UPDATE: Implementing and Extending Inception-inspired LSTM for Next-frame Video Prediction

Krishna Palle
pallekc@bu.edu

1. Task

According to Project proposal, the initial scope of the project was to implement a paper, Inception-inspired LSTM for Next-frame Video Prediction[1] to test it on KITTY[2] dataset and extend it to multiple frames and see if video generation would be possible. Currently, ignoring the possibility of generating a video using the network by modifying it, the new task proposes to implement the [1] paper and implement a GAN loss if time permits to the architecture and see if the model performs better.

2. Related Work

After thoroughly going through the paper [1], it is understood that the paper is a modification to PredNet[3], a deep predictive coding network for video prediction and unsupervised learning. This paper borrows inspiration from 'predictive coding' from the neuroscience literature.

Diving in a little deep into the architecture of [3], it contains four basic parts, an input convolutional layer (A), a recurrent representation layer (R), a prediction layer (A_hat), and an error representation (E). The representation layer, (R), is a recurrent convolutional network that generates a prediction, (A_hat) of what the layer input, (A), will be on the next frame. The network takes the difference between (A) and (A_hat) and outputs an error representation (E). The error (E) is then passed forward through a convolution layer, to become the input to the next layer (A+1). The error (E) is also passed as the input to the (R) layer along with top-down input from the representation layer of the next level of the network (R+1). Below are the equations that define the network.

$$A_l^t = x_t \quad if \ \ l == 0$$
$$A_l^t = MAXPOOL(ReLU\left(CONV\left(E_{\{L-1\}}^t\right)\right) \ \ if > 0$$
$$\hat{A}_l^t = ReLU(CONV(R_l^t)$$
$$E_l^t = \left[ReLU\left(A_l^t - \ \hat{A}_l^t\right)\right]$$
$$R_l^t = ConvLSTM(E_l^{t-1}, R_l^{t-1}, UpSample(R_{l+1}^t))$$

As we see for the representation neurons, convolutional LSTM units were used. This model is trained on L1 loss for predicting the next frame prediction. Below, the architecture diagram gives a better understanding of the network.

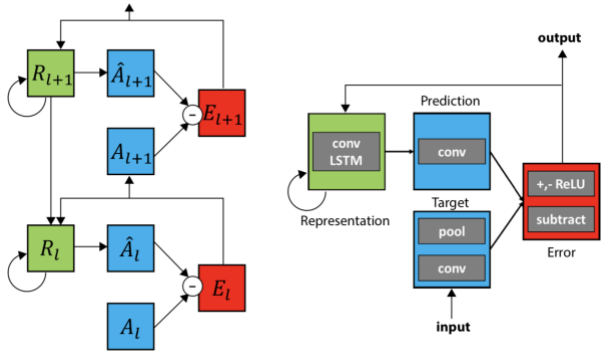

Figure 1: Predictive coding network(PredNet)

The paper, [1], picks up the same architecture and replaces the convolutional LSTM units with inception inspired LSTM units and replicates the same model.

Below are the formulas of the convolution LSTM. The difference between the standard LSTM to the convolution LSTM is replacing the dot product operations with convolutions as follows:

$$i_t = sigmoid(W_{ix} * x_t + W_{ih} * h_{t-1} + b_i)$$
$$f_t = sigmoid\left(W_{fx} * x_t + W_{fh} * h_{t-1} + b_f\right)$$
$$c_t' = i_t \odot tanh(W_{cx} * x_t + W_{ch} * h_{t-1} + b_c)$$
$$c_t = f_t \odot c_{t-1} + c_t'$$

$$O_t = sigmoid(W_{ox} * x_t + W_{oh} * h_{t-1} + b_o)$$
$$h_t = o_t \odot tanh(c_t)$$

Now, an inception inspired LSTM network can be explained with below equations

$$i_t = sigmoid(W_{i1x1} * [x_t, h_{t-1}], W_{i3x3} * [x_t, h_{t-1}], W_{i5x5} * [x_t, h_{t-1}])$$

$$f_t = sigmoid\big(W_{f1x1} * [x_t, h_{t-1}], W_{f3x3} * [x_t, h_{t-1}], W_{f5x5} * [x_t, h_{t-1}]\big)$$

$$g_t = sigmoid\big(W_{g1x1} * [x_t, h_{t-1}], W_{g3x3} * [x_t, h_{t-1}], W_{g5x5} * [x_t, h_{t-1}]\big)$$

$$o_t = sigmoid\big(W_{o1x1} * [x_t, h_{t-1}], W_{o3x3} * [x_t, h_{t-1}], W_{o5x5} * [x_t, h_{t-1}]\big)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

$$h_t = o_t \odot tanh(c_t)$$

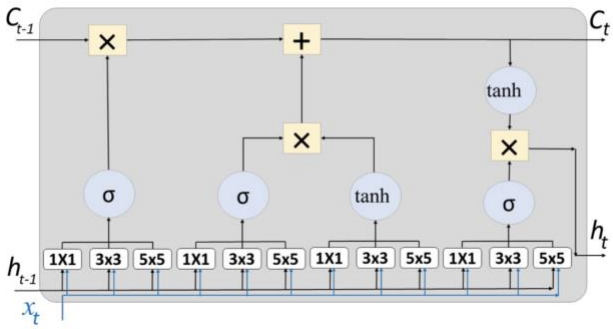

Figure 2: Inception inspired LSTM network v1

Instead of using a single kernel, inception inspired LSTM networks use multiple kernels of different sizes for each input gate.

Another variation of inception inspired LSTM network used in the paper was, instead of using a 5x5 kernel, they use 2 3x3 kernels and follow the same set of architecture as discussed above.

$$i_t = sigmoid\big(W_{i1x1} * [x_t, h_{t-1}], W_{1i3x3} * [x_t, h_{t-1}], W_{2i3x3} * [W_{3i3x3} * [x_t, h_{t-1}]]\big)$$

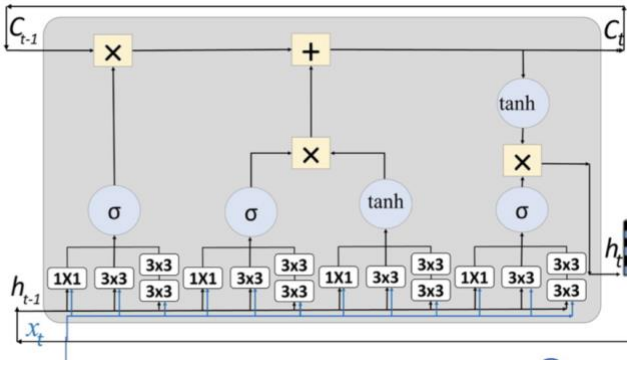

Figure 3: Inception inspired LSTM network v2

3. Approach

To implement the above mentioned network and train it on KITTY[2] dataset. The implemented code for referrals is at [4] for PredNet and [5] for inception inspired LSTM networks for next frame prediction.

If time permits, the extension of the project would be to implement a GAN loss to the same network and see if it improves result. To implement a GAN loss, we need generator and discriminator architectures. The generator architecture is what was discussed until now and for the discriminator, we can have a stack of the same inception inspired LSTM networks discussed above (Needs to discuss about this with the professor and look for inputs) and they can be trained together with a Minmax or Wasserstein Loss (need input).

4. Datasets and Metrics

For the above project, we plan to use KITTY [3] dataset, the computer vision benchmark suite. It is a video dataset where each frame has a resolution of 1392x512 pixels.

http://www.cvlibs.net/datasets/kitti/

The metric we can use to compare with the performance could be MSE and MAE, since those were used in the paper to evaluate the model.

5. Approximate Timeline

Task	Predicted Deadline
Implementation of inception inspired LSTM	04/20/2020
Implementing GAN loss to the architecture	04/25/2020
Taking feedback to improve results	04/27/2020

References

[1] Hosseini, M., Maida, A., Hosseini, M. and Raju, G. (2019). *Inception-inspired LSTM for Next-frame Video Prediction*. [online] arXiv.org. Available at: https://arxiv.org/abs/1909.05622 [Accessed 1 Mar. 2020].

[2] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Re- search*, 32(11):1231–1237, 2013.

[3] Wang, Y., Long, M., Wang, J., Gao, Z. and Yu, P. (2017). *PredRNN: Recurrent Neural Networks for Predictive Learning using Spatiotemporal LSTMs*. [online] Papers.nips.cc. Available at: http://papers.nips.cc/paper/6689-predrnn-recurrent-neural-networks-for-video-prediction-using-spatiotemporal-lstms [Accessed 1 Mar. 2020].

[4] GitHub. 2020. *Coxlab/Prednet*. [online] Available at: <https://github.com/coxlab/prednet> [Accessed 7 April 2020].

[5] GitHub. 2020. *Matinhosseiny/Inception-Inspired-LSTM-For-Video-Frame-Prediction*. [online] Available at: <https://github.com/matinhosseiny/Inception-inspired-LSTM-for-Video-frame-Prediction> [Accessed 7 April 2020].