Co-occurrence Feature Learning for Skeleton based Action Recognition using Regularized Deep LSTM Networks

Wentao Zhu [1*], **Cuiling Lan** [2], **Junliang Xing** [3], **Wenjun Zeng** [2], **Yanghao Li** [4], **Li Shen** [5], **Xiaohui Xie** [1]
[1] University of California, Irvine, USA [2] Microsoft Research Asia, Beijing, China
[3] Institute of Automation, Chinese Academy of Sciences, Beijing, China
[4] Peking University, Beijing, China [5] University of Chinese Academy of Sciences, Beijing, China
wentaoz1@uci.edu, {culan,wezeng}@microsoft.com, jlxing@nlpr.ia.ac.cn,
lyttonhao@pku.edu.cn, li.shen@vipl.ict.ac.cn, xhx@ics.uci.edu

Abstract

Skeleton based action recognition distinguishes human actions using the trajectories of skeleton joints, which provide a very good representation for describing actions. Considering that recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM) can learn feature representations and model long-term temporal dependencies automatically, we propose an end-to-end fully connected deep LSTM network for skeleton based action recognition. Inspired by the observation that the co-occurrences of the joints intrinsically characterize human actions, we take the skeleton as the input at each time slot and introduce a novel regularization scheme to learn the co-occurrence features of skeleton joints. To train the deep LSTM network effectively, we propose a new dropout algorithm which simultaneously operates on the gates, cells, and output responses of the LSTM neurons. Experimental results on three human action recognition datasets consistently demonstrate the effectiveness of the proposed model.

1 Introduction

Recognizing human actions has remained one of the most important and challenging tasks in computer vision. It facilitates a wide range of applications such as intelligent video surveillance, human-computer interaction, and video understanding (Poppe 2010; Weinland, Ronfard, and Boyerc 2011).

Traditional studies on action recognition mainly focus on recognizing actions from RGB videos recorded by 2D cameras (Weinland, Ronfard, and Boyerc 2011). However, capturing human actions in the full 3D space in which they actually occur can provide more comprehensive information. Biological observations suggest that humans can recognize actions from just the motion of a few light displays attached to the human body (Johansson 1973). Motion capture systems (CMU 2003) extract 3D joint positions using markers and high precision camera arrays. Although slightly higher in price, such systems provide highly accurate joint positions for skeletons. Recently, the Kinect device has gained much popularity thanks to its excellent accuracy in human body modeling and affordable price. The bundled SDK for Kinect v2 can directly generate accurate skeletons in real-time. Due to the prevalence of these devices, skeleton based representations of the human body and its temporal evolution has become an attractive option for action recognition.

In this paper, we focus on the problem of skeleton based action recognition. The key to this problem lies mainly in two aspects. One is to design robust and discriminative features from the skeleton (and the corresponding RGB-D images) for intra-frame content representation (Müller, Röder, and Clausen 2005; Wang, Liu, and Junsong 2012; Sung et al. 2012; Yang and Tian 2014; Ji, Ye, and Cheng 2014). The other is to explore temporal dependencies of the inter-frame content for action dynamics modeling, using hierarchical maximum entropy Markov model (Sung et al. 2011), hidden Markov model (Xia, Chen, and Aggarwal 2012) or Conditional Random Fields (Sminchisescu et al. 2005). Inspired by the success of deep recurrent neural networks (RNNs) using the Long Short-Term Memory (LSTM) architecture for speech feature learning and time series modeling (Graves, Mohamed, and Hinton 2013; Graves and Schmidhuber 2005), we intend to build an effective action recognition model based on deep LSTM network.

To this end, we propose an end-to-end fully connected deep LSTM network to perform automatic feature learning and motion modeling (Fig. 1). The proposed network is constructed by inheriting many insights from recent successful networks (Graves 2012; Krizhevsky, Sutskever, and Hinton 2012; Szegedy et al. 2015; Du, Wang, and Wang 2015) and is designed to robustly model complex relationships among different joints. The LSTM layers and feedforward layers are alternately deployed to construct a deep network to capture the motion information. To ensure the model learns effective features and motion dynamics, we enforce different types of strong regularization in different parts of the model, which effectively mitigates over-fitting.

Specifically, two types of regularizations are proposed. (i) For the fully connected layers, we introduce regularization to drive the model to learn co-occurrence features of the joints at different layers. (ii) For the LSTM neurons, we derive a new dropout and apply it to the LSTM neurons in the last LSTM layer, which helps the network to learn complex motion dynamics. With these forms of regularization, we

[*] This work was done when W. Zhu was an intern at Microsoft Research Asia.

validate our deep LSTM networks on three public datasets for human action recognition. The proposed model has been shown to consistently outperform other state-of-the-art algorithms for skeleton based human action recognition.

2 Related Work

2.1 Activity Recognition with Neural Networks

In contrast to the handcrafted features, there is a growing trend of learning robust feature representations from raw data with deep neural networks, and excellent performance has been reported in image classification (Krizhevsky, Sutskever, and Hinton 2012) and speech recognition (Graves, Mohamed, and Hinton 2013). However, there are only few works which leverage neural networks for skeleton based action recognition. A multi-layer perceptron network is trained to classify each frame (Cho and Chen 2014); however, such a network cannot explore temporal dependencies very well. In contrast, a gesture recognition system (Lefebvre et al. 2013) employs a shallow bidirectional LSTM with only one forward hidden layer and one backward hidden layer to explore long-range temporal dependencies. A deep recurrent neural network architecture with handcrafted subnets is utilized for skeleton based action recognition (Du, Wang, and Wang 2015). However, the handcrafted hierarchical subnets and their fusion ignore the inherent co-occurrences of joints. This motivates us to design a deep fully connected neural network which is capable of fully exploiting the inherent correlations among skeleton joints in various actions.

2.2 Co-occurrence Exploration

An action is usually only associated with and characterized by the interactions and combinations of a subset of the skeleton joints. For example, the joints "hand", "arm" and "head" are associated with the action "making telephone call". An actionlet ensemble model exploits this trait by mining some particular conjunctions of the features corresponded to some subsets of the joints (Wang, Liu, and Junsong 2012). Similarly, actions involving two people can be characterized by the interactions of a subset of the two persons' joints (Yun et al. 2012; Ji, Ye, and Cheng 2014). Inspired by the actionlet ensemble model, we introduce a new exploration mechanism in the deep LSTM architecture to achieve automatic co-occurrence mining as opposed to pre-specifying in advance which joints should be grouped.

2.3 Dropout for Recurrent Neural Networks

Dropout has been demonstrated to be quite effective in deep convolutional neural networks (Krizhevsky, Sutskever, and Hinton 2012), but there has been relatively little research on applying it to RNNs. In order to preserve the ability of RNNs to model sequences, dropout applied only to the feedforward (along layers) connections but not to the recurrent (along time) connections is proposed (Pham et al. 2014). This is to avoid erasing all the information from the units (due to dropout). Note that the previous work only considers dropout at the output response for an LSTM neuron (Zaremba, Sutskever, and Vinyals 2014). However, considering that an LSTM neuron consists of internal cell and gate units, we believe one should not only look at the output of the neuron but also into its internal structure to design effective dropout schemes. In this paper, we design an in-depth dropout for LSTM to address this problem.

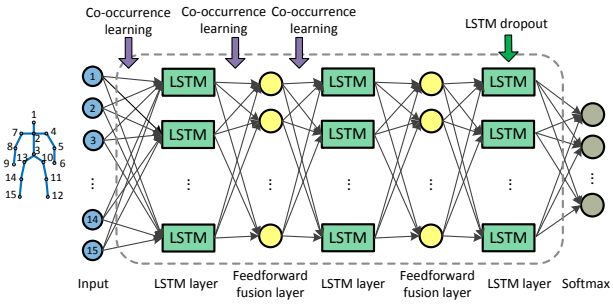


Figure 1: The proposed deep LSTM network with three LSTM layers and two feedforward layers. For clarity, the temporal recurrent structure is not shown.

3 Deep LSTM with Co-occurrence Exploration and In-depth Dropout

Leveraging the insights from recent successful networks, we design a fully connected deep LSTM network for skeleton based action recognition. Fig. 1 shows the architecture of the proposed network, which has three bidirectional LSTM layers, two feedforward layers, and a softmax layer that gives the predictions. The proposed full connection architecture enables one to fully exploit the inherent correlations among skeleton joints. In the network, the co-occurrence exploration is applied to the connections prior to the second LSTM layer to learn the co-occurrences of joints/features. LSTM dropout is applied to the last LSTM layer to enable more effective learning. Note that each LSTM layer uses bidirectional LSTM and we do not explicitly distinguish the forward and backward LSTM neurons in Fig. 1. At each time step, the input to the network is a vector denoting the 3D positions of the skeleton joints in a frame.

In the following, we first review LSTM briefly to make the paper self-contained. Then we introduce our method for co-occurrence exploration in the deep LSTM network. Lastly we describe our dropout algorithm which is designed for the LSTM neurons and enables effective learning of the model.

3.1 Overview of LSTM

The RNN is a successful model for sequential learning (Graves 2012). For the recurrent neurons at some layer, the output responses $\mathbf{h}_t$ are calculated based on the inputs $\mathbf{x}_t$ to this layer and the responses $\mathbf{h}_{t-1}$ from the previous time slot

$$\mathbf{h}_t = \theta\left(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h\right), \quad (1)$$

where $\theta(\cdot)$ denotes the activation function, $\mathbf{b}_h$ denotes the bias vector, $\mathbf{W}_{xh}$ is the matrix of weights between the input and hidden layer and $\mathbf{W}_{hh}$ is the matrix of recurrent weights from the hidden layer to itself at adjacent time steps which is used for exploring temporal dependency.

LSTM is an advanced RNN architecture which can learn long-range dependencies (Graves, Mohamed, and Hinton

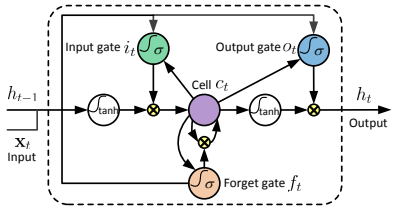


Figure 2: The structure of an LSTM neuron.

2013). Fig. 2 shows a typical LSTM neuron, which contains an input gate i_t, a forget gate f_t, a cell c_t, an output gate o_t and an output response h_t. The input gate and forget gate govern the information flow into and out of the cell. The output gate controls how much information from the cell is passed to the output h_t. The memory cell has a self-connected recurrent edge of weight 1, ensuring that the gradient can pass across many time steps without vanishing or exploding. Therefore, it overcomes the difficulties in training the RNN model caused by the "vanishing gradient" effect. For all the LSTM neurons in some layer, at time t, the recursive computation of activations of the units is

$$\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i\right), \\
\mathbf{f}_t &= \sigma\left(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f\right), \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c), \\
\mathbf{o}_t &= \sigma\left(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o\right), \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right),
\end{aligned} \quad (2)$$

where $\odot$ denotes element-wise product, $\sigma(x)$ is the sigmoid function defined as $\sigma(x) = 1/(1+e^{-x})$, $\mathbf{W}_{\alpha\beta}$ is the weight matrix between α and β (e.g., $\mathbf{W}_{xi}$ is the weight matrix from the inputs $\mathbf{x}_t$ to the input gates $\mathbf{i}_t$), and $\mathbf{b}_\beta$ denotes the bias term of β with $\beta \in \{i, f, c, o\}$. Four weight matrixes are associated with input $\mathbf{x}_t$. To allow the information from both the future and the past to determine the output, bidirectional LSTM can be utilized (Graves 2012).

3.2 Co-occurrence Exploration

The fully connected deep LSTM network in Fig. 1 has very powerful learning capability. However, it is difficult to learn directly due to the huge parameter space. To overcome this problem, we introduce a co-occurrence exploration process to ensure the deep model learns effective features.

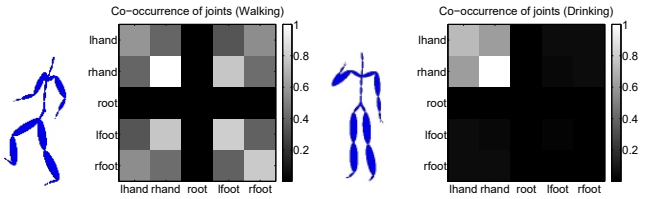


Figure 3: Illustration of co-occurrences of joints for "walking" and "drinking" respectively (using the absolute values of the covariance matrix). Joints from different parts are active simultaneously, e.g., joints of hands and feet for "walking". Different actions have different active joint sets.

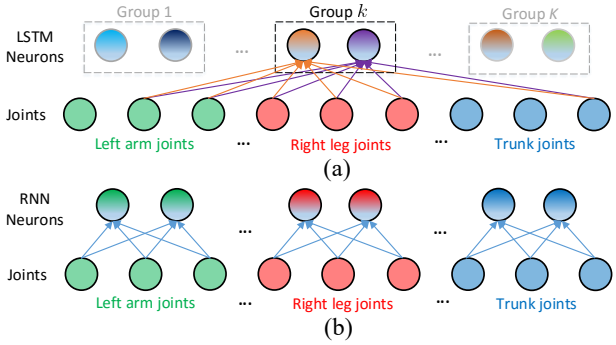
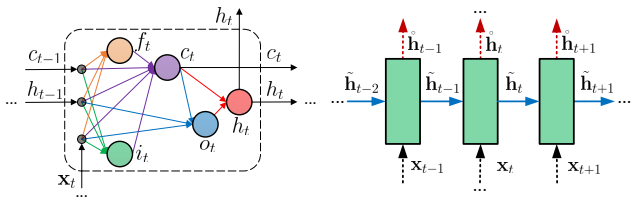


Figure 4: Illustration of the connections between joints and neurons in the first layer. (a) Co-occurrence connections automatically learned for Group k (proposed). (b) Part-based subnet connections (Du, Wang, and Wang 2015), where the co-occurrences of joints across different parts are prohibited.

The co-occurrence of some joints can intrinsically characterize a human action. Fig. 3 shows two examples. For "walking", the joints from hands and feet have high correlations but they all have low correlations with the joint of root. The sets of correlated joints for "walking" and "drinking" are very different, indicating the discriminative subset of joints varies for different types of actions. Two main aspects have been considered in our design of the network and the specialized regularization we propose. (i) We expect the network can automatically explore the conjunctions of discriminative joints. (ii) We expect the network can explore different co-occurrences for different types of actions. Therefore, we design the fully connected network to allow each neuron being connected to any joints (for the first layer) or responses of the previous layer (for the second or higher layer) to automatically explore the co-occurrences. Note that the output responses are also referred to as *features* which are the input of the next layer. We divide the neurons in the same layer into K groups to allow different groups to focus on exploration of different conjunctions of discriminative joints. Taking the k^{th} group of neurons as an example (see Fig. 4 (a)), the neurons will automatically connect the discriminative joints/features. In our design, we incorporate the co-occurrence regularization into the loss function

$$\min_{\mathbf{W}_{x\beta}} \mathcal{L} + \lambda_1 \sum_{\beta \in S} \|\mathbf{W}_{x\beta}\|_1 + \lambda_2 \sum_{\beta \in S} \sum_{k=1}^{K} \left\| {\mathbf{W}_{x\beta,k}}^T \right\|_{2,1}, \quad (3)$$

where $\mathcal{L}$ is the maximum likelihood loss function of the deep LSTM network (Graves 2012). The other two terms are used for the co-occurrence regularization which can be applied to each layer. $\mathbf{W}_{x\beta} = [\mathbf{W}_{x\beta,1}; \cdots; \mathbf{W}_{x\beta,K}] \in \mathbb{R}^{N \times J}$ is the connection weight matrix from inputs to the units associated with $\beta \in S$, with N indicating the number of neurons and J the dimension of inputs (e.g., for the first layer, J is the number of joints multiplied by 3). The N neurons are partitioned into K groups and the number of neurons in a group is $L = \lceil N/K \rceil$, with $\lceil \cdot \rceil$ representing the rounding up operation. $\mathbf{W}_{x\beta,k}$ is the matrix composed of the $(L(k-1)+1)^{th}$ to



(a) (b)

Figure 5: LSTM dropout. (a) An LSTM neuron viewed in unfolded form. Gates, cell and output response (as marked by large circles) can be dropped. (b) Dropout flow. The solid arrows denote the flow where dropout is forbidden and the dashed arrows denote the flow where dropout is allowed. A rectangle box indicates all the LSTM neurons in this layer.

$(Lk)^{th}$ rows of $\mathbf{W}_{x\beta}$. ${\mathbf{W}_{x\beta,k}}^T$ denotes its transpose. S denotes the set of internal units which are directly connected to the input of a neuron. For the LSTM layer, $S=\{i,f,c,o\}$ denotes the gates and cell in LSTM neurons. For the feedforward layer, $S=\{h\}$ denotes the neuron itself.

In the third term, for each group of units, a structural ℓ_{21} norm, which is defined as $\|\mathbf{W}\|_{2,1}=\sum_i\sqrt{\sum_j w_{i,j}^2}$ (Cotter et al. 2005), is used to drive the units to select a conjunction of descriptive inputs (joints/features) since the ℓ_{21} norm can encourage the matrix $\mathbf{W}_{x\beta,k}$ to be column sparse. Different groups explore different connection (co-occurrence) patterns in order to acquire the capability of recognizing multiple categories of actions. The ℓ_1 norm constraint (the second term) helps to learn discriminative joints/features.

The stochastic gradient descent method is then employed to solve (3). The advantage of the co-occurrence learning is that the model can automatically learn the discriminative joint/feature connections, avoiding the fixed a priori blocking of joint co-occurrences across human parts (Du, Wang, and Wang 2015) as illustrated in Fig. 4 (b).

3.3 In-depth Dropout for LSTM

Dropout tries to combine the predictions of many "thinned" networks to boost the performance. During training, the network randomly drops some neurons to force the remaining sub-network to compensate. During testing, the network uses all the neurons together to make predictions.

To extend this idea to LSTM, we propose a new dropout algorithm to allow the dropping of the internal gates, cell and output response for an LSTM neuron, encouraging each unit to learn better parameters. For clarity, an LSTM neuron is shown in Fig. 5 (a) in the unfolded form, where the units are explicitly connected. For recurrent neural networks, the erasing of all the information from a unit is not expected, especially when the unit remembers events that occurred many timesteps back in the past (Pham et al. 2014). Therefore, we allow the influence of dropout in LSTM to flow along layers (marked by dashed arrows) but prohibit it to flow along the time axis (marked by solid arrows) as illustrated in Fig. 5 (b). To control the influence flows, in the feedforward process, the network calculates and records two types of activations as follows. The responses of units to be transmitted along the time without dropout are

$$\begin{aligned}
\tilde{\mathbf{i}}_t &= \sigma\Big(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\tilde{\mathbf{h}}_{t-1} + \mathbf{W}_{ci}\tilde{\mathbf{c}}_{t-1} + \mathbf{b}_i\Big)\,,\\
\tilde{\mathbf{f}}_t &= \sigma\Big(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\tilde{\mathbf{h}}_{t-1} + \mathbf{W}_{cf}\tilde{\mathbf{c}}_{t-1} + \mathbf{b}_f\Big)\,,\\
\tilde{\mathbf{c}}_t &= \tilde{\mathbf{f}}_t \odot \tilde{\mathbf{c}}_{t-1} + \tilde{\mathbf{i}}_t \odot \tanh\Big(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\tilde{\mathbf{h}}_{t-1} + \mathbf{b}_c\Big)\,,\\
\tilde{\mathbf{o}}_t &= \sigma\Big(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\tilde{\mathbf{h}}_{t-1} + \mathbf{W}_{co}\tilde{\mathbf{c}}_t + \mathbf{b}_o\Big)\,,\\
\tilde{\mathbf{h}}_t &= \tilde{\mathbf{o}}_t \odot \tanh\left(\tilde{\mathbf{c}}_t\right).
\end{aligned} \tag{4}$$

The responses of units to be transmitted across layers with dropout applied are

$$\begin{aligned}
\mathring{\mathbf{i}}_t &= \sigma\left(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\tilde{\mathbf{h}}_{t-1} + \mathbf{W}_{ci}\tilde{\mathbf{c}}_{t-1} + \mathbf{b}_i\right)\odot\mathbf{m}_i,\\
\mathring{\mathbf{f}}_t &= \sigma\left(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\tilde{\mathbf{h}}_{t-1} + \mathbf{W}_{cf}\tilde{\mathbf{c}}_{t-1} + \mathbf{b}_f\right)\odot\mathbf{m}_f,\\
\mathring{\mathbf{c}}_t &= \Big(\mathring{\mathbf{f}}_t\odot\tilde{\mathbf{c}}_{t-1} + \mathring{\mathbf{i}}_t\odot\tanh\Big(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\tilde{\mathbf{h}}_{t-1} + \mathbf{b}_c\Big)\Big)\odot\mathbf{m}_c,\\
\mathring{\mathbf{o}}_t &= \sigma\left(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\tilde{\mathbf{h}}_{t-1} + \mathbf{W}_{co}\mathring{\mathbf{c}}_t + \mathbf{b}_o\right)\odot\mathbf{m}_o,\\
\mathring{\mathbf{h}}_t &= \mathring{\mathbf{o}}_t \odot \tanh\left(\mathring{\mathbf{c}}_t\right)\odot\mathbf{m}_h,
\end{aligned} \tag{5}$$

where $\mathbf{m}_i, \mathbf{m}_f, \mathbf{m}_c, \mathbf{m}_o$, and $\mathbf{m}_h$ are dropout binary mask vectors for input gates, forget gates, cells, output gates and output responses, respectively, with an element value of 0 indicating that dropout happens. Note that for the first LSTM layer, the inputs $\mathbf{x}_t$ are the skeleton joints of a frame; for the higher LSTM layer, the inputs $\mathbf{x}_t$ are the response outputs of the previous layer.

During the training process, the errors back-propagated to the output responses $\mathbf{h}_t$ are

$$\begin{aligned}
&\boldsymbol{\epsilon}_h^t = \mathring{\boldsymbol{\epsilon}}_h^t + \tilde{\boldsymbol{\epsilon}}_h^t,\\
&\mathring{\boldsymbol{\epsilon}}_h^t = \boldsymbol{\epsilon}_h^{hier}\odot\mathbf{m}_h,\quad \tilde{\boldsymbol{\epsilon}}_h^t = \boldsymbol{\epsilon}_h^{recur},
\end{aligned} \tag{6}$$

where $\boldsymbol{\epsilon}_h^{hier}$ denotes the vector of errors back-propagated from the upper layer at the same time slot, $\boldsymbol{\epsilon}_h^{recur}$ denotes the vector of errors from the next time slot in the same layer, $\mathring{\boldsymbol{\epsilon}}^t$ and $\tilde{\boldsymbol{\epsilon}}^t$ denote the dropout errors from the upper layer and recurrent errors from the next time slot, respectively.

By taking derivative of $\mathring{\mathbf{h}}_t$ with respect to $\mathring{\mathbf{o}}_t$ based on (5), we get the errors from $\mathring{\mathbf{h}}_t$ to $\mathring{\mathbf{o}}_t$ which represent the errors from upper layer with dropout involved

$$\mathring{\boldsymbol{\epsilon}}_o^t = (\mathring{\boldsymbol{\epsilon}}_h^t \odot \frac{\partial\mathring{\mathbf{h}}_t}{\partial\mathring{\mathbf{o}}_t})\odot\mathbf{m}_o = \mathring{\boldsymbol{\epsilon}}_h^t\odot\tanh\left(\mathring{\mathbf{c}}_t\right)\odot\mathbf{m}_o. \tag{7}$$

Similarly, based on (4), we get the errors back-propagated from $\tilde{\mathbf{h}}_t$ to $\tilde{\mathbf{o}}_t$ which represent the errors from the next time slot in the same layer without dropout

$$\tilde{\boldsymbol{\epsilon}}_o^t = \tilde{\boldsymbol{\epsilon}}_h^t \odot \frac{\partial\tilde{\mathbf{h}}_t}{\partial\tilde{\mathbf{o}}_t} = \tilde{\boldsymbol{\epsilon}}_h^t\odot\tanh\left(\tilde{\mathbf{c}}_t\right). \tag{8}$$

Then, the errors back-propagated to the output gates are the summation of the two types of errors

$$\boldsymbol{\epsilon}_o^t = \mathring{\boldsymbol{\epsilon}}_o^t + \tilde{\boldsymbol{\epsilon}}_o^t. \tag{9}$$

In the same way, we derive the errors propagated to the cells, forget gates, and input gates, based on (4) and (5).

During the testing process, we use all the neurons but multiplying the units of LSTM neurons (in the LSTM layer where dropout is applied) by the probability values of $1-p$, where p is the dropout probability of that unit. Note that the simple dropout which only drops the output responses $\mathbf{h}_t$ (Zaremba, Sutskever, and Vinyals 2014) is a special case of our proposed dropout.

3.4 Action Recognition using the Learned Model

With the learned deep LSTM network, the probability that a sequence $\mathbf{X}$ belongs to the class C_k is

$$\begin{aligned} p(C_k|\mathbf{X}) &= \frac{e^{o_k}}{\sum_{i=1}^{C} e^{o_i}}, \quad k = 1, \cdots, C, \\ \mathbf{o} &= \sum_{t=1}^{T} \left(\mathbf{W}_{\overrightarrow{h}o} \overrightarrow{\mathbf{h}}_t + \mathbf{W}_{\overleftarrow{h}o} \overleftarrow{\mathbf{h}}_t + \mathbf{b}_o \right), \end{aligned} \tag{10}$$

where C denotes the number of classes, T represents the length of the test sequence, $\mathbf{o} = [o_1, o_2, \cdots, o_C]$, $\overrightarrow{\mathbf{h}}_t$ and $\overleftarrow{\mathbf{h}}_t$ denote the output responses of the last bidirectional LSTM layer. Then, the class with the highest probability is chosen as action class.

4 Experiments

We validate the proposed model on the SBU kinect interaction dataset (Yun et al. 2012), HDM05 dataset (Müller et al. 2007), and CMU dataset (CMU 2003) whose groundtruth was labeled by ourselves. We have also tested our model on the Berkeley MHAD action recognition dataset (Ofli et al. 2013) and achieved 100% accuracy. To investigate the impact of each component in our model, we conduct experiments under different configurations represented as follows:

- Deep LSTM is our basic scheme without regularizations;
- Deep LSTM + Co-occurrence is the scheme with our proposed co-occurrence regularization applied;
- Deep LSTM + Simple Dropout is the scheme with the dropout algorithm proposed by Zaremba et al. (Zaremba, Sutskever, and Vinyals 2014) applied to our basic scheme;
- Deep LSTM + In-depth Dropout is the scheme with our proposed in-depth dropout applied;
- Deep LSTM + Co-occurrence + In-depth Dropout is our final scheme with both co-occurrence regularization and in-depth dropout applied.

Down-sampling the skeleton sequences in temporal is performed to have the frame rate of 30fps on the HDM05 dataset and CMU dataset. To reduce the influence of noise in the captured skeleton data, we smooth each joint's position of the raw skeleton using the filter $(-3, 12, 17, 12, -3)/35$ in the temporal domain (Savitzky and Golay 1964; Du, Wang, and Wang 2015). The number of groups (K) in our model is set to 5, 10, and 10 for the first three layers experimentally. We set the dropout probability p to 0.2 for each unit in an LSTM neuron, which makes the overall dropout probability of an LSTM neuron approach 0.5 (this can be derived based on (5)). Note that when dropout is applied, the number of neurons in the corresponding layer is doubled as suggested by previous work (Srivastava et al. 2014). We set the parameters λ_1 and λ_2 in (3) experimentally.

4.1 SBU Kinect Interaction Dataset

The SBU kinect interaction dataset is a Kinect captured human activity recognition dataset depicting two person interaction, which contains 230 sequences of 8 classes (6,614 frames) with subject independent 5-fold cross validation. The smoothed positions of joints are used as the input of the deep LSTM network for recognition. The number of neurons is set to 100×2, 100, 110×2, 110, 200×2 for the first to fifth layers respectively, where 2 indicates bidirectional LSTM is used and thus the number of neurons is doubled.

We have compared our schemes with other skeleton based methods (Yun et al. 2012; Ji, Ye, and Cheng 2014; Du, Wang, and Wang 2015). Note that we add an additional layer to fuse the two subnets corresponding to the two persons when extending Hierarchical RNN scheme for use in the two person interaction scenario (Du, Wang, and Wang 2015). We summarize the results in terms of the average recognition accuracy (5-fold cross validation) in Table 1.

Table 1: Comparisons on SBU kinect interaction dataset.

Methods	Acc.(%)
Raw skeleton (Yun et al. 2012)	49.7
Joint feature (Yun et al. 2012)	80.3
Raw skeleton (Ji, Ye, and Cheng 2014)	79.4
Joint feature (Ji, Ye, and Cheng 2014)	86.9
Hierarchical RNN (Du, Wang, and Wang 2015)	80.35
Deep LSTM	86.03
Deep LSTM + Co-occurrence	89.44
Deep LSTM + Simple Dropout	89.70
Deep LSTM + In-depth Dropout	90.10
Deep LSTM+Co-occurrence+In-depth Dropout	**90.41**

Table 1 shows that our basic scheme of Deep LSTM achieves comparable performance to the method using handcrafted complex features (Ji, Ye, and Cheng 2014). The proposed schemes of Deep LSTM + Co-occurrence and Deep LSTM + In-depth Dropout can improve the recognition accuracy by 3.4% and 4.1% respectively over Deep LSTM, indicating that the co-occurrence exploration boosts the discrimination of features and the proposed LSTM dropout is capable of learning a more effective model. Deep LSTM + In-depth Dropout is superior to Deep LSTM + Simple Dropout. Note that the deep LSTM network achieves remarkable (5.6%) performance improvement in comparison with the hierarchical RNN network (Du, Wang, and Wang 2015). That is because allowing full connection of joints/features with neurons rather than imposing a priori subnet constraints facilitates the interaction among joints especially when the joints do not belong to the same part, or same person. Our scheme with combined regularizations achieves the best performance.

4.2 HDM05 Dataset

The HDM05 dataset contains 2,337 skeleton sequences performed by 5 actors (184,046 frames after down-sampling). For fair comparison, we use the same protocol (65 classes, 10-fold cross validation) as used by Cho and Chen (Cho and Chen 2014). The pre-processing is the same as that done in the hierarchical RNN scheme (Du, Wang, and Wang 2015) (centralize the joints' positions to human center for each frame and smooth the positions). The number of neurons is 100×2, 110, 120×2, 120, 200×2 for the five layers respectively. Table 2 shows the results in terms of average accuracy. Our basic deep LSTM achieves better results than the Multi-layer Perception model, which suggests that LSTM exhibits better motion modeling ability than the MLP. With the proposed co-occurrence learning and in-depth dropout regularization, our full model also performs better than the manually designed hierarchical RNN approach.

Table 2: Comparisons on HDM05 dataset.

Methods	Acc.(%)
Multi-layer Perceptron (Cho and Chen 2014)	95.59
Hierarchical RNN (Du, Wang, and Wang 2015)	96.92
Deep LSTM	96.80
Deep LSTM + Co-occurrence	97.03
Deep LSTM + Simple Dropout	97.21
Deep LSTM + In-depth Dropout	97.25
Deep LSTM+Co-occurrence+In-depth Dropout	**97.25**

4.3 CMU Dataset

We have categorized the CMU motion capture dataset into 45 classes for the purpose of skeleton based action recognition[1]. The categorized dataset contains 2,235 sequences (987,341 frames after down-sampling) and is the largest skeleton based human action dataset so far. This dataset is much more challenging because: (i) the lengths of sequences vary greatly; (ii) the within-class diversity is large, e.g., for "walking", different people walk at different speeds and along different paths; (iii) the dataset contains complex actions such as dance, doing yoga.

We have evaluated the performance on both the entire dataset (CMU) and a subset of the dataset (CMU subset). For this subset, we have chosen eight representative action categories containing 664 sequences (125,667 frames after down-sampling), with actions of *jump*, *walk back*, *run*, *sit*, *getup*, *pickup*, *basketball*, *cartwheel*. The same preprocessing as used for the HDM05 dataset is performed. The number of neurons is set to 100×2, 100, 120×2, 120, 100×2 for the five layers. Three-fold cross validation is conducted and the results in terms of average accuracy are shown in Table 3. We can see that the proposed model achieves significant performance improvement, indicating that it can better learn the discriminative features and model long-range temporal dynamics even for this challenging dataset.

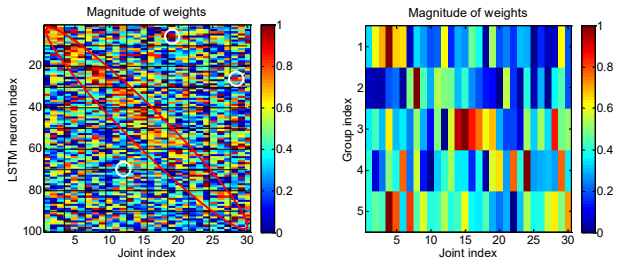
[1]http://www.escience.cn/people/zhuwentao/29634.html

Table 3: Accuracy (%) comparisons on CMU dataset.

Methods	CMU subset	CMU
Hierarchical RNN (Du, Wang, and Wang 2015)	83.13	75.02
Deep LSTM	86.00	79.53
Deep LSTM + Co-occurrence	87.20	79.60
Deep LSTM + Simple Dropout	87.80	80.59
Deep LSTM + In-depth Dropout	88.25	80.99
Deep LSTM+ Co-occurrence+In-depth Dropout	**88.40**	**81.04**

4.4 Discussions

To further understand our deep LSTM network, we visualize the weights learned in the first LSTM layer on the SBU kinect interaction dataset in Fig. 6 (a). Each element represents the absolute value of the weight between the corresponding skeleton joint and input gate of that LSTM neuron. It is observed that the weights in the diagonal positions marked by the red ellipse have high values, which means the co-occurrence regularization helps learn the human parts automatically. In contrast to the part based subnet fusion model (Du, Wang, and Wang 2015), the learned co-occurrences of joints by our model do not limit the connections to be in the same part, as there are many large weights outside the diagonal regions, e.g., in the regions marked by white circles, making the network more powerful for action recognition. This also signifies the importance of the proposed full connection architecture. By averaging the energy of the weights in the same group of neurons for each joint, we obtain Fig. 6 (b) which has five groups of LSTM neurons. It is observed that different groups have different weight patterns, preferring different conjunctions of joints.



(a) (b)

Figure 6: Visualization of the absolute values of input gate weights in the first layer on SBU kinect interaction dataset. Horizontal axis denotes the indexes of 30 joints of two persons. (a) Vertical axis denotes the 100 LSTM neurons. Each element represents the absolute value of the weight between the corresponding joint and input gate unit of that LSTM neuron. Every three nearby joints form a part of a person. (b) Vertical axis denotes the five groups of LSTM neurons. We observe different groups have different weight patterns, preferring different conjunctions of joints.

5 Conclusion

In this paper, we propose an end-to-end fully connected deep LSTM network for skeleton based action recognition. The proposed model facilitates the automatic learning of feature co-occurrences from the skeleton joints through our designed regularization. To ensure effective learning of the deep model, we design an in-depth dropout algorithm for the LSTM neurons, which performs dropout for the internal gates, cell, and output response of the LSTM neuron. Experimental results demonstrate the state-of-the-art performance of our model on several datasets.

Acknowledgment

We would like to thank David Wipf from Microsoft Research Asia for the valuable discussions, and thank Yong Du from Institute of Automation, Chinese Academy of Sciences for providing Hierarchical RNN code for the comparison.

References

Cho, K., and Chen, X. 2014. Classifying and visualizing motion capture sequences using deep neural networks. In *International Conference on Computer Vision Theory and Applications*, 122–130.

CMU. 2003. CMU graphics lab motion capture database. http://mocap.cs.cmu.edu/.

Cotter, S. F.; Rao, B. D.; Engan, K.; and Kreutz-Delgado, K. 2005. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing* 53(7):2477–2488.

Du, Y.; Wang, W.; and Wang, L. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1110–1118.

Graves, A., and Schmidhuber, J. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks* 18(5):602–610.

Graves, A.; Mohamed, A.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 6645–6649.

Graves, A. 2012. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer.

Ji, Y.; Ye, G.; and Cheng, H. 2014. Interactive body part contrast mining for human interaction recognition. In *IEEE International Conference on Multimedia and Expo Workshops*, 1–6.

Johansson, G. 1973. Visual perception of biological motion and a model for it is analysis. *Perception and Psychophysics* 14(2):201–211.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 1097–1105.

Lefebvre, G.; Berlemont, S.; Mamalet, F.; and Garcia, C. 2013. BLSTM-RNN based 3D gesture classification. In *Proceedings of the International Conference on Artificial Neural Networks and Machine Learning*, 381–388.

Müller, M.; Röder, T.; Clausen, M.; Eberhardt, B.; Krüger, B.; and Weber, A. 2007. Documentation mocap database HDM05. Technical Report CG-2007-2, Universität Bonn.

Müller, M.; Röder, T.; and Clausen, M. 2005. Efficient content-based retrieval of motion capture data. *ACM Transactions on Graphic* 24(3):677–683.

Ofli, F.; Chaudhry, R.; Kurillo, G.; Vidal, R.; and Bajcsy, R. 2013. Berkeley MHAD: A comprehensive multimodal human action database. In *Proceedings of the IEEE Workshop on Applications on Computer Vision*, 53–60.

Pham, V.; Bluche, T.; Kermorvant, C.; and Louradour, J. 2014. Dropout improves recurrent neural networks for handwriting recognition. In *International Conference on Frontiers in Handwriting Recognition*, 285–290.

Poppe, R. 2010. A survey on vision-based human action recognition. *Image and Vision Computing* 28(6):976–990.

Savitzky, A., and Golay, M. J. 1964. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry* 36(8):1627–1639.

Sminchisescu, C.; Kanaujia, A.; Li, Z.; and Metaxas, D. 2005. Conditional models for contextual human motion recognition. In *IEEE Conference on Computer Vision*, volume 2, 1808–1815.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Sung, J.; Ponce, C.; Selman, B.; and Saxena, A. 2011. Human activity detection from RGBD images. In *AAAI workshop on Pattern, Activity and Intent Recognition*, 47–55.

Sung, J.; Ponce, C.; Selman, B.; and Saxena, A. 2012. Unstructured human activity detection from RGBD images. In *IEEE International Conference on Robotics and Automation*, 842–849.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.

Wang, J.; Liu, Z.; and Junsong, Y. 2012. Mining actionlet ensemble for action recognition with depth cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1290–1297.

Weinland, D.; Ronfard, R.; and Boyerc, E. 2011. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding* 115(2):224–241.

Xia, L.; Chen, C.-C.; and Aggarwal, J. K. 2012. View invariant human action recognition using histograms of 3D joints. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 20–27.

Yang, X., and Tian, Y. 2014. Effective 3D action recognition using EigenJoints. *Journal of Visual Communication and Image Representation* 25(1):2–11.

Yun, K.; Honorio, J.; Chattopadhyay, D.; Berg, T. L.; and Samaras, D. 2012. Two-person interaction detection using body pose features and multiple instance learning. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 28–35.

Zaremba, W.; Sutskever, I.; and Vinyals, O. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.