

Duale Hochschule Baden-Württemberg Mannheim

Gruppenarbeit

**Steuerung von Agenten in Videospielen basierend auf
dem genetischen Algorithmus NEAT**

Studiengang Wirtschaftsinformatik

Studienrichtung Data Science

Verfasser:	Amos Dinh, David Schäfer, Lasse Friedrich
Matrikelnummer:	5504890, 7086451, 9924680
Kurs:	WWI-21-DSA
Studiengangsleiter:	Prof. Dr.-Ing. habil. Dennis Pfisterer
Dozent:	Dr. Maximilian Scherer
Bearbeitungszeitraum:	13.11.2023 – 07.02.2023

Inhaltsverzeichnis

Abbildungsverzeichnis	ii
Tabellenverzeichnis	iii
Abstract	iv
1 Motivation	1
2 Theoretische Grundlagen	2
2.1 Grundlagen von NEAT	2
2.2 Crossover und Mutation in NEAT	3
2.3 Grundlagen von Computerspielen	5
2.4 Grundlagen zu Visualisierungen	6
3 Praktische Ausarbeitung	10
3.1 Projektbeschreibung	10
3.2 Implementierungskonzept	11
3.3 Visualisierungskonzept	11
4 Evaluation der Ergebnisse	13
4.1 Ergebnisse	13
4.2 Handlungsempfehlungen	15
Literaturverzeichnis	17

Abbildungsverzeichnis

4.1	Abbildung des Fitnessverlaufs der beiden besten Spezies	14
-----	---	----

Tabellenverzeichnis

4.1	Checkliste für funktionale und nicht funktionale Anforderungen	13
4.2	Erfüllte Checkliste für funktionale und nicht funktionale Anforderungen . .	16

Abstract

Dieses Projekt zielt auf die Implementierung und Visualisierung des genetischen Algorithmus NEAT (Neuroevolution of augmenting topologies) ab, angewendet auf das Lunar Lander-Spiel aus der Gymnasium-Bibliothek. Der Fokus liegt auf der Entwicklung eines Lehrvideos, das den Algorithmus und seine Anwendung im Spielkontext veranschaulicht. Unter Verwendung der Python-Bibliothek Manim wurde ein detailliertes Visualisierungskonzept entworfen und umgesetzt, um die Funktionsweise von NEAT eingängig zu erklären.

Im Zuge des Projekts wurde der NEAT-Algorithmus von Grund auf implementiert und in das „Lunar Lander“-Spiel integriert. Durch das vertonte Lehrvideo, das sowohl visuell ansprechend als auch informativ gestaltet wurde, konnte eine klare und verständliche Darstellung des Algorithmus erreicht werden. Das Endergebnis ist ein umfassendes Lehrmittel, das sowohl die technische Implementierung als auch die theoretischen Grundlagen von NEAT effektiv vermittelt.

1 Motivation

Die Faszination für den genetischen Fortschritt und die Anpassungsfähigkeit in der Natur bildet die Grundlage für dieses Projekt. In der biologischen Welt lösen Lebewesen durch genetische Evolution und Mutationen Aufgaben zunehmend effizienter. Diese Beobachtung weckt das Interesse an der Frage, inwieweit ähnliche Prinzipien in Form von unsupervised learning auf den Bereich der Computerwissenschaften übertragen werden könnten. Speziell die Implementierung von genetikbasiertem Lernen in Computer-Agenten, angelehnt an natürliche Vorbilder, stellt ein interessantes Forschungsfeld dar.

Die Natur bietet unzählige Beispiele für effiziente Anpassungs- und Lernprozesse, die durch genetische Variationen und natürliche Selektion getrieben werden. Diese Prozesse haben zu einer beeindruckenden Vielfalt und Spezialisierung der Arten geführt. In der Informatik könnten ähnliche Mechanismen genutzt werden, um lernfähige Systeme zu entwickeln, die sich selbstständig an neue Aufgaben und Umgebungen anpassen können. Die Herausforderung liegt darin, Konzepte der natürlichen Evolution in Algorithmen zu übersetzen, die in der Lage sind, komplexe Probleme effektiv zu lösen.

Der Ansatz, genetikbasierte Lernmethoden auf Computer-Agenten anzuwenden, bietet das Potenzial, die Grenzen herkömmlicher KI-Methoden zu überwinden. Insbesondere könnte dieser Ansatz dazu beitragen, das Problem der Überanpassung (Overfitting) zu verringern, da genetische Algorithmen eine natürliche Tendenz zur Exploration und Diversifikation aufweisen. Zudem ermöglicht die Verwendung von unsupervised learning Methoden eine flexiblere Anpassung an unbekannte oder sich verändernde Umgebungen, was in vielen realen Anwendungsfällen von großer Bedeutung ist.

Diese Arbeit zielt darauf ab, die Machbarkeit und Effizienz von genetikbasiertem Lernen in Computer-Agenten zu untersuchen. Durch die Kombination von Theorien aus der Biologie und Informatik wird versucht, ein tieferes Verständnis dafür zu entwickeln, wie maschinelles Lernen durch die Prinzipien der Evolution bereichert werden kann.

2 Theoretische Grundlagen

2.1 Grundlagen von NEAT

Historische Entwicklung genetischer Algorithmen

Die Entwicklung genetischer Algorithmen ist eng mit dem Aufkommen und der Evolution des maschinellen Lernens verknüpft. Ursprünglich ein bedeutendes Forschungsfeld, wurde die Neuroevolution in weiten Teilen durch Methoden im Bereich des Reinforcement Learning übertroffen. Der NEAT-Algorithmus wendet das Konzept der Evolution auf neuronale Netzwerke an, indem er nicht nur die Gewichte, sondern auch die Struktur der Netzwerke optimiert. Ein Vorteil dieser Methodik liegt darin, dass Lösungen für Probleme gefunden werden können, für die keine explizite Kostenfunktion formuliert werden kann. Belohnungen oder negative Kosten werden oft nur für einzelne Aktionen der Agenten vergeben, wodurch traditionelle Optimierungsverfahren, die eine Funktion optimieren, nicht anwendbar sind. Trotz der Etablierung des gradientenbasierten Deep Learning könnte die Neuroevolution aufgrund ihrer Eigenschaften, wie der einfachen und effizienten Parallelisierung, zukünftig wieder an Bedeutung gewinnen [14].

Bestandteile von NEAT

NEAT (Neuroevolution of augmenting topologies) umfasst mehrere Schlüsselkomponenten [12]:

- **Neural (N):** NEAT beginnt mit einem grundlegenden neuronalen Netz und erhöht dessen Komplexität bei Bedarf.
- **Evolution (E):** Dies beinhaltet Mechanismen wie Mutation, Crossover und Selektion zur Entwicklung der Netze.
- **of augmenting Topologies (AT):** NEAT startet mit der einfachsten Netzwerkstruktur und fügt dann Komplexität hinzu, wenn diese die Leistung der Netzwerke verbessert. Die Komplexität wird durch die Struktur der neuronalen Netze, einschließlich der Anzahl der Neuronen und Layer sowie der Verbindungen zwischen den Layern, ausgedrückt.

Jedes Genom in NEAT besteht aus einer Reihe von Knoten- und Kanten-Genen. Wesentlich ist hierbei, dass jedem neu entstehenden Gen eine global einzigartige Innovationsnummer zugewiesen wird, die den Entstehungszeitpunkt angibt und eine eindeutige Vererbung der Gene ermöglicht.

Der Algorithmus wird mit einer bestimmten Anzahl von Individuen x initialisiert, wobei jedes Individuum ein neuronales Netzwerk darstellt, definiert durch ein spezifisches Genom. Anfangs existieren nur Verbindungen zwischen Input O^m und Output Y^n . NEAT eignet sich für Simulationsaufgaben wie das Erlernen einfacher Videospiele. Dabei repräsentiert der Input-Zustandsvektor den Zustand des Agenten und seiner Umgebung. Basierend auf O berechnet das neuronale Netzwerk die Handlung Y , die dann in der Simulationsumgebung ausgeführt wird. Für bestimmte Zustände der Umgebung oder des Agenten können Belohnungen vergeben werden, die zur Fitness aufsummiert werden. Beispielsweise könnte ein Agent mit menschenähnlichen Gliedmaßen belohnt werden, wenn die Koordinate des Kopfes eine bestimmte Höhe überschreitet, wie bei der Aufgabe, aus einer Liegeposition aufzustehen. Jedes Genom wird einzeln evaluiert, und die zugehörige Fitness wird dokumentiert.

2.2 Crossover und Mutation in NEAT

Crossover-Prozess

Im Crossover-Prozess werden zwei Eltern-Genome kombiniert, um ein neues Kind-Genom zu erstellen. Eine potenzielle Mutation der Genom-Charakteristika ermöglicht die Exploration der Topologie und Gewichtswerte, um das ideale Genom zu finden. Die Auswahl der Eltern-Genome basiert dabei auf der evaluierten Fitness und wird später erläutert.

Der Crossover-Prozess konzentriert sich auf Knoten- und Kanten-Gene. Jedes Kanten-Gen hat typischerweise fünf Attribute: Innovationsnummer, Ursprungsknoten, Endknoten, Gewicht und Aktivierungsstatus. Ein Knoten wird durch zwei Attribute definiert: Innovationsnummer und Knotentyp (Sensor, Hidden, Output). Das Kind-Genom erbt die Gene der Eltern-Genome, wobei ein überlappendes Gen zufällig ausgewählt wird. Die überschüssigen Gene werden nur vom Elternteil mit der höheren Fitness übernommen. Überschüssige Gene sind solche, deren Innovationsnummer über die maximale Innovationsnummer des anderen Genoms hinausgeht.

Es gibt fünf Arten von Mutationen in NEAT, die das neu entstandene Genom durchlaufen kann: 1. Erzeugung einer neuen Verbindung mit einem Gewicht $w \sim \mathcal{P}$, wobei \mathcal{P} beispielsweise als $\mathcal{N}(0, \sigma^2)$ gewählt wird. 2. Teilen einer zufälligen bestehenden Verbindung durch Hinzufügen eines Knotens, wobei die bestehende Verbindung, der erste Teil der Verbindung, erhalten bleibt und die neue Verbindung ein Gewicht von 1 erhält. 3. Aktivieren oder Deaktivieren einer Verbindung. 4. Gewichtsverschiebung eines zufälligen Gewichts durch Addition mit einer Zahl $x \sim \mathcal{Q}$. 5. Gewichtsneusetzung durch einen neuen Wert $w \sim \mathcal{P}$.

Selektion und Speziation in NEAT

$$d(G_1, G_2) = c_1 \cdot \frac{E}{N} + c_2 \cdot \frac{D}{N} + c_3 \cdot \bar{W} \quad (2.1)$$

Bei der Speziation werden Genome basierend auf ihrer genetischen Distanz (Gleichung 2.1) in Spezies gruppiert. Hierbei sind c_1 , c_2 und c_3 Hyperparameter, E die Anzahl der überschüssigen Gene, D die Anzahl der disjunkten Gene und \bar{W} die durchschnittliche Gewichtsdistanz der überlappenden Gene der Genome. Folglich werden Genome, deren Distanz unter einem bestimmten Schwellenwert liegt, in einer Spezies gruppiert. Ist die Distanz zu groß und existiert keine passende Spezies, wird eine neue Spezies mit dem entsprechenden Genom als Basis gebildet.

Die Selektion in NEAT basiert auf dem Fitnesswert verschiedener Genome. Innerhalb der Spezies werden Genome nach ihrer Fitness sortiert und anteilig, beispielsweise die 20% mit den höchsten Fitness-Werten, für die Kreuzung zur Erstellung der nächsten Generation ausgewählt.

Die Speziation ermöglicht es, dass sich in einer Nische ähnliche Genome weiterentwickeln können, ohne sofort auszusterben, nur weil andere Genom-Variationen in der aktuellen Evaluation eine höhere Fitness erzielt haben. Die Kreuzung von ähnlichen Genomen innerhalb einer Nische führt mit höherer Wahrscheinlichkeit zu funktionierenden Kind-Genomen. Hierdurch kann strukturelle Vielfalt bewahrt werden, was die Innovation auf der Topologie-Ebene fördert und die Evolution von Netzwerken durch kleinere, inkrementelle Mutationen ermöglicht. Drastische Mutationen führen zur Bildung neuer Spezies. Die absolut erlaubte Anzahl an Nachkommen pro Spezies wird durch die relative Fitness der Spezies festgelegt. Somit stirbt eine nicht-performante Spezies nicht sofort aus, sondern erhält lediglich weniger Nachkommen zugewiesen.

Der Kreislauf aus Fitness-Evaluation, Crossover, Mutation und Speziation wird über mehrere Generationen hinweg wiederholt, bis sich ein Genom mit gewünschter Performance entwickelt.

2.3 Grundlagen von Computerspielen

Historische Bedeutung und Entwicklung

Die historische Entwicklung von Computerspielen ist eng mit technologischen Fortschritten und gesellschaftlichen Veränderungen verknüpft. Ihren Ursprung finden Computerspiele in den frühen Experimenten an Universitäten in den 1950er Jahren, die sich zu einer bedeutenden Unterhaltungsform in der digitalen Ära entwickelt haben. Ursprünglich Teil von wissenschaftlichen Versuchen, wurden diese Spiele schnell zu einem kulturellen Phänomen, das von militärischer Simulation bis zur familiären Unterhaltung reicht. Die gesellschaftliche Durchdringung von Computerspielen spiegelt ihre Bedeutung in der Analyse menschlichen Verhaltens wider [9].

Charakteristika von Computerspielen

Computerspiele zeichnen sich durch mehrere grundlegende Charakteristika aus. Sie bieten eine interaktive Umgebung, in der Spieler auf variierende Herausforderungen und Szenarien reagieren. Die Spiele definieren spezifische Ziele, sei es das Erreichen eines Highscores oder das Überleben in einer virtuellen Welt. Wettbewerb und Kooperation sind zentrale Aspekte, wobei Spieler oft gegen computergesteuerte Gegner oder in einem Mehrspielermodus antreten. Die Steuerung von Agenten, sei es in Form von Charakteren oder Objekten, in einer meist simplifizierten Welt ist ein weiteres Merkmal. Die visuelle Gestaltung der Spiele reicht von einfachen Grafiken bis zu komplexen 3D-Welten, wobei die visuelle Qualität nicht immer ausschlaggebend für den Erfolg eines Spiels ist.

Rolle der Visualisierung

Die Visualisierung in Computerspielen geht über die ästhetische Komponente hinaus und ist ein integraler Bestandteil des Spielerlebnisses. Sie ermöglicht es, komplexe Welten darzustellen, Geschichten zu erzählen und emotionale Reaktionen zu evozieren. Mit der Weiterentwicklung der Grafiktechnologien haben sich die Darstellungsmöglichkeiten in Spielen erweitert und reichen von einfachen Pixelgrafiken bis zu realistischen 3D-Umgebungen [13].

Zukunftsperspektiven

Die Zukunft von Computerspielen ist vielversprechend und wird durch die Entwicklung neuer Technologien wie Virtual und Augmented Reality, Cloud-Gaming und künstliche Intelligenz geprägt. Diese Entwicklungen deuten auf eine Ära hin, in der die Grenzen zwischen Realität und virtueller Welt zunehmend verschwimmen und neue Formen des Spielens möglich werden.

2.4 Grundlagen zu Visualisierungen

Geschichte der Visualisierung

Die Geschichte der Visualisierung begann in der zweiten Hälfte des 19. Jahrhunderts, als technologische Fortschritte, insbesondere in der Fotografie, neue Möglichkeiten eröffneten. Diese Entwicklung wird oft als ein wichtiger Wendepunkt in der visuellen Darstellung gesehen. Im Laufe der Zeit entwickelte sich auch die Fähigkeit, Bilder und visuelle Inhalte zu verstehen und zu interpretieren, bekannt als visuelle Alphabetisierung. Diese Fähigkeit ist nicht auf einen bestimmten Forschungsbereich beschränkt und hat kein festes Ziel. Sie ist vielmehr wichtig, um zu verstehen, wie wir visuelle Informationen wahrnehmen und wie diese in unserem sozialen Umfeld funktionieren [11].

Bereits 1969 wies John Deeb auf die Bedeutung der visuellen Alphabetisierung hin. Er betonte die Wichtigkeit, visuelle Darstellungen und den Prozess des Sehens zu studieren, und forderte eine Zusammenarbeit über verschiedene Disziplinen hinweg, um die visuelle Kultur besser zu verstehen. Die visuelle Alphabetisierung hat viele verschiedene Definitionen, die je nach Fachgebiet variieren. Die International Visual Literacy Association definiert sie als einen interdisziplinären Ansatz, um den Prozess der visuellen Kommunikation zu verstehen und die dafür notwendigen Kenntnisse, Fähigkeiten und Kompetenzen zu erlernen [16].

Visuelle Wahrnehmung

Visuelle Wahrnehmung und die Verarbeitung visueller Informationen sind wichtige Elemente im Bildungsbereich. Es geht dabei nicht nur um das Verstehen wissenschaftlicher Konzepte, sondern auch um die Fähigkeit, diese Konzepte in neuen Zusammenhängen anzuwenden. Visualisierungen spielen eine große Rolle bei kognitiven Aktivitäten und beim Lernen. Sie ermöglichen es, wissenschaftliche Inhalte auf eine neue Weise zu betrachten und zu verstehen [16].

Visualisierung wird oft auch als „mentales Bild“ oder „mentale Darstellung“ bezeichnet. Diese mentalen Bilder helfen Schülern, komplexe wissenschaftliche Konzepte besser zu erfassen und zu verinnerlichen. Die grafische Darstellung, also die Verwendung von Bildern und Diagrammen, ist ein wichtiger Schritt, um diese Konzepte zugänglich und verständlich zu machen. Sie erleichtert das Verständnis verschiedener Wahrnehmungen der Realität und fördert den Lernprozess [3].

Die Visualisierung soll gesprochene Worte nicht immer ersetzen, sondern vielmehr darauf abzielen, die Aufmerksamkeit der Zuhörenden auf das Wesentliche des Dargebotenen zu lenken und das Verständnis der präsentierten Informationen zu verbessern, sowie den Zugang zum Kern der Inhalte zu erleichtern [6].

Kreativität ist der Schlüssel zur Visualisierung und steigert die Effizienz erheblich. Sie sollte sich vor allem auf drei Aspekte konzentrieren:

- Die Planung der Visualisierung.
- Die Grundpunkte der geplanten Visualisierung.
- Die Kompositionsregeln der Visualisierung.

Bei der Planung sind vor allem Antworten auf die folgenden Fragen wichtig:

- Was möchte ich darstellen?
- Was ist das Ziel der vorbereiteten Präsentation?
- Wen möchte ich erreichen oder überzeugen?

Visuelle Materialien als Bildungswerkzeug

Bildmaterialien dienen als multifunktionale Instrumente im Lernprozess, die komplexe Sachverhalte visuell veranschaulichen. Sie lassen sich in Kategorien wie direkte Darstellungen – Fotos, Bilder, Diagramme – und dynamische Medien – Diashows, Filme – einteilen. Während Fotos die Realität detailgenau wiedergeben können, besteht die Gefahr, dass sie mit irrelevanten Details überladen und somit ablenkend wirken [8].

In Bildungsmaterialien eingesetzte Bilder erfüllen unterschiedliche Funktionen:

- Dekorative Bilder steigern die ästhetische Anziehungskraft und füllen leere Flächen.
- Darstellende Bilder illustrieren Inhalte direkt und unterstützen die Informationsvermittlung.

- Organisierende Bilder tragen zur Strukturierung komplexer Themen bei.
- Interpretierende Bilder liefern Erklärungen und fördern das Verständnis der Lerninhalte.
- Transformierende Bilder ändern die Betrachtungsweise und vertiefen das Verständnis.

Symbole spielen eine signifikante Rolle, indem sie Objekte und Phänomene kennzeichnen und die Kommunikation über Sprachgrenzen hinweg erleichtern.

Die Zielsetzung und die anvisierte Zielgruppe sind bei der Auswahl und Gestaltung von Bildmaterialien wesentlich. Eine strategische Auswahl kann die Verständlichkeit und Erinnerung der Lerninhalte verbessern [1].

Visuelle Informationen variieren weitgehend und umfassen:

- Informativ: Klare und eindeutige Darstellungen wie Piktogramme.
- Wissenschaftlich: Fachspezifische Bilder, darunter Röntgenbilder oder geowissenschaftliche Visualisierungen.
- Medial: Bilder aus Werbung und Film, die emotionale Reaktionen hervorrufen oder überzeugen sollen.

Die Kategorisierung unterstützt die Erkennung der Vielfalt und des Einflusses von Bildmaterialien im Lernkontext und fördert einen bewussten Einsatz zur Bereicherung und Vertiefung des Lernprozesses.

Elemente der Visualisierung

Elemente der Visualisierung sind wesentliche Werkzeuge in Lernumgebungen, die Inhalte auf eine klar strukturierte Weise kommunizieren [1]. Bei der Erstellung von Visualisierungen sollten folgende Punkte beachtet werden:

- Die Auswahl der Präsentationsmedien und Gestaltungselemente muss den Inhalt adäquat reflektieren.
- Lesbarkeit ist entscheidend, wobei einfache Schriftarten wie Arial oder Calibri zu bevorzugen sind.
- Die kulturellen Lesegewohnheiten sind zu respektieren, insbesondere das Lesen von links nach rechts sowie die korrekte Anwendung von Groß- und Kleinschreibung.

- Das Layout und die Ausrichtung des Textes, einschließlich Textumrandungen und Hervorhebungen, verbessern die Übersichtlichkeit.
- Der Einsatz von Farben sollte wohlüberlegt sein, um Kontraste zu schaffen und symbolische Bedeutungen zu nutzen.

Die korrekte Interpretation der Daten in diesen Diagrammen ist für das Verständnis von wissenschaftlichen Zusammenhängen unerlässlich. Schwierigkeiten, wie die Unterscheidung zwischen x- und y-Achse, müssen adressiert werden [7].

Die Anordnung der Präsentationselemente folgt einer spezifischen Komposition, die auf Symmetrie, Übertragung, Rhythmus und Dynamik basiert. Vorlagen können hierbei als Hilfsmittel dienen, um ein konsistentes und logisches Layout zu gewährleisten.

In der Gestaltung von Lernmaterialien durch Computerpräsentationen wird eine Balance zwischen Informationsvermittlung, Motivation und Anpassung an individuelle Lerngeschwindigkeiten angestrebt [15].

Zukunftsperspektiven

Die Zukunft der Visualisierung deutet auf eine verstärkte Integration von virtueller und physischer Realität hin, die neue Wege für immersive Lern- und Erfahrungsräume eröffnet. Diese technologischen Fortschritte versprechen eine tiefere Einbindung und Verständnisförderung in Bildung und Präsentation [8].

3 Praktische Ausarbeitung

3.1 Projektbeschreibung

Das Ziel dieses Projekts, inspiriert durch die Faszination für die Modellierung natürlicher Ereignisse in computergesteuerten Systemen, ist die Entwicklung eines Agenten, basierend auf dem NEAT-Algorithmus, um ein Computerspiel zu meistern.

Für das Projekt wurde ein Standardspiel aus der Gymnasium (gym) Bibliothek ausgewählt [4]. Gym, ein Open-Source-Python-Paket entwickelt von OpenAI, bietet eine Sammlung standardisierter Umgebungen zur Entwicklung und zum Testen von Algorithmen im Bereich des Reinforcement Learning (RL). Es erleichtert die Entwicklung und Vergleichbarkeit von RL-Algorithmen durch eine einheitliche Schnittstelle für unterschiedliche Umgebungen. Der NEAT-Algorithmus eignet sich zur Bewältigung von RL-Aufgaben, weshalb diese Bibliothek Anwendung findet. Gym umfasst zahlreiche Simulationsumgebungen, einschließlich aller Atari-Spiele.

Das ausgewählte Spiel, „Lunar Lander“, zeichnet sich durch ein einfaches Ziel und einfache Steuerungselemente aus. Der Inhalt des Spiels besteht darin, einen Mondlander sicher und waagrecht in einem markierten Bereich auf der Mondoberfläche zu landen. Die Leistungskriterien umfassen den waagrechten Anflug, den Treibstoffverbrauch und ob der Lander in der richtigen Orientierung sicher im Zielbereich landet. Zur Steuerung des Landers werden zwei lineare Signale a und b verwendet, um die Leistung des Haupt- und der Seitentriebwerke zu regulieren. Das Haupttriebwerk wird bei $a > 0,5$ aktiviert, wobei die Intensität bis $a = 1$ proportional zunimmt. Ähnlich feuern das linke oder rechte Triebwerk bei $-1 \leq b < -0,5$ bzw. $1 \geq b > 0,5$.

Das technische Ziel des Projekts ist es, eine Konfiguration für ein neuronales Netz zu finden, sodass basierend auf der relativen Position des Landers die Triebwerke so gesteuert werden, dass eine Landung möglichst waagrecht, mit geringem Treibstoffverbrauch und innerhalb des Zielbereichs erfolgt.

Anschließend wird ein Video konzipiert und erstellt, das den NEAT-Algorithmus erklärt und am Beispiel des Lunar Landers veranschaulicht.

3.2 Implementierungskonzept

Für die Realisierung dieses Projektes wurde die Programmiersprache Python ausgewählt. In Python wurden sowohl die einzelnen Komponenten des NEAT-Algorithmus als auch dessen Integration mit dem ausgewählten Spiel umgesetzt. Die benötigte Simulationsumgebung für das Spiel wird durch die Bibliothek Gymnasium bereitgestellt [4].

Als funktionale Anforderungen wurden folgende Punkte festgelegt:

- Der NEAT-Algorithmus soll basierend auf dem Whitepaper implementiert und getestet werden.
- Der Lander soll ausschließlich durch einen Agenten gesteuert werden, der auf dem NEAT-Algorithmus basiert.
- Alle Aktionen des Landers sollen aus den Beobachtungsvariablen des 8-dimensionalen Inputvektors abgeleitet werden.
- Der finale Agent soll als Lösung mindestens 200 Punkte erreichen [5].

Als nicht-funktionale Anforderung wurde festgelegt:

- Das Programm zur Ermittlung eines optimalen Landers soll einfach und replizierbar sein.

Implementierungsdetails

Die Implementierung des NEAT-Algorithmus folgt der ursprünglichen Beschreibung [12]. An einigen Stellen wurde die Vorgehensweise der existierenden Bibliothek Python NEAT [10] als Orientierung genutzt, beispielsweise bei der zufälligen Generierung von Gewichtswerten mittels Normalverteilung. Diese Aspekte sind in der originalen Beschreibung nicht ausführlich erläutert, sondern nur im dazugehörigen C++-Code zu finden.

3.3 Visualisierungskonzept

Das Visualisierungskonzept dieses Projektes basiert auf den Vorgaben des Kurses. Es wurde Python in Kombination mit Manim verwendet, um das Konzept ansprechend in Videoform zu animieren und darzustellen. Die Vertonung erfolgte durch das Text-to-Speech-Programm

ElevenLabs, und der Schnitt sowie die Videobearbeitung wurden mit ClipChamp durchgeführt.

Die Anforderungen an diesen Projektteil unterteilen sich in funktionale und nicht-funktionale Kriterien:

Funktionale Anforderungen:

- Animation und Rendering des Videos mit Python und Manim.
- Die Videolänge überschreitet 15 Minuten nicht.
- Vollständige Vertonung des Videos.

Nicht-funktionale Anforderungen:

- Ansprechende Animation.
- Vollständige und verständliche Darstellung des NEAT-Konzepts.
- Berücksichtigung von Visualisierungsgrundlagen.
- Ein klar erkennbarer roter Faden.

Bei der Umsetzung war es wichtig, die erarbeiteten Visualisierungsgrundlagen effektiv einzusetzen. Ziel war es, den NEAT-Algorithmus mittels natürlicher Phänomene insbesondere für technisch versierte MINT-Studenten verständlich und inspirierend zu erklären.

In der visuellen Gestaltung lag der Fokus auf darstellenden und organisierenden Bildern, um den Informationsfluss zu strukturieren. Interpretierende Bilder kamen zum Einsatz, um komplexe Theorien zu vermitteln und die Aufmerksamkeit der Zuschauer zu gewinnen. Als Medium wurde das Video gewählt, wobei die Lesbarkeit durch die Schriftart Montserrat und helle Schrift auf dunklem Hintergrund unterstützt wurde. Ergänzt wurde dies durch Umrandungen, Hervorhebungen und gezielten Farbeinsatz. Die Präsentation der Daten erfolgte hauptsächlich über Diagramme, welche die neuronalen Netze abbildeten.

Die Planung des Projekts begann mit einem skizzierten Storyboard, das von allen Gruppenmitgliedern diskutiert und verfeinert wurde. Nach der Finalisierung erfolgte die schrittweise Umsetzung und Animation. Die Vertonung wurde abschnittsweise durchgeführt.

4 Evaluation der Ergebnisse

4.1 Ergebnisse

	Anforderungen	Erfüllt
Visualisierung	Das Video wurde durch Python und Manim animiert und gerendert	<input type="checkbox"/>
	Die Länge des Videos überschreitet nicht 15 Minuten	<input type="checkbox"/>
	Das Video ist vollständig vertont	<input type="checkbox"/>
	Das Konzept von NEAT soll vollständig und verständlich sein	<input type="checkbox"/>
	Visualisierungsgrundlagen sollen berücksichtigt werden	<input type="checkbox"/>
	Ein roter Faden sollte erkennbar sein	<input type="checkbox"/>
	Das Video soll ansprechend animiert sein	<input type="checkbox"/>
Implementierung	NEAT Algorithmus implementieren und testen	<input type="checkbox"/>
	Lander durch NEAT Algorithmus steuern	<input type="checkbox"/>
	Aktionen aus 8-dimensionalen Input Vektoren ableiten	<input type="checkbox"/>
	Agent soll mindestens 200 Punkte erreichen	<input type="checkbox"/>
	Programm soll replizierbar sein	<input type="checkbox"/>

Tabelle 4.1: Checkliste für funktionale und nicht funktionale Anforderungen

Um das Projekt sachgemäß zu evaluieren, wird im Folgenden die Liste der Anforderungen auf ihre Erfüllung hin überprüft.

Die Bewertung der Visualisierung basiert auf dem eingereichten Video. Aus diesem lässt sich erkennen, dass das gesamte Video im Manim-Stil erstellt wurde. Der zugehörige Quellcode ist in einem GitHub-Repository dokumentiert [2]. Die Länge des Videos bleibt unter 15 Minuten, und es ist vollständig vertont. Basierend auf dem Feedback von Kommilitonen wurde das Konzept von NEAT verständlich vermittelt. Dabei hat der Ersteller des Videos aktiv auf wichtige Visualisierungsgrundlagen, wie Farbkomposition und Lesbarkeit, geachtet. Die durchgängige Struktur und Verständlichkeit wird durch das zuvor angefertigte Storyboard sichergestellt. Das Video wurde als angenehm anzusehen bewertet, was größtenteils auf die Qualität der Animationen zurückzuführen ist.

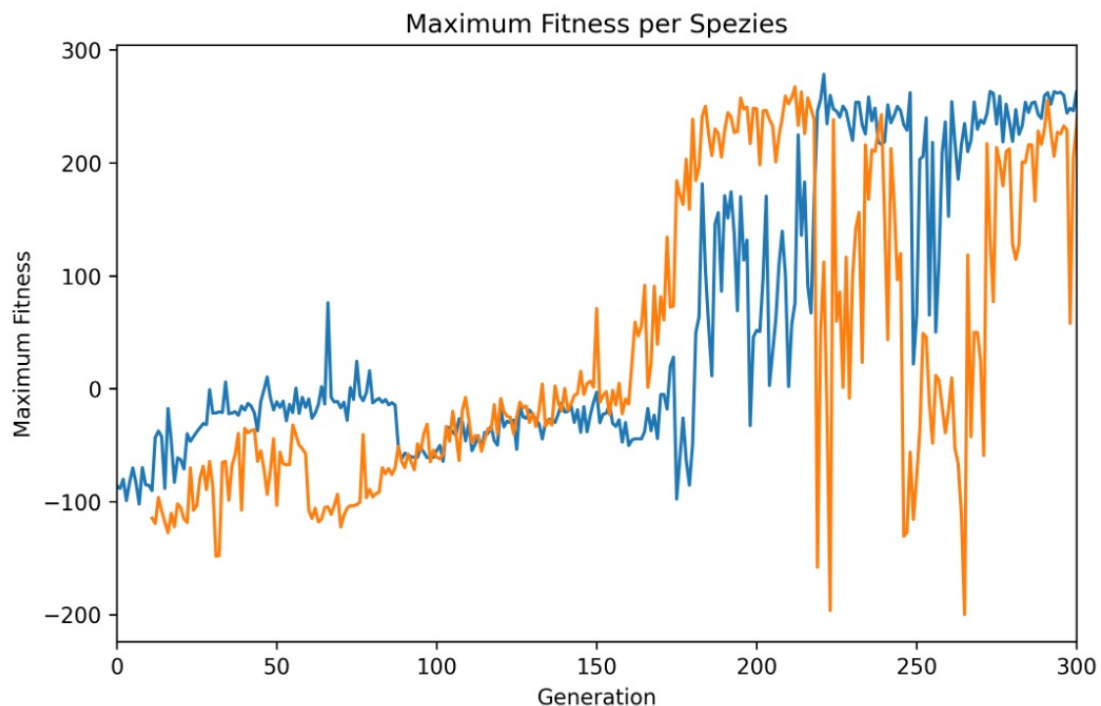


Abbildung 4.1: Abbildung des Fitnessverlaufs der beiden besten Spezies

Die Evaluierung der Implementierung basiert, wie aus der Grafik ersichtlich, auf der maximalen Fitness, die von einer Spezies erreicht wurde. Eine Spezies, die eine Fitness von 200 erreicht, gilt dabei als Lösung. Die Fitness beruht auf einer Metrik, die eine Kombination aus Geschwindigkeit, Triebwerksauslastung, Rotation und Nähe zum Ziel darstellt [5]. Die dargestellten Grafiken zeigen die beiden besten Spezies mit ihren entsprechenden Fitness-Leveln über mehrere Generationen hinweg. Es ist erkennbar, dass der Score von 200 überschritten wird, jedoch eine hohe Fluktuation in der Leistung besteht. Dies könnte auf das Konzept der Spezies zurückzuführen sein, die auch nicht optimale Netzwerke

beinhalten können. Das Gesamtergebnis wurde ausschließlich durch NEAT-basiertes Lernen auf den Input-Vektoren erzielt, welches von Grund auf in einer strukturierten Form in Python implementiert wurde. Verschiedene Spezies haben zur Zielerreichung unterschiedliche Strategien angewandt. Der Entwicklungsprozess reichte von zufälliger Steuerung über eine „Kopf-zuerst“-Strategie bis hin zur finalen Version, die sich zuerst fallen lässt und dann relativ spät die Triebwerke zur Abfederung aktiviert und ins Ziel fliegt. Es wurde ebenfalls festgestellt, dass die kontinuierliche Aktivierung der Triebwerke bessere Ergebnisse erbrachte als die absolute Aktivierung. Während die Anfangsversion das Ziel nie erreichte, konnte die finale Spezies durch NEAT-basiertes Lernen und Selektion die Rakete jedes Mal landen.

Basierend auf diesen Ergebnissen kann die Evaluation als komplett angesehen werden. Alle festgelegten technischen und nicht-technischen Anforderungen konnten umgesetzt werden, sodass das Projekt vollständig nach den Vorgaben implementiert wurde.

4.2 Handlungsempfehlungen

In diesem Beispiel konnte NEAT sehr gute Ergebnisse erzielen und die Anforderungen des Spiels meistern. Der Algorithmus fand eine sehr gute Lösung und konnte die Eingabewerte entsprechend generalisieren und verwenden. Vermutlich hätte ein Deep Learning- oder Reinforcement Learning-Ansatz dies ebenso geschafft, allerdings mit einer wesentlich einfacheren und generalistischeren Implementierung. Daher stellt sich die Frage nach der Zukunft der genetischen Algorithmen. Es scheint, dass NEAT für bestimmte Zwecke sehr gut funktioniert, jedoch eine relativ strikte Struktur voraussetzt. Ein weiteres Problem ist die hohe Fitness-Volatilität zwischen den Epochen, die das Lernen in späteren Phasen verlangsamen kann. Für einen allgemeineren Ansatz hat sich nach unserer Einschätzung Deep Learning als Favorit herauskristallisiert, da es einen generalistischen Ansatz für Lernprobleme verfolgt. In diesem Bereich gibt es zudem erhebliches Forschungspotential, das noch ausgeschöpft werden kann. Eine Kombination von auf natürlichen Prinzipien basierender Selektion während des Deep Learning-Prozesses ist allerdings eine denkbare Möglichkeit.

	Anforderungen	Erfüllt
Visualisierung	Das Video wurde durch Python und Manim animiert und gerendert	✓
	Die Länge des Videos überschreitet nicht 15 Minuten	✓
	Das Video ist vollständig vertont	✓
	Das Konzept von NEAT soll vollständig und verständlich sein	✓
	Visualisierungsgrundlagen sollen berücksichtigt werden	✓
	Ein roter Faden sollte erkennbar sein	✓
	Das Video soll ansprechend animiert sein	✓
Implementierung	NEAT Algorithmus implementieren und testen	✓
	Lander durch NEAT Algorithmus steuern	✓
	Aktionen aus 8-dimensionalen Input Vektoren ableiten	✓
	Agent soll mindestens 200 Punkte erreichen	✓
	Programm soll einfach und replizierbar sein	✓

Tabelle 4.2: Erfüllte Checkliste für funktionale und nicht funktionale Anforderungen

Literaturverzeichnis

- [1] M. Bílek et al. *Selected aspects of the visualization science curriculum subjects*. Hradec Králové: M und V Hradec Králové, 2007, S. 180–190. ISBN: 80-86771-21-0.
- [2] Amos Dinh, David Schaefer und Lasse Friedrich. *Impossible-Game GitHub*. Accessed: 30.12.2023. URL: <https://github.com/LasseFrie/Impossible-Game>.
- [3] R. Duval. „Representation, Vision and Visualization: Cognitive Functions in Mathematical Thinking“. In: *Proceedings of the Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education*. Mexico, 1999, S. 1–25.
- [4] Faram Foundation. *Gymnasium Bibliothek*. Accessed: 30.12.2023. URL: <https://github.com/Farama-Foundation/Gymnasium>.
- [5] Faram Foundation. *Gymnasium Bibliothek*. Accessed: 30.12.2023. URL: https://gymnasium.farama.org/environments/box2d/lunar_lander/#lunar-lander.
- [6] J. Gilbert. *Visualisation: a metacognitive skill in science and science education*. Visualisation in Science Education, 2005, S. 1–27. ISBN: 978-1-4020-3613-2.
- [7] M. Happonen und M. Aksela. „Graphs in upper secondary school chemistry education“. In: *ESERA 2011 Teaching and learning science*. Lyon, France, 2011, S. 77–81. ISBN: 978-9963-700-44-8.
- [8] V. Jan. *The Importance of Visualisation in Education*. University of Ostrava. Fr. Šrámka 3, Ostrava-Mariánské Hory, 709 00, Czech Republic, 2010. URL: <https://depot.ceon.pl/bitstream/handle/123456789/14480/36%20The%20Importance%20of%20Visualisation.pdf?sequence=1>.
- [9] H. Johan. *Homo Ludens, Vom Ursprung der Kultur im Spiel*. Rowohlt Taschenbuch Verlag, 1939. ISBN: 9783499554353.
- [10] Python NEAT. *Python NEAT*. Accessed: 30.12.2023. URL: <https://neat-python.readthedocs.io/en/latest/>.
- [11] M. Šimůnek. *Several suggestions for developing visual literacy*. Media Pedagogy. Media and communication in theory and education. Czech Budejovice, University of South Bohemia in Czech Budejovice. 2009.

- [12] Kenneth O. Stanley und Risto Miikkulainen. „Evolving Neural Networks through Augmenting Topologies“. In: *Evolutionary Computation* 10.2 (Juni 2002), S. 99–127. ISSN: 1063-6560. DOI: 10.1162/106365602320169811. eprint: <https://direct.mit.edu/evco/article-pdf/10/2/99/1493254/106365602320169811.pdf>. URL: <https://doi.org/10.1162/106365602320169811>.
- [13] S. Stephan. *Die Geschichte der Videospiele und Heimcomputer*. Accessed: 30.12.2023. URL: <https://8bit-museum.de/impressum/>.
- [14] Felipe Petroski Such et al. „Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning“. In: *ArXiv abs/1712.06567* (2017). URL: <https://api.semanticscholar.org/CorpusID:5044808>.
- [15] J. Veřmiřovský und M. Bílek. „Opinions of chemistry teachers on ways of using MS PowerPoint and multimedia objects“. In: *XX. International Seminar about Chemistry education - Current trends in ICT Chemistry education*. Hradec Králové: Gaudeamus, 2010, S. 16–20.
- [16] E. Wiebe, A. Clark und E. Haase. „Scientific Visualisation: Linking Science and Technology Education through Graphic Communication“. In: *Journal of Design and Technology Education* 6.1 (2001). ISSN: 1360-1431.