

Künstliche Intelligenz und Maschinelles Lernen für Entwickler

Computer werden immer 'intelligenter': AlphaGo Zero gewinnt das Go-Duell gegen den besten menschlichen Go-Spieler, Spracherkennungen wie Siri, Alexa oder Cortana verstehen zunehmend komplexere Anfragen, und Autos fahren autonom in echtem Straßenverkehr.

Was steckt dahinter?

Der Entwickler Workshop geht dieser Frage nach und gibt einen detaillierten Überblick über Methoden und Techniken des Maschinellen Lernens.

Neben etablierten Standardverfahren und Algorithmen wie z. B. Nearest-Neighbors-Classification, Lineare und Logistische Regression werden auch komplexere Ansätze wie Support Vector Machines und Neuronale Netze vorgestellt.

Neuronale Netze erzielen in vielen Anwendungsbereichen erstaunliche Leistungen und haben sich als Quasi-Standard in der Welt des maschinellen Lernens etabliert. Innerhalb der letzten Jahre hat das Konzept aus dem letzten Jahrtausend, nicht zuletzt dank Cloud-Computing, eine Renaissance erfahren. Das Konzept der Neuronalen Netzwerke ist mehrfach überarbeitet und weiterentwickelt worden. So existieren mittlerweile, optimiert für unterschiedliche Aufgabenstellungen, diverse Ansätze wie z. B. Convolutional Neural Network, Recurrent Neural Network oder das Long short-term memory Konzept.

Anhand ausgesuchter Beispiele wie z. B. Betrugserkennung und NLP (Natural Language Processing) werden die oben genannten Methoden und Techniken des maschinellen Lernens mittels führender Implementierungen wie TensorFlow, Keras und scikit-learn vorgestellt. Die Beispiele werden in kleinen Projekten gemeinsam erarbeitet und eigenständig in Python umgesetzt.

Dauer

3 Tage

Zielgruppe

Software-Entwickler und Software-Architekten

Voraussetzungen

Erfahrungen in einer Programmiersprache. Python Kenntnisse hilfreich.

Inhalt

Einführung und Überblick

- Was ist Lernen und Intelligenz
- Was ist Künstliche Intelligenz und Maschinelles Lernen
- Wie und warum kann eine Maschine lernen und intelligent werden
- Künstliche Intelligenz und Maschinelles Lernen Vergleich, Einordnung und Abgrenzung

Entwicklung und Validierung von Modellen

- Daten - Analyse, Aufbereitung und Import
- Features - Selektion und Reduktion
- Modell - Evaluierung, Diagnose und Optimierung
- Lernen - Training, Test und Validierung
- Fehler - Überanpassung vs. Unteranpassung

Methoden und Techniken

- Regression vs. Klassifikation
- Überwachtes, verstärkendes und nicht überwachtes Lernen
- KNeighbors, Regression, DecisionTrees, Random Forest
- Support-Vektor-Maschinen
- Neuronale Netze und mehrschichtige neuronale Netze
- Feed Forward Network
- Deep Neural Network
- Convolutional Neural Network
- Recurrent Neural Network
- Long short-term memory

Die Werkzeugkiste des Data-Scientists

- Python
- NumPy
- Scikit-Learn
- Pandas
- TensorFlow
- Keras
- Matplotlib
- Jupyter