Action-Affect-Gender Classification using Multi-Task Representation Learning

Timothy J. Shields*, Mohamed R. Amer*, Max Ehrlich, and Amir Tamrakar
SRI International
firstName.lastName@sri.com

Abstract

Recent work in affective computing focused on affect from facial expressions, and not as much on body. This work focuses on body affect. Affect does not occur in isolation. Humans usually couple affect with an action; for example, a person could be running and happy. Recognizing body affect in sequences requires efficient algorithms to capture both the micro movements that differentiate between happy and sad and the macro variations between different actions. We depart from traditional approaches for time-series data analytics by proposing a multi-task learning model that learns a shared representation that is well-suited for action-affect-gender classification. For this paper we choose a probabilistic model, specifically Conditional Restricted Boltzmann Machines, to be our building block. We propose a new model that enhances the CRBM model with a factored multi-task component that enables scaling over larger number of classes without increasing the number of parameters. We evaluate our approach on two publicly available datasets, the Body Affect dataset and the Tower Game dataset, and show superior classification performance improvement over the state-of-the-art.

1. Introduction

Recent work in the field of affective computing [1] focus on face data [2], audiovisual data [3], and body data [4]. One of the main challenges of affect analysis is that it does not occur in isolation. Humans usually couple affect with an action in natural interactions; for example, a person could be walking and happy, or knocking on a door angrily as shown in Fig. 1. These activities are performed differently given the gender of the actor. To recognize body action-affect-gender, efficient temporal algorithms are needed to capture the micro movements that differentiate between happy and sad as well as capture the macro variations between the different actions. The focus of our work is on single-view, multi-task action-affect-gender recognition from skeleton data captured by motion capture or Kinect sensors. Our work leverages the knowledge and work done by the graphics and animation community [5, 6, 7] and uses machine learning to enhance it and make it accessible for a wide variety of applications. We use the Body Affect dataset produced by [7] and the Tower Game [8] dataset as the test cases for our novel multi-task approach.

Time series analysis is a difficult problem that requires efficient modeling, because of the large amounts of data it introduces. There are multiple approaches that designed features to reduce the data dimensionality and then use a simpler model to do classification [9, 10]. We depart from these methods and propose a model that learns shared representation using multi-task learning. We choose Conditional Restricted Boltzmann Machines, which are non-linear probabilistic generative models for modeling time series data, as our building block. They use an undirected bipartite graph with binary latent variables connected to a number of visible variables. A CRBM-based generative model enables modeling short-term phenomenon. CRBMs do not require as many parameters as RNNs and LSTMs since they do not contain any lateral connectivity and they are appropriate for this problem since we are not modeling long term phenomenon. We propose a new hybrid model that enhances the CRBM model with multi-task, discriminative, components based on the work of [11]. This work leads to a superior classification performance, while also allowing us to model temporal dynamics efficiently. We evaluate our approach on the Body Affect [7] and Tower Game [8] datasets and show how our results are superior to the state-of-the-art.

Our contributions: Multi-task learning model for unimodal and multimodal time-series data; Evaluations on two multi-task public datasets [7, 8].

Paper organization: Sec. 2 discusses prior work; Sec. 3 gives a brief background of similar models that motivate our approach, followed by a description of our model; Sec. 4 describes the inference algorithm; Sec. 5 specifies our learning algorithm; Sec. 6 shows quantitative results of our approach, followed by the conclusion in sec. 7.

*Both authors equally contributed to this work

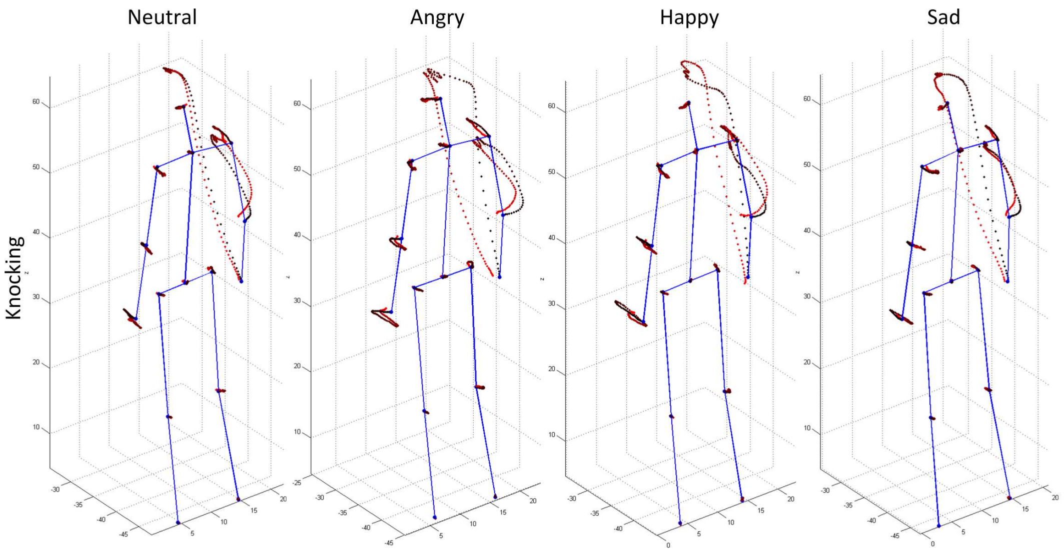


Figure 1. Examples from the Body Affect dataset [7] of a person Knocking with various affects. The trajectory color corresponds to time, where black is the beginning of the sequence, reddish-black is the middle, and red is the end of the sequence.

2. Prior work

In this section we first review literature on activity recognition in RGB-D and Motion Capture Sequences in Sec. 2.1; second we review Multi-Task Learning approaches in Sec. 2.2; finally we review temporal, energy-based, representation learning in Sec. 2.3.

2.1. Body Affect Analysis

Initial work on activity recognition in RGB-D sequences has been popular in recent years with the availability of cheap depth sensors. Since initial work [12], there have been an increasing number of approaches addressing the problem of activity recognition using skeletal data [9]. Prior to activity recognition in RGB-D sequences, datasets were captured using motion capture sensors. During that time, research focused on graphics applications such as generating animation and transitions between animations using signal processing techniques rather than machine learning or computer vision. Their main goal was to generate natural looking skeletons for animation. Some methods used knowledge of signal processing to transform a neutral skeleton pose to reflect a certain emotion [5]. These methods were very constrained to the type of motion and were engineered to reproduce the same motions. Other work used a language based modeling of affect [6] where they modeled actions (verbs) and affect (adverbs) using a graph. They were able to produce results using a combination of low level functions to interpolate between example motions. More recent work [13] modeling non-stylized motion for affect communication used segmentation techniques which divided complex motions into a set of motion primitives that they used as dynamic features. Unlike our approach, their mid-level features were hand engineered rather than learned, which is very limited, does not scale and is prone to feature design flaws. More recent work such as [7] collected natural body affect datasets where they have varied identity, gender, emotion, and actions of the actors but not used it for classification.

2.2. Multi-Task Learning

Multi-task learning is a natural approach for problems that require simultaneous solutions of several related problems [14]. Multi-task learning approaches can be grouped into two main sets. The first set focuses on regularizing the parameter space. The main assumption is that there is an optimal shared parameter space for all tasks. These approaches regularize the parameter space by using a specific loss [15], methods that manually define relationships [16], or more automatic ways that estimate the latent structure of relationships between tasks [17, 18, 19, 20, 21]. The second set focuses on correlating relevant features jointly [22, 23, 24, 25]. Other work focused on the schedule of which tasks should be learned [26]. Multi-task learning achieved good results on vision problems such as: person re-identification [27], multiple attribute recognition [28], and tracking [29]. Recently, Deep Multi-Task Learning (DMTL) emerged with the rise of deep learning. Deep Neural Networks (DNNs) were used to address multi-task learning and were applied successfully to facial landmark detection [30], scene classification [31], object localization and segmentation [32] and attribute prediction [33]. Other work used multi-task autoencoders [34] for object recognition in a generalized domain [35], where the tasks were the different domains. Other work used multi-task RNNs for interaction prediction in still images [36]. Most of the

Deep Multi-task Learning approaches only focused on using DNN-based models applied to still images. Our approach is the first DMTL for temporal and multimodal sequence analysis.

2.3. Representation Learning

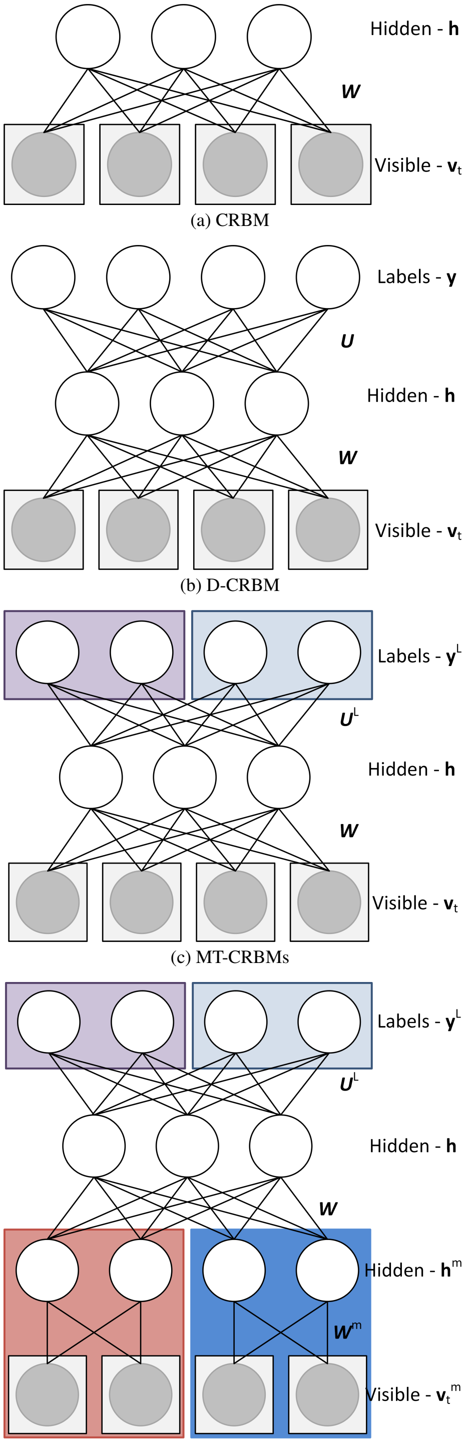
Deep learning has been successfully applied to many problems [37]. Restricted Boltzmann Machines (RBMs) form the building blocks in energy-based deep networks [38, 39]. In [38, 39], the networks are trained using the Contrastive Divergence (CD) algorithm [40], which demonstrated the ability of deep networks to capture the distributions over the features efficiently and to learn complex representations. RBMs can be stacked together to form deeper networks known as Deep Boltzmann Machines (DBMs), which capture more complex representations. Recently, temporal models based on deep networks have been proposed, capable of modeling a rich set of time series analysis problems. These include Conditional RBMs (CRBMs) [41] and Temporal RBMs (TRBMs) [42, 43, 44]. CRBMs have been successfully used in both visual and audio domains. They have been used for modeling human motion [41], tracking 3D human pose [45], and phone recognition [46]. TRBMs have been applied for transferring 2D and 3D point clouds [47], and polyphonic music generation [48].

3. Model

Rather than immediately defining our Multi-Task CRBM (MT-CRBM) model, we discuss a sequence of models, gradually increasing in complexity, such that the different components of our final model can be understood in isolation. We start with the basic CRBM model (sec. 3.1), then we extend the CRBM to a new discriminative (D-CRBM) model (sec. 3.2), then we extend the D-CRBM to our main multi-task model (MT-CRBM) (sec. 3.3), and finally we define a multi-task multimodal model (MTM-CRBM) (sec. 3.4).

3.1. Conditional Restricted Boltzmann Machines (CRBMs)

CRBMs [41] are a natural extension of RBMs for modeling short term temporal dependencies. A CRBM, shown in Figure 2(a), is an RBM which takes into account history from the previous N time instances, $t-N, \ldots, t-1$, when considering time t. This is done by treating the previous time instances as additional inputs. Doing so does not complicate inference. Some approximations have been made to facilitate efficient training and inference, more details are available in [41]. A CRBM defines a probability distribu-



(a) CRBM

(b) D-CRBM

(c) MT-CRBMs

(d) MTM-CRBMs

Figure 2. The deep learning models described in sections 3.1, 3.2, 3.3, and 3.4: (a) CRBM (b) DCRBM (c) MTCRBM (d) MTM-CRBM. The MT-CRBMs learn a shared representation layer for all tasks. In addition to the shared layer, the MTM-CRBMs learn an extra representation layer for each of the modalities, which learn modality-specific representations.

tion p_{C} as a Gibbs distribution (1).

$$p_{\text{C}}(\mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{<t}) = e^{-E_{\text{C}}(\mathbf{v}_t, \mathbf{h}_t|\mathbf{v}_{<t})})/Z(\boldsymbol{\theta}),$$
$$Z(\boldsymbol{\theta}) = \textstyle\sum_{\mathbf{h},\mathbf{v}} e^{-E_{\text{C}}(\mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{<t})},$$
$$\boldsymbol{\theta} = \begin{bmatrix} \{\mathbf{a}, \mathbf{b}\} & \text{-bias,} \\ \{A, B\} & \text{-auto regressive,} \\ \{W\} & \text{-fully connected.} \end{bmatrix} \tag{1}$$

The visible vectors from the previous N time instances, denoted as $\mathbf{v}_{<t}$, influence the current visible and hidden vectors. The probability distributions are defined in (2).

$$\begin{aligned} p_{\text{C}}(v_i|\mathbf{h}, \mathbf{v}_{<t}) &= \mathcal{N}(c_i + \textstyle\sum_j h_j w_{ij}, 1), \\ p_{\text{C}}(h_j = 1|\mathbf{v}, \mathbf{v}_{<t}) &= \sigma(d_j + \textstyle\sum_i v_i w_{ij}), \end{aligned}$$
$$c_i = a_i + \textstyle\sum_p A_{pi} v_{p,<t} \quad , \qquad d_j = b_j + \textstyle\sum_p B_{pj} v_{p,<t}. \tag{2}$$

The new energy function $E_{\text{C}}(\mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{<t})$ in (3) is defined in a manner similar to that of the RBM.

$$E_{\text{C}}(\mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{<t}) = \textstyle\sum_i \frac{(c_i - v_{i,t})^2}{2} - \sum_j d_j h_{j,t} - \sum_{i,j} v_{i,t} w_{ij} h_{j,t}, \tag{3}$$

Note that A and B are matrices defining dynamic biases for $\mathbf{v}_t$ and $\mathbf{h}_t$, consisting of concatenated vectors of previous time instances of $\mathbf{a}$ and $\mathbf{b}$.

3.2. Discriminative CRBMs (D-CRBMs)

We extend the CRBMs to the D-CRBMs shown in Figure 2(b). D-CRBMs are based on the D-RBM model presented in in [11], generalized to account for temporal phenomenon using CRBMs. D-CRBMs define the probability distribution p_{DC} as a Gibbs distribution (4).

$$p_{\text{DC}}(\mathbf{y}_t, \mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{<t}) = e^{-E_{\text{DC}}(\mathbf{y}_t, \mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{<t})}/Z(\boldsymbol{\theta}) \tag{4}$$

The probability distribution over the visible layer will follow the same distributions as in (2). The hidden layer $\mathbf{h}$ is defined as a function of the labels y and the visible nodes $\mathbf{v}$. A new probability distribution for the classifier is defined to relate the label y to the hidden nodes $\mathbf{h}$ (5).

$$\begin{aligned} p_{\text{DC}}(v_{i,t}|\mathbf{h}_t, \mathbf{v}_{<t}) &= \mathcal{N}(c_i + \textstyle\sum_j h_j w_{ij}, 1), \\ p_{\text{DC}}(h_{j,t} = 1|y_t, \mathbf{v}_t, \mathbf{v}_{<t}) &= \sigma(d_j + \textstyle\sum_k y_{k,t} u_{jk} + \sum_i v_{i,t} w_{ij}), \\ p_{\text{DC}}(y_{k,t}|\mathbf{h}) &= \frac{e^{s_k + \sum_j u_{jk} h_j}}{\sum_{k^*} e^{s_{k^*} + \sum_j u_{jk^*} h_j}}. \end{aligned} \tag{5}$$

The new energy function E_{DC} is defined as in (6).

$$\begin{aligned} E_{\text{DC}}(\mathbf{y}_t, \mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{<t}) &= \underbrace{E_{\text{C}}(\mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{<t})}_{\text{Generative}} \\ &- \underbrace{\sum_k s_k y_{k,t} - \sum_{j,k} h_{j,t} u_{jk} y_{k,t}}_{\text{Discriminative}} \end{aligned} \tag{6}$$

3.3. Multi-Task CRBMs (MT-CRBMs)

In the same way the CRBMs can be extended to the DC-RBMs by adding a discriminative term to the model, we can extend the CRBMs to be multi-task MT-CRBMs Figure 2(c). MTCRBMs define the probability distribution p_{MT} as a Gibbs distribution (7). The MT-CRBMs learn a shared representation layer for all tasks.

$$p_{\text{MT}}(\mathbf{y}_t^L, \mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{<t}) = \frac{e^{-E_{\text{DC}}(\mathbf{y}_t^L, \mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{<t})}}{Z(\boldsymbol{\theta})}. \tag{7}$$

The probability distribution over the visible layer will follow the same distributions as in (5). The hidden layer $\mathbf{h}$ is defined as a function of the multi-task labels y^L and the visible nodes $\mathbf{v}$. A new probability distribution for the multi-task classifier is defined to relate the multi-task labels y^L to the hidden nodes $\mathbf{h}$ as shown in (8).

$$\begin{aligned} &p_{\text{MT}}(v_{i,t}|\mathbf{h}_t, \mathbf{v}_{<t}) = \mathcal{N}(c_i + \textstyle\sum_j h_j w_{ij}, 1), \\ &p_{\text{MT}}(h_{j,t} = 1|y_t^L, \mathbf{v}_t, \mathbf{v}_{<t}) \\ &\qquad = \sigma(d_j + \textstyle\sum_{l,k} y_{k,t}^l u_{jk}^l + \sum_i v_{i,t} w_{ij}), \\ &p_{\text{MT}}(y_{k,t}^l|\mathbf{h}) = \frac{\exp[s_k^l + \sum_j u_{jk}^l h_j]}{\sum_{k^*} \exp[s_{k^*}^l + \sum_j u_{jk^*}^l h_j]}. \end{aligned} \tag{8}$$

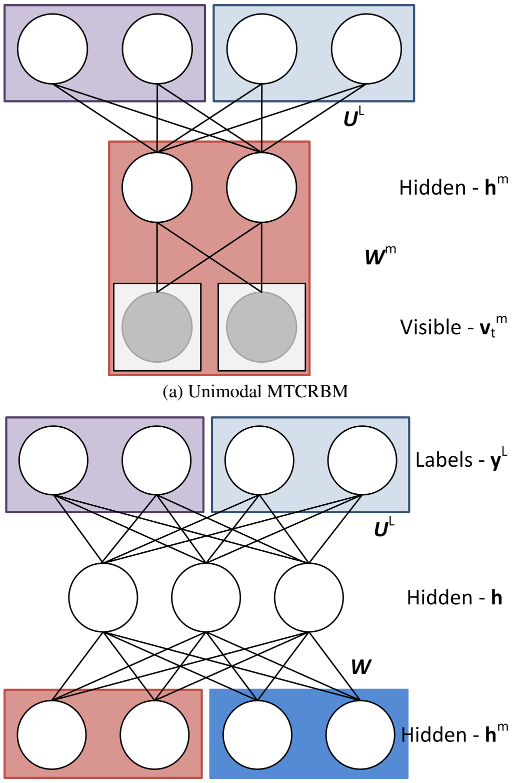
The energy for the model shown in Figure 2(c), E_{MT}, is defined as in (9).

$$\begin{aligned} E_{\text{MT}}(\mathbf{y}_t^L, \mathbf{v}_t, \mathbf{h}_t|\mathbf{v}_{<t}) &= \underbrace{E_{\text{C}}(\mathbf{v}_t, \mathbf{h}_t|\mathbf{v}_{<t})}_{\text{Generative}} \\ &- \underbrace{\sum_{k,l} s_k^l y_{k,t}^l - \sum_{j,k,l} h_{j,t} u_{jk} y_{k,t}^l}_{\text{Multi-Task}} \end{aligned} \tag{9}$$

3.4. Multimodal MT-CRBMs (MTM-CRBMs)

We can naturally extend MT-CRBMs to MTM-CRBMs. A MTM-CRBMs combines a collection of unimodal MT-CRBMs, one for each visible modality. The hidden representations produced by the unimodal MT-CRBMs are then treated as the visible vector of a single fusion MT-CRBMs. The result is a MTMCRBM model that relates multiple temporal modalities to multi-task classification labels. MTM-CRBMs define the probability distribution p_{MTM} as a Gibbs distribution (10). The MTM-CRBMs learn an extra representation layer for each of the modalities, which learns a modality specific representation as well as the shared layer for all the tasks.

$$\begin{aligned} &p_{\text{MTM}}(\mathbf{y}_t^L, \mathbf{h}_t, \mathbf{h}_t^{1:M}, \mathbf{v}_t^{1:M}|\mathbf{v}_{<t}^{1:M}) \\ &\qquad = \exp[-E_{\text{MTM}}(\mathbf{y}_t^L, \mathbf{h}_t, \mathbf{h}_t^{1:M}, \mathbf{v}_t^{1:M}|\mathbf{v}_{<t}^{1:M})]/Z(\boldsymbol{\theta}). \end{aligned} \tag{10}$$



(a) Unimodal MTCRBM

(b) Fusion MTCRBM

Figure 3. This figure specifies the inference algorithm. We first classify the unimodal data by activating the corresponding hidden layers $\mathbf{h}_t^m$ as shown in (a), followed by classifying the multimodal data by activating the fusion layer $\mathbf{h}_t$ as shown in (b).

Similar to the MT-CRBMs(8), the hidden layer $\mathbf{h}$ is defined as a function of the labels y^L and the visible nodes $\mathbf{v}$. A new probability distribution for the classifier is defined to relate the label y^L to the hidden nodes $\mathbf{h}$ is defined as in (11).

$$
\begin{aligned}
&p_{\text{MTM}}(v_{i,t}^m|\mathbf{h}_t^m, \mathbf{v}_{<t}^m) = \mathcal{N}(c_i^m + \textstyle\sum_j h_j^m w_{ij}^m, 1),\\
&p_{\text{MTM}}(h_{j,t}^m = 1|y_t^L, \mathbf{v}_t^m, \mathbf{v}_{<t}^m) = \sigma(d_j^m + \textstyle\sum_{l,k} y_{k,t}^l u_{jk}^l + \sum_i v_{i,t}^m w_{ij}^m),\\
&p_{\text{MTM}}(y_{k,t}^l|\mathbf{h}_t^m) = \frac{\exp[s_k^l + \sum_j u_{jk}^{m,l} h_{j,t}^m]}{\sum_{l*} \exp[s_{k*}^l + \sum_j u_{jk*}^{m,l} h_{j,t}^m]},\\
&p_{\text{MTM}}(h_{n,t} = 1|y_t^L, \mathbf{h}_t^{1:M}, \mathbf{h}_{<t}^{1:M}) =\\
&\qquad\qquad \sigma(f_n + \textstyle\sum_{l,k} y_{k,t}^l u_{nk}^l + \sum_{m,j} h_{j,t}^m w_{jn}^m),\\
&p_{\text{MTM}}(y_{k,t}^l|\mathbf{h}) = \frac{\exp[s_k^l + \sum_j u_{nk}^l h_n]}{\sum_{k*} \exp[s_{k*}^l + \sum_n u_{nk*}^l h_n]}.
\end{aligned}
\tag{11}
$$

where,

$$
\begin{aligned}
c_i^m &= a_i^m + \textstyle\sum_p A_{p,i}^m v_{p,<t}^m,\\
d_j^m &= b_j^m + \textstyle\sum_p B_{p,j}^m v_{p,<t},\\
f_n &= e_n + \textstyle\sum_{m,r} C_{r,n}^m h_{r,<t}^m.
\end{aligned}
\tag{12}
$$

The new energy function E_{MTM} is defined in (13) similar to that of the MT-CRBMs (7).

$$
\begin{aligned}
&E_{\text{MTM}}(\mathbf{y}_t^L, \mathbf{h}_t, \mathbf{h}_t^{1:M}, \mathbf{v}_t^{1:M}|\mathbf{v}_{<t}^{1:M}) =\\
&\underbrace{\sum_m E_{\text{MT}}(\mathbf{y}_t^L, \mathbf{h}_t^m, \mathbf{v}_t^m|\mathbf{v}_{<t}^m)}_{\text{Unimodal}}\\
&\underbrace{-\sum_j f_n h_{n,t} - \sum_{j,k,m} h_{j,t}^m w_{jn} h_{n,t}}_{\text{Fusion}}\\
&\underbrace{-\sum_{k,l} s_k^l y_{k,t}^l - \sum_{n,k,l} h_{n,t} u_{nk}^l y_{k,t}^l}_{\text{Multi-Task}}
\end{aligned}
\tag{13}
$$

4. Inference

We first discuss inference for the MTM-CRBM since it is the most general case. To perform classification at time t in the MTM-CRBM given $\mathbf{v}_{<t}^{1:M}$ and $\mathbf{v}_t^{1:M}$ we use a bottom-up approach, computing the mean of each node given the activation coming from the nodes below it; that is, we compute the mean of $\mathbf{h}_t^m$ using $\mathbf{v}_{<t}^m$ and $\mathbf{v}_t^m$ for each modality, then we compute the mean of $\mathbf{h}_t$ using $\mathbf{h}_{<t}^{1:M}$, then we compute the mean of $\mathbf{y}_t^L$ for each task using $\mathbf{h}_t$, obtaining the classification probabilities for each task. Figure 3 illustrates our inference approach. Inference in the MT-CRBM is the same as the MTM-CRBM, except there is only one modality, and inference in the D-CRBM is the same as the MT-CRBM, except there is only one task.

5. Learning

Learning our model is done using Contrastive Divergence (CD) [40], where $\langle\cdot\rangle_{data}$ is the expectation with respect to the data and $\langle\cdot\rangle_{recon}$ is the expectation with respect to the reconstruction. The learning is done using two steps: a bottom-up pass and a top-down pass using sampling equations from (5) for D-CRBM, (8) for MT-CRBM, and (11) for MTM-CRBM. In the bottom-up pass the reconstruction is generated by first sampling the unimodal layers $p(h_{t,j}^m = 1|\mathbf{v}_t^m, \mathbf{v}_{<t}^m, y_l)$ for all the hidden nodes in parallel. This is followed by sampling the fusion layer $p(h_{t,n} = 1|y_{k,t}^L, \mathbf{h}_t^{1:M}, \mathbf{h}_{<t}^{1:M})$. In the top-down pass the unimodal layer is generated using the activated fusion layer $p(h_{t,j}^m = 1|\mathbf{h}_t, y_{k,t}^L)$. This is followed by sampling the visible nodes $p(v_{t,i}^m|\mathbf{h}_t^m, \mathbf{v}_{<t}^m)$ for all the visible nodes in parallel. The gradient updates are described in (14). Similarly

learning of D-CRBM and MT-CRBM could be done.

$$\begin{array}{llll}
\Delta a_i & \propto & \langle v_i^m \rangle_{data} & - & \langle v_i^m \rangle_{recon}, \\
\Delta b_j & \propto & \langle h_j^m \rangle_{data} & - & \langle h_j^m \rangle_{recon}, \\
\Delta e_n & \propto & \langle h_n \rangle_{data} & - & \langle h_n \rangle_{recon}, \\
\Delta s_k^l & \propto & \langle y_k^l \rangle_{data} & - & \langle y_k^l \rangle_{recon}, \\
\Delta A_{p,i,<t}^m & \propto & v_{k,<t}^m (\langle v_{i,t}^m \rangle_{data} & - & \langle v_{i,t}^m \rangle_{recon}), \\
\Delta B_{p,j,<t}^m & \propto & v_{i,<t}^m (\langle h_{j,t}^m \rangle_{data} & - & \langle h_{j,t}^m \rangle_{recon}), \\
\Delta C_{r,n,<t}^m & \propto & h_{j,<t}^m (\langle h_{n,t} \rangle_{data} & - & \langle h_{n,t} \rangle_{recon}), \\
\Delta w_{i,j}^m & \propto & \langle v_i^m h_j^m \rangle_{data} & - & \langle v_i^m h_j^m \rangle_{recon}, \\
\Delta w_{j,k} & \propto & \langle h_j^m h_n \rangle_{data} & - & \langle h_j^m h_n \rangle_{recon}, \\
\Delta u_{jk}^{l,m} & \propto & \langle y_k^l h_j^m \rangle_{data} & - & \langle y_k^l h_j^m \rangle_{recon}, \\
\Delta u_{nk}^L & \propto & \langle y_k^l h_n \rangle_{data} & - & \langle y_k^l h_n \rangle_{recon}.
\end{array} \tag{14}$$

6. Experiments

We now describe the datasets in (sec 6.1), specify the implementation details in (sec 6.2), and present our quantitative results in (sec 6.3).

6.1. Datasets

Our problem is very particular in that we focus on multi-task learning for body affect. In the literature [4, 9] most of the datasets were either single task for activity recognition, not publicly available, too few instances, or only RGB-D without skeleton. We found two available datasets to evaluate our approach that are multi-task. The first dataset is the Body Affect dataset [7], collected using a motion capture sensor, which consists of a set of actors performing several actions with different affects. The second dataset is the Tower Game [8], collected using a Kinect sensor, which consists of an interaction between two humans performing a cooperative task, with the goal of classifying different components of entrainment. In the following subsections we describe the datasets.

Body Affect Dataset: This dataset [7] consists of a library of human movements captured using a motion capture sensor, annotated with actor, action, affect, and gender. The dataset was collected for studying human behavior and personality properties from human movement. The data consists of 30 actors (15 female and 15 male) each performing four actions (walking, knocking, lifting, and throwing) with each of four affect styles (angry, happy, neutral, and sad). For each actor, there are 40 data instances: 8 instances of walking (2 directions x 4 affects), 8 instances of knocking (2 repetitions x 4 affects), 8 instances of lifting (2 repetitions x 4 affects), 8 instances of throwing (2 repetitions x 4 affects), and 8 instances of the sequences (2 repetitions x 4 affects). For knocking and lifting and throwing there were 5 repetitions per data instances. Thus, the 24 records of knocking, lifting, and throwing contain 120 separate instances, yielding a total of 136 instances per actor and a total of 4,080 instances. We split dataset into 50% training using 15 actors and 50% testing using the other 15 actors.

Tower Game Dataset: This dataset [8] is a simple game of tower building often used in social psychology to elicit different kinds of interactive behaviors from the participants. It is typically played between two people working with a small fixed number of simple toy blocks that can be stacked to form various kinds of towers. The data consists of 112 videos which were divided into 1213 10-second segments indicating the presence or absence of these behaviors in each segment. Entrainment is the alignment in the behavior of two individuals and it involves simultaneous movement, tempo similarity, and coordination. Each measure was rated low, medium, or high for the entire 10 seconds segment. 50% of that data was used for training and 50% were used for testing. In this dataset we call each person's skeletal data a modality, where our goal is to model mocap-mocap representations.

6.2. Implementation Details

For pre-processing the Tower Game dataset, we followed the same approach as [50] by forming a body centric transformation of the skeletons generated by the Kinect sensors. We use the 11 joints from the upper body of the two players since the tower game almost entirely involves only upper body actions and gestures are done using the upper body. We used the raw joint locations normalized with respect to a selected origin point. We use the same descriptor provided by [51, 52]. The descriptor consists of 84 dimensions based on the normalized joints location, inclination angles formed by all triples of anatomically connected joints, azimuth angles between projections of the second bone and the vector on the plane perpendicular to the orientation of the first bone, bending angles between a basis vector, perpendicular to the torso, and joint positions. As for the Body Affect dataset we decided to use the full body centric representation [53] for motion capture sensors resulting in 42 dimensions per frame.

For the Body Affect dataset we trained a three-task model for the following three tasks: Action (AC) $\in$ {Walking, Knocking, Lifting, Throwing}, Affect (AF) $\in$ {Neutral, Happy, Sad, Angry}, Gender (G) $\in$ {Male, Female}. The data is split into a training set consisting of 50% of the instances, and a test set consisting of the remaining 50%. For the Tower Game dataset we trained a three-task model for the following tasks,: Tempo Similarity (TS), Coordination (C), and Simultaneous Movement (SM), each in {Low, Medium, High}. The data is split into a training set consisting of 50% of the instances, and a test set consisting of the remaining 50%.

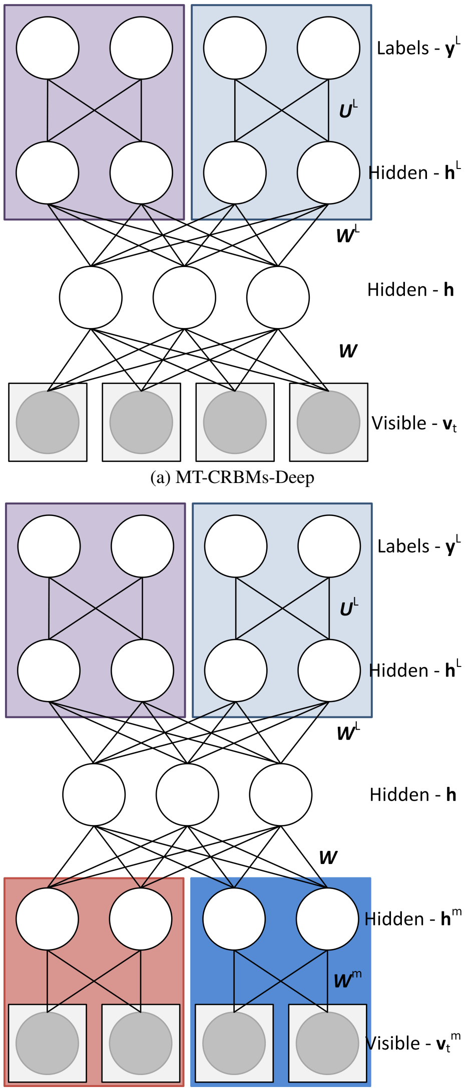
We tuned our model parameters. For selecting the model parameters we used a grid search. We varied the number of hidden nodes per layer in the range of $\{10, 20, 30, 50, 70, 100, 200\}$, as well as the auto-regressive nodes in the range of $\{5, 10\}$, resulting a total of 2744 trained models. The best performing model on the Body Affect dataset has the following configuration $v = 42, h = 30, v_{<t} = 42 \times 10$ and the best performing model on the Tower Game dataset has the following configuration $v^m = 84, h^m = 30, v^m_{<t} = 10 \times 84$ for each of the modalities and for the fusion layer in the Tower Game dataset $h^{1:M} = 60, h = 60, h^{1:M}_{<t} = 10 \times 60$.

Note that in our MT-CRBM model, the tasks are assumed conditionally independent given the hidden representation. Thus the number of parameters needed for the hidden-label edges is $H \cdot \sum_{k=1}^{L} Y_k$, where H is the dimensionality of the hidden layer and Y_k is the number of classes for task k. Contrast this to the number of parameters needed if instead the tasks are flattened as a Cartesian product, $H \cdot \prod_{k=1}^{L} Y_k$. Our factored representation of the multiple tasks uses only linearly many parameters instead of the exponentially many parameters needed for the flattened representation.

6.3. Quantitative Results

We first define baselines and variants of the model, followed by the average classification accuracy results on the two datasets.

Baselines and Variants: Since we compare our approach against the results presented in [8] we decided to use the same baselines they used. They used SVM classifiers on a combination of features. *SVM+RAW:* The first set of features consisted of first order static and dynamic handcrafted skeleton features. The static features are computed per frame. The features consist of relationships between all pairs of joints of a single actor, and the relationships between all pairs of joints of both the actors. The dynamic features are extracted per window (a set of 300 frames). In each window, they compute first and second order dynamics of each joint, as well as relative velocities and accelerations of pairs of joints per actor, and across actors. The dimensionality of their static and dynamic features is (257400 D). *SVM+BoW100* and *SVM+BoW300:* To reduce their dimensionality they used, Bag-of-Words (BoW) (100 and 300 D) [54, 52]. We also evaluate our approach using *HCRF* [55]. We define our own model's variants, *D-CRBMs* which is our single-task model presented in Section 3.2, *MT-CRBMs* which is our multi-task model presented in Section 3.3, *MTM-CRBMs* the multi-modal multi-task model presented in Section 3.4 and *DM-CRBMs* an extension to the D-CRBMs to be multimodal similar to MTM-CRBMs. We also add two new variants[1] *MT-CRBMs-Deep* and *MTM-CRBMs-Deep* shown in Fig.4, which are a deeper version of the original models, by adding a task specific representation layer.



(a) MT-CRBMs-Deep

(b) MTM-CRBM-Deep

Figure 4. Deep variants of the models presented in sections 3.3 and 3.4. (a) MT-CRBMs-Deep (b) MTM-CRBMs-Deep. The Deep variants add an extra representation layer for each of the tasks, which learns a task specific representation.

Classification: For the Body Affect dataset, Table 1 shows the results of the baselines as well as our model and its variants. For the Tower Game dataset, Table 2 shows our average classification accuracy using different features and baselines combinations as well as the results from our models. We can see that the *MT-CRBMs-Deep* model outperforms all the other models for both cases, thereby demonstrating its effectiveness on predicting multi-task labels cor-

[1] This model is initially prototyped by [56] in the deep learning book.

Table 1. Average Classification Accuracy on The Body Affect Dataset.

Classifier (labels)	AC(4)	AF(4)	G(2)
Random Guess	25.0	25.0	50.0
SVM+Raw	35.6	32.2	65.1
SVM+BoW100	41.3	34.1	71.4
SVM+BoW300[52]	39.9	32.8	69.5
HCRF[55]	44.8	34.7	74.1
D-CRBMs	52.6	30.7	78.4
MT-CRBMs	53.5	31.2	78.2
MT-CRBMs-Deep	54.5	32.7	78.4

Table 2. Average Classification Accuracy on The Tower Game Dataset.

Classifier (labels)	TS (3)	C (3)	SM (3)
Random Guess	33.3	33.3	33.3
SVM+Raw [8]	59.3	52.2	39.5
SVM+BoW100 [8]	65.6	55.8	44.3
SVM+BoW300 [52]	54.4	47.5	42.8
HCRF[55]	67.2	58.8	44.5
DM-CRBMs	76.5	62.0	49.2
MTM-CRBMs	86.2	70.0	63.5
MTM-CRBMs-Deep	87.2	70.0	72.8

rectly. Furthermore, the *MTM-CRBMs-Deep* model outperforms all the SVM variants which used high dimensional handcrafted features, demonstrating its ability to learn a rich representation starting from the raw skeleton features. Note that only the *MTM-CRBMs* and *MTM-CRBMs-Deep* performed well on predicting the different tasks simultaneously with a relatively large margin better than the other models, using a shared representation that uses less parameters than our D-CRBMs model that treats all the labels flat.

7. Conclusion and Future Work

We have proposed a collection of hybrid models, both discriminative and generative, that model the relationships in and distributions of temporal, multimodal, multi-task data. An extensive experimental evaluation of these models on two different datasets demonstrates the superiority of our approach over the state-of-the-art for multi-task classification of temporal data. This improvement in classification performance is accompanied by new generative capabilities and an efficient use of model parameters via factorization across tasks.

The factorization of tasks used in our approach means the number of parameters grows only linearly with the number of tasks and classes. This is seen to be significant when contrasted with a single-task model that uses a flattened Cartesian product of tasks, where the number of parameters grows exponentially with the number of tasks. Our factorized approach makes adding additional tasks a trivial matter.

The generative capabilities of our approach enable new and interesting applications. A future direction of work is to further explore and improve these generative applications of the models.

Acknowledgments

This research was partially developed with funding from the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL). The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government

References

[1] Picard, R.W.: Affective Computing. MIT Press (1995) 1

[2] Calvo, R., D'Mello, S., Gratch, J., Kappas, A., Cohn, J.F., Torre, F.D.L.: Automated face analysis for affective computing (2014) 1

[3] Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S.: A survey of affect recognition methods: Audio, visual, and spontaneous expressions. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(1) (2009) 39–58 1

[4] Kleinsmith, A., Bianchi-Berthouze, N.: Affective body expression perception and recognition: A survey. IEEE Transactions on Affective Computing **4**(1) (2013) 15–33 1, 6

[5] Amaya, K., Bruderlin, A., Calvert, T.: Emotion from motion. In: GI. (1996) 1, 2

[6] Rose, C., Bodenheimer, B., Cohen, M.F.: Verbs and adverbs: Multidimensional motion interpolation using radial basis functions. In: Computer Graphics and Applications. (1998) 1, 2

[7] MA, Y., PATERSON, H.M., POLLICK, F.E.: A motion capture library for the study of identity, gender, and emotion perception from biological motion. In: BMR. (2006) 1, 2, 6

[8] Salter, D.A., Tamrakar, A., Behjat Siddiquie, M.R.A., Divakaran, A., Lande, B., Mehri, D.: The tower game dataset: A multimodal dataset for analyzing social interaction predicates. In: ACII. (2015) 1, 6, 7, 8

[9] Zhang, J., Li, W., Ogunbona, P.O., Wang, P., Tang, C.: Rgb-d-based action recognition datasets: A survey. In: arxiv. (2016) 1, 2, 6

[10] Chaquet, J.M., Carmona, E.J., Fernández-Caballero, A.: A survey of video datasets for human action and activity recognition. CVIU **117**(6) (2013) 633 – 659 1

[11] Larochelle, H., Bengio, Y.: Classification using discriminative restricted boltzmann machines. In: ICML. (2008) 1, 4

[12] Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, IEEE (2010) 9–14 2

[13] Bernhardt, D., Robinson, P.: Detecting affect from non-stylised body motions. In: ACII. (2007) 2

[14] Caruana, R.: Multitask learning. In: Machine Learning. (1997) 2

[15] Evgeniou, T., Pontil, M.: Regularized multi?task learning. In: KDD. (2004) 2

[16] Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. In: JMLR. (2005) 2

[17] Ciliberto, C., Rosasco, L., Villa, S.: Learning multiple visual tasks while discovering their structure. In: CVPR. (2015) 2

[18] Maurer, A., Pontil, M., Romera-Paredes, B.: The benefit of multitask representation learning. In: ArXiv. (2015) 2

[19] Maurer, A., Pontil, M., Romera-Paredes, B.: Sparse coding for multitask and transfer learning. In: ICML. (2013) 2

[20] Kumar, A., III, H.D.: Learning task grouping and overlap in multi-task learning. In: ICML. (2012) 2

[21] Zhou, J., Chen, J., Ye, J.: Clustered multi-task learning via alternating structure optimization. In: NIPS. (2011) 2

[22] Argyriou, A., Evgeniou, T., Pontil., M.: Convex multi-task feature learning. In: Machine Learning. (2008) 2

[23] Kang, Z., Grauman, K.: Learning with whom to share in multi-task feature learning. In: ICML. (2011) 2

[24] Romera-Paredes, B., Aung, H., Bianchi-Berthouze, N., Pontil, M.: Multilinear multitask learning. In: ICML. (2013) 2

[25] Yang, Y., Hospedales, T.M.: A unified perspective on multi-domain and multi-task learning. In: ICLR. (2015) 2

[26] Pentina, A., Sharmanska, V., Lampert, C.H.: Curriculum learning of multiple tasks. In: CVPR. (2015) 2

[27] Su, C., Yang, F., Zhang, S., Tian, Q., Davis, L.S., Gao, W.: Multi-task learning with low rank attribute embedding for person re-identification. In: ICCV. (2015) 2

[28] Chen, L., Zhang, Q., Li, B.: Predicting multiple attributes via relative multi-task learning. In: CVPR. (2014) 2

[29] Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via structured multi-task sparse learning. In: IJCV. (2012) 2

[30] Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Facial landmark detection by deep multi-task learning. In: ECCV. (2014) 2

[31] Lapin, M., Schiele, B., Hein, M.: Scalable multitask representation learning for scene classification. In: CVPR. (2014) 2

[32] Wang, X., Zhang, L., Lin, L., Liang, Z., Zuo, W.: Deep joint task learning for generic object extraction. In: NIPS. (2014) 2

[33] Abdulnabi, A.H., Wang, G., Lu, J.: Multi-task cnn model for attribute prediction. In: arXiv. (2016) 2

[34] Zhuang, F., Luo, D., Jin, X., Xiong, H., Luo, P., He, Q.: Representation learning via semi-supervised autoencoder for multi-task learning. In: ICML. (2015) 2

[35] Ghifary, M., Kleijn, W.B., Zhang, M., Balduzzi, D.: Domain generalization for object recognition with multi-task autoencoders. In: ICCV. (2015) 2

[36] Chu, X., Ouyang, W., Yang, W., Wang, X.: Multitask recurrent neural network for immediacy prediction. In: ICCV. (2015) 2

[37] Bengio, Y.: Learning deep architectures for ai. In: FTML. (2009) 3

[38] Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. In: NC. (2006) 3

[39] Salakhutdinov, R., Hinton, G.E.: Reducing the dimensionality of data with neural networks. In: Science. (2006) 3

[40] Hinton, G.E.: Training products of experts by minimizing contrastive divergence. In: NC. (2002) 3, 5

[41] Taylor, G.W., Hinton, G.E., Roweis, S.T.: Two distributed-state models for generating high-dimensional time series. In: Journal of Machine Learning Research. (2011) 3

[42] Sutskever, I., Hinton, G.E.: Learning multilevel distributed representations for high-dimensional sequences. In: AISTATS. (2007) 3

[43] Sutskever, I., Hinton, G., Taylor, G.: The recurrent temporal restricted boltzmann machine. In: NIPS. (2008) 3

[44] Hausler, C., Susemihl, A.: Temporal autoencoding restricted boltzmann machine. In: CoRR. (2012) 3

[45] Taylor, G.W., et. al.: Dynamical binary latent variable models for 3d human pose tracking. In: CVPR. (2010) 3

[46] Mohamed, A.R., Hinton, G.E.: Phone recognition using restricted boltzmann machines. In: ICASSP. (2009) 3

[47] M. D. Zeiler, G. W. Taylor, L.S., Matthews, I., Fergus, R.: Facial expression transfer with input-output temporal restricted boltzmann machines. In: NIPS. (2011) 3

[48] Lewandowski, N.B., Bengio, Y., Vincent, P.: Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In: ICML. (2012) 3

[49] Salakhutdinov, R., Hinton, G.E.: Deep boltzmann machine. In: AISTATS. (2009)

[50] Neverova, N., Wolf, C., Taylor, G., Nebout, F.: Moddrop: adaptive multi-modal gesture recognition. In: PAMI. (2014) 6

[51] Neverova, N., Wolf, C., Taylor, G.W., Nebout, F.: Multi-scale deep learning for gesture detection and localization. In: ECCV-W. (2014) 6

[52] Zanfir, M., Leordeanu, M., Sminchisescu, C.: The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In: ICCV. (2013) 6, 7, 8

[53] Zheng, Q., Hao, Z., Huang, H., Xu, K., Zhang, H., Cohen-Or, D., Chen, B.: Skeleton-intrinsic symmetrization of shapes. In: Computer Graphics Forum. Volume 34. (2015) 275–286 6

[54] Niebles, J., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. IJCV **79**(3) (2008) 299–318 7

[55] Wang, S.B., Quattoni, A., Morency, L.P., Demirdjian, D., Darrell, T.: Hidden conditional random fields for gesture recognition. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. (2006) 7, 8

[56] Ian Goodfellow, Y.B., Courville, A.: Deep learning. Book in preparation for MIT Press (2016) 7