



HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT  
DRESDEN

SPRACHVERARBEITUNG MIT PYTHON  
PROJEKTSEMINAR

# Textgeneration mithilfe von Keras und Tensorflow

## Dokumentation

*Christian Grieb*

unter Aufsicht von  
Prof. Dr. rer. nat. Boris HOLLAS

11. März 2024

# Inhaltsverzeichnis

# 1 Aufgabenstellung

Es soll als Maschinelles Lernverfahren ein zeichenbasiertes, Rekurrentes Neuronales Netzwerk erstellt werden, welches anhand von selbst gewählten und beschafften Trainingsdaten das nächste Zeichen einer Sequenz errät.

Who is the Audience? (scientific background or no point of contact before or just a bit?)

## 2 Werkzeuge

Programmiersprache	Python
Open-Source Bibliotheken	Keras Tensorflow

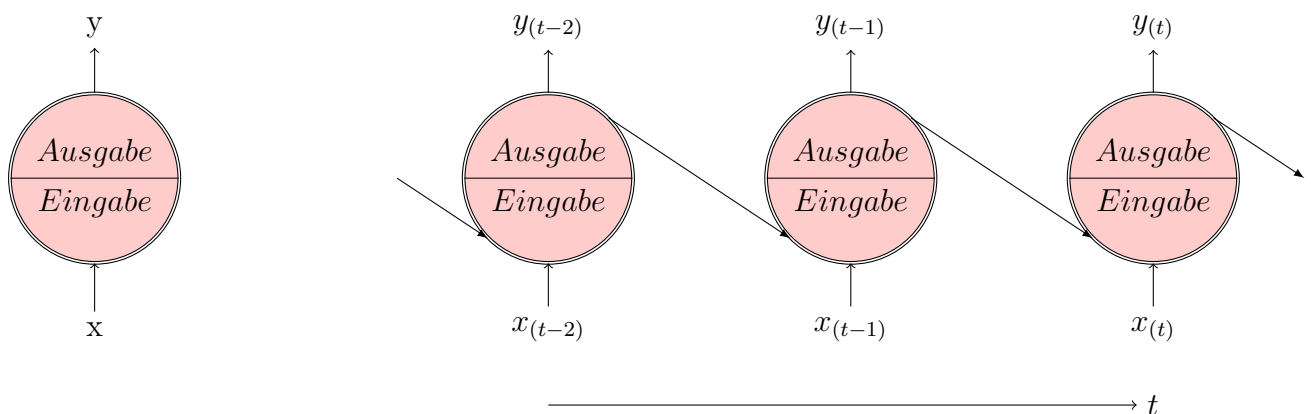
## 3 Was ist Maschinelles Lernen?

Maschinelles Lernen ist das Fachgebiet, das Computern die Fähigkeit zu lernen verleiht ohne explizit programmiert zu werden

- Arthur Samuel 1959

## 4 Was sind Rekurrente Neuronale Netzwerke?

Ein Rekurrentes Neuronales Netzwerk hat im gegensatz zu anderen neuronalen Netzen (z.B. Feed-forward Neural Network) Rückwärts gerichtete Neuronen. Die einfachste Variante besteht aus nur einem Neuron, dass Eingaben erhält, eine Ausgabe produziert und diese Ausgabe wieder an sich selbst schickt.



Einzelnes Neuron links

entlang der Zeitachse aufgerollt rechts

Durch die Rückkopplungen können lokale Optima entstehen, die zu unübersichtlichen Netzzuständen führen. Rekurrente neuronale Netze sind nicht in der Lage, weit entfernte oder zurückliegende Informationen miteinander zu verknüpfen. Netze mit sogenannten LSTM-Zellen (Long Short Term Memory) lösen dieses Problem, indem die Zellen neben einem Eingang und einem Ausgang ein Merk- und Vergesstor besitzen. Es entsteht eine Art Kurzzeitgedächtnis, das Informationen relativ lange vorhalten kann und sich an früher gemachte Erfahrungen erinnert.

Rekurrente neuronale Netze eignen sich im Gegensatz zu den Feedforward-Netzen nicht nur für die einfache Mustererkennung. Sie sind in der Lage, komplexere Sequenzen zu verarbeiten.

## 4.1 ML-Modell

< *Grafik* >

Grobe Beschreibung, was, warum, in welcher Reihenfolge gemacht wird.

### 4.1.1 GRU-Layer

Why GRU (Vorteile, Nachteile)

Verallgemeinerte(vereinfachte) Grafik (Update Gate, Reset Gate, Current Memory)

Grafik Update Gate

etc.

### 4.1.2 LSMT-Layer

Placeholder

### 4.1.3 Dense-Layer

Placeholder

### 4.1.4 Embedding-Layer

Placeholder

## 5 Codevorstellung und Beschreibung(*Einzelschritte?*)

Placeholder incl. Auswertungsgrafiken

**6 Discussion**

**7 Summary**

**8 References**