Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A SPARSE ENCODING SYMMETRIC MACHINES PRE-TRAINING FOR TEMPORAL DEEP BELIEF NETWORKS FOR MOTION ANALYSIS AND SYNTHESIS

[1]MILYUN NI'MA SHOUMI, [2]MOHAMAD IVAN FANANY
Faculty of Computer Science, University of Indonesia, Depok, West Java, Indonesia
E-mail: [1]milyun.nima@ui.ac.id, [2]ivan@cs.ui.ac.id

ABSTRACT

We present a modified Temporal Deep Belief Networks (TDBN) for human motion analysis and synthesis by incorporating Sparse Encoding Symmetric Machines (SESM) improvement on its pre-training. SESM consisted of two important terms: regularization and sparsity. In this paper, we measure the effect of these two terms on the smoothness of synthesized (or generated) motion. The smoothness is measured as the standard deviation of five bones movements with three motion transitions. We also address how these two terms influence the free energy and reconstruction error profiles during pre-training of the Restricted Boltzmann Machines (RBM) layers and the Conditional RBM (CRBM) layers. For this purpose, we compare gait transitions by bifurcation experiments using four different TDBN settings: original TDBN; modified-TDBN(R): a TDBN with only regularization constraint; modified-TDBN(S): a TDBN with only sparsity constraint; and modified-TDBN(R+S): a TDBN with regularization plus sparsity constraints. These experiments shows that the modified-TDBN(R+S) reaches lower energy faster in RBM pre-training and reach lower reconstruction error in the CRBM training. Even though the smoothness of the synthesized motion from the modified-TDBN approaches is slightly less smooth than the original TDBN, they are more responsive to the action command to change a motion (from run to walk or vice versa) while preserving the smoothness during motion transitions without incurring much overhead computation time.

Keywords: *Temporal Deep Belief Network (TDBN), Sparse Encoding Symmetric Machines (SESM), Restricted Boltzmann Machine (RBM), Conditional RBM (CRBM)*

1. INTRODUCTION

The advent of motion capture technology has enabled deeper analysis and synthesis not only of human motion and action, but also general human activity and even behavior. Human activity recognition and reconstruction is one of the most active research in the field of computer vision, computer animation, computer graphics, and human computer interface. Its application is covering vast area of fields including medical, security in a public area, industrial film, games development, and education. In one review [1], methodologies used by previous researches to detect human activity can be classified into two categories: a single-layer approach and a hierarchical approach. Recent developments indicate that the hierarchical approach can provide a significant improvement compared to the single-layer approach.

Deep model is one of the hierarchical approaches, where this model can learn multiple-layer of features hierarchically and can automatically generate multi-level representation of input data. Layer at the bottom detects simple features and provide the result to layer above, which is subsequently used to detect more complex features. In addition, one of the main advantage of deep learning framework is the recognition process for action analysis and the reconstruction process for action synthesis can be performed in a single unit. Deep Belief Networks (DBNs) [2] is a deep model with several layers that have been successfully applied for a variety of machine learning problems, such as handwritten digit recognition [2] and also human motion recognition from motion capture data [3]. The DBNs architecture is composed from several Restricted Boltzmann Machine (RBM), i.e., a neural network with two layers: visible and hidden layers. RBM was initially limited for statistical pattern, like an image, but [3] shows that RBM can be modified to learn motion data by adding an additional layer to keep the past state of the visible layer, which is called Conditional RBM (CRBM).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Sukhbaatar et al. [4] proposed a system that utilizes RBM and CRBM in a single architecture for motion analysis and synthesis which is called as Temporal Deep Belief Network (TDBN). TDBN is a DBN with two layers, where the first layer consists of 5 RBM that represent the body part, and the second layer consists of single CRBM.

One of the shortcoming of RBM as the main component of TDBN is their objective function cannot be maximized directly, so there is no obvious function to monitor and decide when to stop the training process, and required a mechanism for adjust the computational time as well as the amount of memory required for computation [5]. Thus, the process is inefficient and weights as a result of training is not optimal. Sparse Encoding Symmetric Machine (SESM) is intended to deal with this problem by inserting a sparsity penalty into the Contrastive Divergence (CD) method in RBM. SESM is an alternative method that can be used for pre-training, where SESM allows the objective function to be optimized directly and it can be used as an indicator to stop the training process. SESM was first implemented for hand-written digit recognition [6], while [5] improved the SESM algorithm and implement it on speech recognition.

In this paper, we evaluate an analysis and synthesis of motion capture data by the modified TDBN. The Modification is conducted by adding regularization and sparsity penalty to RBM during the pre-training process of TDBN. Our experiments show that the addition of regularization and sparsity constraints can improve performance and provide a smooth transition between motion (walking and running) on the synthesis result.

The reminder of this paper is organized as follows: In Section 2 we briefly review some previous works on analysis and synthesis of human motion using a hierarchical approach. In Section 3, we describes the methods used to construct modified TDBN system. Details of data set, implementation, experiments, and also evaluation results is described briefly in Section 4.

2. RELATED WORKS

Numerous research on analysis and synthesis of human motion using a hierarchical approach with deep model has been carried out in recent years, and it is estimated that the number of research in this area will continuously increases. Song et al. [7] developed a model of Hierarchical Sequence Summarization (HSS) using Conditional Random Fields (CRF) with a set of hidden variables to learn the sequential data and modify the function of the standard features to learn the representation of different features automatically. Baccouche et al. [8] developed a model for classifying human actions without the use of prior knowledge. The scheme is an extension of Convolutional Neural Network (CNN) that automatically learn the spatiotemporal features, and Recurrent Neural Networks (RNN) which are trained to classify each sequence. Le et al. [9] propose a model, which is named Space-Time Deep Belief Network (ST-DBN) using Convolutional Restricted Boltzmann Machine (CRBM) as a basis for processing. While Wolf et al. [10] developed a Conditional Deep Belief Networks (CDBNs) which consists of the arrangement of Conditional RBM (CRBM) with the purpose of classifying human activities on very short sequences. Feature generated by CDBN subsequently be classified using a Support Vector Machine (SVM).

Our research is inspired by a research paper of Sukhbaatar et al. [4], where the goal is to re-construct human motions using two-layer DBN which is called Temporal Deep Belief Networks (TDBN). The first layer consists of multiple RBM which is used to encode the spatial pattern of motion frames into a compact representation. The number of RBM in the first layer according to the amount of different body parts. They used five RBM on the first layer, and each of them is used for one of the body part, that is right arm, left arm, right leg, left leg, and trunk. So, RBM that corresponding to the right foot will only be used to train any movement on the right food, as well as four other RBM. While the layer above is one Conditional RBM (CRBM), which are used to learn the temporal constraint contained in the transition of compact representation that is generated in the previous layer.

3. METHODS

In this paper, we modify Temporal Deep Belief Network (TDBN) method that has been developed by Sukhbaatar [4]. TDBN is a Deep Belief Network (DBN) with two layers that consist of: (1) Five RBMs in the first layer where each represents 5 groups of body parts (right hand, left hand, right foot, left foot, trunk); (2) A CRBM in the second layer will accept the result of training in the first layer as input. The RBMs are trained using Contrastive Divergence (CD) that utilizes Gibbs sampling algorithm. Hidden layer on the first layer is used as an input or visible layer on the second

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

layer. The second layer serves to unify the training result on the first layer, as well as to control the synchronization of the body parts that previously separated. CRBM on the second layer is also trained using CD algorithm as RBM. The illustration of TDBN architecture can be seen in Figure 1. The system used in this study is an extension of TDBN where RBM in the first layer added with regularization and sparsity penalty. Modification that applied to TDBN system can provide better reconstruction result

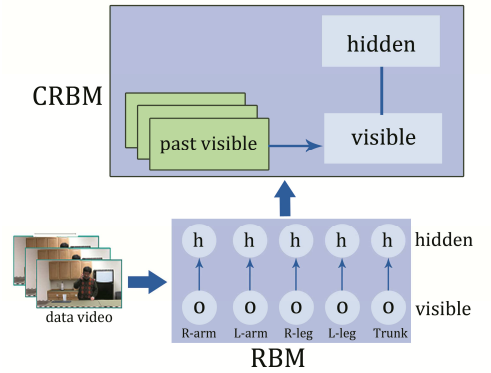

Figure 1: TDBN architecture with two layer

Restricted Boltzmann Machine (RBM)

Restricted Boltzmann Machine (RBM) is a type of Boltzmann Machine that has a special structure, in which its units are divided into visible layer and hidden layer [11]. Each unit on the visible layer are connected with hidden layer units, but each unit in the same layer have no connection with each other. This kind of bi-partite structure ensures that units in the hidden layer are conditionally independent to the units in the visible layer, and vice versa. The simple structure and inference accuracy are the main advantages of the use of RBM compared to conventional Boltzmann Machine, which has all it sunits are connected. Comparison of the structure between RBM and Boltzmann Machine can be seen in Figure 2.

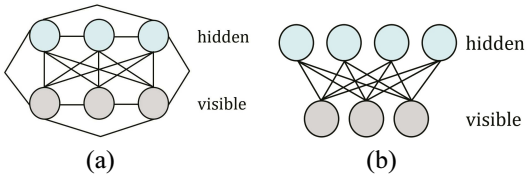

Figure 2: Structure of (a) Boltzmann Machine and (b) RBM

In RBM, the probability value for each joint of visible unit *v* and hidden unit *h* defined as follows:

$$p(v,h) = \frac{\exp(-E(v,h))}{Z} \quad (1)$$

where $E(v,h)$ is a function of energy and Z is a normalization constant or commonly called partition function, which is used to calculate the number of possible joint with equation $Z = -\sum_{v',h'} E(v',h')$. If the visible unit and hidden unit are binary with value of 1 or 0, the energy function can be written as follows:

$$E(v,h) = -\sum_{ij} w_{ij} v_i h_j - \sum_i a_i v_i - \sum_j b_j h_j \quad (2)$$

where v_i is a node on the visible unit i, while h_j is a node on the hidden unit j. The a_i is bias on the visible unit, while b_j is bias on the hidden unit. The w_{ij} is the weight of edge that connects v_i and h_j. To make a visible unit with real-value, the energy function need to be transformed into

$$E(v,h) = \frac{1}{2}\sum_i v_i^2 - \sum_{ij} w_{ij} v_i h_j - \sum_i a_i v_i - \sum_j b_j h_j \quad (3)$$

Conditional Restricted Boltzmann Machine (CRBM)

Initially, RBM can only model the static frame in data (such as images) and does not include temporal information thus cannot models sequential data. Conditional RBM (CRBM) that was introduced by Taylor et al. [3] is an extension model of RBM that can learn transition on time-series data. The general concept of CRBM is combining temporal information in the data. Temporal information of data will be combined by adding a layer called past visible layer, and two types of directed connections: (1) *Autoregressive connection* from *n* time frame of previous visible unit to the current visible unit. (2) *Directed connection* from *m* time frame of previous visible unit to the current hidden unit. The structure of CRBM can be seen in Figure 3.

CRBM can be trained with the same training algorithm like RBM, because input from the past-visible unit can be handled in the same way as bias, but has different effect because of hidden units in CRBM are influenced by past-visible units. To reconstruct human motion using CRBM, the past-visible units and visible units must be real-valued because motion is represented as a real-valued joint-angles.

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

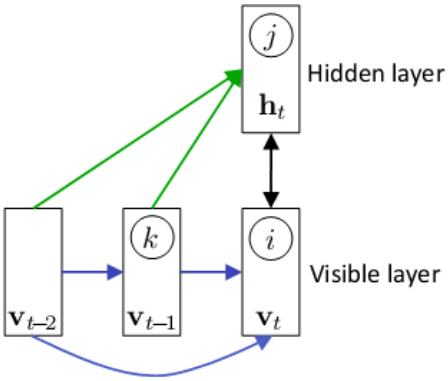

Figure 3: Structure of CRBM with n=2 [3]

Contrastive Divergence

An algorithm which commonly used to train RBM with the purpose to optimize the weight vector *W* is Contrastive Divergence (CD) which was developed by [12]. CD using Gibbs sampling algorithm and also implement gradient descent procedure to perform weight update calculation. Briefly, the procedure which performed one phase of CD algorithm (CD-1) in the one sample can be written as follows:

1. For the training sample *v*, calculate the probability of hidden unit and sampling hidden activation vector *h* from probability distribution of calculation result.
2. Calculate the outer product of *v* and *h*. The result of this calculation called positive gradient.
3. From *h*, sampling on the reconstruction result of visible unit v'. From this sampling result, resampling on hidden activation *h'*.
4. Calculate the outer product of *v'* and *h'*. The result of this calculation called negative gradient.

The weights W_{ij} are updated by the difference value between positive gradient and negative gradient. Illustration of the training process using CD algorithm can be seen in Figure 4.

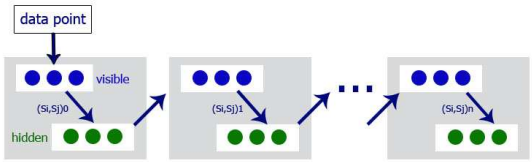

Figure 4: Training process using CD

Sparse Encoding Symmetric Machine (SESM)

Training the weights by greedily propagating back and forth from one layer to the next layer is an effective and efficient method for determining the initial weights in a neural network, rather than just initializing the weights with random values. Such greedy training is commonly called pre-training. In the pre-training phase, unsupervised model is defined as a distribution of input vector *Y* on energy function *E(Y,Z,W)* as follows:

$$P(Y \mid W) = \int_z P(Y, z \mid W) = \frac{\int_z e^{-\beta E(Y,z,W)}}{\int_{y,z} e^{-\beta E(y,z,W)}} \quad (4)$$

where β is a constant variable. Weight matrix *W* will be updated during the training to obtain optimal *Z* value. The equation to calculate weights that can minimize the loss function, which is equal to the negative log likelihood on the training data:

$$L(W,Y) = -\frac{1}{\beta}\int_z e^{-\beta E(Y,z,W)} + -\frac{1}{\beta}\int_{y,z} e^{-\beta E(y,z,W)} \quad (5)$$

The first term is free energy and the second term is called log partition function. The log partition function ensures that we observed low energy for the (true) data distribution and high values anywhere else [5]. Two approaches can be used to estimate the log partition function. RBM uses the concept of Contrastive Divergence (CD) to obtain the log partition function, whereas SESM replacing the log partition by adding sparsity constraint on the prediction of the output of neural network. Sparseness in SESM allows for the optimization of the objective function directly.

The training of SESMs is performed by simply minimizing the average energy in combination with the additional sparseness term of the output. Similar to the RBM, SESM follows the encoder-decoder paradigm. The encoder and decoder are described by

$$f_{enc}(Y) = W^T Y + b_{enc}, f_{dec}(Z) = Wl(Z) + b_{dec} \quad (6)$$

where *l* function is a logistic non-linearity $l(x) = 1/(1+\exp(-gx))$ with gain *g* = *1* for all experiments. While the energy function of SESM can be described as follows:

$$E(Y,Z) = \alpha_e \| Z - f_{enc}(Y) \|^2 + \| Y - f_{dec}(Z) \|^2 \quad (7)$$

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Overall, the optimization of the loss function during SESM training can be calculated using:

$$L(W) = E(Y,Z) + \alpha_s h(Z) + \alpha_r |W|_1$$
$$= \alpha_e \| Z - f_{enc}(Y) \|_2^2 + |Y - f_{dec}(Z)|_2^2 + \alpha_s h(Z) + \alpha_r |W|_1 \quad (8)$$

where $h(Z) = \sum_d \log(1 + l^2(z_d))$. The loss contains the free energy (Eqn. 6), a sparseness term (h(Z)) as an approximation to the log partition function and a l1-regularization term on the weights. Instead of sampling the output as for RBM, SESM uses the output of the encoder directly. In this paper, however, we use l2 regularization.

4. EXPERIMENTS AND EVALUATIONS

Data used in this study is an Acclaim format, with two types of file: (1) Acclaim Skeleton File (ASF) which is used to store skeleton information and (2) Acclaim Motion Capture (AMC) which is used to store motion information. Overall data derived from CMU Graphics Lab Motion Capture Database (http://mocap.cs.cmu.edu). Our experiment is performed on 8x Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz processor with 32909MB (13668MB used) memory and Ubuntu 14.04.1 LTS operating syste

We perform gait transitions by bifurcation experiment similar to Sukhbaatar et al., [4]. In our experiments, however, we test and compare four different TDBN schemes: original TDBN, modified-TDBN(R), modified-TDBN(S), and modified-TDBN(R+S). We train the model on two different gait styles: walking and running. The TDBN learns and generates two different motions by a single TDBN and generates a natural gait transition motion. The walking and running frames have to be learned in a random order to prevent a bias to one gait. We give the motion change command at the 100-th and 200-th frames for all four TDBNs are more responsive to the action command to change a motion (from run to walk or vice versa).

Our TDBN parameter settings are as follows. The number of hidden nodes in RBMs is 30, whereas in the CRBMs we set 15 hidden nodes. According to [4] more units boost the effect of the gait parameter on the hidden layer, making the transition more sudden and short. On the contrary, fewer units will increase the effect of the past hidden states on the hidden layer, making transition slower and even impossible in some cases. However, in our experiments, we observed that the number of hidden nodes (30 for RBMs, and 15 for CRBM) gives visually natural transitions. According to the epoch that is used by [4], RBMs in the first layer is trained with 200 epochs, and the CRBMs is trained with 40 epochs. The regularization and sparsity constants are respectively set to 1.0×10^{-5} and 2.0×10^{-7}. The smoothness of gait transition is measured by statistical evaluation on the variation of the rotation angles of five bones movement (root, left hand, right hand, left foot, and right foot) as shown in Table 1. The smoothness differences between the modified TDBN and the original TDBN is relatively small.

Table 1: Bone motions smoothness measure as standard deviation of points in bones trajectory (the smaller is the better).

Bone Name	Original TDBN	Mod-TDBN (R)	Mod-TDBN (S)	Mod-TDBN (R+S)
Root RX	0.00597	0.00797	0.00866	0.02491
Root RY	0.01175	0.01406	0.01342	0.01455
Root RZ	0.00925	0.01273	0.01264	0.01371
lhand RX	0.00264	0.04546	0.00413	0.00437
lhand RZ	0.01289	0.04685	0.01786	0.02942
rhand RX	0.00441	0.04024	0.00518	0.03373
rhand RZ	0.01045	0.04051	0.01318	0.06286
lfoot RX	0.06138	0.07779	0.07644	0.07626
lfoot RZ	0.03010	0.03804	0.03368	0.04423
rfoot RX	0.06288	0.07421	0.07306	0.08200
rfoot RZ	0.02548	0.03183	0.03139	0.02947
mean	0.02313	0.04217	0.02809	0.03906

As a result preview, the comparison between the reconstruction result of original TDBN and modified-TDBN for walking and running motions, along with the transition between those motions can be seen in Figure 5. Gait transitions from the four TDBN schemes show that the modified TDBN is more responsive that the original TDBN while preserving the smoothness between motions transitions. Figure 6 shows that the free energy during training of RBMs, which is real-binary pair layers, in the modified TDBNs get lower faster than the RBM in the original TDBN, whereas the reconstruction error are about the same. Figure 7 shows that the energy of CRBM, which is a binary-binary layer, in the modified TDBN diverge but

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

attain lower reconstruction error than the original TDBN.

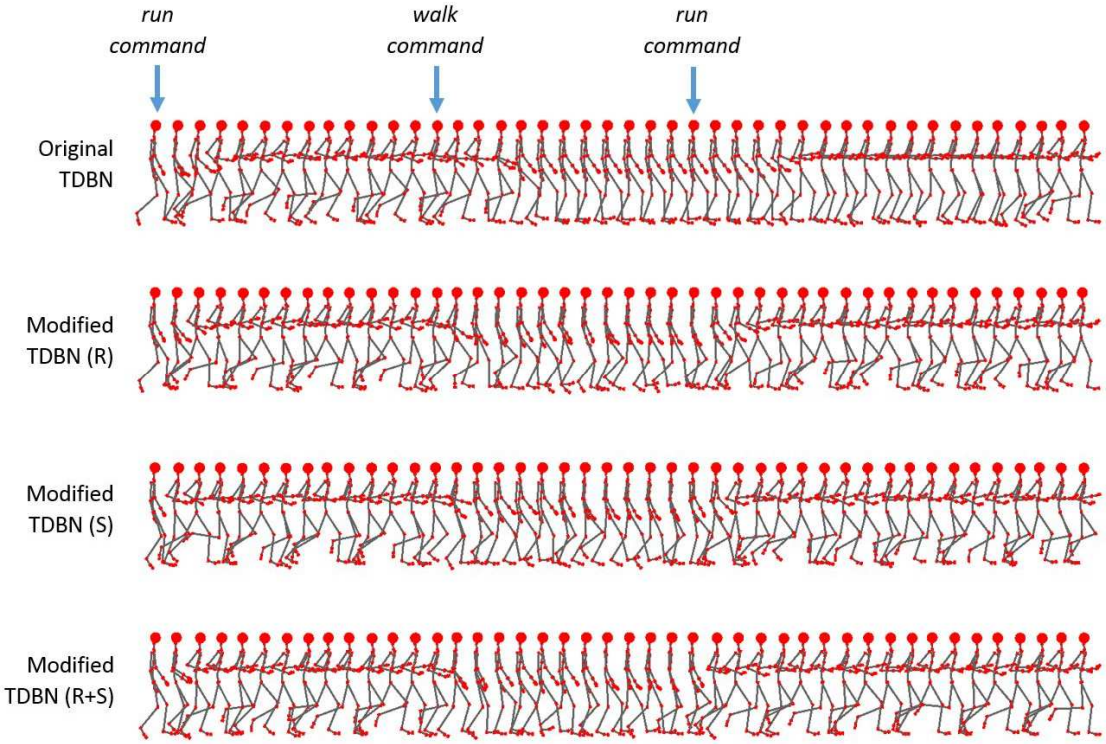

Figure 5: Reconstructed motions

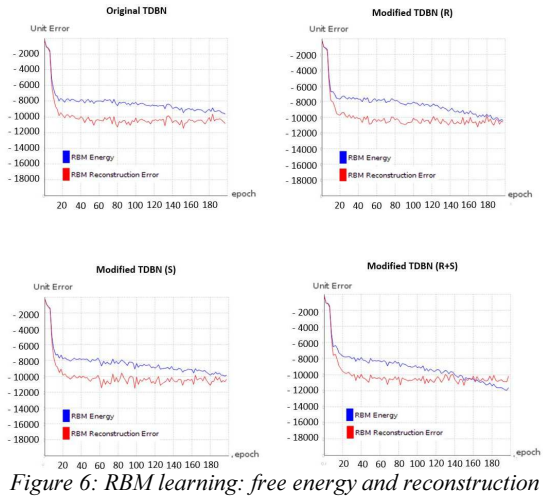

Figure 6: RBM learning: free energy and reconstruction error

Table 2 shows that the modified TDBN learning does not incur much additional total computation time. Figure 8 shows the transition of the right-hand bone from walk to run of the rotation angle with respect to Z axis. The transition profile of the modified TDBN is comparably similar to the original TDBN.

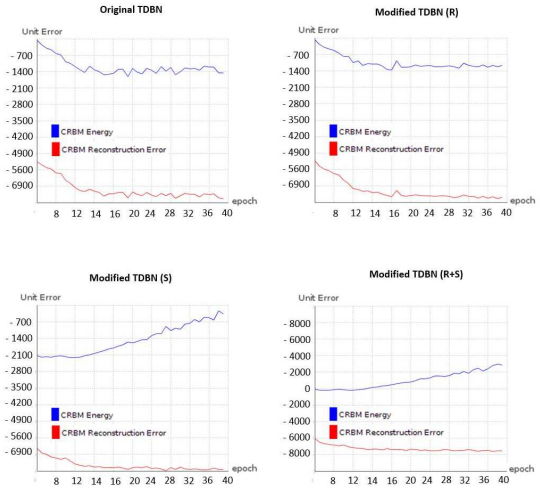

Figure 7: CRBM learning: free energy and reconstruction error

5. CONCLUSION

This paper presents an application of SESM pre-training into TDBN for human motion analysis and synthesis. The application is SESM includes the regularization and sparsity constraints into the CD

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

algorithm inside RBM and CRBM of a single TDBN. We compare gait transitions by bifurcation experiments using four different TDBN settings.

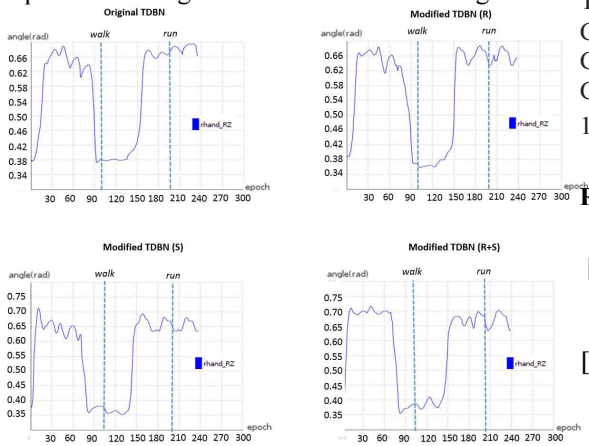

Figure 8: The smoothness of transition of rhand RZ rotation

These experiments show that the modified-TDBN reaches lower energy faster in RBM pre-training and reach lower reconstruction error in the CRBM training. Even though the synthesized motions from the modified-TDBN approaches is slightly less smooth than the original TDBN, the modified-TDBN is more responsive to the action command to change a motion (from run to walk or vice versa) while preserving the smoothness during motion transitions without incurring much overhead computation time. For our future works, we plan to try another types of motion data such as dances and sport movement. We also plan to use the auto-encoder method in the first layer of TDBN to replace RBM.

Table 2: TDBN schemes computation time

training	Original TDBN	Mod-TDBN (R)	Mod-TDBN (S)	Mod-TDBN (R+S)
RBM	39.696 s	40.076s	40.962s	40.941s
CRBM	1m 29.983s	1m 28.443s	1m 28.041	1m 30.45s
Total	2m 9.679s	2m 8.519s	2m 9.003s	2m 11.391s

Table 3: TDBN schemes memory usage (in MB)

training	Original TDBN	Mod-TDBN (R)	Mod-TDBN (S)	Mod-TDBN (R+S)
RBM	119.4	114.1	94.3	116.4
CRBM	81.2	79.1	60.8	78.6
Total	200.6	193.2	155.0	195.1

ACKNOWLEDGEMENT

This work is supported by International Collaboration Grant from Directorate Research and Community Engagement Universitas Indonesia Contact No.
1402/H2.R12/HKP.05.00/Perjanjian/2014

REFRENCES:

[1] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A Review", ACM Computing Surveys (CSUR), 43(3): 16, 2011.

[2] G.E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for Deep Belief Nets", Neural Computation, 18, pp. 1527-1554, 2006.

[3] G. W. Taylor, G. E. Hinton, and S. Rowies, "Modeling human motion using binary latent variables", Adv. In Neural Information Processing Systems, 2007.

[4] S. Sukhbaatar, T. Makino, K. Aihara, and T. Chikayama, "Robust generation of dynamical patterns in human motion by Deep Belief Nets", Journal of Machine Learning Research (JMLR): Workshop and Conference Proceedings 20, 231-246, 2011.

[5] C. Plahl, T.N. Sainath, B. Ramabhadran, and D. Nahamoo, "Improved Pre-Training of Deep Belief Networks Using Sparse Encoding Symmetric Machines", IEEE, ICASSP, 2012.

[6] M.A. Ranzato, Y.L. Boureau, Y. LeCun, "Sparse Feature Learning for Deep Belief Networks", Advances in Neural Information Processing Systems (NIPS 2007), 2007.

[7] Y. Song, L.P. Morency, and R. Davis, "Action recognition by hierarchical sequence summarization", In Computer Vision and Pattern Recognition (CVPR 2013), Portland, OR, 2013.

[8] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition", 2nd International Workshop on Human Behavior Understanding (HBU), Lecture Notes in Computer Science, pp. 29–39, Springer 2011.

[9] Q.V. Le, W.Y. Zou, S.Y. Yeung, and Andrew Y.Ng., "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis", In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '11, pp. 3361–3368, Washington, DC, USA, 2011.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[10] C. Wolf, G.W. Taylor and J.M. Jolion, "Learning individual human activities from short binary shape sequences", Technical Report RR-LIRIS-2011-018 Laboratoire d'Informatique en Images et Systèmes d'Information (LIRIS), INSA de Lyon, France, 2011.

[11] P. Smolensky. 1986. "Information Processing in Dynamical Systems: Foundations of Harmony Theory". In D. E. Rumelhart, J. L. McClelland, et al., editors, Parallel Distributed Processing: Volume 1: Foundations, pages 194-281. MIT Press, Cambridge, MA.

[12] G. E. Hinton. 2002. Training Products of Experts by Minimizing Contrastive Divergences. Neural Computation, 14(8): 1771-1800.