

# Simulation selbstfahrender Autos mithilfe von künstlichen neuronalen Netzen und evolutionären Algorithmen

Bachelorarbeit von Eike Stein

Gutachter: Dr. Marco Grawunder, Cornelius Ludmann

26.09.2016

# Inhalt

- Einleitung
  - Motivation
  - Forschungsfrage
- Grundlagen
  - Autonomes Fahren
  - Evolutionäre Algorithmen
  - Künstliche neuronale Netze
- Konzept
- Demo
- Fazit und Ausblick
  - Fazit
  - Grenzen
  - Ausblick

# Einleitung

# Motivation

- Jährlich sterben in Deutschland rund 3500 Menschen bei Verkehrsunfällen [1]
  - Häufig ist menschliches Versagen die Ursache
  - Es werden weitere Sicherheitssysteme benötigt
- Bestehende Sicherheitssysteme nur unterstützend
  - Sicherheit muss immer noch durch den menschlichen Fahrer garantiert werden
- Autonomes Fahren neuartiger Lösungsansatz
- Viele unvorhersehbare Faktoren wirken auf den Entscheidungsprozess beim Fahren ein
  - Keine einfachen *Wenn-Dann* Beziehungen möglich
  - Systeme benötigt, die selbstständig auf neue Situationen angemessen reagieren können
  - Zum Beispiel künstliche neuronale Netze

# Forschungsfrage

- Eignen sich künstliche neuronale Netze, um Sensordaten eines Fahrzeugs zu verarbeiten und dieses zu steuern?
    - Fahrverhalten sollte schnell, sicher und *menschenähnlich* sein
  - Zur Beantwortung kommt ein neuentwickeltes Softwaresystem zum Einsatz
- Inwieweit eignet sich ein Softwaresystem zur Durchführung und Auswertung von Simulationen, in denen Fahrzeuge mithilfe von künstlichen neuronalen Netzen gesteuert werden?

# Grundlagen

# Autonomes Fahren

- Ziel ist die vollständige Kontrolle von Autos, Bussen und Lastwagen
- Einer der ersten Beiträge von Tsukuba Mechanical Engineering Laboratory
  - Autos konnten weißen Linien auf Testgelände folgen [2]
- Mittlerweile sind große Unternehmen wie Google und Tesla führend in der Entwicklung [3][4]
- Verschiedene Sensoren kommen zum Einsatz
  - Kamera
  - Ultrasound
  - GPS
  - Laser
  - ...
- Sensordaten → Steuerung des Fahrzeuges
  - Fuzzy-Logik
  - Künstliche neuronale Netze

# Evolutionäre Algorithmen (1)

- Anlehnung an natürliche Selektion in der Natur [5]
  - *Survival of the fittest*
- Individuum  $\Leftrightarrow$  Lösung für Problem
  - Form des Flügels eines Windrades
  - Form der Karosserie eines Rennwagens
  - Fahrverhalten eines autonomen Autos
- Lösungskandidaten werden durch *Gene* kodiert [6]
  - Gen  $\Leftrightarrow$  Bit oder Dezimalzahl
  - Mehrere Gene definieren eine Lösung



## Evolutionäre Algorithmen (2)

- Iterative (kleine) Veränderungen [7]
    - Mutation
    - Rekombination von Individuen
  - Selektion der besten Individuen in nächste Generation
- ➔ Nach vielen Generationen existieren Individuen, die akzeptable Lösungen kodieren

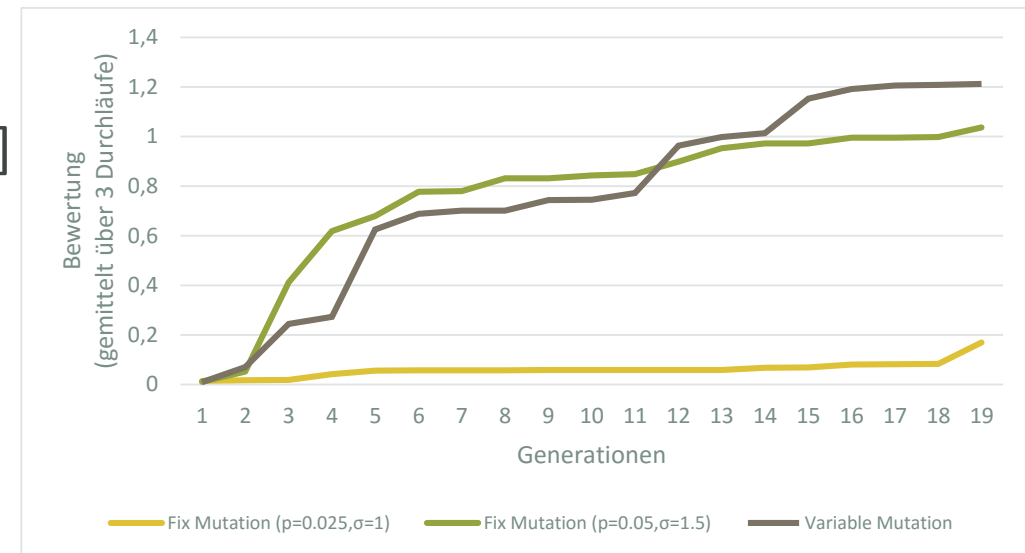
## Evolutionäre Algorithmen (3) – Ablauf

1. Zufällige Ausgangspopulation
2. Fitnessfunktion bewertet Individuen
  - Luftwiderstand im Windtunnel
3. Selektion von Individuen anhand ihrer Fitness
  - I. Rekombination
  - II. Mutation
  - III. Bewertung der Fitness
4. Auswahl für nächste Generation aus alten und neuen Individuen
5. Gehe zu Schritt 3

Quelle: in Anlehnung an [7]

## Evolutionäre Algorithmen (4)

- Laufen prinzipiell unbegrenzt lange [7]
  - ➔ Abbruchkriterien
    - Zeit
    - Anzahl Generationen
    - Bewertung der Individuen
- Problem der Parameterbestimmung [8]
  - Mutationswahrscheinlichkeit
  - Selektionsmechanismus
  - ...



# Künstliche neuronale Netze (1)

- Versuchen Brücke zu schlagen zwischen dem Intellekt von Menschen und der Rechengeschwindigkeit von Computern
- Inspiriert von neuronalen Verbindungen im Gehirn [9]
- Aufbau über Ebenen
  - Eingabeebene
  - *Hidden layers*
  - Ausgabebene

## Künstliche neuronale Netze (2)

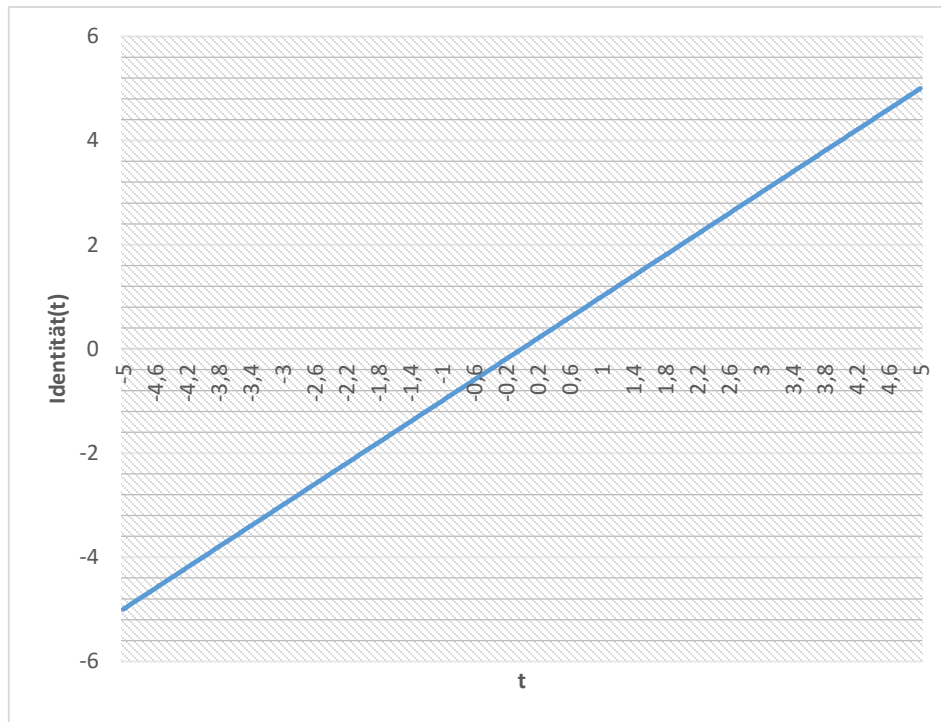
- Jede Ebene setzt sich aus Neuronen zusammen [10][11]
  - Meistens ist jedes Neuron mit jedem Neuron der nächsten Ebene verbunden
  - Jede Verbindung ist gewichtet
  - Gewichtete Summe alle Ausgaben der vorherigen Ebene ergibt Eingabewert  
 $\rightarrow input = \sum_{i=0}^n output_i * weight_i$
  - Ausgabe wird durch Aktivitätsfunktion aus Eingabe errechnet
- Daten werden an den Neuronen der Eingabeebene angelegt
  - Vergleichbar mit Helligkeitswahrnehmung durchs Auge
- *Hidden layers* stellen die Daten in Verbindung zueinander
  - Genauer Wirkmechanismus noch unklar
- In der Ausgabebene wird das Ergebnis dargestellt

## Künstliche neuronale Netze (3)

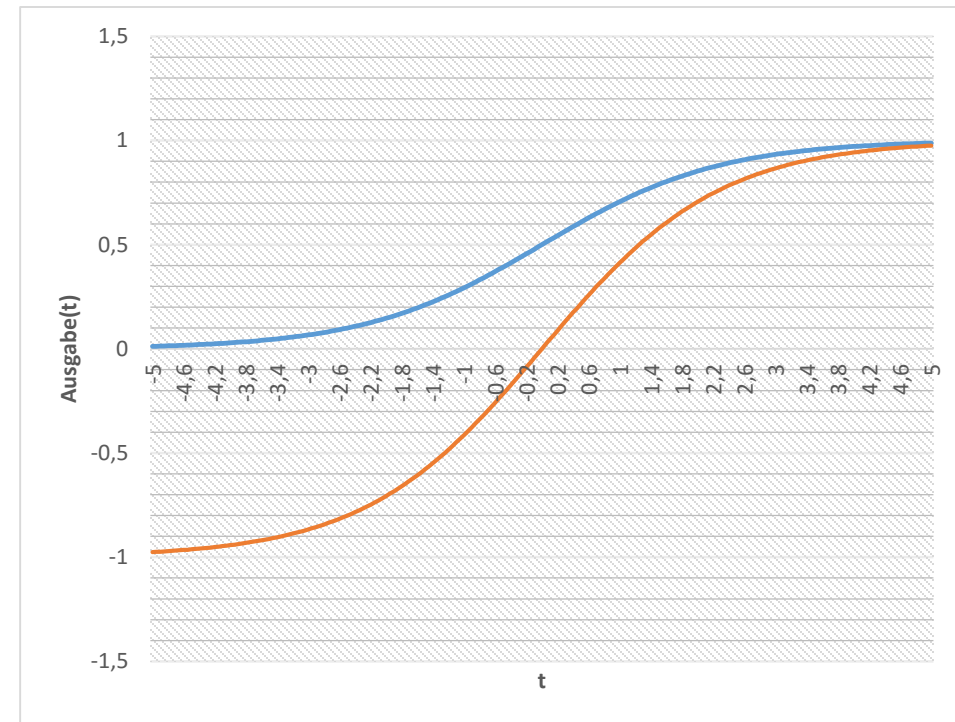
- Aktivitätsfunktion projiziert Werte [11]
  - Sigmoid in  $[0,1]$
  - TanH in  $[-1,1]$
- Damit die Ausgabe des Netzwerkes sinnvoll genutzt werden kann, müssen die Verbindungsgewichte trainiert werden
  - Ein Trainingsverfahren ist der Einsatz von evolutionären Algorithmen

# Künstliche neuronale Netze (4) – Aktivitätsfunktionen

## Identität



## Sigmoid/TanH [11]



# Konzept

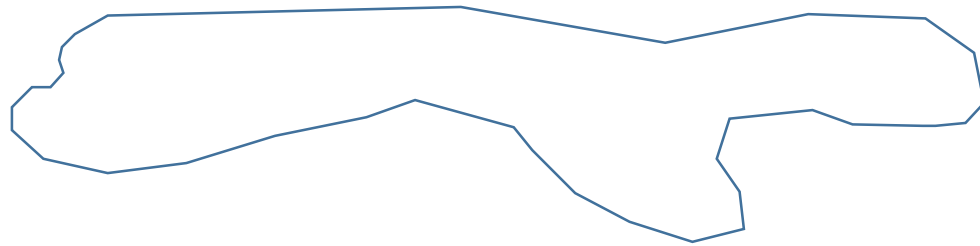


# Ansatz

- Ziel ist es ein Fahrzeug über ein künstliches neuronales Netz zu steuern
- Eingabedaten des neuronalen Netzwerkes sind Abstandssensordaten der Umgebung
- Ausgabe gibt Fahrverhalten an
- 2D physikalische Simulation ➔ 2D virtuelle Umgebung

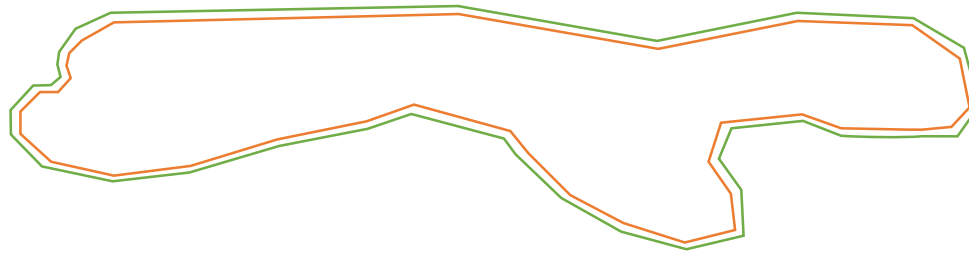
# Streckengenerierung (1)

- Rennstrecken eignen sich aufgrund abwechslungsreicher Abschnitte
- Nur ein Fahrzeug zur Zeit
- Quelle ist die Webseite GPSies [12]
- Aus GPS Koordinaten wird Polygon errechnet



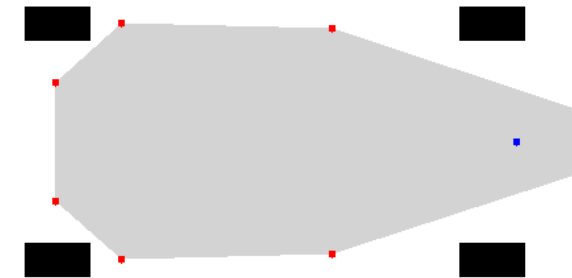
## Streckengenerierung (2)

- Innere und äußere Streckenbegrenzung durch vergrößern und verkleinern des Polygons



# Fahrzeug

- In Anlehnung an einen PKW
- Form über Polygon definiert
- Form des Fahrzeuges nicht nur kosmetisch relevant
  - Abstand der Reifen antiproportional zur Manövrierbarkeit
  - Größere Karosserie → höheres Gewicht → größere Trägheit
- Sensoren messen den Abstand ausgehend vom blauen Punkt in verschiedene Richtungen
  - Orientiert an der menschlichen Position im Fahrzeug



# Sensorverarbeitung (1)

- Die gemessenen Sensordaten dienen als Eingabedaten für das Netzwerk
- Ausgabe ist die Lenkausrichtung und Zielgeschwindigkeit
  - Lenkausrichtung in  $[-1,1]$
  - Zielgeschwindigkeit in  $[0.1,1]$
- Eingabedaten müssen normalisiert werden
  - Sensoren besitzen maximale Entfernung
  - ➔  $Eingabe = \frac{\text{gemessene Entfernung}}{\text{maximale Entfernung}}$
- Wahl der Topologie des Netzwerkes stellt Balance zwischen Lerngeschwindigkeit und maximal erreichbarer Performanz dar [14]

## Sensorverarbeitung (2)

- Festlegung der Anzahl der Sensoren vergleichbar mit Komplexität des Netzwerkes
- Platzierung paarweise, sodass das *Sichtfeld* symmetrisch ist
- In der Arbeit wurden die Winkel  $-40^\circ$ ,  $-20^\circ$ ,  $-4^\circ$ ,  $4^\circ$ ,  $20^\circ$  und  $40^\circ$  relativ zur Fahrtrichtung untersucht

# Simulationsablauf

- Fahrzeug wird auf ausgewählter Strecke platziert
- 1. Sensoren nehmen Umgebung wahr
- 2. Ausgabe des künstlichen neuronalen Netzwerks wird berechnet
- 3. Fahrzeugphysik wird berechnet
  - I. Reifen werden entsprechend gedreht
  - II. Beschleunigungskräfte werden auf die Reifen angewendet
  - III. Luftwiderstand und Reibung werden berechnet
- 4. Simulationsschritt wird ausgeführt
  - I. Kollisionen werden aufgelöst
  - II. Kräfte werden berechnet
  - III. Objekte werden bewegt

} *Farseer Physics Engine [13]*
- 5. Falls Zeit abgelaufen ist oder Fahrzeug mit Wand kollidiert ist, wird die Simulation abgebrochen
- 6. Ansonsten weiter mit Schritt 1.

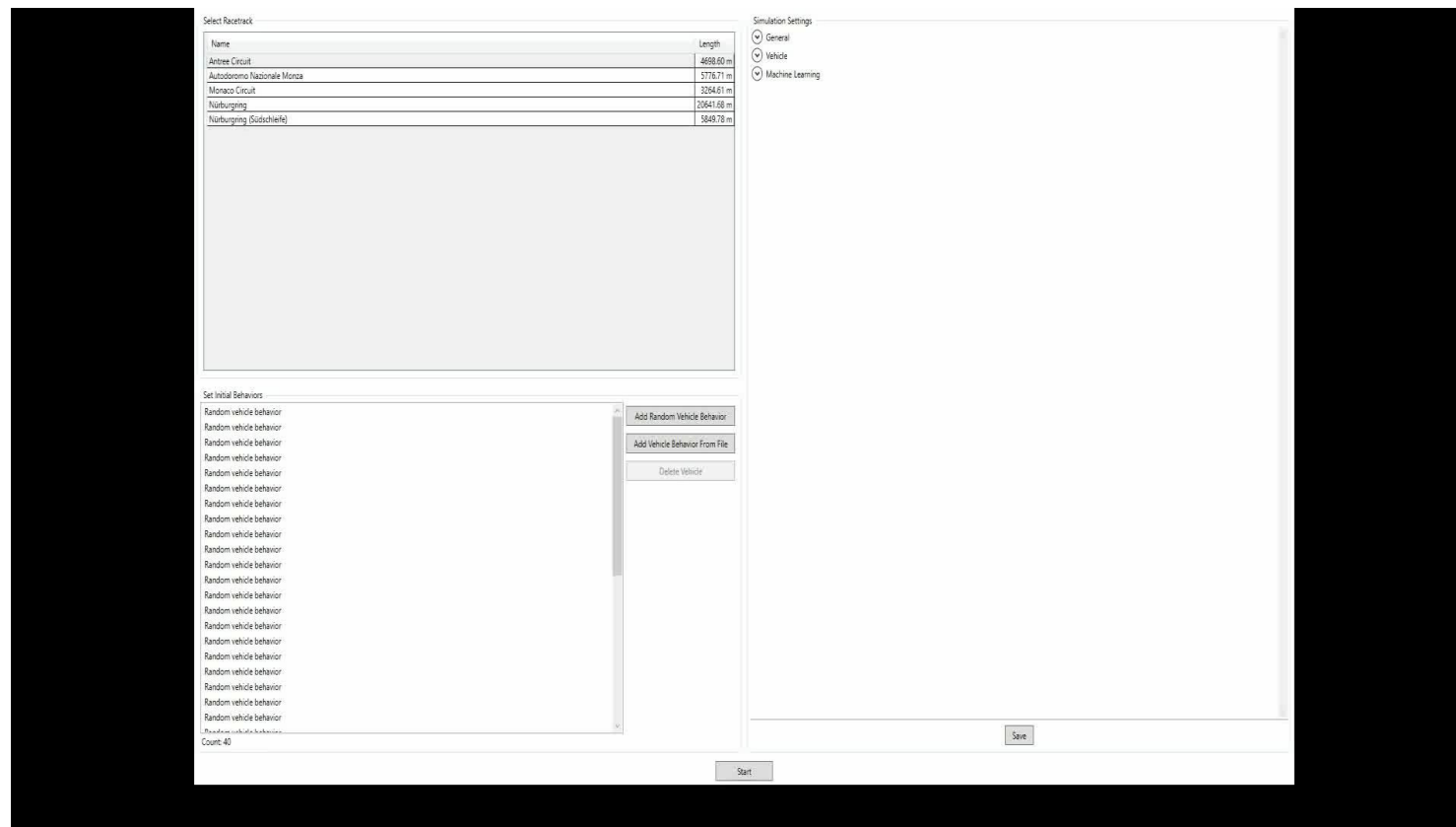
## Berechnung der nächsten Generation

- Die künstlichen neuronalen Netze mit dem *besten* Fahrverhalten werden unverändert in die nächste Generation übernommen
  - Die verbleibenden Plätze werden durch gewichtete Zufallsauswahl gefüllt (Roulette-Selektion) [15]
    - Die Gewichte der ausgewählten Netzwerke werden mutiert
- ➔ Streng monoton steigende Bewertung der besten Individuen



# Demo

# Video-Demo



# Fazit und Ausblick

## Fazit (1)

- In recht kurzer Zeit lernen künstliche neuronale Netzwerke ein Fahrzeug einen virtuellen Rundkurs zu durchsteuern
- Nach wenigen Generationen gelingt kollisionsfreies Durchfahren der ausgewählten Strecke
- Geschwindigkeit nimmt zunächst eine untergeordnete Rolle ein
  - Auch in späteren Generationen weiteres Optimierungspotential
- Das Fahrverhalten kann ausschließlich subjektiv beurteilt werden
  - Es wirkt etwas riskant
  - Pendeln des Fahrzeuges auch in späteren Generationen sichtbar
  - Nichtsdestotrotz recht zuverlässig

## Fazit (2)

- Trotz der erwähnten Probleme stellen künstliche neuronale Netzwerke ein vielversprechenden Ansatz zur Kontrolle von Fahrzeugen dar
- Das Softwaresystem stellt die nötige Grundlage für die Auswertung dar
  - Ohne dieses wäre eine systematische Auswertung kaum möglich
  - Die Diagramme, sowie die visuelle Simulation des Fahrverhaltens ermöglichen eine einfachere Entscheidungsfindung bei der Parameterbestimmung

# Grenzen

- Nur ein Fahrzeug zur selben Zeit
- Nur in Simulation getestet
- Keine Stördaten und Umgebung sehr simpel
- (Fahrzeug-)Physik nicht 100% realistisch

# Ausblick

- Nur ein kleiner technologischer Aspekt autonomer Autos untersucht
- Künstliche neuronale Netze sind vielversprechend aber weitere Untersuchungen sind nötig
  - Nicht untersuchte Anpassungen an der Simulation
  - Durchführung von Experimenten mit realen Fahrzeugen
- Weiterführende Fragen
  - Welche Probleme der Steuerung können ausschließlich von künstlichen neuronalen Netzen gelöst werden?
  - Wo müssen andere Technologien eingesetzt werden?
  - Inwieweit lassen sich diese kombinieren?

# Vielen Dank für Ihre Aufmerksamkeit

Haben Sie Fragen?



# Quellen

- [1] o.A., „Statistisches Bundesamt,“ 12 07 2016. [Online]. Available: [https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2016/07/PD16\\_242\\_46241.html;jsessionid=D A13AA6640CB70FA934860DCC9167B62.cae2](https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2016/07/PD16_242_46241.html;jsessionid=D A13AA6640CB70FA934860DCC9167B62.cae2). [Zugriff am 20 08 2016]
- [2] M. K. Alex Forrest, Autonomous Cars and Society, Worcester, 2007
- [3] o.A., „Google,“ o.J.. [Online]. Available: <https://www.google.com/selfdrivingcar/> [Zugriff am 31 Mai 2016]
- [4] o.A., „Tesla,“ o.J.. [Online]. Available: [https://www.tesla.com/de\\_DE/presskit/autopilot](https://www.tesla.com/de_DE/presskit/autopilot) [Zugriff am 08 18 2016]
- [5] C. Darwin, Über die Entstehung der Arten durch natürliche Zuchtwahl, Books on Demand, 2016
- [6] M. G. Xinjie Yu, „Representation and Evaluation,“ in Introduction to Evolutionary Algorithms, Springer, 2010, pp. 15-16
- [7] M. G. Xinjie Yu, „Simple Genetic Algorithm Infrastructure,“ in Introduction to Evolutionary Algorithms, Springer, 2010, pp. 17-23
- [8] C. F. L. Z. M. F.J. Lobo, Parameter Setting in Evolutionary Algorithms, Springer, 2007
- [9] P. v. d. S. Ben Kröse, An introduction to Neural Networks, Amsterdam, 1996
- [10] P. v. d. S. Ben Kröse, An introduction to Neural Networks, Amsterdam, 1996, p. 33
- [11] P. v. d. S. Ben Krose, An introcution to Neural Networks, Amsterdam, 1996, p. 17
- [12] o.A., „GPSies,“ o.J.. [Online]. Available: <http://www.gpsies.com/> [Zugriff am 20 08 2016]
- [13] e. a. Ian Qvist, „Codeplex,“ 26 08 2013. [Online]. Available: <https://farseerphysics.codeplex.com/> [Zugriff am 21 08 2016]
- [14] A. e. a. Fizelew, Finding Optimal Neural Network Architecture Using, Buenos Aires Institute of Technology, o.J.
- [15] F. H. Thomas Bäck, Extended Selection Mechanisms in Genetic Algorithms, Dortmund, o.J.