Grundlagen der Künstlichen Intelligenz

Wintersemester 25/26

Vorlesung 10

Rekurrente Neuronale Netze

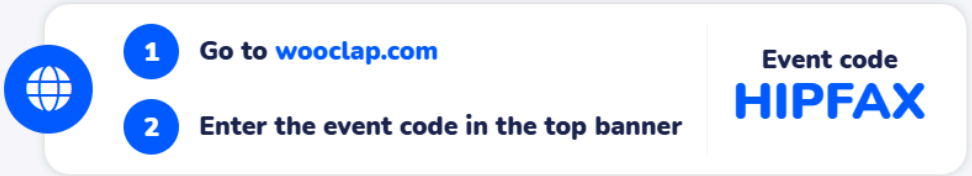
Anwendung: Natural Language Processing

12. Januar 2026

T.T.-Prof. Dr. Peer Nowack
Prof. Dr. Pascal Friederich

Wir werden immer mal wieder Umfragen machen...

Sie können sich bereits einloggen. Nutzen Sie das KIT Wi-Fi bei schlechtem Empfang.

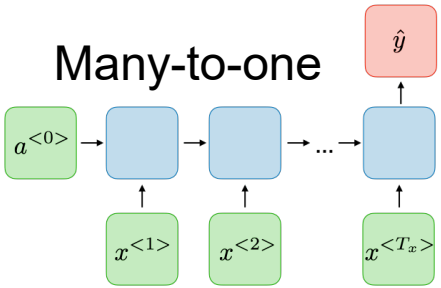


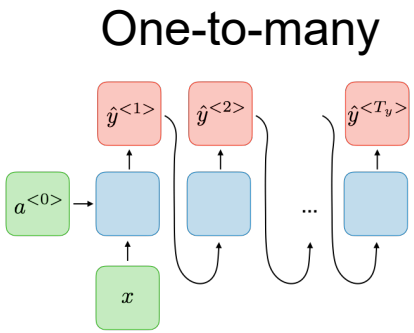
You cannot vote anymore

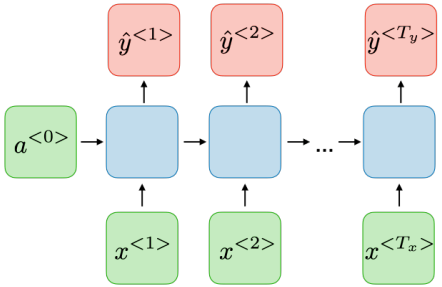
Welche der folgenden Beispiele sind geordnete Sequenzdaten (alle richtigen Antworten ...

1	Messdaten von Lufttemperatur in Karlsruhe	72%	73 ✓
2	Videos	78%	79 ✓
3	Fotos von handgeschriebenen Postleitzahlen	14%	14
4	Tonspur eines Podcasts	90%	91 ✓
5	Der Text eines Zeitungsartikels	82%	83 ✓
6	Eine Zeitreihe für ein Regressionsproblem in dem Samples zur Kreuzvalidierung gemischt wurden.	26%	26

Sequenzdaten (Textdaten, Zeitreihen von Messdaten, ...)



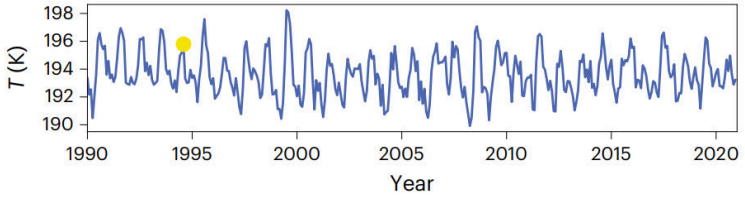


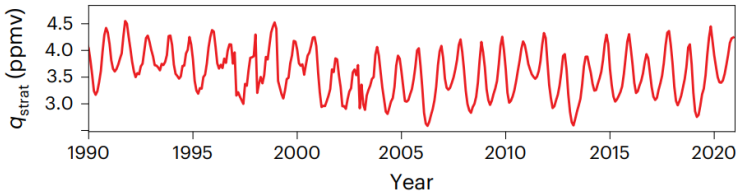


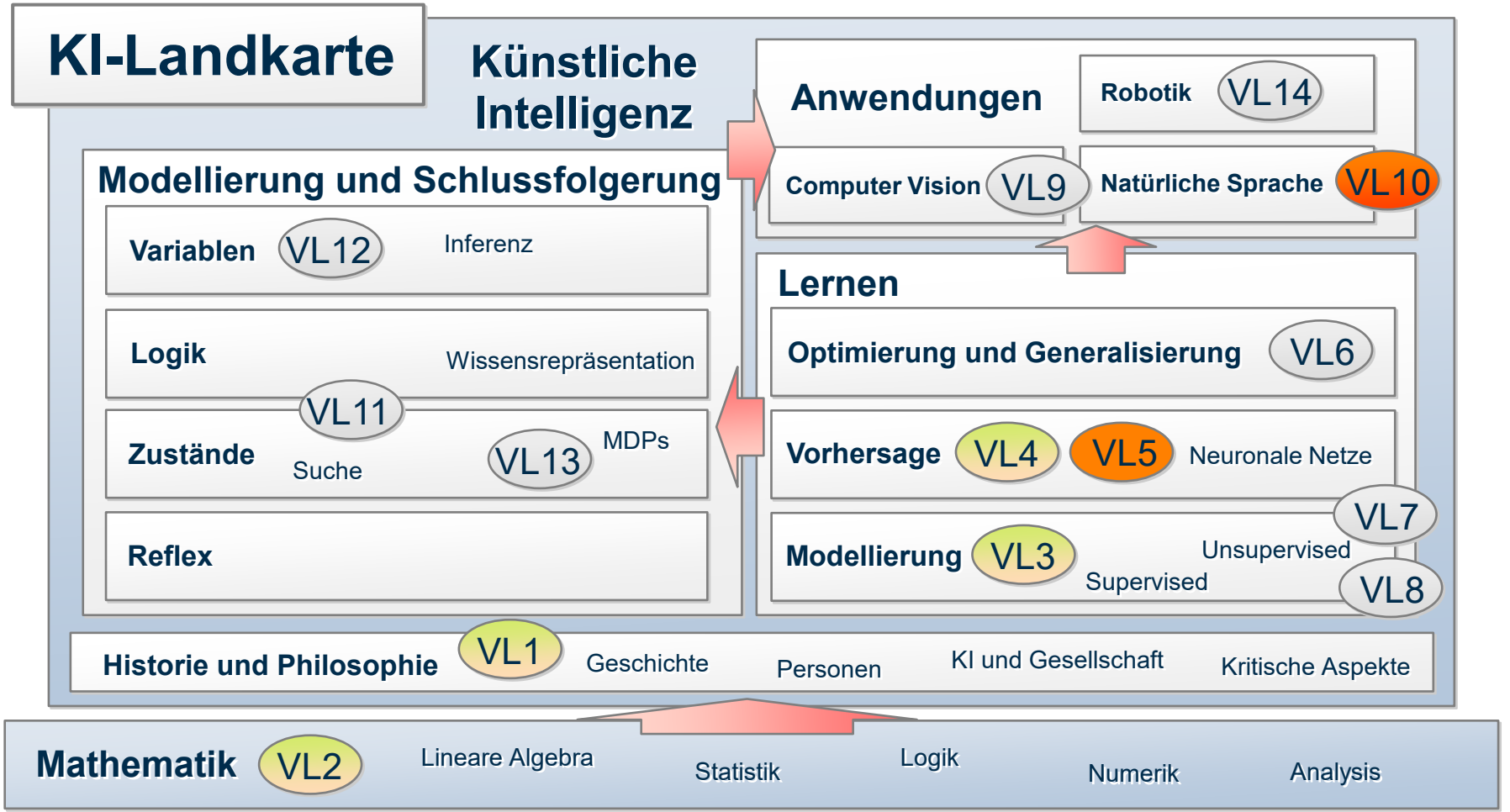
Many-to-many

Stanford University
Cheng et al. (2016)

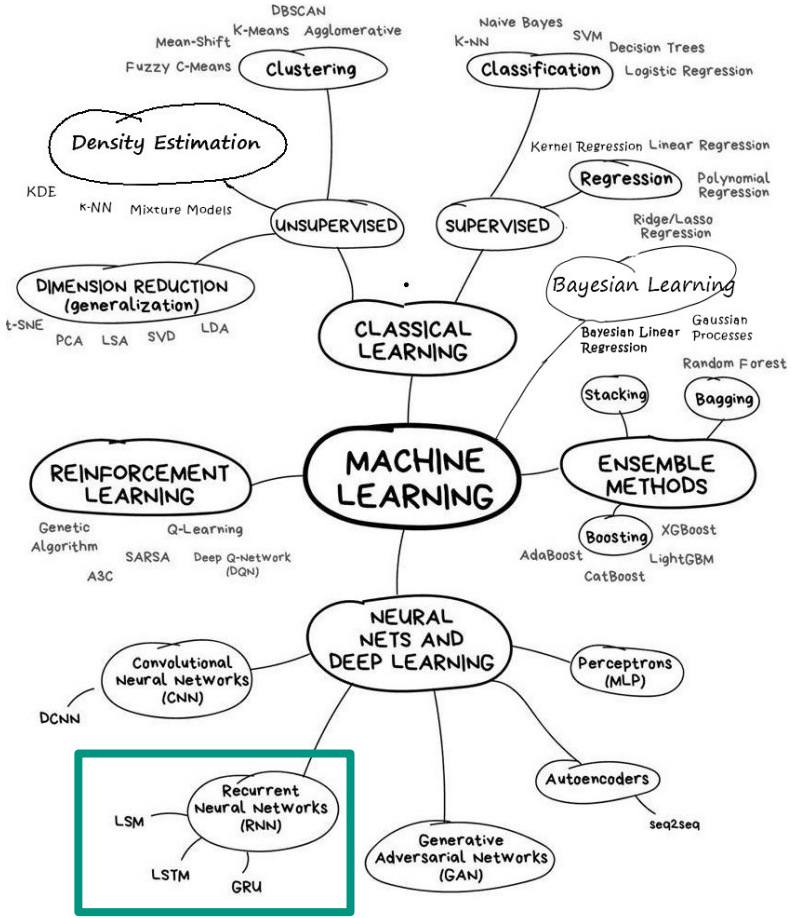
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .






KI-Landkarte
Künstliche Intelligenz
Anwendungen
Robotik
VL14
Computer Vision
VL9
Natürliche Sprache
VL10
Modellierung und Schlussfolgerung
Variablen
VL12
Inferenz
Logik
Wissensrepräsentation
VL11
Zustände
Suche
VL13
MDPs
Reflex
Lernen
Optimierung und Generalisierung
VL6
Vorhersage
VL4
VL5
Neuronale Netze
VL7
Modellierung
VL3
Unsupervised
Supervised
VL8
Historie und Philosophie
VL1
Geschichte
Personen
KI und Gesellschaft
Kritische Aspekte
Mathematik
VL2
Lineare Algebra
Statistik
Logik
Numerik
Analysis

Der ML-Zoo...


DBSCAN
K-Means
Agglomerative
Mean-Shift
Fuzzy C-Means
Clustering
Naive Bayes
K-NN
SVM
Decision Trees
Classification
Logistic Regression
Density Estimation
KDE
K-NN
Mixture Models
UNSUPERVISED
SUPERVISED
Kernel Regression
Linear Regression
Regression
Polynomial Regression
Ridge/Lasso Regression
DIMENSION REDUCTION (generalization)
t-SNE
PCA
LSA
SVD
LDA
CLASSICAL LEARNING
Bayesian Learning
Bayesian Linear Regression
Gaussian Processes
Random Forest
Stacking
Bagging
REINFORCEMENT LEARNING
MACHINE LEARNING
ENSEMBLE METHODS
Genetic Algorithm
A3C
Q-Learning
SARSA
Deep Q-Network (DQN)
Boosting
XGBoost
AdaBoost
LightGBM
CatBoost
NEURAL NETS AND DEEP LEARNING
Convolutional Neural Networks (CNN)
DCNN
Perceptrons (MLP)
Recurrent Neural Networks (RNN)
LSM
LSTM
GRU
Autoencoders
seq2seq
Generative Adversarial Networks (GAN)

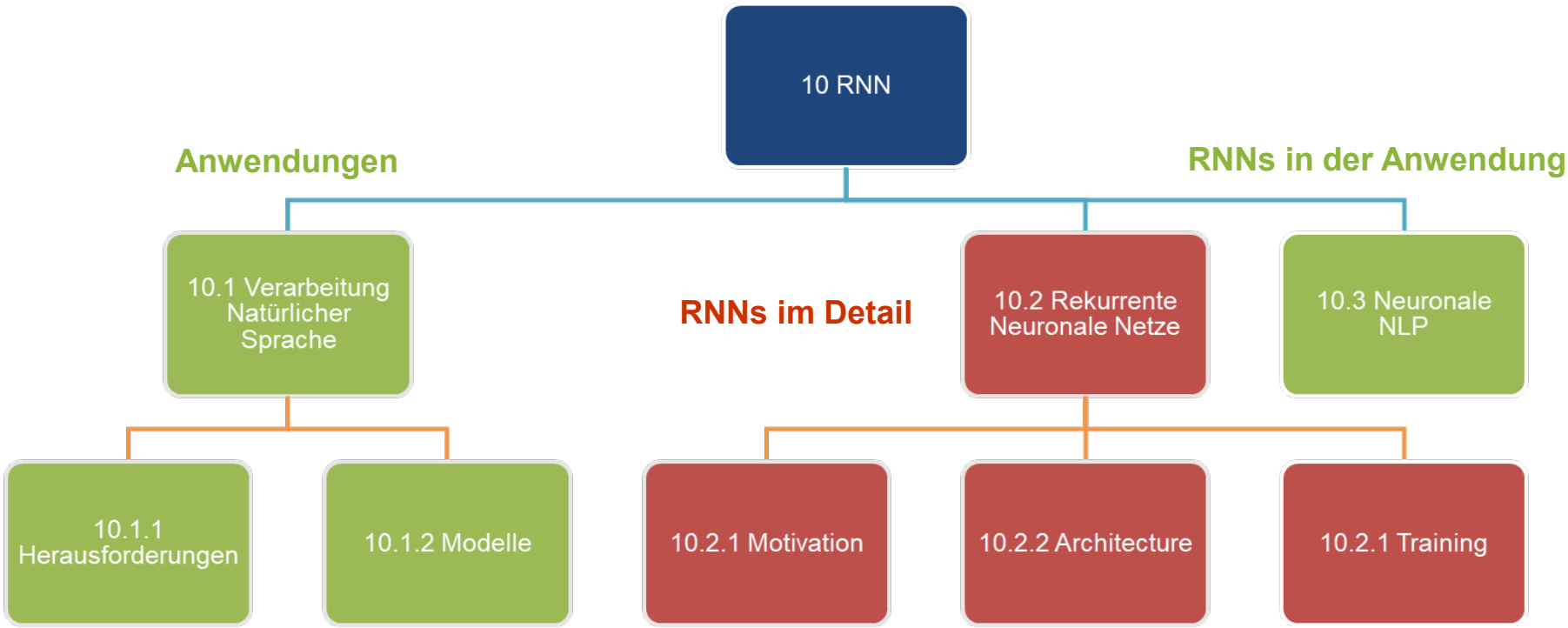
Die heutigen Vorlesungsfolien sind zu einem großen Teil basierend auf der Gastvorlesung von

Professor Jan Niehues

Lehrstuhl KI für Sprachtechnologien

Institut für Anthropomatik

Überblick


10 RNN
Anwendungen
RNNs in der Anwendung
10.1 Verarbeitung Natürlicher Sprache
RNNs im Detail
10.2 Rekurrente Neuronale Netze
10.3 Neuronale NLP
10.1.1 Herausforderungen
10.1.2 Modelle
10.2.1 Motivation
10.2.2 Architecture
10.2.1 Training

Lernziele der heutigen Vorlesung

Nach dieser Vorlesung werden Sie in der Lage sein:

- Anwendungsbeispiele von **KI zur Text-/Sprachverarbeitung und Generierung** zu beschreiben und zu kontrastieren.
- Wichtige **Herausforderungen** in der Verarbeitung natürlicher Sprache zu benennen und aktuelle Lösungsstrategien zu erklären.
- **Recurrent Neural Networks (RNNs)** als eine bedeutende KI-Methode zu formulieren, die Architektur zu begründen, und deren Stärken und Schwächen zu bewerten.
- Das **Trainieren von RNNs mittels Backpropagation in der Zeit** zu erklären, sowie Limitierungen wie das **Vanishing/Exploding Gradients Problem** zu beschreiben.

Verarbeitung natürlicher Sprache

Was versteht man unter der „Verarbeitung natürlicher Sprache“?

ChatGPT 5.2:

Was ist die "Verarbeitung natürlicher Sprache"?

Verarbeitung natürlicher Sprache (auf Englisch *Natural Language Processing*, kurz **NLP**) ist ein Teilgebiet der **Künstlichen Intelligenz** und der **Informatik**, das sich damit beschäftigt, wie Computer **menschliche Sprache verstehen, verarbeiten und erzeugen** können.

Einfach erklärt

NLP sorgt dafür, dass Maschinen mit Sprache arbeiten können – so wie Menschen es tun: lesen, schreiben, zuhören, sprechen und Bedeutung erfassen.

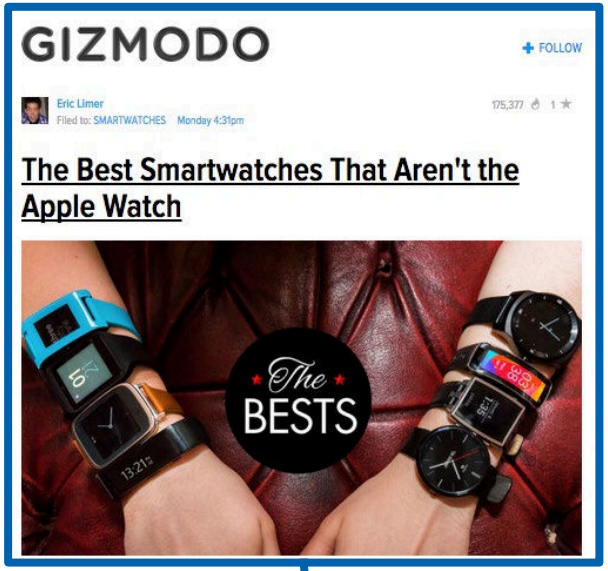
Typische Aufgaben der NLP

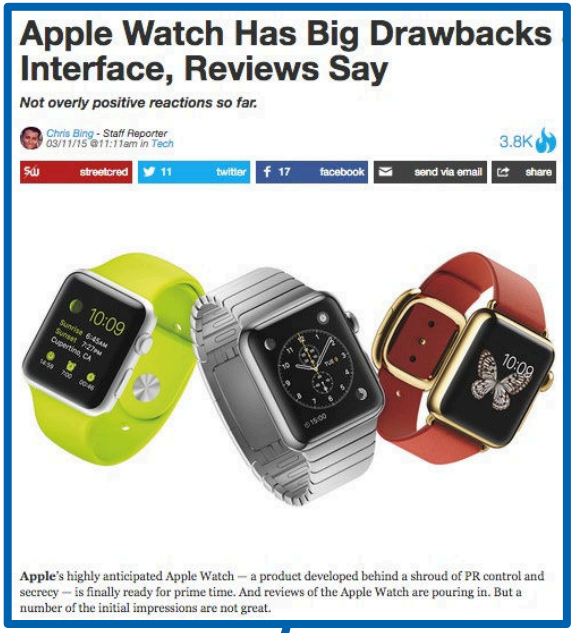
- **Text verstehen**: Worum geht es in einem Text? (z. B. Themen, Schlüsselbegriffe)
- **Spracherkennung**: Gesprochene Sprache in Text umwandeln
- **Maschinelle Übersetzung**: Texte von einer Sprache in eine andere übersetzen
- **Sentimentanalyse**: Erkennen, ob ein Text positiv, negativ oder neutral ist
- **Textgenerierung**: Automatisches Schreiben von Antworten, Zusammenfassungen oder Artikeln
- **Chatbots & Sprachassistenten**: Natürliche Dialoge mit Menschen führen

Wo begegnet dir NLP im Alltag?

- Sprachassistenten wie Siri oder Alexa
- Automatische Übersetzungen (z. B. Online-Übersetzer)
- Autokorrektur und Textvorschläge auf dem Smartphone
- Spam-Filter für E-Mails
- Suchmaschinen

Automatische Zusammenfassung





The Apple Watch has drawbacks. There are other smartwatches that offer more capabilities.

Maschinelle Übersetzung

"Il est impossible aux journalistes de rentrer dans les régions tibétaines"

Bruno Philip, correspondant du "Monde" en Chine, estime que les journalistes de l'AFP qui ont été expulsés de la province tibétaine du Qinghai "n'étaient pas dans l'illégalité".

Les faits Le dalaï-lama dénonce l'"enfer" imposé au Tibet depuis sa fuite, en 1959

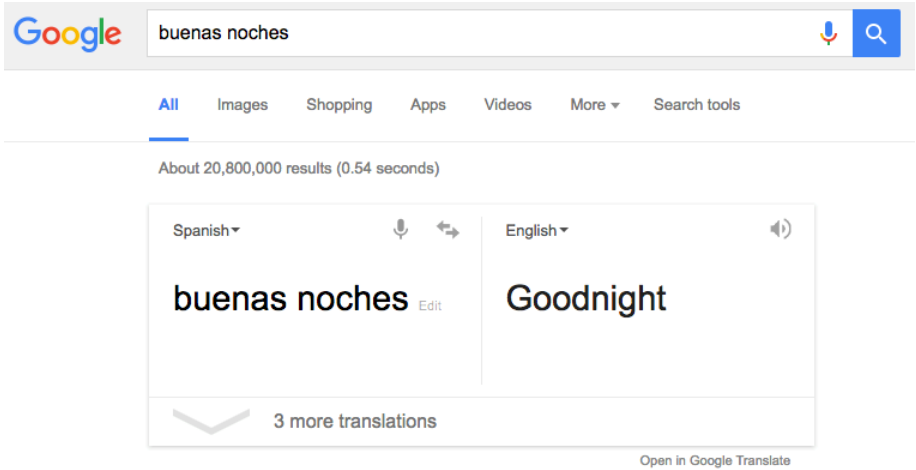
Vidéo Anniversaire de la rébellion

"It is impossible for journalists to enter Tibetan areas"

Philip Bruno, correspondent for "World" in China, said that journalists of the AFP who have been deported from the Tibetan province of Qinghai "were not illegal."

Facts The Dalai Lama denounces the "hell" imposed since he fled Tibet in 1959

Video Anniversary of the Tibetan rebellion: China on guard



Persönliche Assistenten

Siri

amazon alexa

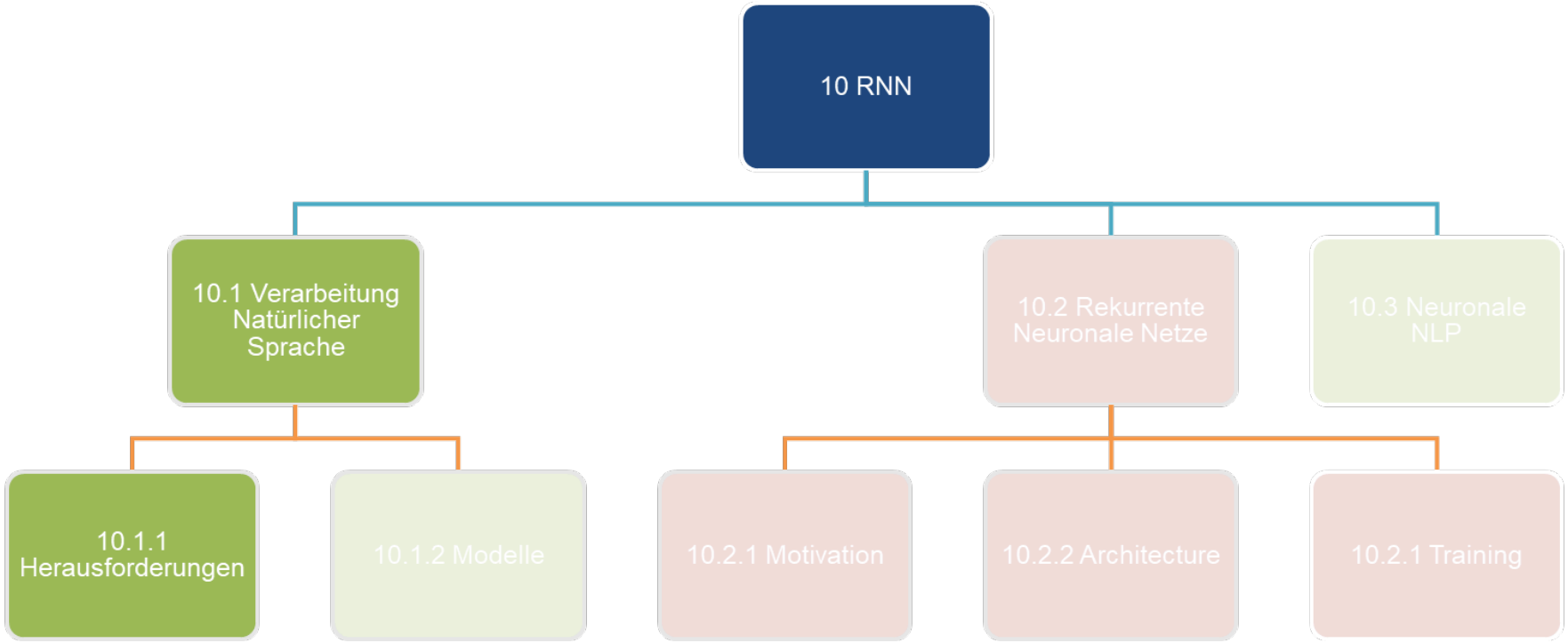
Question answering

NLP, formal

- Algorithmen zum **Verarbeiten, Analysieren und Verstehen** von Texten in natürlicher Sprache
    - **Struktur** verstehen
    - **Bedeutung** verstehen

- Schnittmenge zu vielen Bereichen
    - Informatik
        - Künstliche Intelligenz
    - Linguistik

Überblick


10 RNN
10.1 Verarbeitung Natürlicher Sprache
10.2 Rekurrente Neuronale Netze
10.3 Neuronale NLP
10.1.1 Herausforderungen
10.1.2 Modelle
10.2.1 Motivation
10.2.2 Architecture
10.2.1 Training

Verarbeitung natürlicher Sprache

ChatGPT 5.2

Was sind Herausforderungen bei der Verarbeitung natürlicher Sprache?

Bei der **Verarbeitung natürlicher Sprache (NLP)** gibt es mehrere zentrale Herausforderungen:

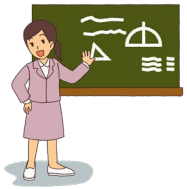
- **Mehrdeutigkeit**: Wörter und Sätze können je nach Kontext unterschiedliche Bedeutungen haben („Bank“ als Geldinstitut oder Sitzgelegenheit).
- **Kontextverständnis**: Die Bedeutung eines Satzes hängt oft vom vorherigen Text, vom Weltwissen oder von impliziten Annahmen ab.
- **Grammatik und Sprachvielfalt**: Sprachen haben komplexe Grammatikregeln, Dialekte, Umgangssprache und regionale Varianten.
- **Ironie und Sarkasmus**: Diese sind für Menschen oft leicht, für Computer aber sehr schwer zu erkennen.
- **Fehlerhafte oder unvollständige Sprache**: Tippfehler, Abkürzungen oder gesprochene Sprache mit Pausen und Füllwörtern erschweren die Verarbeitung.
- **Kulturelle und emotionale Nuancen**: Gefühle, Höflichkeit oder kulturelle Anspielungen sind schwer eindeutig zu interpretieren.
- **Datenabhängigkeit**: Gute NLP-Modelle benötigen große, qualitativ hochwertige und oft sprachspezifische Datensätze.

Diese Herausforderungen machen NLP zu einem komplexen, aber sehr aktiven Forschungsgebiet.

Herausforderungen

Herausforderungen

- Mehrdeutigkeiten

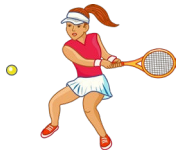


Vorzeichen

sign

omen

I am happy to introduce one of the best tennis players in the world.



Mehrdeutigkeit

Find as many meanings as you can of this sentence:

```
I made her duck
```

- I cooked waterfowl for her benefit (to eat)
- I cooked waterfowl belonging to her
- I created the (plaster?) waterfowl she owns
- I caused her to quickly lower her head or body
- I waved my magic wand and turned her into undifferentiated waterfowl

Mehrdeutigkeiten: Phonetik

Aye mate, her duck

I mate or duck

I'm eight or duck

Eye maid; her duck

I maid her duck

I'm aid her duck

I mate her duck

I'm ate her duck

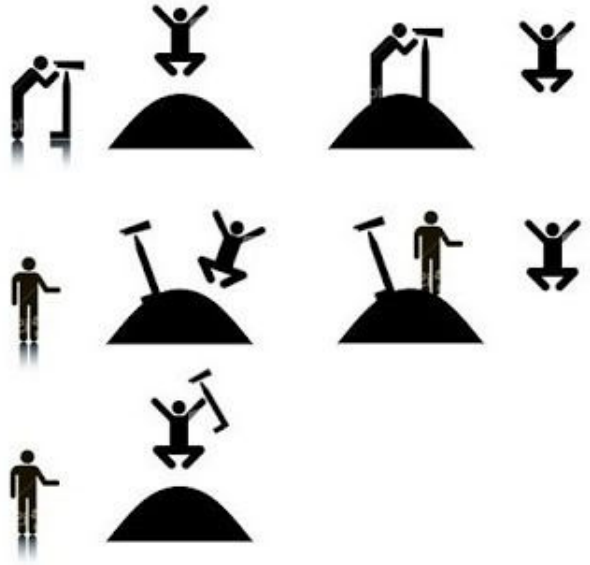
I'm ate or duck

I mate or duck

Mehrdeutigkeit - Struktur

- “I saw the man on the hill with a telescope.”

- I saw the man. The man was on the hill. I was using a telescope.
- I saw the man. I was on the hill. I was using a telescope.
- I saw the man. The man was on the hill. The hill had a telescope.
- I saw the man. I was on the hill. The hill had a telescope.
- I saw the man. The man was on the hill. I saw him using a telescope.

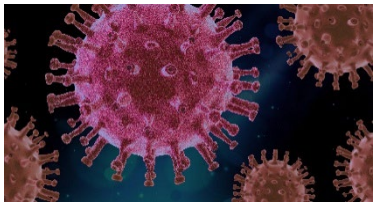


Co-reference

„If the **baby** does not like the **milk**, boil **it**“

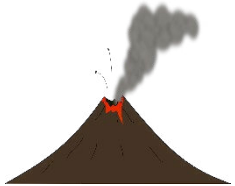
Herausforderungen

- Mehrdeutigkeiten
- Neue Wörter



COVID

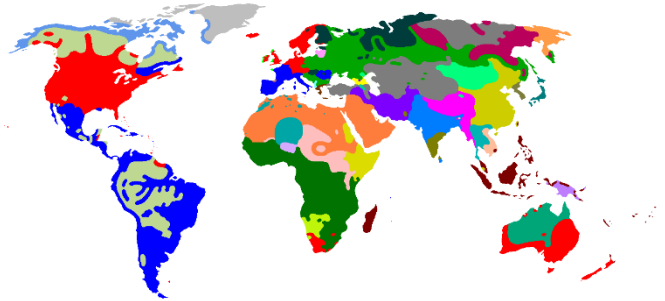
Brexit



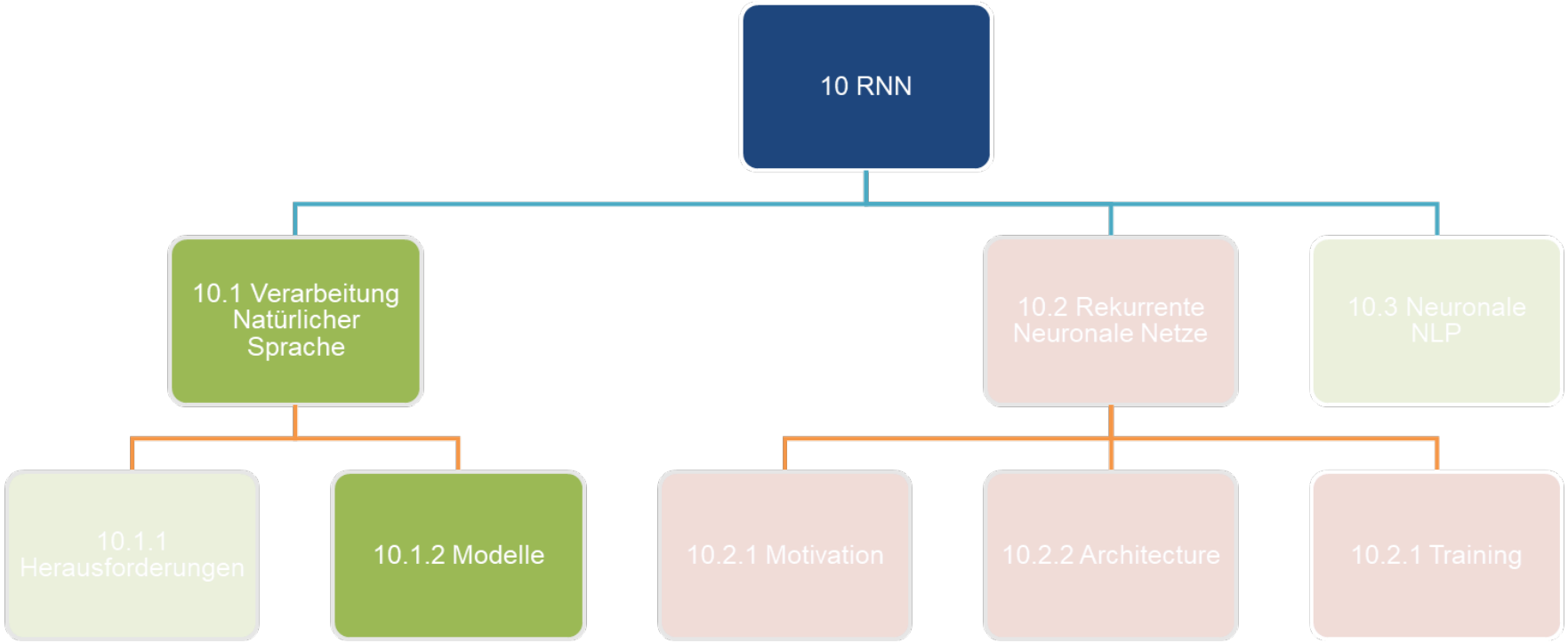
Eyjafjallajökull

Herausforderungen

- Mehrdeutigkeiten
- Neue Wörter
- Diversität
  - 6000-7000 Sprachen
  - Modalitäten



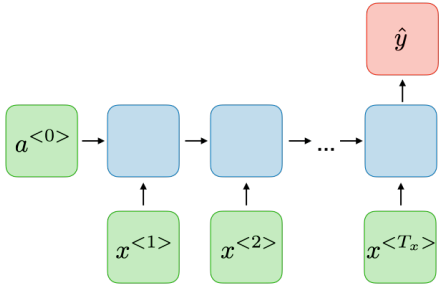
Überblick


10 RNN
10.1 Verarbeitung Natürlicher Sprache
10.2 Rekurrente Neuronale Netze
10.3 Neuronale NLP
10.1.1 Herausforderungen
10.1.2 Modelle
10.2.1 Motivation
10.2.2 Architecture
10.2.1 Training

Modelltypen

Sequenzklassifikation:

- Eingabe: Sequenz mit variabler Länge
- Ausgabe: Klasse



Beispiel: Sentimentanalyse

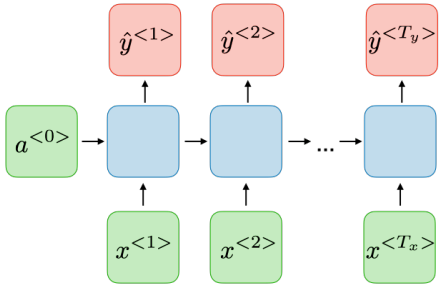
„Ich bin mit dem Produkt äußerst zufrieden. Die Qualität ist hervorragend und die Lieferung erfolgte schneller als erwartet. Auch der Kundenservice war freundlich und sehr hilfsbereit.“

„Man könnte meinen, die Qualität sei hochwertig, zumindest wenn man sich an den Beschreibungen orientiert. In der Praxis wirkt das Produkt jedoch alles andere als überzeugend, und die versprochene Zuverlässigkeit habe ich persönlich nicht wirklich erlebt. Wer einen reibungslosen Ablauf und hilfreichen Support erwartet, wird vermutlich ähnliche Erfahrungen machen.“

Modelltypen

Sequenzannotation:

- Eingang: Sequenz mit variabler Länge
- Ausgabe: Labelsequenz mit gleicher Länge



Beispiel: Erkennung von Entitäten (Named Entity Recognition) in

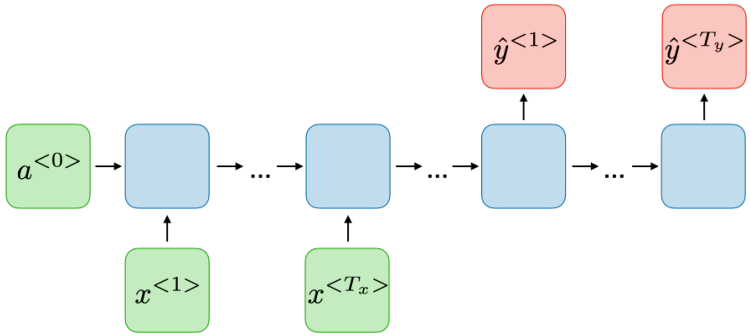
„Ich fliege mit American Airlines nach New York“

ich	fliege	mit	American	Airlines	nach	New	York
✗	✗	✗			✗		

Modelltypen

Sequenz-zu-Sequenz-Modell:

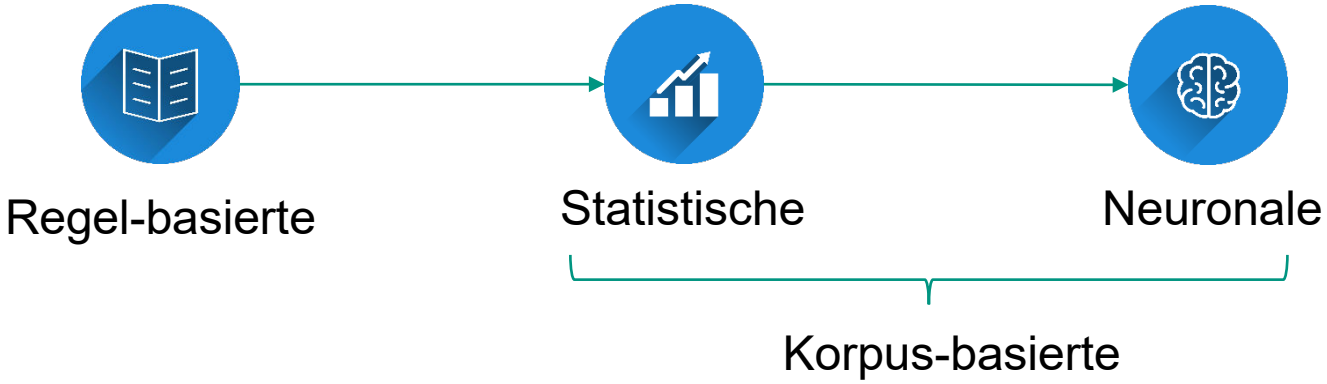
- Eingabe: Sequenz mit variabler Länge
- Ausgabe: Sequenz mit variabler Länge



Beispiel: Automatische Textzusammenfassung

Good quality summary output
S: a man charged with the murder last year of a british backpacker confessed to the slaying on the night he was charged with her killing , according to police evidence presented at a court hearing tuesday . ian douglas previte , ## , is charged with murdering caroline stuttle , ## , of yorkshire , england **T**: man charged with british backpacker 's death confessed to crime police officer claims **O**: man charged with murdering british backpacker confessed to murder

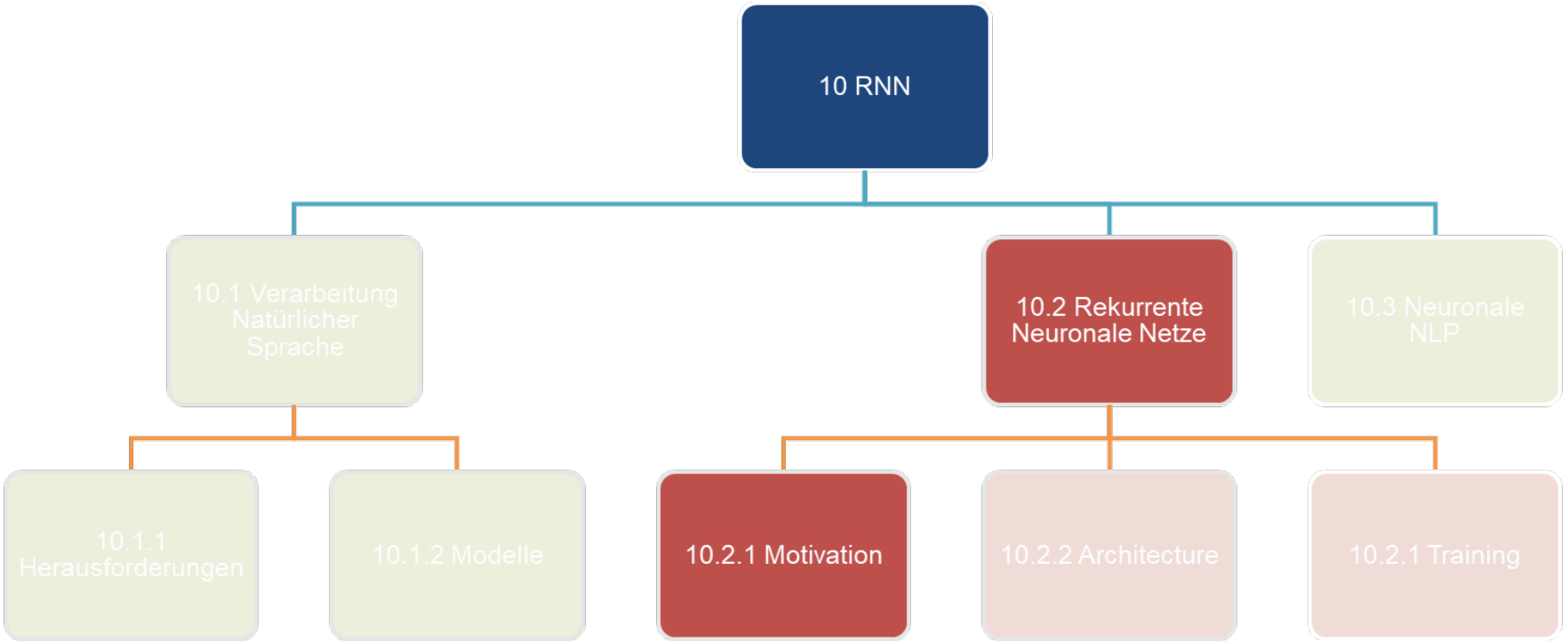
Ansätze



Experten erstellen Regeln

Lernen aus Daten

Überblick


10 RNN
10.1 Verarbeitung Natürlicher Sprache
10.2 Rekurrente Neuronale Netze
10.3 Neuronale NLP
10.1.1 Herausforderungen
10.1.2 Modelle
10.2.1 Motivation
10.2.2 Architecture
10.2.1 Training

Kurze Pause: Lehrevaluation (Vorlesung – Nowack)

Vielen herzlichen Dank für Ihre Zeit und Ihr konstruktives Feedback!

Sachlich, spezifisch, begründet, lösungsorientiert:

- Was war gut und warum?
- Was könnte verbessert werden und wie?
- Ihre Kommentare und Vorschläge sind mir sehr willkommen!

https://onlineumfrage.kit.edu/evasys/online.php?p=W28CQ

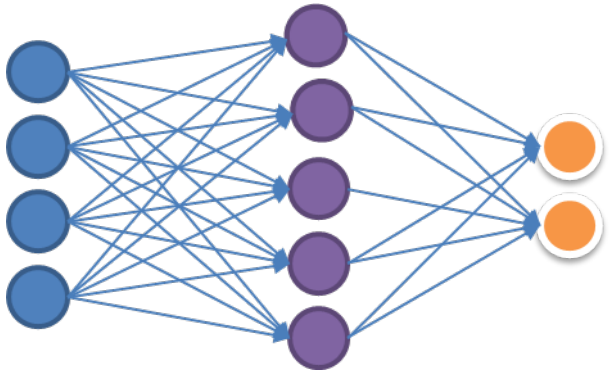
Frühere Vorlesungen

Mehrschichtiges Perzeptron (MLP)

- Annäherung von Funktionen: Zuordnen von Eingaben $\mathbf{X}$ zu Ausgaben $\mathbf{Y}$ mit Parametern $\boldsymbol{\theta}$

$$\mathbf{Y} = f(\mathbf{X}; \boldsymbol{\theta})$$

- Feste Anzahl von Eingabedimensionen → ein Problem bei der Verarbeitung von Sequenzen unterschiedlicher Länge



Recurrent Neural Networks (RNNs)

Goodfellow et al. Kapitel 10

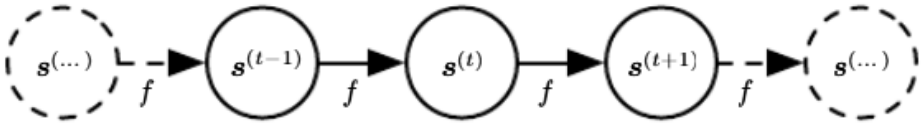
Eine Familie von neuronalen Netzwerken die spezialisiert darauf sind Sequenzdaten $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\tau$ von oftmals variabler Länge zu prozessieren.

Dafür wird wieder die Idee von „Parameter sharing“ zentral sein.

Notation: $\boldsymbol{x}_t$ beschreibt Vektoren, die die Sequenz bilden, mit dem Zeitschrittindex t. Dieser kann, muss sich aber nicht, auf eine wahre Zeitachse beziehen: „Die Position in der Sequenz.“

Zentral: Die Idee einer rekursiven/rekurrenten Berechnung eines dynamischen Systems mit Zustandsvektor $\boldsymbol{s}_t$

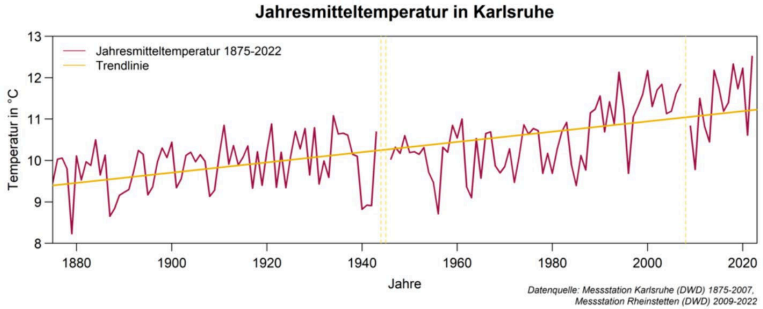
$$\boldsymbol{s}_t = f(\boldsymbol{s}_{t-1}; \boldsymbol{\theta})$$



Sequenzen

Viele verschiedene Arten von Sequenzdaten

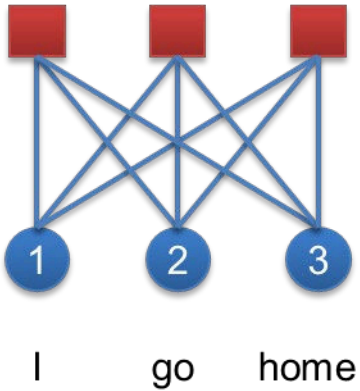
- Text
- Audio
- Videos
- Sensordaten
- Wetterdaten
- Finanzdaten (Aktienkurse)
- …

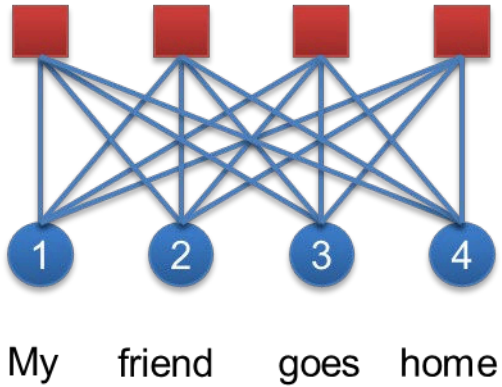


Stadt Karlsruhe

Sequenzen

Herausforderung: Eingaben mit variabler Länge





MLP: lerne alle Verbindungen über alle möglichen Positionen im Satz

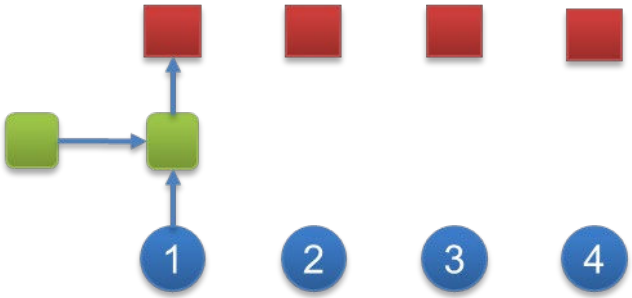
Herausforderung: Information unabhängig von der Position im Satz extrahieren:

„I went to Nepal in 2009.“

„In 2009, I went to Nepal.“

Neuronale Netze für Sequenzen

Idee: Speichern des Verlaufs in einem Zustand (Vektor)



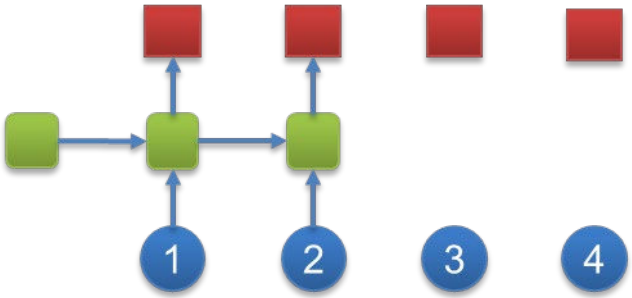
Ausgaben/Output $\boldsymbol{o}_t$

Versteckter/Hidden State $\boldsymbol{h}_t$

Eingaben/Input $\boldsymbol{x}_t$

Neuronale Netze für Sequenzen

Idee: Speichern des Verlaufs in einem Zustand (Vektor) – Reihenfolge wichtig!



Ausgaben/Output $\boldsymbol{o}_t$

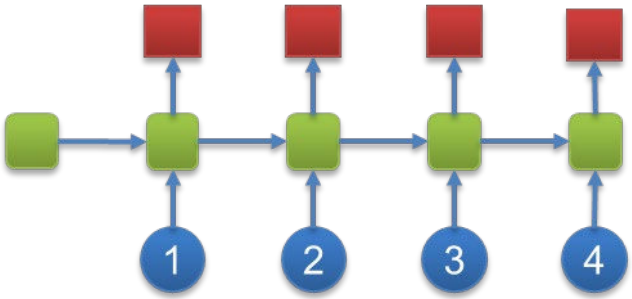
Versteckter/Hidden State $\boldsymbol{h}_t$

Eingaben/Input $\boldsymbol{x}_t$

“Mary kills John” ist anders als “John kills Mary”

Neuronale Netze für Sequenzen

Idee: Speichern des Verlaufs in einem Zustand (Vektor) $\boldsymbol{h}_t$

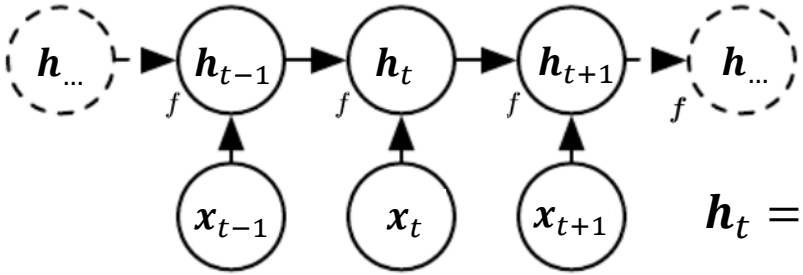


Ausgaben/Output $\boldsymbol{o}_t$

Versteckter/Hidden State $\boldsymbol{h}_t$

Eingaben/Input $\boldsymbol{x}_t$

$$\boldsymbol{s}_t = f(\boldsymbol{s}_{t-1}; \boldsymbol{\theta})$$



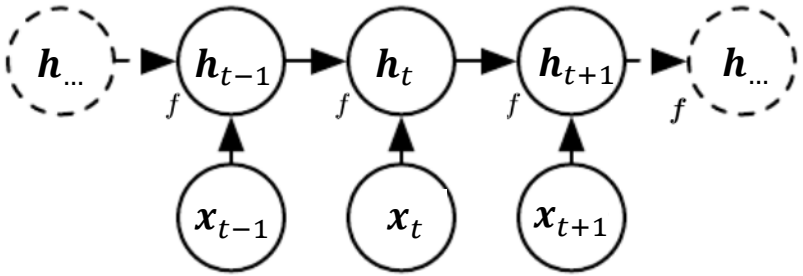
$$\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t; \boldsymbol{\theta})$$

Überblick


10 RNN
10.1 Verarbeitung Natürlicher Sprache
10.2 Rekurrente Neuronale Netze
10.3 Neuronale NLP
10.1.1 Herausforderungen
10.1.2 Modelle
10.2.1 Motivation
10.2.2 Architecture
10.2.1 Training

Was ist die Architektur solcher RNNs?

- Effektiv könnte jede neuronale Funktion f die einer rekurrenten Beschreibung wie z.B. $\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t; \boldsymbol{\theta})$ folgt als RNN betrachtet werden…
- Daher viele Optionen!
- Es gibt aber z.B. Abweichungen darin wie der hidden state basierend auf den Eingaben und Ausgaben definiert wird.



Gemeinsame Vorteile:

- Unabhängig von Sequenzlänge
- Können die gleiche Übergangsfunktion f für jeden Zeitschritt nutzen

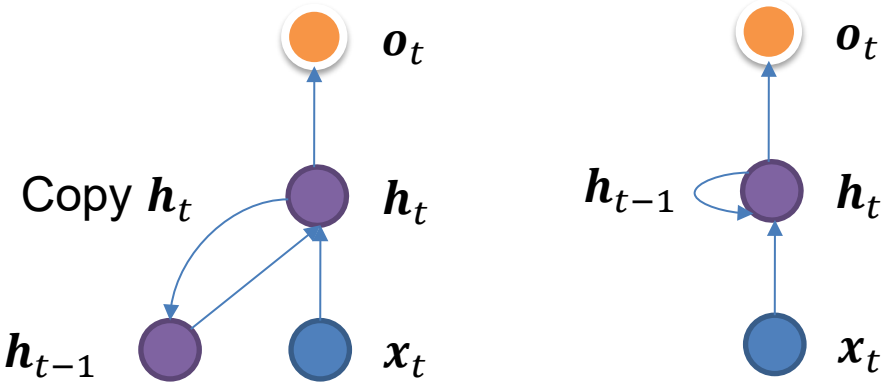
Elman-Netzwerke

Architektur:

- Vorheriger hidden state wird zur Eingabe für den nächsten Zeitschritt

Implementierung:

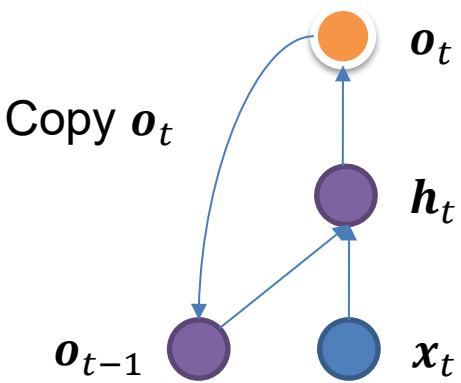
- Kopiere den hidden state des vorherigen Zeitschritts

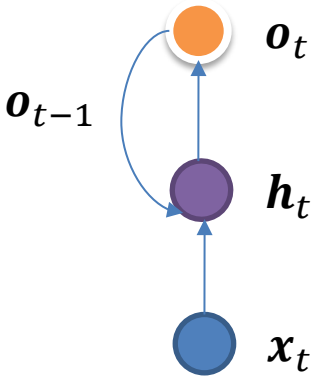


Jordan-Netzwerke

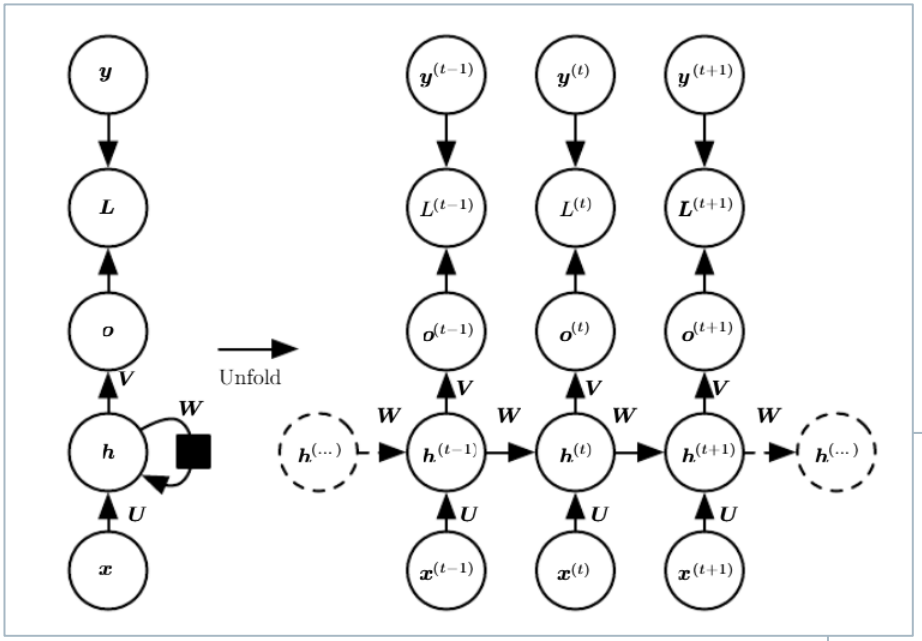
Gleicher Kopiermechanismus

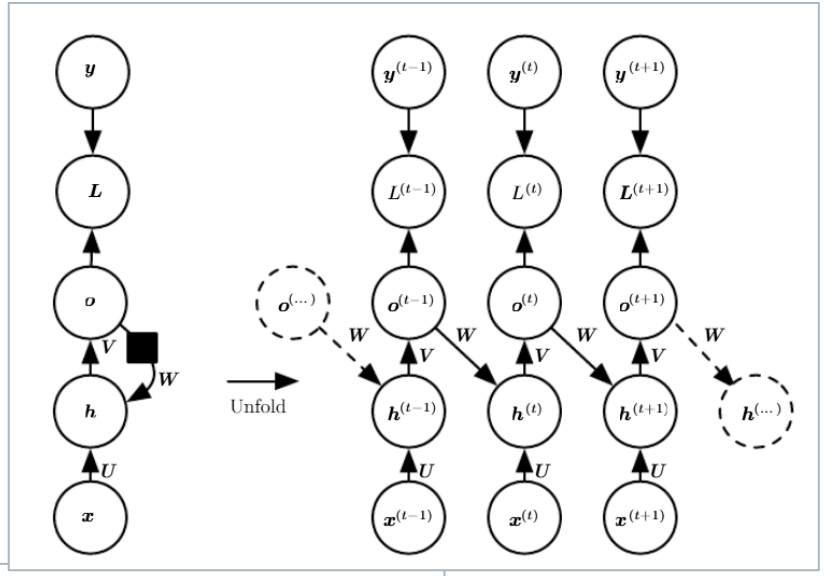
- Aber für die Ausgabeschicht

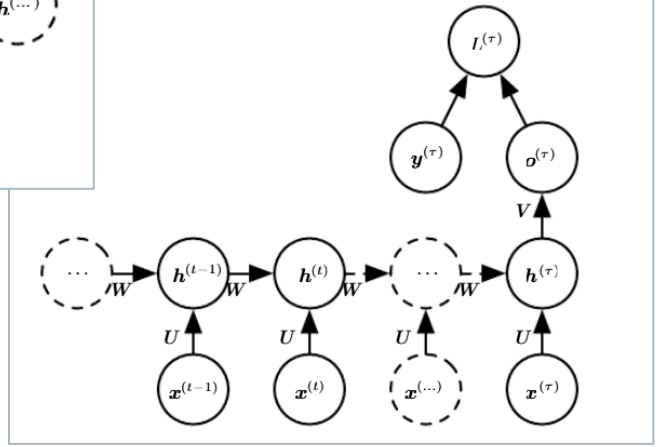




RNN-Architekturbeispiele







Was sind L, y?

Elman? Jordan?

Wie werden RNNs in der Praxis implementiert?

Berechnung der versteckten Schicht: $\boldsymbol{h}_t = f(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}, \boldsymbol{\theta})$

Viele verschiedene Implementierungen/Arten von RNNs, z.B.:

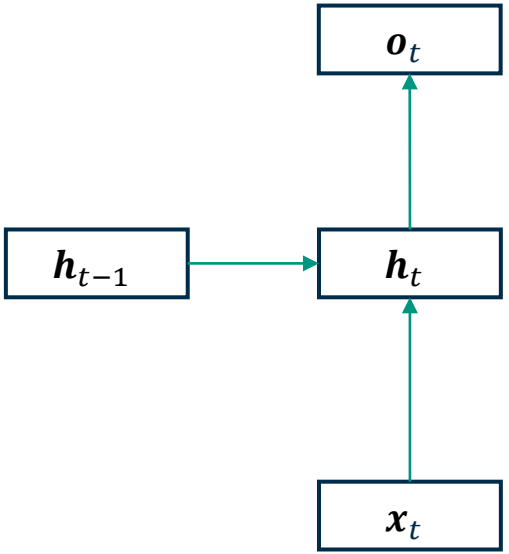
- Einfache rekurrente neuronale Netzwerke
  Rumelhart et al. 1986
- Long Short-Term Memory (LSTM)
  Hochreiter und Schmidhuber 1997; adressiert Vanishing Gradients Problem
- Gated Recurrent Unit (GRU)
  Cho et al. 2014; recheneffizientere Alternative

Einfache rekurrente neuronale Netze

$$\boldsymbol{h}_t = f_{\text{act}}(\mathbf{W}^h \boldsymbol{h}_{t-1} + \mathbf{W}^x \boldsymbol{x}_t + \boldsymbol{b})$$

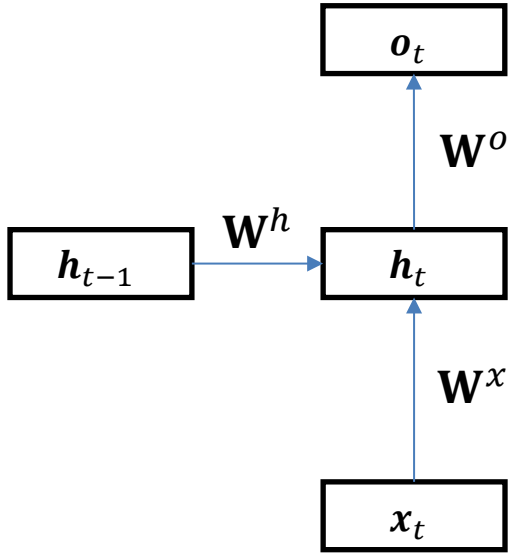
f_{act}: eine Aktivierungsfunktion (Sigmoid, Tanh, ReLU)

Innerhalb der Klammer: eine lineare Kombination zwischen dem Speicher $\boldsymbol{h}_{t-1}$ und der Eingabe $\boldsymbol{x}_t$

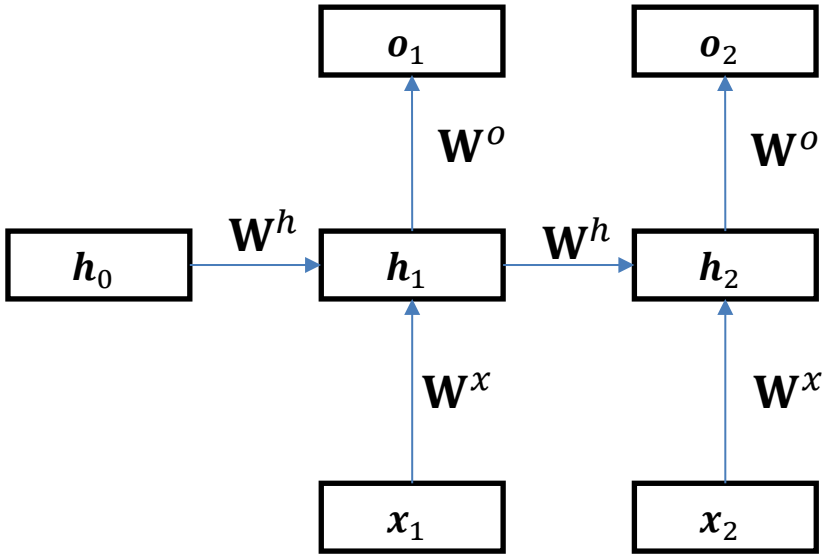


Ausrollen/Roll-out

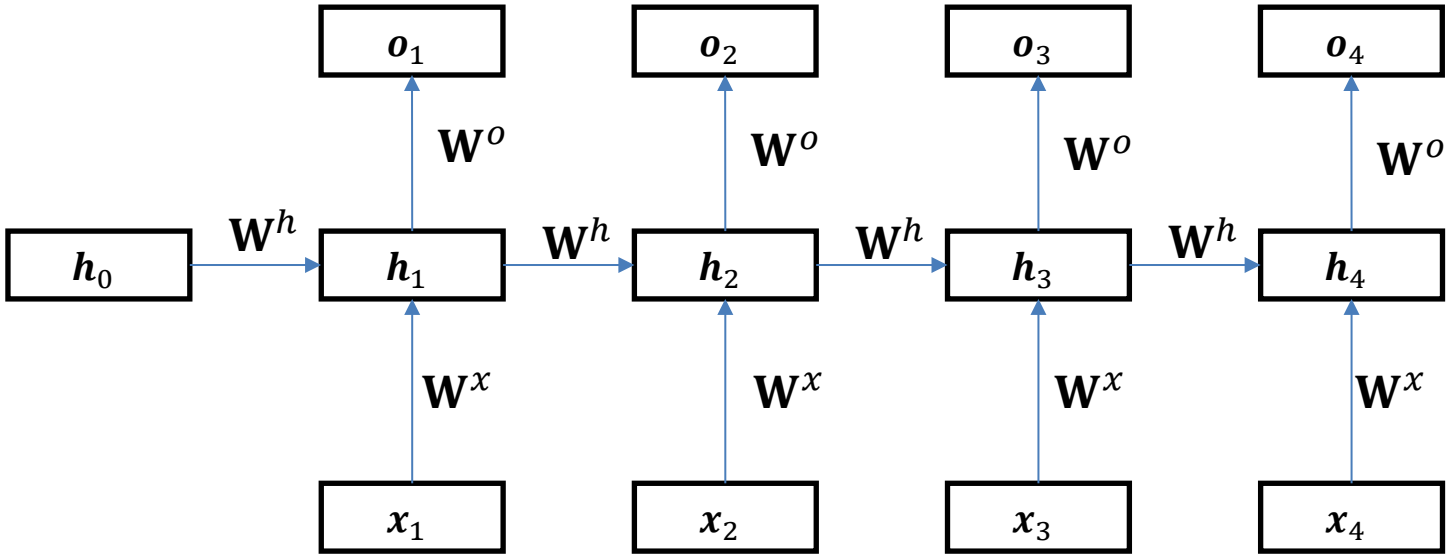
- Ausrollen des Netzwerks über die Zeit
  → geteilte Gewichte!
- D.h. die gleichen drei Gewichtsmatrizen $\mathbf{W}^h, \mathbf{W}^x$ und (für die Ausgabe) $\mathbf{W}^o$ werden zu jedem Zeitschritt angewendet
- Damit haben wir ein Netzwerk, welches unabhängig von der Sequenzlänge eingesetzt werden kann
- Einfaches RNN: Feedforward-Netzwerk für jede neue Eingabe $\boldsymbol{x}_t$, zudem basierend auf dem vorherigen hidden state $\boldsymbol{h}_{t-1}$.



Ausrollen/Roll-out



Ausrollen/Roll-out

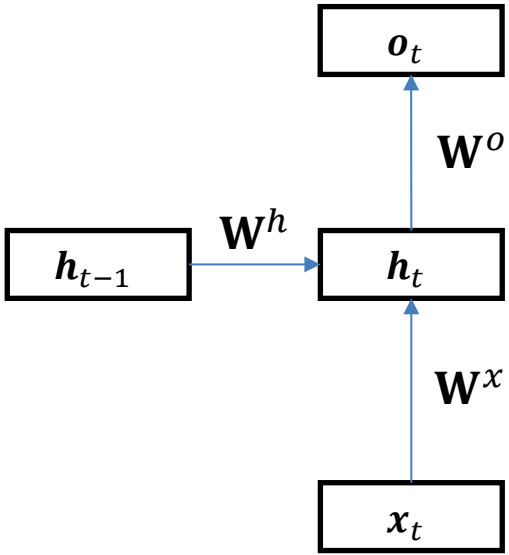


Beispiel

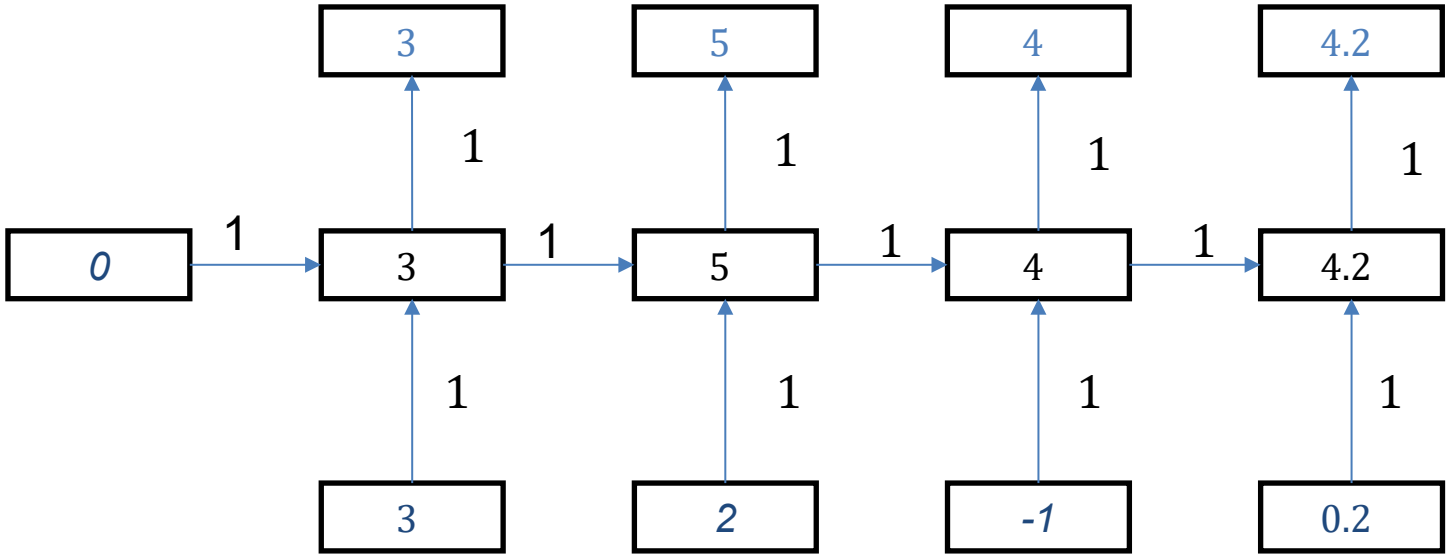
$$\boldsymbol{o}_t = \mathbf{W}^o \boldsymbol{h}_t$$

$$\boldsymbol{h}_t = \mathbf{W}^h \boldsymbol{h}_{t-1} + \mathbf{W}^x \boldsymbol{x}_t$$

Frage: Wann wird das Output einfach zur Summe der Eingabesequenz?

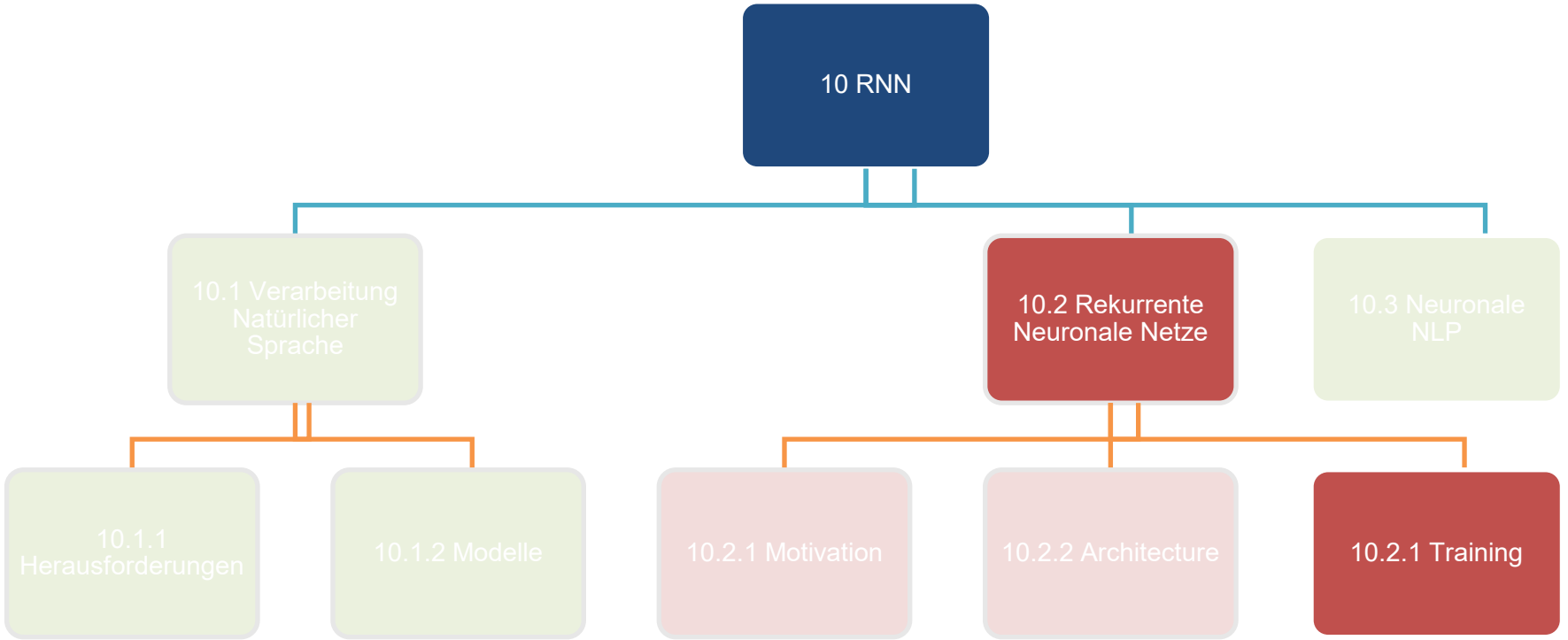


Beispiel



Fragen?

Überblick


10 RNN
10.1 Verarbeitung Natürlicher Sprache
10.2 Rekurrente Neuronale Netze
10.3 Neuronale NLP
10.1.1 Herausforderungen
10.1.2 Modelle
10.2.1 Motivation
10.2.2 Architecture
10.2.1 Training

Training? → Backpropagation durch die Zeit

Werbos (1990) Backpropagation through time: what it does and how to do it

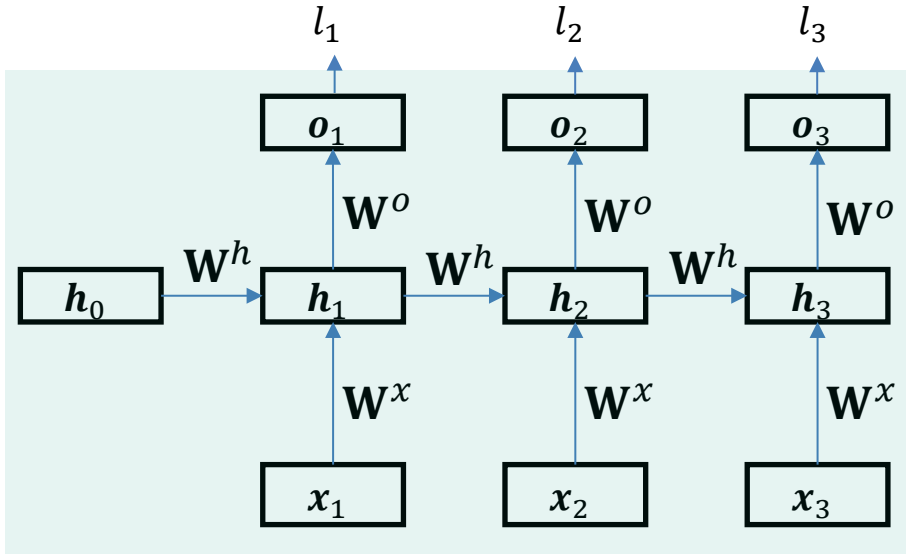
- Backpropagation through time (BPTT)
  - Für neuronale Netzwerke in denen der versteckte Zustand $\boldsymbol{h}_t$ rekurrent propagiert wird.
  - Gegeben: geordnete (→ Kausalität) Sequenz von Input-Output Paaren $(\boldsymbol{x}_t, \boldsymbol{y}_t)$ sowie Anfangszustand (oft Nullvektor)
- Anfang: Forward-Pass um Aktivierungen zu berechnen.
- Dann: Backpropagation um die Gradienten der Loss-Funktion bezüglich allen Netzwerkparametern zu finden; über die verschiedenen Zeitschritte hinweg.
- Je nach RNN-Architektur: berechne Loss abhängig von Vorhersagen zu jedem Zeitschritt, oder nur am Ende, oder …
- Hier: wegen parameter sharing, werden die gleichen Gewichtsmatrizen $\boldsymbol{W}^x, \boldsymbol{W}^h, \boldsymbol{W}^o$ zu jedem Zeitschritt genutzt. Daher werden wir auch die Updates der Gewichte über alle Zeitschritte hinweg mitteln/summieren/gewichten.

Backpropagation durch die Zeit

$$\boldsymbol{h}_t = \mathrm{f}_{\mathrm{act}}(\mathbf{W}^x \boldsymbol{x}_t + \mathbf{W}^h \boldsymbol{h}_{t-1} + \boldsymbol{b}_h)$$

$$\boldsymbol{o}_t = f_{\mathrm{out}}(\mathbf{W}^o \mathbf{h}_t + \boldsymbol{b}_o)$$

$$L = \frac{1}{\tau} \sum_{t=1}^{\tau} l_t(\boldsymbol{o}_t, \boldsymbol{y}_t)$$



Für Gradient Descent

Gradienten bezüglich den drei Gewichtsmatrizen an jedem t mitteln:

$$\mathbf{W}^o = \mathbf{W}^o - \alpha \frac{\partial L}{\partial \mathbf{W}^o} \qquad \frac{\partial L}{\partial \mathbf{W}^o} = \frac{1}{\tau} \sum_{t=1}^{\tau} \frac{\partial l_t}{\partial \mathbf{W}^o}$$

$$\mathbf{W}^h = \mathbf{W}^h - \alpha \frac{\partial L}{\partial \mathbf{W}^h} \qquad \frac{\partial L}{\partial \mathbf{W}^h} = \frac{1}{\tau} \sum_{t=1}^{\tau} \frac{\partial l_t}{\partial \mathbf{W}^h}$$

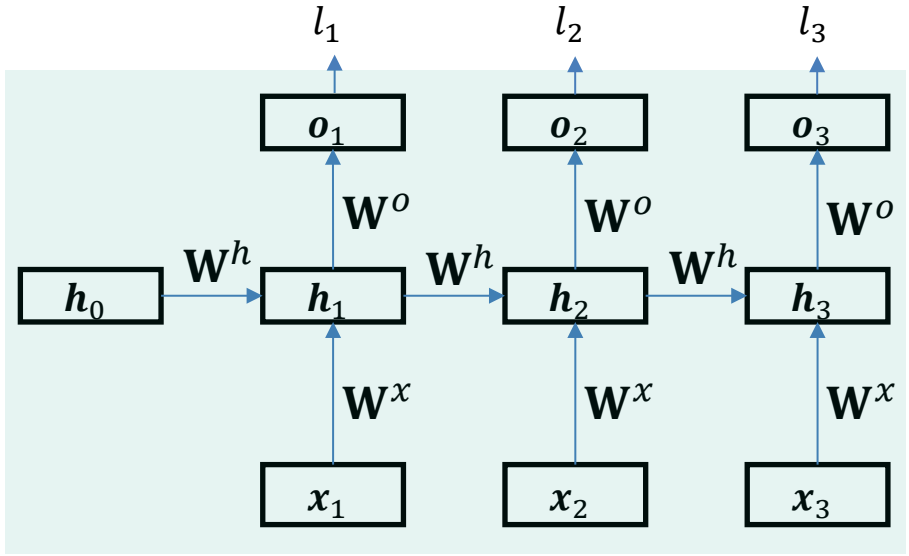
$$\mathbf{W}^x = \mathbf{W}^x - \alpha \frac{\partial L}{\partial \mathbf{W}^x} \qquad \frac{\partial L}{\partial \mathbf{W}^x} = \frac{1}{\tau} \sum_{t=1}^{\tau} \frac{\partial l_t}{\partial \mathbf{W}^x}$$

Backpropagation durch die Zeit

$$\boldsymbol{h}_t = \mathrm{f}_{\mathrm{act}}(\mathbf{W}^x \boldsymbol{x}_t + \mathbf{W}^h \boldsymbol{h}_{t-1} + \boldsymbol{b}_h)$$

$$\boldsymbol{o}_t = f_{\mathrm{out}}(\mathbf{W}^o \mathbf{h}_t + \boldsymbol{b}_o)$$

$$L = \frac{1}{\tau} \sum_{t=1}^{\tau} l_t(\boldsymbol{o}_t, \boldsymbol{y}_t)$$



Nutze den ausgerollten Graphen, um alle Abhängigkeiten zu sehen, die in der Berechnung der Gradienten berücksichtigt werden müssen.

Zum Beispiel für $\boldsymbol{h}_3$ sind die direkten Abhängigkeiten: $\boldsymbol{W}^x$, $\boldsymbol{W}^h$, $\boldsymbol{h}_2$, $\boldsymbol{x}_3$

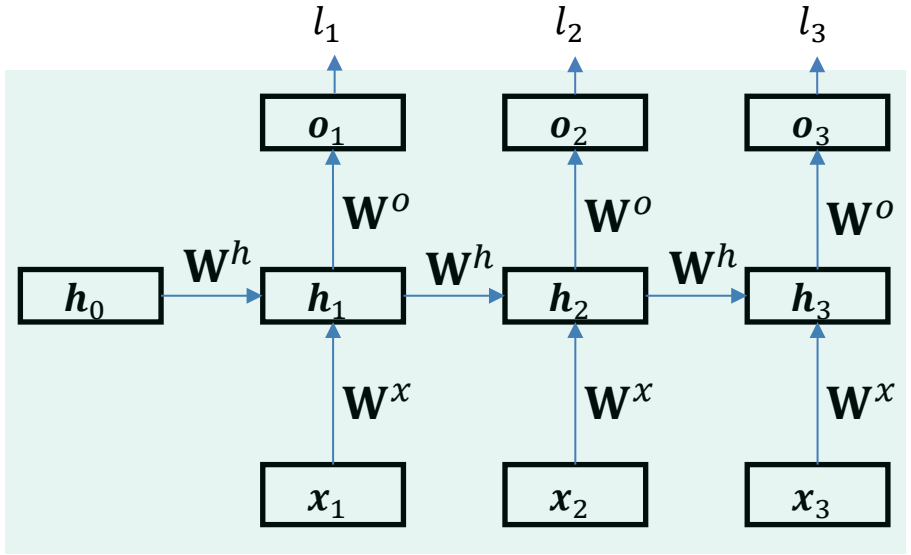
Ableitungen des Losses hinsichtlich der drei Gewichtsmatrizen: $\frac{\partial L}{\partial \mathbf{W}^x}, \frac{\partial L}{\partial \mathbf{W}^h}, \frac{\partial L}{\partial \mathbf{W}^o}$

Backpropagation durch die Zeit

$$\boldsymbol{h}_t = \mathbf{W}^x \boldsymbol{x}_t + \mathbf{W}^h \boldsymbol{h}_{t-1}$$

$$\boldsymbol{o}_t = \mathbf{W}^o \mathbf{h}_t$$

$$L = \frac{1}{\tau} \sum_{t=1}^{\tau} l_t(\boldsymbol{o}_t, \boldsymbol{y}_t)$$



Fehler hinsichtlich Vorhersage:

$\boldsymbol{g}_t = \frac{\partial l_t}{\partial \boldsymbol{o}_t}$ und somit $\frac{\partial L}{\partial \boldsymbol{o}_t} = \frac{1}{\tau} \boldsymbol{g}_t$

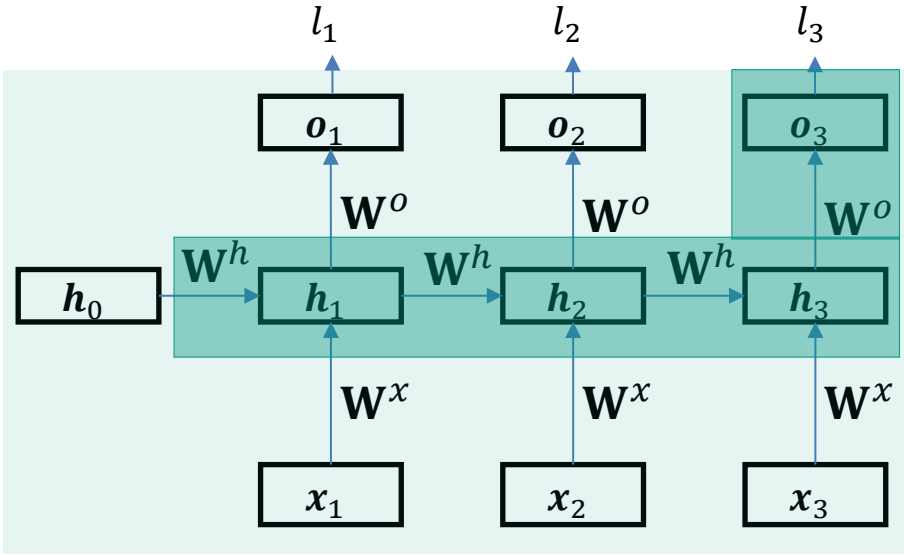
Annahmen:

- $\mathrm{f}_{\mathrm{act}}, \mathrm{f}_{\mathrm{out}}$ sind Einheitsfunktionen
- keine $\boldsymbol{b}$ Terme
- $\boldsymbol{h}_0$ wird als konstant, z.B. Nullvektor angenommen
- $\boldsymbol{o}_3$ ist das letzte Output

Hilfreiches Video von Herman Kamper

Backpropagation durch die Zeit

$$\boldsymbol{h}_t = \mathbf{W}^x \boldsymbol{x}_t + \mathbf{W}^h \boldsymbol{h}_{t-1}$$
$$\boldsymbol{o}_t = \mathbf{W}^o \mathbf{h}_t$$
$$L = \frac{1}{\tau} \sum_{t=1}^{\tau} l_t(\boldsymbol{o}_t, \boldsymbol{y}_t)$$



Gradienten hinsichtlich $\mathbf{W}^o$ und $\mathbf{W}^h$:

$$\frac{\partial l_3}{\partial \mathbf{W}^o} = \frac{\partial l_3}{\partial \boldsymbol{o}_3} \frac{\partial \boldsymbol{o}_3}{\partial \mathbf{W}^o} = \frac{\partial l_3}{\partial \boldsymbol{o}_3} \boldsymbol{h}_3^T$$

$\boldsymbol{h}_0$ Nullvektor

$$\frac{\partial l_3}{\partial \mathbf{W}^h} = \frac{\partial l_3}{\partial \boldsymbol{o}_3} \frac{\partial \boldsymbol{o}_3}{\partial \boldsymbol{h}_3} \frac{\partial \boldsymbol{h}_3}{\partial \mathbf{W}^h} + \frac{\partial l_3}{\partial \boldsymbol{o}_3} \frac{\partial \boldsymbol{o}_3}{\partial \boldsymbol{h}_3} \frac{\partial \boldsymbol{h}_3}{\partial \boldsymbol{h}_2} \frac{\partial \boldsymbol{h}_2}{\partial \mathbf{W}^h} + \frac{\partial l_3}{\partial \boldsymbol{o}_3} \frac{\partial \boldsymbol{o}_3}{\partial \boldsymbol{h}_3} \frac{\partial \boldsymbol{h}_3}{\partial \boldsymbol{h}_2} \frac{\partial \boldsymbol{h}_2}{\partial \boldsymbol{h}_1} \frac{\partial \boldsymbol{h}_1}{\partial \mathbf{W}^h} + \frac{\partial l_3}{\partial \boldsymbol{o}_3} \frac{\partial \boldsymbol{o}_3}{\partial \boldsymbol{h}_3} \frac{\partial \boldsymbol{h}_3}{\partial \boldsymbol{h}_2} \frac{\partial \boldsymbol{h}_2}{\partial \boldsymbol{h}_1} \frac{\partial \boldsymbol{h}_1}{\partial \boldsymbol{h}_0} \frac{\partial \boldsymbol{h}_0}{\partial \mathbf{W}^h}$$

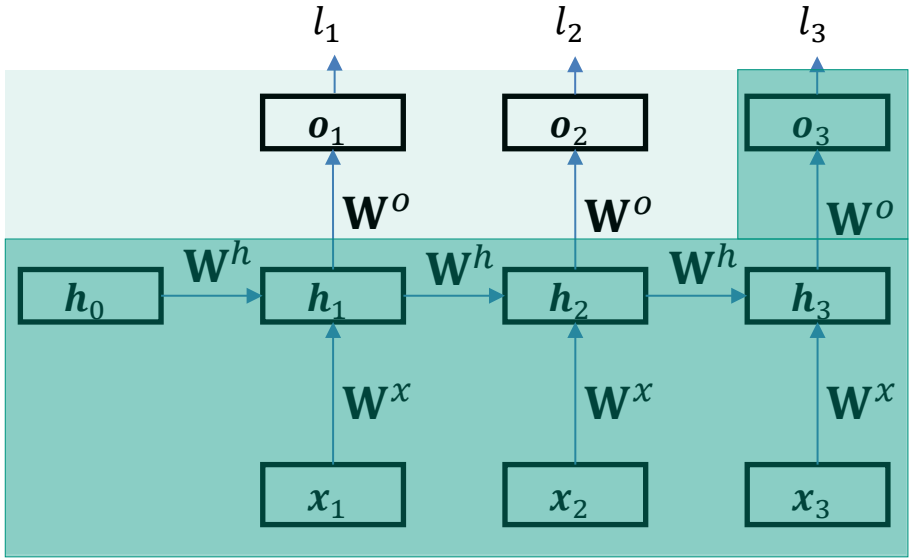
$\mathbf{W}^o$ $\boldsymbol{h}_2^T$

$$= \sum_{t=1}^{3} \frac{\partial l_3}{\partial \boldsymbol{o}_3} \frac{\partial \boldsymbol{o}_3}{\partial \boldsymbol{h}_3} \frac{\partial \boldsymbol{h}_3}{\partial \boldsymbol{h}_t} \frac{\partial \boldsymbol{h}_t}{\partial \mathbf{W}^h}$$

Nutze Kausalität in der Zeit um immer von fixen Quantitäten auszugehen!

Backpropagation durch die Zeit

$$\boldsymbol{h}_t = \mathbf{W}^x \boldsymbol{x}_t + \mathbf{W}^h \boldsymbol{h}_{t-1}$$
$$\boldsymbol{o}_t = \mathbf{W}^o \mathbf{h}_t$$
$$L = \frac{1}{\tau} \sum_{t=1}^{\tau} l_t(\boldsymbol{o}_t, \boldsymbol{y}_t)$$



Gradienten hinsichtlich $\mathbf{W}^x$ (läuft wieder über $\boldsymbol{h}_t$, daher sehr ähnlich)

$$\frac{\partial l_3}{\partial \mathbf{W}^x} = \frac{\partial l_3}{\partial \boldsymbol{o}_3}\frac{\partial \boldsymbol{o}_3}{\partial \boldsymbol{h}_3}\frac{\partial \boldsymbol{h}_3}{\partial \mathbf{W}^x} + \frac{\partial l_3}{\partial \boldsymbol{o}_3}\frac{\partial \boldsymbol{o}_3}{\partial \boldsymbol{h}_3}\frac{\partial \boldsymbol{h}_3}{\partial \boldsymbol{h}_2}\frac{\partial \boldsymbol{h}_2}{\partial \mathbf{W}^x} + \frac{\partial l_3}{\partial \boldsymbol{o}_3}\frac{\partial \boldsymbol{o}_3}{\partial \boldsymbol{h}_3}\frac{\partial \boldsymbol{h}_3}{\partial \boldsymbol{h}_2}\frac{\partial \boldsymbol{h}_2}{\partial \boldsymbol{h}_1}\frac{\partial \boldsymbol{h}_1}{\partial \mathbf{W}^x}$$

$$= \sum_{t=1}^{3} \frac{\partial l_3}{\partial \boldsymbol{o}_3}\frac{\partial \boldsymbol{o}_3}{\partial \boldsymbol{h}_3}\frac{\partial \boldsymbol{h}_3}{\partial \boldsymbol{h}_t}\frac{\partial \boldsymbol{h}_t}{\partial \mathbf{W}^x}$$

$\mathbf{W}^o$ $\boldsymbol{x}_3^T$

Analog für andere Outputs, bias-Terme, eventuell $\boldsymbol{h}_0$ Parameter…

Probleme...

- Vanishing gradient problem / Exploding gradient problem

  → durch wiederholte Multiplikation verschwinden/explodieren Gradienten

  Illustrative Annahme: $\boldsymbol{h}_t = \mathbf{W}^{\mathrm{T}} \boldsymbol{h}_{t-1}$

  Dann ist die Rekurrenz effektiv auf einfache Art und Weise beschrieben durch

  $$\boldsymbol{h}_t = (\mathbf{W}^t)^{\mathrm{T}} \boldsymbol{h}_0$$

  Falls $\mathbf{W}$ eine Eigenwertzerlegung $\mathbf{W} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$ zulässt, erhalten wir

  $$\boldsymbol{h}_t = \mathbf{V}^{\mathrm{T}} \boldsymbol{\Lambda}^t \mathbf{V} \boldsymbol{h}_0$$

  |Eigenwerte| < 1 „verschwinden“ und „explodieren“ andernfalls…

- Lösungsansätze: vor allem angepasste Architekturen, z.B. das LSTM

Fragen?

You cannot vote anymore

Gegeben sei ein rekurrentes neuronales Netz mit geteilten Gewichten W^h, einer Ausgabeloss l_τ am letzten Zeitschritt τ und Hidden States h_t. Welche der folgenden Gleichungen beschreibt korrekt den Gradienten des Losses bezüglich der rekurrenten Gewichte W^h?

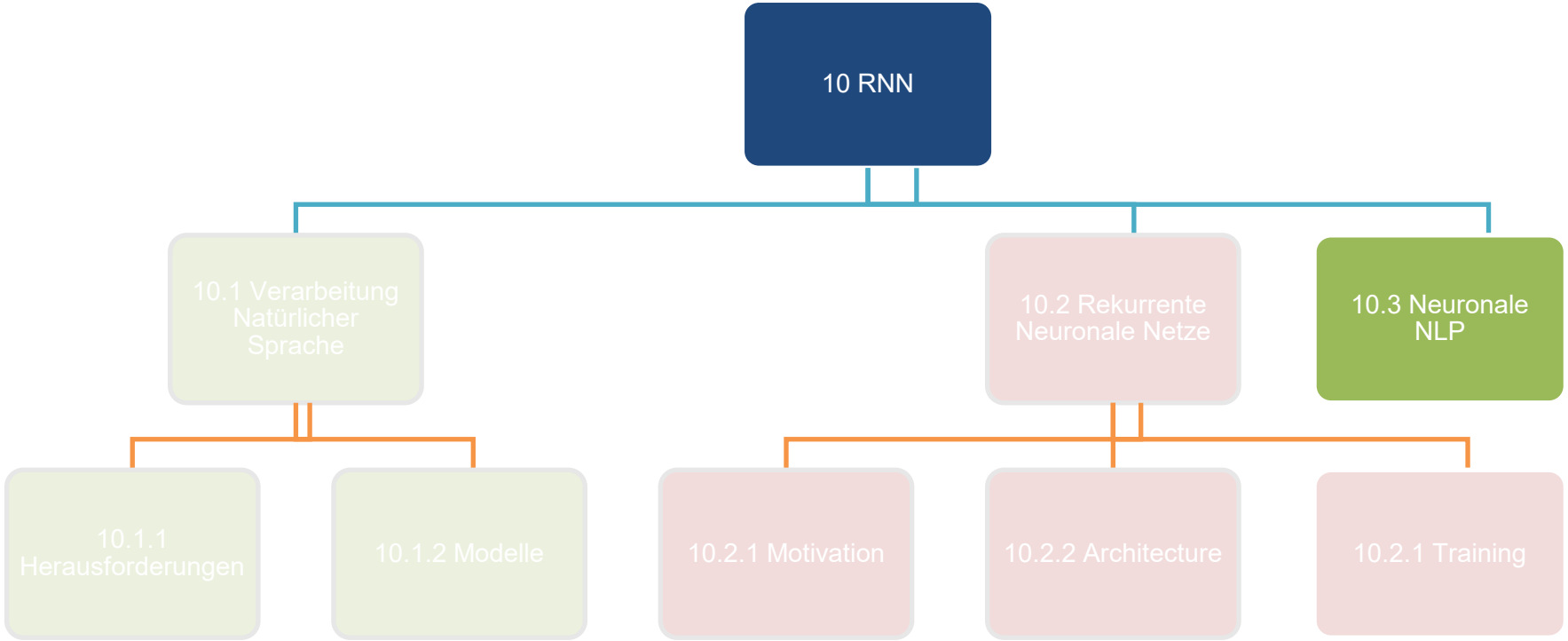
Click on the projected screen to start the question

1 $\frac{\partial l_\tau}{\partial W^h} = \frac{\partial l_\tau}{\partial o_\tau} \frac{\partial o_\tau}{\partial h_\tau} \frac{\partial h_\tau}{\partial W^h}$ 8% 6

2 $\frac{\partial l_\tau}{\partial W^h} = \sum_{t=1}^{\tau} \frac{\partial l_\tau}{\partial o_\tau} \frac{\partial o_\tau}{\partial h_\tau} \frac{\partial h_\tau}{\partial h_t} \frac{\partial h_t}{\partial W^h}$ 53% 40 ✓

3 $\frac{\partial l_\tau}{\partial W^h} = \sum_{t=1}^{\tau} \frac{\partial l_\tau}{\partial o_\tau} \frac{\partial o_\tau}{\partial h_\tau} \frac{\partial h_\tau}{\partial h_t} \frac{\partial h_t}{\partial W^h}$ 39% 29

Überblick


10 RNN
10.1 Verarbeitung Natürlicher Sprache
10.2 Rekurrente Neuronale Netze
10.3 Neuronale NLP
10.1.1 Herausforderungen
10.1.2 Modelle
10.2.1 Motivation
10.2.2 Architecture
10.2.1 Training

Modelltypen

Sequenzklassifikation:

- Eingabe: Sequenz mit variabler Länge
- Ausgabe: Klasse

Beispiel: Sentimentanalyse

„Ich bin mit dem Produkt äußerst zufrieden. Die Qualität ist hervorragend und die Lieferung erfolgte schneller als erwartet. Auch der Kundenservice war freundlich und sehr hilfsbereit.“

„Man könnte meinen, die Qualität sei hochwertig, zumindest wenn man sich an den Beschreibungen orientiert. In der Praxis wirkt das Produkt jedoch alles andere als überzeugend, und die versprochene Zuverlässigkeit habe ich persönlich nicht wirklich erlebt. Wer einen reibungslosen Ablauf und hilfreichen Support erwartet, wird vermutlich ähnliche Erfahrungen machen.“

Grundlegendes Deep Learning Modell für NLP

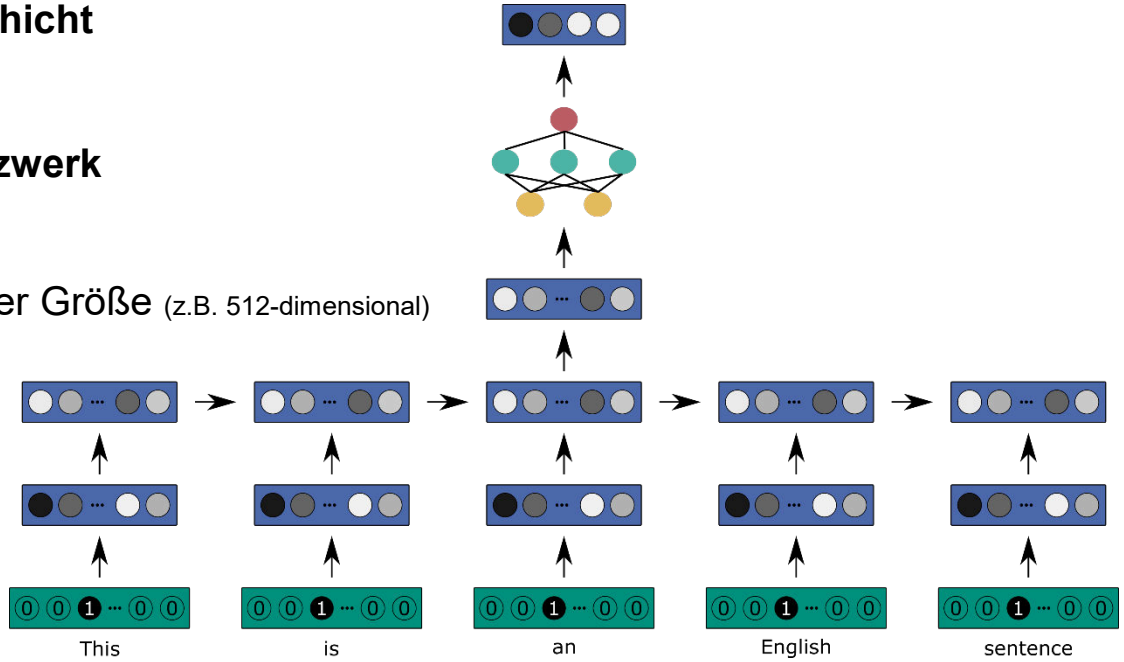
Klassifizierungsschicht

Feed-Forward-Netzwerk

Repräsentation fester Größe (z.B. 512-dimensional)

Encoder

Satz als Eingabe



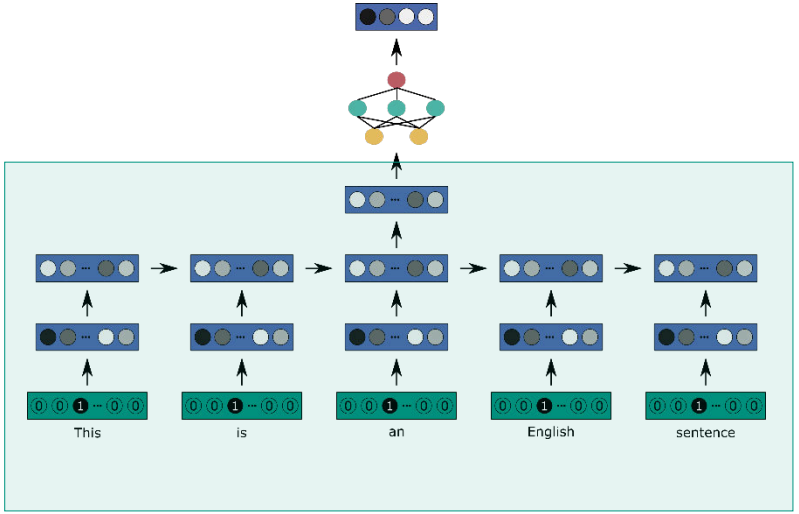
Grundlegendes Deep Learning Modell für NLP

Encoder:

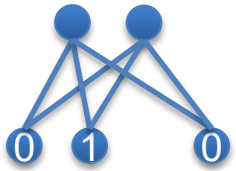
Hier: eine Funktion zur Darstellung einer Wortfolge als Vektor fester Größe.

Teile eines solchen Textencoders:

- Word-Embeddings
- Sequenzschicht
- Aggregationsschicht



Word embeddings



Erst **Zuordnung** von einem **Wort zu** einem kontinuierlichen **Vektor**

- Z.B. **one-hot-encoding Vektoren** von Wörtern/Tokens, bei dem das i-te Wort im Vokabular mit einem 1-bit in der i-ten Position und einer 0 in allen anderen Positionen encodiert ist. Besser: Zuordnung mittels V Integer IDs.
- Sehr ineffizient und kann auch Ähnlichkeiten zwischen Wörtern nicht darstellen.
- Lernen **niedrigerdimensionale Vektoren** die jedes Wort repräsentieren (ähnliche Worte haben dann ähnliche Vektoren).
- Haben eine Sequenz von Wörtern in eine Sequenz von Vektoren gemapped…

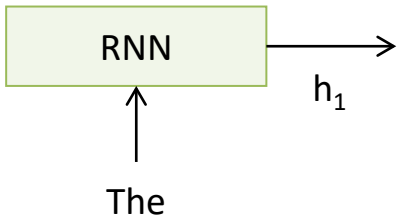
Herausforderungen:

- Die meisten Parameter
- Festes Vokabular

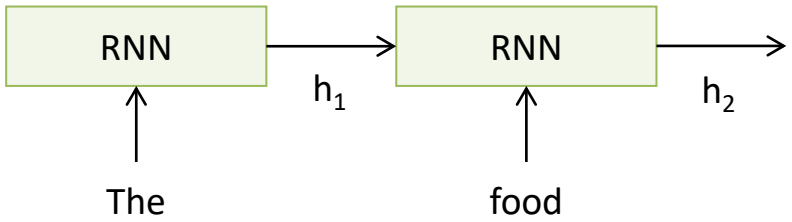
$$
\begin{aligned}
\text{“aardvark”} &= [-0.7, +0.2, -3.2, \ldots] \\
\text{“abacus”} &= [+0.5, +0.9, -1.3, \ldots] \\
&\ldots \\
\text{“zyzzyva”} &= [-0.1, +0.8, -0.4, \ldots].
\end{aligned}
$$

Russel & Norvig, 4th edition, 2021

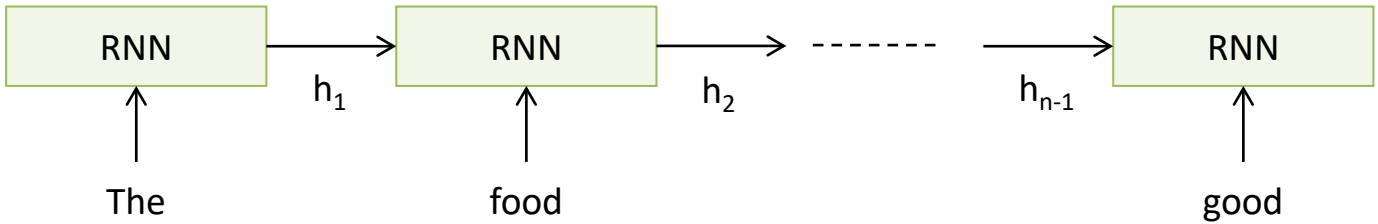
Sequenzschicht (z.B. unser RNN)



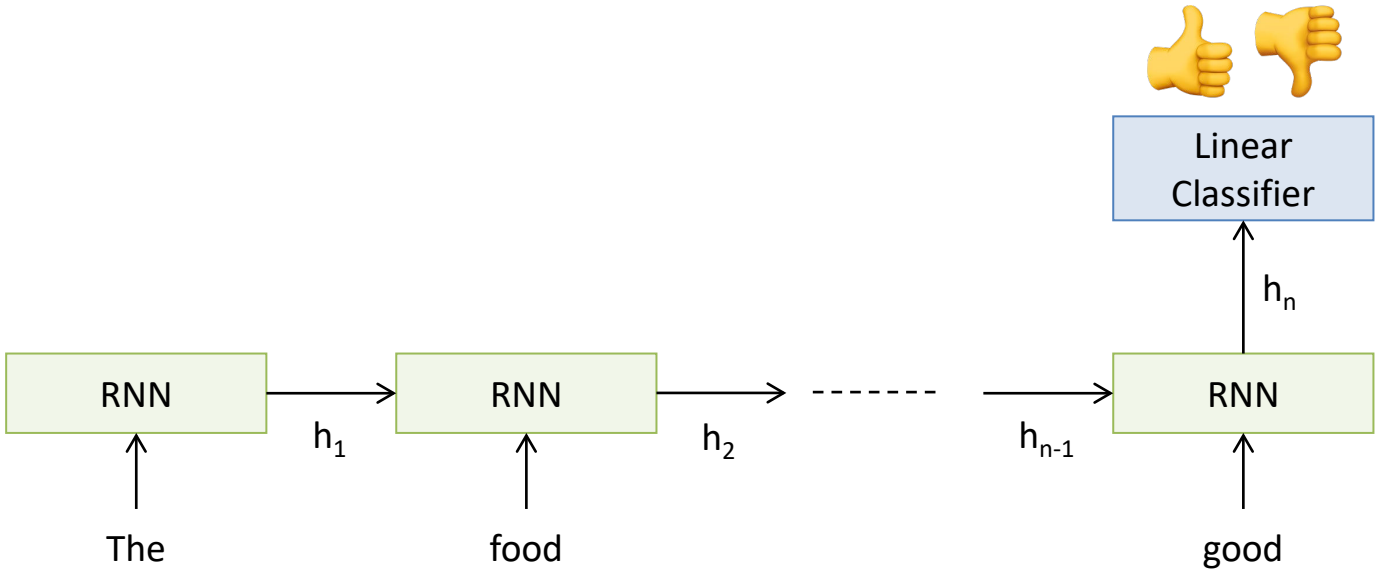
Sequenzschicht (z.B. unser RNN)



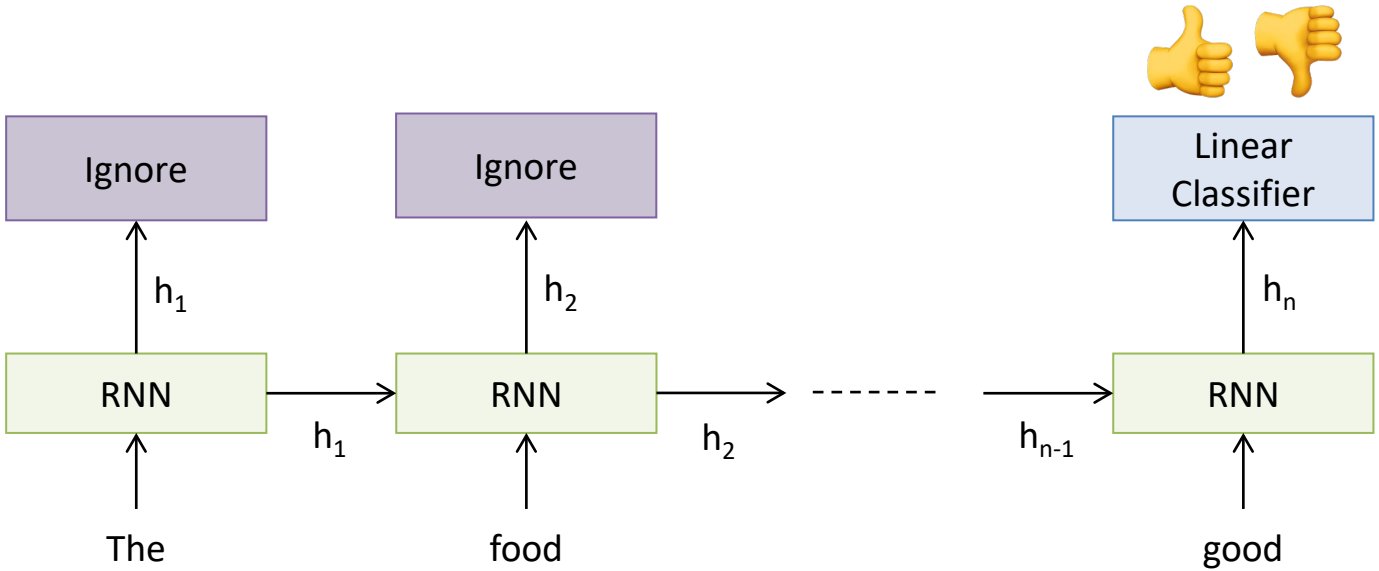
Sequenzschicht (z.B. unser RNN)



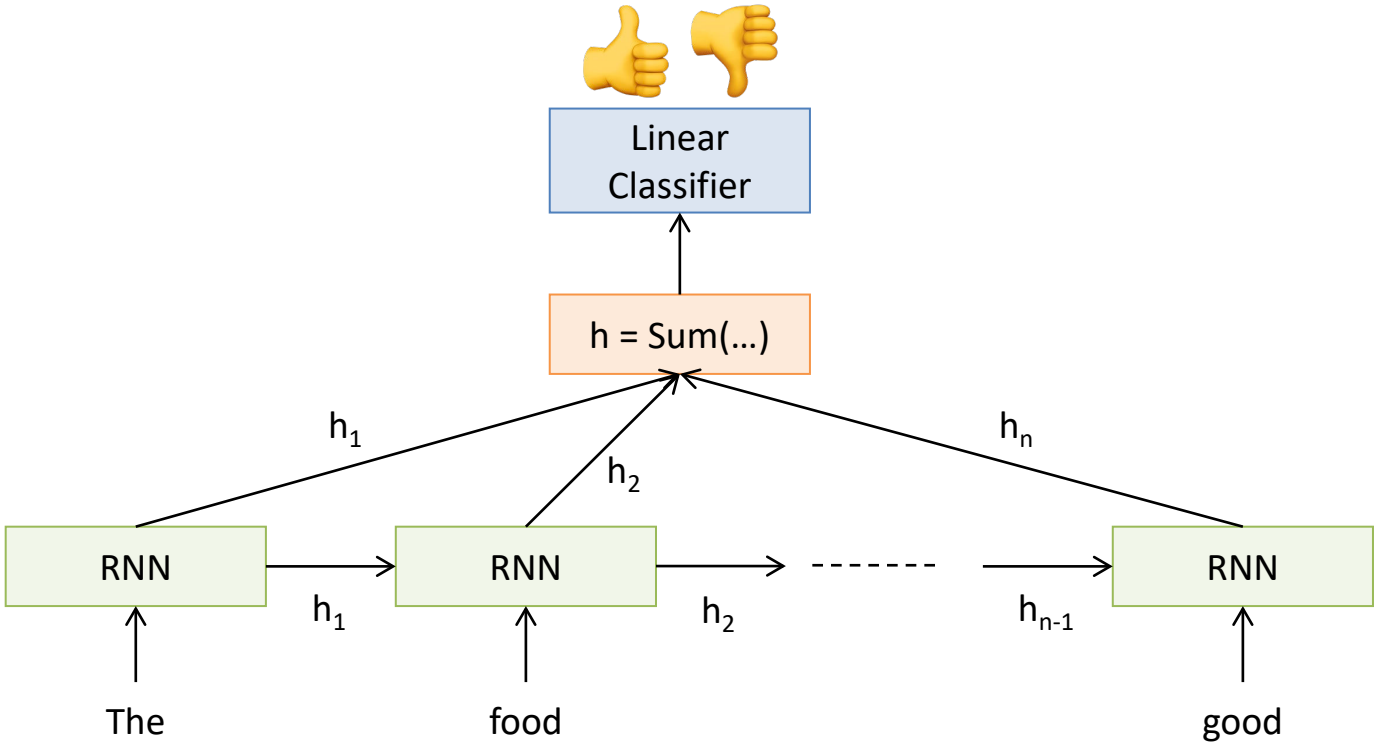
Aggregationsschicht



Aggregationsschicht

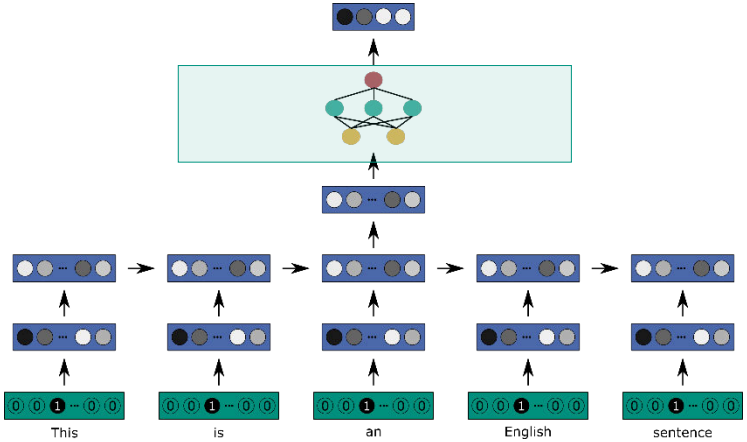


Aggregationsschicht



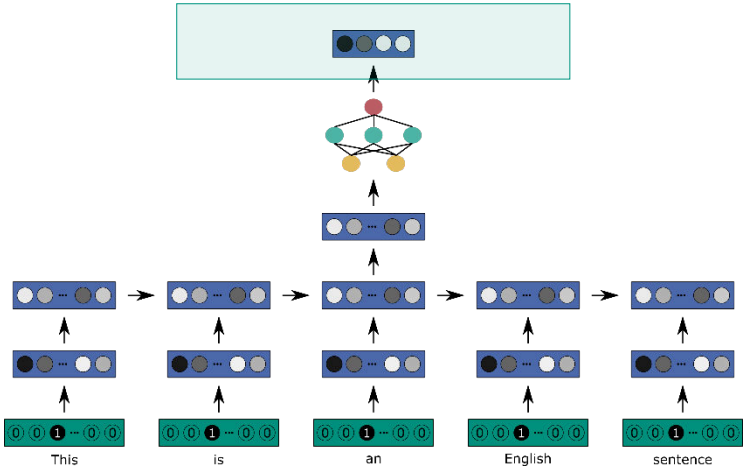
Grundlegendes Deep Learning Modell für NLP

- Encoder
- Neuronale Netzwerkschichten
  - Bessere Darstellung des Satzinhalts
  - Neuronales Feedforward-Netzwerk
  - Eingabe: Darstellung des Satzes in fester Größe
  - Ausgabe: Darstellung mit fester Größe

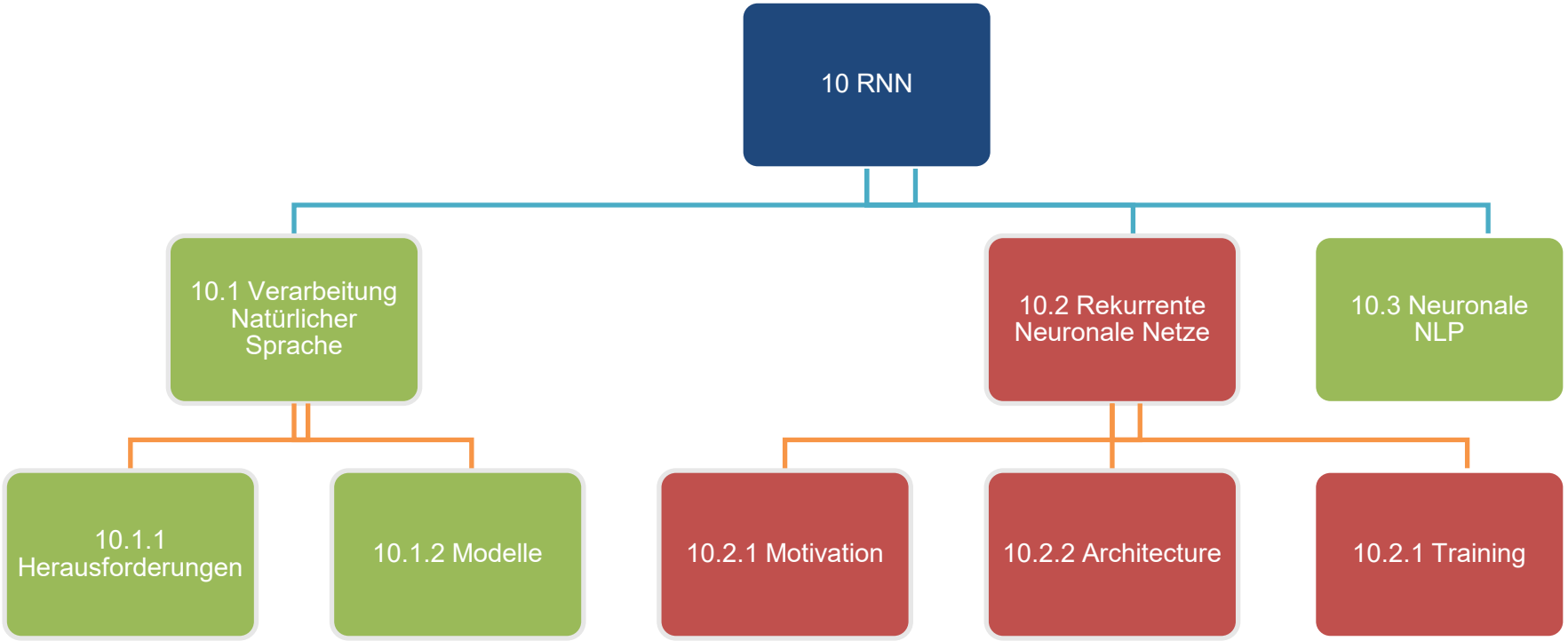


Grundlegendes Deep Learning Modell für NLP

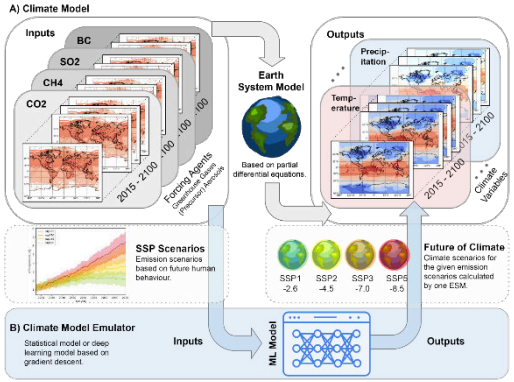
- Encoder
- Neuronale Netzwerkschichten
- Klassifizierungsschicht
  - Softmax-Schicht:
    - Ein Neuron pro Klasse
    - Aktivierung = Wahrscheinlichkeit dieser Klasse
    - $o_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$
  - Loss Funktion: Cross-entropy Loss



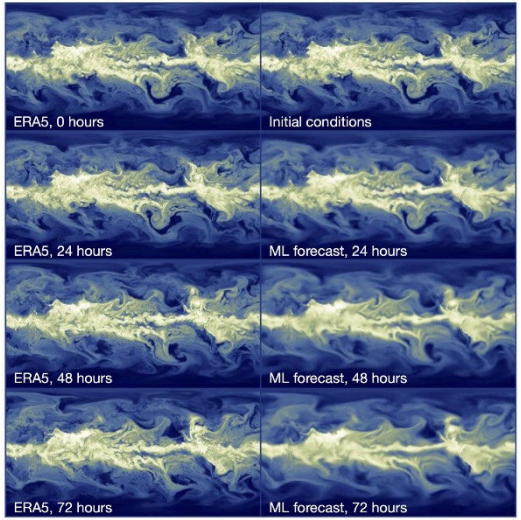
Überblick


10 RNN
10.1 Verarbeitung Natürlicher Sprache
10.2 Rekurrente Neuronale Netze
10.3 Neuronale NLP
10.1.1 Herausforderungen
10.1.2 Modelle
10.2.1 Motivation
10.2.2 Architecture
10.2.1 Training

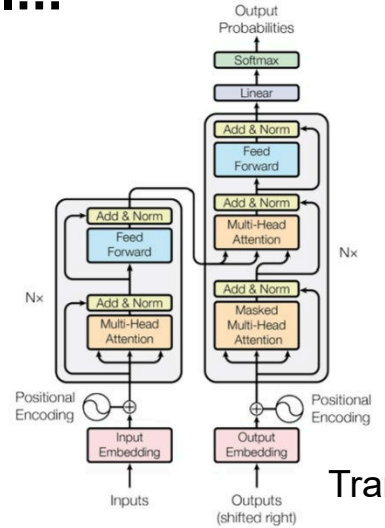
Ausblick auf unser Mastermodul...



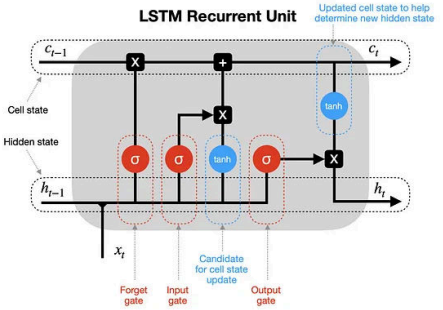
Climate model emulation



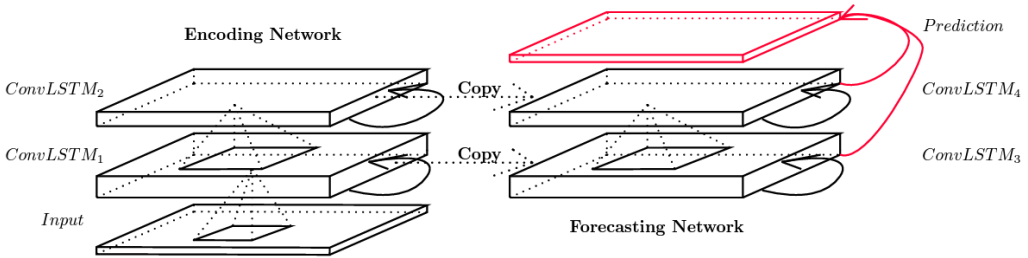
Weather forecasting



Transformers



LSTMs

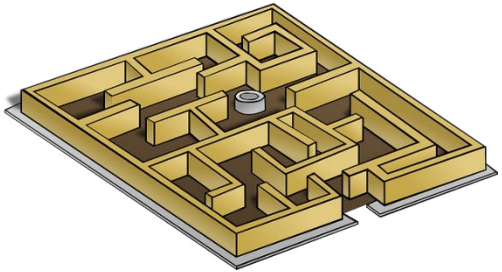


ConvLSTMs

Zusammenfassung

Verarbeitung natürlicher Sprache

Recurrent Neural Network



Herausforderungen

Backpropagation through time

Lehrevaluation

Vielen herzlichen Dank für Ihre Zeit und Ihr konstruktives Feedback!

Übung

https://onlineumfrage.kit.edu/evasys/online.php?p=KPQEH

Vorlesung Friederich

https://onlineumfrage.kit.edu/evasys/online.php?p=Y7FPN