LSTMs ZUR ERKENNUNG MENSCHLICHER AKTIVITÄT

Kolloquium Christoph Jabs 29.01.2020 Hochschule Reutlingen

BOSCH

Motivation Welche Frage soll beantwortet werden?

- Sind Long-Short-Term-Memory Netzwerke zur Analyse von Zeitreihendaten inertialer Sensoren geeignet?
- → Beantwortung der Frage basierend auf der Erkennung menschlicher Aktivität



Agenda

- 1. Human Activity Recognition
- 2. LSTMs zur Erkennung menschlicher Aktivität
- 3. Experimente und Ergebnisse
 - 1. Architekturstudie
 - 2. Vergleich mit CNNs
- 4. Zusammenfassung und Ausblick



Human Activity Recognition Was ist HAR? / Datensätze

Erkennung und Klassifizierung menschlicher Aktivität aus Inertialsensordaten

- Anwendungen
 - Aktivitätstracking in Smartphones/-watches
 - Aufzeichnung eines Gesundheitsprofils
- Verwendete Sensoren
 - Beschleunigungssensor
 - Gyroskop
 - Magnetometer
- Aktives Forschungsfeld
 - ► Viele Ergebnisse für Vergleiche

Property	Opportunity [4]	mHealth [2]	
Subjekte	4	10	
Daten	8 Stunden	6,75 Stunden	
Sensorpos.	14 (verwendbar und am Körper)	3	
Aktivitäten	4 (+Null Klasse)	5 (+Null Klasse) (nach Reduzierung)	



Human Activity Recognition Beispiele aus den Daten / Feature-Extraktion



Automotive Electronics | Christoph Jabs (AE/PJ-SW4) | 29.01.2020



LSTMs zur Erkennung menschlicher Aktivität Was sind LSTMs?

Long-Short-Term-Memory

- Form von Recurrent Neural Networks (RNN)
- RNN plus zusätzliches Langzeitgedächtnis
- Reduziertes Risiko von explodierenden/verschwindenden Gradienten
 - ► LSTM: $c_t = f c_{t-1} + ig$
 - Gradient enthält Multiplikation von f
 - $\blacktriangleright \text{ RNN: } c_t = \Phi(W_{xh}x_t + W_{hh}c_{t-1})$
 - Gradient enthält Multiplikation von W_{hh}



Input Layer x_t

Experimente und Ergebnisse Architekturstudie – Überblick

- Vier grundlegende Architekturen
- Basierend auf Daten von Opportunity
- Insgesamt 380 Netzwerke
 - ► 95 Varianten pro grundlegender Architektur



Grundlegende Architekturen

- SequenceLSTM
- SequenceLSTMnoFC
- RelaxedLSTM





BOSCH

Automotive Electronics | Christoph Jabs (AE/PJ-SW4) | 29.01.2020

Experimente und Ergebnisse Ergebnisse auf Opportunity – Vergleich mit CNN



Automotive Electronics | Christoph Jabs (AE/PJ-SW4) | 29.01.2020

© Robert Bosch GmbH 2020. Alle Rechte vorbehalten, auch bzgl. jeder Verfügung, Verwertung, Reproduktion, Bearbeitung, Weiterg abe sowie für den Fall von Schutzrechtsanmeldungen.

BOSCH

Experimente und Ergebnisse Ergebnisse auf Opportunity – Vergleich mit CNN – Details



 (\square)

BOSCH

Q Automotive Electronics | Christoph Jabs (AE/PJ-SW4) | 29.01.2020

Experimente und Ergebnisse Ergebnisse auf mHealth – Vergleich mit CNN



10 Automotive Electronics | Christoph Jabs (AE/PJ-SW4) | 29.01.2020



Experimente und Ergebnisse Ergebnisse auf mHealth – Vergleich mit CNN – Details



 (\mathbb{H})

BOSCH

11 Automotive Electronics | Christoph Jabs (AE/PJ-SW4) | 29.01.2020

LSTMs eignen sich zur Erkennung menschlicher Aktivität und generell zur Analyse von Zeitreihendaten inertialer Sensoren.

12 Automotive Electronics | Christoph Jabs (AE/PJ-SW4) | 29.01.2020 © Robert Bosch GmbH 2020. Alle Rechte vorbehalten, auch bzgl. jeder Verfügung, Verwertung, Reproduktion, Bearbeitung, Weiterg abe sowie für den Fall von Schutzrechtsanmeldunger



Zusammenfassung und Ausblick Weiterführende Arbeit

- ► Mehr Trainingsdaten
 - ► Größere Datensätze
 - Datensätze kombinieren
- Training auf komplexeren Sequenzen
 - Sequenzen variiernder Länge
 - Klassenübergänge in Sequenzen
- Optimierung der Features
 - Weniger Features
 - Optimierung der Frequenzbänder
- CNN-LSTM [5]



VIELEN DANK!



Bibliografie Referenzen der Präsentation

- [1] Charu C. Aggarwal. *Neural Networks and Deep Learning*. Springer International Publishing, 2018.
- [2] Oresti Banos et al. 'Design, implementation and validation of a novel open framework for agile development of mobile health applications'. In: *BioMedical Engineering OnLine* 14 Suppl 2 (2015), S6.
- [3] Sojeong Ha and Seunhjin Choi. 'Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors'. In: *2016 International Joint Conference on Neural Networks (IJCNN).* IEEE, July 2016.
- [4] Daniel Roggen et al. 'Collecting complex activity datasets in highly rich networked sensor environments'. In: 2010 Seventh International Conference on Networked Sensing Systems (INSS). IEEE, June 2010.
- [5] Tara N. Sainath, Oriol Vinyals, Andrew Senior and Hasim Sak. *Convolutional, Long Short-Term Memory fully connected Deep Neural Networks.* IEEE, April 2015.



Experimente und Ergebnisse Architekturstudie – Ergebnisse







Experimente und Ergebnisse Vergleich der unterschiedlichen Ansätze

Dataset	Accuracy	LSTM ¹ (beste ValAcc)	LSTM ² (beste SeqAcc)	CNN2D ³
Opportunity	Validation Accuracy	78,75%	75,15%	75,62%
	Sequence Accuracy	68,79%	72,91%	61,99%
mHealth	Validation Accuracy	77,81%	84,51%	76,50% ⁴
	Sequence Accuracy	74,51%	80,03%	75,86%

¹ RelaxedLSTMnoFC; 2 Layer (100, 30)

² SequenceLSTM; 4 Layer (50, 30, 20, 10)

³ Basierend auf [3]

⁴ Quelle erreicht >84%



Datensätze Opportunity Sequenzübersicht



Automotive Electronics | Christoph Jabs (AE/PJ-SW4) | 29.01.2020



Datensätze mHealth Sequenzübersicht



19 Automotive Electronics | Christoph Jabs (AE/PJ-SW4) | 29.01.2020



Feature Extraktion Überblick über die Features

Signal Magnitude Area

•
$$SMA = \frac{1}{T} \int_0^T x^2 + y^2 + z^2$$

- Spectral Entropy
 - $H = \sum_{i=1}^{N} p_i \log p_i$

- Leistung in bestimmten Frequenzbändern
 - ▶ 0 2.5Hz
 - ▶ 2.5 5Hz
 - ▶ 5 7.5Hz
 - ▶ 7.5 10Hz
 - ▶ 10 15Hz



How to train your Network Basic Layer Overview

- ► Fully Connected Layer
 - ► Goal: weigh all inputs into one output
 - ▶ Parameters: weights *W* and bias *b*
 - Sometimes with activation function
 - Equation: $Y = W \cdot X + b$
- Softmax Layer
 - ► Goal: norm inputs to [0,1]
 - Parameters: None

• Equation: $Y_i = \frac{\exp(x_i)}{\sum_i \exp(x_i)}$

► *Y*: Output, *X*: Input

- ► Pooling Layers
 - ► Goal: reduce input dimension, gain invariance
 - Parameters: Padding, Stride (=step size), size



- Equation: Complicated
- Dropout Layers
 - ► Goal: increase input significance
 - Parameters: dropout probability
 - ► Leaves out different nodes each full cycle



How to train your Network RNNs and LSTMs

- Feedforward Net not capable of accounting for previous, no longer available information
- RNNs work like FFNs, but store a hidden state that is weighed into the activation function input and updated each iteration
- Problem: Backpropagation leads to exploding/vanishing gradient
- LSTM Layer reduces vanishing/exploding gradient problem via forget gate (RNN gradient for comparison)

$$\frac{\partial E_{k}}{\partial W} = \frac{\partial E_{k}}{\partial H_{k}} \frac{\partial H_{k}}{\partial C_{k}} \frac{\partial C_{k}}{\partial C_{k-1}} \cdots \frac{\partial C_{2}}{\partial C_{1}} \frac{\partial C_{1}}{\partial W}$$

$$\approx \frac{\partial E_{k}}{\partial H_{k}} \frac{\partial H_{k}}{\partial C_{k}} \left(\prod_{t=2}^{k} \sigma(W_{f} \cdot [H_{t-1}, X_{t}]) \right) \frac{\partial C_{1}}{\partial W}$$

$$\neq \frac{\partial E_{k}}{\partial H_{k}} \frac{\partial H_{k}}{\partial C_{k}} \left(\prod_{t=2}^{k} \sigma(W_{f} \cdot [H_{t-1}, X_{t}]) \right) \cdot W_{c} \right) \frac{\partial C_{1}}{\partial W}$$
(2) Automotive Electronics | Christoph Jabs (AE/PJ-SW4) | 29.01.202



• $i_t = \sigma_g (W_i x_t + R_i h_{t-1} + b_i)$ Input Gate - controls what new information will be encoded into the cell state $f_t = \sigma_g (W_f x_t + R_f h_{t-1} + b_f)$ Forget Gate - controls what information in cell state to forget

 $g_t = \sigma_c (W_g x_t + R_g h_{t-1} + b_g)$ Cell Candidate – controls the importance of each new information

 $o_t = \sigma_g (W_o x_t + R_o h_{t-1} + b_o)$ Output Gate - controls what cell state is sent to the network as input in the following time step

• σ_g = gate activation fcn, σ_c = state activation fcn

$$\bullet \quad c_t = f_t \ c_{t-1} + i_t \ g_t \ \text{Cell State}$$

 $h_{t} = o_t \sigma_c(c_t)$ Hidden State



How to train your Network CNNs

Convolutional Neural Nets convolute an input, i.e. parse the input using different filters. A convolution is defined as

$$(fst g)(x):=\int_{\mathbb{R}^n}f(au)g(x- au)\mathrm{d} au$$

- The filters/weights are trained to detect patterns in the input and are well suited to image classification
- Sensor data can be interpreted as a 1-channel image, where the width corresponds to time



