Proposing Multimodal Integration Model Using LSTM and Autoencoder

Wataru Noguchi
Department of Engineering
Hokkaido University
North14 West9, Sapporo,
060-0814, Japan

Hiroyuki Iizuka
Graduate School of
Information Science and
Technology
Hokkaido University
North14 West9, Sapporo,
060-0814, Japan

Masahito Yamamoto
Graduate School of
Information Science and
Technology
Hokkaido University
North14 West9, Sapporo,
060-0814, Japan

{noguchi, iizuka, masahito}@complex.ist.hokudai.ac.jp

ABSTRACT

We propose an architecture of neural network that can learn and integrate sequential multimodal information using Long Short Term Memory. Our model consists of encoder and decoder LSTMs and multimodal autoencoder. For integrating sequential multimodal information, firstly, the encoder LSTM encodes a sequential input to a fixed range feature vector for each modality. Secondly, the multimodal autoencoder integrates the feature vectors from each modality and generate a fused feature vector which contains sequential multimodal information in a mixed form. The original feature vectors from each modality are re-generated from the fused feature vector in the multimodal autoencoder. The decoder LSTM decodes the sequential inputs from the regenerated feature vector. Our model is trained with the visual and motion sequences of humans and is tested by recall tasks. The experimental results show that our model can learn and remember the sequential multimodal inputs and decrease the ambiguity generated at the learning stage of LSTMs using integrated multimodal information. Our model can also recall the visual sequences from the only motion sequences and vice versa.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms

Keywords

multimodal integration, deep learning, autoencoder, Long Short Term Memory

1. INTRODUCTION

Human use information from multiple sources in order to recognize various representation, such as objects, speech, motion, etc. In machine learning, also, using multimodal information for recognition tasks is efficient, because multiple modalities make recognition more robust than single modality.

Recently, deep neural networks improved accuracy in image recognition, and several researches show multimodal integration learning using deep neural network is efficient. While deep network has significantly improved static image learning, sequential information learning such as video and speech learning is still a challenging task. Studying more efficient deep network learning architecture for temporal information is catching great attention. Ngiam et al. used multimodal deep autoencoder architecture for speech recognition task with mouth motion [6]. In their model, information from audio and vision is fused in a central hidden layer in a deep autoencoder. The multimodal representation can be constructed in the layer for the recognition task. Their model shows good recognition performance in the task, however the length of time series information which can be learnt is restricted to the network input size.

Noda et al. studied multimodal learning for long time series information of sensory-motor coordination in real-world robotics [7]. They used time-delay neural network (TDNN) architecture dealing with long time series. Recurrent neural network (RNN) is often used for long time series learning, however unfortunately RNN's learning is slow and unstable, particularly in high dimensional input like videos. Unlike RNN, TDNN does not have any recurrent mechanisms. TDNN's learning becomes stable even in high dimensional input. This is why their model works well. However TDNN also has limitation that it cannot remember context information longer than input length in principle.

The Long Short Term Memory (LSTM) architecture, which is a kind of RNN, has shown better performance than simple RNN. Srivastava et al. proposed unsupervised video learning architecture using LSTM ([9]) based on the model proposed by Sutskever et al. [11]. The input to the model is raw pixel data of the images, which means that any dimension reduction techniques are not applied. They showed that the LSTM

BICT 2015, December 03-05, New York City, United States

DOI 10.4108/eai.3-12-2015.2262505

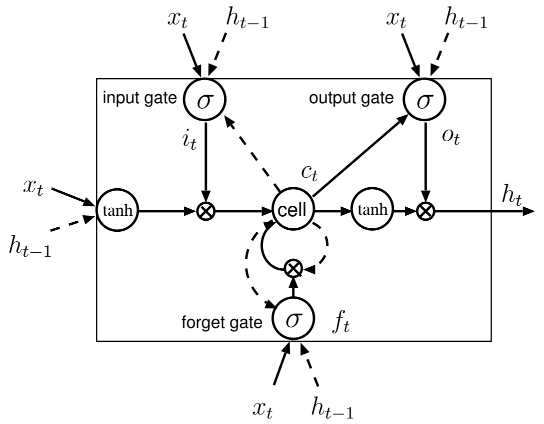

Figure 1: LSTM unit [2].

memorizes and recalls the images with a high precision in spite of using high dimensional inputs.

The LSTM autoencoder model has no limitation for sequential information learning different from TDNN and shows good performance even for high dimensional inputs. Therefore, we combine the LSTM architecture to learn sequential inputs with a multimodal autoencoder to integrate different modality information in order to realize temporal multimodal representation. We use vision and motion as learning modalities, having a motivation to apply our study to imitation learning that requires fusing both modalities in future. We evaluate our model in terms of cross-modal recall ability. Cross-modal recall is another aspect of effectiveness of multimodal learning, because it enables generating missing modality from the other and more flexible situation awareness by using information across modalities.

2. MODEL

2.1 Long Short Term Memory

RNN is an artificial neural network model that has feedback connections in the hidden units. Because the previous states in the hidden units are used as inputs, RNN can store historical information like memory and can solve context-dependent tasks with the architecture. The popular method of training RNN is gradient descent such as backpropagation through time (BPTT) [12] and real time recurrent learning (RTRL) [8]. The gradient based training of RNN has a problem that derivatives propagated via recurrent connections become too small or too large. This vanishing and explosion gradient problem makes learning of RNN difficult.

Long Short Term Memory (LSTM) is a special type of RNN architecture to overcome the vanishing gradient problem [4]. The schematic view of LSTM is shown in Fig. 1. LSTM units receive external inputs and generate hidden outputs. LSTM consists of three gates (input, output, and forget gates) and a memory cell. The gates and memory cell are internally connected with weighted links, and the gates are also connected with external sources, which are current sequential inputs, $\boldsymbol{x}_t$ and previous hidden states, $\boldsymbol{h}_{t-1}$. The hidden output, $\boldsymbol{h}_t$, is calculated from $\boldsymbol{x}_t$, $\boldsymbol{h}_{t-1}$, and previous state of the memory cell, $\boldsymbol{c}_{t-1}$. The equations of LSTM can be expressed as follows.

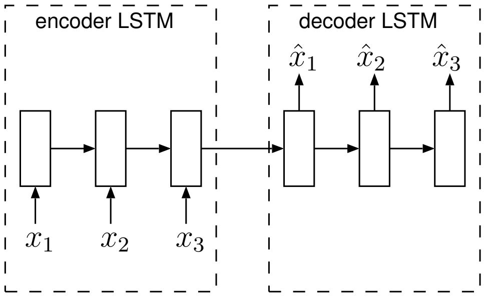

Figure 2: LSTM autoencoder.

$$\boldsymbol{i}_t = \sigma\left(\mathbf{W}_{xi}\boldsymbol{x}_t + \mathbf{W}_{hi}\boldsymbol{h}_{t-1} + \mathbf{W}_{ci}\boldsymbol{c}_{t-1} + \boldsymbol{b}_i\right) \quad (1)$$

$$\boldsymbol{f}_t = \sigma\left(\mathbf{W}_{xf}\boldsymbol{x}_t + \mathbf{W}_{hf}\boldsymbol{h}_{t-1} + \mathbf{W}_{cf}\boldsymbol{c}_{t-1} + \boldsymbol{b}_f\right) \quad (2)$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t\boldsymbol{c}_{t-1} + \boldsymbol{i}_t \tanh\left(\mathbf{W}_{xc}\boldsymbol{x}_t + \mathbf{W}_{hc}\boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right) \quad (3)$$

$$\boldsymbol{o}_t = \sigma\left(\mathbf{W}_{xo}\boldsymbol{x}_t + \mathbf{W}_{ho}\boldsymbol{h}_{t-1} + \mathbf{W}_{co}\boldsymbol{c}_t + \boldsymbol{b}_o\right) \quad (4)$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \tanh\left(\boldsymbol{c}_t\right) \quad (5)$$

where σ is the sigmoid function, $\boldsymbol{i}$, $\boldsymbol{f}$, $\boldsymbol{o}$, $\boldsymbol{c}$ and $\boldsymbol{h}$ are the input, forget, output gates, memory cell and hidden activation vectors, respectively.

The weights of the connection from α to β are denoted by $\mathbf{W}_{\alpha\beta}$, which is a matrix (e.g. $\mathbf{W}_{hf}$ means hidden-forget gate weight matrix). The bias terms are denoted by $\boldsymbol{b}_\gamma$ (e.g. $\boldsymbol{b}_i$ means the bias for the input gate activation). The role of gates is as follows. Unless the input gate is open, the state of memory cell is not overwritten. Also, unless the output gate is open, the activation of the network is not transmitted. This prevents the LSTM from storing useless or noisy input information. The forget gate releases the memory that becomes no longer required in order to store new useful memory. These mechanisms work to suppress the vanishing gradient problem. To prevent the explosion problem, we clipped the derivatives that are propagated via recurrent connections in the LSTM within a predefined range. This technique is used by a predecessor [2]. The details of LSTM are described in [1].

For obtaining the output, an additional activation layer is attached to LSTM unit. $\hat{\boldsymbol{x}}_t$ is calculated as follows.

$$\hat{\boldsymbol{x}}_t = a\left(\mathbf{W}_{\hat{x}h}\boldsymbol{h}_t + \boldsymbol{b}_{\hat{x}}\right) \quad (6)$$

where a is an activation function. There are two kinds of activation functions, i.e., sigmoid and linear function, one of which is chosen for the different modalities. The detail is described later in Section 3.

2.2 LSTM Autoencoder Model

LSTM autoencoder model proposed by Srivastava et al. [9]. consists of encoder LSTM and decoder LSTM. The encoder LSTM receives input sequences and encode them to a fixed range feature vector as the normal LSTM generates hidden outputs from the external inputs. Then, the decoder LSTM receives the feature vector and decodes it into the original input sequences as the autoencoder. The schematic view of LSTM autoencoder is shown in Fig. 2. Srivastava et al. also proposed future predictor model where the decoder

LSTM predicts the future input sequences instead of the original inputs. These models are based on the sequence to sequence learning framework [11]. The sequence to sequence learning allows to use different lengths of input and output sequences, however it should be same in the LSTM autoencoder. Because it is reported that reversed order decoding makes learning easier [9, 11], it is used in our model. For example, if the input sequence is $\{\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \ldots, \boldsymbol{x}_{t+k}\}$, then the target sequence is $\{\boldsymbol{x}_{t+k}, \boldsymbol{x}_{t+k-1}, \ldots, \boldsymbol{x}_t\}$. The reason for this is that because the last input has a stronger correlation to the feature vector, which is sent to the decoder, than the earlier inputs, decoding the feature vector to the last input is easier than to the first input. Recalling the earlier inputs can be performed through gradual memory state transitions from the last to the first inputs.

2.3 Multimodal LSTM autoencoder

For multimodal learning, we use encoder and decoder LSTMs for each modality. The modalities are integrated by a multimodal autoencoder, which consists of three layers, i.e., input, hidden, and output layers. The schematic view of our proposed model is shown in Fig. 3. Our model consists of encoder part and decoder part similar to vanilla autoencoder. The advantages of the multimodal autoencoder are as follows. One is that memorized patterns can be reconstructed from incomplete or single modality information. Another is that the multimodal encoder can be naturally combined with the encoder and decoder LSTMs and the whole structure can be learned with the simple gradient descent method. We call our model multimodal LSTM autoencoder.

The multimodal LSTM autoencoder can encode the sequential inputs of multiple modalities to a fused feature vector and decode the vector to the separated sequential outputs of each modality. There are four stages of encoding or decoding in the multimodal LSTM autoencoder. The first stage is encoding the sequence to fixed range feature vector by the encoder LSTM for each modality. The feature vectors are combined and converted to a fused feature vector once in the multimodal autoencoder as the second stage. The multimodal autoencoder also decode the fused feature vector to the two separated feature vectors which are sent to the decoder LSTMs at the third stage. At the last stage, the decoder LSTM reproduces the original input sequences from the feature vectors for each modality.

The training procedure of our model has three stages to train the encoder and decoder LSTMs and multimodal encoder in a separate and integrated ways. Firstly, the encoder and decoder LSTMs are trained to be able to encode an input sequence to a feature vector and to decode the feature vector to the input sequence in a usual way of LSTM learning. Secondly, only multimodal autoencoder is trained to encode the feature vectors of two modalities, which is passed by the encoder LSTMs, to the fused feature vector and decode the vector to the original vectors of two modalities. By the second stage, the encoder and decoder LSTMs and multimodal autoencoder can be used as a multimodal LSTM autoencoder if learning of both LSTMs and multimodal autoencoder is perfect. However, because the multimodal autoencoder cannot completely recall the feature vectors, recalling ability of simply connected LSTMs and multimodal autoencoder becomes poor. Therefore, the encoder and decoder LSTMs and the multimodal autoencoder are trained as a whole network as the third stage. Although training entire network in a single stage is possible in principle, the training is more difficult and slower than the three stages training in practice.

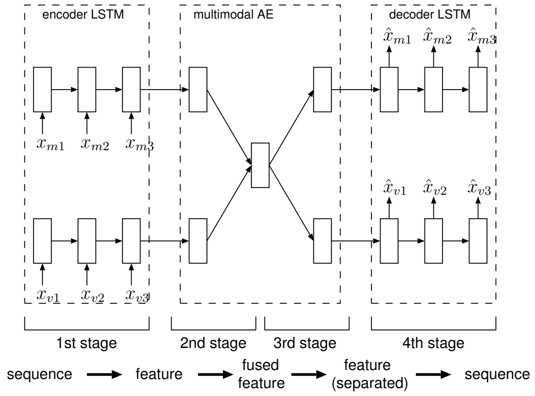
encoder LSTM
multimodal AE
decoder LSTM
$\hat{x}_{m1}$ $\hat{x}_{m2}$ $\hat{x}_{m3}$
x_{m1} x_{m2} x_{m3}
$\hat{x}_{v1}$ $\hat{x}_{v2}$ $\hat{x}_{v3}$
x_{v1} x_{v2} x_{v3}
1st stage
2nd stage
3rd stage
4th stage
sequence
feature
fused feature
feature (separated)
sequence

Figure 3: Multimodal LSTM autoencoder.

After training, it is expected that the fused feature vector encoded by the multimodal autoencoder contains information of multiple modalities in a mixed form. However, in case that the number of units in the hidden layer of multimodal autoencoder is same as the sum of each modality's or more, the vector might be merely identical mapping of modalities, and that ends up result that we get no benefit of using multimodal integration approach. By reducing the number of nodes in the hidden layer of multimodal autoencoder less than the sum of each modality's, we can force the output to be mixed.

3. EXPERIMENTS

In order to evaluate our proposed method, it is applied to crossmodal recalling and recognition between motion and vision.

3.1 Dataset

We used motion capture (mocap) data from CMU Graphics Lab Motion Capture Database for motion patterns (see acknowledgments). The mocap data consists of time series of skeleton data of human figure, i.e., joint angles of body parts, and motion data that contains translational and rotational movements of a reference point of the human figure. The endpoints of the other body parts can be sequentially calculated from the reference. Because the learning effect of crossmodality is evaluated here, we eliminate the translational movements of the reference point. Because the degree of freedom of whole body is 56 and the rotational degree of freedom of the reference point is three, the dimensions of input motion patterns become 59. The visual images for vision are created from the mocap data manually. The body parts are colored with white and black for the background. The frame size of the visual images is 64×64. As the translational movements of the reference are eliminated, the position of the reference point is fixed at the center of the frame. The examples of input patterns for vision and motion are shown in Fig. 4. Several people participated in CMU database as subjects to capture motion data, and data are separated by

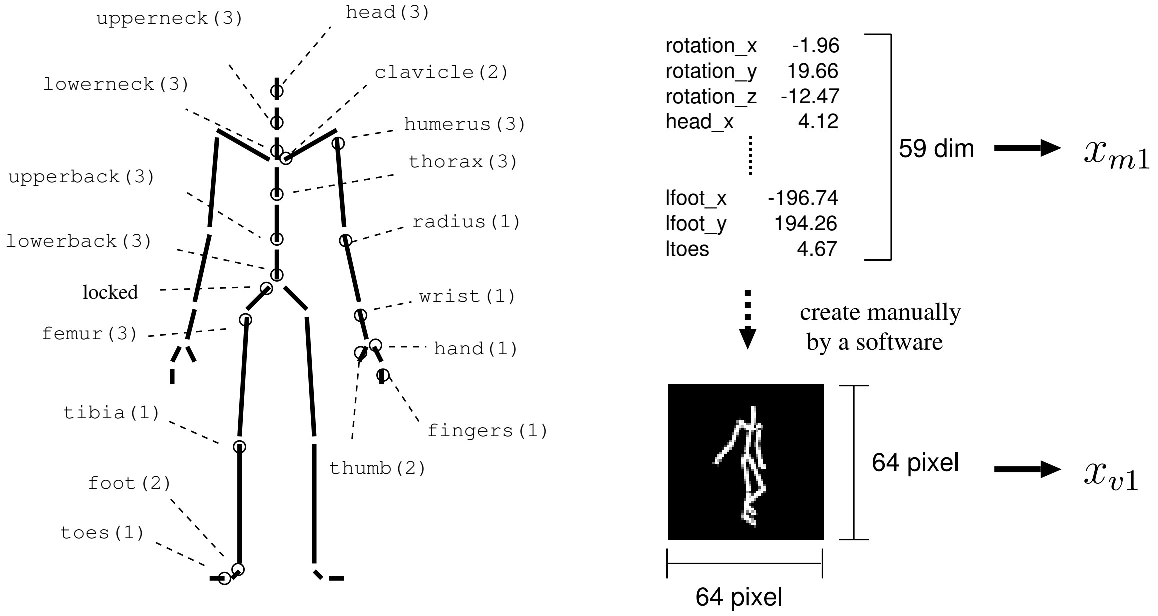

Figure 4: Left: Joints and degrees of freedom of the body skeleton in the motion patterns. Right: Example of 1-shot motion and visual data (denoted by x_{m1} and x_{v1} respectively). The motion and visual image sequences consist of multiple data of this.

subjects number. We used data of subject #1 for training our model. The data contains 14 trials. One trial is composed of a continuous motion sequence that possibly contains more than two kinds of motion. Data have been captured in 120 fps. In the original fps data, the differences between succeeded visual images or motions are often so small that the data are downsampled to 8 fps for input visual and motion data. The total number of frames in downsampled data is about 3,600 frames. The length of the recalled data is set to 10, which corresponds to 1.125 sec. Finally, we obtain 360 input data sequences of vision and motion for training.

3.2 Memorizing and Recalling Visual Images

At the first stage of training our model, we train the encoder and decoder LSTMs that receive and recall visual images. As described before, the activation function of eq. (6) for the visual images is a sigmoid activation function. For the learning, we used stochastic gradient descent with learning rate of 0.7. To compute the gradient, the reconstruction error is computed by cross entropy loss function. The size of inputs for the LSTM is 4096, and we set the number of hidden units of the LSTM to 1024. The parameters are updated on mini-batch method, and the size of mini-batch is 10.

Figure 5 shows the recall error of the output patterns to the target input patterns. The all errors gradually decrease and converge by 250 epochs[1]. The LSTM recalls the last-input visual image first and the fist-input in the end (reverse order). That is why the error of the first recalling output become the smallest. Figure 6 (the second row from the top) shows the visual images recalled by the best trained LSTM at 250 epochs. The left and right most images are the first and last outputs of LSTM, respectively. In the opposite direction, the left most of the target is the last input. It is remarkable that the images of recalled outputs are similar to the target even the middle recalled images are blurry. It means that the training has been reasonably successful and the LSTM can recall the sequential visual information.

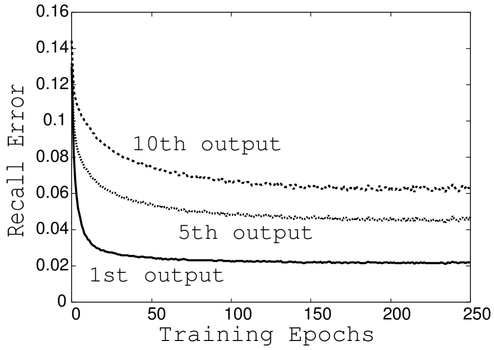

Figure 5: Recall errors during training of visual LSTM autoencoder.

3.3 Memorizing and Recalling Motion Data

For the motion data, another LSTM is trained in the same way as the visual images except for a few implementations. Because a value of the motion data has no limit, sigmoid activation function which has bound in $(0, 1)$ and cross entropy loss function which assumes the input in $(0, 1)$, are not suitable for the implementations. Therefore, we use a linear activation function in eq. (6) and euclidean loss function to train on the motion data and recall the data. The

[1] In one epoch, a network is trained on each data in training set once.

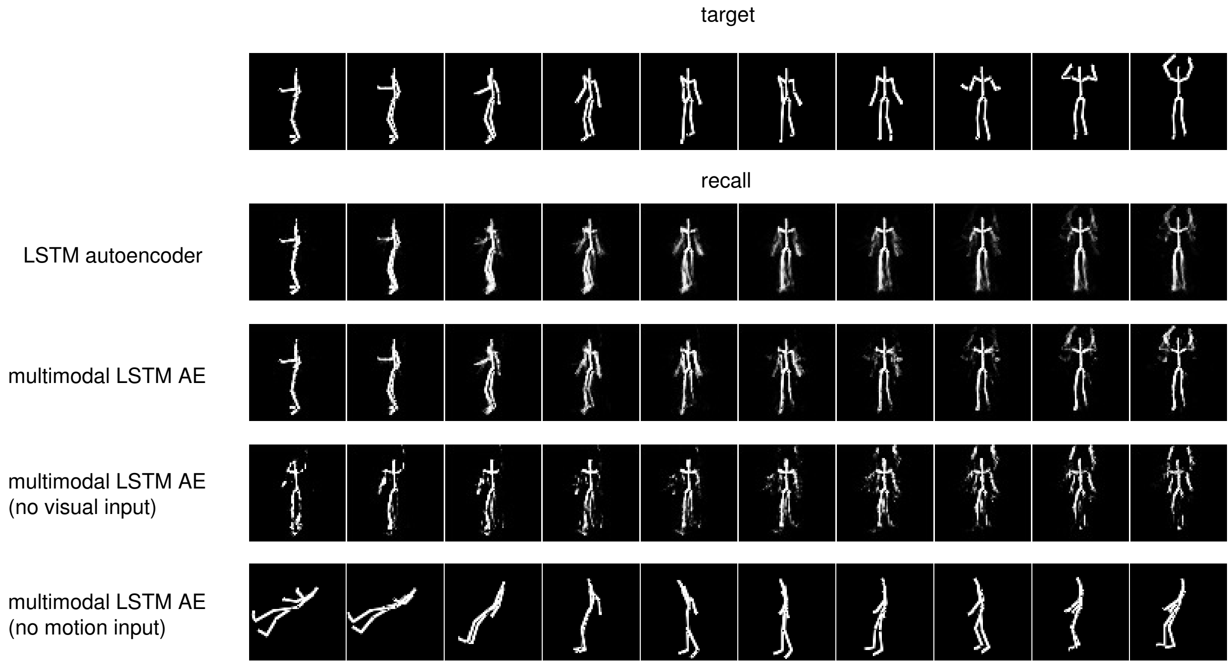

Figure 6: Examples of the target sequence, recalled visual images by LSTM autoencoder (AE) and multimodal LSTM AE, visual images recalled from only motion, and visualized motion sequences recalled from only vision (See the main text for the details).

learning rate is 0.7 for the weights that direct inward. For the weights of outward connections (i.e. the connections to the linear activation layer), the learning rate is 0.007. That is because the linear activation has no limit in their values and updating the parameters should be performed carefully. The size of inputs for the LSTM is 59, and the number of hidden units is 512. Although the number of hidden units is relatively large to the input size, the parameter is determined to balance between vision and motion at the multimodal integration stage. If the number of hidden units of motion LSTM is much less than that of vision LSTM, motion modality might not be able to contribute the fused representation in multimodal autoencoder efficiently.

Figure 7 shows recall error as training epochs, and it shows that the training has been done successfully. All errors become very small and the training has been done successfully. Though it is possible to visualize recalled motion data in the same way as creating video data, almost no difference can be seen between the images of target and recall, and the visualization is not shown.

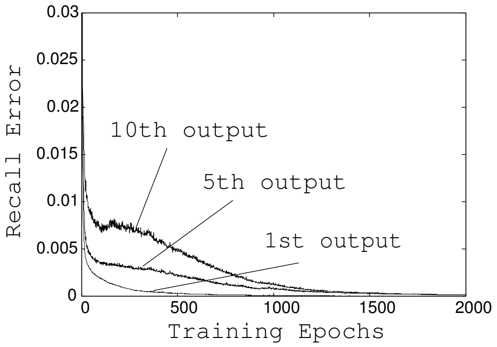

Figure 7: Recall errors during training of motion LSTM autoencoder.

3.4 Experiment on Multimodal Integration

The trained visual and motion LSTMs are integrated by a multimodal autoencoder. Firstly, the multimodal autoencoder is trained to receive and reconstruct the feature vectors which is the output from the trained encoder LSTM. In this stage, the parameters of the only multimodal autoencoder are updated to reconstruct the feature vectors. After that, the trained multimodal autoencoder is connected to the encoder and decoder LSTMs (also trained in previous section), and we train them as a whole network. These greedy layer-wise training process is the practical way of the learning on deep network. The number of nodes at the hidden layer of the multimodal autoencoder is 256. For the multimodal autoencoder, the learning rate is 0.7. For the encoder and decoder LSTMs, the learning rate is also 0.7 except for the linear output layer of the decoder LSTM for motion described in the above section, and we decrease the rate linearly 0.7 to 0.1 after the monitoring errors looks converged.

Figure 8 shows the errors of the visual images and motion recalled by the multimodal LSTM autoencoder during training as a whole deep network. The recall error of visual sequence is lower than that showed in the previous section even before the learning rate has been annealed. However, the error

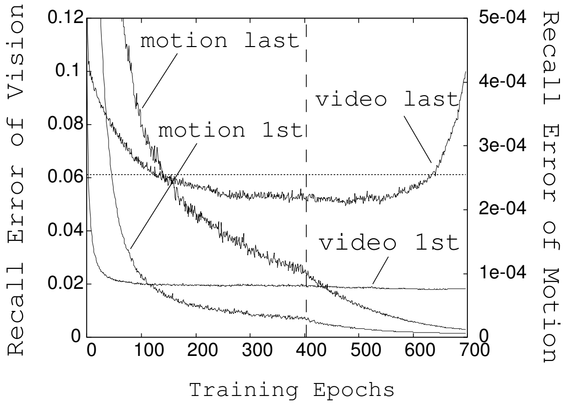

Figure 8: Recall errors during training of multimodal LSTM autoencoder. The vertical dashed line indicates the epoch when learning rate annealing starts. The horizontal dashed line indicates the best recall error of the LSTM autoencoder for visual images.

of the motion sequence becomes higher. While the error of motion sequence is decreasing after annealing the learning rate, that of video sequence is increasing. We discuss about the reason of this in Section 4.

3.4.1 Recalling from multiple modalities

For the evaluation of the multimodal recall, we use the trained model at 550 epoch which is the time before the error of visual images start increasing. In Fig. 6, the third row shows the recalled visual images of multimodal LSTM autoencoder when the motion and visual images are given at the same time. The recall by the multimodal LSTM autoencoder is clearer than the recall of the single modal information by the encoder and decoder LSTM. The details of the arms and legs in the human figure are recalled better. These results show that the two modalities are integrated by the hidden unit of the multimodal autoencoder and the motion input sequences contribute to disambiguate the recalled visual image sequences.

3.4.2 Recalling of Both Vision and Motion from Either Modality

The multimodal LSTM autoencoder can be also used to generate missing inputs. For example, even if the input sequences of visual images are not given, the visual images can be recalled from the only motion sequences. However, recall obtained with missing input is much noisy and not similar to the target. To make the noisy output clear to some extent, we can use the output as the input which is absent. By iteratively inputting the output to the missing modality, the output become more closer to the target in some cases. It should be noted that there are some cases that the output is still far from the target.

The forth and fifth row in Fig. 6 show the results of recalling from a single modal information, i.e., visual images recalled from motion sequences and motion sequences recalled from visual image sequences, respectively. The results of motion sequences are visualized in the same way as creating visual images from mocap data. Although the recall from only motion input (the forth row in Fig. 6) is very blurry and noisy, the feature of the target can be observed (hands-up in the recalled images). In the case of recall from only visual images (the fifth row in Fig. 6), while the facing direction in the recalled visual images is correct however it fails to recall the features of hands-up and turning on the way. Figure 9 shows another example of recall from missing visual images. The facing direction of the visual images recalled from the motion is wrong, however, it is interesting that the images of hands-up is successfully recalled. This is probably because the distance between turning away and turning front is far in the feature space. As the first recall is noisy and sometimes close to another training input rather than the target, once the feature is trapped in unfavorable subspace, which is far from where the target is, it is difficult to escape from there. Figure 10 shows changes of the recall errors of the visual images and motion while obtaining the result shown in Fig. 9 when the recalled outputs are fed back into the input of the multimodal LSTM autoencoder repeatedly. While the errors of the motion become nearly zero, that of visual images increased rather than decreased. This is caused due to the same reason described just before.

3.5 Generalization Capability of Trained Model

In this section, we test generalization capability of trained model. We prepare extra data for the test from data of subject #2 in CMU database, on which the model is not trained. Although the test data is unknown visual images and motion patterns, if the trained model has high generalization ability, the model can output the visual images from only motion patterns or motion patterns from only visual image inputs. Figure 11 shows the recall of visual images without input visual images and visualization of motion sequences without input motion sequences. The recalled sequence captures at most one feature of target, and the feature appears in quite different style. The sample shown in the Fig.11 is one of the most successful sample, and the other recall is not good. It show that generalization ability of our model is fairly low.

4. DISCUSSION

Our results showed that the proposed model can integrate the temporal multimodal sequences and can recall missing sequential information from either modality. However, there is an asymmetric ability of recalling between vision and motion. The recalling from motion to vision is better.

The encoder LSTM is trained to map the input sequences to the fixed dimensional feature vectors. At this stage, the different sequences should be mapped to different feature vectors with certain distances, but it is not an easy task for high dimensional inputs such as visual image sequences. On the other hand, the number of dimensions of motion sequence data is much less than the vision and the differences of trained motions are relatively large. In fact, the gap between the recall errors of the first and last outputs for vision is bigger than for motion. Therefore, there is a difference of LSTM abilities between vision and motion to map the input sequences to the feature vectors. Those feature vectors are fused in the multimodal autoencoder, however it is just a multi-layer autoencoder. If the feature vectors are not clearly separated, the fused feature vectors and reconstructed fea-

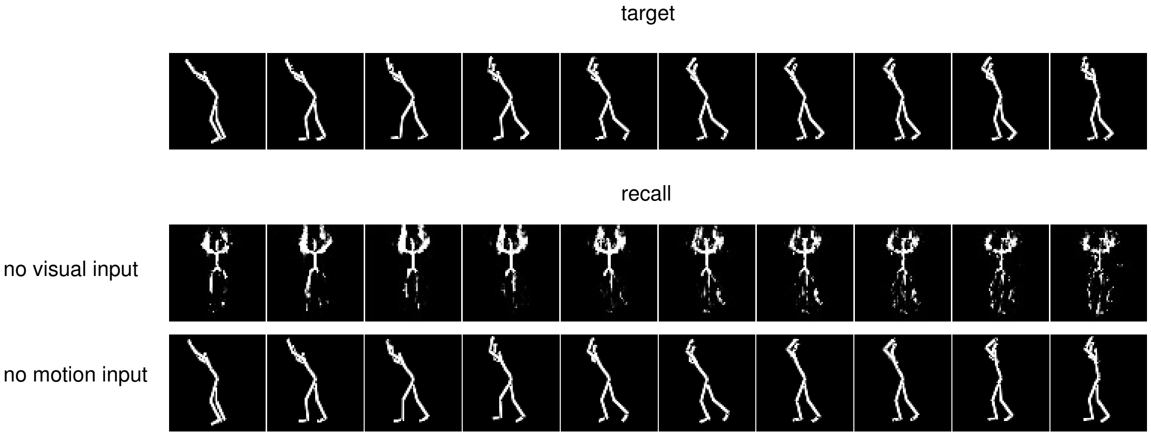

Figure 9: Another example of visual images recalled from the only motion sequence and visualized motion sequence recalled from the only visual image sequence.

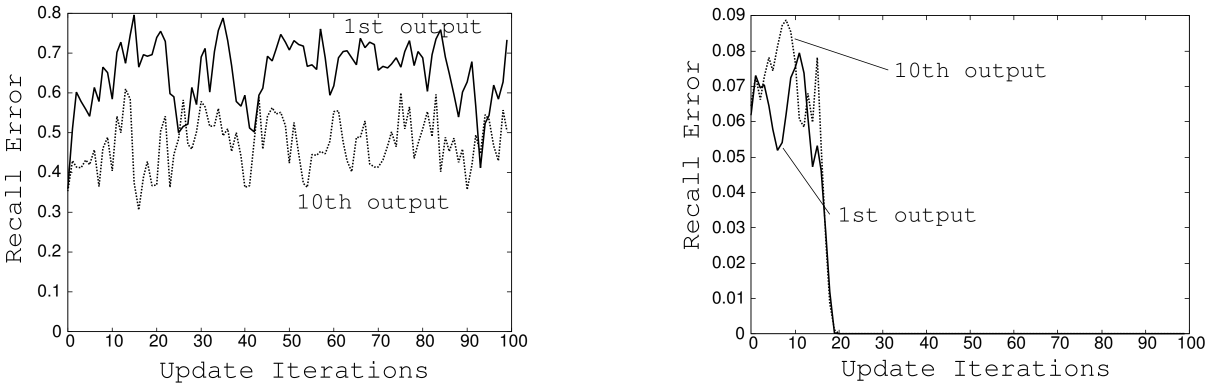

Figure 10: Recall errors of the visual images (left) and motion sequences (right) when the recalled outputs are fed back into the input of the multimodal LSTM autoencoder repeatedly, and the obtainend result is shown in Fig. 9.

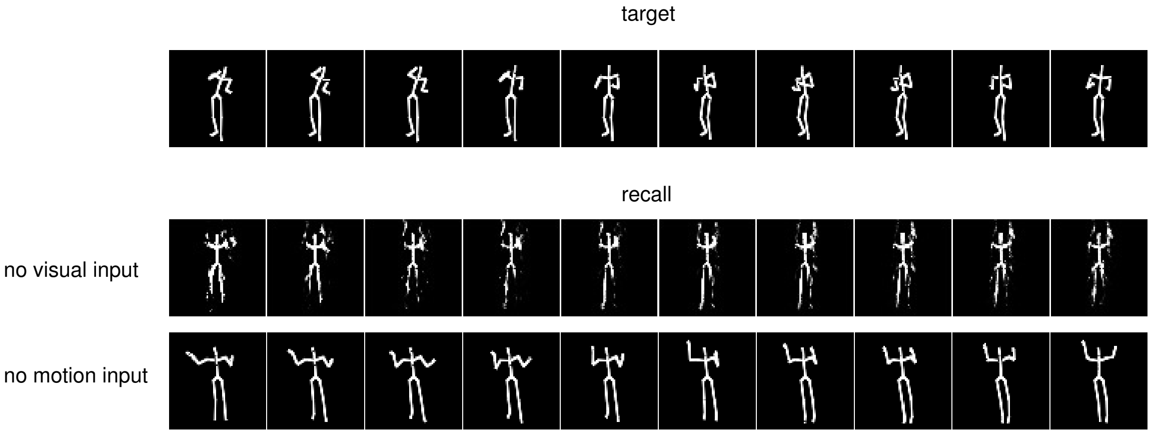

Figure 11: Visual images recalled from the only motion sequences and visualized motion sequences recalled from the only visual images. The input and target is not used to train the model. The rotation of motion pattern is set to that of target to clarify the motion of limbs.

ture vectors of each modality would be vague. When the only motion sequences are given and the visual images are reconstructed, the feature vectors of the motion sequences are clearly separated, and the fused feature vectors and reconstructed feature vectors of each modality would approach to the original trained feature vectors during repetition of recalling while compensating the missing modality. However, when the only visual image sequences are given, the feature vectors are not so clearly separated that the repeated recalling in multimodal autoencoder would not improve. That is because there is an asymmetricity between vision and motion. It might be related to the multimodal processes of human. It is relatively easy to imagine how we have moved during motion, but it is usually difficult to imitate the other's motion after looking.

When the LSTM autoencoder is trained with only vision, the recall error could not be small enough to show sharp motion figures. It might be because the feature vectors of vision encoded by LSTM were not separated among different input sequences in the feature space as described above. The errors of vision were slightly improved in the multimodal training (Fig. 8). The motion sequences help to classify the feature vectors to be separated. However, the motion errors were improved and not for vision in the end when the learning rate was set to the smaller values. This might be because the classification of fused feature vectors in the multimodal autoencoder was specialized too much for the motion, which broke down the decoded feature vectors of vision in the multimodal autoencoder and it also breaks the trained structure of the decoded LSTM autoencoder of vision. The fused feature vectors should be classified in the feature space properly for both vision and motion. This balancing problem can be overcome by a deep LSTM model because the deep LSTM model can build up high level representation in deep layers ([3]) and multimodal integration using the high level representation which has no statistical deviation depending on input modality is easier than using low level representation [10].

The current generalization ability of our model is not good enough as shown in Section 3.5. The possible reasons except for deficiency of LSTM are that the number of trained samples is too small and that our model is not a probabilistic model such as Boltzmann machine. A probabilistic version of autoencoder has been proposed, which is called variational autoencoder[5] and it is compatible with LSTM because it can be trained by backpropagation. One of our future works is to introduce the variational autoencoder to our multimodal LSTM model to obtain high generalization ability among modalities and apply it to imitation.

5. CONCLUSION

We proposed an architecture of neural network that can learn and integrate sequential multimodal information using LSTM. Our model consists of encoder and decoder LSTMs and multimodal autoencoder. Each LSTM deal with sequential information and multimodal autoencoder integrates multimodal information. We tested our proposed model by recall tasks on the visual and motion sequence. The experimental results have shown that our model has ability to learn and remember the sequential multimodal inputs, and even decrease the ambiguity generated at the learning stage of LSTMs using integrated multimodal information. Our model can also recall the visual sequences from the only motion sequences and vice versa. However, generalization ability for unknown input of our model is fairly low, therefore improving the ability is our future work.

Although we used two modalities, vision and motion, for multimodal information in this paper, our model can also deal with more than two modalities, for example vision, motion and sound. It is expected that additional modalities contribute to decrease the ambiguity of fused information. It is another future work to test our model on additional modalities.

6. ACKNOWLEDGMENTS

The data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.

7. REFERENCES

[1] A. Graves. *Supervised sequence labelling with recurrent neural networks*. PhD thesis, München, Techn. Univ., Diss., 2008, 2008.

[2] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, pages 1–43, 2013.

[3] A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.

[4] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[5] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[6] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal Deep Learning. *Proceedings of The 28th International Conference on Machine Learning (ICML)*, pages 689–696, 2011.

[7] K. Noda, H. Arie, Y. Suga, and T. Ogata. Multimodal integration learning of robot behavior using deep neural networks. *Robotics and Autonomous Systems*, 62(6):721–736, 2014.

[8] A. J. Robinson and F. Fallside. The utility driven dynamic error propagation network. *Technical Report CUED/F-INFENG/TR.1*, 1987.

[9] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv preprint arXiv:1502.04681*, 2015.

[10] N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012.

[11] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[12] R. J. Williams and D. Zipser. Gradient-based learning algorithms for recurrent networks and their computational complexity. *Back-propagation: Theory, architectures and applications*, pages 433–486, 1995.