On Attention Models for Human Activity Recognition

Vishvak S. Murahari
Georgia Institute of Technology
Atlanta, USA
vishvak.murahari@gatech.edu

Thomas Plötz
Georgia Institute of Technology
Atlanta, USA
thomas.ploetz@gatech.edu

ABSTRACT

Deep Learning methods have become very attractive in the wider, wearables-based human activity recognition (HAR) research community. The majority of models are based on either convolutional or explicitly temporal models, or combinations of both. In this paper we introduce attention models into HAR research as a data driven approach for exploring relevant temporal context. Attention models learn a set of weights over input data, which we leverage to weight the temporal context being considered to model each sensor reading. We construct attention models for HAR by adding attention layers to a state-of-the-art deep learning HAR model (DeepConvLSTM) and evaluate our approach on benchmark datasets achieving significant increase in performance. Finally, we visualize the learned weights to better understand what constitutes relevant temporal context.

ACM Classification Keywords

H.1.2. User/Machine System; I.5. Pattern Recognition

Author Keywords

Activity Recognition; Attention; Deep Learning

INTRODUCTION

In Human Activity Recognition (HAR) we analyze and model sequential, that is time-series data. In order to do so we need to look into the temporal context of every single sensor reading, which forms the basis for modeling and eventually recognition. This has traditionally been done through sliding window approaches [2], which use a fixed size window to model the temporal context of every single sensor reading. Sliding window procedures also (and still) play a crucial role for many recent Deep Learning based HAR methods. For example, Convolutional Neural Networks (CNNs) in HAR employ a sliding window procedure to map the timeseries data to a fixed 2D representation that is fed into the convolution layers [9].

Decisions regarding any sliding window procedure are hard and often final decisions that impact the recognition procedure as a whole. As such mistakes made here are critical and errors made are difficult to recover from.

ISWC '18, October 8–12, 2018, Singapore, Singapore

ISBN 978-1-4503-5967-2/18/10…$15.00

DOI: https://doi.org/10.1145/3267242.3267287

Alternative approaches use sequential models that could overcome aforementioned issues through explicit segmentation of the activities of interest. The recent adoption of recurrent (deep) neural networks for HAR applications has led to impressive recognition results, but these methods come with their own set of problems [6]. For example, Long Short Term Memory (LSTM [7] models can learn infinite temporal contexts. However, it is not reasonable to assume that an event in the distant past would actually influence current events in the same way a more recent event does. Vanilla modeling is not able to capture such aspects. This is even more pertinent for HAR problems as there is typically only little, if any, relation between current and distant past activities [2].

Such observations lead us to the question about what would be the temporal context that is actually relevant for a model to consider in order to successfully represent activities of interest, and whether a model could make such a decision automatically. If that was the case, then an externalization of such a data-driven decision regarding the relevant temporal context would lead to insights about the analyzed data, possibly up to improved segmentation procedures. Ultimately, we aim for a model to automatically learn its relevant temporal context.

In this paper we build on sliding window based analysis procedures for the initial capture of temporal context. We then integrate attention models to automatically fine-tune the weights of past sensor readings, and thus to adapt their importance for label prediction in HAR. Essentially, attention models help a model learn a set of weights over a set of representations–data input–which signify the relative importance of each of the representations. For the case of activity analysis, these models would learn the contributions of all previous sensor readings that are considered for the analysis of a sample. We use attention models for supervised learning tasks in HAR giving the model the ability to generate weight distributions over the history of a sample. In doing so the model is incentivized to generate weights which place the weight on the context that is relevant for a classification decision.

We explore the potential of attention models by adding an attention layer to a state-of-the-art, deep learning based HAR model – DeepConvLSTM [9]. We evaluate our approach on standard benchmarks (Opportunity, PAMAP2, Skoda), and results demonstrate significantly increased performance over current approaches, which emphasizes the relevance of the proposed approach. We further explore what the models have learned by visualizing the attention model's weights, which provides additional insights into the model behavior.

BACKGROUND

Recent work in sequence modeling in HAR has mainly focused on convolutional networks (CNN), and on recurrent models such as LSTMs. The attraction of CNNs lies in the fact that end-to-end learning is possible due to their stacked filtering layers that automatically capture hierarchical feature representations of the data. Combined with clever combinations of pooling, that is, subsampling, and linear layers, very powerful recognition systems have been realized [1, 15, 14]. CNNs are applicable for analyzing sequential data only because of the sliding window trick, that slices out–typically fixed size–analysis frames from the input sequence of sensor data and promotes these through the network independently. Some recent work has randomized this sliding window procedure in order to generate data variability that is exploited in Ensemble based approaches [4]. However, the general sliding window principle remains unchanged.

Recurrent Models have also been applied very successfully in challenging HAR scenarios. The vast majority of these approaches is based on–variants of–LSTMs [7]. Such models incorporate specific gates into individual cells that allow for keeping an internal memory by feeding back a cell's output and by keeping track of the internal state. [5] extensively analyzed the behavior of deep learning models in the wider HAR context, and one of the most promising current models represents a combination of CNNs–for representation learning– and LSTMs–for sequence learning: DeepConvLSTM [9].

ATTENTION FOR HAR

Previous deep learning approaches have focused on representing and modeling a fixed size temporal context for all sensor readings. Arguably, this approach works very well as such models currently dominate the most challenging HAR benchmarks (such as Opportunity [3]). However, especially such challenging tasks exhibit both a substantial intra- as well as inter-class variance with regards to durations of the activities (cf. [4] for a detailed analysis of current benchmark datasets). As such, using fixed window lengths with uniform distribution of sample weights will not naturally lead to ideal modeling and hence classification performance.

Instead, we explore how attention models can be employed for automatically determining the temporal context that is relevant for modeling activities. Attention models have been introduced for natural language processing tasks for part of speech (POS) tagging [8]. The formal idea is that a set of linear layers and non-linearities are used to learn weights over k vectors each of dimension d. Most architectures have a set of linear layers that map the dimension of these d-dimensional vectors to a one-dimensional score and these scores are then passed through the Softmax function to give the set of k weights. The way each of these k vectors is mapped to a one dimensional score is architecture-specific. For instance, a linear layer could directly map the d-dimensional vector to a single dimension or one could add an intermediate linear layer to initially map the d-dimensional vector to, for example, a dimension $d/2$ and then a subsequent linear layer to map the $d/2$ dimensional vector to the one dimensional score. Fig. 1 illustrates the general principle of adding attention to a deep learning model.

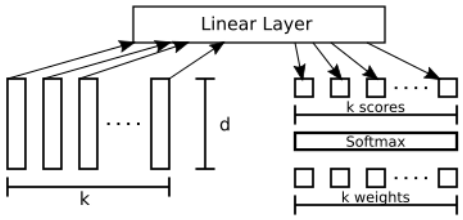

Figure 1. Illustration of adding attention (see text for description).

We construct our HAR models by adding an attention layer to the state of the art architecture from [9] – DeepConvLSTM (see below for model details). The general idea is to start with a large enough temporal context (sliding window) that was used in previous work and led to reasonable recognition performance. We then add an attention layer to automatically rescale the weights of all samples in the analysis frame according to their relevance for modeling, which is according to our hypothesis that not all historic samples in an analysis frame are of (the same) relevance for modeling. This relevance weighting of a sensor reading's history is a direct outcome of the attention layers, which we exploit for improving HAR models. We do not change any other (hyper-) parameters to focus our exploration on the effect that introducing attention has on state-of-the-art activity recognition.Fig. 2 illustrates this architecture. Details are given in what follows.

DeepConvLSTM and Attention

DeepConvLSTM models [9] represent the state-of-the-art for deep learning based HAR applications, which motivates us to explore the addition of attention layers to this model architecture. Model input are $24 \times d$ frames each representing 24 samples with d features. Frames are fed into four consecutive convolution layers with standard rectified linear units (Fig. 2). Input frames are obtained through a standard sliding window procedure (window length: 24; shift: 12). Convolution filters of size 5×1 in all convolution layers expand the $24 \times d$ input frame to a two-dimensional array of size $8 \times (d \times f)$, where d is the number of features and f (set to 64) denotes the number of filters in each convolution layer. This sequence of 8 resulting feature vectors is then modeled by a two-layer LSTM with 128 hidden units. The final hidden layer of the LSTM represents the embedding of the input frame, which is fed into a linear layer and a softmax to produce the prediction for an input frame. To incorporate the attention mechanism, we analyze the 8 hidden states of the LSTM that represent the embeddings for the different parts of an input frame. We consider the first 7 hidden states as the historical temporal context and learn 7 weights corresponding to these states:

$$\text{past context} = [h_1, h_2, h_3, ...h_7] \quad (1)$$

$$\text{current} = h_8 \quad (2)$$

$$\text{transformed context} = \tanh(W_1 \times \text{past context} + b_1) \quad (3)$$

$$\text{weights} = \text{softmax}(W_2 \times \text{transformed context} + b_2) \quad (4)$$

$$\text{final embedding} = \text{past context} \times \text{weights} + \text{current} \quad (5)$$

b_1 and b_2 denote the biases in the two linear layers, and W_1 and W_2 represent the 2D matrices in the linear layers. We initially apply a linear transformation accompanied by a tanh linearity transforming each of these seven vectors of size 128 into seven new vectors of size 128 (Eq. 3). Another linear transformation

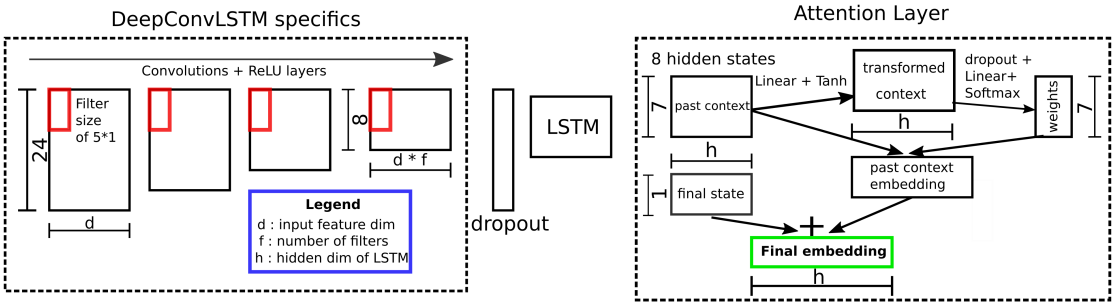

Figure 2. Adding attention to human activity recognition based on the DeepConvLSTM Architecture. The final embedding, highlighted in green, is used for prediction as opposed to the final hidden state in the original model (see text for description).

converts these 7 vectors each to size 1 essentially giving us scores for each of the hidden states. These scores are then passed through a softmax to give the final set of weights (Eq. 4). These weights are used to calculate a weighted sum of all the 7 hidden states to give the final embedding for the past context. This past context is added to the last hidden state to give the final embedding for the input frame (Eq. 5). This final embedding is used for classification as opposed to the last hidden state used by DeepConvLSTM.

Note that the addition of the last hidden state to the embedding of the past context can be interpreted as a skip connection from the recurrent layers to the attention layer. Considering the computational graph that corresponds to this model, we observe that the model may decide to propagate the gradient only to the recurrent layers and could avoid the attention layers altogether. This is actually beneficial for HAR as datasets are often relatively small overfitting needs to be avoided, which could be realized explicitly through aggressive regularization, through the dropout layers [11] shown in Fig. 2 , or implicitly through these skip connections.

EXPERIMENTS

Our explorations of the benefits that attention models may bring to human activity recognition are based on experimental evaluations on standard datasets from the field: Opportunity [3], PAMAP2 [10], and Skoda [12]. These datasets are very diverse in terms of the nature of activities and the distribution of activities representing robust benchmarks for evaluating HAR systems. We employ standard training and evaluation protocols based on hold out datasets as they have been defined in the original publications (and summarized, e.g., in [4]).

We used a sliding window procedure to extract initial processing frames (length: 1s, overlap: 50%). The resulting frames are randomly shuffled before training to avoid bias and the label of each frame is set to be the label of the last sample in the window. During evaluation all studied models produce sample-wise predictions, which is–in contrast to frame-wise prediction–more realistic for practical applications [5, 4]. The sample-wise prediction is generated by considering a 24 sample window preceding, and containing, the sample. This window is then fed into the model to get the label for the sample. For the first 23 samples in the time series, a valid window does not exist and for these samples the majority class is predicted.

Number of Conv. Filters(f)	64
Optimization Algorithm	RMSProp [13]
Convolution Filter Size	5×1
Input Dimension(d)	Varies for each dataset
Initial Learning rate	1e-3
Hidden Units(h)	128
Hidden Layers	2
Batch Size	100

Table 1. Summary of hyper-parameter values.

Models were trained using cross-entropy loss. Learning rate was decayed every epoch and the decay rate, along with dropout, were optimized for all models and these parameters seemingly have substantial impact on recognition performance. All hyperparameters and their values as they were used for our explorations are listed in Table 4. All code along with the best model weights for each of the datasets and the best hyper-parameters is available on our github page for reference[1].

RESULTS

Given the imbalance in class-distributions for all three datasets, we report results as mean f1 scores. Statistical significance tests are based on Wilson score interval with 95% confidence. Recognition results for all benchmark datasets are given in Table 5 as mentioned in [4]. It can be seen that the incorporation of attention models leads to significant increase in performance over the state -of-the-art for both Opportunity and PAMAP2. For Skoda we only see marginal improvements when introducing attention, which is similar to what has been reported in the literature for (other) model evaluations on this datasets [4] [5] [9]. Note that we have not changed the model architecture across the three datasets to keep the evaluation consistent. However, changing hyper-parameters may have an impact on numerical values.

DISCUSSION

Figure 3 visualizes the weights of the best model. While evaluating, each sample has a set of 7 weights associated with the relative importance of the first 7 hidden states of the LSTM. We take the median of these weights across all samples belonging to a certain activity. The visualization shows interesting insights into the model's behavior. We notice that most of the weight is concentrated on the last few hidden

[1] https://bitbucket.org/vmurahari3/deepconvlstmattention

Modeling Variant	Datasets Opportunity	PAMAP2	Skoda
DeepConvLSTM [9]	67.2	74.8	91.2
b-LSTM-S [5]	68.4	83.8	92.1
Att. Model	**70.7**	**87.5**	91.3
Confidence Interval	± **.003**	± **.002**	± .004

Table 2. Sample-wise recognition results (class-averaged F1). Significant improvements over non-attention baselines are given in bold (Wilson).

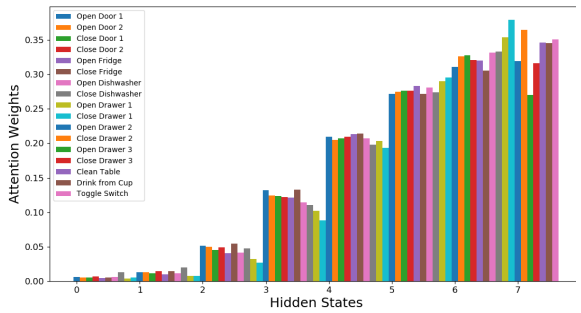

Figure 3. Visualizing weights learned by the best attention model on the Opportunity test dataset (best viewed in color).

states. This is reasonable as these states capture the summary of the input frame and through the LSTM recurrence the last hidden states capture more information than previous ones, and hence have a more dominant contribution to the final context embedding. However, only using the final hidden state–as LSTMs do–is detrimental as there may be some important information at the start of the input frame. Therefore, through using attention models we see a significant amount of the weight being placed on the past hidden states as well and this allows the model to capture the context more effectively as opposed to only relying on the last hidden state of the LSTM. The improvements in recognition performance confirm the benefit of adding attention to deep, recurrent HAR models.

We also observe that for all activities analyzed, the weight on the first two hidden states is close to zero. This is likely due to the first few hidden states not yet being able to capture anything valuable because the history at this stage is too short and thus rather uninformative. The attention model explicitly downweights those initial states to not "waste" model parameters if included. In summary, the attention mechanism effectively shrinks and focuses the history of a sensor reading that a HAR model needs to focus on.

We also observe that among all the (Opportunity) activities, the activity "Open Door 3" has the most spread out weights on all hidden states. This suggests that this activity might involve multiple distinct segments as the model distributes the weight evenly on hidden states. On further inspection, we realize that this activity is about opening the lowest drawer in a cupboard containing three drawers and hence one might need to perform multiple smaller activities such as bending down, opening the drawer and rising up to perform this activity. Therefore, the model is incentivized to distribute the weight more evenly to capture these sub-activities. This last aspect is the basis for future developments and applications as–essentially–it is the starting point for novel segmentation schemes.

REFERENCES

1. S. Bhattacharya and N.D. Lane. 2016. Sparsification and separation of deep learning layers for constrained resource inference on wearables. In *Proc. Int. Conf. Embedded Network Sensor Systems*.
2. A. Bulling, U. Blanke, and B. Schiele. 2014. A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors. *ACM Comput. Surv.* 46, 3, Article 33 (Jan. 2014), 33 pages.
3. R. Chavarriaga, H. Sagha, A. Calatroni, S.T. Digumarti, G. Tröster, J.R. Millán, and D. Roggen. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters* 34, 15 (2013), 2033–2042.
4. Y. Guan and T. Plötz. 2017. Ensembles of Deep LSTM Learners for Activity Recognition Using Wearables. *PACM IMWUT* 1, 2 (June 2017), 11:1–11:28.
5. N.Y. Hammerla, S. Halloran, and T. Ploetz. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *Proc. IJCAI*.
6. S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, and others. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. (2001).
7. S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
8. A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proc. ICML*.
9. F.J. Ordóñez and D. Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16, 1 (2016), 115.
10. A. Reiss and D. Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring. In *Proc. ISWC*.
11. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR* 15, 1 (2014), 1929–1958.
12. T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tröster. 2008. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing* 7, 2 (2008).
13. T. Tieleman and G. Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* 4, 2 (2012), 26–31.
14. J. Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, and Shonali Krishnaswamy. 2015. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition.. In *Proc. IJCAI*.
15. M. Zeng, L.T. Nguyen, B. Yu, Ole J. Mengshoel, J. Zhu, P. Wu, and J. Zhang. 2014. Convolutional neural networks for human activity recognition using mobile sensors. In *Proc. MobiCASE*.