A Lightweight Temporal Convolutional Network for Human Motion Prediction

*WANG You, QIAO Bing**

College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, P.R.China

(Received 8 May 2022; revised 23 July 2022; accepted 28 August 2022)

Abstract: A lightweight multi-layer residual temporal convolutional network model (RTCN) is proposed to target the highly complex kinematics and temporal correlation of human motion. RTCN uses 1-D convolution to efficiently obtain the spatial structure information of human motion and extract the correlation in the time series of human motion. The residual structure is applied to the proposed network model to alleviate the problem of gradient disappearance in the deep network. Experiments on the Human 3.6M dataset demonstrate that the proposed method effectively reduces the errors of motion prediction compared with previous methods, especially of long-term prediction.

Key words: human motion prediction; temporal convolutional network; short-term prediction; long-term prediction; deep neural network

CLC number: TP242.6 **Document code:** A **Article ID:** 1005-1120(2022)S-0150-08

0 Introduction

Understanding and predicting human motion has a variety of applications in the fields such as computer vision and robotics, which has been attracting more and more research interest. For instance, in unmanned driving, predicting the perceived intentions of pedestrians ahead can effectively avoid unnecessary accidents to a large extent. In the application of robots, anticipating the movement of workers can prevent a potential collision and can work better for the robot with human workers.

Human motion prediction is considered as a task to predict and generate future human body motion sequences based on a given sequence of body motion. The complexity of human motion is primarily reflected in two aspects: The kinematic constraints and the biological constraints between the various joints of space. The other is the non-linearity in temporal and great uncertainty in the motion velocity at different moments. Therefore, how to improve the accuracy of human motion prediction is generally considered from these two perspectives.

Since recurrent neural network (RNN) is a very effective method to deal with timing problems, a lot of researches have been done on capturing timing changes based on RNN. Fragkiadaki et al.[1] proposed an encoder-recurrent-decoder (ERD), in which the pose is mapped to the hidden state for the first time and propagated through the long short-term memory(LSTM) layer. The human motion sequence was formulated as spatial-temporal graphs through the structural-RNN designed by Jain et al.[2]. Martinez et al.[3] proposed a residual-based gate recurrent unit (GRU) (Res-GRU) model by predicting the gradient of human motion rather than human motion directly. Though these RNN based models can effectively extract the temporal correlation information of human motion sequence, it is challenging to learn the constraints of various parts of the body. With the approaches based on RNNs since human motion contains complex spatial information, the relative motion of limbs and trunk is biologically restricted, such as jumping and talking

*Corresponding author, E-mail address: bqiao@nuaa.edu.cn.

How to cite this article: WANG You, QIAO Bing. A lightweight temporal convolutional network for human motion prediction[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2022, 39(S): 150-157.

http://dx.doi.org/10.16356/j.1005-1120.2022.S.020

and those highly non-periodic actions. Furthermore, as the sequence length grows, learning the spatial structure of the human body becomes more challenging and errors accumulate more easily. For the extraction and utilization of spatial information of human motion sequence, convolutional neural network (CNN) based model can solve this problem well. Unlike RNNs, CNN models can efficiently and quickly learn high-level semantic cues with their naturally equipped excellent ability to extract high-level information. However, temporal information extraction is a problem that must be considered when processing human motion predictions based on CNNs. Therefore, some research on human motion sequence tries to make up for CNN's based model from the perspective of time sequence. Wang[4] proposed the joint trajectory map (JTM), representing spatial configuration and dynamics of joint trajectories into three texture images through color encoding. Guo et al.[5] adopted the long-term encoder and short-term encoder so that both distant and nearby temporal motion information can be used for future prediction. Yanshan and Rongjie[6] proposed the shape-motion representation from geometric algebra, which addressed the importance of both joints and bones. A sequence-to-sequence method was proposed by Li et al.[7] to alleviate the deficiency of recurrent networks to some extent. However, the improvement is not apparent, which makes it difficult to trade-off tight coherence in temporal and spatial because it enhances the connectedness of the human joints while weakening the continuity between the various skeleton sequences.

To effectively handle the above mentioned problems, an multi-layer residual temporal convolutional network (RTCN) model based on CNN is proposed in this paper. For RTCN, a two-dimensional spatiotemporal tensor extended from a series of human skeleton is used as input to the model, and the skeleton sequence is processed by 1-D convolution, which not only maintains the advantage of CNN spatial feature extraction but also reflects the sequential relationship of time and strengthens the continuity of time. In addition, the prediction target will be generated frame by frame, and each input will be pushed back frame by frame to ensure consistency in the timing sequence. To strengthen the spatial semantic association, it uses quaternion to represent skeletal joints[8].

1 Residual Temporal Convolutional Network

Human motion prediction is a sequence-related task, of which the input and output are all a sequence. The sequence-related task generally uses the encoder-decoder structure, in which the encoder encodes the input sequence by transforming it into an intermediate semantic vector $\boldsymbol{C}$ through a function, while the decoder is used to decode it. It means that all the previous information contained in the input sequence passed to the decoder layer through the intermediate semantic vector $\boldsymbol{C}$. however, there is a problem that the entire input sequence information cannot be fully expressed, especially when the length of the input sequence increases, which means on long-term prediction tasks, more information will be lost.

In response to the above problems, the method of this paper is to directly input the skeleton sequence into the convolutional network to predict the future target sequence. The prefix human motion sequence x_1—x_N is given as input, features are extracted through the RTCN model and the target pose $\hat{y}$ is generated, then it is fed back to the next epoch as input, finally, the final predicted sequence $\hat{y}_1$—$\hat{y}_M$ is generated, as shown in Fig.1. In Fig.1, the RTCN is based on a temporal convolutional network (TCN)[9] module followed by a batch normalization[10] layer and a leaky ReLu activation[11] layer. TCN operation are followed by a batch normalization and leaky ReLU, and then a residual connection is added, as shown in Fig.2. Cover all values contained in the input sequence by the dilated causal convolutional layer. Weight norm normalizes the convolutional layer weights in the residual block, adds ReLU to introduce nonlinearity, and introduces dropout regularization to avoid overfitting. Note that the Conv1-D is effective when the input channel and output channel are not equal. In addition, to al-

leviate the problem of gradient disappearance in the deep network, a residual connection[12] is added. Next, the detail will be introduced.

The temporal convolutional network is based upon two principles: The network produces an output of the same length as the input, so that there can be no leakage from the future into the past. As shown in Fig.3, to solve the first point, a 1-D fully-convolutional network (FCN) architecture[13] is used, in which each hidden layer has the same length as the input layer and zero paddings of length (kernel size-1) is added to keep the subsequent layers the same length as previous ones. For the second point, causal convolutions are used, where the output of time t is convolved only with time t and the elements at the previous layer. However, a simple causal convolution could only look back at history with a size linear in the depth of the network, so that an intense network or extensive filters are needed. But this will increase the computational effort and make the network more complex.

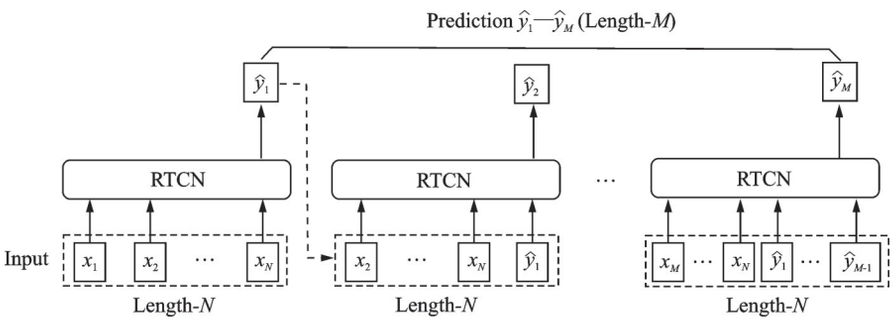

Fig.1 Model structure of human motion prediction

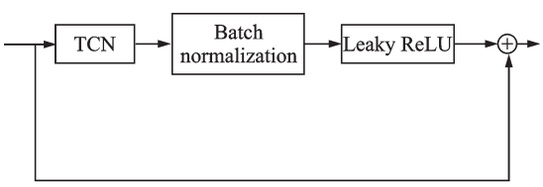

Fig. 2 Structure of RTCN

In order to solve the above problem, following the Ref. [14], the dilated convolutions are employed to enable an exponentially sizeable receptive field. More formally, for a 1-D sequence input $\boldsymbol{x}\in\mathbf{R}^n$ and a filter f: $\{0,\cdots,k-1\}\rightarrow\mathbf{R}$, the dilated convolution operation F on element s of the sequence is defined as

$$F(s)=\sum_{i=0}^{k-1}\boldsymbol{f}(i)\cdot\boldsymbol{x}_{s-d\cdot i} \tag{1}$$

where d is the dilation factor, k is the filter size, and $s-d\cdot i$ accounts for the past direction. Thus, the larger filter sizes k and dilation factor d can be chosen to increase the receptive field. Moreover, the depth of the network would not be increased.

A residual block contains a branch leading out to a series of transformations Λ, including weight norm, ReLU, and dropout, in which outputs are added to the input $\boldsymbol{x}$ of the block

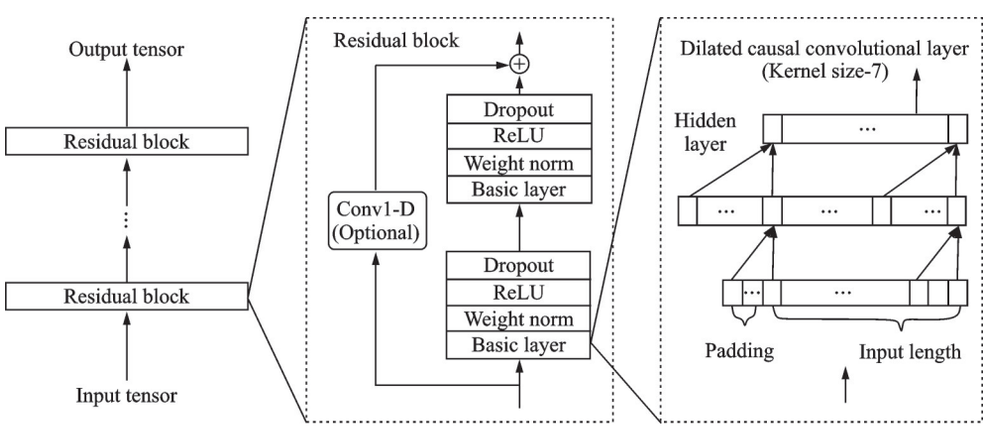

Fig.3 Illustration of the detailed structure of TCN

$$\boldsymbol{o}=\text{Activation}\left(\boldsymbol{x}+\Lambda(\boldsymbol{x})\right) \tag{2}$$

where $\boldsymbol{o}$ is output, and Activation$(\cdot)$ is an activation function. The above equation effectively allows layers to learn modifications to the identity mapping rather than the entire transformation. When the network reaches a certain depth, in the backpropagation, the gradient tends to 0, that is, the problem of gradient disappearance. The residual connection adds the input of the previous layer to the original direct mapping to ensure the effective and stable update of the gradient.

The whole training process of the model: Given the length-N prefix sequence of human motion skeleton sequence $\boldsymbol{X}=[x_1,x_2,\cdots,x_N]\in\mathbf{R}^{N\times J\times K}$, where each of the frames $\{\boldsymbol{x}_n^j\}_{j=1}^{J}$ represents the single skeleton, containing J joints data. $\boldsymbol{x}_n^j\in\mathbf{R}^K$ is a minimal per-joint representation at the n-th frame and the j-th joint, in which K is the feature dimension representing human joint data. $K=4$ is adopted for quaternion as this format is free of discontinuity and singularity[8]. The goal of human motion prediction is to generate consecutive length-M target sequence $\hat{\boldsymbol{Y}}=[\hat{y}_1,\hat{y}_2,\cdots,\hat{y}_M]$ for the next T frame poses. Note that, in this paper, the last frame of the output will be taken as the final prediction for each epoch, the generated output at time t will be the input at time $t+1$, as shown in Fig.1, the detail equation is

$$\{\boldsymbol{X}\}_{n=t}^{N+t}=\{\boldsymbol{X}\}_{n=t}^{N}+\left\{\hat{\boldsymbol{Y}}\right\}_{m=1}^{t} \tag{3}$$

$$\hat{y}_t=O\left(\{\boldsymbol{X}\}_{n=t-1}^{N+t-1}\right) \tag{4}$$

where $\{X\}_{n=t}^{N+t}$ denotes the N frames input at time t, $\left\{\hat{\boldsymbol{Y}}\right\}_{m=1}^{t}$ the prediction range from 1 to t, $\hat{y}_t$ the prediction at time t, and $O(\cdot)$ is the output function.

This paper applies the average L1 distance as the reconstruction loss

$$\mathcal{L}=\frac{1}{J\times M}\sum_{j=1}^{J}\sum_{t=1}^{M}\left\|\hat{y}_t^j-y_t^j\right\|_1 \tag{5}$$

where $\hat{y}_t^j$ denotes the predicted skeleton of the j-th joint in the t-th frame, and y_t^j the corresponding ground truth.

2 Experiments and Results

In this section, the experiments are carried out on popular motion capture benchmarks dataset: Human 3.6M[15] and compared with the current state-of-the-art and ablation studying.

2.1 Datasets and implementation details

Human 3.6M is currently the most prominent 3-D video pose dataset. It was recorded by four static cameras and 15 active scenes performed by seven different professional actors. There are two video sequences for each active scene, each of which is between 3 000 and 5 000 frames, due to the different theme and camera views. Each frame contains 34 sets of data, including a global translation, a global rotation, and 32 joint rotations relative to the parent joint. Each joint is represented as an exponential map (axis angle). Following the standard protocol[3,8], all sequences are de-sampled to a rate of 25 frame/s and global translation and global rotation are discarded. For evaluation, we use the Euler angle to calculate the Euler error. Specifically, we use the Euler angle of each action to measure the Euclidean distance between our prediction and the ground truth, and then calculate the mean value of all sequences randomly selected from the test set. In addition, The Subject 5 (S5) is used in testing while the others are used in training.

Two layers of RTCN with residual connections are contained in the model, in which TCN sets up six hidden layers in each layer, and each hidden layer has 50 channels. The kernel size is set to seven for TCN. Each TCN is performed with the same dilated and padding mode to ensure that the temporal size remains the same. Leaky ReLU is used as nonlinear activation, and the activation rate is 0.01. The model is lightweight with 0.86 MB.

The model is implemented in the Pytorch framework, and the ADAM[16] optimizer is used to optimize the model. The initial learning rate is 0.001, and the decay is 0.999. The gradient clipping norm is set to 0.1, and the mini-batch consists of 60 samples. The model is trained on a single NVIDIA RTX 2080 SUPER GPU.

2.2 Evaluation

In this paper, the results reported on all 15 actions in the Human 3.6M dataset for short-term prediction of 80, 160, 320, 400 ms and long-term prediction of 560 and 1 000 ms. Three recent advanced methods on human motion prediction will be compared to evaluate the performance of the model, in which Res-GRU[3] and SkelNet[17] are based on RNNs, ConvSeq2Seq[7] is based on CNN, as well as one baseline method Zero-velocity[3], on short-term and long-term. All the other results are referred from their original papers.

2.2.1 Short-term prediction

Firstly, all the actions of the method in this paper are compared with other methods in the short-term. As shown in Table 1, it can be seen that the method outperforms the other methods in most actions. According to the mean value comparison in the last column, in the first 80 ms, the results are consistent with those of other methods, but in subsequent 160, 320, and 400 ms, the performance of this method is improved. Besides it performs better on periodic movements, such as Walking and Smoking, it significantly performs better on non-periodic movements such as Discussion and Walking dog. This is because the method is better at capturing spatial variations in human motion. However, there is a great difference in action, "Phoning" is noticed in all actions. Through the analysis, it can be found that the error is caused by discontinuous test data, but the abnormal data is from the end of the finger joints so that it will have little impact on the whole 3-D space. The problem is also discussed here[18].

Table 1 Mean angle error for short-term motion prediction on Human 3.6M for all 15 actions

Action	Walking				Eating				Smoking				Discussion			
Milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Zero-velocity[3]	0.39	0.68	0.99	1.15	0.27	0.48	0.73	0.86	0.26	0.48	0.97	0.95	0.31	0.67	0.94	1.04
Res-GRU[3]	**0.27**	0.47	0.70	0.78	0.25	0.43	0.71	0.87	0.33	0.61	1.04	1.19	0.31	0.69	1.03	1.12
ConvSeq2Seq[7]	0.33	0.54	0.68	0.73	0.22	0.36	0.58	0.71	0.26	0.49	0.96	0.92	0.32	0.67	0.94	1.01
SkelNet[17]	0.31	0.50	0.69	0.76	**0.20**	**0.31**	0.53	0.69	**0.25**	0.50	0.93	0.89	**0.30**	0.64	0.89	0.98
RTCN (Ours)	**0.27**	**0.38**	**0.53**	**0.60**	0.22	0.33	**0.46**	**0.56**	**0.25**	**0.36**	**0.61**	**0.60**	0.40	**0.60**	**0.69**	**0.83**
Action	Direction				Greeting				Phoning				Posing			
Milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Zero-velocity[3]	0.39	0.59	0.79	0.89	0.54	0.89	1.30	1.49	0.64	1.21	1.65	0.83	**0.28**	0.57	1.13	1.37
Res-GRU[3]	**0.26**	**0.47**	0.72	0.84	0.75	1.17	1.74	1.83	**0.23**	**0.43**	**0.69**	**0.82**	0.36	0.71	1.22	1.48
ConvSeq2Seq[7]	0.39	0.60	0.80	0.91	0.51	0.82	1.21	1.38	0.59	1.13	1.51	1.65	0.29	0.60	1.12	1.37
SkelNet[17]	0.36	0.58	0.77	0.86	**0.50**	0.84	1.28	1.45	0.58	1.12	1.52	1.64	0.29	0.62	1.19	1.44
RTCN (Ours)	0.40	0.48	**0.53**	**0.55**	0.53	**0.62**	**0.79**	**0.86**	0.62	1.07	1.29	1.36	0.34	**0.46**	**0.67**	**0.70**
Action	Purchases				Sitting				Sitting down				Taking photo			
Milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Zero-velocity[3]	0.62	0.88	1.19	1.27	0.40	1.63	1.02	1.18	0.39	0.74	1.07	1.19	0.25	0.51	0.79	0.92
Res-GRU[3]	0.51	0.97	1.07	1.16	0.41	1.05	1.49	1.63	0.39	0.81	1.40	1.62	0.24	0.51	0.90	1.05
ConvSeq2Seq[7]	0.63	0.91	1.19	1.29	0.39	0.61	1.02	1.18	0.41	0.78	1.16	1.31	**0.23**	0.49	0.88	1.06
SkelNet[17]	**0.58**	0.84	1.17	1.24	0.40	0.61	1.01	1.15	0.37	0.72	1.05	1.17	0.24	0.47	0.78	0.93
RTCN (Ours)	0.61	**0.60**	**0.78**	**0.72**	**0.34**	**0.47**	**0.61**	**0.70**	**0.36**	**0.59**	**0.65**	**0.66**	**0.23**	**0.39**	**0.45**	**0.51**
Action	Waiting				Walking dog				Walking together				Average			
Milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Zero-velocity[3]	0.34	0.67	1.22	1.47	0.60	0.98	1.36	1.50	0.33	0.66	0.94	0.99	0.42	0.74	1.12	1.20
Res-GRU[3]	**0.28**	0.53	1.02	1.14	0.56	0.91	1.26	1.40	0.31	0.58	0.87	0.91	0.39	0.72	1.08	1.22
ConvSeq2Seq[7]	0.30	0.62	1.09	1.30	0.59	1.00	1.32	1.44	0.27	0.52	0.71	0.74	0.38	0.68	1.01	1.13
SkelNet[17]	0.30	0.63	1.17	1.40	**0.54**	0.88	1.20	1.35	0.27	0.53	0.68	0.74	**0.36**	0.64	0.99	1.02
RTCN (Ours)	0.31	**0.49**	**0.73**	**0.84**	0.55	**0.75**	**0.89**	**0.97**	**0.24**	**0.41**	**0.47**	**0.50**	0.38	**0.53**	**0.68**	**0.73**

2.2.2 Long-term prediction

The method is also compared with other methods for long-term prediction of all actions. As shown in Table 2, it shows the results obtained

based on open source baselines and running your own code, and the places where there is no open code baseline are left blank. It is clear that the method outperforms all other methods in the long-term predictions of 560 and 1 000 ms. For example, the performance gain is 0.42 and 0.37 for 560 and 1 000 ms of "Smoking". In addition, the improvement in the long-term forecast is more significant than the improvement in the short-term forecast. For example, the performance gain in the reaches of 0, 0.12, 0.32, and 0.29 for 80, 160, 320, and 400 ms of "Smoking" has not been significantly improved compared to the long-term mission. This shows that the method can effectively alleviate the problem of excessively fast error accumulation in RNNs in long-term generation, and the effect is better than other convolutional networks.

Table 2 Mean angle error for long-term motion prediction on Human 3.6M for all 15 actions

Action	Walking		Eating		Smoking		Discussion		Direction		Greeting		Phoning		Posing	
Milliseconds	560	1 000	560	1 000	560	1 000	560	1 000	560	1 000	560	1 000	560	1 000	560	1 000
Zero-velocity[3]	1.35	1.32	1.04	1.38	1.02	1.69	1.41	1.96	1.02	1.50	1.79	1.80	1.81	2.04	1.81	2.78
Res-GRU[3]	0.93	1.03	0.95	1.08	1.25	1.50	1.43	1.69	0.96	1.42	1.68	1.76	1.56	1.77	1.78	2.29
ConvSeq2Seq[7]	0.86	0.92	0.89	1.24	0.97	1.62	1.44	1.86	0.93	1.42	1.57	1.79	1.66	1.83	1.75	2.78
SkelNet[17]	0.79	0.83	0.84	1.06	0.98	1.21	1.39	1.75								
RTCN (Ours)	**0.72**	**0.79**	**0.65**	**0.83**	**0.56**	**0.84**	**1.09**	**1.31**	**0.66**	**0.91**	**1.09**	**1.16**	**1.21**	**1.29**	**1.17**	**1.43**

Action	Purchases		Sitting		Sitting down		Taking photo		Waiting		Walking dog		Walking together		Average	
Milliseconds	560	1 000	560	1 000	560	1 000	560	1 000	560	1 000	560	1 000	560	1 000	560	1 000
Zero-velocity[3]	1.64	2.45	1.26	1.63	1.36	1.80	1.03	1.27	1.89	2.63	1.74	1.96	1.10	1.52	1.42	1.85
Res-GRU[3]	1.41	2.30	1.24	1.51	1.28	1.72	0.95	1.17	1.64	2.30	1.69	1.73	0.80	1.43	1.30	1.65
ConvSeq2Seq[7]	1.44	2.38	1.15	1.48	1.26	1.75	0.92	1.23	1.70	2.37	1.62	1.78	0.79	1.45	1.26	1.73
SkelNet[17]																
RTCN (Ours)	**0.89**	**1.42**	**0.66**	**0.78**	**0.72**	**0.97**	**0.54**	**0.55**	**0.89**	**1.14**	**1.10**	**1.12**	**0.59**	**1.01**	0.84	1.03

2.3 Ablation study

Effects of the number of hidden layers: In order to test the influence of the number of hidden layers in TCN on the prediction error, other parameters are kept unchanged, where the kernel size of the convolution is set to 7, and the numbers of hidden layers are set as 2, 4, 6, 8, and 10 for experiments.

Table 3 shows the mean error of all actions corresponding to 80, 160, 320, 400, 560, and 1 000 ms under different layers. It can be seen that the number of hidden layers at 80 ms has almost no effect on the error but has a more significant effect on the following, with a different range of 0.7—0.14. In addition, when the number of hidden layers is 6, the overall effect is well, and when the number of hidden layers continues to increase, the effect is also good in some periods. For example, the minimum error is reached at 80 ms. However, the overall improvement is minor. Furthermore, with the increase in the number of layers, the time consumption of training and testing will greatly aggravate, and the model parameters and the amount of calculation will also increase. As shown in Fig.4(a), as the number of hidden layers increases, the volume of the model becomes more extensive and grows linearly. In summary, the number of hidden layers is selected as 6 in this paper.

Table 3 Average mean angle error of different values of hidden layer both on short-term and long-term

ms

Layer	80	160	320	400	60	1 000
2	0.39	0.60	0.75	0.81	0.93	1.17
4	0.38	0.55	0.70	0.76	0.89	1.12
6	0.38	0.53	0.68	0.73	0.84	1.03
8	0.38	0.54	0.68	0.73	0.82	1.05
10	0.37	0.53	0.68	0.74	0.86	1.08

Generally, in the convolutional network model, the convolution kernel size has a relatively critical effect on model performance. In order to test the influence of convolution check on prediction error in our model RTCN, as above, other parameters are kept unchanged, among which the number of hidden layers is set to 6, and the size of the convolu-

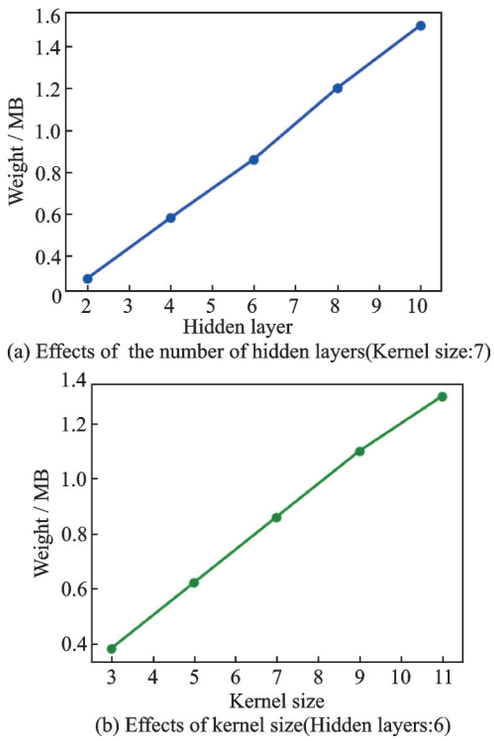

Fig. 4 Illustration of effects of both hidden layers and kernel size for weight of model of Human 3.6M dataset

tional kernel is set as 3, 5, 7, 9, 11 for experiment, respectively. Table 4 gives the error means of all actions at different moments. It can be observed that when the kernel size grows from 3 to 5, 7, the effect becomes better and better. For the model, the wider its convolution kernel, the larger the corresponding receptive field, and the more complete the historical information retained. However, when the convolution kernel continues to increase, the effect does not improve more. This is because when the convolution kernel is so large that it can cover all the history of the input under the limited number of hidden layers (the number of layers is six here). In the case of information, the increase of the convolution kernel does not have much impact on the extraction of past information. However, it products redundancy and reuses information, and leads to a more bloated model. As shown in Fig. 4 (b), as the convolution kernel grows, the volume of the model becomes more prominent and rises linearly. In summary, the size of the convolution kernel is selected as 7 in this paper.

Table 4 Average mean angle error of different values of kernel size both on short-term and long-term

ms

K-size	80	160	320	400	560	1 000
3	0.39	0.56	0.74	0.80	0.93	1.16
5	0.38	0.54	0.70	0.75	0.87	1.08
7	0.38	0.53	0.68	0.73	0.84	1.03
9	0.38	0.54	0.68	0.75	0.86	1.12
11	0.38	0.54	0.69	0.74	0.87	1.12

3 Conclusion

In this work, rapid accumulation of errors based on RNNs and excessive loss of spatial information in human motion prediction tasks are analyzed, and some methods based on CNN cannot extract timing relationship information well. To this end, the method RTCN based on CNN is proposed in this paper using 1-D convolution to extract spatial information while retaining the timing information. Experiments show that the method performs better than previous methods, especially on long-term prediction tasks. In addition, the model is more straightforward and has the fewer parameters, which reduces the weight of the model and speeds up training.

References

[1] FRAGKIADAKI K, LEVINE S, FELSEN P, et al. Recurrent network models for human dynamics[C]// Proceedings of IEEE International Conference Computer Vision(ICCV). [S.l.]: IEEE, 2015:4346-4354.

[2] JAIN A, ZAMIR A R , SAVARESE S, et al. Structural-RNN: Deep learning on spatio-temporal graphs [C]// Proceedings of IEEE Conference Computer Vision Pattern Recognition (CVPR). [S. l.] : IEEE, 2016:5308-5317.

[3] MARTINEZ J, BLACK M J, ROMERO J. On human motion prediction using recurrent neural networks [C]// Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition(CVPR).[S.l.] IEEE, 2017: 4674-4683.

[4] WANG P, LI W, LI C, et al. Action recognition based on joint trajectory maps with convolutional neural networks[C]//Proceedings of ACM on Multimedia Conference.[S.l.]: ACM, 2016: 1044-1048.

[5] GUO X, CHOI J. Human motion prediction via learning local structure representations and temporal dependencies[C]//Proceedings of AAAI Conference Artifical Intelligence.[S.l.]:AAAI, 2019: 2580-2587.

[6] YANSHAN L, RONGJIE X, XING L, et al. Learning shapemotion representations from geometric algebra spatio-temporal model for skeleton-based action recognition[C]//Proceedings of IEEE International

Conference on Multimedia & Exposition. [S. l.]: IEEE, 2019:674-679.

[7] LI C, ZHANG Z, LEE W S, et al. Convolutional sequence to sequence model for human dynamics[C]// Proceedings of IEEE/CVF Conference Computer Vision Pattern Recognition.[S.l.]: IEEE, 2018: 5226-5234.

[8] PAVLLO D, GRANGIER D, AULI M. Quater-Net: A quaternion-based recurrent model for human motion [EB/OL]. (2018-07-31)[2022-06-08]. https://arxiv.org/abs/1805.06485V2.

[9] Van Den OORD A, DIELEMEN S, ZEN H, et al. WaveNet: A generative model for raw audio [EB/OL]. (2016-09-12)[2022-06-12]. https://arxiv.org/abs/1609.03499v2.

[10] IOFFE S, SZEGEDY C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]// Proceedings of the 32nd International Conference on Machine Learning. [S.l.]: PMLR, 2015:448-456.

[11] MAAS A L, HANNUN A Y, NG A Y. Rectifier nonlinearities improve neural network acoustic models [C]//Proceedings of International Conference on Machine Learning(ICML). [S.l.]: ICML, 2013:51-58.

[12] HE KAIMING, ZHANG Xiangyu, REN Shaoqing, et al. Deep residual learning for image recognition [C]//Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition(CVPR).[S.l.]: IEEE,2016:770-778.

[13] Vohra R, GOEL K, SAHOO J K. Modeling temporal dependencies in data using a DBN-LSTM[J]. Data Science and Advanced Analytics(DSAA), 2015, 18(16):143-149.

[14] SCOTT W, THOMAS P, JOHN H, et al. Full-capacity unitary recurrent neural networks[J]. Conference and Workshop on Neural Information Processing Systems(NIPS), 2016, 24(31): 103-111.

[15] IONESCU C P D, OLARU V. Human 3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014,36(7): 1325.

[16] KINGMA D, Ba J. Adam: A method for stochastic optimization[C]// Proceedings of the 3rd International Conference on Learning Representations (ICLR). [S.l.]: ICLR,2015.

[17] GUO X, CHOI J. Human motion prediction via learning local structure representations and temporal dependencies[C]//Proceedings of AAAI Conference Artifical Conference Artifical Intelligence. [S.l.]: AAAI, 2019:2580-2587.

[18] MAO W, LIU M, SALZMANN M, et al. Learning trajectory dependencies for human motion prediction [C]//Proceedings of Conference Computer Vision Pattern Recognition.[S.l.]: IEEE, 2019:9488-9496.

Authors Mr. **WANG You** is a master of Engineering, Nanjing University of Aeronautics and Astronautics, majoring in aerospace engineering. Main research fields: image processing, human motion prediction, and robotics.

Dr. **QIAO Bing** is an associate researcher, graduated from Nanjing University of Aeronautics and Astronautics School of Mechanical and Electrical Engineering in 1999, majoring in mechanical and electronic engineering. From August 2008 to February 2009, he was a visiting scholar in the Department of Mechanical and Aerospace Engineering, New Mexico state University. The main research areas include: intelligent robotics, space robotics, spacecraft autonomous on-orbit service technology and modular spacecraft design.

Author contributions Mr. **WANG You** designed the study, complied the models, conducted the analysis, interpreted the results and wrote the manuscript. Dr. **QIAO Bing** contributed to the discussion and background of the study. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: LIU Yandong)

基于时间卷积轻量级网络的人体运动预测

汪 友，乔 兵

(南京航空航天大学航天学院，南京 210016，中国)

摘要：针对人体运动高度复杂的运动学和时间相关性，本文提出了一种轻量级的多层残差时间卷积网络模型(Residual temporal convolutional network, RTCN)。该模型使用一维卷积高效获取人体运动的空间结构信息，提取人体运动时间序列中的相关性。在本文所提出的网络模型中应用残差结构来缓解深度网络中梯度消失的问题。在Human 3.6M数据集上进行的实验表明，与最新的方法相比，本文方法有效地改善了运动预测的误差，特别是在长期预测方面。

关键词：人体运动预测；时间卷积网络；短期预测；长期预测；深度神经网络