



Technische
Universität
Braunschweig



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

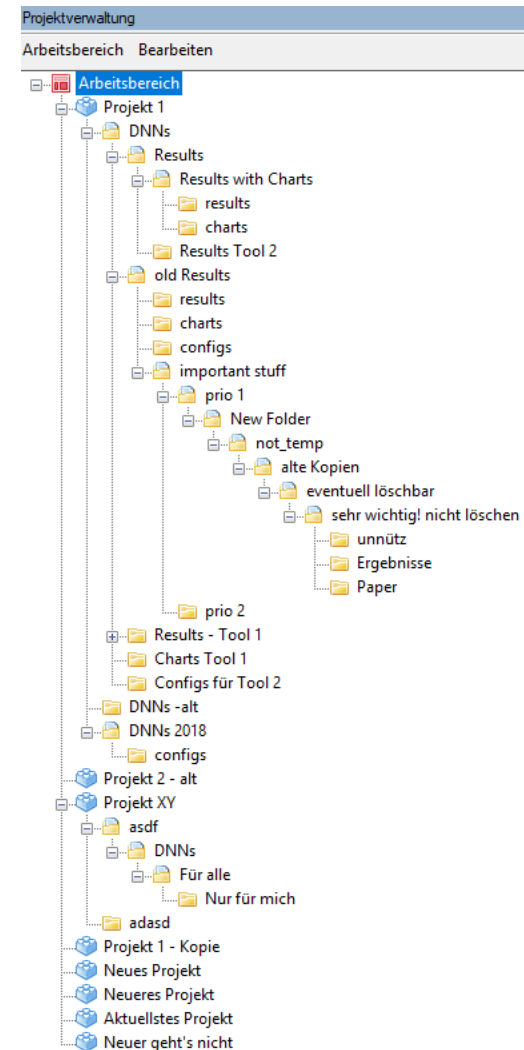
3. Vorstellung der Anwendung

4. Fazit und Ausblick

Problemstellung

- Arbeit mit Deep Neural Networks:
 - Verfügbare Tools divers
 - Großer und ggf. unterschiedlicher Parameterraum
 - Hunderte trainierte Netzwerke
 - Viele Ausgabedateien

➡ Übersicht zu behalten kann anstrengend sein



Lösungsansatz

Webfrontend „Administrative Neural Network Application“ (ANNA)

- Verwaltung von Projekten
- Speichern von Konfigurationsdateien und Ergebnissen
- Ausgaben vereinheitlichen
- Filterfunktion
- Bessere Übersicht
- Einfaches Teilen der Arbeit mit anderen



[2]

Gliederung

1. Einführung in die Thematik

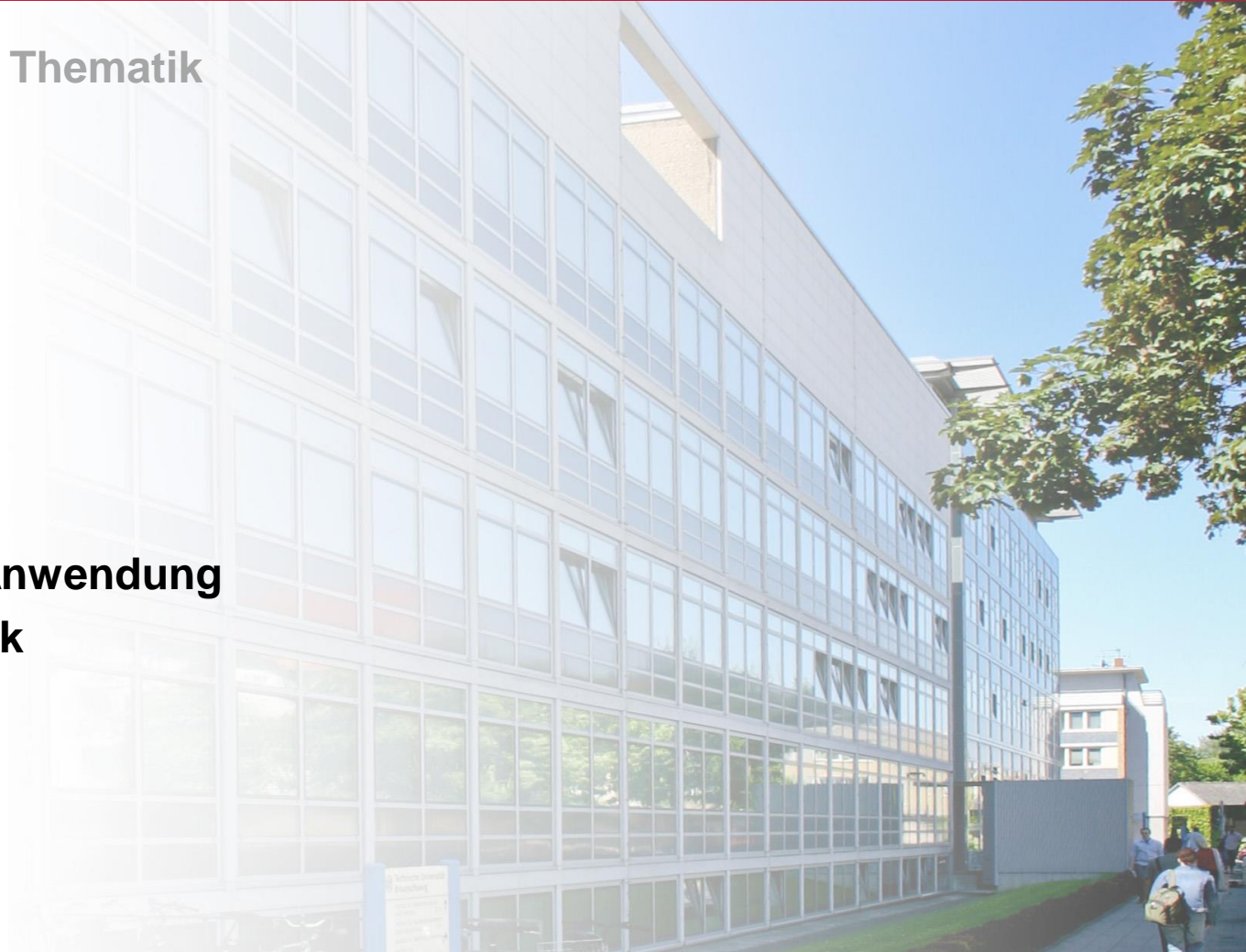
- Problemstellung
- Lösungsansatz

2. Aufbau

- TypeScript
- Serverseite
- Clientseite
- Datenbank

3. Vorstellung der Anwendung

4. Fazit und Ausblick



TypeScript

- Nutzung von JavaScript
- TypeScript als Hilfe
 - Klassen, Vererbung, Interfaces, anonyme Funktionen, etc.
- Java-ähnlich



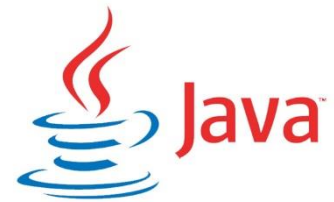
[3]

```
1  var Person = (function () {  
2  
3      function Person(name) {  
4          this.name = name  
5      }  
6  
7      Person.prototype.getName = function () {  
8          return this.name  
9      }  
10  
11      return Person  
12  })
```

TypeScript

[4]

```
1  class Person {  
2  
3      private name: string;  
4  
5      constructor(name: string) {  
6          this.name = name;  
7      }  
8  
9      public getName(): string {  
10         return this.name;  
11     }  
12  
13 }
```



[5]

```
1  public class Person {  
2  
3      private String name;  
4  
5      public Person(String name) {  
6          this.name = name;  
7      }  
8  
9      public String getName() {  
10         return this.name;  
11     }  
12  
13 }
```

[6]

Serverseite – Meteor

- Meteor auf Basis von NodeJS
- Einfache Adaption und schneller Start
- Plattformunabhängig
- TypeScript bzw. JavaScript
- Reaktiv mithilfe von Meteors Distributed Data Protocol
- Erweiterung mit Packages



[6]

+



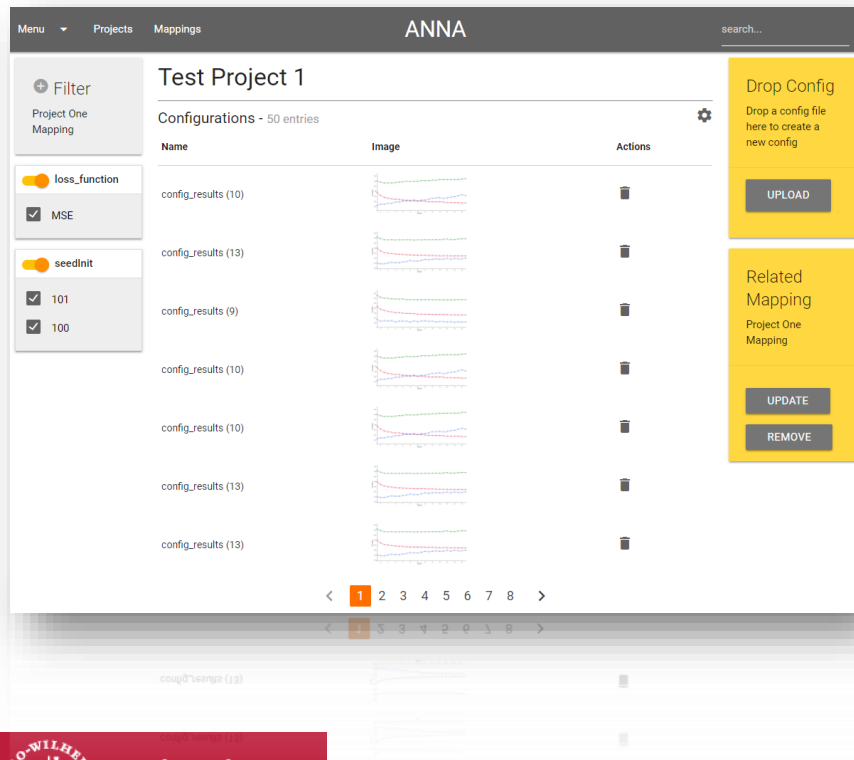
[7]

Clientseite – Angular

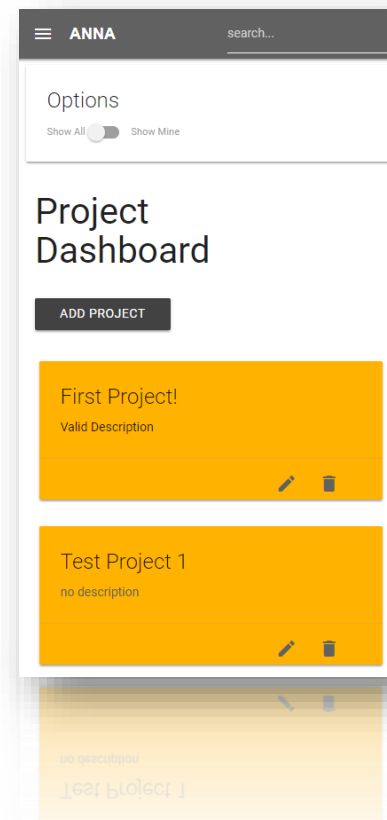
- Javascript Framework von Google
- Model-View-Controller Pattern
- Plattformunabhängig
- Reaktiv



[8]



[10]



[9]

Datenbank – MongoDB

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



[11]

```
{
  "_id" : "S7ZLXmpkQAzRxsKu6",
  "name" : "config_results (60)",
  "projectID" : "kHATnCoXLW8aB3eYf",
  "results" : [
    {
      "name" : "Training",
      "epochs" : [
        1.2240057732735,
        1.1969204147687,
        1.183273264675
      ]
    },
    {
      "name" : "Test",
      "epochs" : [
        1.3207248981952,
        1.3021228351597,
        1.2979549815629
      ]
    },
    {
      "name" : "Validation",
      "epochs" : [
        1.416297544425,
        1.405216438302,
        1.3958830064919
      ]
    }
  ],
  "creator" : "bm7MHBte86XwtPei8"
}
```

[12]

1. Einführung in die Thematik

- Problemstellung
- Lösungsansatz

2. Aufbau

- TypeScript
- Serverseite
- Clientseite
- Datenbank

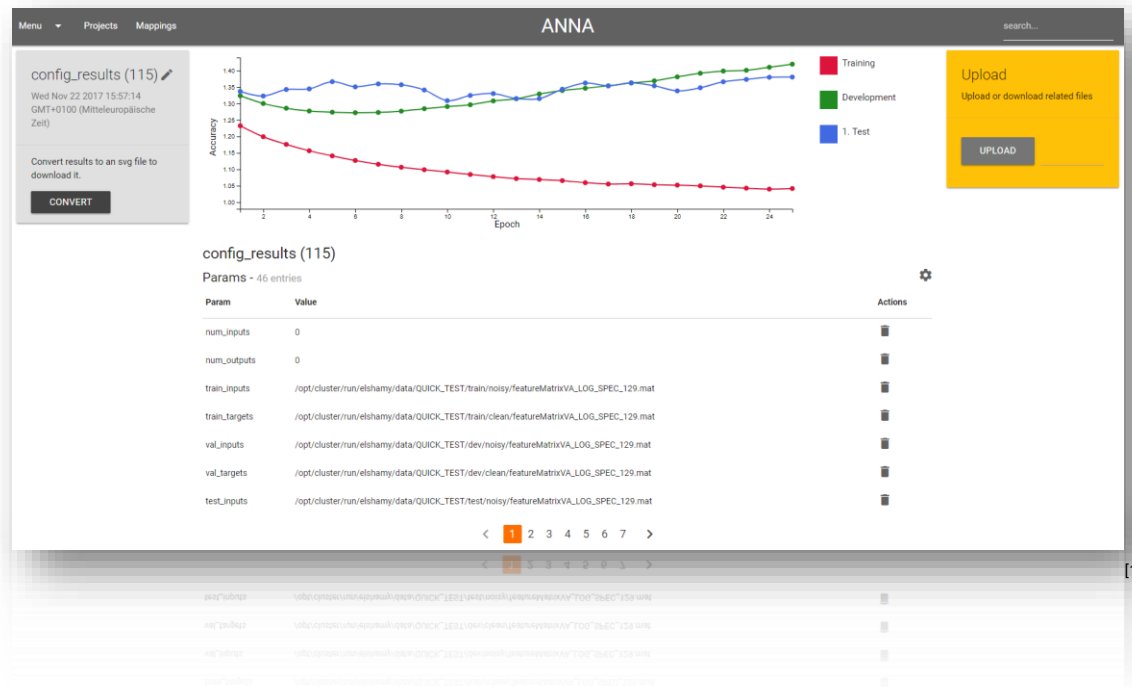
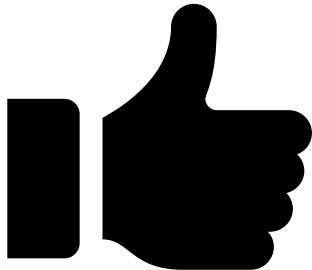
3. Vorstellung der Anwendung

4. Fazit und Ausblick



Fazit und Ausblick

- Schnelle, reaktive Anwendung
- Besserer Überblick
- Hilfreiche Filterfunktionen
- Einfach erweiterbar, z.B.:
 - GPU-Cluster Anbindung
 - Metastudien



[13]

Vielen Dank für Ihre Aufmerksamkeit.

Lukas Güldenhaupt

lukas.gueldenhaupt@tu-bs.de



Technische
Universität
Braunschweig



Institut für Nachrichtentechnik

Quellen

- [1] unübersichtliche Ordnerstruktur – erstellt von Lukas Güldenhaupt
- [2] vom Chaos zur Ordnung – erstellt von Lukas Güldenhaupt
- [3] JavaScript Logo – <https://commons.wikimedia.org/wiki/File:JavaScript-logo.png>
- [4] TypeScript Logo - <https://github.com/Microsoft/TypeScript/blob/master/doc/logo.svg>
- [5] Java Logo - <https://blog.newrelic.com/2014/04/04/java-developers-to-follow/>
- [6] Codevergleich (JavaScript, TypeScript, Java) – erstellt von Lukas Güldenhaupt
- [7] NodeJS Logo – <https://nodejs.org/en/>
- [8] Meteor Logo - <https://www.meteor.com/>
- [9] ANNA Dashboard – erstellt von Lukas Güldenhaupt
- [10] ANNA Projektseite –erstellt von Lukas Güldenhaupt
- [11] MongoDB Logo - <https://www.mongodb.com/>
- [12] Collection Eintrag – erstellt von Lukas Güldenhaupt
- [13] ANNA Configurationsdateiseite –erstellt von Lukas Güldenhaupt