In: Proceedings of the European Symposium on Artificial Neural Networks,
Computational Intelligence and Machine Learning (ESANN), pp. 213-218, Bruges, Belgium (2016)

Gesture Recognition with a Convolutional Long Short-Term Memory Recurrent Neural Network

Eleni Tsironi, Pablo Barros and Stefan Wermter *

University of Hamburg - Department of Computer Science
Vogt-Koelln-Strasse 30, D-22527 Hamburg - Germany
http://www.informatik.uni-hamburg.de/WTM/

Abstract. Inspired by the adequacy of convolutional neural networks in implicit extraction of visual features and the efficiency of Long Short-Term Memory Recurrent Neural Networks in dealing with long-range temporal dependencies, we propose a Convolutional Long Short-Term Memory Recurrent Neural Network (CNNLSTM) for the problem of dynamic gesture recognition. The model is able to successfully learn gestures varying in duration and complexity and proves to be a significant base for further development. Finally, the new gesture command TsironiGR-dataset for human-robot interaction is presented for the evaluation of CNNLSTM.

1 Introduction

Gestures constitute a crucial element in human communication, as well as in human-robot interaction, thus, gesture recognition has been a field of particular interest in computer science. More specifically, dynamic gesture recognition is a challenging task, since it requires the accurate detection of the body parts involved in the gesture, their tracking and the interpretation of their sequential movement. There have been many approaches proposed by the research community, differing in the sensor modalities (e.g. RGB cameras, depth sensors, wearable devices); in the process of segmenting the body parts involved in the gesture (e.g. skin colour segmentation by thresholding, motion analysis); in the representation of the segmented body parts (e.g. hand orientation); and in the recognition process of the gesture (e.g. Hidden Markov Models [1], Dynamic Time Warping [2], Recurrent Neural Networks [3], Echo State Networks [4]).

Motivated by the efficiency of Convolutional Neural Networks (CNNs) in implicit feature extraction and their successful application in form of Multichannel Convolutional Neural Networks (MCCNNs) in gesture recognition [5] and by the ability of Long Short-Term Memory recurrent neural networks (LSTMs) with forget gates and peephole connections [6] in modeling long-range dependencies of sequential data, this paper proposes a Convolutional Long Short-Term Memory recurrent neural network (CNNLSTM) for the task of dynamic gesture recognition. The proposed architecture aims to deal with the limitations caused by the use of feedforward networks in sequential tasks such as the requirement of using fixed-size windows and their lack of flexibility in learning sequences of

*This work was partially supported by the CAPES Brazilian Federal Agency for the Support and Evaluation of Graduate Education (p.n.5951-13-5).

In: Proceedings of the European Symposium on Artificial Neural Networks,
Computational Intelligence and Machine Learning (ESANN), pp. 213-218, Bruges, Belgium (2016)

different sizes. At the same time, we want to avoid the process of explicit feature extraction, by jointly training the CNN and the LSTM using the convolutional layers as a trainable feature detector. The system proposed below requires the input of an RGB camera and uses motion representation in order to extract the body parts involved in the gesture. Similar forms of combinations of CNNs and LSTMs have been successfully used in other sequential tasks such as activity recognition [7] and speech recognition [8], reinforcing the assumption that such a model can also yield a significant improvement in the field of dynamic gesture recognition. Finally, CNNLSTM is a type of an Elman recurrent neural network and consequently, can be trained with Back Propagation Through Time (BPTT). To evaluate our model, we create a dynamic gesture corpus with nine different Human-Robot Interaction commands. We show that our model outperforms the common Convolutional Neural Network and it is able to learn the temporal aspects of the gestures.

2 CNNLSTM

The proposed CNNLSTM architecture for dynamic gesture recognition consists of two consecutive convolutional layers, a flattening layer, a Long Short-Term Memory recurrent layer and a softmax output layer. At each time-step a differential image [9] Δ is given as input to the first convolutional layer. The differential image is the output of the segmentation process and represents the body motion. More precisely, the segmentation process that generates a differential image is a three-frame differencing operation, followed by a bitwise AND operation, as described in equation 1.

$$\Delta = (I_t - I_{t-1}) \wedge (I_{t+1} - I_t), \tag{1}$$

where I_t, I_{t-1}, I_{t+1} are the frames at the current time-step t, the previous time-step $t-1$ and the next time-step $t+1$ respectively, and $\wedge$ is the *bitwise* AND operation.

Once the differential image given as input to the CNNLSTM is processed by the first convolutional layer, a set of feature maps is produced, which is further processed by the second convolutional layer. Thereupon, the feature maps produced by the second convolutional layer are flattened to form the input for the hidden layer, which in Figure 1, for better visualisation, is depicted only by two units. The vector is fed to the LSTM blocks of the recurrent layer, which makes use of the past context. The output of the recurrent layer is squashed by a softmax activation function, which assigns a gesture label to the current differential image. CNNLSTM is a deep recurrent architecture that can be trained with standard Backpropagation Through Time.

A convolutional layer consists of two consecutive convolution and max-pooling operations coupled by means of a squashing function as shown in the equation

In: Proceedings of the European Symposium on Artificial Neural Networks,
Computational Intelligence and Machine Learning (ESANN), pp. 213-218, Bruges, Belgium (2016)

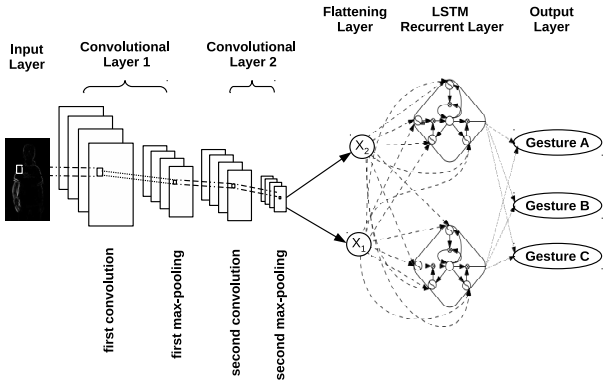

Fig. 1: The architecture of the proposed CNNLSTM.

below:

$$x_j^l = tanh(pooling_{max}(x_j^{l-1} * k_{ij}) + b_j^l), \tag{2}$$

where x_j^l are the feature maps produced by the convolutional layer l, x_j^{l-1} are the feature maps of the previous convolutional layer $l-1$, k_{ij} are the trained convolution kernels and b_j^l the additive bias. Finally, $pooling_{max}(\cdot)$ is the max-pooling operation and $tanh(\cdot)$ is the hyperbolic activation function.

An LSTM block of the recurrent layer is defined by the following set of equations.

$$\begin{aligned} i_t &= \sigma(x^t W_{xi} + h_{t-1} W_{hi} + c_{t-1} W_{ci} + b_i), \\ f_t &= \sigma(x^t W_{xf} + h_{t-1} W_{hf} + c_{t-1} W_{cf} + b_f), \\ o_t &= \sigma(x^t W_{xo} + h_{t-1} W_{ho} + c_t W_{co} + b_o), \\ c_t &= f_t \circ c_{t-1} + i_t \circ tanh(x^t W_{xc} + h_{t-1} W_{hc} + b_c), \\ h_t &= o_t \circ tanh(c_t), \end{aligned} \tag{3}$$

where x^t is the input to the LSTM block, i_t, f_t, o_t, c_t, h_t are the input gate, the forget gate, the output gate, the cell state and the output of the LSTM block respectively at the current time step t. W_{xi}, W_{xf}, W_{xo} are the weights between the input layer and the input gate, the forget gate and the output gate respectively. W_{hf}, W_{hi}, W_{ho} are the weights between the hidden recurrent layer and the forget gate, the input gate and the output gate of the memory

In: Proceedings of the European Symposium on Artificial Neural Networks,
Computational Intelligence and Machine Learning (ESANN), pp. 213-218, Bruges, Belgium (2016)

block respectively. W_{ci}, W_{cf}, W_{co} are the weights between the cell state and the input gate, the forget gate and the output gate respectively and finally, b_i, b_f, b_o are the additive biases of the input gate, the forget gate and the output gate respectively. The set of activation functions consists of the sigmoid function $\sigma(\cdot)$, the element-wise multiplication $\circ(\cdot)$ and the hyperbolic activation function $tanh(\cdot)$.

3 Experiments & Results

For the evaluation of the proposed CNNLSTM architecture the new gesture command TsironiGR-dataset for human-robot interaction was created. The dataset includes nine gesture classes; "abort", "circle", "hello", "no", "stop", "warn", "turn left", "turn" and "turn right", as shown in Figure 2. For the collection of the dataset, six subjects were recorded and each of them performed each gesture approximately ten times in a random order. Each of the gesture sequences is segmented and labeled with their correct gesture class label. The dataset consists of 543 gesture sequences in total. The gestures were captured by an RGB camera with a resolution of 640x480, recorded with a frame rate of 30 FPS. For the experiments the dataset was split in 60% for training, 20% for validation and 20% for testing. Each experiment was repeated five times.

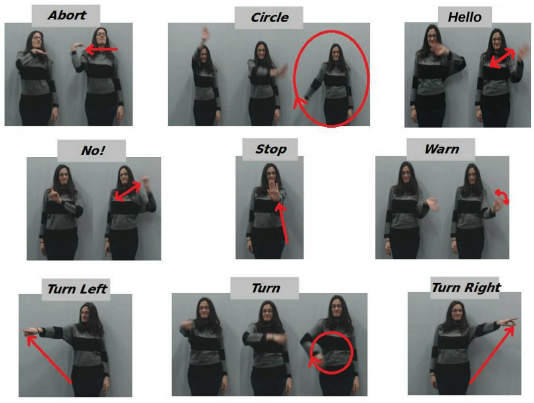

Fig. 2: The motion pattern of each of the training gesture commands

The performance of the CNNLSTM was compared with a common CNN baseline system. The CNN architecture consists of two consecutive convolutional layers, connected to a fully-connected hidden layer, which consequently is connected to a softmax output layer. The input to the network is a motion history

In: Proceedings of the European Symposium on Artificial Neural Networks,
Computational Intelligence and Machine Learning (ESANN), pp. 213-218, Bruges, Belgium (2016)

image, which consists of the accumulation of all the subsequent differential images of a gesture sequence. The Loss function used for the training the CNN is a negative log likelihood function and is trained with Backpropagation.

For the training of the CNN baseline model and the CNNLSTM model presented in the previous section, we randomly initialised all the model weights, except for the biases which were initialised with zeros. At the beginning of each epoch, the order of the training dataset was randomised. The backward pass of the CNNLSTM, and therefore the weight update, was done only after a whole sequence had been propagated forward through the network. The error signals and therefore the weight updates, have been calculated with respect to the mean of cross entropy loss function. During the testing phase, each gesture sequence has been classified separately. More specifically, the classifications of each of the differential images belonging to the sequence were processed to compute the gesture label with the highest frequency and assign it as a label to the whole gesture sequence.

The input size for the differential and motion history images was resized to 64x48, the convolutional layers had the same parameters in both architectures with the size of the first layer kernels being 11x11, the size of the max-pooling window 2x2 and the number of feature maps 5. The size of the convolution kernel of the second convolutional layer was 6x6, the size of the second max-pooling window 2x2 and the number of the produced feature maps 10. The difference between the two architectures concerns the type of the fully-connected layers following the flattening of the output of the second convolutional layer. The hidden fully-connected layer of the CNN is a simple feedforward hidden layer and that of the CNNLSTM is a hidden recurrent LSTM layer. The number of the hidden neurons of the fully-connected layer of the CNN was 500, the same as the number of the LSTM blocks in the recurrent fully-connected layer of the CNNLSTM model. The proposed CNNLSTM outperformed the simple CNN in terms of accuracy, precision, recall and F1-measure as shown in Table 1. We notice as well, that in the five times each model was run, the CNNLSTM seems to be more consistent, with smaller turbulences, as revealed by the standard deviation. The CNNLSTM converged quite fast in all runs, around the 19^{th} epoch. Moreover, the label frame patterns per sequence are very consistent, with the majority of the frames either being all correctly classified or just the first few frames being misclassified, which can be explained by the fact that the first frames in most gestures may be confusing since all gestures in the dataset were performed starting from the same resting position.

Model	Accuracy	Precision	Recall	F1-measure
CNN	77.78%±3.75%	79.87%±3.64%	77.78%±4.19%	76.56% ±4.27%
CNNLSTM	91.67%±1.13%	92.25% ±1.02%	91.67%±1.13%	91.63% ±1.15%

Table 1: Metrics table for the CNN and CNNLSTM models

In: Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pp. 213-218, Bruges, Belgium (2016)

4 Conclusion

We presented a Convolutional Long Short-Term Memory Recurrent Neural Network (CNNLSTM) architecture for the task of gesture recognition. The model was evaluated on the new gesture command TsironiGR-dataset, and outperformed the common CNN baseline. CNNLSTM extends the simple CNN by modelling the temporal evolution of the body postures while the gesture is a performed, which is a crucial element for gesture recognition. Therefore, the proposed model exhibited very good performance in sequence level classification and could be further improved by training the CNNLSTM with a Connectionist Temporal Classification (CTC) loss function [10]. With CTC we can eliminate the need for training the model on temporally pre-segmented gestures, and at the same time get a classification label immediately after a whole sequence is recognised, overcoming in this way the limitations of frame level classification in the task of gesture recognition. Moreover, the system can be also easily extended to accept three-dimensional input such as depth information.

References

[1] Gerhard Rigoll, Andreas Kosmala, and Stefan Eickeler. High performance real-time gesture recognition using hidden markov models. In *Gesture and Sign Language in Human-Computer Interaction*, pages 69–80. Springer, 1998.

[2] Trevor Darrell and Alex Pentland. Space-time gestures. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 335–340. IEEE, 1993.

[3] Natalia Neverova, Christian Wolf, Giacomo Paci, Giacomo Sommavilla, Graham W Taylor, and Florian Nebout. A multi-scale approach to gesture detection and recognition. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 484–491. IEEE, 2013.

[4] Doreen Jirak, Pablo Barros, and Stefan Wermter. Dynamic gesture recognition using echo state networks. pages 475–480, 2015.

[5] Pablo Barros, German I. Parisi, Doreen Jirak, and Stephan Wermter. Real-time gesture recognition using a humanoid robot with a deep neural architecture. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 646–651, Nov 2014.

[6] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. Learning precise timing with lstm recurrent networks. *The Journal of Machine Learning Research*, 3:115–143, 2003.

[7] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. 2014.

[8] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Hasim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *in Proceedings ICASSP*, 2015.

[9] Robert T Collins, Alan Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt, et al. *A system for video surveillance and monitoring*, volume 2. Carnegie Mellon University, the Robotics Institute Pittsburg, 2000.

[10] Alex Graves et al. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer, 2012.