Neural Networks: Architectures and Applications for NLP

Übungssitzung 1: Organisation und Orientierung

Julian Hitschler

ICL, Universität Heidelberg, WiSe 2016/17

27.10.2016

Inhalt

- Vorstellung
- Organisatorisches
- Voraussetzungen
- Implementierungsprojekte

Vorstellung

- Der Dozent
  - Julian Hitschler
  - zuständig für
    - komplementäre Übungssitzungen zum Vorlesungsteil der Veranstaltung
    - Betreuung der Implementierungsprojekte
    - Übungsaufgaben
  - `hitschler@cl.uni-heidelberg.de`
  - Deep-Learning-Anfänger (wie fast alle in der NLP-Community)
- Die Studierenden

Organisatorisches

- Zeit: Donnerstag, 09:15 - 10:45
- Raum: INF 346, SR 10
- Sofern verfügbar, bitte eigene Laptops mitbringen

Voraussetzungen

- Grundkenntnisse Programmieren
  - Grundwissen `Python`
  - zum Beispiel aus Programmieren I
- Grundlagen der linearen Algebra
  - Notation für Matrizen und Vektoren
  - grundlegende Operationen auf Matrizen und Vektoren
- Grundlagen der Analysis
  - (partielle) Ableitungen
  - Gradienten(-felder)

Implementierungsprojekte

- Ziel: Ein trainierbares neuronales Netzwerk selbst implementieren und damit Experimente durchführen
- eher keine Gruppenarbeit
  - selbständiges Verständnis der Materie und Arbeiten mit Frameworks
  - begründete Ausnahmen bitte frühzeitig absprechen
- zwei Optionen
  1. eigene Implementierung eines einfachen NN
  2. Implementierung eines mäßig komplexen NN in einem Framework
- Experiment-Skript
- Dokumentation

Implementierungsprojekte

1. Eigene Implementierung eines NN
   - Programmiersprache nach Wahl: `C++`, `Python`, `Java`, ...
   - keine Verwendung von Deep-Learning Frameworks und Bibliotheken
   - keine Verwendung von Bibliotheken für lineare Algebra, symbolische Ableitung etc.
   - nur Verwendung von Standardbibliotheken (soweit möglich)
   - Lösung eines einfachen Klassifizierungsproblems
     - NIST-Datensatz (Ziffernerkennung)
     - einfacher Datensatz zur Dokumentklassifikation
     - oder ähnliches
   - mindestens ein hidden Layer
   - Gradientenabstieg/Backpropagation kann architekturspezifisch implementiert sein
   - Mindestumfang der Experimente: Vergleich von zwei einfachen NN-Architekturen (z.B.: Restricted Boltzmann Machine vs. Multilayer-Perceptron)

Implementierungsprojekte

2. Implementierung einer NN-Architektur in einem Framework
   - Auswahl eines Frameworks: `Theano` oder `TensorFlow`
   - Implementierung eines RNN-Sprachmodells oder einer vergleichbar komplexen Architektur
     - mindestens einmal Rekursion oder Rekurrenz als Teil der Netzwerkarchitektur
     - Verwendung von RNN-Modulen in `TensorFlow` ist erlaubt!
     - Lösung eines Sequenzklassifikations- oder (probabilistischen) Sequenzmodellierungsproblems
   - Preprocessing für Korpora in Standardformaten (Textdateien, CSV, XML, ConLL, etc.)
     - Austausch des Trainingskorpus muss für "den Laien" (Julian) möglich sein

Implementierungsprojekte

2. Implementierung einer NN-Architektur in einem Framework
   - maximale Trainingszeit: 48 Stunden auf 10 Cores auf `last`
     - *hartes Limit!*
     - kleine Layer
     - flaches Netzwerk
     - kleiner Datensatz
     - sinnvolles Minibatching
     - effiziente Implementierung
     - gute Hyperparameter wollen experimentell gefunden werden
   - keine Verwendung (Copy&Paste) von Tutorial-Code
     - Dozentin und Dozent haben übersinnliche Fähigkeiten und könnes dies erspüren
     - es ist nicht empfehlenswert, Funktionen der Frameworks zu verwenden, die man nicht genau versteht
   - Projekt kann ein Einstiegspunkt für eigene Forschung (Abschlussarbeit, etc.) sein!

Implementierungsprojekte

- Projektvorschlag
  - kurze Zusammenfassung des geplanten Projekts (1-2 Seiten)
  - bis 16.12.2016 per Mail an `hitschler@cl.uni-heidelberg.de` (Betreff: "Projekt NNAA")
  - wer im Seminarteil einen Vortrag hält, muss kein Projekt implementieren, darf dies aber gerne tun
- Dokumentation des Projekts
  - README (DozentInnen-sicher)
  - 1×`Enter`-Experimentskript (Shell-Skript)
  - Dokumentation des Systems und der Experimente als Short Paper (4 Seiten + Quellen)
- Abgabetermin (Mail an Julian) Code+Dokumentation: 31.3.2017,23:59

Noch Fragen?

Vielen Dank für die Aufmerksamkeit!