



Ferdinand
Braun
Schule

Technische Schulen der Stadt Fulda

Dokumentation zur Abschluss Projektarbeit

Autonomes Überwachungsfahrzeug

Deep Neural Network mit Arduino Mikrokontroller

Camilo Fernandez

Fachinformatiker für Anwendungsentwicklung

Ausbildungsbetrieb: EELOY AG

Inhaltsverzeichnis

1. Ausgangssituation
 - a. Projektziel und Teilaufgaben, Kundenwünsche
 - b. Technisches Umfeld und Schnittstellen
2. Ressourcen- und Aufgabenplanung
 - a. Personal-, Sachmittel-, Termin- und Kostenplanung
 - b. Ablaufplan
3. Durchführung und Auftragsbearbeitung
 - a. Prozessschritte
 - b. Anpassungen und Abweichungen
4. Projektergebnis
 - a. Test und Abnahme
 - b. Bewertung von Ergebnis und Prozess
 - c. Ausblick
5. Kunden Dokumentation
6. Anhang
 - a. Literaturverzeichnis
 - b. Code-Listings

1. Ausgangssituation:

a. Projektziel und Teilaufgaben, Kundenwünsche:

Original: Der Kunde wünscht sich ein Modell, das auf Basis von Machine Learning Verkehrssituationen mit selbstfahrenden Autos simuliert. Dazu soll in einem ersten Schritt eine zweidimensionale Testumgebung mit einem einzelnen Auto erstellt werden.

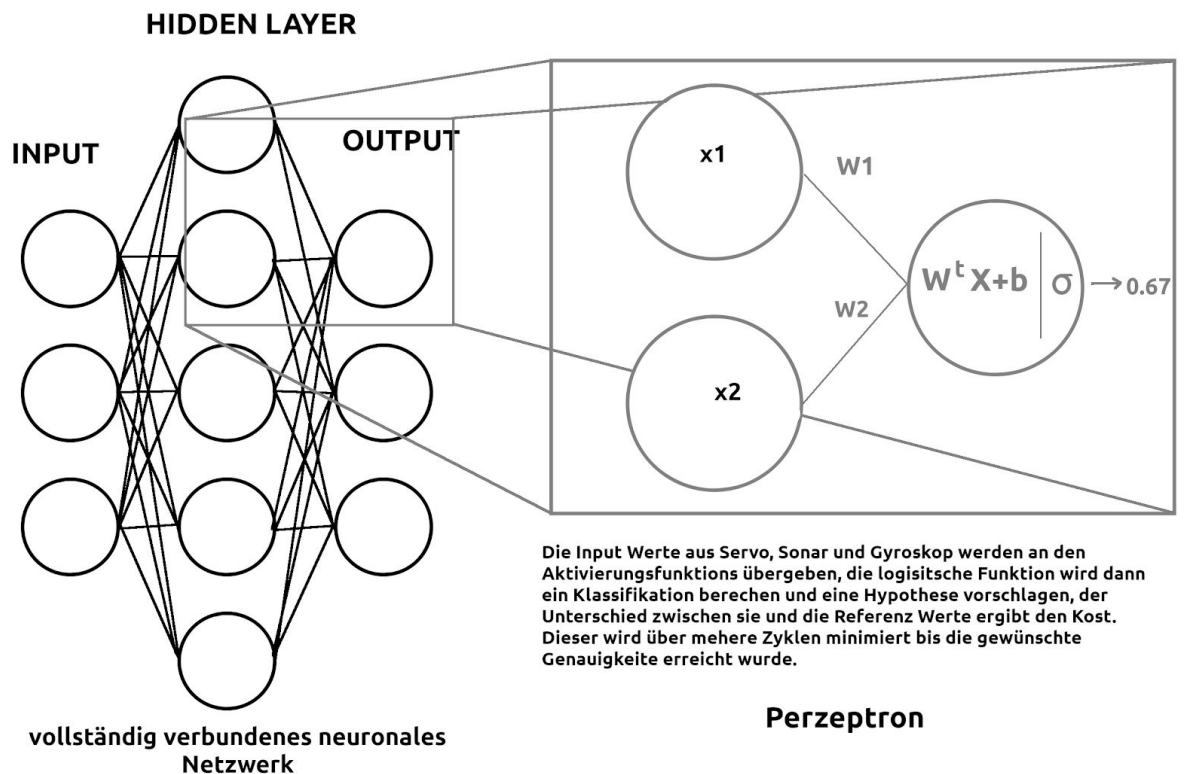
Neu: der Kunde sucht eine Lösung für die autonome Überwachung ihrer Fabrik während Auszeiten. Das Roboter kann sich in fremden oder wechselnden Umgebungen orientieren und Hindernisse umzugehen. Das Produkt soll

b. Technisches Umfeld und Schnittstellen

- Die Ultraschallsensoren dienen auch als Schnittstelle und sammeln wichtige Informationen aus der Umgebung wie die Distanz und relative Position zu Hindernissen und Fahrtrichtung des Fahrzeugs.
- **Original:** Das Machine Learning Modell besteht aus einem Agent, der Eingabedaten aus der Umgebung sammelt und am Anfang zufällig reagiert. Eine Police, die den Markov-Entscheidungsprozess auslöst, dient dazu, dass der Agent eine positive Bestärkung zu gewünschten Aktionen bekommt. Nach mehreren Iterationen werden die Bestärkungen und Strafen zusammengerechnet und der Agent wird die optimale Handlungsreihenfolge als Ausgabe des Systems lernen.

Neu: Weil die Trainingszeit und Speicherplatz exponential pro Variable steigt, habe ich die Umsetzung von den Markov-Entscheidungsprozess auf einen Neuronales Netzwerk umgestellt. Die Eingaben aus der Umgebung, vermittelt durch die Sensoren und Winkel der Micro Servo SG51, werden normalisiert und als Eingabe Schicht mit 3 Neuronen übermittelt.

Ein Hidden Layer mit 5 neuen Neuronen empfängt die Daten und aktiviert die Gewicht für jeden Synapse mit Logistische Regression, die Ausgabe Schicht wird dann auch 3 Neuronen verfügen. Nach circa 20000 Iterationen der selben Prozess durch Überprüfung mit eine Hypothese Funktion werden die Optimale werte für die Gewichte (nach Vorgabe) gefunden und für weitere Nutzung gespeichert.



- Der GY-521 Modul ist mit einem MPU6050 Chip versehen, der einen MEMS (Microelectromechanical Sensor) oder Vibrationskreisel Beschleunigungssensor und einen MEMS Gyroskop verfügt. Die Messungen der Drehwinkel in den 3 Achsen wird über einen I2C Schnittstelle an der Arduino kommuniziert. Die analogen Signale von 16-Bits für jeden Kanal werden in diesen Hardware einzel digitalisiert, was bedeutet genauen und schnellen gleichzeitige Messungen.
- Für das Endprodukt habe ich entschieden nur das Z-Achse zu nutzen, denn die innenraum Oberflächen sind komplett flach. Diese hat auch den Vorteil, dass befreit Speicherplatz in der Arduino UNO und beschleunigt das Trainingsprozess mehr als 20%.

2. Ressourcen- und Aufgabenplanung:

a. *Personal-, Sachmittel-, Termin- und Kostenplanung*

Hardware:

- Kettenfahrzeug
- Jumperkabel
- Widerstände-Kit
- Lötkolben
- Ultraschall-Sensoren HC SR04
- Laptops
- Steckplatine
- USB-Kabel
- H-Brücke L298N
- Beschleunigungssensor und Gyroskop GY-521 mit MPU6050

Software:

- Arduino IDE
- Arduino Bibliotheken
 - NewPing
 - NN, Neural Network Library
 - Tiny MPU
 - Arduino.h
- USB Port Driver
- Git

Personal:

Das Projekt wurde für 2 Personen konzipiert aber Florian Wiegard hat entschieden aus unbekannten Grunde aus dem Projekt auszutreten. Die Konzipierung, Durchführung, Dokumentation, Kosten und Programmierung wurden ausschließlich von Camilo Fernandez gemacht.

Kostenübersicht:

Artikel	Menge	Preis
Jumper Kabel, Steckplatine Kit	1	8,99€
Ultraschall-Sensor	3	6,46 €
Lötkolben Kit	1	14,99 €
GY-521 MPU-6050 3-Achsen Gyroskop und Beschleunigungssensor	1, Defekt	5,97€
GY-521 MPU-6050 3-Achsen Gyroskop und Beschleunigungssensor	3	6,69€
Dual H-Brücke DC-Schrittmotor Controller Modul L298N für Arduino	5	15,49€
ELEGOO UNO R3 Microcontroller Board ATmega328P ATMEGA16U2 Hauptplatine	1	13,99€
	Total	72.40€

b. *Aufgabenplan, Termine*

Siehe *Git Repository*:

<https://github.com/camiloAF/SelfDrivingCar/blob/master/doc/Aufgabenplanung.xlsx>

3. Durchführung und Auftragsbearbeitung

a) Prozessschritte

- Bereitstellen des Hardware:
 - Entwurf des Schaltplans
 - Austausch von Mikrokontroller
 - Ultraschall Sensoren anschließen
 - Provisorisches Netzteil besorgen
 - Spannung mit Multimeter prüfen
 - Ultraschall Sensoren mit Test Code ausprobieren
 - H-Brücke L298N mit Motoren und Akku anschließen
 - Spannung mit Multimeter prüfen
 - Motoren Probelauf
 - Installation von Beschleunigungssensor
 - Beschleunigungstest
 - Roll-Nick-Gier-Winkel Messungen Test
 - Defekt Beschleunigungssensor ersetzen und testen
- Software:
 - Test Code für Ultraschall Sensoren programmieren
 - Code für Betrieb von Motoren hochladen und Pulsdauermodulation Werte austesten
 - Paralleles Auslesen den 3 Ultraschall Sensoren
 - Test Code für Motoren programmieren
 - Test Code für Probelauf von Motoren programmieren
 - Arduino Bibliotheken für Ultraschall Sensoren und Motoren testen
 - Test Code für Beschleunigungssensor programmieren
 - Arduino Bibliotheken für MPU Chip testen
 - Alpha version für gesamten Programm schreiben (Objektorientiert)
 - Code verzweigen neues Feature für Machine Learning
 - Probleme mit Speicherplatz lösen, Code minimieren und optimieren
 - Eingabewerte und Ausgabe für Neuronales Netzwerk auswählen
 - Machine Learning Architektur entwerfen
 - Referenzwerte für Netzwerk berechnen
 - Aktivierungs Funktion auswählen
 - Hypothese Funktion schreiben
 - Verlustfunktion für Kosten der Neuronalen Netzwerk berechnen
 - Arduino Bibliothek für Neuronales Netzwerk testen
 - Werte der Ausgabe Schicht mit Steuerung verknüpfen
 - Allgemeinen Probelauf Code schreiben

b. Abweichungen vom Projektantrag

Nach Vorschlag von Herr Reuter haben wir uns entschieden, ein Kettenfahrzeug mit eingebautem Arduino aus einem früheren Projekt zu nutzen. Das Fahrzeug soll mit Ultraschallsensoren ausgestattet werden und mit Hilfe von Machine Learning das kollisionsfreie Fahren lernen.

Wegen der fehlender Zeit und außergewöhnliche Schul- und arbeitssituation, habe ich die virtuelle Simulationsumgebung wegfallen lassen.

Weitere nötige Anpassungen:

- **Problem:** Originales Fahrzeug Arduino Board lässt sich nicht mit der Arduino IDE verbinden, dadurch ist das Hochladen von Code nicht möglich.
Lösung: Wir erhalten von der Schule ein anderes Board.
- **Problem:** Anzahl von Pin in Arduino Uno reicht nicht für 3 Ultraschall Sensoren.
Lösung: 2 von 3 festgebundenen Sensoren entfernen und das bleibende auf ein Mikro Servo montieren um die Messwinkel zu erhöhen. Diese hat ein positives Ergebnis weil jede Sensor hätte nur ein 15° Winkel, aber der Servo hat ein 180° Drehwinkel.
- **Problem:** Das Fahrzeug hat keine Orientierung in Raum.
Lösung: Beschleunigungssensor und Gyroskop hinzufügen.
- **Problem:** Trotz Architektur Änderungen ist der Arduino Uno Speicherplatz für dynamische Variablen ungenügend
Lösung: Code Optimierung durch Entfernen von Klassen Objekten, Refactoring Datenstrukturen von Integer und Double auf Float und Byte, Entfernung von Kommentare und Test Funktionen von der Steuerung des Kettenfahrzeuges, Entfernen von Wifi Steuerung Funktionen um die volle Kontrolle aufs das Neuronalen Netzwerk zu überlassen, Entfernen von Eingabevariablen für X, Y Achsen, Normalisierung von allen Werten
- **Problem:** Der Beschleunigungssensor GY-521 ergibt falsche Werte für die Roll-Nick-Gier-Winkel.
Versuche:
 - Geschobene Register Werte als unsigned int ausgeben lassen
 - Einzelne Register Werte auslesen
 - Empfindlichkeit der Vibrationskreisel von 2g (Standard Schwerkraft), 4g, 8g und 16g austesten.
 - Empfindlichkeit der Vibrationskreisel von 250 grad/sek, 500 grad/sek, 1000 grad/sek und 2000 grad/sek austesten.**Lösung:** Das Modul war defekt nach dem Austausch hat normal funktioniert.

- **Problem:** Die Antriebsmotoren haben eine Ausgabe Spannung von 12V je, aber die Arduino Uno, gibt bis 5V.
Lösung: H-Brücke L298N direkt zur Akku mit 12V Ausgänge anschließen.
- **Problem:** Fehlender Deckel für das Fach von Akku
Lösung: Einen neuen skizziert und in 3D Drucker drucken.
- **Problem:** Neural Network braucht bis 20 Minuten fürs Training
Lösung: Input Arraytyp von Byte auf float geändert für eine 200% Zeitverkürzung
- **Problem:** Fehlende Zeit um verschieden Referenz Input Output Werte fürs Neural Network auszuprobieren.
Lösung: Noch offen.
- **Problem:** Trainingszeit sind trotzdem sehr unregelmässig.
- **Lösung:** Input und Output Neurone für Servosteuerung entfernt
- **Problem:** Aktivierungsfunktion bleibt in lokalen optima und minimiert die Fehler nicht weiter
- **Lösung: Offen**

4. Projektergebnis

a) *Test und Abnahme*

Soll-Aufnahme	Ist-Aufnahme
Das Fahrzeuge kann selbständig fahren	Nicht erfüllt
Das Fahrzeuge kann Hindernisse ausweichen	Nicht erfüllt
Das Fahrzeuge ist anpassungsfähig	Nicht erfüllt
Das Projekt hat eine simulationsumgebung	Nicht erfüllt
Der Nutzer soll nichts konfigurieren	erfüllt

b) *Bewertung von Ergebnis und Prozess*

Von 1 unzufrieden bis 10 sehr zufrieden

Thema	Bewertung
Erfolg des Projektes	5
Vorbereitung und Durchführung des Planes	6
Problemlösung	8
Arbeitsverhalten	10
Pünktlichkeit	4

c) *Ausblick*

Das Endprodukt bietet sich am Besten für autonome Innenraum Navigation wie Gebäude, Lagerhäuser oder ähnliches, jedoch das Fahrzeug kann mit Kameras, GPS für Outdoor Überwachung, Ärmern und andere Sensoren wie Abgasmessgeräte, Temperatur oder Feuchtigkeit Sensoren um eigenständigen Kontroll, Reparatur, Transport und/oder Überwachung Routinen durchzuführen.

Weitere Funktionalitäten könnten ohne große Probleme eingefügt wie Indoor Mapping eingefügt werden aber eine Upgrade von der Platine wäre nötig um bessere Hardware und zusätzliche Schnittstellen zu addieren.

Ebenso habe ich gelernt dass es besser ist, erst eine Testumgebung zu entwickeln bevor man zum Produkt und Code programmieren anfangt.

5. Kunden Documentation



Bedienungsanleitung

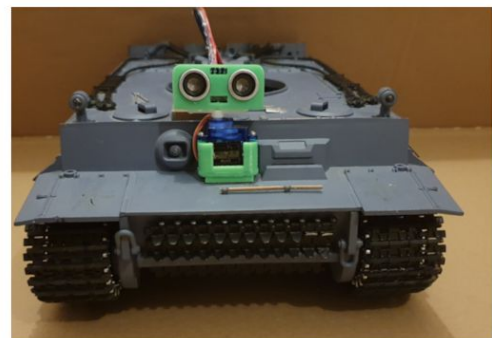
Vielen Dank, dass Sie sich für unser Produkt entschieden haben.
Bitte lesen Sie diese Bedienungsanleitung sorgfältig durch, bevor Sie das Fahrzeug einschalten, da sie alle Informationen enthält, die Sie zur korrekten Verwendung dieses Produkts benötigen.

Autonomes Fahrzeug mit künstlicher Intelligenz

Einrichtung

Batterie und Ladegerät enthalten

- Laden Sie den Akku 6 Stunden lang auf und lassen Sie ihn nicht unbeaufsichtigt
- Öffnen Sie die untere Abdeckung, indem Sie den Haltedeckel schieben
- Stecken Sie die Motorbatterie ein
- Setzen Sie die Abdeckung wieder auf
- Schalten Sie den Schalter der Unterseite ein
- Warten Sie auf den Bordcomputer, um das Training der künstlichen Intelligenz zu beenden. Dieser Vorgang kann bis zu 6 Minuten dauern. Wenn das Fahrzeug nach dieser Zeit den Einrichtungsvorgang nicht gestartet hat, schalten Sie es bitte aus und wieder ein.



Anwendung

Das Fahrzeug kann autonom ohne zusätzliche Einstellungen Innenräumen navigieren und mit seinem Sonar Hindernissen ausweichen. Es ist nützlich, Bereiche zu überwachen, in denen biologische Gefahren für Ihre Mitarbeiter bestehen können.

Der Robotrack benötigt keinerlei Benutzerkonfiguration. Bitte öffnen Sie die obere Abdeckung nicht, da dadurch die Garantie erlischt.

Wenn Sie das Fahrzeug mit zusätzlicher Hardware wie GPS, Kamera oder einem anderen chemischen Erkennungssystem anpassen möchten, wenden Sie sich bitte an unsere Hotline unter +4912345678910

6. Referenz

- https://en.wikipedia.org/wiki/Logistic_regression
- <http://robotics.hobbizine.com/arduinoann.html>
- <https://www.youtube.com/watch?v=7FMGXCAwbnc>
- https://en.wikipedia.org/wiki/Neural_network
- <https://data-science-blog.com/blog/2018/08/31/funktionseise-kunstlicher-neuronaler-netze/>