Ausarbeitung – Interaktives Lern-Tool für neuronale Netze

# Arbeitsaufteilung

|  |  |  |
| --- | --- | --- |
| Phase | Patrick | Lukas |
| Analysieren | Ausarbeitung Idee | Ausarbeitung Idee |
|  | Erstellung Personas |
|  | Erstellung Szenario |
| Interpretieren | Recherche nach Tools | Erstellung Figma Kontext |
|  | Testen NN (neuronales Netz) from Scratch |  |
| Kreieren | Tests mit ThreeJS (Kreise und Linien | Erstellung Wheel und Slider Interaktion mit ThreeJS |
|  | Tests Tensorflow |  |
|  | Ergänzen von Infos bei Zwischenpräsentation | Umsetzung Zwischenpräsentation |
|  | Dynamisches Erzeugen von Layers + Tensorflow | Erzeugung TikTakToe und Studentenwohnheim Datensatz |
|  | Tests-NN TikTakToe Problem | Tests-NN TikTakToe Problem + Tests Studentenwohnheim |
|  | Gestaltung von Prototypen für Experteninterview | Umsetzung eines Prototyps für das Experteninterview (Interaktion + Studenten NN) |
|  | Debuggen Prototyp bei Generierung von Ergebnissen in Output-Layer | Erzeugung von Ergebnissen, nachdem neuer Input ausgewählt wurde |
| Bewerten | Vorbereitung + Durchführung von Experteninterview | Erstellung Fragebogen für Interview + Anfragen |
| Analysieren | Auswertung Interview |  |
| Interpretieren | Planung aus dem Ergebnis des Interviews |  |
| Kreieren | Vorbereiten des gemeinsamen Standes für die Einarbeitung der Ergebnisse des Interaktionsprototypen | Einarbeitung der Funktionen des Interatkionsprototypen |
| Bewerten | Durführung des AB-Tests | Durchführung des AB-Tests |

# Idee: Patrick und Lukas

Wir haben gemeinsam eine Idee evaluiert, die darauf abzielt, ein Tool zu entwickeln, mit dem Lernende durch einen konstruktivistischen Lernansatz ein neuronales Netzwerk durch Interaktion kennenlernen können. Es gibt bereits ähnliche Tools ([Interaktiv: Neuronales Netz – Hart und Trocken](https://www.hartundtrocken.de/my-product/interaktiv-neuronales-netz/)), jedoch keines, bei dem die internen Parameter des Netzwerks angepasst werden können. Die Ausarbeitung der Idee und die Entwicklung der Anwendung wurden von uns gemeinsam erarbeitet und vorangetrieben. Eine strikte Trennung der Arbeitsteilung war hier nicht möglich. Trotzdem gab es in bestimmten Bereichen Schwerpunkte, die im Folgenden aufgezeigt werden.

# Arbeitsbereich Lukas Willmann

## Analysieren

Für den ersten Abschnitt habe ich vier Personas erstellt. ([ID\_XAI\_TS/Documents/Personas at main · Z3r0cks/ID\_XAI\_TS (github.com)](https://github.com/Z3r0cks/ID_XAI_TS/tree/main/Documents/Personas)) Die Personas bilden lernwillige Personen ab, welche sich fortbilden wollen. Anhand dieser Personas erstellte ich ein Szenario für eine der Personas in einem möglichen Studenten-Alltag. ([ID\_XAI\_TS/Documents/Szenario at main · Z3r0cks/ID\_XAI\_TS (github.com)](https://github.com/Z3r0cks/ID_XAI_TS/tree/main/Documents/Szenario))

## Interpretieren

Mithilfe der Personas konnte ein „klickbarer“-Figma Prototyp erstellt werden, welcher für uns den Use-Case verdeutlichte. ([NN\_Explainer – Figma](https://www.figma.com/file/IPAPgOgXqxD9lLfVlNVUVv/NN_Explainer?type=design&node-id=0-1&mode=design)) Lernende würden an einem Desktop-PC auf einer Website Informationen über Neuronale Netze in verschiedenen Kapitel abrufen können. Auf dieser Website wäre das interaktive Tool als eigenes Kapitel oder am Ende eines passenden Kapitels eingebettet.

In unserem Projekt konzentrierten wir uns auf die Umsetzung des Tools. Nicht auf die komplette Umsetzung der Website mit dem Tool, da dies den Rahmen für uns gesprengt hätte. Dieser Figma-Kontext half uns aber das Tool auf bestimmte Weise zu entwerfen, wie z.B. die Beachtung von verschiedenen Desktop-Größen (Kamera-Handling in der Szene), jedoch nicht die Auslegung für Mobile, da wir davon ausgehen, dass die Nutzer für eine konzentrierte Lernatmosphäre eher an Desktop-PCs sitzen mit einer Scroll-Möglichkeit (Maus oder Touchpad eines Laptops).

## Kreieren

Damit die Kamera zentral auf die Layers gerichtet ist in Abhängigkeit von der Anzahl der Neuronen wie viele in einem Layer sind und wie viele Layer es gibt habe ich eine Kamera-Steuerungsklasse erstellt, mit welcher das neuronale Netz zentriert im Bild bleibt. ([ID\_XAI\_TS/src/classes/CameraHandler.ts at main · Z3r0cks/ID\_XAI\_TS (github.com)](https://github.com/Z3r0cks/ID_XAI_TS/blob/main/src/classes/CameraHandler.ts))

Um die Kommunikation zwischen unseren Programmierteilen zu verbessern habe ich ein Klassendiagramm erzeugt. ([ID\_XAI\_TS/Documents/interactUML.drawio at main · Z3r0cks/ID\_XAI\_TS (github.com)](https://github.com/Z3r0cks/ID_XAI_TS/blob/main/Documents/interactUML.drawio))

Die Interaktionen des Nutzers werden über die Klassen in der Interaktions-Lib gesteuert. ([ID\_XAI\_TS/src/classes/InteractionLib at main · Z3r0cks/ID\_XAI\_TS (github.com)](https://github.com/Z3r0cks/ID_XAI_TS/tree/main/src/classes/InteractionLib)) Durch einen Parameter in der „InteractionFactory“ ([ID\_XAI\_TS/src/classes/InteractionLib/InteractionFactory.ts at main · Z3r0cks/ID\_XAI\_TS (github.com)](https://github.com/Z3r0cks/ID_XAI_TS/blob/main/src/classes/InteractionLib/InteractionFactory.ts)) kann gesteuert werden, ob deine „Wheel“ oder eine „Slider“ Interaktion erzeugt werden soll. Diese Klassen wurden zuerst in einem JavaScript Projekt erstellt. Bei der Umstellung auf TypeScript wurden die Klassen aber angepasst.

Um Datensätze zu haben mit welchen wir NNs trainieren konnten erzeugte ich einmal einen TikTakToe-Datensatz und einen Wohnungen-Datensatz. Diese wurden durch Brute-Force-Python Scripte erzeugt. Bei dem TikTakToe Datensatz wurde viel mit Zufalls-Zahlen gearbeitet und in dem Datensatz der Wohnungen wurden logische Regeln und ebenfalls Zufallszahlen verwendet um die Daten zu generieren und zu labeln. ([ID\_XAI\_TS/DataSets at main · Z3r0cks/ID\_XAI\_TS (github.com)](https://github.com/Z3r0cks/ID_XAI_TS/tree/main/DataSets))

Um die NNs zu testen welche aus den Datensätzen entstehen können habe ich mit Jupyter-Notebooks gearbeitet. ([ID\_XAI\_TS/DataSets/JupyterTests at main · Z3r0cks/ID\_XAI\_TS (github.com)](https://github.com/Z3r0cks/ID_XAI_TS/tree/main/DataSets/JupyterTests))

Da die Kombination zwischen unserem Hauptprojekt und dem Projekt an welchem ich mit den Interaktion gearbeitet habe zu komplex war um in der Zeit bis zur Endpräsentation zu kombinieren, habe ich einen ersten testbaren Prototyp in meinem Projekt erzeugt. Das Modell ist in seinen Layern, Neuronen und Output Möglichkeiten statisch. Das Modell wird beim Start 200 mal mit dem Wohnungs-Datensatz trainiert. Danach kann der Tester die beiden Interaktionen unter Anleitung testen. [LksWllmnn/NeuroInteract: Interaction with a small Neuronal Network (github.com)](https://github.com/LksWllmnn/NeuroInteract)

## Bewerten

Um unsere wissenschaftliche Fragen für das Projekt zu bearbeiten habe ich einen einfachen Fragebogen erstellt. ([ID\_XAI\_TS/Documents/Fragebogen Interaktion der Anwendung KI.pdf at main · Z3r0cks/ID\_XAI\_TS (github.com)](https://github.com/Z3r0cks/ID_XAI_TS/blob/main/Documents/Fragebogen%20%20Interaktion%20der%20Anwendung%20KI.pdf)) Im ersten Teil des Fragebogens gibt es qualitative Fragen zur Nutzung des Prototypen. Hierbei war die Intuition, dass wir ein genaueres Verständnis bekommen welche Interaktion für NNs angenehmer ist und nicht nur ob eine Testperson das Mausrad oder einen Slider lieber mag. Im zweiten Teil haben wir den wissenschaftlichen „SUS“-Fragebogen verwendet um einen wissenschaftlich Bewährten Fragebogen für unsere Evaluation heran ziehen zu können. ([(PDF) SUS: A quick and dirty usability scale (researchgate.net)](https://www.researchgate.net/publication/228593520_SUS_A_quick_and_dirty_usability_scale))

# Arbeitsbereich Patrick Kaserer

|  |  |
| --- | --- |
| Analysieren | Ausarbeitung Idee |
|  |
|  |
| Interpretieren | Recherche nach Tools |
|  | Testen NN (neuronales Netz) from Scratch |
| Kreieren | Tests mit ThreeJS (Kreise und Linien |
|  | Tests Tensorflow |
|  | Ergänzen von Infos bei Zwischenpräsentation |
|  | Dynamisches Erzeugen von Layers + Tensorflow |
|  | Tests-NN TikTakToe Problem |
|  | Gestaltung von Prototypen für Experteninterview |
|  | Debuggen Prototyp bei Generierung von Ergebnissen in Output-Layer |
| Bewerten | Vorbereitung + Durchführung von Experteninterview |
| Analysieren | Auswertung Interview |
| Interpretieren | Planung aus dem Ergebnis des Interviews |
| Kreieren | Vorbereiten des gemeinsamen Standes für die Einarbeitung der Ergebnisse des Interaktionsprototypen |
| Bewerten | Durführung des AB-Tests |

## Interpretieren

Für die Umsetzung der Anwendung, wurde eine Recherche gestartet, um passende Werkzeuge zu finden. Obwohl anfangs noch die Idee bestand die Inhalte von Grund auf selbst zu entwickelt, wurde diese Idee recht schnell wieder fallen gelassen. Für die Umsetzung wurde mehrere Repositories. Das erste war eine reine JavaScript-Umsetzung (<https://github.com/Z3r0cks/Interaktionsdesign_XAI>) bei der im Code auch noch Ansätze zu finden sind, wie die Entwicklung ohne Libraries hätte gemacht werden können (z.B. die Datei [src/simple\_NN.js](https://github.com/Z3r0cks/Interaktionsdesign_XAI/blob/main/src/simple_NN.js#L61)). Wichtig war hierbei, dass die Anwendung möglichst einfach ist, und keine Wert auf hohe Komplexität gesetzt wird. Es soll ein einfaches neuronales Netz sein, welches die XAI-Anwendung mit einem einfachen Model repräsentiert.

## Kreieren

Durch den zu hohen Aufwand entschieden wir uns auf das Tool TensorFlow.js (TsJs) zu wechseln, was die Implementation einfacher gestaltet. Da erste Repository als ein Testdurchlauf angelegt wurde, entschieden wir uns daraufhin für eine strukturiertere Variante mit TsJs und Typescript. Der erste richtige Prototyp war [dieser Prototyp](https://github.com/Z3r0cks/ID_XAI_TS). Hierbei wurde stark auf die Lesbarkeit des Codes geachtet, und auf die Modulierung der Inhalte mit Klassen. Dieser Prototyp mit der Bezeichnung „Dynamischer Prototyp“ ist dafür da, die Eingabe im Front-End zu repräsentieren, bei dem dynamisch die Anzahl der Layer und Neuronen bestimmt werden kann. Im Hintergrund ist TsJs integriert, bei dem ein TensorFlow-Model erstellt wird, sobald der entsprechende Button gedrückt wurde. Dieser Prototyp sollte die Hauptanwendung darstellen, bei dem alle Inhalte im Nachhinein implementiert werden.

Als Problem erkannten wir, dass wir für die Anwendung auch ein einfaches mathematisches Problem benötigen, auf welches das Model approximiert wird. Die Idee entstand dabei, ein TicTacToe-Spiel mit allen möglichen Ergebnissen zu entwerfen und das Model darauf zu trainieren. Trotz mehrerer Anläufe und intensiver Zeit beim Experimentieren und Debuggen, konnte das Model nicht zufriedenstellend fertiggestellt werden. Auch beim Nachfolgemodel (Studentenwohnheim) fand noch Debugging statt.

## Bewerten

Für das Experteninterview musste entsprechende Formulare erstell und vorbereitet werden. Anschießend wurde das Experteninterview mit Prof. Dr. Lasowski durchgeführt. Da hier nicht der ursprüngliche Prototyp wurde nicht verwendet. Da die Interaktion in der Anwendung von Lukas Willmann prototypisch implementiert wurde (<https://github.com/LksWllmnn/NeuroInteract>). Hierfür wurde zur Vorbereitung noch Veränderungen an der Anwendung durchgeführt, um den Test auf die Nutzerinteraktion anzupassen und das Hauptaugenmerk auf die Fragenstellung zu lenken. Da der Teil der Visualisierung nicht im aktuellen Repository zu finden ist, hier ein Ausschnitt des Sliders vorher und nachher:Ein Bild, das Text, Screenshot, Rechteck, Electric Blue (Farbe) enthält.

Automatisch generierte Beschreibung

Beim Expertentest wurde die Anwendung im Vornherein erläutert und dann zum Testen der beiden Eingabebereiche übergeben. Anschließend wurden zwei Fragebogen ausgefüllt und ein Interview über die Anwendung und die Eingaben durchgeführt, um ein ausführliches Ergebnis zu erhalten.

## Analysieren

Das Experteninterview wurde Anschließend analysiert.