##### 

**Vorlage für die DIPLOMARBEIT**

**an der HTL Anichstraße**

**Version: 1.4**

**Gültig ab: 31.07.2021**

**Verteilermethode: schulintern – moodle**

**Dokumentenstatus: freigegeben**

**Es darf nur mit aktuellen Originaldokumenten gearbeitet werden.**

**Diese Seite und die folgenden 2 Seiten dienen der Dokumentenlenkung und müssen in der Diplomarbeit gelöscht werden.**

**Änderungshistorie:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Datum** | **Änderungsgrund** | | **Änderungsstand** | | **Verantwortliche** |
| **28.11.18** | **Rechtschreib- und Formatierungsfehler** | | **V1.1** | | **YH** |
| **11.03.19** | **Inhaltsverzeichnis an V1.0 angepasst** | | **V1.2** | | **YH** |
| **21.03.20** | **Seite 4 angepasst Kopfzeile** | | **V1.3** | | **YH** |
| **15.07.21** | **Neues Logo** | | **V1.4** | | **YH** |
| Erstellt: | | | geprüft | | Freigegeben: | | |
| 15.07.21, YH | | | 15.07.21, YH | | 15.07.21, YH | | |
| Datum, Ersteller | | | Datum, Qualitätsbeauftragte | | Datum, Direktion | | |

Information zur nachfolgenden Dokumentation der Diplomarbeit

Sehr geehrte Diplomandinnen, sehr geehrte Diplomanten,

alle Abteilungen der HTL Anichstraße haben sich entschlossen, eine einheitliche Dokumentation der Diplomarbeiten einzuführen.

Dafür wurde nachfolgende Vorlage erarbeitet, die am Moodle Server zur Verfügung gestellt wird.

Die blau geschriebenen Texte sind zur Unterstützung gedacht und müssen in der Diplomarbeit auf die Standardtextfarbe (schwarz) umgestellt werden. Das Inhaltsverzeichnis ist bindend, kann aber natürlich mit Unterpunkten beliebig erweitert werden. Optionale Teile können entfallen.

Die Vorlage ist verpflichtend zu verwenden!

**Zusatzinformationen:**

**Wie zitiere ist?**

Es gibt unterschiedliche Vorgaben für ein Zitat, die HTL Anichstraße bezieht sich auf die Regeln der Diplomarbeitswebseite des Bildungsministeriums und diese Zitierregeln sind anzuwenden:

<http://www.diplomarbeiten-bbs.at/hinweise-zum-wissenschaftlichen-arbeiten/zitation-plagiate>

**FAQ**

Auf der vom Ministerium ausgearbeiteten Seite sind sehr viele Fragen zur und über die Diplomarbeit sehr gut beantwortet. Bei Unklarheiten wenden Sie sich aber gerne an die Betreuerin, den Betreuer oder Abteilungsvorstand.

<http://www.diplomarbeiten-bbs.at/faq>

**Gendern**

Aus Gründen der besseren Lesbarkeit wird in dieser Diplomarbeit die Sprachform des generischen Maskulinums angewendet. Es wird darauf hingewiesen, dass die ausschließliche Verwendung der männlichen Form geschlechtsunabhängig verstanden werden soll.

**Sperrvermerk**

Der Sperrvermerk wird dann ins Dokument eingebunden, wenn die Auftraggeberin, der Auftraggeber dies aus verständlichen Gründen vorgibt. WENN ein Sperrvermerk notwendig ist, dann werden die Seiten, die es betrifft in der Bibliotheksversion NICHT in die DA eingebunden (Vorgabe der HTL Anichstraße), die Seiten bleiben LEER, damit beugen wir vor, dass ein gesperrtes Exemplar versehentlich öffentlich ausgegeben werden kann. Die Betreuerin, der Betreuer benötigen für die Notenfindung ein vollständiges Dokument.

Wir möchten darauf hinweisen, dass die Präsentation trotz Sperrvermerk ÖFFENTLICH ist!

**DIPLOMARBEIT**

**TONE**

**Höhere Technische Bundeslehr- und Versuchsanstalt Anichstraße**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Elektronik**

**Elektronik und Informatik**

|  |  |  |
| --- | --- | --- |
| Ausgeführt im Schuljahr 2021/22 von: |  | Betreuer/Betreuerin: |
| Crispin Holleis  Manolo Jaschke  Nils Kuhn |  | Dipl.-Ing. Markus Signitzer |

Projektpartner: -

Innsbruck, am TT.MM.JJJJ

Abgabevermerk: Betreuer/in:

Datum:

**SPERRVERMERK**

Auf Wunsch der Firma Vollständige Bezeichnung der Firma ist die vorliegende Diplomarbeit

für die Dauer von drei / fünf / sieben Jahren

für die öffentliche Nutzung zu sperren.

Veröffentlichung, Vervielfältigung und Einsichtnahme sind ohne ausdrückliche Genehmigung der Firma \*\*\* und der Verfasser

bis zum TT.MM.JJJJ nicht gestattet.

Innsbruck, TT.MM.JJJJ

Verfasser:

Vor- und Zuname Unterschrift

Vor- und Zuname Unterschrift

Firma:

Firmenstempel

## Kurzfassung /Abstract

## Projektergebnis

**Inhaltsverzeichnis**

[Kurzfassung /Abstract i](#_Toc96972235)

[Projektergebnis ii](#_Toc96972236)

[1 Einleitung 1](#_Toc96972237)

[2 Vertiefende Aufgabenstellung 1](#_Toc96972238)

[2.1 Crispin Holleis 1](#_Toc96972239)

[2.2 Manolo Jaschke 1](#_Toc96972240)

[2.3 Nils Kuhn 1](#_Toc96972241)

[3 Dokumentation der Arbeit 2](#_Toc96972242)

[3.1 Grundkonzept 2](#_Toc96972243)

[3.2 Theoretische Grundlagen 2](#_Toc96972244)

[3.2.1 …. 2](#_Toc96972245)

[3.3 Praktische Umsetzung 2](#_Toc96972246)

[4 Erklärung der Eigenständigkeit der Arbeit 3](#_Toc96972247)

[I. Abbildungsverzeichnis I](#_Toc96972248)

[II. Tabellenverzeichnis I](#_Toc96972249)

[III. Literaturverzeichnis I](#_Toc96972250)

[IV. Abkürzungs- und Symbolverzeichnis III](#_Toc96972251)

[Anhang IV](#_Toc96972252)

[A1 Pflichtenheft (OPTIONAL) IV](#_Toc96972253)

[A2 Schlussfolgerung / Projekterfahrung IV](#_Toc96972254)

[A3 Projektterminplanung IV](#_Toc96972255)

[A4 Arbeitsnachweis Diplomarbeit IV](#_Toc96972256)

[A5 Datenblätter (OPTIONAL) V](#_Toc96972257)

[A6 Technische Zeichnungen (OPTIONAL) V](#_Toc96972258)

1. Einleitung
2. Vertiefende Aufgabenstellung

## Crispin Holleis

Konzeptionierung und Implementieren des steuernden neuronalen Netzwerks, einschließlich kurzer Erläuterung der allgemeinen Funktionsweisen verwendeten Algorithmen im Rahmen der Dokumentation.

## Manolo Jaschke

Simulation des Verkehrs als Produkt individueller Ziele einzelner Fahrzeuge, außerdem Entwicklung einer koordinierenden Schnittstelle zwischen Simulation, neuronalem Netzwerk sowie Userinterface.

## Nils Kuhn

Konzeptausarbeitung eines Userinterfaces und Entwicklung der grafischen Oberfläche sowie Visualisierung des Simulationsvorganges und zugehöriger Daten mittels einer Engine.

# NeuroEvolution

## Einführung

Die Steuerung des Straßenverkehrs soll von einem neuronalen Netzwerk übernommen werden, dieses übernimmt nicht nur die Wegfindung, sondern auch die Aufgabe, die einzelnen Verkehrsteilnehmer möglichst effizient an ihr jeweiliges Ziel zu bringen. Hierzu wird die Auswirkung von dichtem Verkehr in der Straßennetzsimulation berücksichtigt: Je mehr Autos sich auf derselben Straße befinden, desto langsamer wird ihre Geschwindigkeit.

Ziel dieses Mechanismus ist es, das neuronale Netzwerk zu einer Aufteilung der Fahrzeuge auf verschiedene Routen zu animieren.

## Auswahl des Algorithmus

Mit zunehmender Popularität von Machine Learning haben sich eine Vielzahl verschiedener Techniken entwickelt. *Machine Learning* ist ein Teilgebiet der künstlichen Intelligenz und bezeichnet Algorithmen, die sich durch Erfahrung und/oder Datenanalyse automatisiert verbessern können. Eine Verbesserung bedeutet in diesem Kontext einen Fortschritt beim Erreichen eines expliziten Ziels.

Die Aufgabe der Verkehrsoptimierung impliziert eine Restriktion, welche die möglichen Algorithmen stark einschränkt: Die Lösung ist nicht bekannt, nur die Qualität der Antwort kann gemessen werden. Dies entspricht einem *Reinforcement Learning* Szenario. Dieser Bereich des Machine Learning beschäftigt sich mit der Aufgabe, Entscheidungen zu treffen, welche zu einer maximalen Belohnung führen. Zur Berechnung der Belohnung dient oftmals eine *Fitness-Function*, welche den Erfolg einer gewissen Entscheidungskette quantifiziert.

Ein prominenter Vertreter des Reinforcement Learnings sind die biologisch-evolutionär inspirierte *Evolutionäre Algorithmen (engl. Genetic Algorithms),* welche mittels Selektion und Operatoren wie Mutation und Paarung eine Lösung suchen. Eine gefundene Lösung ist hierbei nicht garantiert ideal, meistens aber ausreichen gut.

Der große Vorteil von genetischen Algorithmen gegenüber anderen Techniken liegt in der relativen Einfachheit ihrer Kernmechanismen. Da diese selbst implementiert werden sollen, wurde mit dem NEAT-Algorithmus ein Vertreter dieser Kategorie gewählt. Im folgenden Kapitel wird die Funktionsweise von NEAT zusammengefasst.

## NEAT

### Allgemeines

Das Akronym NEAT steht für *NeuroEvolution of Augmenting Topologies* und ist ein genetischer Algorithmus zur Entwicklung künstlicher neuronaler Netzwerke.

Ein *neuronales Netzwerk* stellt eine Struktur mit einer fixierten Anzahl von Ein- und Ausgängen dar, welches aus miteinander verbundenen *Neuronen* besteht. Dieses Netz besteht aus zwei fundamentalen Komponenten:

* Die **Neuronen** fungieren als Knoten mit beliebig vielen Eingängen und einem Ausgang. Ist das Netzwerk aktiv, summieren die Neuronen ihre Eingänge auf und übergeben die Summe an eine nichtlineare *Aktivierungsfunktion*. Das Ergebnis liegt dann am Ausgang an.
* **Verbindungen** vernetzen ein Source- und ein Target-Neuron und verfügen über einen konstanten Faktor, ein sogenanntes *Weight*. Ihre Aufgabe besteht aus der Weiterleitung des Ausgangswertes des Source-Neurons an einen der Eingänge des Target-Neurons. Der weitergeleitete Wert wird hierbei mit dem Weight multipliziert. Je nach Vorzeichen kann die Verbindung folglich stimulierend oder hemmend wirken.

Ein neuronales Netz von ausreichender Größe kann bewiesenermaßen jede mathematische Funktion abbilden.

### Funktionsweise

Im Gegensatz zu vielen anderen neuroevolutionären Methoden optimiert NEAT nicht ein statisches neuronales Netzwerk, sondern erlaubt das Hinzufügen (und in einigen Implementierungen Löschen) von Neuronen und Verbindungen. Dies ermöglicht zusätzlich zur Anpassung der Gewichtungen die Entwicklung einer möglichst effizienten Topologie, führt allerdings auch zu einer Reihe von Schwierigkeiten.

Die Evolution geschieht in *Generationen,* welche eine gewisse Anzahl von Individuen enthalten, entsprechend der *Populationsgröße.* In jeder Generation wird die Fitness der Individuen bestimmt, je erfolgreicher diese sind, desto höher die Wahrscheinlichkeit auf Fortpflanzung in die nächste Generation.

### Genetic Encoding

Genetische Kodierung bezeichnet die Art, auf die das neuronale Netzwerk gespeichert wird. Im Fall von NEAT muss diese Struktur zwei fundamentale Operatoren erlauben: *Mutation* und *Paarung (engl. mating oder crossover).*

* **Mutation** bezeichnet eine zufällige Veränderung eines Netzes. Dies kann die Veränderung eines Connection-Weights, das Hinzufügen (bzw. Entfernen) einer Verbindung sowie das Hinzufügen eines Neurons durch Zweiteilung einer vorhandenen Verbindung sein.
* **Paarung** entspricht der Kreuzung zweier unabhängiger Netzwerke. Hierbei muss sichergestellt werden, dass das Ergebnis aussagekräftig und sinnvoll ist.

Aufgrund der variablen Netzstruktur werden nicht nur die verschiedenen Verbindungsgewichtungen gespeichert, sondern auch die Topologie. Als Folge hiervon treten drei große Probleme auf:

* **Competing Conventions** bedeutet, dass unterschiedliche Layouts dieselbe Funktion erfüllen können. Dies ist beispielsweise bei gespiegelten Netzen der Fall.
* Die **variable Länge** der entwickelten Lösungen bedeutet, dass auch die interne Repräsentation entsprechend dynamisch sein sollte, um einerseits keine willkürliche Obergrenze zu setzen und andererseits auch bei kleinen Topologien Ressourcen effizient nutzen zu können. Die Konsequenz dieser Anforderung ist die Unvereinbarkeit mit einigen klassischen Ansätzen wie beispielsweise Connection-Bitmasks.
* Zusätzlich ist beim **Crossover** sicherzustellen, dass der erzeugte Nachwuchs mit angemessener Wahrscheinlichkeit funktional ist. Hierzu ist eine Identifikation von funktionsgleichen Komponenten notwendig, um diese sinnvoll kreuzen zu können. Das Problem der competing Conventions erschwert diese Aufgabe zusätzlich, denn auch Bestandteile mit derselben Funktion können unterschiedliche Layouts aufweisen.

NEAT löst diese Schwierigkeiten mit *Genomen* und dem Tracking ihrer Abstammung.

Genome repräsentieren Verbindungen oder Neuronen des Netzwerks, stellen also eine Kodierung eines Multigraphs dar. Verbindungsgenome bestehen aus Source- sowie Target-Neuron und Weight, Neurongenome lediglich aus einer Neuronen-ID.

Grafik Repräsentation

Die Ursprungsverfolgung erfolgt über die sogenannte *Innovation Number*, eine einzigartige Zahl, welche einer Verbindung beim Entstehen zugeordnet wird. Werden die Verbindungen vererbt, können sie mittels dieser Nummer auch nach beliebig vielen Generationen noch verglichen werden. Besitzen zwei Verbindungen dieselbe Nummer, so geht der Algorithmus auch von einer gleichen Funktion aus. Um Netzwerke auf diese Weise kompatibel zu halten, darf es auch bei den Neuronen-IDs keine Duplikate geben.

Der Crossover-Mechanismus nimmt das erfolgreichere Elternnetz (höhere Fitness) als Grundlage und bildet den arithmetischen Mittelwert der Gewichtungen aller übereinstimmenden Verbindungen. Bei gleicher Fitness werden alle nicht übereinstimmenden Genome mit 50 prozentiger Wahrscheinlichkeit übernommen.

Grafik Mating

### Speciation

Ein weiteres Problem ist die Tatsache, dass neue Mutationen zu Beginn oftmals schädlich für das Fitness-Rating eines Netzwerks sind, auch wenn sie auf längere Sicht vorteilhaft wären. Dies ist der Fall, da hinzugefügte Verbindungen erst angepasst werden müssen, bevor sie ihre Funktion erfüllen können. Um dies zu ermöglichen, wird die Bevölkerung in Spezies unterteilt. Jede Spezies erhält einen gewissen Anteil des Nachwuchses, wobei größere Spezies proportional weniger Nachwuchs erhalten. Die Fitness einer Spezies errechnet sich als

wobei N die Anzahl der Individuen einer Spezies, fi die Fitness eines Individuums und fk die Fitness der gesamten Spezies darstellt. Im weiteren Verlauf sowie im Source-Code wird auf diese Formel als *Quadratdurschnitt (engl. square average)* verwiesen.

Grafik Fitnessverlauf

Der Nachwuchs einer Spezies wird als direkter Anteil an der Gesamtfitness berechnet:

Nk steht hierbei für die Anzahl an Spezies, fk für den entsprechenden Quadratdurchschnitt, P für die Populationsgröße und nk für die zugeordnete Nachwuchsanzahl.

Die Einteilung in Spezies geschieht mittels einer *Distance-Function* sowie einem *Compatibility-Threshold*. Unterschreitet der berechnete Unterschied den Schwellenwert, so sind gehören zwei Individuen zur selben Spezies. Der Unterschied ist abhängig von der Anzahl an unterschiedlichen Verbindungen (U) sowie der durchschnittlichen Weight-Differenz übereinstimmender Verbindungen:

Die originale NEAT-Implementation unterscheidet zusätzlich zwischen zwei unterschiedlichen Arten von differenten Verbindungen, verwendet jedoch denselben Zahlenwert als Koeffizienten, folglich kann dies zu obiger Formel vereinfacht werden.

### Minimal Structure

Um eine möglichst effiziente Lösung, d.h. ein Netzwerk mit einer möglichst geringen Anzahl an Neuronen und Verbindungen zu finden, wählt NEAT den Ansatz einer minimalen Anfangsstruktur. Die Anfangspopulation besteht hierbei aus Netzwerken mit lediglich einer einzigen, zufällig erzeugten Verbindung. Dies vermeidet unnötige Komplexität, da die Topologie erst im Verlauf der Evolution graduell erhöht wird, solange kein hinreichendes Ergebnis erzielt wird. Zusätzlich verringert diese Technik den *Search Space,* da zu Beginn nur wenig veränderliche Parameter existieren und diese in späteren Nachkommen bereits angepasst sind.

## Implementation

### Einleitung

Die Grundfunktionalität des NEAT-Algorithmus steht wie im vorigen Kapitel erläutert bereits fest, dennoch gibt es einige Voraussetzung, die für einige Entscheidungen von Bedeutung sind.

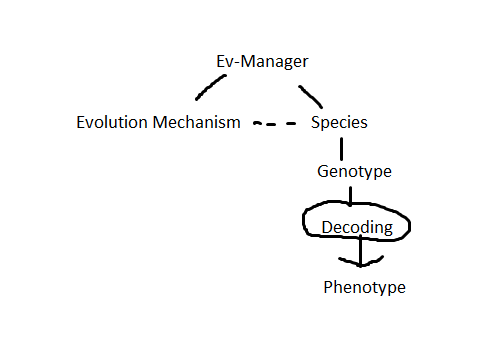
Der wichtigste Punkt ist die Performance. Für die Aufgabe der Verkehrsoptimierung ist ein verhältnismäßig großes neuronales Netzwerk zu erwarten, die Implementation muss fähig sein, dies zu bewältigen. Zusätzlich könnte Multithreading aufgrund des spezifischen Designs der Straßennetzsimulation nicht möglich sein, dies wäre ebenfalls durch ein leistungsfähigeres Programm auszugleichen.

Eine weitere essentielle Anforderung ist die Kompatibilität mit der Programmiersprache C#, da die graphische Oberfläche in der Grafik-Engine Unity und somit in C# erstellt wird. Das Programm muss also folglich entweder über ein entsprechendes Interface verfügen oder nativ kompatibel sein.

Zudem ist ein objektorientiertes Design anzustreben, da dies die Implementierung der komplexen Struktur vereinfacht und die originale Publikation bereits implizit eine Struktur umreißt. Außerdem ist C# rein objektorientiert, eine ähnliche Programmarchitektur vereinfacht die Implementation eines Interfaces.

Obwohl der NEAT-Algorithmus sich prinzipiell in jeder modernen Programmiersprache realisieren lässt, gibt es angesichts oben genannter Gründe bessere und weniger gut geeignete Kandidaten. Letztendlich fiel die Wahl auf C++, eine sehr performante, objektorientierte Sprache, welche mittels PInvoke (Platform Invoke) eine Interfacemöglichkeit zu C# besitzt.

### Programmarchitektur



Fehlt: Genome/Connection (?)

Diese Gruppierung der Kernkomponenten ist wie erwähnt bereits in der ursprünglichen NEAT-Publikation grob vorhanden. Zusätzlich Inspiration, besonders hinsichtlich des Genotype-Phenotype-Dualismus des Designs stammt von der *SharpNEAT* Library von Colin Green.

### Probleme

## Layout des neuronalen Netzwerks

# Erklärung der Eigenständigkeit der Arbeit

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe. Meine Arbeit darf öffentlich zugänglich gemacht werden, wenn kein Sperrvermerk vorliegt.

|  |  |  |
| --- | --- | --- |
| Ort, Datum |  | Verfasser 1 |

|  |  |  |
| --- | --- | --- |
| Ort, Datum |  | Verfasserin 2 |

…

# Abbildungsverzeichnis

[Abbildung 1: XYZ vi](#_Toc415201720)

[Abbildung 2:ABC [VI](#_Toc415201721)**.**](#_Toc415201721)

# Tabellenverzeichnis

[Tabelle 1: Arbeitsaufstellung IV](#_Toc415201490)

# Literaturverzeichnis

# Abkürzungs- und Symbolverzeichnis

Anhang

# A1 Pflichtenheft (OPTIONAL)

A2 Schlussfolgerung / Projekterfahrung

A3 Projektterminplanung

## A4 Arbeitsnachweis Diplomarbeit

A5 Datenblätter (OPTIONAL)

A6 Technische Zeichnungen (OPTIONAL)