Inception LSTM for Next-frame Video Prediction (Student Abstract)

Matin Hosseini,[1] **Anthony S. Maida, Majid Hosseini, Gottumukkala Raju**
[1]School of Computing and Informatics University of Louisiana at Lafayette
{mxh0212, Maida, Seyedmajid.hosseini1, Raju}@louisiana.edu

Abstract

In this paper, we proposed a novel deep-learning method called Inception LSTM for video frame prediction. A standard convolutional LSTM uses a single size kernel for each of its gates. Having multiple kernel sizes within a single gate would provide a richer features that would otherwise not be possible with a single kernel. Our key idea is to introduce inception like kernels within the LSTM gates to capture features from a bigger area of the image while retaining the fine resolution of small information. We implemented the proposed idea of inception LSTM network on PredNet network with both inception version 1 and inception version 2 modules. The proposed idea was evaluated on both KITTI and KTH data. Our results show that the Inception LSTM has better predictive performance compared to convolutional LSTM. We also observe that LSTM with Inception version 1 has better predictive performance compared to Inception version 2, but Inception version 2 has less computational cost.

Introduction

Video frame prediction has generated a lot of interest given its utility in several computer vision applications such as video transcoding, video frame synthesis, autonomous vehicle navigation and robotic motion planning. Both convolutional recurrent neural network and inception neural network architectures were proven to be quite successful in image processing. Recent work by Shi et. al. (Xingjian et al. 2015) and Lotter et. al. (Lotter, Kreiman, and Cox 2016) demonstrate the effectiveness of convolutional LSTM, where the affine weight multiplication is replaced with convolutions for processing image sequences. The standard convolutional LSTM has a fixed kernel size for all the gates. Inception neural networks (Szegedy et al. 2015) use multiple kernel sizes r for each convolution that has a different receptive field. This makes the inception network adaptable to varying scales of objects in the image.

This paper introduces a novel Inception-inspired LSTM architecture that uses multiple-kernels with different sizes for each input gate within the LSTM. The idea of using different kernel sizes to capture the magnitude of motion (i.e. slow vs fast motion) in sequence of images was originally inspired from convolutional LSTM from Shi et. al. (Xingjian et al. 2015). But the kernel size in convolution LSTM is typically set as a hyper parameter. In our proposed architecture, we incorporated multiple-kernels with different sizes. These kernels are passed to the gate activation functions by concatenating multiple kernels. The proposed model is implemented with two variations of inception, namely inception-v1 and inception-v2 networks. The proposed model is compared with two standard widely used video frame prediction data-sets, namely Kitti and KTH.

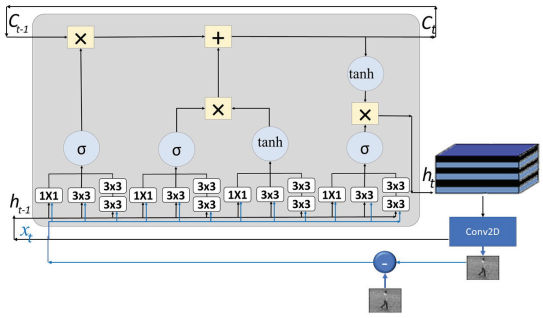

Figure 1: Our Inception-inspired version 2 LSTM module in one layer PredNet architecture.

The Inception LSTM

The general idea of Inception LSTM is to introduce an inception layer that has the ability to perform convolution with multiple kernel sizes instead of a single kernel. Inception v1 (Szegedy et al. 2016) architecture has 3 different kernels with sizes 1×1, 3×3, and 5×5 . In the case of Inception v2, the 5×5 is replaced with 2 stacked 3×3 kernels. Figure 1 shows the architecture of Inception LSTM cell with Inception v2. We can see that the modified inception module (without max pooling and the 1×1 kernel) is added to each gate in the LSTM cell.

The equations for Inception LSTM v2 are given by:

		KITTI	
Model	MAE	MSE	SSIM
ConvLSTM(2L)	0.053306	0.010216	0.811534
ConvLSTM(3L)	0.047526	0.008185	0.847114
ConvLSTM(4L)	0.045806	0.007612	0.857651
Incv1LSTM(2L)	0.049663	0.009095	0.833583
Incv1LSTM(3L)	**0.044761**	**0.007345**	**0.863714**
Incv1LSTM(4L)	**0.043640**	**0.007028**	**0.868226**
Incv2LSTM(2L)	0.049966	0.009114	0.831361
Incv2LSTM(3L)	**0.045021**	**0.007481**	**0.861877**
Incv2LSTM(4L)	**0.044115**	**0.007191**	**0.867645**
		KTH	
ConvLSTM(2L)	0.044115	0.007191	0.867645
ConvLSTM(3L)	0.010629	0.000449	0.961777
ConvLSTM(4L)	0.011604	0.000573	0.955879
Incv1LSTM(2L)	0.010624	0.000479	0.961024
Incv1LSTM(3L)	**0.010326**	**0.000406**	**0.963680**
Incv1LSTM(4L)	0.010959	0.000524	0.958901
Incv2LSTM(2L)	0.010752	0.000503	0.960048
Incv2LSTM(3L)	**0.010429**	**0.000438**	**0.962630**
Incv2LSTM(4L)	**0.010637**	**0.000463**	**0.961332**

Table 1: Performance on the KITTI and KTH data sets. Inc denotes the inception results and the number in the parenthesis indicates the number of layers.

$$i_t, f_t, g_t, o_t = \sigma \begin{bmatrix} W_{1\times1} * [x_t, h_{t-1}], \\ W_{3\times3} * [x_t, h_{t-1}], \\ W_{2i3\times3} * [W_{3i3\times3} * [x_t, h_{t-1}]] \end{bmatrix} \quad (1a)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (1b)$$

$$h_t = o_t \odot tanh(c_t) \quad (1c)$$

$W_{1\times1}$ denotes the weights for the corresponding gate and 1×1 shows the kernel size. The equations for i_t, f_t,g_t and o_t are similar and only the weights change respectively. The full set of equations are provided in (Hosseini et al. 2019) The output of the three convolutions (indicated by square brackets) are stacked and passed to the input gate. The cell state and recurrent connection (h) are similar to the original convolution LSTM.

Inception version 2 uses two 3x3 convolution kernels instead of one 5x5. Equations for Inception LSTM version 1 are provided below. The equation for f_t,g_t,o_t are identical, with the exception of weights.

$$i_t = \sigma \begin{bmatrix} W_{i1\times1} * [x_t, h_{t-1}], \\ W_{1i3\times3} * [x_t, h_{t-1}], \\ W_{5\times5} * [x_t, h_{t-1}]] \end{bmatrix} \quad (2a)$$

Experimental setup and results

The two variants of the Inception LSTM are implemented within the PredNet architecture (Lotter, Kreiman, and Cox 2016). The gate activation functions use hard Sigmoids similar to PredNet implementation. The proposed model is compared for 2-layer, 3-layer and 4-layer architectures. The number of channels vary for each layer. For instance, the 4-layer architecture has 3, 48, 96 and 192 channels. The source code for this implementation is made available at httpsgithub.com/matinhosseiny/Inception-inspired-LSTM-for-Video-frame-Prediction. The proposed models are compared using widely accepted KITTI (Geiger et al. 2013) and walking video data from KTH (Schüldt, Laptev, and Caputo 2004) data sets. Three quantitative measures were used for performance comparison. The Mean Squared Error (MSE), the Structural Similarity Index (SSIM) and Mean Absolute Error (MAE).

Table 1 provides performance comparison of the proposed Inception LSTM (version 1 and version 2) with convolutional LSTM for the KITTI and KTH data sets. Inception version 1 has the best performance with respect to MSE. The original PredNet uses 150 epochs for training. We use only 50 epochs in our model for all the experimental results. We can observe that Inception LSTM with 3-layer outperforms 4-layer convolutional LSTM.

Discussion and Conclusions

This paper presents Inception LSTM architecture for video frame prediction. We observe that larger range of kernels create a rich feature set that help improve prediction accuracy. The proposed model made some minor modifications to the original Inception module by removing the max pooling and 1x1 convolution. We observed that both versions of Inception-inspired LSTM show performance improvements when compared to the original convolutional LSTM.

References

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* 32(11):1231–1237.

Hosseini, M.; Maida, A. S.; Hosseini, M.; and Raju, G. 2019. Inception-inspired lstm for next-frame video prediction. *arXiv preprint arXiv:1909.05622*.

Lotter, W.; Kreiman, G.; and Cox, D. 2016. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*.

Schüldt, C.; Laptev, I.; and Caputo, B. 2004. Recognizing human actions: a local SVM approach. In *Proc. Int. Conf. Pattern Recognition (ICPR'04)*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, 802–810.