Two-Stage Temporal Multimodal Learning for Speaker and Speech Recognition

Qianli Ma[1,2(✉)], Lifeng Shen[1], Ruishi Su[1], and Jieyu Chen[3]

[1] School of Computer Science and Engineering,
South China University of Technology, Guangzhou 510006, China
qianlima@scut.edu.cn, scuterlifeng@foxmail.com
[2] Guangdong Key Laboratory of Big Data Analysis and Processing,
Guangzhou 510006, China
[3] Linguistic Department, University of California, San Diego, CA 92093, USA
jic387@ucsd.edu

Abstract. Temporal information prevails in multimodal sequence data, such as video data and speech signals. In this paper, we propose a two-stage learning to model the temporal information in multimodal sequences. At the first learning stage, static representative features are extracted from each modality at every time step. Then joint representations across various modalities are effectively learned within a joint fusion layer. The second one is to transfer the static features into corresponding dynamical features by jointly learning the temporal information and dependencies between different time steps with a Long Short-Term Memory (LSTM). Compared with previous multimodal methods, the proposed model is efficient in learning temporal joint representations. Evaluated on Big Bang Theory speaker recognition dataset and AVLetters speech recognition dataset, our model proves to outperform other methods.

Keywords: Temporal multimodal learning · Speaker recognition · Speech recognition

1 Introduction

In most cases, both visual and audio information plays a vital role in understanding a given circumstance by computers. For example, by employing facial information can we deal with speaker recognition to some extent. However, it would fail if real-world data had illumination variations or blurring information. Also, audio information alone is not sufficient for speech recognition either, because speech signals often contain noises. Thus, researchers tend to take both visual and audio information into account to reduce recognition errors, which has been verified by previous research [8,10,12].

In recent years, a number of approaches have been proposed to fuse audio and visual in-formation for better recognition. Recurrent Temporal Multimodal RBM (RTMRBM) [6] added joint layers on the top of Multimodal RBMs

© Springer International Publishing AG 2017
D. Liu et al. (Eds.): ICONIP 2017, Part II, LNCS 10635, pp. 64–72, 2017.
https://doi.org/10.1007/978-3-319-70096-0_7

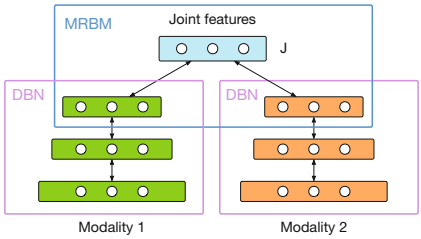

Fig. 1. Structure of Multimodal Deep Belief Network (MDBN)

(MRBMs) [13], which connects the sequence of MRBMs to learn temporal joint representation. RTMRBM attempted to model the overall joint distribution of the entire multimodal time series directly. Since its learning at each time step relies on the joint distribution of the whole time frames, it will significantly increase the training cost of the model. Most recently, Jimmy Ren [11] built the Multimodal Long Short-term Memory (Multimodal LSTM), and tried to explicitly model the long-term dependencies in a single modal both within the same modality and across modalities. To this end, multimodal LSTM duplicated the internal nodes but kept the parameters shared for each modality. However, different modalities will, more often than not, provide various information, and it is hard to extract all important information from various modalities with just one same set of weights.

In this paper, we propose a two-stage learning to model the temporal information in multimodal sequences. At the first learning stage, static representative features are extracted from each modality at each time step by a Deep Belief Network (DBN) [3]. Then joint representations across various modalities are effectively learned within a Multimodal Restricted Boltzmann Machine (MRBM). The second stage is to transfer static features into corresponding dynamical features by jointly learning the temporal information by a LSTM. Since there is no need of distribution estimation of the whole temporal data, our method will noticeably decrease the training cost at Stage I. Furthermore, our experimental results show that it is more efficient to learn temporal information and dependencies in high-level abstract features than in low-level ones. Our model achieves a better performance on Big Bang Theory dataset (speaker recognition), and AVLetters dataset (speech recognition) than other methods.

2 Background

2.1 Multimodal Deep Belief Network

Multimodal Deep Belief Network (MDBN) [13] is a generative model, which employs a Multimodal Restricted Boltzmann Machine (MRBM) [13] to extract joint multimodal features from the top layer of Deep Belief Network (DBN) [3].

For each modality, MDBN establishes a DBN to extract specific abstract features, and learn joint representations from specific features with a MRBM, as illustrated in Fig. 1.

2.2 Long Short-Term Memory

Long Short Term Memory (LSTM) [4] is a recurrent neural network (RNN) and is well-suited to learn sequence data. LSTM units are often implemented in "block" with several "gates" to control the flow of information into or out of their memory.

3 The Proposed Model

The schema of two-stage temporal multimodal learning (TS-TML) is illustrated in Fig. 2. Instead of directly modeling the overall joint distribution of the whole time frames, we only extract and fuse static features by a MDBN without taking into account temporal dependencies at Stage I. Following Stage I, we will obtain the high-level common abstract concept features of multiple modalities. At Stage II, we transfer static features into corresponding dynamical features by jointly learning the temporal in-formation with a LSTM.

Compared with previous multimodal methods, our two-stage learning yields two main benefits: One is that it will dramatically decrease the training cost at Stage I because there is no need of distribution estimation of the whole temporal sequences, which stands as a striking contrast to RTMRBM. The other one is that it is more efficient to learn temporal information and dependencies in high-level abstract features than in low-level ones from scratch.

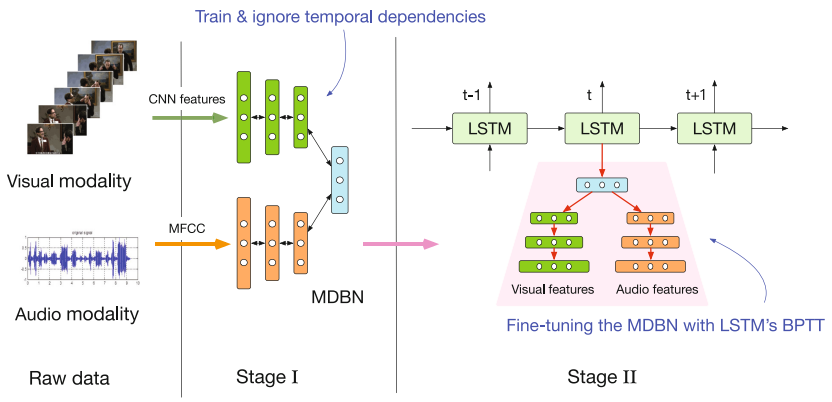

Fig. 2. Two-Stage Temporal Multimodal Learning (TS-TML)

3.1 Stage I

At Stage I, we employ a MDBN to extract static features from each modality and fuse them into common features J on the top of the MDBN, where $J \in \mathbb{R}^{N_{joint}}$, N_{joint} is the dimension of the common feature layer. Given a MDBN in Fig. 1, it joins two branches corresponding to two modalities (modal 1 and modal 2). Every two adjacent layers of the MDBN can be viewed as an RBM. The training of RBM depends on this inference and learning process. Firstly, we defined the joint distribution of adjacent layers. Let $v \in \mathbb{R}^{n_0}$ be the inputs of one modality, $h^1 \in \mathbb{R}^{n_1}$ denote the 1st hidden layer states and $h^2 \in \mathbb{R}^{n_2}$ denote the 2nd hidden layer states, and then we can formulate their joint distributions of RBM as follows:

$$P_\Theta(\nu, h^1) = \frac{1}{Z(\theta)} exp(\sum_{i=1}^{n_0} \sum_{j=1}^{n_1} \omega_{ij} \nu_i h_j^1 + \sum_{i=1}^{n_0} \nu_i \alpha_i + \sum_{j=1}^{n_1} h_j^1 \beta_j) \quad (1)$$

$$P_\Theta(h^1, h^2) = \frac{exp(\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \omega_{ij} h_j^2 h_i^1 + \sum_{i=1}^{n_1} h_i^1 \beta_i + \sum_{j=1}^{n_2} h_j^2 \gamma_j)}{Z(\theta)} \quad (2)$$

Similarly, the joint distribution between two modalities hidden states (h_1^2, h_2^2) and the joint hidden states J can be defined as:

$$P_\Theta(h_1^2, h_2^2, J) = \frac{exp((h_1^2)^T W_1 J + (h_2^2)^T W_2 J + (h_1^2)^T \gamma_1 + (h_2^2)^T \gamma_2 + J^T \sigma)}{Z(\theta)} \quad (3)$$

where h_1^2 denotes 2nd hidden layer states of modal 1 and h_2^2 denotes 2nd hidden layer states of modal 2, J denotes joint hidden units states on the top of two 2nd hidden layers, α, β, γ, σ are the bias of the input, the 1st hidden layer, the 2nd hidden layer, and the joint hidden units, respectively. W_1 and W_2 are the weights between the 2nd hidden layers corresponding to two modals and the joint hidden layer. $Z(\Theta)$ is the partition function.

Having obtained the formulation of joint distributions, we can train DBN layer by layer with Contrastive Divergence algorithm (CD-k) [3]:

Step1: Given visible layer's input $v \in \mathbb{R}^{n_v}$, sample hidden layer's value $h \in \mathbb{R}^{n_h}$ from the conditional distribution:

$$P(h_i = 1|v) = \frac{e^{-E(v,h)}}{\sum_v e^{-E(v,h)}} = \sigma(\sum_{j=0}^{n_v} \omega_{ij} v_j + c_i) \quad (4)$$

Step 2: Reconstruct visible layer's value v' with the sampled hidden layer state h:

$$P(v_j^{'} = 1|h) = \frac{e^{-E(v^{'},h)}}{\sum_h e^{-E(v',h)}} = \sigma(\sum_{i=0}^{n_h} \omega_{ij} h_i + b_i) \quad (5)$$

Repeating Step 1 and Step 2 for k times, we will obtain the last visible layer's value v^k. Then we use the data distribution and the model distribution

to compute the gradients of parameters, such as the gradients of weights be formulated by

$$\frac{\partial L(\theta)}{\partial \omega_{ij}} = \sum\nolimits_{v} P(h_j = 1|v)v_i - \sum\nolimits_{i=1} P(h_j = 1|v_i^k)v_i^k \tag{6}$$

3.2 Stage II

After the MDBN being pre-trained at stage I, we can feed joint features J into the top LSTM at each time step. At this stage, we aim to employ the temporal modeling capacity of LSTM to adjust the learned joint features J and build a temporal dependencies in its joint feature spaces. At each time step, we output a predicted label for the current frame. And the entire sequence classification performance will be averaged over all time steps. The training loss is formulated by

$$E = \frac{1}{N} \sum_{n}^{N} \sum_{k}^{K} y_{nk} \log(O_{nk}) \tag{7}$$

where n denotes the n-th frame ($n = 1, 2, \ldots, N$), $y \in \mathbb{R}^k$ is one-hot vector and denotes groundtruth label. O_{nk} is the output at k-th unit at the time n. According to this average category loss, we adopt the back propagation through time (BPTT) algorithm to train the whole system, which includes the fine-tuning of the MDBN. In summary, we present our learning processes in Algorithm 1.

Algorithm 1. Two-Stage Temporal Multimodal Learning.

Require:
Number of modal, M;
Depth of DBN, L ($L = 3$ in Fig. 1);
Length of multimodal sequences, N;
Label sequences, y;

1: **for** modal m :1→ M **do**
2: **for** i : 1→ $L - 1$ do: **do**
3: Train the weights between h_i and h_{i+1} as a RBM with CD-k;
4: **end for;**
5: **end for.**
6: Fix all pre-trained weights for each modal.
7: Obtain the L-th hidden states in each modal and learn a joint layer in a MRBM as the top of MDBN, which has the same training process as RBM.
8: Obtain N-length static joint representations for given multimodal sequences.
9: Input joint representation sequences and train LSTM, which includes fine-tuning the MDBN at each time.

Fig. 3. Speaker recognition on BBT data. From first line to second line, they are two speakers identified by our method: Shelton and Lenord, respectively.

Table 1. The speaker recognition accuracy (%) on Big Bang Theory dataset.

Models	Size of sliding windows					
	0.5	1.0	1.5	2.0	2.5	3.0
MLR (2013) [2]	-	-	-	-	77.8	-
MRF (2012) [14]	-	-	-	-	80.8	-
Multimodal CNN (2015) [7]	74.93	77.24	79.35	82.12	82.8	83.42
Multimodal LSTM (2016) [11]	86.59	89.00	90.45	90.84	91.1	91.38
Ours TS-TML	**98.01**	**98.22**	**98.45**	**98.56**	**99.20**	**99.30**

4 Experiments

4.1 Speaker Recognition

This task is to identify a person who is currently talking in a continuous multi-character conversation scene video and identify the person's identity through the person's facial features and vocal characteristics. We choose the Big Bang Theory (BBT) dataset as our experimental data. Due to illumination variations and blurring information of human face images, speaker recognition from BBT is a very challenging task.

We first use the face detection algorithm to locate faces in each video frame. These frames are manually annotated by five classes: Shelton, Lenord, Howard, Raj and Penny. Consequently, we have 310,000 consecutive hand-labeled frames of these five characters. For the audio data, we extract each character's speech in the video, and combine them into one audio file, and label them according to corresponding characters in image sequences. We use the first 6 episodes from the second season for training, and the first 6 episodes from the first season for testing.

We first pre-train a Caffenet with BBT dataset, and extract image features with this Caffenet. As for the audio data, we use the 20-millisecond-sliding-window-10-millisecond-step MFCCs [9] feature to preprocess them. Each sequence length is 49.

We set up 6 different sliding windows with 0.5 s, 1.0 s, 1.5 s, 2.0 s, 2.5 s, and 3.0 s, respectively. The results compared with the other models are shown in Table 1.

From Table 1, we can see that our model achieves a far better performance than other multimodal methods. In various sliding windows, our method improves 7.74%–11.42% accuracies. It proves that learning temporal information in high-level abstract features is more efficient than leaning it in low-level ones. In addition, with the increasing size of a time window, the accuracy of multimodal methods improves as well, indicating that more information will help to improve the performance of multimodal learning.

To present our results visually, we demonstrate a few frames obtained by our method in Fig. 3. Our method can identify the two main characters in BBT with audiovisual features in various scenarios.

Finally, we compare our method with our baseline models and LSTM which only employ visual or audio data. The results are listed in Table 2. It is noteworthy that, our temporal multimodal method outperforms the single modality one, which is also consistent with previous multimodal methods' results [6,13].

Table 2. Accuracy (%) of single modality and multimodal methods.

Models	Visual	Audio	Audio-visual
LSTM	88.27	46.4	-
Ours (single modality)	90.1	51	-
Ours TS-TML	-	-	**98.01**

Table 3. The average accuracy (%) of speech recognition on AVLetters.

Models	Acc.
MDAE (2015) [5]	62.90
CRBM (2014) [1]	64.8
RTMRBM (2016) [6]	66.04
Ours TS-TML	**66.51**

4.2 Speech Recognition

This task is to recognize what a certain person is speaking, given the lip motion and dubbing in a video clip.

We choose the AVLetters dataset with 780 short video clips. In each video there is a person reading letters from A to Z. There are 10 individuals, and each

one reads the 26 letters for three times. The size of lip images in all frames is 60×80. The dataset also provides MFCC features of the audio data. We take the first two as a training set, and the last one as a testing set. Our experiment is speaker-dependent. To match the visual and audio frames' length, we input one image frame with four audio frames simultaneously into our model.

Our method is compared with the Multimodal Deep Auto Encoder (MDAE) [5], Conditional Restricted Boltzmann Machine (CRBM) [1], RTMRBM [6] on AVLet-ters dataset, and the results are shown in Table 3.

From Table 3, we can see that our method outperforms other multimodal models on this speech recognition task. Compared with non-temporal model MDAE and single temporal modality model CRBM, our method has a much better performance. Furthermore, instead of directly modeling the overall joint distribution of the whole video and audio frames as RTMRBM did, we only extract and fuse static features of each video and audio frame without considering temporal dependencies at Stage I and learn the temporal information at Stage II. The results show that our two-stage learning strategy has a better performance than RTMRBM. It demonstrates that learning temporal information and dependencies in high-level abstract features is more efficient than learning them in low-level ones from scratch.

4.3 Discussions of Computational Complexity

All our experiments are conducted on machine with an Intel Core i5-6500, 3.20-GHz CPU 32-GB RAM and a GeForce GTX 980-Ti 6G. Take AVLetters for example, there are 20800 multimodal data in total (dim of video frame: 4096, dim of audio: 104). We spent about 32s/epoch to pretrain the MDBN, 10s/epoch to train the LSTM and fine-tune our whole system. From these observations, we find that learning static joint features costs most of runtime, which depends on the complexity of MDBN. Compared with the strategy of learning joint distribution over the whole multimodal sequences, our learning method adopts the idea of transferring from static features to dynamic ones in two stages, which largely reduces the training cost of estimation the joint distribution.

5 Conclusion

We have proposed a two-stage learning to model the temporal information in multi-modal sequences. Instead of directly modeling the overall joint distribution of the whole time frames, our method merely extracts and fuses static features of each frame without considering temporal dependencies at Stage I, while it learns the temporal information at Stage II. These two stages make it easier to train the model and decrease the training cost. Our experimental results show that the proposed method performs better than single modality methods, non-temporal multimodal networks, as well as other temporal multimodal methods in the tasks of speaker and speech recognition.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (Grant No. 61502174, 61402181), the Natural Science Foundation of Guangdong Province (Grant No. S2012010009961, 2015A030313215), the Science and Technology Planning Project of Guangdong Province (Grant No. 2016A040403046), the Guangzhou Science and Technology Planning Project (Grant No. 201704030051, 2014J4100006), the Opening Project of Guangdong Province Key Laboratory of Big Data Analysis and Processing (Grant No. 2017014), and the Fundamental Research Funds for the Central Universities (Grant No. D2153950).

References

1. Amer, M.R., Siddiquie, B., Khan, S., Divakaran, A., Sawhney, H.: Multimodal fusion using dynamic hybrid models. In: IEEE Winter Conference on Applications of Computer Vision, pp. 556–563, March 2014
2. Bauml, M., Tapaswi, M., Stiefelhagen, R.: Semi-supervised learning with constraints for person identification in multimedia data. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2013
3. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
5. Hong, C., Yu, J., Wan, J., Tao, D., Wang, M.: Multimodal deep autoencoder for human pose recovery. IEEE Trans. Image Process. **24**(12), 5659–5670 (2015)
6. Hu, D., Li, X., Lu, X.: Temporal multimodal learning in audiovisual speech recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
7. Hu, Y., Ren, J.S., Dai, J., Yuan, C., Xu, L., Wang, W.: Deep multimodal speaker naming. In: Proceedings of the 23rd ACM International Conference on Multimedia, MM 2015, NY, USA, pp. 1107–1110 (2015). doi:10.1145/2733373.2806293
8. Huang, J., Kingsbury, B.: Audio-visual deep learning for noise robust speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7596–7599 (2013)
9. Jamil, M., Rahman, G.R.S.: Speaker identification using Mel frequency cepstral coefficients (2004)
10. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28–July 2, pp. 689–696 (2011)
11. Ren, J., Hu, Y., Tai, Y.W., Wang, C., Xu, L., Sun, W., Yan, Q.: Look, listen and learn – a multimodal LSTM for speaker identification. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI 2016, pp. 3581–3587. AAAI Press (2016)
12. Sohn, K., Shang, W., Lee, H.: Improved multimodal deep learning with variation of information. In: International Conference on Neural Information Processing Systems, pp. 2141–2149 (2014)
13. Srivastava, N., Salakhutdinov, R.: Multimodal learning with deep boltzmann machines. J. Mach. Learn. Res. **15**, 2949–2980 (2014). http://jmlr.org/papers/v15/srivastava14b.html
14. Stiefelhagen, R.: "Knock! knock! who is it?" probabilistic person identification in TV-series. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR 2012, pp. 2658–2665 (2012). http://dl.acm.org/citation.cfm?id=2354409.2354974