Temporal Deep Belief Network for Online Human Motion Recognition

François Lasson, Mihai Polceanu, Cédric Buche, Pierre De Loor
LAB-STICC, ENIB, France
{lasson, polceanu, buche, deloor}@enib.fr

Abstract

Interaction between humans and machines, like social robots, requires real time recognition of human actions. Most approaches to this problem wait for the end of the gesture to perform classification. In this paper we present a deep learning approach to online gesture recognition that allows for an estimation of the current gesture since its beginning. Our approach is to modify the existing Temporal Deep Belief Network (TDBN) architecture. The result is a Discriminative Temporal Deep Belief Network (DTDBN) which we apply to the online classification of motion capture streams. We optimize and evaluate our model in comparison with related work.

Introduction

Real time human action recognition is an increasingly important capability in social robots. Extensive research has been done in this direction (for detailed review see (Moeslund and Granum 2001; Aggarwal and Ryoo 2011)), but not many focus on predicting the action before the gesture has finished. However, such capabilities could be crucial for real time interactive applications for which decisions must be taken as fast as possible and can benefit from the anticipation of the situation. The problem of sequential pattern recognition can be seen as a classification problem and is often dealt with through Hidden Markov Models (HMM), Support Vector Machines (SVM) or Dynamic Time Warping (DTW). However, DWT is known to perform slowly when the gesture database increases in size (Keogh and Pazzani 1999). SVMs seem to provide faster recognition than HMMs (Chen et al. 2015), but these models are still relatively hard to train on high dimensional data. Hence, we turn to more scalable models that have been proven to handle high dimensional input efficiently, namely deep learning. The goal of our work is to recognize the gesture as fast as possible, through this kind of approach, with a decent accuracy, instead of waiting for the end of the motion to obtain a result. An important point with deep learning is that, after the learning phase is finished, the recognition is very fast and is not dependent on the number of gesture classes to recognize.

There are two major families of deep architectures: Convolutional Neural Networks (CNN) and Deep Belief Networks (DBN). Because of their operating principle and their capacity to process large data, CNNs are very efficient for visual analysis and thus have wide applications in image and video recognition (Krizhevsky, Sutskever, and Hinton 2012). In our case, we are particularly interested in the DBNs which allow the recognition of static patterns in general. The architecture of a DBN consists in a stack of Restricted Boltzmann Machines (RBM), each of which contains a visible layer and a hidden layer that enables the extraction of increasingly more abstract features from the given data. The training algorithm for this network is the greedy layer-wise unsupervised algorithm (Hinton, Osindero, and Teh 2006), an iterative algorithm to train a DBN by successively applying the algorithm of contrastive divergence (CD) to each of the RBMs which compose it.

In this paper, we first discuss several extensions of the DBN architecture that enable pattern recognition in time series. We then present our approach which consists in modifying a Temporal Deep Belief Network (TDBN) (Sukhbaatar et al. 2011) to perform classification on motion capture data streams in real time. We optimize the parameters of the obtained model and evaluate its classification performance and recognition speed on two motion capture databases and compare the results to related approaches.

Related Work

Hidden Markov Models (HMM) are considered to be a state-of-the-art technology for the recognition of temporal patterns such as sound signals, despite the fact that they can not determine abstract concepts on input data. HMM based recognition systems also need a large amount of training data to obtain good recognition rates while approaches using Dynamic Time Warping (DTW) has low performance when the variance in the recorded motion is significant. For all these systems, the number of recognizable gestures is limited.

Deep belief networks are dedicated to the recognition of static patterns. It is difficult to use them to generate dynamic patterns, first because their hierarchical structure is not explicitly adaptable to temporal contexts, but also because the Restricted Boltzmann Machines used in the DBNs are not able to learn time constraints. However, there are extensions of DBNs that allow sequential patterns to be recognized, which perform better than Conditional Ran-

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

dom Fields (CRFs) (Lafferty, McCallum, and Pereira 2001). The latter are commonly used for temporal pattern recognition despite their inferior latent representations (Andrew and Bilmes 2012).

Hybrid architectures that combine DBNs with HMMs have been proposed, notably Back-Propagation DBN (BP-DBN) and Associative Memory DBN (AM-DBN) (Mohamed, Dahl, and Hinton 2009), for sound signal recognition. These two architectures show slight differences in their structures, but both have mechanisms to avoid overfitting. They are also pre-trained via the greedy layer-wise unsupervised algorithm.

Taylor et al. proposed an extension of the RBM that is able to learn gestures and therefore temporal data (Taylor, Hinton, and Roweis 2006). This new model is called Conditional RBM (CRBM). A CRBM is the addition of an extended visible layer to the RBM structure, called "past visible", that has the role of keeping the previous state of the visible layer. The connections between this new layer and the other layers of the model are unidirectional and correspond to the temporal constraints between two consecutive visible states. The training of a CRBM is identical to that of a conventional RBM, the neurons of the past-visible layer being comparable to biases. Despite their interest, the CRBMs have some shortcomings in our use case. Indeed, the capture motion corresponds to a set of angular data represented by continuous variables. As a result, the past-visible and visible layers are associated with continuous values when gestures are generated. The problem is that the connections between these two layers create an instability in the training of the CRBMs.

To deal with this issue, it is possible to evolve the CRBM architecture by adding a hierarchical structure. This idea introduces a new model, the Temporal DBN (TDBN) (Sukhbaatar et al. 2011). TDBNs are 2-layer DBNs in which the initial layer is composed of several RBMs in parallel, that can be seen as a large RBM with fairly scattered connections. This architecture makes TDBNs a robust and flexible model and avoids the problem of instability associated with training the CRBM directly on the data, since features are extracted by the RBMs and supplied to the CRBM which only focuses on the temporal aspect of the data. Hence, the top layer of the CRBM can be seen as an abstract control layer which can be used to impose a gesture to generate.

For our purpose – *i.e.* online human action recognition – the robustness shown by the TDBN architecture in gesture generation (Sukhbaatar et al. 2011) makes it a good candidate for extension to our classification problem. In the following section we describe our modifications to this model and the steps taken to optimize its performance.

Proposition: Discriminative TDBN

Online gesture recognition requires processing high dimensional temporal data, on which classic classification algorithms are inefficient. Sukhbaatar et al. propose a modification to the DBN architecture for human gesture generation (*i.e.* temporal pattern generation) (Sukhbaatar et al. 2011). Their approach consists in a Conditional Restricted Boltzmann Machine (CRBM) (Taylor, Hinton, and Roweis 2006) coupled with a set of RBMs, one for each body part (left/right arm/leg and trunk). The use of multiple RBMs enables considerably limiting the training data dimensions for each RBM, while the top CRBM layer is trained with abstract binary features obtained from the bottom RBMs and handles temporal dynamics. This spatial-temporal separation enables the generation and recognition of dynamic behaviors.

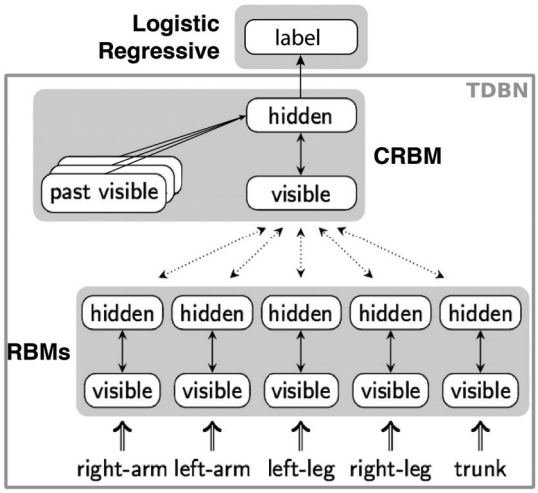


Figure 1: Illustration of the Discriminative Temporal Deep Belief Network architecture (DTDBN).

Although a TDBN control layer allows to impose the action to generate, it is associated with labels learned in an unsupervised manner, as the CRBM is only trained with the Contrastive Divergence algorithm. The model proposed by Sukhbaatar et al. does not include a supervised fine-tuning phase.

Because our goal is to perform gesture recognition rather than generation, we must add a layer of labels (logistic regression). The architecture shown in Figure 1 is hence obtained, which we have chosen to name Discriminative Temporal Deep Belief Network (DTDBN).

For the purpose of supervised learning, the TDBN architecture must be slightly changed. According to the literature and more precisely the work of Hinton (Hinton, Osindero, and Teh 2006), the most common way to obtain this type of learning is to add an extra layer at the top of the network, which would be trained with different labels in a supervised fine-tuning phase. In our case, this is a logistic regression layer, one of the most common linear classifiers for predicting the class of an input. The X input data belongs to the class Y for which the probability (1) is the highest.

$$\Pr(Y = i|X, W, b) = softmax(W * X + b) \qquad (1)$$

Optimization

For the purpose of exploring how the model parameters influence its performance, we used part of the INGREDIBLE database (Stankovic et al. 2013), namely the Fitness dataset

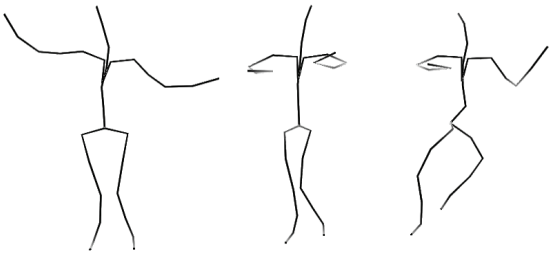


Figure 2: INGREDIBLE database gesture examples.

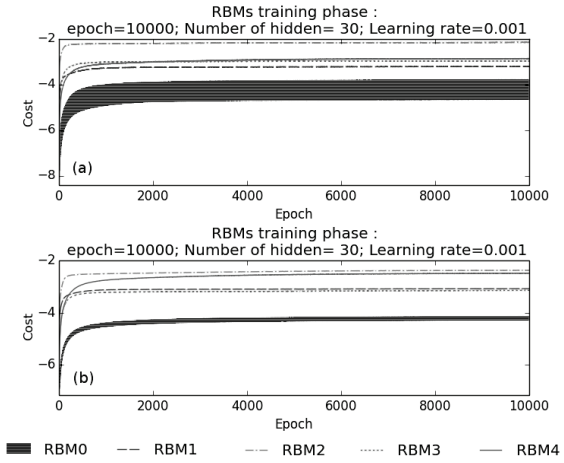


Figure 3: Plot of RBM training cost (negative log-likelihood) without (a) and with (b) data shuffling.

because it is more complex (more similarities between different gestures) and therefore much more of a challenge to perform classification. The data consists in a set of BVH files which contain the skeleton bone rotations at each frame (see Figure 2).

The proposed model is used to classify 9 actions (more details in the following section, table 4). For each of them, 4 samples are used for learning, 4 for validation and 2 for the test phase ($\sim$ 400 frames per sample). We perform two experiments to optimize the training cost of the RBMs and the time window for the CRBM.

In the first experiment we tested the importance of order of training data. The hypothesis was that learning the gestures one after the other could introduce bias and instability in learning. Because the weight matrix is updated by packets and not continuously, certain actions could potentially have greater importance than others. Therefore, we chose to train the RBMs and CRBM by shuffling all the frames of the training data. In the case of the CRBM, the ordered frame packets were mixed to maintain the temporal aspect and be able to update past-visible layers - *i.e.* the frame order is kept for each training packet, but the order of the packets is randomized. We have plotted the evolution of the RBM training cost without and with data shuffling (see Figure 3).

For the same training parameters and conditions, we notice a strong oscillation of the cost when no shuffling is performed, which signals weight matrix variations in the RBMs and thus a learning difficulty. As expected, the oscillation is greatly reduced with shuffling due to more uniformly represented gesture classes in each minibatch.

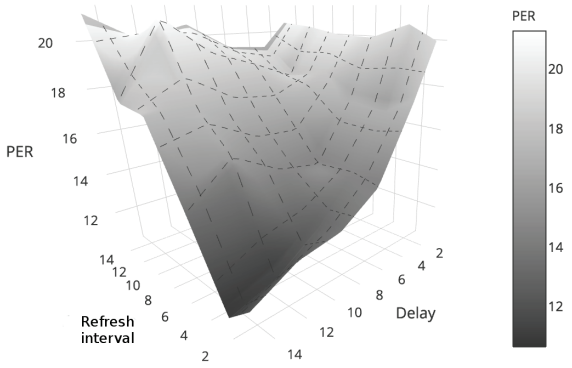


Figure 4: Plot of the influence of refresh interval and delay parameters on the PER.

In the second experiment, we wanted to test the benefits of a memory refresh rate, a principle used by (Sukhbaatar et al. 2011). The module of (Jost et al. 2015) operates at 30 frames per second (FPS), which corresponds to the frequency of perception of the human eye. Our module operates at 120 FPS due to the capture frequency of gestures present in the INGREDIBLE database, therefore the temporal influence is $(D+1)*R*1/120$ where D is the delay (number of past-visible layers) and R is the refresh interval (R-1 frames are skipped between each past-visible layer in order to increase the temporal influence). A CRBM that has 10 past-visible layers updated at 120 FPS without skipping frames has a temporal influence of $(10+1)*1*1/120 = 0.092s$. This time window seems too short to represent the evolution of a human gesture. The idea is then to set up a memory refresh interval to update every R frames, with R a natural number. The interest of this parameter is that it increases the time window of the CRBM without increasing the size of its architecture. As evidenced by the 3D graphics shown in Figure 4, the Prediction Error Rate ($PER = 1 - accuracy$) is low when the time window of the CRBM is around $(14+1)*2*1/120 = 0.250s$. Moreover, the model is more efficient for classification when the number of past-visible layers is high and the refresh rate is low. This result is consistent, a large number of past-visible layers means many synaptic connections with the discriminative layer and therefore has more influence on the latter.

It is also necessary to determine the influence of the number of neurons present in the hidden layer of the RBMs and the CRBM. We have varied these two parameters while observing the evolution of the error rate. According to the graph presented in Figure 5, the results in classification performance are optimal for RBMs having 30 and CRBM having 300 neurons in the hidden layer.

Regarding the learning rates associated with the different training algorithms, we have varied them between $1e{-}1$, $1e{-}2$ and $1e{-}3$ and observed their impact on the error rate.

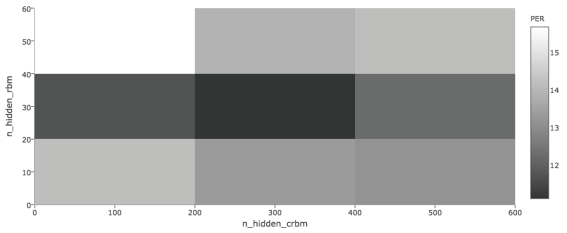


Figure 5: Heat map showing the influence of the number of neurons present in the hidden layers of RBMs and CRBM.

The bold values in Table 1 are those used hereafter.

	RBM training			CRBM training			Finetuning		
L. rate	1e-1	1e-2	1e-3	1e-1	1e-2	1e-3	1e-1	1e-2	1e-3
PER	24.54%	17.57%	**16.88%**	**14.27%**	16.39%	16.88%	**20.26%**	22.97%	27.92%

Table 1: Influence of learning rates on the recognition error.

Consequently, it is possible to train the DTDBN with these different parameters which we summarized in Table 2. The duration necessary for this training is 29.46 minutes, however this time can be significantly reduced by lowering the number of training epochs, with negligible decrease in performance.

Param	Epochs			Learning rates			Hidden layers		Memory	
	RBM	CRBM	Finetune	RBM	CRBM	Finetune	RBM	CRBM	Delay	Freq
Value	10000	10000	10000	1e-3	1e-1	1e-1	30	300	14	2

Table 2: Summary of optimal parameters.

Evaluation

We compare the performance of our approach with recent existing results, from the point of view of classification accuracy (final decision, regardless of how many frames were analyzed) and from that of recognition speed (how many frames are required to predict a gesture).

In their paper, (Jost et al. 2015) use the idea of chunk and shift. Rather than comparing the predicted and actual labels on each frame, the idea is to calculate the average label on a set of frames called a chunk. For our model we use a shift of 1 in order not to drastically reduce the number of data to be classified. This principle is illustrated in Figure 6.

With regard to performance, we conducted tests using four versions of our model:

- Test 1 corresponds to a gesture after gesture recognition per frame (chunk size 1);
- Test 2 corresponds to gesture after gesture recognition using chunks (chunk size 240, in order to match the 2 second window as used by (Jost et al. 2015));
- Test 3 corresponds to recognition of gestures in a continuous stream without using chunks. A continuous stream means that previous frames are implicitly used as memory initialization for the recognition of the subsequent ones;
- Test 4 corresponds to recognition of gestures in a continuous stream using chunks.

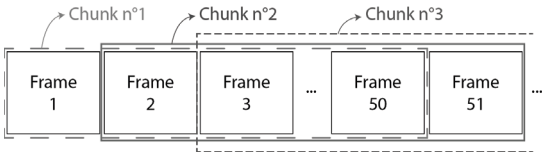


Figure 6: Illustration of chunks composed of 50 frames with a shift of 1 (49 overlapping frames).

The results of our approach were obtained using the following configuration: Intel Xeon E5-1603 2.8GHz, 8GB RAM, GeForce GTX Titan.

Error Rate Estimation

There are many indicators when evaluating the classification performance of a model, commonly calculated based on the confusion matrix. As in the case of our work, the trade-off between true positive (TP) rate (sensitivity) and false positive (FP) rate (1-specificity) has less impact than in other fields such as medicine, we do not investigate different thresholds for classification (ROC curve). Nevertheless, we analyze four performance indicators: Precision, Recall, Accuracy and F-measure (see Table 3).

Comparing the results of tests 1 and 2 with tests 3 and 4 respectively makes it possible to determine the impact that continuous stream recognition has on the model's classification performance. Given that the results are not significantly different, this experiment proves that the DTDBN is an efficient model for segmentation of gestures. This is noteworthy, since continuous stream segmentation appears to be one of the weaknesses of SVM classifiers (Kadu and Kuo 2014).

Classification Performance

In their model, (Jost et al. 2015) have taken into consideration 233 features such as speed, acceleration, symmetry or fluidity, obtained using Principal Component Analysis (PCA). Following this analysis, the labeled learning gestures are divided into multiple chunks for which the various features determined above are calculated. These chunks are then stored in a dictionary where the chunk-label association is performed. The recognition principle is quite similar, and has the advantage that the prediction is performed by a simple comparison of chunks. Since the size of these chunks is fixed, recognition does not depend directly on the number of frames contained in the gestures. This makes it possible to overcome the usual problems of segmentation of gestures during a continuous stream recognition.

We tested the system of (Jost et al. 2015) to a new set of gestures from the INGREDIBLE database (Fitness dataset) to compare the two models on more complex gestures. The comparison results are illustrated in Table 4.

The results presented in Table 4 show a great improvement brought by the DTDBN model in terms of classification performance and robustness both for gesture after gesture recognition and for segmentation of gestures in a continuous stream.

As to further validate the robustness of the proposed model, we also compare our approach to the work of (Kadu

	Precision				Recall				Accuracy				F-measure			
	test 1	test 2	test 3	test 4	test 1	test 2	test 3	test 4	test 1	test 2	test 3	test 4	test 1	test 2	test 3	test 4
G0	86.9 %	100.0 %	86.9 %	100.0 %	97.1 %	100.0 %	95.9 %	99.2 %	98.7 %	100.0 %	98.7 %	99.9 %	91.7 %	100.0 %	91.2 %	99.6 %
G1	91.0 %	100.0 %	91.7 %	99.4 %	96.8 %	100.0 %	97.4 %	99.7 %	98.9 %	100.0 %	99.0 %	99.9 %	93.8 %	100.0 %	94.4 %	99.6 %
G2	90.7 %	100.0 %	89.1 %	99.6 %	88.0 %	100.0 %	89.7 %	99.8 %	98.4 %	100.0 %	98.5 %	100.0 %	89.3 %	100.0 %	89.4 %	99.7 %
G3	98.1 %	100.0 %	98.6 %	99.9 %	91.4 %	100.0 %	92.0 %	94.9 %	98.6 %	100.0 %	98.8 %	99.3 %	94.6 %	100.0 %	95.2 %	97.3 %
G4	74.0 %	100.0 %	70.8 %	94.8 %	76.1 %	100.0 %	74.1 %	99.8 %	93.7 %	100.0 %	93.1 %	99.3 %	75.0 %	100.0 %	72.4 %	97.2 %
G5	86.9 %	100.0 %	87.3 %	98.8 %	70.9 %	100.0 %	73.4 %	92.1 %	93.9 %	100.0 %	94.4 %	98.7 %	78.1 %	100.0 %	79.8 %	95.3 %
G6	78.9 %	100.0 %	81.5 %	91.8 %	94.0 %	100.0 %	95.6 %	99.0 %	96.5 %	100.0 %	97.0 %	98.7 %	85.8 %	100.0 %	88.0 %	95.3 %
G7	92.4 %	100.0 %	92.5 %	100.0 %	81.8 %	100.0 %	79.9 %	96.3 %	96.6 %	100.0 %	96.2 %	99.5 %	86.8 %	100.0 %	85.7 %	98.1 %
G8	76.2 %	100.0 %	77.8 %	95.7 %	88.3 %	100.0 %	87.1 %	100.0 %	95.9 %	100.0 %	95.9 %	99.5 %	81.8 %	100.0 %	82.2 %	97.8 %

Table 3: Performance indicators on the INGREDIBLE database.

#	Jost et al. version		DTDBN version	
	v1	v2	v1	v2
0	96.7%	91.8%	100%	100%
1	97.3%	94.5%	100%	99.4%
2	57.1%	58.1%	100%	99.6%
3	98.4%	75.5%	100%	99.9%
4	97.3%	94.8%	100%	94.8%
5	88.6%	88.2%	100%	98.8%
6	99.6%	88.0%	100%	91.8%
7	96.4%	94.2%	100%	100%
8	92.4%	97.7%	100%	95.7%
Total	91.5%	87.0%	100%	97.8%

Table 4: Recognition performance results on INGREDIBLE Database. The first version (v1) corresponds to a recognition gesture after gesture (same as test 2 for DTDBN) and the second (v2) corresponds to continuous stream recognition (same as test 4 for DTDBN).

and Kuo 2014), who describe a set of techniques which they combine to produce superior classifiers. They use the CMU database which consists of 30 different gestures, with 278 motion capture files containing a total of approximately 0.5 million frames. They obtain recognition rates of 99.2%. Their results were obtained with the following configuration: Intel Core 2 Duo T6500 2.1GHz, 4GB RAM. We replicate their 5-fold cross-validation with the same version of our model (without CMU database specific optimization). Our preliminary results are presented in Table 5.

+	0	1	2	3	4	5	6	7	8	9
0	63	89	22	100	80	100	100	86	80	76
10	60	80	60	71	60	60	83	100	92	20
20	33	83	17	100	19	93	100	80	93	60
Total: 71.7%										

Table 5: Recognition accuracy results on CMU Database in %. Gesture number composed by adding row and column numbers (ex. Gesture 22 = 20+2).

Considering the results presented in Table 5, we quickly realize that the DTDBN is much less efficient than the set of classifiers proposed by (Kadu and Kuo 2014) in the case of a gesture after gesture recognition. Although a recognition rate above 70% is reasonable for a classification of 30 gestures, we note that the PER has increased considerably compared to the classification of 9 gestures of the INGREDIBLE database that we realized previously. We can therefore conclude that the increase in the number of gestures to be recognized leads to confusion in learning, which implies a rapid increase in the PER.

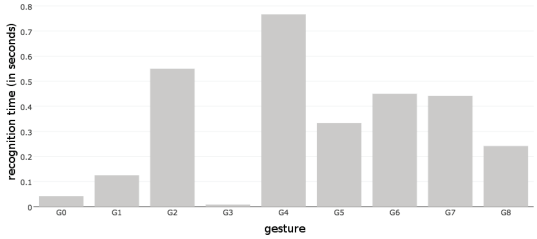


Figure 7: DTDBN gesture recognition delays in a continuous stream.

However, it is important to note that because of their operating principle, these classifiers prove to be much less efficient in recognizing gestures in a continuous stream, that is to say to perform segmentation of gestures, whereas the DTDBN, as discussed in the following subsection, is able to recognize a gesture while it is happening. The SVMs are therefore perfect candidates for classifying time series one after the other but they do not answer our problem of online recognition.

Recognition Speed

In their paper, (Jost et al. 2015) also evaluated the average time needed for their model to recognize a gesture from the INGREDIBLE database in a continuous stream, and compared it to the average human delay. We conducted a test using the DTDBN on the same data in order to obtain a fair comparison. Recognition delays for our model are plotted for each gesture in Figure 7.

The average gesture duration in the INGREDIBLE database (Fitness dataset) is $\sim$4 seconds. Therefore, our model requires to perceive on average 8% of the gesture before it can determine its category. The results of the comparison are presented in Table 6.

	Human	Jost et al.	DTDBN
Avg. rec. time (sec)	0.48	0.41	0.3287

Table 6: Average recognition time on continuous stream. Comparison with results from (Jost et al. 2015).

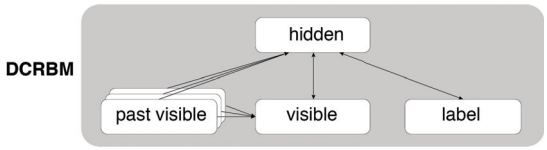


Figure 8: Discriminative CRBM (DCRBM).

In addition to being more efficient and robust, the DTDBN model also has advantages thanks to its principle of operation. The module of (Jost et al. 2015) is based on a comparison of chunks to a dictionary. Therefore, when the database becomes too large, the time required for this comparison would increase and make it difficult to recognize gestures in real time. In the case of the DTDBN, the size of the database only influences the training time.

Conclusions and Future Work

In this work, we sought to answer the problem of online human action classification by using a Temporal Deep Belief Network. Our approach was to adapt this generative model to a classification context. This resulted in the Temporal Discriminative Deep Belief Network, a model dedicated to motion capture data recognition. We sought to evaluate our classifier in terms of performance and robustness on the INGREDIBLE and CMU databases. The approach has promising results: a recognition rate of 100% in a gesture recognition after gesture and 97.8% in the case of gesture segmentation in a continuous stream of motion capture data, on the INGREDIBLE database for which it was optimized. Results on the CMU database are also satisfactory, given the low online recognition time ($\sim$0.32s). Such results show the robustness of the DTDBN since the databases used present variability in terms of fluidity, amplitude and rate of execution of the gestures.

Regarding future work, we intend to explore the possibility to modify the CRBM architecture to add an internal classification layer (Figure 8), similar to discriminative RBMs proposed by (Larochelle and Bengio 2008).

This new approach would replace the top layer of the DTDBN and the current CRBM with a DCRBM with the aim to asses and compare performance with our current approach. We also intend to vary the complexity of DTDBN parameters by replacing the RBMs by DBNs and automatically optimizing the size of the hidden layer of the CRBM.

Acknowledgments

The work in this paper was partially funded by the ANR project SOMBRERO (ANR-14-CE27-0014).

References

Aggarwal, J. K., and Ryoo, M. S. 2011. Human activity analysis: A review. *ACM Computing Surveys (CSUR)* 43(3):16.

Andrew, G., and Bilmes, J. 2012. Sequential deep belief networks. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4265–4268. IEEE.

Chen, Y.; Ding, Z.; Chen, Y.-L.; and Wu, X. 2015. Rapid recognition of dynamic hand gestures using leap motion. In *Information and Automation, 2015 IEEE International Conference on*, 1419–1424. IEEE.

Hinton, G. E.; Osindero, S.; and Teh, Y.-W. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18(7):1527–1554.

Jost, C.; De Loor, P.; Nédélec, L.; Bevacqua, E.; and Stanković, I. 2015. Real-time gesture recognition based on motion quality analysis. In *Intelligent Technologies for Interactive Entertainment (INTETAIN), 2015 7th International Conference on*, 47–56. IEEE.

Kadu, H., and Kuo, C.-C. J. 2014. Automatic human mocap data classification. *IEEE Transactions on Multimedia* 16(8):2191–2202.

Keogh, E. J., and Pazzani, M. J. 1999. Scaling up dynamic time warping to massive datasets. In *European Conference on Principles of Data Mining and Knowledge Discovery*, 1–11. Springer.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In Pereira, F.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc. 1097–1105.

Lafferty, J.; McCallum, A.; and Pereira, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, 282–289.

Larochelle, H., and Bengio, Y. 2008. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, 536–543. ACM.

Moeslund, T. B., and Granum, E. 2001. A survey of computer vision-based human motion capture. *Computer vision and image understanding* 81(3):231–268.

Mohamed, A.-r.; Dahl, G.; and Hinton, G. 2009. Deep belief networks for phone recognition. In *Nips workshop on deep learning for speech recognition and related applications*, volume 1, 39.

Stankovic, I.; De Loor, P.; Demulier, V.; Nédélec, A.; and Bevacqua, E. 2013. The ingredible database: A first step toward dynamic coupling in human-virtual agent body interaction. In *IVA*, 430–431. Springer.

Sukhbaatar, S.; Makino, T.; Aihara, K.; and Chikayama, T. 2011. Robust generation of dynamical patterns in human motion by a deep belief nets. In *ACML*, 231–246.

Taylor, G. W.; Hinton, G. E.; and Roweis, S. T. 2006. Modeling human motion using binary latent variables. In *Advances in neural information processing systems*, 1345–1352.