



Klassifikation von Multidimensionalen Zeitreihen mittels Deep Learning

Manuel Meyer

Master Hauptseminar SS 2018

Betreuender Professor: Prof. Dr.-Ing. Andreas Meisel



Outline

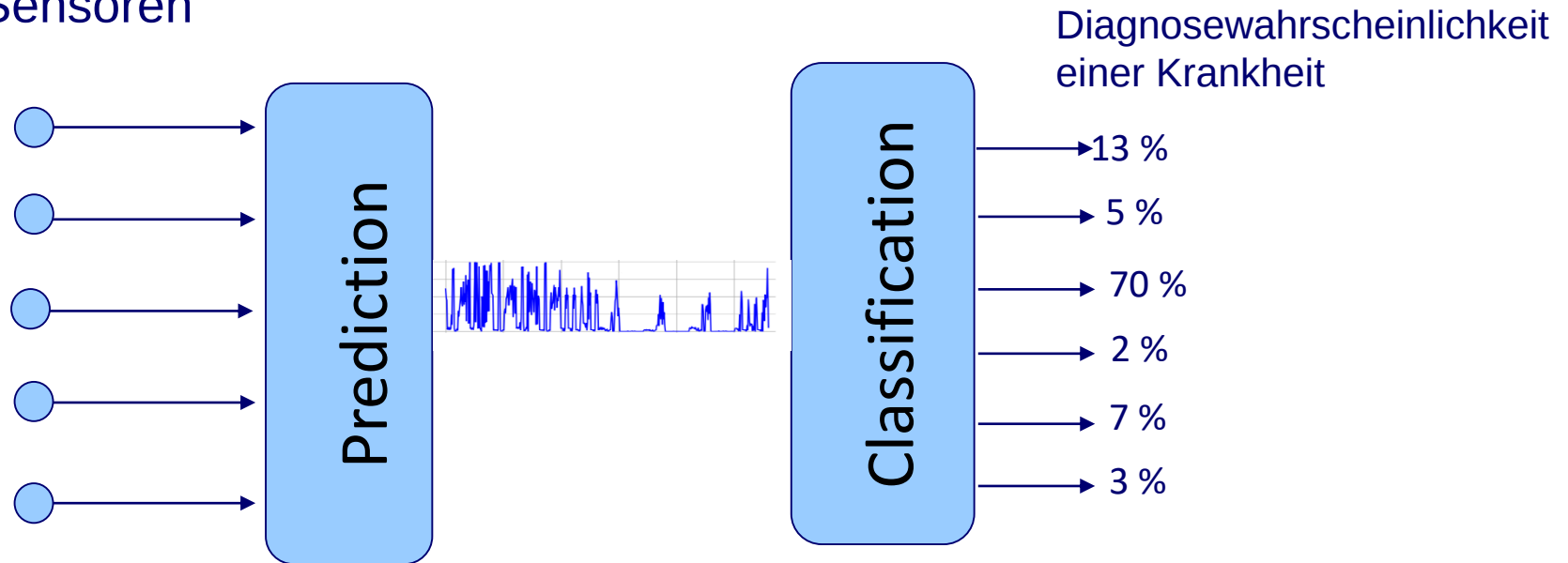
1. **Motivation**
2. **Ein kurzer Rückblick**
3. **Convolutional Neural Networks**
4. **Recurrent Neural Networks**
5. **Architekturen mit gates – LSTM**
6. **Datensatz**
7. **Aktuelle Entwicklung meiner Arbeit**
8. **Zukünftige Entwicklungen meiner Arbeit**
9. **Stand der Technik I+II**



Motivation

- Health Daten

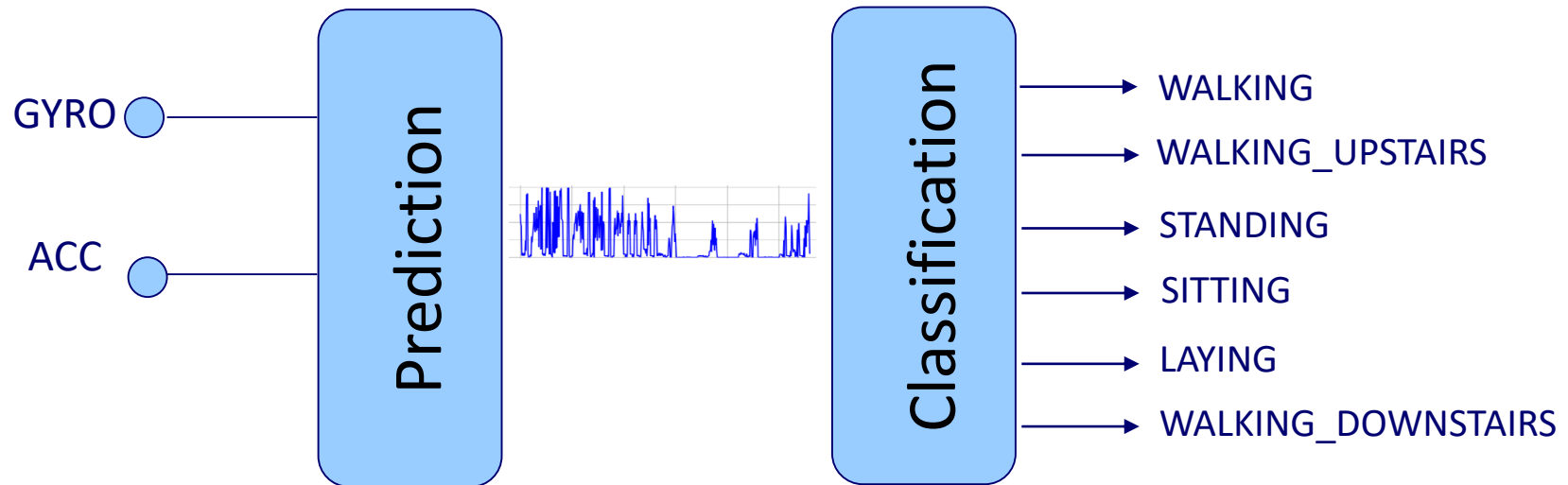
Sensoren





Motivation

- Human Activity Recognition Daten





Ein kurzer Rückblick

- Sensoren erzeugen in zeitlichen Abständen Signale
- Erkennung von Systematiken in Impulsfolgen
 - Vorhersage (zukünftige Ereignisse)
 - Klassifikation
- KNN um nichtlineare Zusammenhänge herzustellen
- Dynamische Verhalten von Systemen lernen
- Convolutional Neuronale Netze + Netze mit Rückkopplung
 - Heterogenen Sensordaten (Multidimensionale Daten)
 - Klassifikation, Vorhersage, Prognose



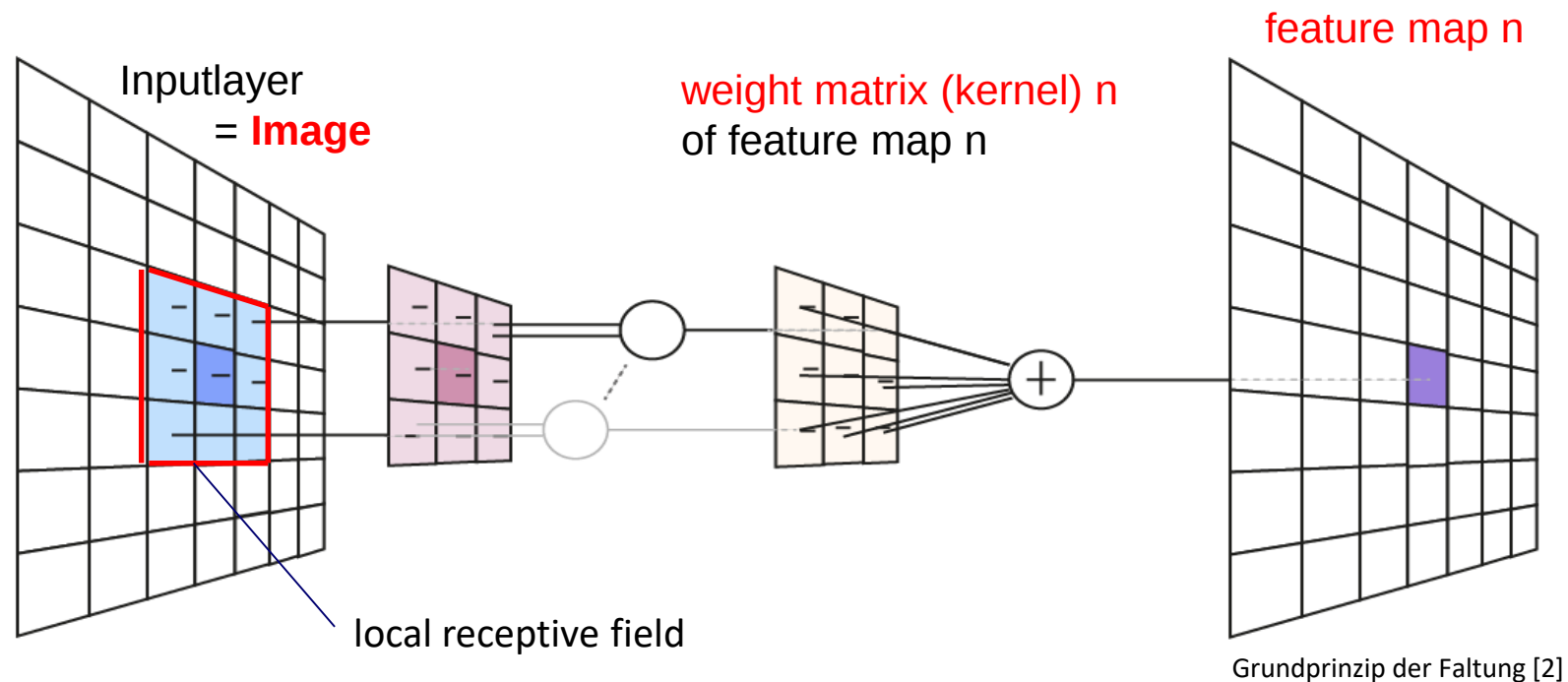
Convolutional Neural Networks

- **Eine Variante von KNN mit mehreren Schichten**
- **Faltungsnetzwerk**
- **Robust und unempfindlich gegen Rotation, Translation, Skalierung, etc.**
- **Liefern gute Ergebnisse: Bilderkennung, Gesichtserkennung, Spracherkennung....**



Faltung

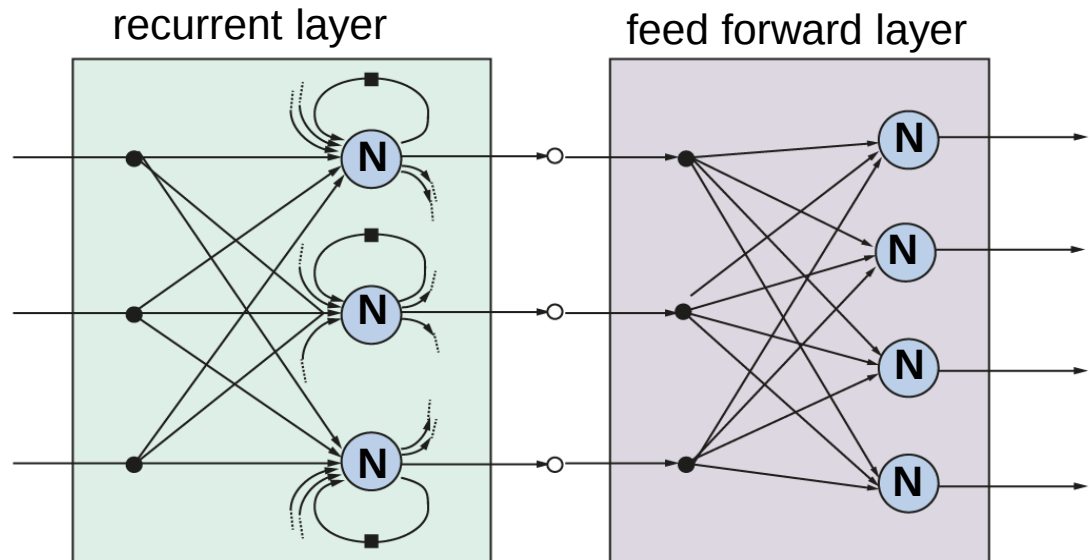
- Eingabedaten werden mittels Faltungsmaske bearbeitet
- Korrespondierende Punkte berechnet -> neuer Pixel wird erzeugt
- Merkmale/Features hervorheben oder unterdrücken
- Faltungskerne trainiert





Recurrent Neural Networks

- Netze mit Rückkopplung
- Können zeitliche Vorgeschichte speichern
- Erkennung von Systematiken in Zeitfunktion
- Zeitreihen und Sequenzen beliebiger Länge
 - Daten, Beobachtungen, Messungen zeitliche sortiert
 - Sensorsignale (ECG, EEG, heart rate acceleration, temperature, ...)
 - Audio ...,
 - Text

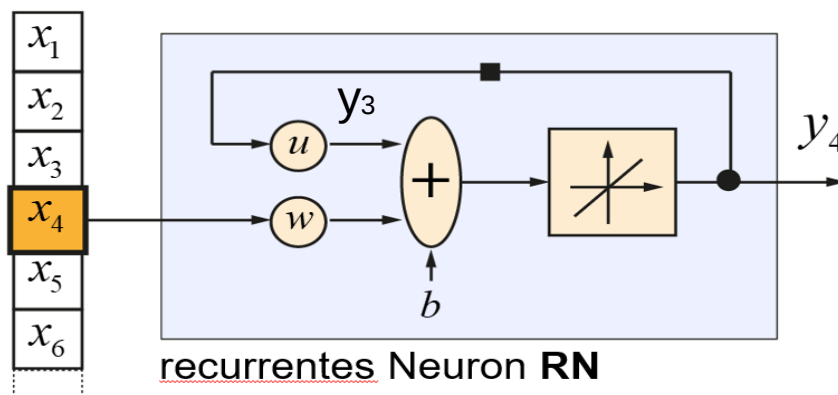


Exemplarische Layer Darstellung eines RNN und FFN[2]

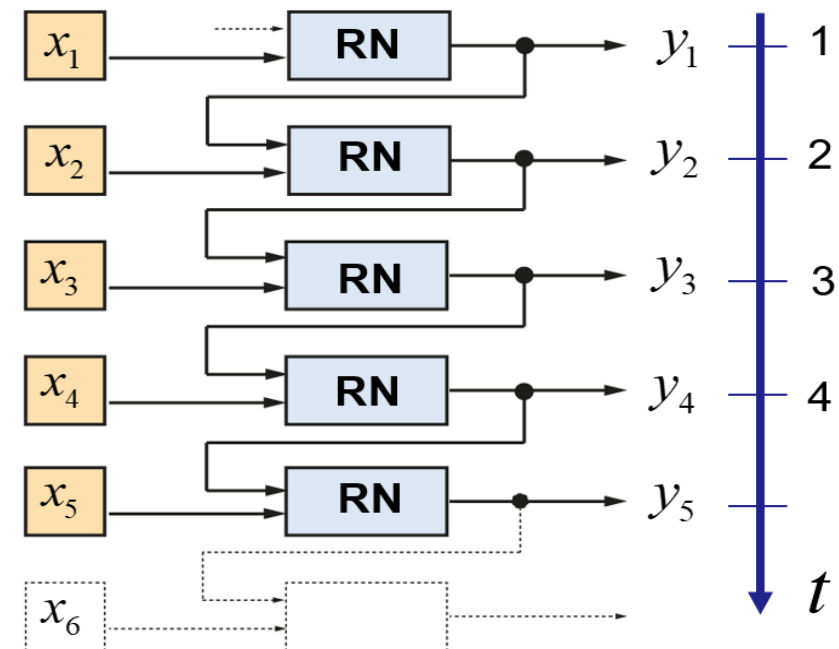


Recurrent Neural Networks – zeitlich aufgerollt

Beispiel mit **single** Neuron



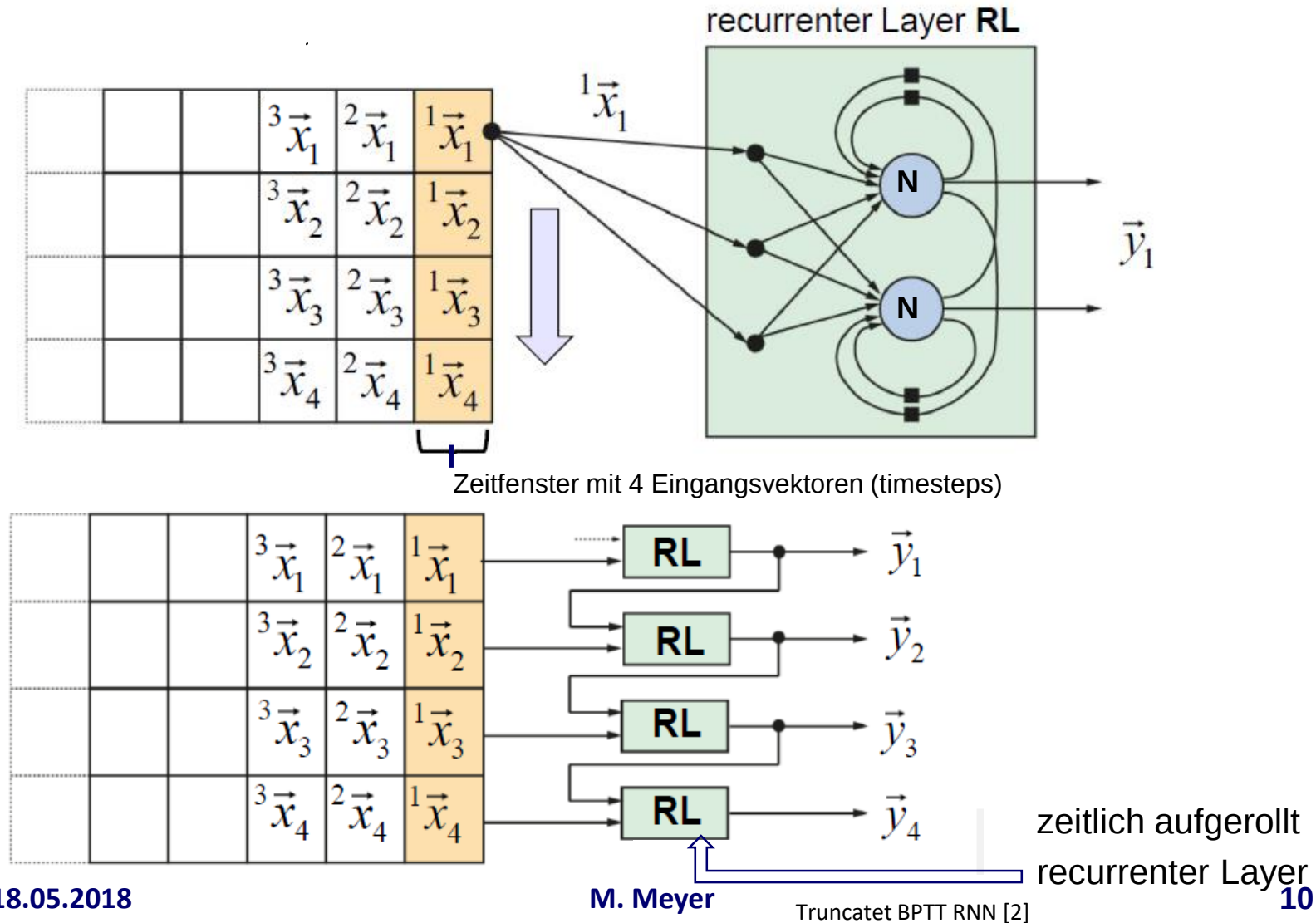
zeitlich aufgerollt



Darstellung eines RNN [2]



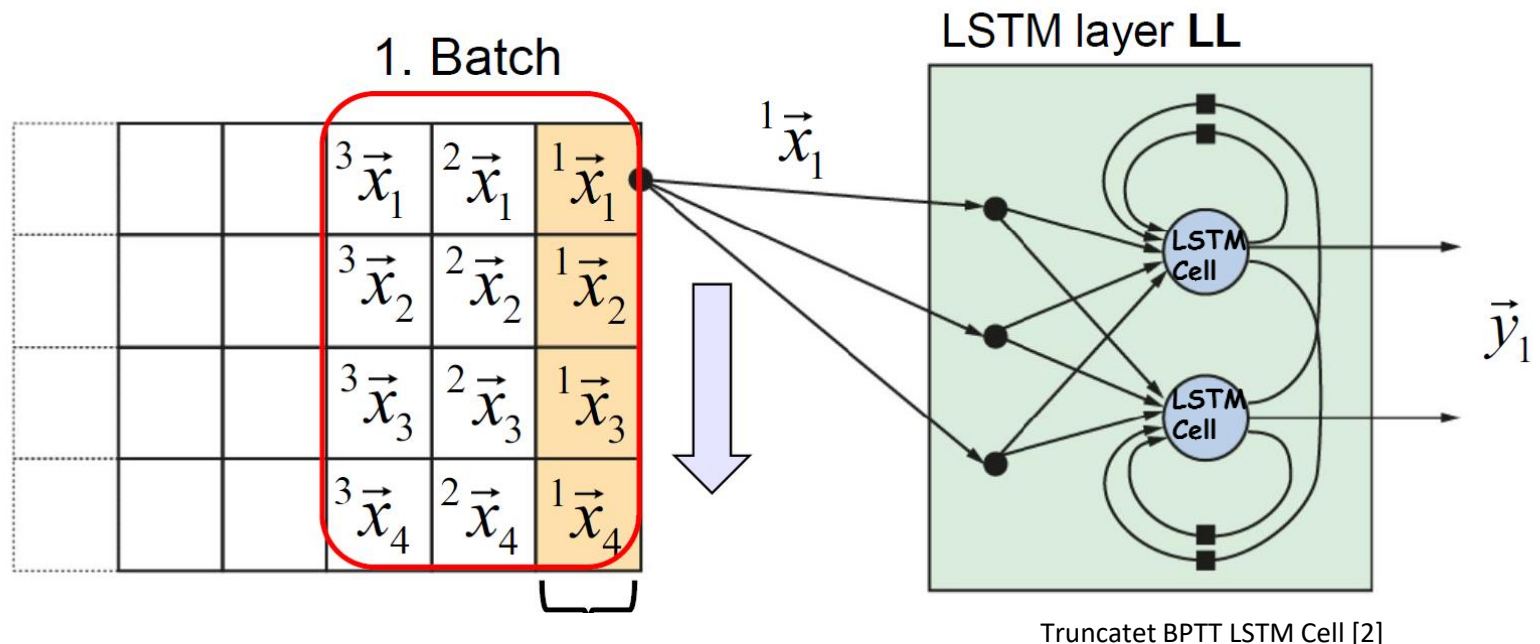
Recurrent Layer – zeitlich aufgerollt





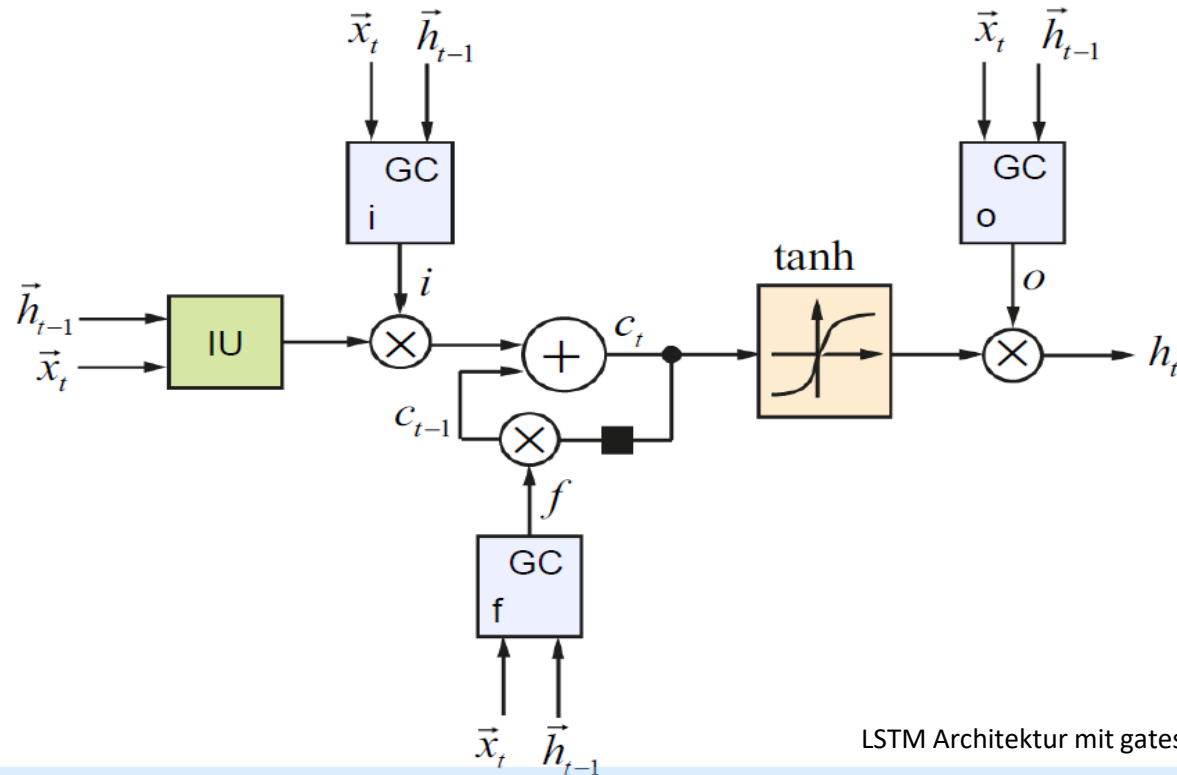
Architekturen mit gates – LSTM layer

- RNNs verliert den Einfluss von früherer Faktoren auf spätere Berechnungen
- LSTM können temporale Abhängigkeiten in Daten weitreichend erkennen
- Deswegen führt man sog. *gates* ein (*input-, output- and forget-gate*)





Architekturen mit gates - LSTM



LSTM Architektur mit gates [2]

Der gate- Mechanismus:

- a) behält *cell*-Informationen für eine lange Zeit während der Arbeit
- b) schützt den Gradient in der Zelle vor *vanishing* während dem Training
- c) das Training konvergiert schneller



Datensatz

A Public Domain Dataset for Human Activity Recognition Using Smartphones^[1]

- **Forschungsteam der Universität (Davide Anguita und sein Team)**
 - **Universitat Politècnica de Catalunya, Spanien**
 - **University of Genova, Italien**
- **Forschungsbereich - Human-Centered Computing**
- **Anwendung mittels Sensoren/Smartphone Aktivitäten zu erfassen**
- **30 Teilnehmer (19-48)**



Datensatz

- **ca. 10.000 Datensätze**
 - 70% Train
 - 30% Test
- **Datensatz auf Plattformen veröffentlicht wie:**
 - UCI Machine Learning Repository
 - Kaggle
 -



Datensatz

- 3-Achsige Signale Gyroskop / Accelerometer (50 Hz)
- 6 Klassifikationen – [WALKING, STANDING, SITTING, LAYING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS]
- Rauschreduzierung durch Medianfilter
- *sliding window* mit der festen Breite 2.56 sec mit 50% Überlappung
 - (2.54 sec x 50Hz = 128 cycles)
- Zusätzlich ein feature Vektor mit 561 Dim.
 - z.B. FFT, Energie, Durchschnitt, Standardabweichung, Integrale/Differenziale, Winkeländerungen, etc.



Aktuelle Arbeit/Stand

- Untersuchung mit feature Vektor 561 Dim.
- Eigene Bibliothek mit nützlichen Funktionen und Transformer:
 - Aufbereiten Daten
 - Umwandeln Daten
 - mit Regularisierungen Probleme entgegenwirken
 - Vanish/Exploding Gradient,
 - Underfitting/Overfitting, etc.
- Zwei Modelle
 - Klassifikation mit CNN+LSTM mit 96 % acc
 - Experimente für Vorhersage mit LSTM mit 20% acc

CNN+LSTM	acc
Train	0.994595
Val	0.978935
Test	0.965337

LSTM_Prediction	acc
Train	0.413944
Val	0.169309
Test	0.192538



Zukünftige Arbeit

- **Untersuchung des Modells mit:**
 - Rohdaten ohne Merkmalsselektion
 - Nach und nach Merkmale hinzufügen/kombinieren
 - Warum verändert sich das Modell bei bestimmten Parameter?
- **Reduzierung Komplexität:**
 - Dimensionsreduktion durch PCA
 - Feste Reihenfolge der 6 Klassifikation
- **Feinabstimmung der Hyperparameter durch Estimator**
- **Fragestellungen:**
 - Welches sind geeignet Merkmale um Aktivitäten zu erkennen?
 - Verbessert die Transformation der Sensordaten die Erkennung?



Stand der Technik I

A Hybrid Deep Representation Learning Model for Time Series Classification and Prediction^[6]

[Yang Guo*, Zhenyu Wu†, Yang Ji*, 16.November 2017]

- 3 Modelle für Zeitreihen Klassifikation und Vorhersage
- Datensatz
 - 5 Teilnehmer
 - 4 Klassifikationen (STAND, WALK, SIT, LIE)
 - Feste sortierte Reihenfolge
 - Input Vektor 113 Dim.
 - ca. 20.000 Trainingsdaten
- Vorverarbeitung
 - missing values -> linear interpolation
 - normalisiert



Stand der Technik I

TABLE I. F_1 score performance on different tasks

Model	Classification	Prediction
LR-ConvLSTM	87.47%	80.21%
LRCN	86.54%	74.25%
DeepConvLSTM	84.34%	null

- **LR-ConvLSTM**
 - CNN + LSTM (2xLL) + LSTM (2xLL)
- **LRCN (CNN+LSTM)**
 - Vorhersage mit Variabler Länge TIMSTEPS führten zum selben Ergebnis
- **DeepConvLSTM (CNN+LSTM)**
 - Keine Vorhersage aufgrund unterschiedlichen Inputvektor
- **TIMESTEPS: 12, 24, 36, 48**

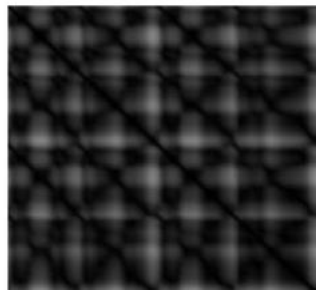
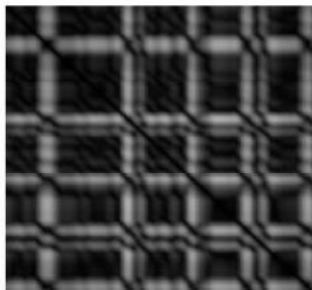


Stand der Technik II

Effect of Dynamic Feature for Human Activity Recognition using Smartphone Sensors [8]

[Kotaro Nakano, Basabi Chakraborty, 15.Jan 2018]

- Selber Datensatz wie in dieser Arbeit
- Vorverarbeitung:
 - Statische Merkmale:
 - Standardabweichung
 - Dynamische Merkmale:
 - Sensordaten in *recurrence plot* konvertiert



- moderne Methode der nichtlinearen Datenanalyse
- Visualisierung rekurrenter Zustände
- dient zur Darstellung von höherdimensionale Räume



Stand der Technik II

- **Verwendete Klassifizierer**
 - **CNN (Convolutional Neural Network)**
 - **MLP (multilayer perceptron)**
 - **SVM (support vector machines)**
 - **k-NN (k-nearest neighbor)**
-
- Dynamische Merkmale**
- Statische Merkmale**



TABLE I
CLASSIFICATION ACCURACY OF DIFFERENT CLASSIFIERS WITH STATIC
FEATURES

Activity	MLP	SVM	k-NN
Walking	0.9899	0.9859	0.9778
Walking Upstairs	0.9363	0.9660	0.9002
Walking downstairs	0.9333	0.9405	0.7857
Sitting	0.8941	0.9063	0.7923
Standing	0.9680	0.9831	0.9304
Laying	0.9963	1.00	0.9944
av.of first 3	0.9554	0.9652	0.9011
av. of all	0.9014	0.9634	0.9012

TABLE II
CLASSIFICATION ACCURACY OF CNN WITH DYNAMIC FEATURES

Activity	Accelerometer	Gyro	Both
Walking	0.9859	0.9012	0.9980
Walking upstairs	0.9681	0.9681	0.9703
Walking downstairs	0.9857	0.8571	0.9905
Sitting	0.4400	0.6680	0.7739
Standing	0.7368	0.8308	0.8909
Laying	0.7765	0.6164	0.8100
av. of first 3	0.9799	0.9088	0.9862
av.of all	0.8155	0.8069	0.9056



Danke für Ihre Aufmerksamkeit

Gibt es noch Fragen?



Literaturverzeichnis

- [1] Anguita, D. ; Ghio, A. ; Oneto, L. ; Parra, X. ; L Reyes-Ortiz, J. : A Public Domain Dataset for Human Activity Recognition using Smartphones. (2013), 01
- [2] A. Meisel. Vorlesungsskript „Modellierung dynamischer Systeme “ 2018.
- [6] Guo, Y. ; Wu, Z. ; Ji, Y. : A Hybrid Deep Representation Learning Model for Time Series Classification and Prediction. In: 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM), 2017, S. 226-231
- [7] Kim, Y. ; An, J. ; Lee, M. ; Lee, Y. : An Activity-Embedding Approach for Next- Activity Prediction in a Multi-User Smart Space. In: 2017 IEEE International Conference on Smart Computing (SMARTCOMP), 2017, S. 1-6
- [8] Nakano, K. ; Chakraborty, B. : Effect of dynamic feature for human activity recognition using smartphone sensors. In: 2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST), 2017, S. 539-543