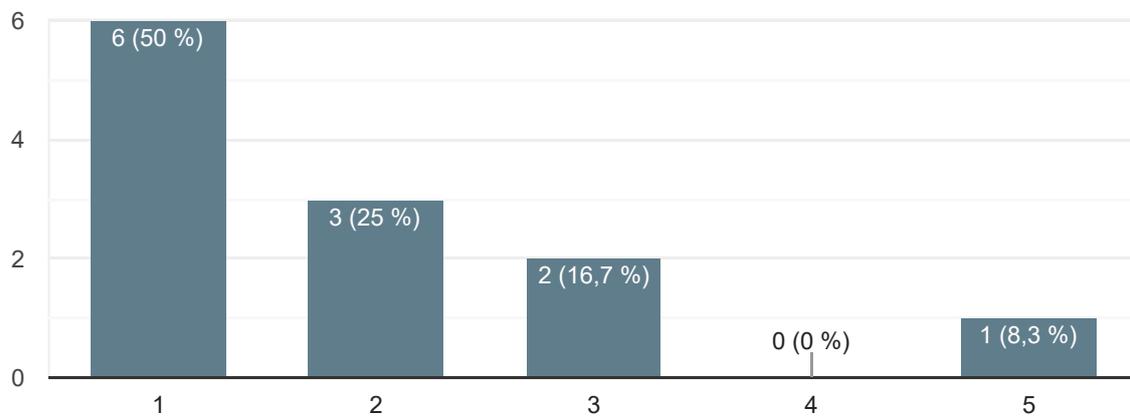
12 Antworten



Ich habe zu keinem Zeitpunkt während der Simulation an der Vorhersagegenauigkeit gezweifelt.

 Kopieren

12 Antworten



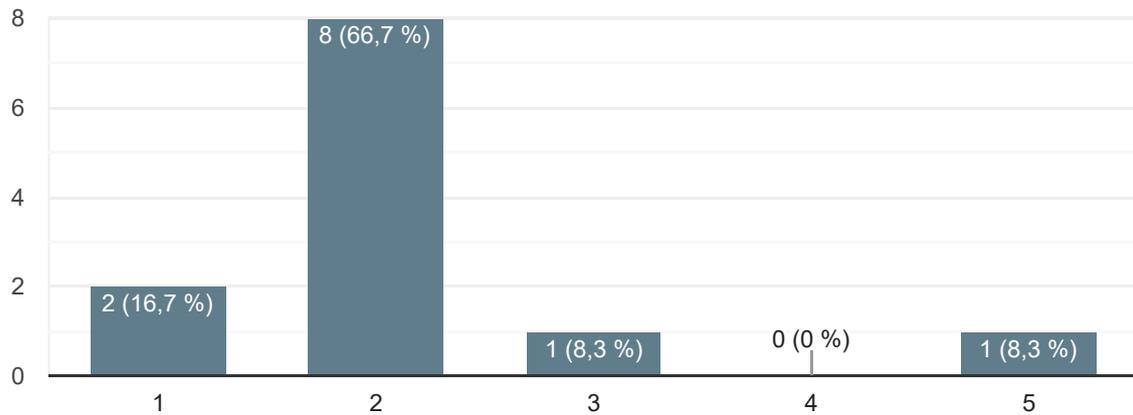
## Fragen VI



Ich bin davon überzeugt, dass ich durch die Nutzung der XAI-Anwendung ein gutes Verständnis von NEAT entwickelt habe.

 Kopieren

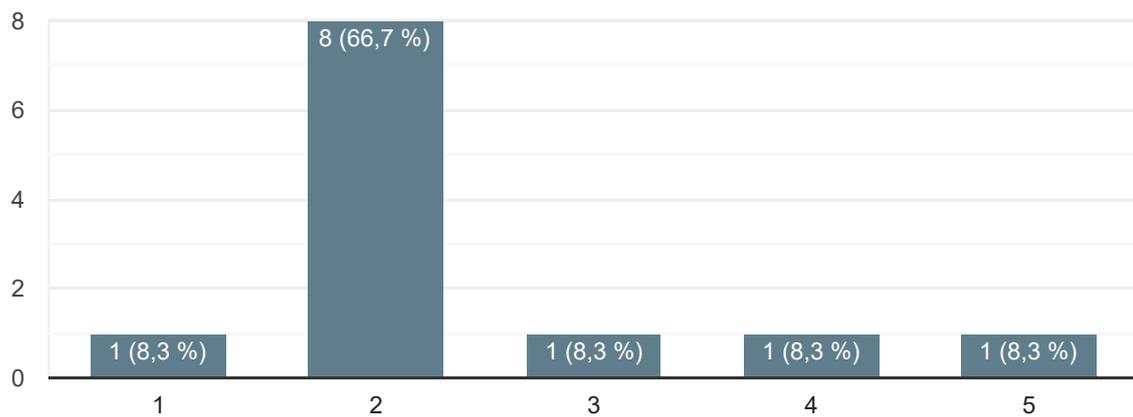
12 Antworten



Ich bin davon überzeugt, dass ich durch die Nutzung der XAI-Anwendung ein gutes Verständnis von HyperNEAT entwickelt habe.

 Kopieren

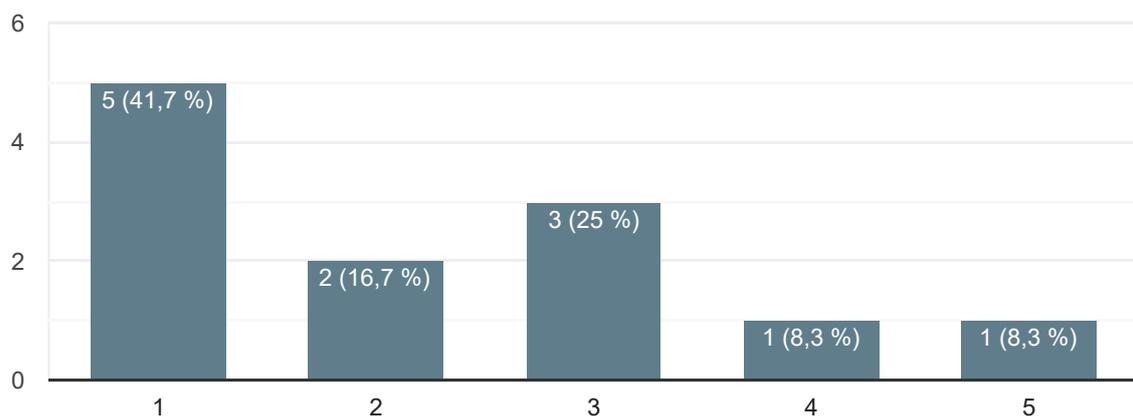
12 Antworten



Ich habe Interesse daran entwickelt, mich weiter mit den Algorithmen auseinanderzusetzen.

 Kopieren

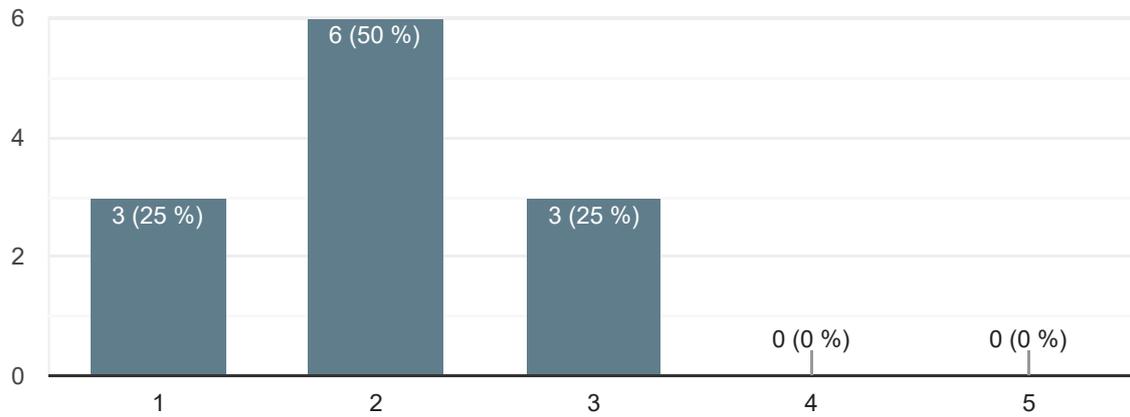
12 Antworten



Die Anwendung hat mich davon überzeugt, dass die Algorithmen in der Praxis nützlich und zuverlässig sind.

 Kopieren

12 Antworten

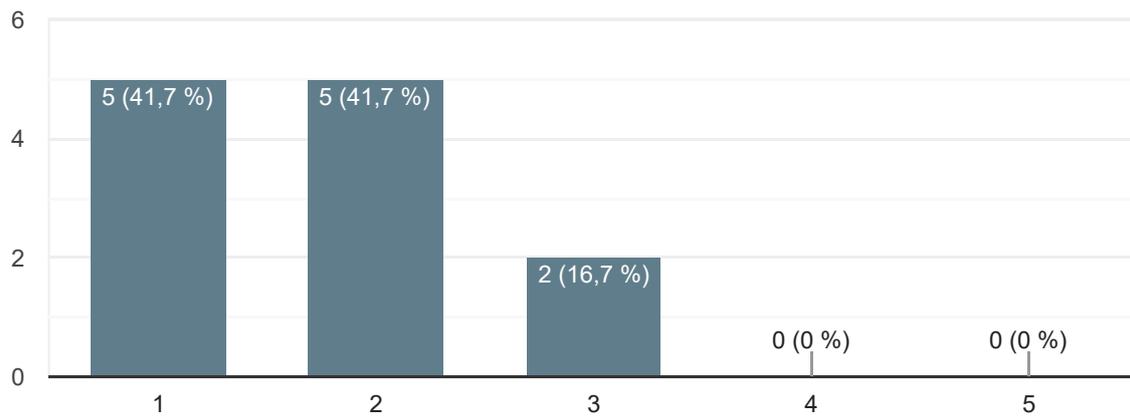


### Fragen VII

Ich bin insgesamt zufrieden mit den bereitgestellten Erklärungen.

 Kopieren

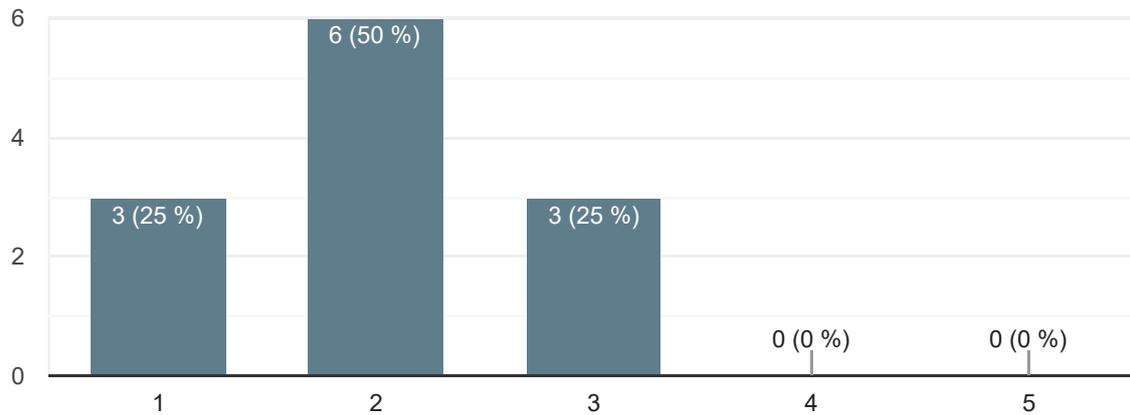
12 Antworten



Die XAI-Anwendung hat meine Erwartungen an die Erklärbarkeit von NEAT und HyperNEAT erfüllt.

 Kopieren

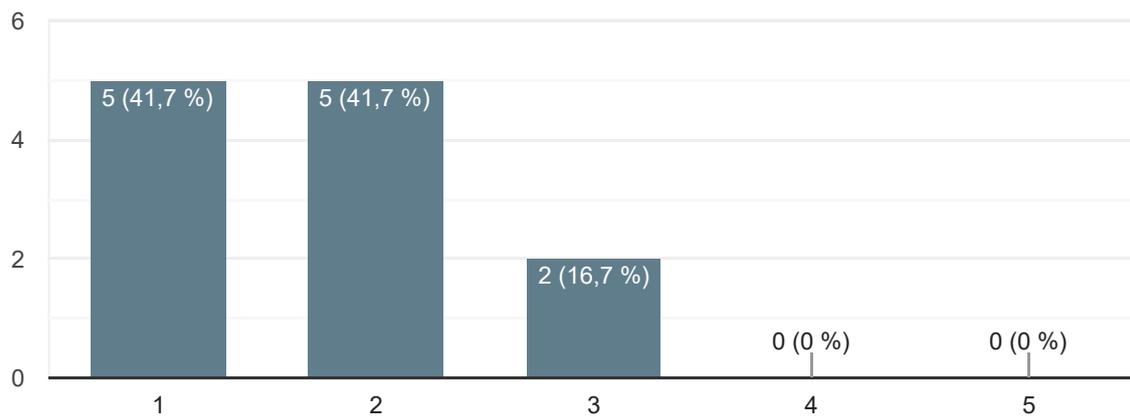
12 Antworten



Die Erklärungen waren ausgewogen, indem sie Text, Bilder, Videos und Simulationen angemessen kombinierten.

 Kopieren

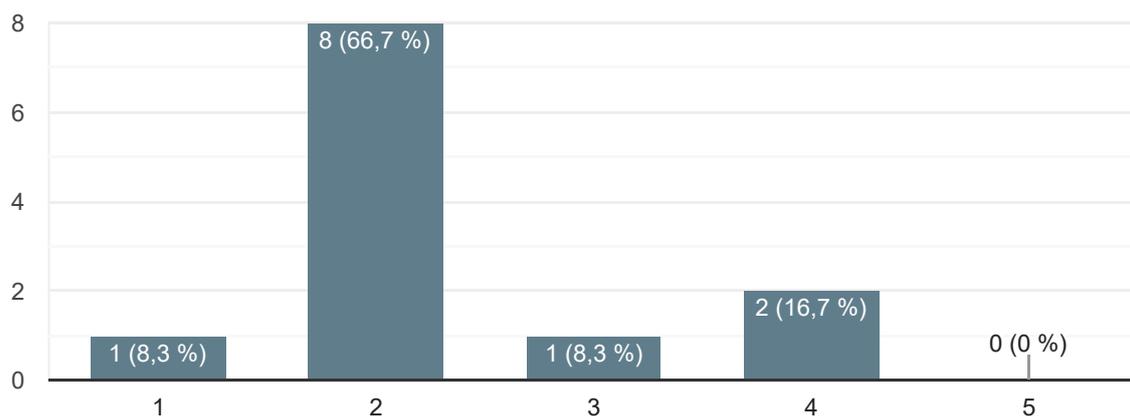
12 Antworten



Das Design der XAI-Anwendung entspricht meinen Erwartungen an eine solche Anwendung.

 Kopieren

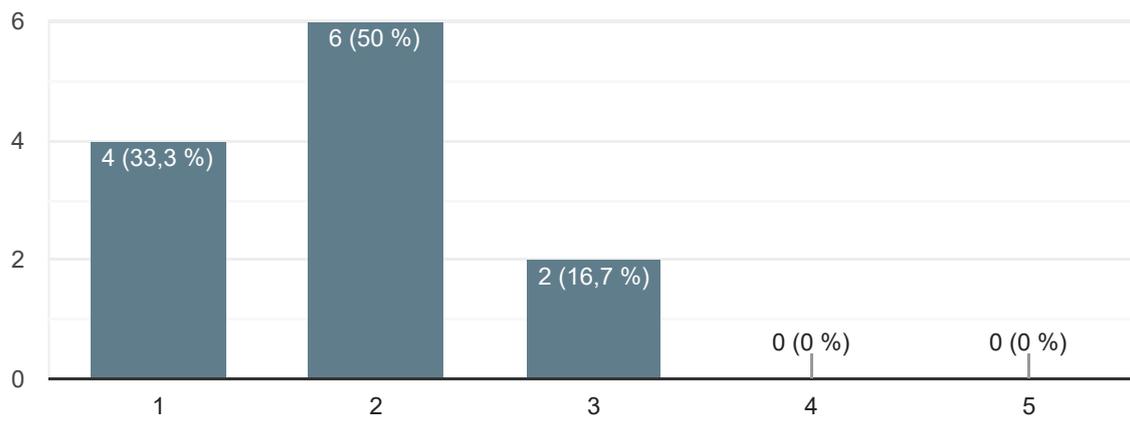
12 Antworten



Ich würde die XAI-Anwendung weiterempfehlen.

 Kopieren

12 Antworten

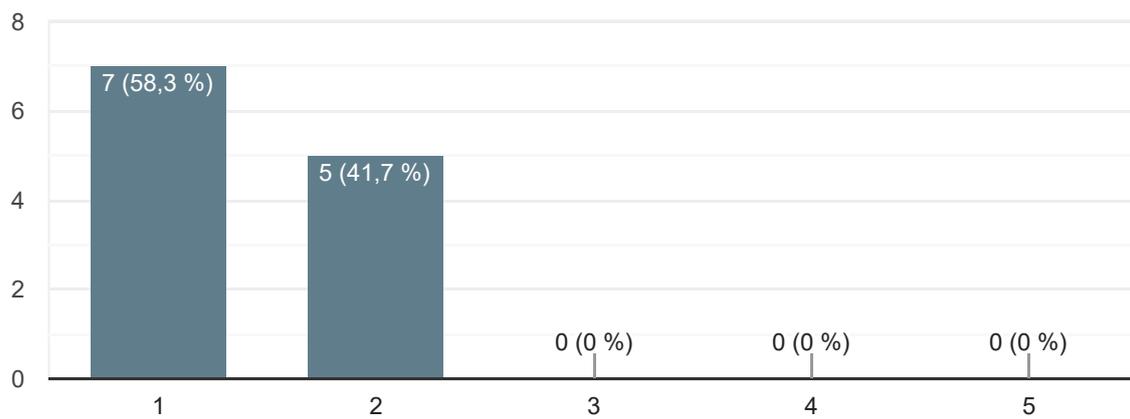


VIII

Die XAI-Anwendung hat mein Wissen über evolutionäre Algorithmen erweitert.

 Kopieren

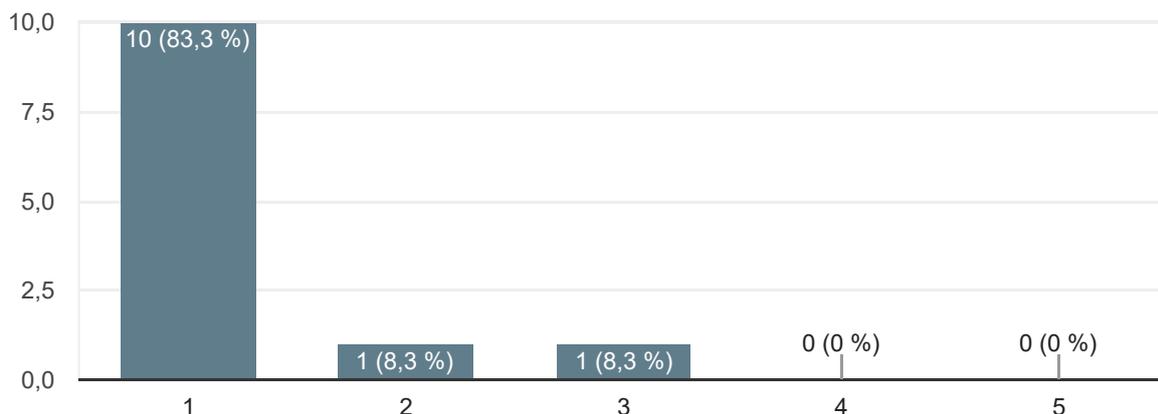
12 Antworten



Das Problem der Bildklassifizierung war ein leicht verständliches Problem.

 Kopieren

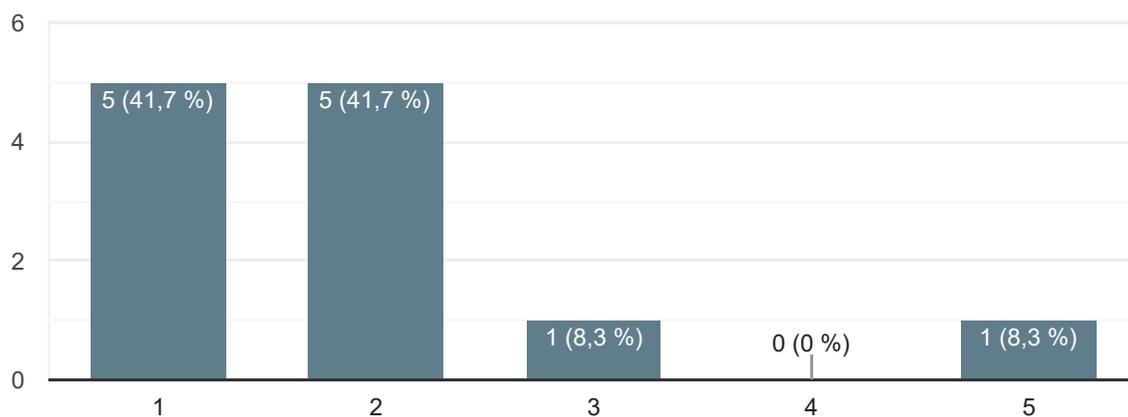
12 Antworten



Die XAI-Anwendung hat mir geholfen, schneller ein tieferes Verständnis der Algorithmen zu entwickeln.

 Kopieren

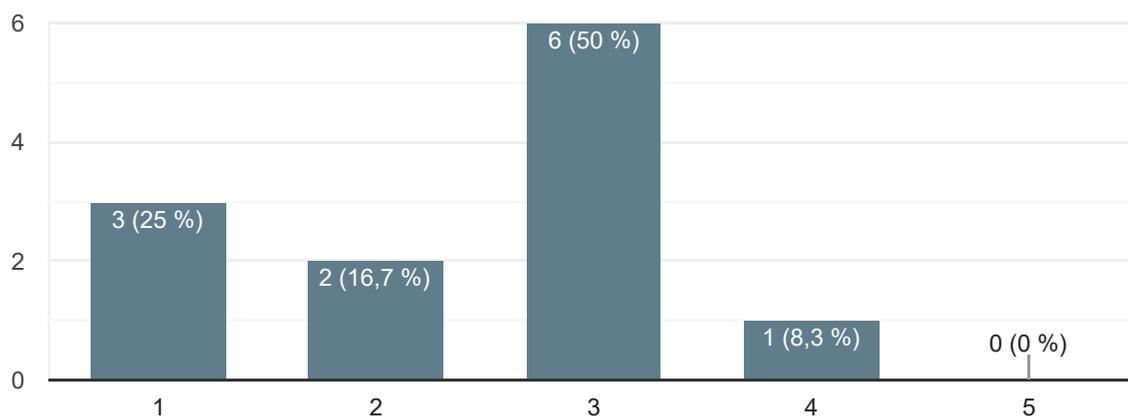
12 Antworten



Ich würde grundsätzlich eher auf eine XAI Anwendung zurückgreifen, statt mich mit wissenschaftlicher Lektüre auseinanderzusetzen.

 Kopieren

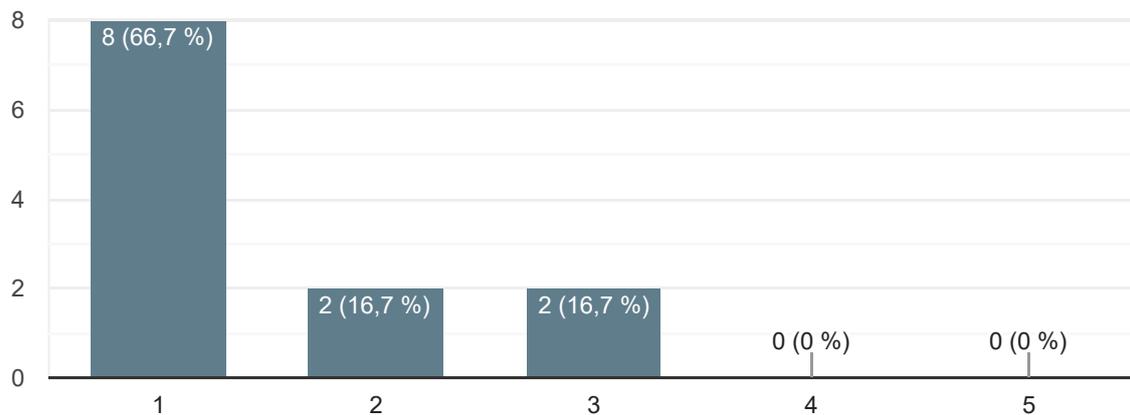
12 Antworten



Ich sehe weiteres Entwicklungspotential in der Anwendung, um die Algorithmen noch besser verständlich zu machen.

 Kopieren

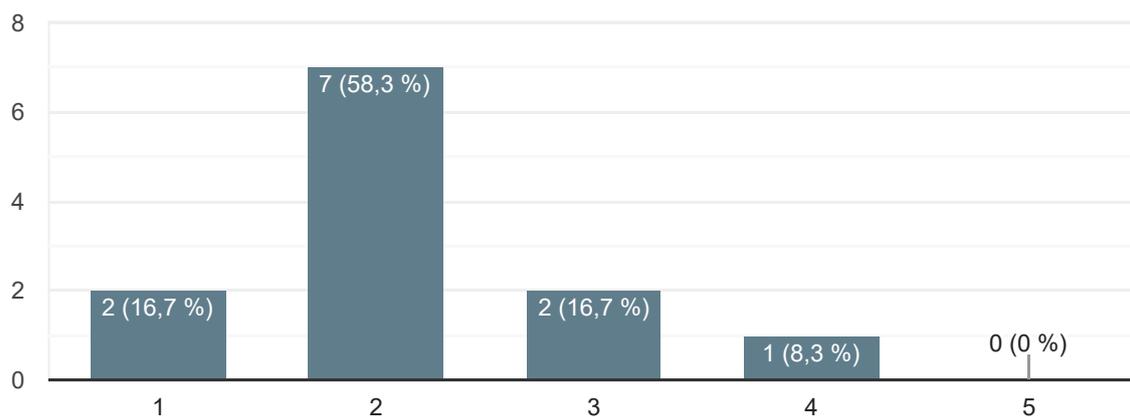
12 Antworten



Der Umfang der bereitgestellten Informationen war genau richtig, um ein umfassendes Verständnis zu ermöglichen.

 Kopieren

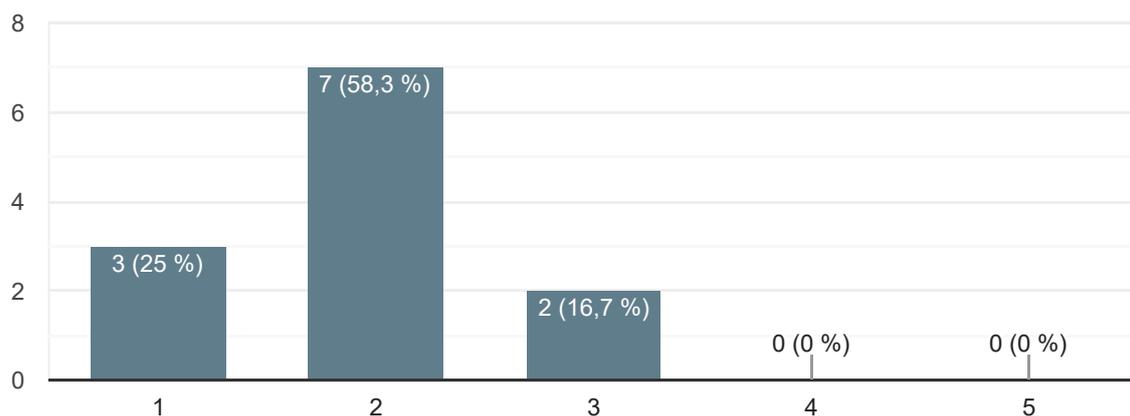
12 Antworten



Die Anwendung bot ausreichend Flexibilität in der Konfiguration und Nutzung.

 Kopieren

12 Antworten

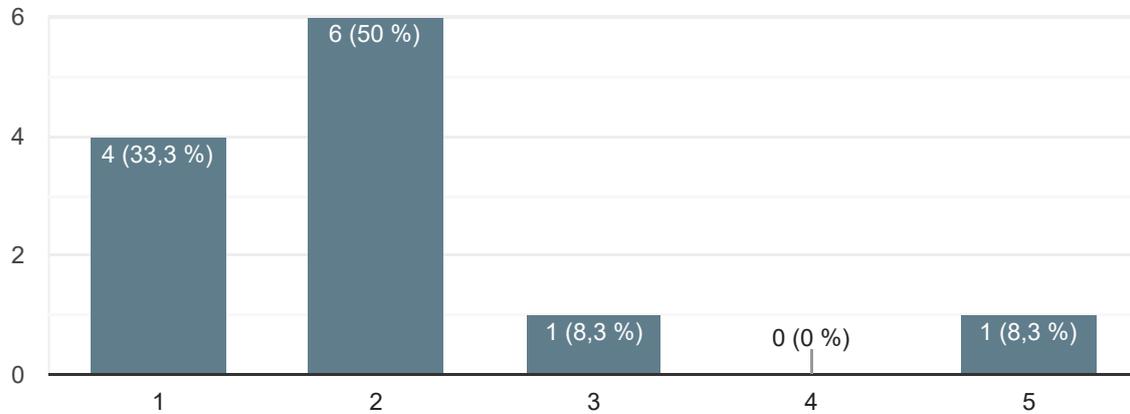


## Fragen IX

Nach der Nutzung der XAI-Anwendung fühle ich mich in der Lage, die Grundzüge von NEAT zu erklären.

 Kopieren

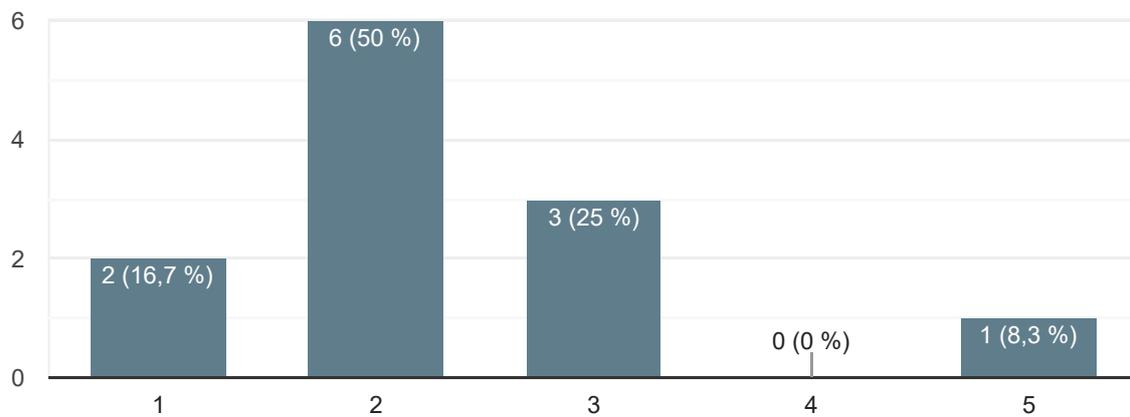
12 Antworten



Nach der Nutzung der XAI-Anwendung fühle ich mich in der Lage, die Grundzüge von HyperNEAT zu erklären.

 Kopieren

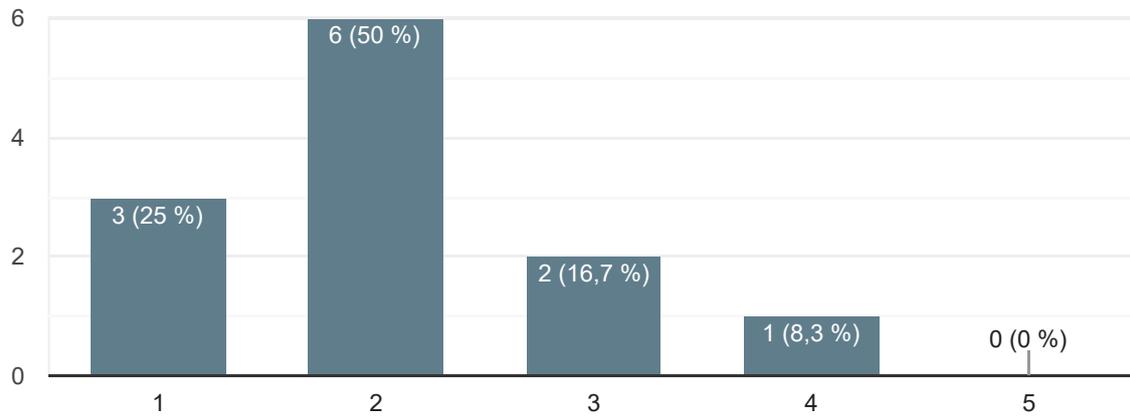
12 Antworten



Die Erklärungen haben mir geholfen, komplexe Konzepte einfacher zu verstehen.

 Kopieren

12 Antworten

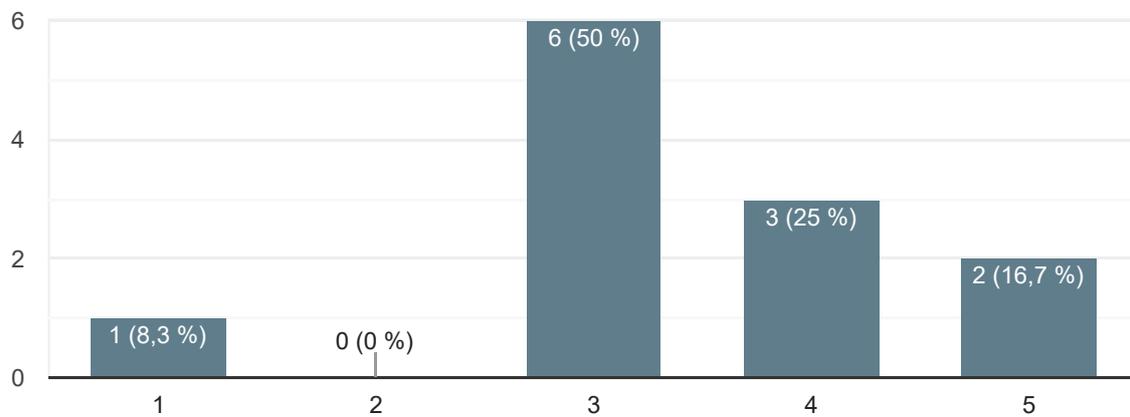


Fragen X

Ich fühle mich dazu befähigt, Fehler in der Simulation zu identifizieren.

 Kopieren

12 Antworten



Fragen XI



## Hast du noch Verbesserungsvorschläge?

6 Antworten

GUI ist sehr rudimentär, aber auch verständlich. Ich denke es gibt da Verbesserungspotential.

Ein "saubereres" Design könnte für die Übersichtlichkeit helfen.

Vielleicht das Video direkt einbinden?

Eventuell wären noch mehr interaktive Möglichkeiten, abseits der Simulation möglich? Z.B. ein Quiz am Ende jeder "Erklär"-Seite.

Ein visuelles Beispiel z.B. anhand der Acht würde es ermöglichen die Anwendung besser zu verstehen und den Unterschied zwischen Neat und Hypertext herauszustellen. Wie ist der Ordnungs- und Optimierungsprozess bei Neat vs hyperneat.

Kurze Erklärungen in Form eines Glossars an den Faktoren des Algorithmus würden helfen und als Erinnerungstütze dienen.

Einzelne Infos zu Schwellenwerten (ab wann kann ich ein System als "belastbar" ansehen) in der Anwendung selbst würden einen guten Orientierungspunkt bieten.

Die Anwendung sollte nach gängigen technischen Anforderungen gestaltet werden um eine reibungslose Inbetriebnahme zu gewährleisten.

Insgesamt ein spannendes Tool.

Inhalt:

Es könnte eine Erläuterung ergänzt werden, warum manche Parameter fest vorgegeben sind (z.B. Reproduktionsrate von 0.75 und maximale Speziesanzahl) -> Gibt es Paper, die zeigen, dass diese Zahl am besten funktioniert oder sind die Gründe technische / zeitliche Limitationen o.ä.?

Ui:

Die Trennung des "Simulation"-Headers von den Informations-Headern könnte noch deutlicher erfolgen, z.B. alle Informationsheader links und der Simulationsheader ganz rechts. In der aktuellen Reihenfolge ("Simulation" zwischen "Parameterübersicht" und "Evaluation") ist nicht klar, ob man nach "Parameterübersicht" am besten die Simulation macht, und erst danach "Evaluation", "Selektion" und Co liest oder zuerst alle Infos lesen soll.

Die Informations-Popups an den i-Symbolen sind sehr schön, verschwinden allerdings nach einer Zeit automatisch, obwohl die Maus noch auf dem i-Symbol hovert. Dies könnte User einschränken, die mehr Zeit zum Lesen benötigen.

Vielleicht die Lesbarkeit der Parameter in der Simulation zu verbessern.

Die Möglichkeit mehrere Generationen durchlaufen zu lassen oder eine geringere Generationsanzahl auszuwählen, damit wenn man wissen möchte wie das Ende aussieht, man nicht 1000 auf nächste Generation klicken muss.



## **Erklärung zur selbständigen Bearbeitung**

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

---

Ort

Datum

Unterschrift im Original