ATLANTIS PRESS

Advances in Engineering Research, volume 166

3rd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE 2018)

Frame Prediction Using Recurrent Convolutional Encoder with Residual Learning

Bo-xuan YUE
State Key Laboratory of Industrial Control Technology,
Department of Control Science and Engineering, Zhejiang University
Hangzhou, P. R. China
e-mail: ybx90@outlook.com

Jun Liang
State Key Laboratory of Industrial Control Technology,
Department of Control Science and Engineering,
Zhejiang University
Hangzhou, P. R. China
e-mail: jliang@zju.edu.cn

***Abstract*—The prediction for the frame of a video is difficult but in urgent need in auto-driving. Conventional methods can only predict some abstract trends of the region of interest. The boom of deep learning makes the prediction for frames possible. In this paper, we propose a novel recurrent convolutional encoder and deconvolutional decoder structure to predict frames. We introduce the residual learning in the convolution encoder structure to solve the gradient issues. The residual learning can transform the gradient backpropagation to an identity mapping. It can reserve the whole gradient information and overcome the gradient issues in Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). Besides, compared with the branches in CNNs and the gated structures in RNNs, the residual learning can save the training time significantly. In the experiments, we use UCF101 dataset to train our networks, the predictions are compared with some state-of-the-art methods. The results show that our networks can predict frames fast and efficiently. Furthermore, our networks are used for the driving video to verify the practicability.**

Keywords- residual learning; recurrent convolutional networks; frame prediction.

I. INTRODUCTION

Representation learning is one of the potential direction of video process with the development of hierarchical networks. The deep networks are now not limited for classification and recognition. Researchers have succeeded to predict the sequences. In this paper, we address the issue of predicting the 2D sequence signals, i.e. frame prediction, which is used for predicting the scene for an auto-driving vehicle. Compared with sequence predictions, videos have sequential features on no matter timeline and space. These features provides sufficient information for frame prediction. However, more requirements are raised by frame prediction. Other than image reconstruction, frame prediction outputs the next frame, and the predicted frame requires high resolution and clear edges. Therefore, the issue of frame prediction is to output the sharp future frame.

The researchers have focused on frame prediction for long. Van Hateran and Ruderman [5] apply ICA (Independent Component Analysis) method to small video cubes of patches, and so do Hurri and Hyvarinen [6]. Wiskott and Sejnowski [7] develop a method based on the slowness features through time. Bilinear models based on the transformations between nearby frames are widely investigated, the related researches are proposed in the works of Memisevic and Hinton [8], as well as Miao and Rao [9]. Currently, the hierarchical networks plays an import role on frame prediction. Ranzato et al. [3] define a recurrent convolutional network structure based on the language modeling, and predict frames in a discrete space of patch clusters. Srivastava et al. [10] adapt the LSTM model [11]for 2D sequence to predict frames. Oh et al. [12] define an action conditional auto-encoder model to predict next frames in the Atari-like games. Mathieu et al. introduce GAN [4] to predict the frames and reformulate the loss function with the combination of the MSE (mean square error) and the gradients.

To predict a sharp frame and overcome the drawbacks, we propose a frame prediction method using recurrent convolutional networks with residual learning. It is well known that the gradient vanishing and explosion exist in the backpropagation of deep convolutional networks and long-term recurrent networks, which lead in non-convergence. To solve this issue, the gate functions are exploited in the recurrent networks, and ReLU function and assistant loss functions are exploited in the deep convolutional networks. Both of them avoid gradient issues with the cost of computation complexity. To train the networks efficiently, we introduce residual learning in the networks. The learned representation from the previous

frames are restored to a frame by deconvolution networks. The major contributions of this paper are summarized as follows: (1) we propose a novel structure to solve the gradient issues of recurrent convolutional networks; (2) we construct a recurrent encoder and a decoder to predict frames; (3) we prove the residual learning solve the gradient issues theoretically.

II. Recurrent Convolutional Encoder with Residual Learning

A. Gradient Issues of Deep Recurrent Convolutional Networks

It is well known that the training of RNNs [13] and CNNs [14] is difficult when the networks are very deep . The approach of updating gradients for RNNs and CNNs is backpropagation and backpropagation through time (BPTT), respectively. The recurrent model can be unfolded as a multi-layer one with connections in the same layer, and backpropagation through time is similar to the backpropagations in other hierarchical networks, such as Deep Brief Networks (DBN), Auto-Encoder (AE) and CNNs. We take a generic RNN for example to demonstrate the gradient issues as follows. A generic RNN, with an input $\mathbf{x}_t \in \mathbb{R}^n$ and the state $\mathbf{s}_t \in \mathbb{R}^m$ is given by

$$\mathbf{s}_t = f(\mathbf{x}_t, \mathbf{s}_{t-1}, \boldsymbol{\theta}), \tag{1}$$

where $\boldsymbol{\theta}$ is the collection of the input weight matrix $\mathbf{W}$, recurrent weight matrix $\mathbf{U}$ and the bias $\mathbf{b}$. In details, the state at the timestamp t is described as

$$\mathbf{s}_t = \mathbf{U}\sigma(\mathbf{s}_{t-1}) + \mathbf{W}\mathbf{x}_t + \mathbf{b}, \tag{2}$$

where σ is the sigmoid function. Denoting the cost function as $\mathcal{E}$, the BPTT is formulated as

$$\frac{\partial \mathcal{E}}{\partial \boldsymbol{\theta}} = \sum_{1<t<T} \frac{\partial \mathcal{E}_t}{\partial \boldsymbol{\theta}}, \tag{3}$$

$$\frac{\partial \mathcal{E}_t}{\partial \boldsymbol{\theta}} = \sum_{1 \leqslant k \leqslant t} \left(\frac{\partial \mathcal{E}_t}{\partial \mathbf{s}_t} \frac{\partial \mathbf{s}_t}{\partial \mathbf{s}_k} \frac{\partial^+ \mathbf{s}_k}{\partial \boldsymbol{\theta}} \right), \tag{4}$$

$$\frac{\partial \mathbf{s}_t}{\partial \mathbf{s}_k} = \prod_{t>i>k} \frac{\partial \mathbf{s}_i}{\partial \mathbf{s}_{i-1}} = \prod_{t>i>k} \mathbf{W}_{rec}^T \operatorname{diag}(\sigma'(\mathbf{s}_{i-1})), \tag{5}$$

where T is the time distributed length. As Bengio et al. [15-18] discussed and (5) shows, the gradient issues are relative to the continuous multiplications. Because of long-term states, the gradient exploding happens when the gradient grows exponentially, while the gradient vanishing happens when the gradient go exponentially fast to norm 0. The gradient issues of CNNs are similar to RNNs. The gradient issues lead to no convergence.

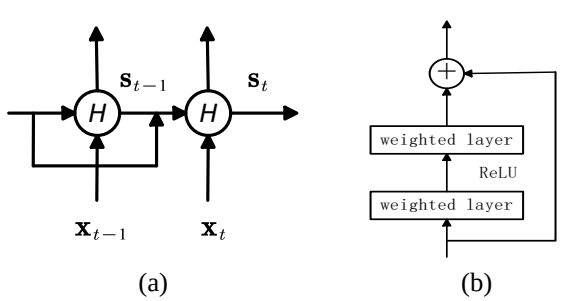

Figure 1. Residual recurrent structure (a) and residual convolutional structure (b) of Residual Recurrent Convolutional Encoder

B. Residual Recurrent Convolutional Encoder

To solve the gradient issues, we propose a recurrent convolutional encoder with residual learning. The recurrent convolutional encoder is composed of two parts, recurrent blocks and convolutional blocks.

Recurrent blocks: As $f(\mathbf{x}_t, \mathbf{s}_{t-1}, \boldsymbol{\theta})$ is an underlying mapping from state to state, we learn this mapping with a residual, given by

$$\mathcal{F}(\mathbf{x}_t, \mathbf{s}_{t-1}) = f(\mathbf{x}, \mathbf{s}_{t-1}, \boldsymbol{\theta}) - \mathbf{s}_{t-1}. \tag{6}$$

Thus, the state is calculated as

$$\mathbf{s}_t = \mathbf{s}_{t-1} + \mathscr{F}(\mathbf{x}, \mathbf{s}_{t-1}, \boldsymbol{\theta}). \tag{7}$$

Equation (7) is composed of a shortcut connection and an element-wise addition. The shortcut connection does not introduce extra parameters and computation complexity. This reconstruction makes the loss function approximate to an identity mapping. When the recurrent connections are formulated as identity mapping, the training error should be non-increasing. To drive (7) to approach an identity mapping, the weights of nonlinear block in (7) are tuned towards zero. Considering the dimension equation, a linear projection is introduced to match the dimensions. Thus, the forward propagation of recurrent networks is shown as

$$\mathbf{y} = \mathbf{W}_{im}\mathbf{s}_{t-1} + \mathscr{F}(\mathbf{x}, \mathbf{s}_{t-1}, \boldsymbol{\theta}), \tag{8}$$

where $\mathscr{F}$ is the residual function, $\mathbf{W}_{im}$ is the linear projection weight and an identity mapping.

The state is the addition of an identity mapping and a residual function. The activation function is also an identity mapping. For the reason of recurrent transmission and the residual learning as (8), the state $\mathbf{s}_t$ can be unrolled as

$$\mathbf{s}_t = \mathbf{s}_{t-T} + \sum_{1<k<T} \mathscr{F}(\mathbf{x}, \mathbf{s}_{t-k}, \boldsymbol{\theta}), \tag{9}$$

where T is the length of sequential dependencies. The accumulation equation (9) shows that any state s_t can be represented as a former state and a sum of residuals. When the initial state $\mathbf{s}_0$ is 0, the state $\mathbf{s}_t$ of timestamp t is the sum of series of residuals. Equation (9) results in a better backward propagation. Considering the loss function represented in (3-5), with the residual accumulation in (9), the gradient in BPTT is

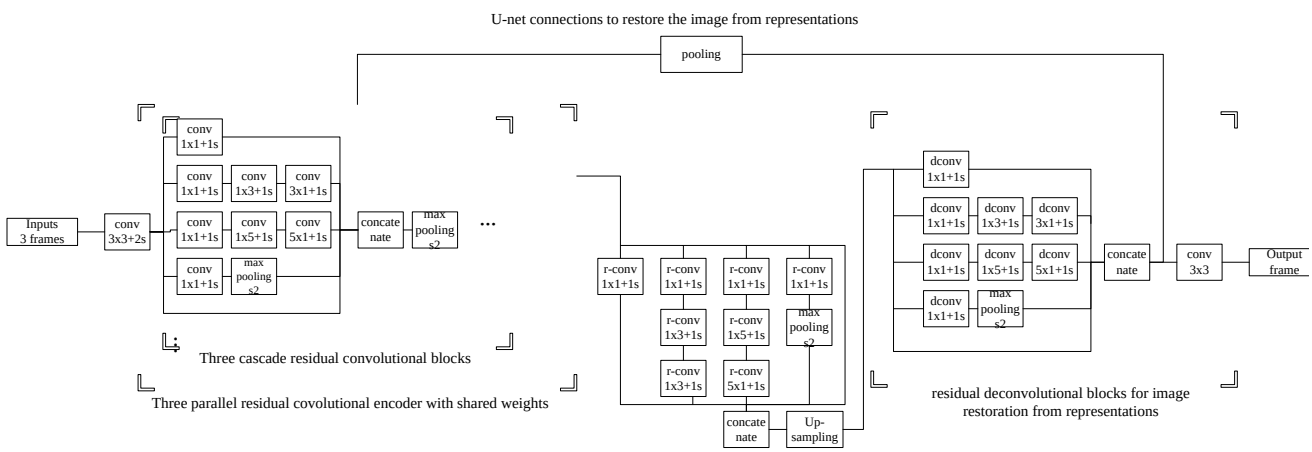

Figure 2. The model with residual recurrent convolutional encoder composed of residual recurrent structure and residual convolutional structure, U-net connections and residual deconvolutional blocks for image restoration from representations.

$$\begin{aligned}\frac{\partial \mathcal{E}}{\partial \mathbf{s}_{t-T}} &= \frac{\partial \mathcal{E}}{\partial \mathbf{s}_t}\frac{\partial \mathbf{s}_t}{\partial \mathbf{s}_{t-T}} \\ &= \frac{\partial \mathcal{E}}{\partial \mathbf{s}_t}\left(1 + \frac{\partial}{\partial \mathbf{s}_{t-T}} \sum_{1<k<T} \mathscr{F}(\mathbf{x}, \mathbf{s}_{t-k}, \boldsymbol{\theta})\right).\end{aligned} \tag{10}$$

Equation (10) describes that the gradient $\frac{\partial \mathcal{E}}{\partial \mathbf{s}_{t-T}}$ can be decomposed into two additive parts. One is $\frac{\partial \mathcal{E}}{\partial \mathbf{s}_t}$, which propagates information directly without weights, another is the accumulation, $\frac{\partial \mathcal{E}}{\partial \mathbf{s}_t}\left(\frac{\partial}{\partial \mathbf{s}_{t-T}} \sum_{1<k<T} \mathscr{F}(\mathbf{x}, \mathbf{s}_{t-k}, \boldsymbol{\theta})\right)$, which needs the inner product with weight matrices. The direct propagation part ensures the gradient information can be propagated to any state. The direct gradient part comes from the two identity mappings. It suggests the information can be entirely propagated both forward and backward, even when the weights are arbitrarily small. Because the accumulation part, $\frac{\partial \mathcal{E}}{\partial \mathbf{s}_t}\left(\frac{\partial}{\partial \mathbf{s}_{t-T}} \sum_{1<k<T} \mathscr{F}(\mathbf{x}, \mathbf{s}_{t-k}, \boldsymbol{\theta})\right)$, can not always be -1, the gradient of a state does not vanish through BPTT in a mini-batch.

Convolutional Blocks: Convolutional blocks are described in [14] and the theatrical analysis are shown in [19]. The structures of recurrent blocks and convolutional blocks are shown in Fig. 1.

C. Model and Training

Based on the residual recurrent convolutional encoder discussed above, we build the model for frame prediction. To improve the scale-invariance of our model, we introduce the inception structure [20]. After learning the representation from

(a) (b) (c) (d)

Figure 3. Comparison with on the Basketball Dunk and Ice Dancing clips from UCF101 [2] appearing in Ranzato's method [3] and Mathieu's method [4]. We display frame predictions for each method along with two zooms of each image.

the frame sequence, the deconvolution block is designed to restore a frame from representations, and a structure of u-net [21] improve the performance. The details and structure of the model are shown in Fig. 2.

In the training of the model, we use patches to accelerate the training and predict the frame with a complete frame sequence. The loss function is the weighted sum of mean square error (MSE) and gradient distribution.

III. EXPERIMENTS

The experiment runs on the Linux with a GPU of GTX970. The models are trained with the UCF101 dataset [2]. We predict the frame with three previous frames, and compare them with the state-of-the-art methods (Ranzato's method [3] and Mathieu's method [4]). The results are shown in Fig. 3. The comparative quantification is shown in Table 1. The results shows that our results are better than other methods. The scene prediction for auto-driving with the dataset from comma.ai [1] is shown in Fig. 4.

ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China (U1664264, U1509203, 61174114).

REFERENCES

[1] E. Santana and G. Hotz, "Learning a Driving Simulator," 2016.

[2] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild," *Computer Science*, 2012.

[3] M. A. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra, "Video (language) modeling: a baseline for generative models of natural videos," *Eprint Arxiv*, 2014.

[4] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," 2015.

[5] J. H. van Hateren and D. L. Ruderman, "Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex," *Proceedings Biological Sciences*, vol. 265, p. 2315, 1998.

[6] J. Hurri and A. Hyvärinen, *Simple-cell-like receptive fields maximize temporal coherence in natural video*: MIT Press, 2003.

[7] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: unsupervised learning of invariances," *Neural Computation*, vol. 14, p. 715, 2002.

[8] R. Memisevic and G. E. Hinton, "Learning to represent spatial transformations with factored higher-order boltzmann machines," *Neural computation*, vol. 22, p. 1473, 2010.

[9] X. Miao and R. P. Rao, "Learning the Lie groups of visual invariance," *Neural Computation*, vol. 19, p. 2665, 2007.

[10] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised Learning of Video Representations using LSTMs," pp. 843-852, 2015.

[11] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735-1780, 1997.

[12] J. Oh, X. Guo, H. Lee, R. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in Atari games," in *International Conference on Neural Information Processing Systems*, 2015, pp. 2863-2871.

[13] A. Gustavsson, A. Magnuson, B. Blomberg, M. Andersson, J. Halfvarson, and C. Tysk, "On the difficulty of training Recurrent Neural Networks," *Computer Science*, vol. 52, pp. 337-345, 2013.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[15] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 8624-8628.

[16] Y. Bengio, P. Frasconi, and P. Simard, "The problem of learning long-term dependencies in recurrent networks," in *IEEE International Conference on Neural Networks*, 1993, pp. 1183-1188 vol.3.

[17] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, p. 157, 1994.

[18] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, pp. 157-166, 2002.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," 2016.

[20] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," 2016.

[21] T. V. Eicken, A. Basu, V. Buch, and W. Vogels, "U-Net: A User-Level Network Interface for Parallel and Distributed Computing," *Acm Sigops Operating Systems Review*, vol. 29, pp. 40-53, 1995.

Figure 4. Scene prediction for auto-driving with the dataset from comma.ai [1].