Neuronale Netze und einige ihrer Anwendungen

Timo Baumann

OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

Lehrerfortbildung NeuNe 11/2023

Sequenzdatenverarbeitung mit Neuronalen Netzen

OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

Inhalt 2. Block

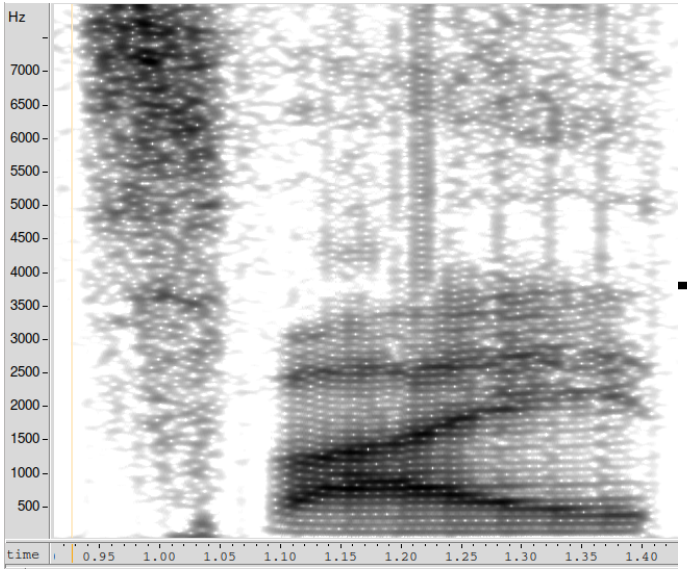
- Rekurrente Neuronale Netze

  aggregieren variabel lange Sequenzen zu Vektoren fixer Länge

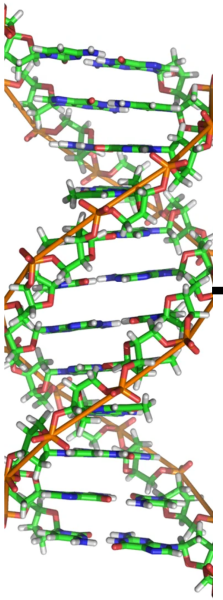
- Klassifikation von Sequenzen

- Beispiel(e) in PyTorch

Sequenzklassifikation



- happy
- sad
- **surprised**
- angry



- useless
- low potential
- **high potential**

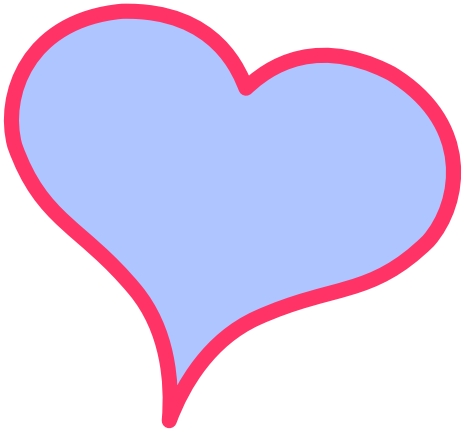
Lorem ipsum dolor sit amet, consectetur adipisici elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat.

- echter Text
- **Blindtext**

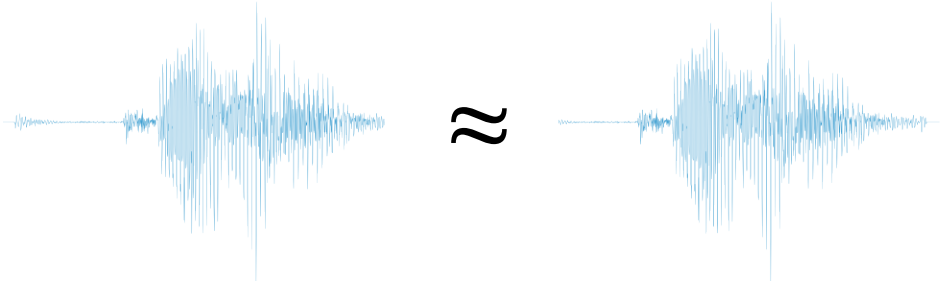
Was ist überhaupt das Problem?

- Sequenzen haben beliebige Längen (auch für dasselbe Problem sind sie häufig nur zufällig gleichlang)
- Neuronale Netze haben fixe Anzahl Eingabeneuronen und fixe Anzahl Ausgabeneuronen

Was ist die Chance?



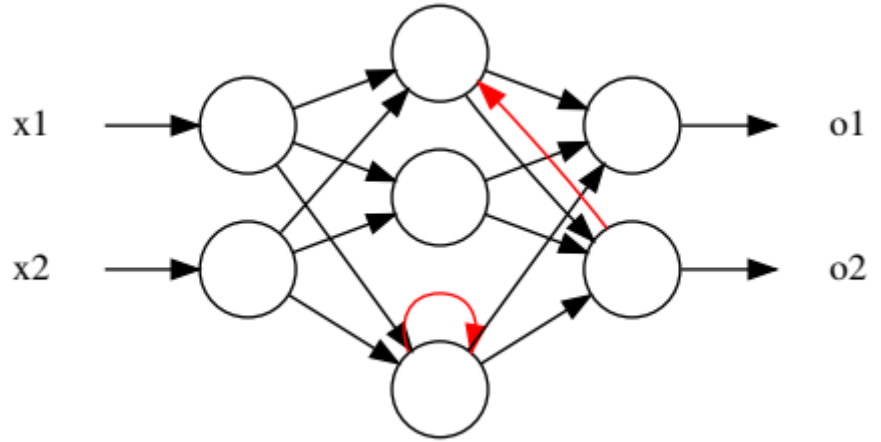
- Sequenzen können sich abschnittsweise sehr ähnlich sein
- Sequenzen können einfach verschoben sein



→solche Ähnlichkeiten sollten beachtet werden

Rekurrente Neuronale Netze

Rekurrentes Netz?

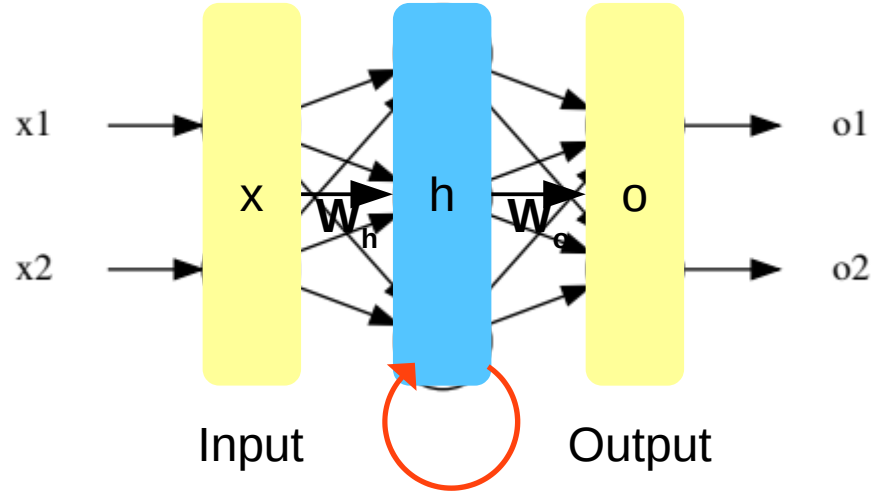


Vorwärtskanten

Rekurrenzen

- Vorwärtseigenschaft erlaubt Aktivierungen zu berechnen und mittels *Backpropagation* und Gradientenabstieg zu lernen
- Rekurrenzen/Rückkopplungen verhindern dies (zunächst) weil sie Zyklen in den Berechnungsgraph einfügen

Schichtendarstellung

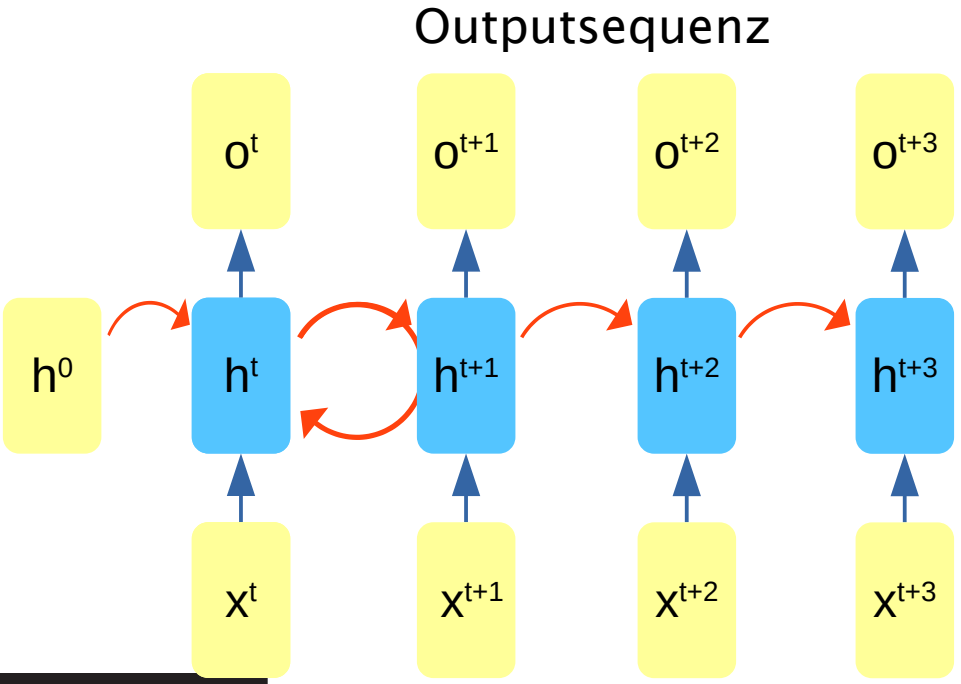


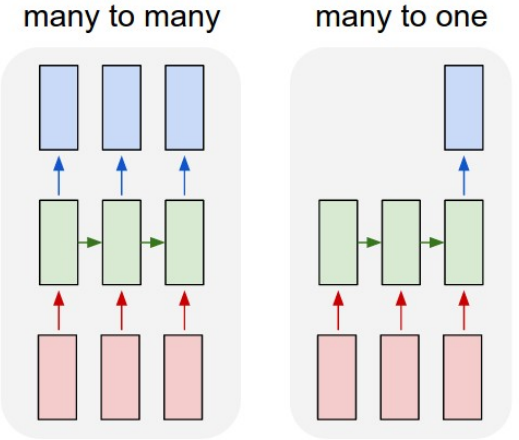
Rekurrenz

- typischerweise stellen wir ganze Neuronenschichten dar
- Pfeil: Eingang in die Schicht
- innerhalb der Schicht (z.B. für h): $\mathbf{h} = \tanh(\boldsymbol{W}_h \cdot \boldsymbol{x} + \boldsymbol{b}_h)$

Zeitschritte

- Betrachtung des rekurrenten Netzes über diskrete Zeitschritte





- Rückkopplungen sind verschwunden!
- $\mathbf{h}^{t+1} = \tanh(\boldsymbol{W}_{xh} \cdot \boldsymbol{x}^{t+1} + \boldsymbol{W}_{hh} \cdot \boldsymbol{h}^{t} + \boldsymbol{b}_{h})$
- die Parameter **W** und **b** sind allen Zeitschritten gemein!
- Parameter sind unabhängig von Anzahl der Zeitschritte

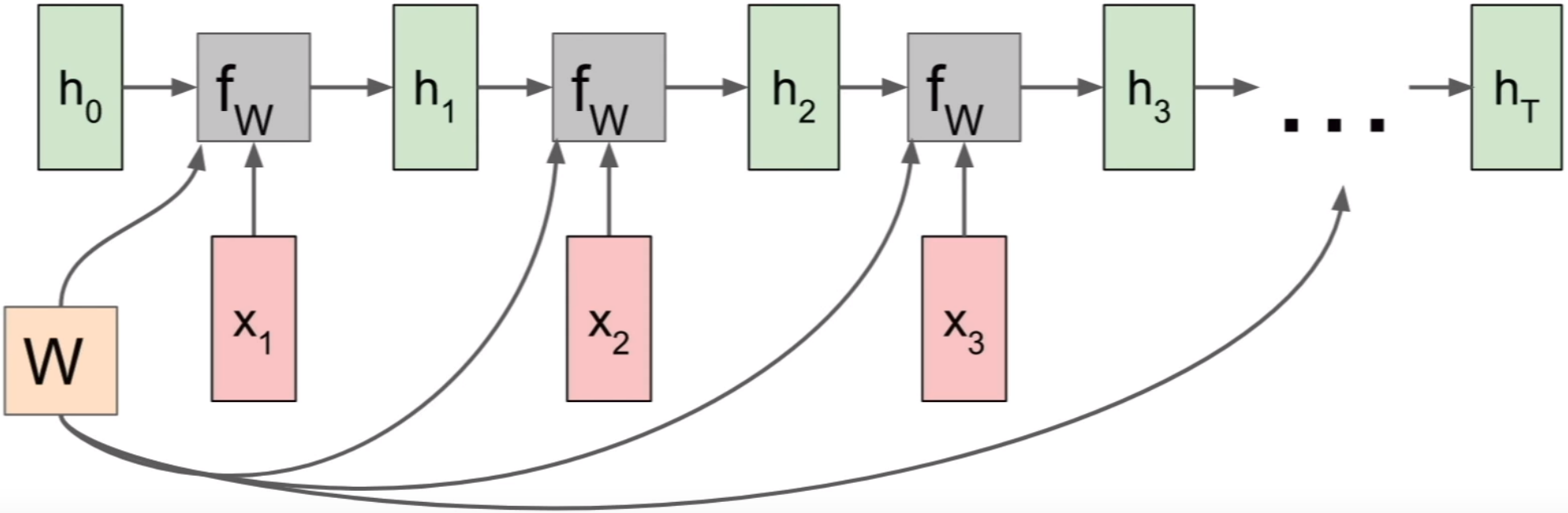
Kurzes Zwischenfazit

- Rekurrentes Netz:
  - Rückkopplungen von Neuronen mit sich selbst
  - Neuronenwert ist abhängig von sich selbst
    - das ist erstmal schlecht
    - von sich selbst *in einem früheren Zeitschritt* →abrollen über die Zeit
  - ermöglicht Behandlung beliebig langer Sequenzen (in der Zeit)

Berechnungsgraph eines RNN

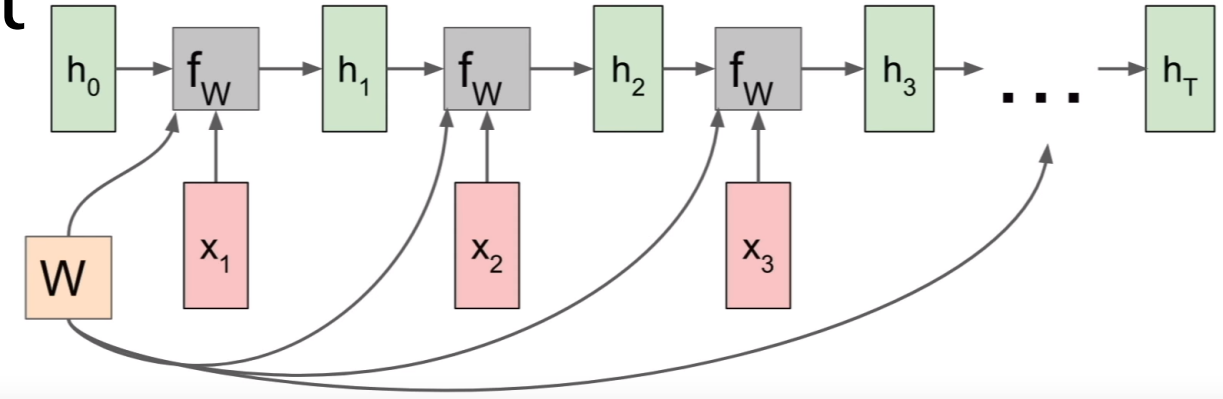
- zeichnen Sie den Berechnungsgraphen eines RNNs
  - die Eingabesequenz besteht aus 4 Elementen x_1 ... x_4
  - die Ausgabesequenz vernachlässigen wir

Berechnungsgraph eines RNN



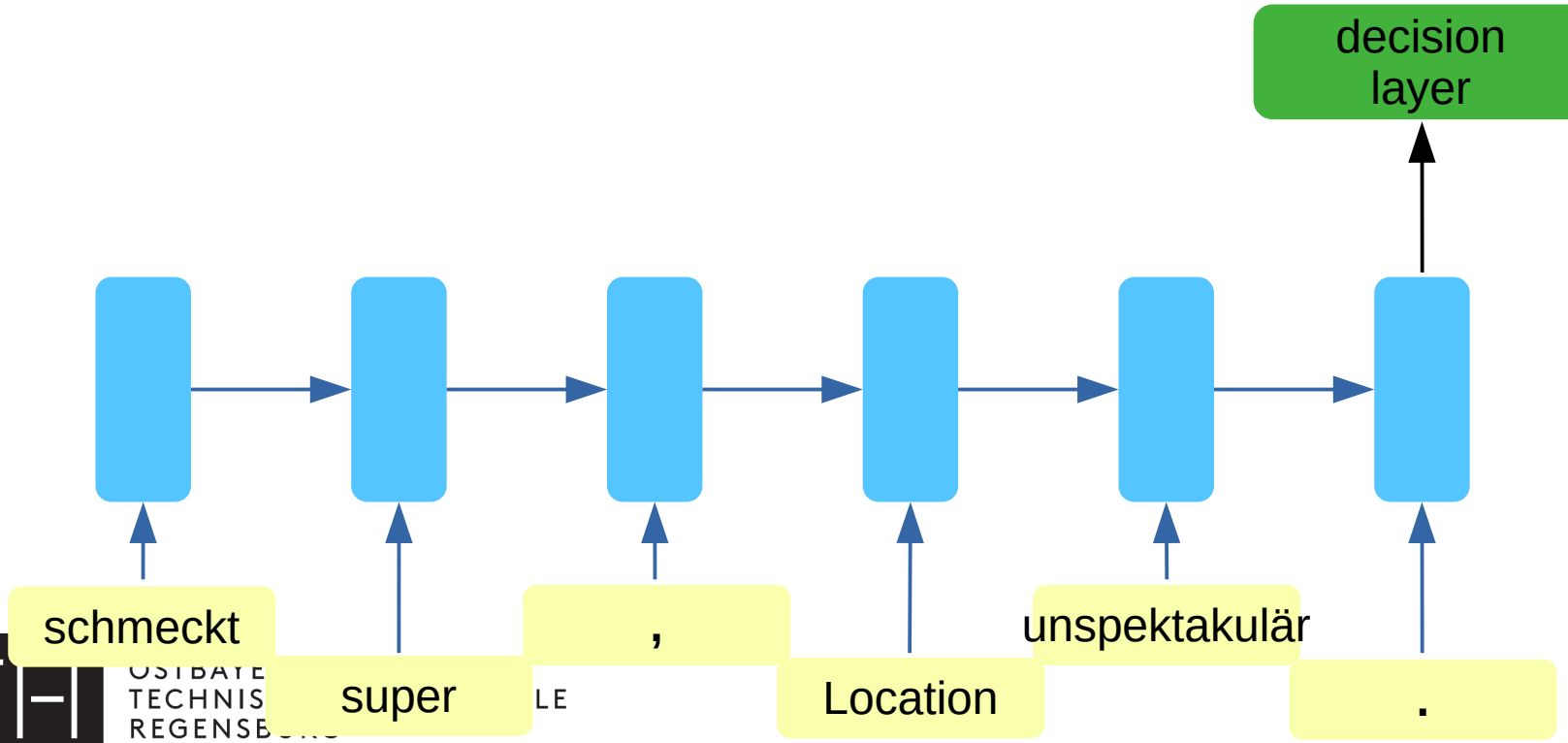
Berechnungsgraph eines RNN

- die Tiefe des neuronalen Netzes wird flexibel der Länge der Eingabesequenz angepasst
  - “abrollen über die Zeit” des rekurrenten Netzes
- dieselben Parameter W werden zu jedem Zeitschritt benutzt
  - unabhängig von der Sequenzlänge



RNNs als Encoder

- Beispiel: Sentiment Classification



Neuronale Textverarbeitung

z.B.

- Klassifikation von Texten
- Klassifikation der Wörter in einem Text (z.B. Wortarten annotieren, Orte/Namen herausfinden, ...)
- Ähnlichkeit von Texten

Text als Sequenzdaten

- diskrete Eingabesymbole (zeitdiskret)
  - Wörter, Buchstaben, oder “Tokens”
- Spezifika von Sprache:
  - Häufigkeit der Symbole ist sehr ungleich verteilt
  - Symbole sind sich unterschiedlich ähnlich
- Lösung: Tokenisierung und Embeddings

Tokens statt Wörter

- seltene Wörter aus Teilen zusammensetzen:
  - Regensburg = Reg + ens + burg
- statistisches Verfahren zur Distillation der Tokens aus Zeichenfolgen
  - keine linguistische Analyse (Regen + s + burg)
  - keine besondere Betrachtung von Leer- oder Satzzeichen (alles Teil der Tokens)

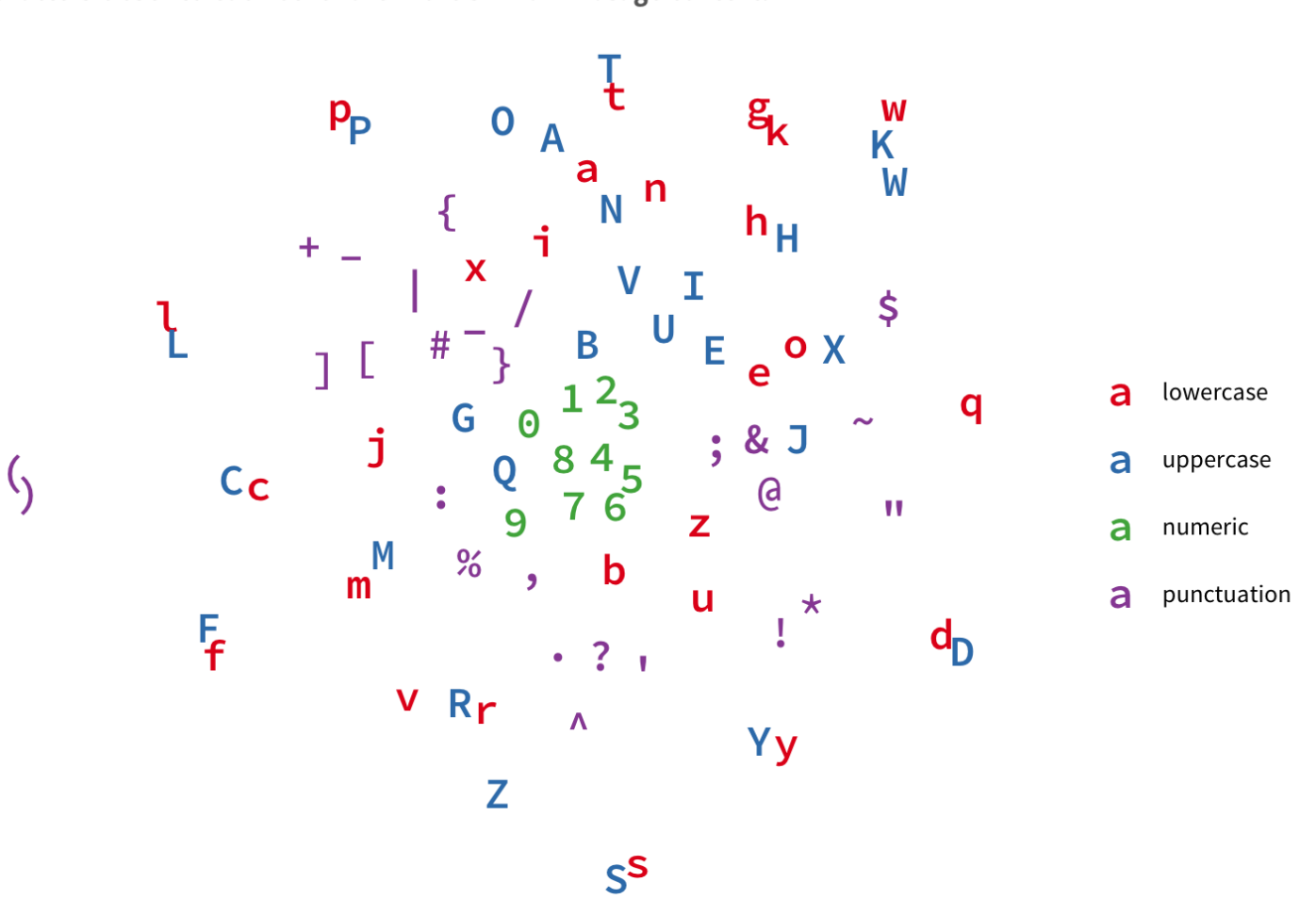
Heute ist der Himmel blau

https://platform.openai.com/tokenizer

Embeddings

- one-hot-Encoding von Wörtern ist ineffizient
  - Wortähnlichkeit entfällt
  - hochdimensionaler Input bei großen Vokabularen
- Idee:
  - Einbettung in vglw. niedrigdimensionalen Raum (z.B. 300 Dimensionen für Wörter, 16 für Buchstaben)
  - Position im Raum wird z.B. gleichzeitig mit dem NN trainiert (alternativ: vorab, z.B. mit zusätzlichen Daten)

Beispiel: Buchstaben-Embedding



ein paar Verfeinerungen

Generelle Probleme mit RNNs

Lernen in RNNs ist

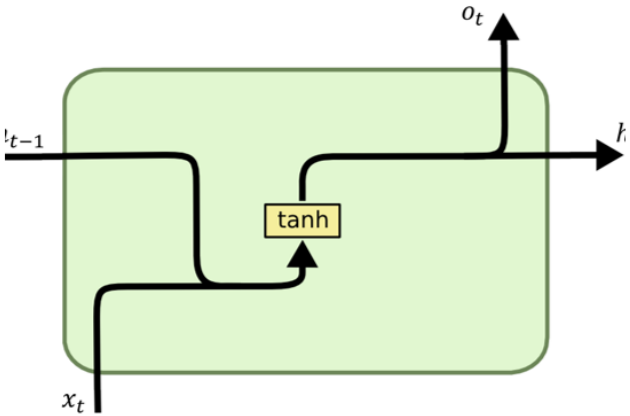
- insbesondere bei langen Sequenzen
- wenig effektiv (vanishing / exploding gradient-Problem)
- langsam (wenig Parallelisierung, da das Netz sehr tief wird)

Gegenmittel

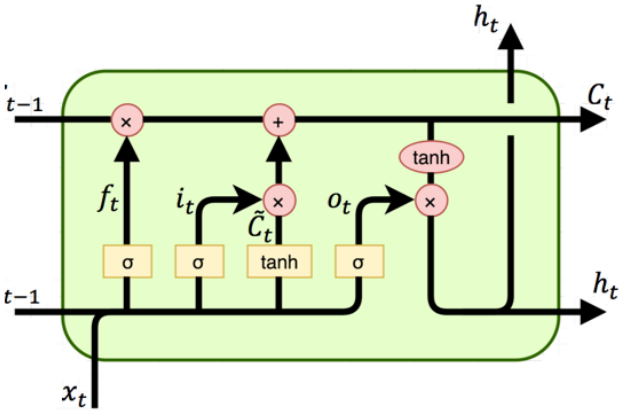
- Repräsentation durch kürzere Sequenzen: Wörter statt Zeichen, Merkmalsextraktion mit Fensterung, ...
- alternative rekurrente Einheiten (LSTM, GRU, ...)
- rekurrente Verbindungen “kappen”: Transformer

Alternative rekurrente Einheiten

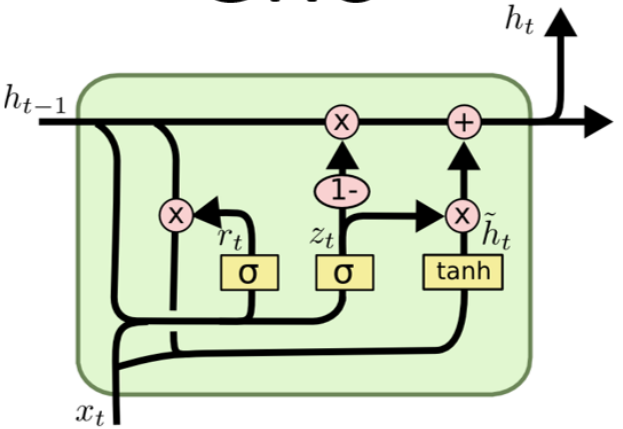
RNN



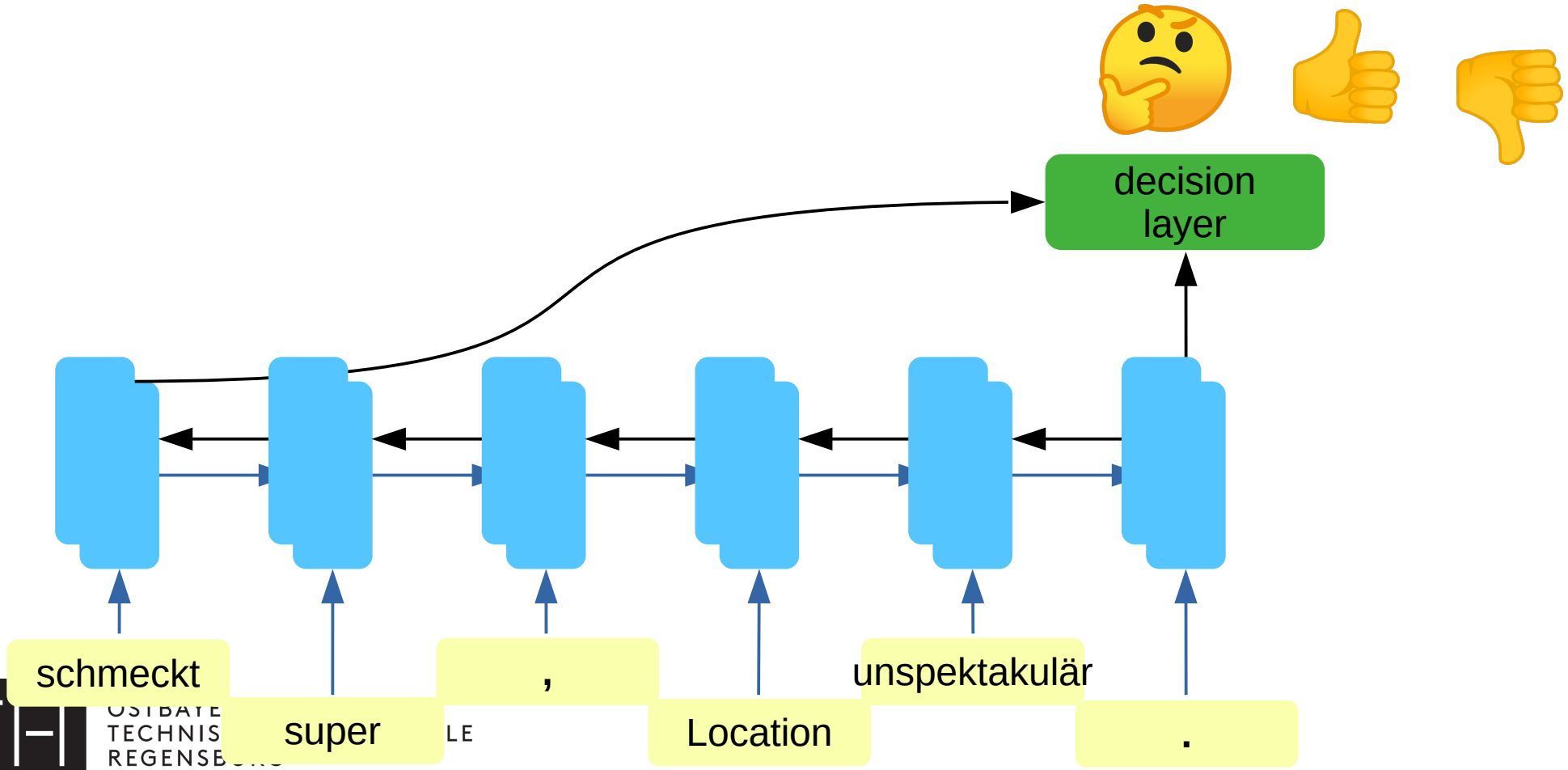
LSTM



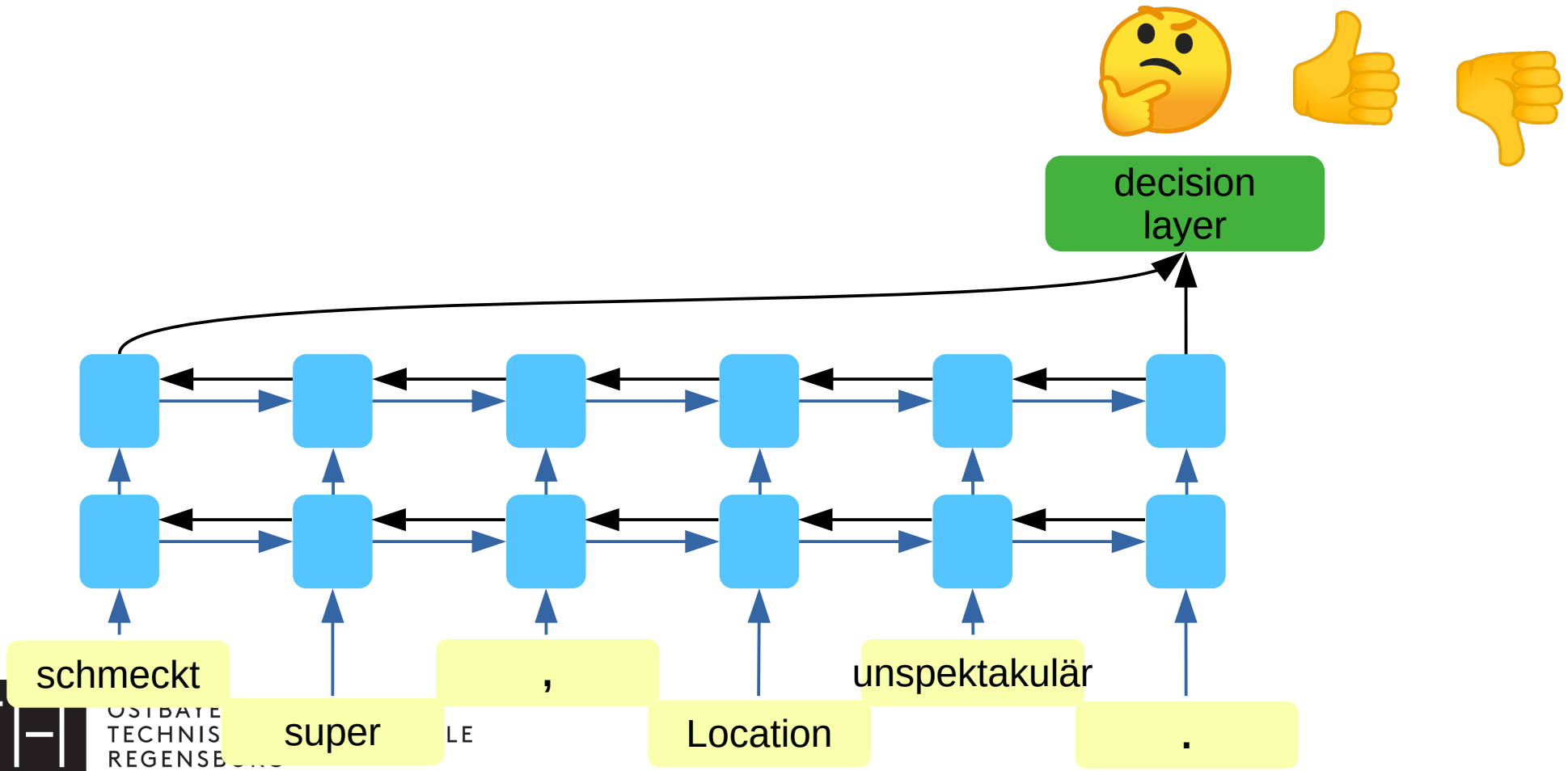
GRU



Bi-RNNs als Encoder



Stacked (Bi-)RNNs als Encoder



kontinuierliche Signale als Sequenzdaten

- empfehlenswert:
  - feste Taktung (soweit keine fachlich passendere Einteilung möglich scheint)
  - Fensterung des Signals und Repräsentation der Fenster ins RNN füttern
    - bekannt “sinnvolle” Vorverarbeitungsschritte zur Dimensionsreduktion (FFT?, Grundfrequenz-exktraktion, Maxima, Minima, Mittelwerte, ...) zuerst probieren
    - ggfs. CNNs zur Aggregation innerhalb des Fensters lernen
  - Sie haben zusätzlich symbolische Information? auch benutzen!

Learning to Determine Who is the Better Speaker

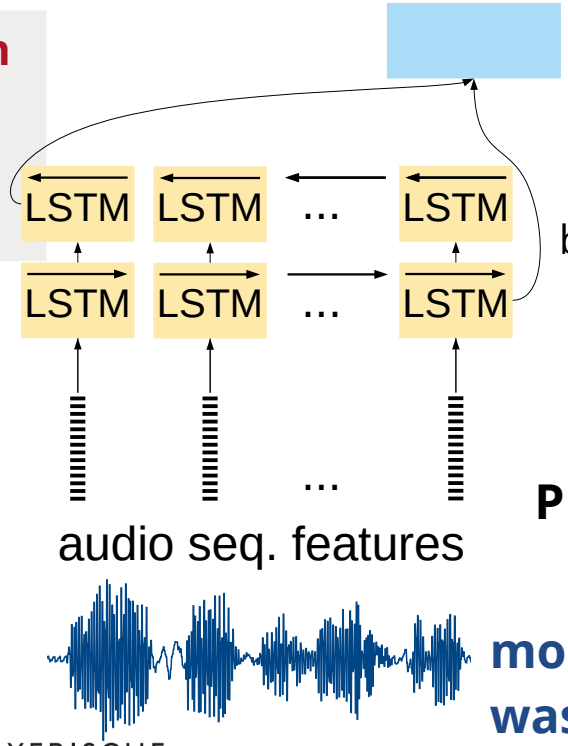
Timo Baumann

OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG

Carnegie Mellon University
Language Technologies Institute

RNN modeling of the speech sequence

gets fixed-length representation from variable-length sequence



a multi-dimensional representation of the stimulus quality

bi-directional LSTMs work better than uni-directional LSTMs.

not easily interpretable. Sorry.

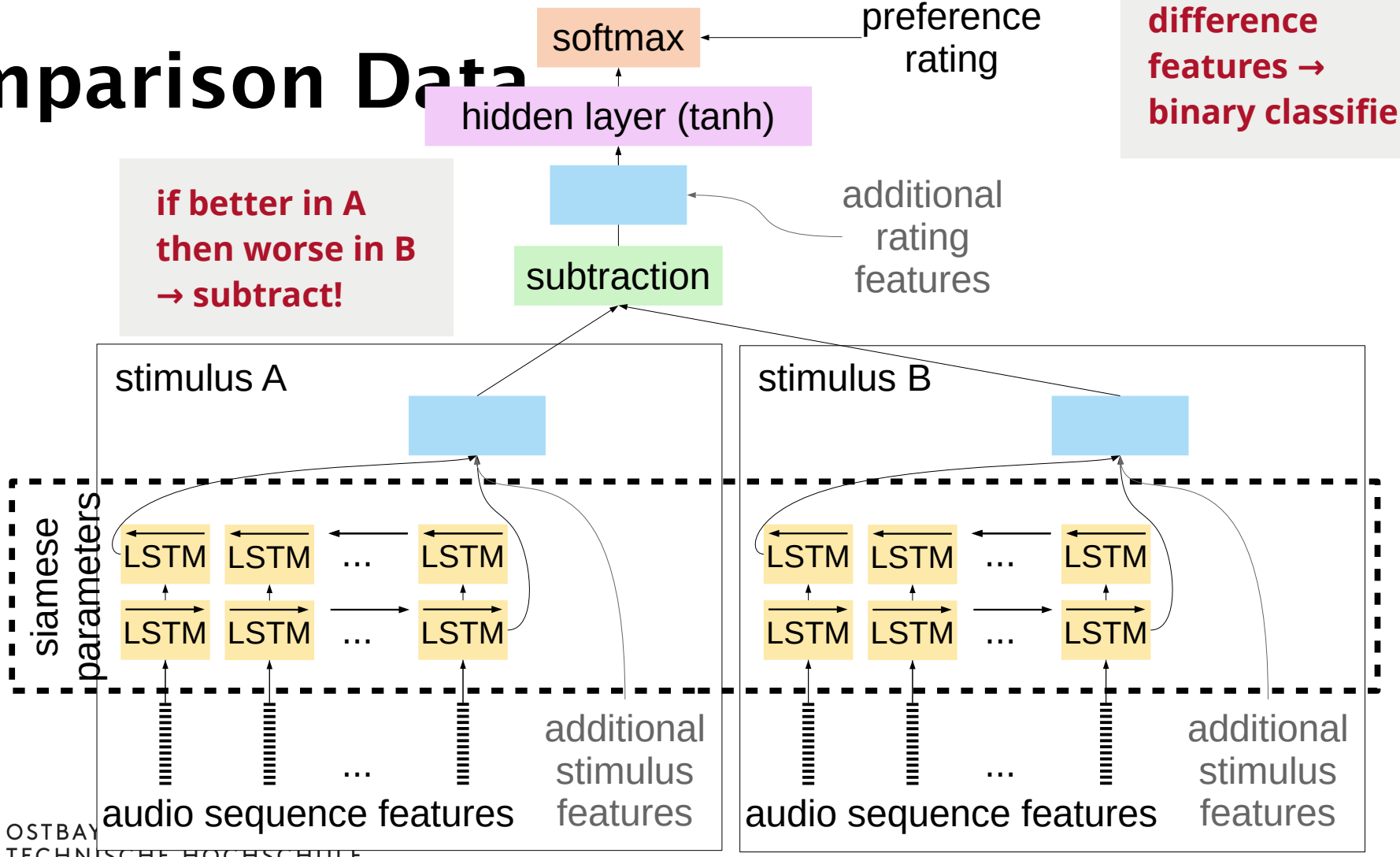
MFCC + FFV
PPQ,APQ,HMN
+ phone ID

almost as simple as you could imagine

e.g., I was too lazy to z-normalize

model can relate prosody to what was spoken (beyond *Gestalt*)

Comparison Data



Bromley et al. 1994 (NIPS)

Zusammenfassung

- Rekurrentes Netz:
  - Neuronenwert ist abhängig von sich selbst *in einem früheren Zeitschritt* →abrollen über die Zeit

- ermöglicht Behandlung beliebig langer Sequenzen mit fixer Anzahl Parametern

Vielen Dank! Ihre Fragen?

timo.baumann@oth-regensburg.de