Pattern Recognition Letters 66 (2015) 100–108

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Factored four way conditional restricted Boltzmann machines for activity recognition ☆

Decebal Constantin Mocanu [a,*], Haitham Bou Ammar [b], Dietwig Lowet [c], Kurt Driessens [d], Antonio Liotta [a], Gerhard Weiss [d], Karl Tuyls [e]

[a] Eindhoven University of Technology, Department of Electrical Engineering, Den Dolech 2, Eindhoven, 5612 AZ, The Netherlands
[b] University of Pennsylvania, GRASP Laboratory, 3330 Walnut Street, Philadelphia, PA 19104-6228, USA
[c] Philips Research, Human Interaction and Experiences, High Tech Campus 34, Eindhoven, 5656 AE, The Netherlands
[d] Maastricht University, Department of Knowledge Engineering, Bouillonstraat 8-10, Maastricht, 6211 LH, The Netherlands
[e] University of Liverpool, Department of Computer Science, Ashton Street, Liverpool, L69 3BX, United Kingdom

ARTICLE INFO

Article history:
Available online 4 February 2015

Keywords:
Activity recognition
Deep learning
Restricted Boltzmann machines

ABSTRACT

This paper introduces a new learning algorithm for human activity recognition capable of simultaneous regression and classification. Building upon conditional restricted Boltzmann machines (CRBMs), Factored four way conditional restricted Boltzmann machines (FFW-CRBMs) incorporate a new label layer and four-way interactions among the neurons from the different layers. The additional layer gives the classification nodes a similar strong multiplicative effect compared to the other layers, and avoids that the classification neurons are overwhelmed by the (much larger set of) other neurons. This makes FFW-CRBMs capable of performing activity recognition, prediction and self auto evaluation of classification within one unified framework. As a second contribution, sequential Markov chain contrastive divergence (SMcCD) is introduced. SMcCD modifies Contrastive Divergence to compensate for the extra complexity of FFW-CRBMs during training. Two sets of experiments one on benchmark datasets and one a robotic platform for smart companions show the effectiveness of FFW-CRBMs.

1. Introduction

Robotic support for elderly people requires (possibly among others) capabilities such as monitoring and coaching [1,2], e.g., emergency detection and medication reminders, and accurate activity detection is vital for such services. On the monitoring side, a system that recognises human activity patterns allows for automated health guidance, as well as providing an objective measure for medical staff. Specifically, the fashion in which these daily activities are executed (e.g., speed, fluency) can serve as an important early indicator of possible problems. Accurate activity recognition is made difficult by the continuous nature of typical activity scenarios, which makes the task highly similar to time series prediction.

Much research has been aimed at detecting human activities based on the output of a variety of low-power, low-bandwidth sensors, such as passive infrared (PIR) sensors, and power and pressure meters placed either around the home, or on-body (e.g. accelerometers [3,4]). The drawback of such an approach lies in the inability to capture sufficiently reliable data that allows to differentiate between subtly different activities. In principle, the most accurate and suited sensors for activity recognition would be video-cameras in combination with advanced computer vision algorithms to interpret the data, but this approach leads to significant privacy issues.

As an alternative, we make use of motion capture data. More exactly, we use a Kinect® sensor[1] to generate a 3D point cloud and to extract the human skeleton joints from it. This approach yields relatively easy data to process and, as we will show, sufficient information to accurately recognise human activities.

Literature provides other techniques that can do both classification (i.e. from a set of possible time series categories determine to which category a new observation belongs or, in our case, recognise the activity performed by a person during a specific moment of time) and time series prediction (i.e. starting from the near history observations forecast the next values for a specific time series or, in our case, forecast the human body's movements or poses in the near future), each of them with their advantages and disadvantages. Among them, Linear Dynamic Systems such as Autoregression and Kalman filters are well suited to model linear time series. Although

☆ This paper has been recommended for acceptance by G. Sanniti di Baja.

* Corresponding author. Tel.: +31 685 342 994.
E-mail address: d.c.mocanu@tue.nl (D.C. Mocanu).

[1] http://en.wikipedia.org/wiki/Kinect, [Accessed 8th June 2014].

http://dx.doi.org/10.1016/j.patrec.2015.01.013
0167-8655/

extensions for non-linear systems exist, they still have difficulties with high non-linearity. Another successful class of time series models are hidden Markov models (HMMs). HMM have reached a lot of success in speech recognition. However, HMM models are also less suited for highly non-linear data and become unwieldy when the state space is large.

Recent research has profiled deep learning (DL) methods [5] as a promising alternative for pattern recognition problems. DL makes small steps towards mimicking the behaviour of the human brain [6,7]. It has been successfully applied to, for example, multi-class classification [8], collaborative filtering [9] and information retrieval [10]. Due to the success of DL based on restricted Boltzman machines (RBMs) [11] in modelling static data, a number of extensions for modelling time series have been developed. A straightforward extension of restricted Boltzmann machines to model time series are Temporal RBMs (TRBM) as described in [12]. Conceptually, a TRBM consists of a succession of RBMs (one for each time frame) with directed connections between the nodes representing consecutive timeframes. However, a lack of an efficient training method limits their application to real-world problems. Conditional RBMs (CRBM) propose a different extension of RBMs for modelling time sequences where two separate visible layers represent (i) the values from N previous time frames and (ii) those of the current time frame [13]. A CRBM can be viewed as adding AutoRegression to RBMs and hence are especially suited for modelling linear time variations. They have been successfully applied to motion capture data. To enable also the modelling of non-linear time variations, the CRBM concept has been further extended by incorporating three-way neural nodes interactions that are connected by a 3-way weight tensor [13]. To overcome the computational complexity induced by the 3-way weight tensor, the tensor can be factored resulting in a Factored Conditional RBM (FCRBM) [14]. These FCRBMs have been shown to give excellent results in modelling and predicting motion capture data. They are able to predict different human motion styles and combine two different styles into a new one.

To our knowledge, FCRBMs represent the current state of the art for capturing and predicting human motion, and therefore, we chose them as a basis for our work on activity recognition. However, the FCRBM is still not optimally suited to classify human motion or activities. The reason for this is that the hidden neurons in the FCRBM are used to model how the next frame of coordinates depends on the historic frames. The most natural way to extend the FCRBM to include classification capabilities is by letting the hidden neurons gate the interactions between the label and the prediction neurons. This results in a model with four-way neuron interactions.[2]

Hence, in this paper we propose a novel model, namely *Factored Four Way Conditional Restricted Boltzmann Machine* (FFW-CRBM) capable of both classification and prediction of human activity in one unified framework. An emergent feature of FFW-CRBM, so called self auto evaluation of the classification performance, may be very useful in the context of smart companions. It allows the machine to autonomously recognise that an activity is undetected and to trigger a retraining procedure. Due to the complexity of the proposed machine, the standard training method for DL models is unsuited. As a second contribution, we introduce *Sequential Markov chain Contrastive Divergence* (SMcCD), an adaptation of contrastive divergence (CD) [16]. To illustrate the efficacy and effectiveness of the model, we present results from two sets of experiments using real world data originating from (i) our previous developed smart companion robotic platform [17] and (ii) a benchmark database for activity recognition [18].

The remaining of this paper is organised as follows. Section 2 presents the mathematical definition of the problem tackled in this article. Section 3 presents background knowledge on deep learning for the benefit of the non-specialist reader. Section 4 details the mathematical model for the unfactorised version of the proposed method. Section 5 describes the FFW-CRBM model including the mathematical modelling. Section 6 describes the experiments performed and depict the achieved results. Finally, Section 7 concludes and presents directions of future research.

[2] Four-way (and higher) interactions are also biologically plausible since they appear to be necessary to explain the workings of the human brain [15].

2. Problem definition

In essence, in this paper, we aim at solving time series classification and prediction simultaneously in one unified framework. Let $i \in \mathbb{N}$ represent the index of available instances, $t \in \mathbb{N}$ to denote time, $\mathbb{R}^d$ a d-dimensional feature space, $t-N : t-1$ the temporal window of observations recorded in the N time steps before t, $\mathcal{C} = \{0, 1, \ldots, k\}$ the set of possible classes, and $\mathbf{\Theta}$ the parameters of a generic mathematical model. The targeted problem can then be written as:

Given a data set $\mathcal{D} = \{\mathbf{X}^{(i)}, \mathbf{y}^{(i)}\}$ for all instances i, where:

- $\mathbf{X}^{(i)} \in \mathbb{R}^{d\times(t-N:t-1)}$, is a real-valued input matrix consisting of d rows of features, and $N-1$ columns corresponding to the associated temporal window $t-N : t-1$,
- $\mathbf{y}_t^{(i)} \in \mathbb{R}^d \times \mathcal{C}$ is the corresponding multidimensional output vector consisting of the d-dimensional real-valued features at time t and an associated class label (e.g. a robotic companion that recognises an activity and predict the corresponding human poses to avoid collision).

Determine $p(\mathbf{Y}|\mathbf{\Gamma};\mathbf{\Theta})$, with $\mathbf{Y} = \{\mathbf{y}^{(i)}\}\ \forall i$ and $\mathbf{\Gamma} = \{\mathbf{X}^{(i)}\}\ \forall i$ representing the concatenation of all outputs and inputs respectively, such that: $\mathrm{KL}(p_{\mathrm{model}}(\mathbf{Y}|\mathbf{\Gamma};\mathbf{\Theta})||p_{\mathrm{empirical}}(\mathbf{Y}|\mathbf{\Gamma}))$ is minimised. KL represents the Kullback Leibler divergence between the empirical and approximated (i.e., model) distributions. This is signified by $p_{\mathrm{model}}(\mathbf{Y}|\mathbf{\Gamma};\mathbf{\Theta})$, which defines a joint distribution over $\mathbb{R}^d \times \mathcal{C}$ space.

3. Background

This section provides background knowledge needed for the remainder of the paper. Firstly, restricted Boltzmann machines (RBMs), being at the basis of the proposed technique, are detailed. Secondly, contrastive divergence, the algorithm used to fit the RBM's hyperparameters is detailed. Finally, factored conditional restricted Boltzmann machines, constituting the main motivation behind this work, are explained.

3.1. Restricted Boltzmann machine

Restricted Boltzmann machines (RBM) [11] are energy-based models for unsupervised learning. These models are stochastic with stochastic nodes and layers, making them less vulnerable to local minima [14]. Further, due to their neural configurations, RBMs posses excellent generalisation capabilities [5].

Formally, an RBM consists of visible and hidden binary layers. The visible layer represents the data, while the hidden increases the learning capacity by enlarging the class of distributions that can be represented to an arbitrary complexity. This paper uses the following notation: i represents the indices of the visible layer, j those of the hidden layer, and w_{ij} denotes the weight connection between the ith visible and jth hidden unit. Further, v_i and h_j denote the state of the ith visible and jth hidden unit, respectively. Using to the above notation, the energy function of an RBM is given by:

$$E(v,h) = -\sum_{i=1}^{n_v}\sum_{j=1}^{n_h} v_i h_j w_{ij} - \sum_{i=1}^{n_v} v_i a_i - \sum_{j=1}^{n_h} h_j b_j \tag{1}$$

where, a_i and b_j represent the biases of the visible and hidden layers, respectively; n_v and n_h are the number of neurons in the visible and hidden layer, respectively. The joint probability of a state

of the hidden and visible layers is defined as: $P(v,h) = \frac{\exp(-E(v,h))}{Z}$ with $Z = \sum_{x,y}\exp(-E(x,y))$. To determine the probability of a data point represented by a state v, the marginal probability is used. This is determined by summing out the state of the hidden layer as: $p(v) = \sum_h P(v,h) = \frac{\sum_h(\exp(-\sum_{i,j} v_i h_j w_{ij} - \sum_i v_i a_i - \sum_j h_j b_j))}{Z}$. In order to maximise the likelihood of the model, the gradients of the energy function with respect to the weights have to be calculated. Unfortunately, in RBMs maximum likelihood cannot be straightforwardly applied due to intractability problems. To circumvent these problems, contrastive divergence was introduced.

3.2. Contrastive divergence

In contrastive divergence (CD) [16], learning follows the gradient of:

$$\mathrm{CD}_n \propto D_{\mathrm{KL}}(p_0(\mathbf{x})||p_\infty(\mathbf{x})) - D_{\mathrm{KL}}(p_n(\mathbf{x})||p_\infty(\mathbf{x})) \tag{2}$$

where, $p_n(.)$ is the resulting distribution of a Markov chain running for n steps. To derive the update rules for w_{ij}, the energy function is re-written in a matrix form as: $E(\mathbf{v},\mathbf{h};\mathbf{W}) = -\mathbf{h}^{\mathrm{T}}\mathbf{W}\mathbf{v} - \mathbf{v}^{\mathrm{T}}\mathbf{a} - \mathbf{h}^{\mathrm{T}}\mathbf{b}$. $\mathbf{v} = [v_1,\ldots,v_{n_v}]$ is a binary vector collecting all visible units v_i, with n_v the index of the last visible neuron. $\mathbf{h} = [h_1,\ldots,h_{n_h}]$ is a binary vector collecting all the hidden units h_j, with n_h the index of the last hidden neuron. $\mathbf{W} \in \mathbb{R}^{n_h \times n_v}$ represents the matrix of all weights w_{ij}, $\mathbf{a} \in \mathbb{R}^{n_v}$,$\mathbf{b} \in \mathbb{R}^{n_h}$ are vectors containing the biases of $\mathbf{v}$ and $\mathbf{h}$, respectively. Since the visible units are conditionally independent given the hidden units and vice versa, learning can be performed using one step Gibbs sampling, which is carried in two half-steps: (1) update all the hidden units, and (2) update all the visible units. Thus, in CD_n the weight updates are done as follows: $w_{ij}^{\tau+1} = w_{ij}^{\tau} + \alpha(\langle\langle h_j v_i\rangle_{p(\mathbf{h}|\mathbf{v};\mathbf{W})}\rangle_0 - \langle h_j v_i\rangle_n)$ where τ is the iteration, α is the learning rate, $\langle\langle h_j v_i\rangle_{p(\mathbf{h}|\mathbf{v};\mathbf{W})}\rangle_0 = \frac{1}{N_l}\sum_{k=1}^{N_l} v_i^{(k)} P(h_j^{(k)} = 1|\mathbf{v}^{(k)};\mathbf{W})$ and $\langle h_j v_i\rangle_n = \frac{1}{N_l}\sum_{k=1}^{N_l} v_i^{(k)(n)} P(h_j^{(k)(n)} = 1|\mathbf{v}^{(k)(n)};\mathbf{W})$ where N_l is the total number of input instances, the superscript $^{(k)}$ shows the kth input instance. The superscript $^{(n)}$ indicates that the states are obtained after n iterations of Gibbs sampling from the Markov chain starting at $p_0(\cdot)$.

3.3. Factored conditional restricted Boltzmann machine

Conditional restricted Boltzmann machines (CRBM) [13] are an extension over RBMs used to model time series data and human activities. They use an undirected model with binary hidden variables $\mathbf{h}$, connected to real-valued (i.e. Gaussian) visible ones $\mathbf{v}$. At each time step t, the hidden and visible nodes receive a connection from the visible variables of the last N time-steps. The history of the values up to time t is collected in the real-valued history vector $\mathbf{v}_{<t}$. It is constructed by starting with the observations recorded at time step $t-N$, and after that, by adding sequentially after its last element, the observations recorded until time step $t-1$, where N represents the size of the temporal window considered. Thus, $\mathbf{v}_{<t} = [v_{1,t-N},\ldots,v_{n_v,t-N},\ldots,v_{1,t-1},\ldots,v_{n_v,t-1}]$. We mention that in any formula in the paper, we note with the subscript $_{,t}$ the present time step, and with the subscript $_{,<t}$ the previous N time steps, for any vector. The total energy of CRBM is given by:

$$E = \sum_{i=1}^{n_v}\frac{(\hat{a}_{i,t} - v_{i,t})^2}{2\sigma_i^2} - \sum_{j=1}^{n_h}\hat{b}_{j,t}h_{j,t} - \sum_{i=1}^{n_v}\sum_{j=1}^{n_h} W_{ij}\frac{v_{i,t}}{\sigma_i}h_{j,t} \tag{3}$$

where $\hat{a}_{i,t} = a_i + \sum_{k=1}^{n_{v_{<t}}} A_{ki}v_{k,<t}$ and $\hat{b}_{j,t} = b_j + \sum_{k=1}^{n_{v_{<t}}} B_{kj}v_{k,<t}$ represent the "dynamic biases", which include the static bias and the contribution from the past, $n_{v_{<t}} = n_v(N-1)$ is the number of elements in $\mathbf{v}_{<t}$, and σ_i represents the standard deviation.

To predict different types of time series within the same model, Taylor added three-way interactions between neurons. To reduce the computational complexity these three-way interactions are factored, resulting in factored CRBM (FCRBMs) [13]. Their energy is:

$$E = \frac{1}{2}\sum_{i=1}^{n_v}(\hat{a}_{i,t} - v_{i,t})^2 - \sum_{j=1}^{n_h}\hat{b}_{j,t}h_{j,t} - \sum_{f=1}^{n_F}\sum_{i=1}^{n_v}\sum_{j=1}^{n_h}\sum_{p=1}^{n_z} W_{if}^v W_{jf}^h W_{pf}^z v_{i,t}h_{j,t}z_{p,t} \tag{4}$$

where, $\mathbf{W}^v$, $\mathbf{W}^h$, and $\mathbf{W}^z$, represent the factored visible, factored hidden, and factored features weights, respectively. n_F is the number of factors, $\mathbf{z}_t$ is a vector for the deterministic features layer, and n_z the number of deterministic features. FCRBMs have been successfully used to model different styles of human motion and time series predictions. Interested readers are referred to [14] for a more comprehensive discussion.

Although successful, FCRBMs are not capable of performing classification and predictions in one unified framework. The proposed methods, explained next, solve this problem by introducing: (1) an additional label layer, and (2) four-way multiplicative interactions between neurons.

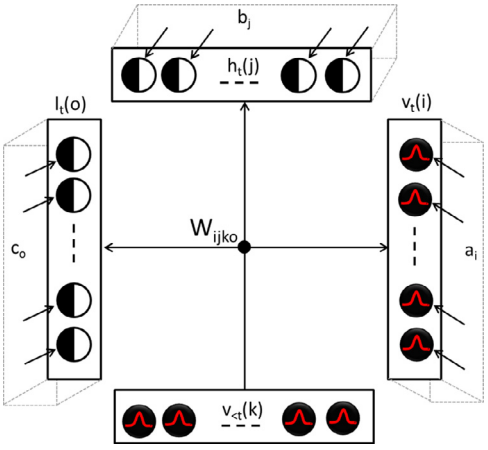

Fig. 1. Overall schematic of the proposed FW-CRBM showing the four layer configuration of the machine.

4. Four way conditional restricted Boltzmann machines

FFW-CRBMs are derived by factorizing the weight tensor of the original four-way conditional restricted Boltzmann machine (FW-CRBM). For ease of presentation, this section introduces the full FW-CRBM, which is then factorised leading to the FFW-CRBM in Section 5.

To classify and regenerate human motion, FW-CRBMs make use of a four layer configuration, shown in Fig. 1. The history layer $\mathbf{v}_{<t}$ describes past frames of an activity, the present layer $\mathbf{v}_t$ describes the current time step (prediction), the hidden layer $\mathbf{h}_t$ assures that the machine is complex enough to model the intended activities, and the label layer $\mathbf{l}_t$ guarantees that the machine is capable of classifying different types of activities. These layers are connected using a fourth order tensor $\mathbf{W}_{ijko} \in \mathbb{R}^{n_\mathbf{v}\times n_\mathbf{h}\times n_{\mathbf{v}_{<t}}\times n_\mathbf{l}}$, where $n_\mathbf{v}$, $n_\mathbf{h}$, $n_{\mathbf{v}_{<t}}$, $n_\mathbf{l}$ represent the number of neurons from the present, hidden, history and label layers, respectively. Furthermore, each of the present, hidden and label layers include $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ biases, respectively.

Formally, an FW-CRBM defines a joint probability over $\mathbf{v}_t$, $\mathbf{h}_t$, and $\mathbf{l}_t$. This distribution is conditioned by $\mathbf{v}_{<t}$, and the model parameters Θ. Therefore, the probability distribution is defined as follows: $P(\mathbf{v}_t,\mathbf{h}_t,\mathbf{l}_t|\mathbf{v}_{<t},\Theta) = \frac{\exp(-\mathbf{E}(\mathbf{v}_t,\mathbf{h}_t,\mathbf{l}_t|\mathbf{v}_{<t},\Theta))}{Z}$, where, $\mathbf{E}(\mathbf{v}_t,\mathbf{h}_t,\mathbf{l}_t|\mathbf{v}_{<t},\Theta)$ represents the total energy of the model detailed in Eq. (5), and Z is the normalisation term, called the partition function, and calculated according to: $Z = \sum_{\mathbf{v}_t,\mathbf{h}_t,\mathbf{l}_t}\exp(-\mathbf{E}(\mathbf{v}_t,\mathbf{h}_t,\mathbf{l}_t|\mathbf{v}_{<t},\Theta))$.

4.1. The energy of FW-CRBMs

FW-CRBMs' energy function is defined as:

$$\mathbf{E}(\mathbf{v}_t, \mathbf{h}_t, \mathbf{l}_t|\mathbf{v}_{<t}, \Theta) = -\sum_{i=1}^{n_v}\frac{(v_{i,t}-a_i)^2}{{\sigma_i}^2} - \sum_{j=1}^{n_h} h_{j,t}b_j - \sum_{o=1}^{n_l} l_{o,t}c_o - \sum_{i=1}^{n_v}\sum_{j=1}^{n_h}\sum_{k=1}^{n_{v<t}}\sum_{o=1}^{n_l} W_{ijko}\frac{v_{i,t}}{\sigma_i}h_{j,t}\frac{v_{k,<t}}{\sigma_k}l_{o,t} \quad (5)$$

where, σ_i and σ_k represent the standard deviation for the corresponding neurons from the present and history layer respectively. $\sum_{i=1}^{n_v}\frac{(v_{i,t}-a_i)^2}{{\sigma_i}^2}$, $\sum_{j=1}^{n_h} h_{j,t}b_j$, $\sum_{o=1}^{n_l} l_{o,t}c_o$ represent the energy contributions of each of the visible, hidden, and label neurons respectively and $\sum_{i=1}^{n_v}\sum_{j=1}^{n_h}\sum_{k=1}^{n_{v<t}}\sum_{o=1}^{n_l} W_{ijko}\frac{v_{i,t}}{\sigma_i}h_{j,t}\frac{v_{k,<t}}{\sigma_k}l_{o,t}$ describes the contribution of the weight tensor to the overall energy function.

4.2. Probabilistic inference in FW-CRBMs

Since there are no connections between the neurons in the same layer, inference can be performed in parallel. The overall input of each of the hidden unit $s^h_{j,t}$, visible unit $s^v_{i,t}$, and labelled unit, $s^l_{o,t}$ is calculated according to:

$$s^h_{j,t} = \sum_{i=1}^{n_v}\sum_{k=1}^{n_{v<t}}\sum_{o=1}^{n_l} W_{ijko}\frac{v_{i,t}}{\sigma_i}\frac{v_{k,<t}}{\sigma_k}l_{o,t} \text{ (for } j\text{th hidden unit)} \quad (6)$$

$$s^v_{i,t} = \sum_{j=1}^{n_h}\sum_{k=1}^{n_{v<t}}\sum_{o=1}^{n_l} W_{ijko}h_{j,t}\frac{v_{k,<t}}{\sigma_k}l_{o,t} \text{ (for } i\text{th visible unit)} \quad (7)$$

$$s^l_{o,t} = \sum_{i=1}^{n_v}\sum_{j=1}^{n_h}\sum_{k=1}^{n_{v<t}} W_{ijko}\frac{v_{i,t}}{\sigma_i}h_{j,t}\frac{v_{k,<t}}{\sigma_k} \text{ (for } o\text{th labelled unit)} \quad (8)$$

For each of the jth hidden ith visible and oth labelled unit, inference is performed according to:

$$p(h_{j,t}=1|\mathbf{v}_t, \mathbf{v}_{<t}, \mathbf{l}_t) = \text{sigmoid}(-b_j - s^h_{j,t}) \quad (9)$$

$$p(v_{i,t}=x|\mathbf{h}_t, \mathbf{v}_{<t}, \mathbf{l}_t) = \mathcal{N}(a_i + s^v_{i,t}, \sigma_i^2) \quad (10)$$

$$p(l_{o,t}=1|\mathbf{v}_t, \mathbf{v}_{<t}, \mathbf{h}_t) = \text{sigmoid}(-c_o - s^l_{o,t}) \quad (11)$$

where, sigmoid$(\cdot)$ is the sigmoidal function, and $\mathcal{N}$ is the Gaussian distribution.

4.3. Learning in FW-CRBMs

The update rules are attained by deriving the energy function with respect to the free parameters (i.e., the weights tensor, and the biases of each of the layers) leading to:

$$\Delta W_{ijko} \propto v_{k,<t}\langle v_{i,t}h_{j,t}l_{o,t}\rangle_0 - v_{k,<t}\langle v_{i,t}h_{j,t}l_{o,t}\rangle_K \quad (12)$$

$$\Delta a_i \propto \langle v_{i,t}\rangle_0 - \langle v_{i,t}\rangle_K \quad (13)$$

$$\Delta b_j \propto \langle h_{j,t}\rangle_0 - \langle h_{j,t}\rangle_K \quad (14)$$

$$\Delta c_o \propto \langle l_{o,t}\rangle_0 - \langle l_{o,t}\rangle_K \quad (15)$$

with K being the number of steps of a Markov Chain, W_{ijko} representing the weights connecting the four layers and a_i, b_j, and c_o denoting the ith bias of the present, the jth bias of the hidden, and the oth bias of the label layers, respectively.

Although successful, FW-CRBMs incur a computational complexity of $\mathcal{O}(n^4)$ making them unsuitable for real-world applications. Next, the more efficient counter-part, the *factored* FW-CRBM (i.e., FFW-CRBM) is introduced.

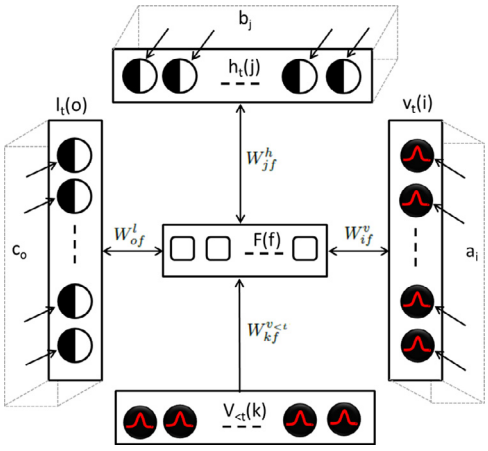

Fig. 2. Overall schematic of the proposed FFW-CRBM showing the four layer configuration of the machine as well as the factored weight tensor.

5. Factored four way conditional restricted Boltzmann machine

To reduce the computational complexity of FW-CRBM from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^2)$, FFW-CRBM factors the 4th order weight tensor (i.e., W_{ijko}) to a sum of products of second order tensors [19]. A high level schematic depicting such a factorisation is shown in Fig. 2. Factoring of the four-way weight tensor is achieved according to:

$$W_{ijko} = \sum_{f=1}^{n_F} W^v_{if}W^h_{jf}W^{v_{<t}}_{kf}W^l_{of} \quad (16)$$

where n_F is number of factors and i, j, k, and o represent the indices of the visible layer neurons $\mathbf{v}_t$, the hidden layer neurons $\mathbf{h}_t$, the history layer neurons $\mathbf{v}_{<t}$ and the labelled layer neurons $\mathbf{l}_t$ respectively. Furthermore, $\mathbf{W}^v$, $\mathbf{W}^h$, $\mathbf{W}^l$ represent the bidirectional and symmetric weights from the visible, hidden and label layers to the factors, respectively. Moreover, $\mathbf{W}^{v_{<t}}$ denotes the directed weights from the history layer to the factors.

Although, FFW-CRBMs behave similar to FW-CRBMs (i.e., having two informational flows, one for classification and one for prediction), the mathematical formalisation needs to be re-derived using the factored weights of Eq. (16).

5.1. Energy for the factored model

In the case of FFW-CRBMs' energy, the first three terms of Eq. (5) remain unchanged. However, the fourth term makes use of the factoring in Eq. (16), yielding:

$$\mathbf{E}(\mathbf{v}_t, \mathbf{h}_t, \mathbf{l}_t|\mathbf{v}_{<t}, \Theta) = -\sum_{i=1}^{n_v}\frac{(v_{i,t}-a_i)^2}{{\sigma_i}^2} - \sum_{j=1}^{n_h} h_{j,t}b_j - \sum_{o=1}^{n_l} l_{o,t}c_o - \sum_{f=1}^{n_F}\sum_{i=1}^{n_v} W^v_{if}\frac{v_{i,t}}{\sigma_i}\sum_{j=1}^{n_h} W^h_{jf}h_{j,t}\sum_{k=1}^{n_{v<t}} W^{v_{<t}}_{kf}\frac{v_{k,<t}}{\sigma_k}\sum_{o=1}^{n_l} W^l_{of}l_{o,t} \quad (17)$$

5.2. Probabilistic inference in the factored model

Inference in FFW-CRBMs is conducted in parallel as in the FW-CRBM case. Nonetheless, the inputs, for each of the hidden, visible and label nodes are given respectively, by:

$$s^h_{j,t} = \sum_{f=1}^{n_F} W^h_{jf}\sum_{i=1}^{n_v} W^v_{if}\frac{v_{i,t}}{\sigma_i}\sum_{k=1}^{n_{v_{<t}}} W^{v_{<t}}_{kf}\frac{v_{k,<t}}{\sigma_k}\sum_{o=1}^{n_l} W^l_{of}l_{o,t} \quad (18)$$

$$s^v_{i,t} = \sum_{f=1}^{n_F} W^v_{if}\sum_{j=1}^{n_h} W^h_{jf}h_{j,t}\sum_{k=1}^{n_{v_{<t}}} W^{v_{<t}}_{kf}\frac{v_{k,<t}}{\sigma_k}\sum_{o=1}^{n_l} W^l_{of}l_{o,t} \quad (19)$$

$$s^l_{o,t} = \sum_{f=1}^{n_F} W^l_{of} \sum_{i=1}^{n_v} W^v_{if} \frac{v_{i,t}}{\sigma_i} \sum_{j=1}^{n_h} W^h_{jf} h_{j,t} \sum_{k=1}^{n_{v<t}} W^{v_{<t}}_{kf} \frac{v_{k,<t}}{\sigma_k} \tag{20}$$

These are then substituted in Eqs. (9)–(11) for determining the probability distributions for each of the visible, hidden and label layers.

5.3. Learning in the factored model

This section is devoted to the learning procedure of FFW-CRBMs. First, the update rules are derived, then an explanation of SMcCD is detailed.

5.3.1. Update rules

The general update rule for the hyper-parameters $\mathbf{\Theta}$ is given by:

$$\mathbf{\Theta}_{\tau+1} = \mathbf{\Theta}_\tau + \rho \Delta \mathbf{\Theta}_\tau + \alpha(\Delta \mathbf{\Theta}_{\tau+1} - \gamma \mathbf{\Theta}_\tau) \tag{21}$$

where τ, ρ, α, and γ represent the update number, momentum, learning rate, and weights decay, respectively. The interested reader is referred to [20] for a more thorough discussion on the choice of such parameters. The *delta* rule for each of the hyper-parameters can be computed by deriving the energy function from Eq. (17), yielding:

$$\Delta W^v_{if} \propto \left\langle v_{i,t} \sum_{j=1}^{n_h} W^h_{jf} h_{j,t} \sum_{k=1}^{n_{v<t}} W^{v_{<t}}_{kf} v_{k,<t} \sum_{o=1}^{n_l} W^l_{of} l_{o,t} \right\rangle_0 - \left\langle v_{i,t} \sum_{j=1}^{n_h} W^h_{jf} h_{j,t} \sum_{k=1}^{n_{v<t}} W^{v_{<t}}_{kf} v_{k,<t} \sum_{o=1}^{n_l} W^l_{of} l_{o,t} \right\rangle_\lambda \tag{22}$$

$$\Delta W^{v_{<t}}_{kf} \propto \left\langle v_{k,<t} \sum_{j=1}^{n_h} W^h_{jf} h_{j,t} \sum_{i=1}^{n_v} W^v_{if} v_{i,t} \sum_{o=1}^{n_l} W^l_{of} l_{o,t} \right\rangle_0 - \left\langle v_{k,<t} \sum_{j=1}^{n_h} W^h_{jf} h_{j,t} \sum_{i=1}^{n_v} W^v_{if} v_{i,t} \sum_{o=1}^{n_l} W^l_{of} l_{o,t} \right\rangle_\lambda \tag{23}$$

$$\Delta W^l_{of} \propto \left\langle l_{o,t} \sum_{k=1}^{n_{v<t}} W^{v_{<t}}_{kf} v_{k,<t} \sum_{j=1}^{n_h} W^h_{jf} h_{j,t} \sum_{i=1}^{n_v} W^v_{if} v_{i,t} \right\rangle_0 - \left\langle l_{o,t} \sum_{k=1}^{n_{v<t}} W^{v_{<t}}_{kf} v_{k,<t} \sum_{j=1}^{n_h} W^h_{jf} h_{j,t} \sum_{i=1}^{n_v} W^v_{if} v_{i,t} \right\rangle_\lambda \tag{24}$$

$$\Delta W^h_{jf} \propto \left\langle h_{j,t} \sum_{k=1}^{n_{v<t}} W^{v_{<t}}_{kf} v_{k,<t} \sum_{i=1}^{n_v} W^v_{if} v_{i,t} \sum_{o=1}^{n_l} W^l_{of} l_{o,t} \right\rangle_0 - \left\langle h_{j,t} \sum_{k=1}^{n_{v<t}} W^{v_{<t}}_{kf} v_{k,<t} \sum_{i=1}^{n_v} W^v_{if} v_{i,t} \sum_{o=1}^{n_l} W^l_{of} l_{o,t} \right\rangle_\lambda \tag{25}$$

$$\Delta a_i \propto \langle v_{i,t} \rangle_0 - \langle v_{i,t} \rangle_\lambda \tag{26}$$

$$\Delta b_j \propto \langle h_{j,t} \rangle_0 - \langle h_{j,t} \rangle_\lambda \tag{27}$$

$$\Delta c_o \propto \langle l_{o,t} \rangle_0 - \langle l_{o,t} \rangle_\lambda \tag{28}$$

with λ being a Markov chain step running for a total number of K steps and starting at the original data distribution.

5.3.2. Sequential Markov chain contrastive divergence

Due to the fact that in the negative phase of the parameters update, the present and label layers have to be modified abide their common dependency, common contrastive divergence cannot be directly applied. To remedy this problem, sequential Markov chain contrastive divergence (SMcCD) is introduced. SMcCD extends CD by running two sequential Markov chains as shown in Algorithm 1. The first reconstructs $\mathbf{v}_t$, after the initialisation of all neurons from $\mathbf{v}_t$ with 0, using the current machine's configuration, while fixing the values at the label and past layers ($\mathbf{l}_t$ and $\mathbf{v}_{<t}$) to the current training instance. The second tries to reconstruct the label layer $\mathbf{l}_t$ from the $\mathbf{v}_t$ and $\mathbf{v}_{<t}$. The weights updates are performed at each step of the Markov chain. After a number of iterations over the training data, the weights (and thus the FFW-CRBM), representing the minimised energy level, can then be used for classification as well as prediction.

Sequential Markov Chain Contrastive Divergence:
Inputs: TD - set of training data;
K - number of Markov Chain steps;
Initialization: $\mathbf{\Theta} \leftarrow \mathcal{N}(0, \sigma^2)$;
set α, ρ, γ;
for *all epochs* **do**
  for *each Sample* $\in$ *TD* **do**
    %%First Markov Chain to reconstruct $\mathbf{v}_t$;
    $\mathbf{v}_t \leftarrow$ initialization with 0;
    $\mathbf{l}_t$ = Sample.Label;
    $\mathbf{v}_{<t}$ = Sample.History;
    $\mathbf{h}_t$ = InferHiddenLayer($\mathbf{v}_t$,$\mathbf{l}_t$,$\mathbf{v}_{<t}$,$\mathbf{\Theta}$);
    for $\lambda = 0; \lambda < K; \lambda++$ **do**
      %%Positive phase;
      pSt=GetPosStats($\mathbf{h}_t$,Sample.Present,$\mathbf{l}_t$,$\mathbf{v}_{<t}$,$\mathbf{\Theta}$);
      %%Negative phase;
      $\mathbf{v}_t$=InferPresentLayer($\mathbf{h}_t$,$\mathbf{l}_t$,$\mathbf{v}_{<t}$,$\mathbf{\Theta}$);
      $\mathbf{h}_t$ = InferHiddenLayer($\mathbf{v}_t$,$\mathbf{l}_t$,$\mathbf{v}_{<t}$,$\mathbf{\Theta}$);
      nSt=GetNegStats($\mathbf{h}_t$,$\mathbf{v}_t$,$\mathbf{l}_t$,$\mathbf{v}_{<t}$,$\mathbf{\Theta}$);
      $\mathbf{\Theta}$=UpdateWeights(**pSt**,**nSt**,$\mathbf{\Theta}$,α,ρ,γ);
    end
    %%Second Markov Chain to reconstruct $\mathbf{l}_t$;
    $\mathbf{l}_t \leftarrow$ initialization with 0;
    $\mathbf{v}_t$ = Sample.Present;
    $\mathbf{v}_{<t}$ = Sample.History;
    $\mathbf{h}_t$ = InferHiddenLayer($\mathbf{v}_t$,$\mathbf{l}_t$,$\mathbf{v}_{<t}$,$\mathbf{\Theta}$);
    for $\lambda = 0; \lambda < K; \lambda++$ **do**
      %%Positive phase;
      pSt=GetPosStats($\mathbf{h}_t$,Sample.Label,$\mathbf{v}_t$,$\mathbf{v}_{<t}$,$\mathbf{\Theta}$);
      %%Negative phase;
      $\mathbf{l}_t$=InferLabelLayer($\mathbf{h}_t$,$\mathbf{v}_t$,$\mathbf{v}_{<t}$,$\mathbf{\Theta}$);
      $\mathbf{h}_t$ = InferHiddenLayer($\mathbf{v}_t$,$\mathbf{l}_t$,$\mathbf{v}_{<t}$,$\mathbf{\Theta}$);
      nSt=GetNegStats($\mathbf{h}_t$,$\mathbf{v}_t$,$\mathbf{l}_t$,$\mathbf{v}_{<t}$,$\mathbf{\Theta}$);
      $\mathbf{\Theta}$=UpdateWeights(**pSt**,**nSt**,$\mathbf{\Theta}$,α,ρ,γ);
    end
  end
end

Algorithm 1: Sequential Markov Chain Contrastive Divergence

5.3.3. Self auto evaluation (SAE) of the classification performance

Given that FFW-CRBMs are capable of performing classification and prediction using the same free parameters, a three steps procedure, can be used in real time applications to evaluate classification performance. Firstly, the machine classifies the current observation (i.e., finding the label $\mathbf{l}_t$ at time t, based on history, $\mathbf{v}_{<t}$, and present frames, $\mathbf{v}_t$). Secondly, a prediction of the next values on the visible neurons $\mathbf{v}_{t+1}$ at time $t+1$, using the previously obtained label $\mathbf{l}_t$, and history frames, is performed. Finally, the Root Mean Square Error (RMSE) can be used to compare the prediction $\mathbf{v}_{t+1}$ with the actual observation acquired from sensory data.

6. Experiments and results

Two sets of experiments were performed to test the proposed models. In the first, FFW-CRBMs were used to classify and predict on skeleton data gathered using a Microsoft Kinect™ sensor, as shown in Fig. 3. In the second experiment (i.e., Section 6.2) FFW-CRBM was tested on the Berkeley multimodal human dataset [18]. Results in each of the above experiments demonstrate that FFW-CRBMs outperform state-of-the-art techniques, such as SVMs [21], CRBMs, and FCRBMs.

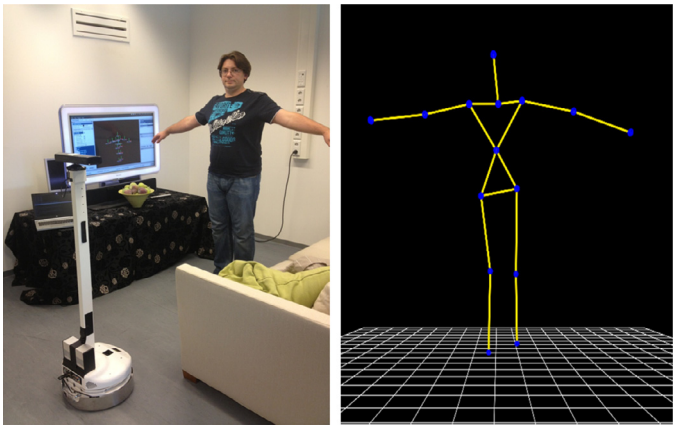

Fig. 3. The overall setup of detecting human activities. On the left a person performs a certain activity in front of a robot equipped with the Microsoft Kinect™ sensor. This is then encoded using the coordinates of the 15 joints shown on the right.

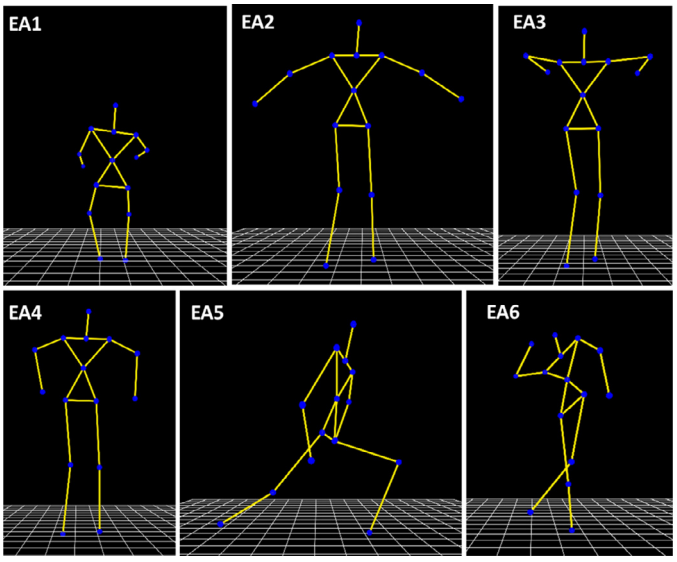

Fig. 4. Screen-shots with the human skeleton joints for the exercise activities experiment.

6.1. Human activity recognition

In this set of experiments, FFW-CRBMs were tested on real-world data acquired through a Microsoft Kinect™ sensor operating under the Robotic Operating System[3] (ROS) framework[4] installed on our robot [17]. Two classes of experiments were conducted. In the first, exercise activities (EA) were performed in front of the Kinect sensor and the positions of the joints of the exercising subjects were collected. These exercises involved: "body squats" (EA1), "vertical to horizontal hand movements" (EA2), "opening and closing of arms while moving" (EA3), "jumping" (EA4), "leg lunges" (EA5), and "walking" (EA6), as depicted in Fig. 4. In the second set, a more difficult scenario was considered. Here, users performed table activities (TA), which included: "phoning" (TA1), "typing" (TA2), "eating a sandwich" (TA3), "eating using a knife and a fork" (TA4), "reading" (TA5), and "writing" (TA6). The goals in each of the two experiments were to: (1) classify the activity, (2) predict the human poses for each of the activities, and (3) assess the SAE procedure. To determine the performance of the model we used 5-fold cross-validation, by splitting the data in 5 folds. Within each fold, to avoid altering the time series, we kept the data in their chronological order (i.e. keep the continuity of the human poses when an activity is performed). This splitting was helpful to have a ground truth for assessing the multi-step prediction performance. Results show that FFW-CRBM outperforms each of: (1) SVMs (in classification) and (2) CRBMs and FCRBMs (in prediction).

The inputs were frames of 45 dimensions, corresponding to 15 joints, at a certain time instance t. Each joint is represented in the three dimensional space, by the (x, y, z) absolute coordinates. The origin of the coordinates system is situated in the RGB camera of the Kinect sensor. The layers of the FFW-CRBM were set to 45 neurons in the visible layer (one for each of the dimensions), 6 neurons in the label layer (one for each of the activities), and 1080 history neurons in the history layer corresponding to 24 frames. Different values for the number of hidden neurons and number of factors were tried, by performing cross-validation on a small amount of instances picked randomly from the datasets. As a result, the number of hidden neurons was set to 40 with 40 factors. In this configuration the FFW-CRBM had 46 931 weight parameters. The initial learning rate was 10^{-4} to guarantee a bounded reconstruction error. The initial number of the Markov Chain steps in the training phase was set to 10. The initialisation of the weights was $\mathcal{N}(0, 0.3)$. In this model, the four way tensor is written as a product of four two way tensors and $0.3^4 = 0.0081$, which represents the usual value for the variance of the weight initialisations in standard RBMs. Further particularities, such as the momentum and the weight decay were set to 0.9 and 0.0002 respectively [20]. After a given initial number of iterations, the learning rate was decreased to 10^{-5} and the SMcCD steps were increased to 50 and a new set of iterations was started.

6.1.1. Classification on EA

In the case of exercise activities, the data set initially consisted of 3876 instances covering all six activities (i.e. EA1, EA2, EA3, EA4, EA5, EA6). Table 1 reports the classification results and the comparison between FFW-CRBM and SVMs using a radial basis function (RBF) kernel and the default parameters from LIBSVM [22]. It is clear that FFW-CRBM outperforms SVMs. For instance, FFW-CRBM achieved about 89% accuracy, compared to 65% for SVMs classifying EA6 (i.e., walking).

6.1.2. Predictions on EA

In the second phase of this experiment, one-step and multi-step predictions of human skeleton joint coordinates was considered. Here, a class label was fixed and the machine re-generates the joints values. In other words, given a class label, the task was to determine the 45-dimensional real valued output on the visible layer. Due to the properties of FFW-CRBMs, there is no need to re-train the inverse of the learned classifier which is nearly impossible with existing techniques such as SVMs. Using the FFW-CRBM, the task was performed by running a time step in the network to determine the visible unit values (i.e., the inverse problem). The results of this re-generation are shown in Tables 2 and 3, where the errors between the true values and the predicted values of FFW-CRBMs, CRBMs, and FCRBMs are presented with mean and standard deviation, over all testing instances. The metric used to measure these errors was RMSE.

One-step prediction: From the results shown in Table 2, it is clear that FFW-CRBM outperforms current state-of-the-art techniques in one-step predictions, where it attains the lowest reconstruction error of 0.018 compared to 0.036 for FCRBM and 0.106 for normal CRBMs.

Multi-step prediction: In this experiment the machine was allowed to progress for a number of steps. Prediction errors for each of the activities were monitored. In Table 3 the minimum, mean, and maximum incurred errors over all the activities at $10, 20, \ldots, 50$ steps,

[3] http://wiki.ros.org/ [Accessed 8th June 2014].

[4] Please note, the FW-CRBM is too computationally expensive to apply in real-world experiments. It is for this reason, that FFW-CRBM was used.

Table 1
Confusion matrix in percentages (with mean ± standard deviation) for FFW-CRBM vs. SVM on classifying human exercise activities. In the top of the table are the FFW-CRBM results, while the bottom represents the SVM results.

Method	Activities	EA1	EA2	EA3	EA4	EA5	EA6
FFW-CRBM	EA1	**98 ± 1.5**	1.3 ± 0.8	0	0	0	0.7 ± 0.6
	EA2	1.3 ± 1.2	**98.7 ± 1.2**	0	0	0	0
	EA3	0.6 ± 0.9	0	**94 ± 2.6**	1.2 ± 1.1	4.2 ± 0.7	0
	EA4	6 ± 1.2	0	0	**94 ± 1.2**	0	0
	EA5	0	0	0	0	**98.9 ± 1.1**	1.1 ± 1.1
	EA6	0	0	7.4 ± 2.1	0	3.6 ± 1.6	**89 ± 2.3**
SVM	EA1	94 ± 2.1	0	0	0	0.7 ± 0.8	5.3 ± 1.9
	EA2	8.5 ± 3.1	91.5 ± 3.1	0	0	0	0
	EA3	15 ± 2.3	2.3 ± 1.4	82.1 ± 3.4	0	0.6 ± 0.8	0
	EA4	47 ± 5.3	1 ± 1.2	0	52 ± 4.7	0	0
	EA5	6 ± 1.6	0	0	0	94 ± 1.6	0
	EA6	28 ± 2.3	3.1 ± 1.6	0	1.9 ± 1.4	2 ± 1.8	65 ± 2.7

Table 2
One step prediction, RMSE values (with mean ± standard deviation) of human skeleton joints averaged over all test instances for human exercise activities using FFW-CRBM, CRBM and FCRBM.

Activities	CRBM	FCRBM	FFW-CRBM
EA1	0.110 ± 0.005	0.054 ± 0.002	**0.028 ± 0.008**
EA2	0.138 ± 0.003	0.036 ± 0.006	**0.018 ± 0.012**
EA3	0.106 ± 0.012	0.044 ± 0.021	**0.023 ± 0.011**
EA4	0.126 ± 0.011	0.094 ± 0.008	**0.027 ± 0.014**
EA5	0.125 ± 0.004	0.068 ± 0.011	**0.026 ± 0.009**
EA6	0.123 ± 0.026	0.093 ± 0.007	**0.048 ± 0.019**

are shown for CRBM, FCRBM, and FFW-CRBM. FFW-CRBM outperforms the other techniques, having a maximum averaged prediction error of 0.094 after 50 prediction steps.

6.1.3. Self auto evaluation on EA

In Fig. 5, the results of the SAE procedure are averaged for all joint-coordinates, over all correctly or incorrectly classified instances for exercise activities. For any activity, the prediction error of incorrectly classified instances is at least 2.5 times higher than the prediction error for correct classified instances, with a maximum of 7 times bigger for EA6. These demonstrate that as the classification is wrong the prediction error increases dramatically. In such a scenario, even a very simple technique, e.g., thresholding, can be adopted to decide whether to retrain the model, on novel unknown activities.

6.1.4. Classification on TA

The same experiments were repeated in a more difficult scenario. In table activities there are a lot of similarities between the joint movements making it harder to differentiate among them. The set-up of the FFW-CRBM was identical to the previous case. The dataset however, consisted of 12483 instances. Classification results shown in

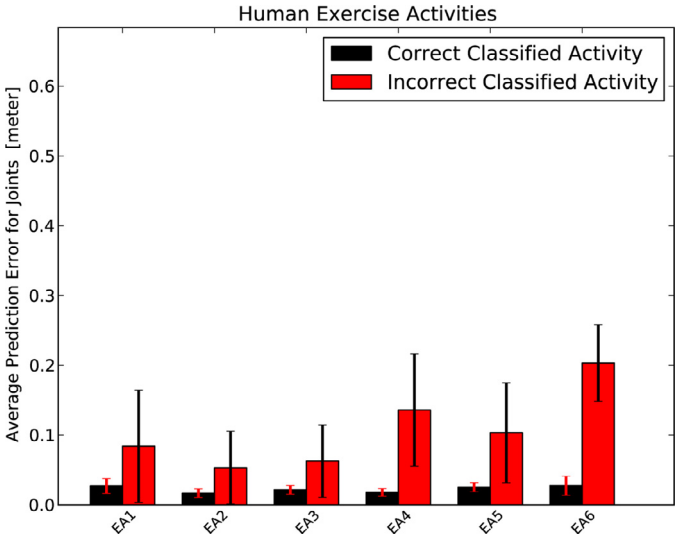

Fig. 5. Self auto evaluation results of FFW-CRBM in the case of human exercise activities. For any activity classified correctly, the black histograms show the mean of the prediction error, and the top red lines represent the standard deviations. The red histograms show the same, but in the case of wrong classified instances.

Table 4 demonstrate that FFW-CRBM is again capable of outperforming SVMs with RBF kernels where, for instance, FFW-CRBM achieved 81% accuracy compared to 66.5% on TA1 (i.e., phoning).

6.1.5. Predictions on TA

Here again, the task was to regenerate joint movements from class labels. Results are summarised in Table 5 for one step predictions and in Table 6 for multi-step predictions.

Table 3
Multi-step prediction results of CRBM, FCRBM and FFW-CRBM on exercise activities. The table represents the minimum, mean and maximum incurred errors over all activities at different numbers of prediction steps with mean (μ) and standard deviation (σ). (For interpretation of the references to colour in this table legend, the reader is referred to the web version of this article.)

Prediction		Minimum			Mean			Maximum		
		CRBM	FCRBM	FFW-CRBM	CRBM	FCRBM	FFW-CRBM	CRBM	FCRBM	FFW-CRBM
10 steps	μ	0.102	0.039	**0.022**	0.116	0.048	**0.038**	0.128	0.076	**0.071**
	σ	0.002	0.003	0.004	0.007	0.016	0.021	0.011	0.007	0.021
20 steps	μ	0.112	0.037	**0.028**	0.121	0.046	**0.043**	0.132	0.083	0.093
	σ	0.004	0.002	0.003	0.008	0.019	0.026	0.011	0.006	0.014
30 steps	μ	0.109	0.038	**0.037**	0.121	0.049	**0.047**	0.124	0.088	**0.077**
	σ	0.004	0.002	0.009	0.011	0.021	0.017	0.014	0.005	0.012
40 steps	μ	0.110	0.037	**0.035**	0.118	0.059	**0.056**	0.131	0.139	**0.078**
	σ	0.003	0.003	0.012	0.011	0.035	0.028	0.004	0.017	0.016
50 steps	μ	0.110	0.037	**0.037**	0.119	0.086	**0.059**	0.126	0.332	**0.094**
	σ	0.006	0.003	0.004	0.012	0.102	0.027	0.006	0.082	0.031

Table 4
Confusion matrix in percentages (with mean ± standard deviation) for FFW-CRBM versus SVM on classifying human table activities, clearly manifesting that the former outperforms the latter.

Method	Activities	TA1	TA2	TA3	TA4	TA5	TA6
FFW-CRBM	TA1	**81 ± 4.7**	4 ± 2.3	9.7 ± 2.6	0	3.6 ± 2.1	1.7 ± 1.2
	TA2	2.1 ± 0.8	**91.3 ± 3.2**	0	2.5 ± 2.9	0	4.1 ± 2.8
	TA3	1.4 ± 1.8	2.3 ± 1.9	**86.3 ± 4.2**	0	4 ± 0.9	6 ± 3.1
	TA4	0	0	0	**93 ± 4.2**	7 ± 4.2	0
	TA5	0	0	0	8.5 ± 1.3	**91.5 ± 1.3**	0
	TA6	0.6 ± 0.9	0	0	1.5 ± 1.2	1.8 ± 0.6	**96.1 ± 1.6**
SVM	TA1	66.5 ± 4.2	7 ± 1.6	21 ± 4.3	5.5 ± 3.2	0	0
	TA2	7 ± 3.2	78.1 ± 2.6	0	13.7 ± 2.9	0	1.2 ± 1.4
	TA3	19.4 ± 3.8	6 ± 2.4	70.2 ± 5.7	3 ± 2.8	1.4 ± 0.6	0
	TA4	1.4 ± 1.8	0	3 ± 1.3	82.5 ± 2.6	7.1 ± 1.8	6 ± 3.7
	TA5	0	14.1 ± 2.1	3.9 ± 1.2	5 ± 2.7	65 ± 4.2	12 ± 3.1
	TA6	0	5 ± 2.1	0	0	7.7 ± 0.9	87.3 ± 1.7

Table 5
One step prediction, RMSE values (with mean ± standard deviation) of human skeleton joints averaged over all test instances for human table activities. FFW-CRBM is compared against CRBM and FCRBM and it outperforms them in almost all cases.

Activities	CRBM	FCRBM	FFW-CRBM
TA1	0.050 ± 0.002	**0.032 ± 0.006**	0.043 ± 0.005
TA2	0.042 ± 0.006	0.046 ± 0.003	**0.022 ± 0.003**
TA3	0.070 ± 0.003	0.048 ± 0.001	**0.041 ± 0.002**
TA4	0.058 ± 0.001	0.051 ± 0.003	**0.032 ± 0.006**
TA5	0.082 ± 0.004	0.070 ± 0.004	**0.018 ± 0.005**
TA6	0.063 ± 0.002	0.044 ± 0.003	**0.031 ± 0.002**

One-step prediction: Results of Table 5 show that FFW-CRBM outperforms FCRBMs and CRBMs where, for example, FFW-CRBM achieves a minimum reconstruction error of 0.018 compared to 0.070 and 0.082 on TA4.

Multi-step prediction: The same experiments were performed for multi-step predictions as in the exercise activities case. Prediction error results are summarised in Table 6. Also here the proposed method outperforms the state-of-the-art techniques in multi-step predictions.

6.1.6. Self auto evaluation on TA

The SAE procedure was again applied but using TA dataset. In this case, Fig. 6 confirms the previous results in which, when the classification is wrong, the prediction error increases substantially, most clearly illustrated for TA5.

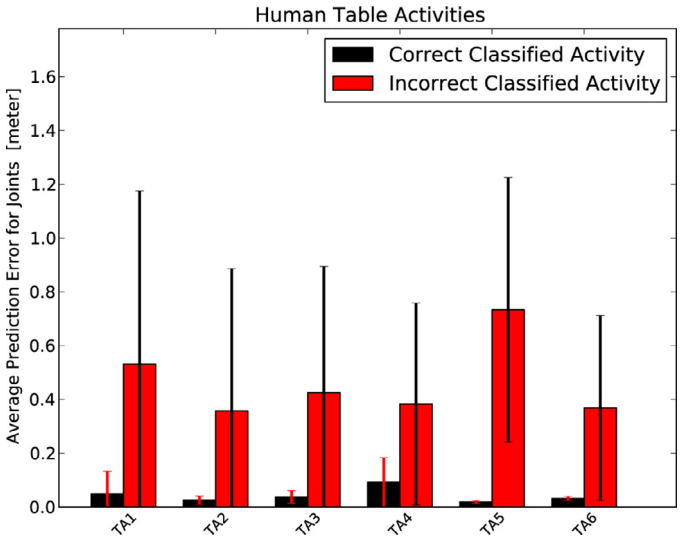

Fig. 6. Self auto evaluation results of FFW-CRBM in the case of human table activities. For any activity classified incorrectly, the red histogram show the mean of the prediction error, and the top black lines represent the standard deviations. The black histograms show the same, but in the case of correct classified instances.

6.2. Berkeley multimodal human action database

In order to benchmark the classification accuracy of FFW-CRBM method, it was tested on the Berkeley Multimodal Human Action Database (MHAD) [18] benchmark. Given the complex nature of this dataset, such an experiment can better judge the classification robustness of the proposed method. This dataset contains 11 activities performed by 12 persons. The difficulty in this experiment is that the training and test data come from different distributions.

Data from the optical mocap containing 93 dimensions was used. The original data contained around 400 frames per second. To speed up the learning process the data was split in temporal windows, each represented by the mean of 20 original frames. The layers of the FFW-CRBM were set to 93 neurons in the visible layer, 11 neurons in the

Table 6
Multi-step prediction results of CRBM, FCRBM and FFW-CRBM on table activities. The table represents the minimum, mean and maximum incurred errors over all activities at different numbers of prediction steps with mean (μ) and standard deviation (σ). (For interpretation of the references to colour in this table legend, the reader is referred to the web version of this article.)

Prediction		Minimum			Mean			Maximum		
		CRBM	FCRBM	FFW-CRBM	CRBM	FCRBM	FFW-CRBM	CRBM	FCRBM	FFW-CRBM
10 steps	μ	0.041	0.036	**0.029**	0.045	0.028	**0.027**	0.055	0.045	**0.038**
	σ	0.004	0.005	0.003	0.004	0.007	0.004	0.002	0.006	0.003
20 steps	μ	0.045	0.032	**0.028**	0.047	0.032	**0.029**	0.051	0.047	**0.039**
	σ	0.002	0.015	0.006	0.008	0.007	0.004	0.005	0.007	0.003
30 steps	μ	0.041	0.036	**0.028**	0.043	0.039	**0.034**	0.052	0.045	**0.041**
	σ	0.003	0.006	0.003	0.004	0.005	0.008	0.003	0.004	0.004
40 steps	μ	0.042	0.038	**0.027**	0.046	0.041	**0.033**	0.049	0.047	**0.041**
	σ	0.003	0.009	0.004	0.004	0.006	0.003	0.004	0.005	0.004
50 steps	μ	0.044	0.036	**0.030**	0.048	0.039	**0.034**	0.051	0.048	**0.042**
	σ	0.004	0.009	0.003	0.003	0.005	0.006	0.003	0.009	0.006

Table 7
Classification accuracy on MHAD database.

Model	Accuracy (%)
1-NN classifier [18]	74.82
3-NN classifier [18]	75.55
K-SVM [18]	79.93
FFW-CRBM	**81.12** ± **1.3**

labelled layer, and 1860 neurons in the history layer. The number of hidden neurons was set to 10 with 10 factors. The initial number of the Markov chain steps in the training phase was set to 10. The initialisation of the weights was set to $\mathcal{N}(0, 0.3)$. The model was trained for 13 iterations.

The same classification scenario was followed as in [18], in which the first 7 users were used to train the model and the last 5 users were used for testing. To assess the stochastic nature of FFW-CRBM and to compensate the absence of *k*-fold cross-validation technique in the original classification scenario, the training/testing procedures were repeated 10 times. As depicted in Table 7, the accuracy of FFW-CRBM is higher than the accuracies reported in the original paper.

7. Conclusions and future work

In this paper, a new machine learning technique for activity recognition and prediction is proposed. Factored four way conditional restricted Boltzmann machines, together with an adapted training algorithm SMcCD are capable of: (1) classification, (2) prediction, and (3) self auto evaluation of their classification performance within one unified framework. The efficacy and performance of FFW-CRBM has been demonstrated on real-world data acquired from our previously developed robotic platform for smart companions and on benchmark datasets. Results showed that FFW-CRBMs are capable of outperforming current state-of-the-art machine learning algorithms in both classification and regression.

Even though FFW-CRBMs are successful, the choice of their parameters such as the number of hidden units, the learning rate or the number of factors might be troublesome, as is the case for other machine learning algorithms. Furthermore, computational complexity in case of a very large number of the past frames might be potentially a problem. Both drawbacks present opportunities for future explorations.

Acknowledgements

This research has been partly funded by the FP7 European project Florence under grant ICT-2009-248730 (Multi Purpose Mobile Robot for Ambient Assisted Living).

References

[1] D. Lowet, M. Isken, W.P. Lee, F. van Heesch, H. Eertink, Robotic telepresence for 24/07 remote assistance to elderly at home, workshop on social robotic telepresence, in: Ro-Man 2012, 21st IEEE International Symposium on Robot and Human Interactive Communication, 2012.
[2] A. Manzanares, L. Martinez, M. Isken, D. Lowet, A. Remazeilles, User studies of a mobile assistance robot for supporting elderly: methodology and results, in: Workshop at IROS 2012 on Assistance and Service Robotics in a Human Environment, At Vila Moura, Portugal, 2012.
[3] S. Zhang, M.H. Ang, W. Xiao, C.K. Tham, Detection of activities for daily life surveillance: Eating and drinking, in: HealthCom 2008 – 10th International Conference on e-Health Networking, Applications and Services (2008) 171–176. doi:10.1109/HEALTH.2008.4600131
[4] L. Chen, J. Hoey, C. Nugent, D. Cook, Z. Yu, Sensor-based activity recognition, IEEE Syst., Man, Cybernet., Part C: Applications and Reviews, IEEE Transactions on 42 (6) (2012) 790–808.
[5] Y. Bengio, Learning deep architectures for ai, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.
[6] N. Jones, Computer science: The learning machines, Nature 505 (7482) (2014) 146–148.
[7] J. Laserson, From neural networks to deep learning: zeroing in on the human brain, ACM Crossroads 18 (1) (2011) 29–34.
[8] H. Larochelle, Y. Bengio, Classification using discriminative restricted Boltzmann machines 2008, pp. 536–543.
[9] R. Salakhutdinov, A. Mnih, G. Hinton, Restricted boltzmann machines for collaborative filtering, in: In Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2004). ACM, AAAI Press, 2007, pp. 791–798.
[10] P.V. Gehler, A.D. Holub, M. Welling, The rate adapting poisson model for information retrieval and object recognition, in: Proceedings of 23rd International Conference on Machine Learning (ICML'06, ACM Press, 2006, p. 2006.
[11] P. Smolensky, Information processing in dynamical systems: foundations of harmony theory, in: D.E. Rumelhart, J.L. McClelland, et al. (Eds.), Parallel Distributed Processing. vol. 1: Foundations, MIT Press, Cambridge, 1987, pp. 194–281.
[12] I. Sutskever, G.E. Hinton, Learning multilevel distributed representations for high-dimensional sequences, in: M. Meila, X. Shen (Eds.), AISTATS. vol. 2: JMLR Proceedings, JMLR.org, 2007, pp. 548–555.
[13] G.W. Taylor, G.E. Hinton, Factored conditional restricted Boltzmann machines for modeling motion style, in: Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, 2009, pp. 1025–1032.
[14] G.W. Taylor, G.E. Hinton, S.T. Roweis, Two distributed-state models for generating high-dimensional time series, J. Mach. Learn. Res. 12 (2011) 1025–1068.
[15] S. Yu, H. Yang, H. Nakahara, G.S. Santos, D. Nikolić, D. Plenz, Higher-order interactions characterized in cortical activity, J. Neurosci. 31 (48) (2011) 17514–17526.
[16] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (8) (2002) 1771–1800.
[17] D. Lowet, F. van Heesch, Florence – a multipurpose robotic platform to support elderly at home, in: Workshop on Ambient Intelligence Infrastructures (WAmIi), Pisa, Italy, 2012.
[18] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy., Berkeley mhad: a comprehensive multimodal human action database, in: Proceedings of the IEEE Workshop on Applications on Computer Vision (WACV), 2013.
[19] R. Memisevic, G.E. Hinton, Learning to represent spatial transformations with factored higher-order Boltzmann machines, Neural Comput. 22 (6) (2010) 1473–1492.
[20] G. Hinton, A Practical Guide to Training Restricted Boltzmann Machines, Technical Report, 2010.
[21] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[22] C.C. Chang, C.J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27.