



Development of an Administrative Web Frontend for Deep Learning Research

Lukas Güldenhaupt, 19.01.2018

Gliederung

- 1. Einführung in die Thematik
 - Problemstellung
 - Lösungsansatz
- 2. Aufbau
 - Typescript
 - Serverseite
 - Clientseite
 - Datenbank
- 3. Vorstellung der Anwendung
- 4. Fazit und Ausblick





Problemstellung

- Verfügbare Tools divers
- Großer und unterschiedlicher Parameterraum
- Hunderte trainierte Netzwerke
- Viele Ausgabedateien
- Keine Übersicht





Lösungsansatz

Webfrontend ANNA (Administrative Neural Network Application)

- Verwaltung von Projekten
- Speichern von Konfigurationsdateien und Ergebnissen
- Ausgaben vereinheitlichen
- Filterfunktion
- Mehr Übersicht





Gliederung

- 1. Einführung in die Thematik
 - Problemstellung
 - Lösungsansatz
- 2. Aufbau
 - Typescript
 - Serverseite
 - Clientseite
 - Datenbank
- 3. Vorstellung der Anwendung
- 4. Fazit und Ausblick





Typescript

- Nutzung von Javascript
- Typescript als Hilfe
- Java-Ähnlich
- Einfacher Einstieg





Serverseite – Meteor

- Meteor auf Basis von NodeJS
- Einfache Adaption und schneller Start
- Plattformunabhängig
- Typescript bzw. Javascript
- Reaktiv mithilfe von DDP
- Erweiterung mit Packages





Clientseite – Angular

- Javascript Framework von Google
- Model-View-Controller Pattern
- Plattformunabhängig
- Reaktiv





Datenbank – MongoDB

- NoSQL Datenbank
- Dokumentbasiert (JSON-Formate)
- Skalierbarkeit
- Gute Meteor-Integration
- Keine Datenkonventionen
- Perfekt für abweichende Datensätze





1. Einführung in die Thematik

- Problemstellung
- Lösungsansatz
- 2. Aufbau
 - Typescript
 - Serverseite
 - Clientseite
 - Datenbank
- 3. Vorstellung der Anwendung
- 4. Fazit und Ausblick





Fazit und Ausblick

- Schnelle, reaktive Anwendung
- Besserer Überblick
- Einfach erweiterbar
- GPU-Cluster Anbindung
- Metastudien







Institut für Nachrichtentechnik

Literaturverzeichnis



