Video Prediction

Independent Work Report (MAE 340, Spring 2021)

Matthew Coleman

April 26, 2022

This project represents my own work, in accordance with the University regulations.

/s/Matthew Coleman

Contents

1 Introduction

While humans cannot perfectly predict the future, they are indeed capable of inferring a great deal of information about near events in the future, and this knowledge greatly aids them in planning out their actions, such as which movements to take to reach a goal. This ability to forecast the future is a direct result of an understanding of causality that is learned through observation and interaction [2].

A great amount of human predictions are, of course, erroneous in major respects, but even the humans least adept at inferring far-off outcomes and consequences still are masters of learning very near-term ones. For example, humans have a good sense for where a car will move in the street, or which direction a pedestrian may continue walking. Even a young child can predict where to toss a football to a moving receiver, and even this small knowledge reveals an infinite wisdom compared to the most advanced video prediction methods.

The task of video prediction is comprised of several open challenges in computer vision; it uses some of the most recent model architectures that have been developed and it even contends directly with an impossible task altogether, which is to predict the future. Although it is a particularly confusing task, it also has the potential for immense impact and immediate practical applications, such as in autonomous driving [5], video interpolation [7] and most interesting in the context of this report, robotic control systems [3].

This project will examine the current state-of-the-art in video prediction models, report on several experiments carried out by implementing and testing such a model on various existing datasets, and attempt to make meaningful conclusions about video prediction and learning causality.

2 The Task of Video Prediction

The task of video prediction is to construct an approximation for the completion of a sequence of frames, given only the initial sequence of frames. Formally, given an ordered set of n image frames $\boldsymbol{X} = (X_1, X_2, X_3 \cdots X_n)$, the task is to predict the latter m frames of the sequence $\boldsymbol{Y} = (Y_1, Y_2, Y_3 \cdots Y_m)$, each frame of which having the same dimensions, for example with c channels, height h, and width w. At each inference in training, the model predictions $\hat{\boldsymbol{Y}} = \left(\hat{Y}_1, \hat{Y}_2, \hat{Y}_3 \cdots \hat{Y}_n\right)$, with the same dimensions as the inputs, are conditioned on the input sequence $\boldsymbol{X}$, and the model weights are updated typically by the gradient of a loss function computed between the predictions and ground truth sequence $\boldsymbol{Y}$ directly. Critically, since there is no human intervention or labeling required for the model to do this, and models typically are able to learn from the implicit temporal organiziation of the video data, video prediction is a self-supervised task [8].

3 Families of Prediction Models

Modern video prediction models tend to adopt several canonical architectures, which are normally simple and easily generalizable for specific tasks, such that they can be used as building blocks or blueprints within larger architectures. Understanding what each architecture seeks to do on a high level is imperative to understanding what kind of output one should expect from the model, since they are each very unique, and research implementations make use of them in different ways.

For example, RNNs and generative networks are each successful and well-tested paradigms used in many other machine learning domains outside of computer vision, but they are especially utilized within video prediction because of their key properties and strengths, particularly of RNNs to work on time-sequential data and of generative networks to "imagine" new data within a distribution. These paradigms are used extensively in Convolutional LSTM, FutureGAN [1], and SAVP models [6], as well as many other models in cutting-edge research. A short description of each family and its relevance to video prediction is given below:

3.1 Recurrent Neural Networks

Formally, a Recurrent Neural Network (RNN) is the end-result of a mathematical analysis of a nonlinear first-order non-homogeneous ordinary differential equation describing the time-evolution of a state signal $\vec{s}$ as a function of time, along with an input signal $\vec{x}$. The canonical statement of an RNN is given below in the form of a discrete Delay Difference Equation (DDE) [9]:

$$\begin{aligned} \vec{s}[n] &= W_s\vec{s}[n-1] + W_r\vec{r}[n-1] + W_x\vec{x}[n] + \vec{\theta_s} \\ \vec{r}[n] &= G(\vec{s}[n]) \end{aligned} \tag{1}$$

With W_s, W_r, and W_x as weight matrices which are either multiplied or convolved with their respective signals, and $\vec{\theta_s}$ as a bias term which is typically added in element-wise fashion to $\vec{s}$. This equation also includes the read-out or output signal $\vec{r}$, which is the activation $G(z)$ of the state signal, and can be seen as the "output" of the network. All together, this equation describes all of the moving parts of an RNN, however, it can be written more succinctly in the following form, ignoring the cell's signal from memory $\vec{s}[n-1]$, since its affect on the signal's trajectory is negligible as long as $\vec{r}[n-1]$ is still present:

$$\begin{aligned} \vec{s}[n] &= W_r\vec{r}[n-1] + W_x\vec{x}[n] + \vec{\theta_s} \\ \vec{r}[n] &= G(\vec{s}[n]) \end{aligned} \tag{2}$$

A more understandable depiction of the RNN described in Equation 2 can be found in Figure 1. This figure also shows the "unfolding" or "unrolling" of the model, which is just a way of seeing each time-step of the state signal $\vec{s}[t]$, input signal $\vec{x}[t]$, and ouput signal $\vec{r}[t]$ (now given by $\vec{y}[t]$) as they are produced over time.

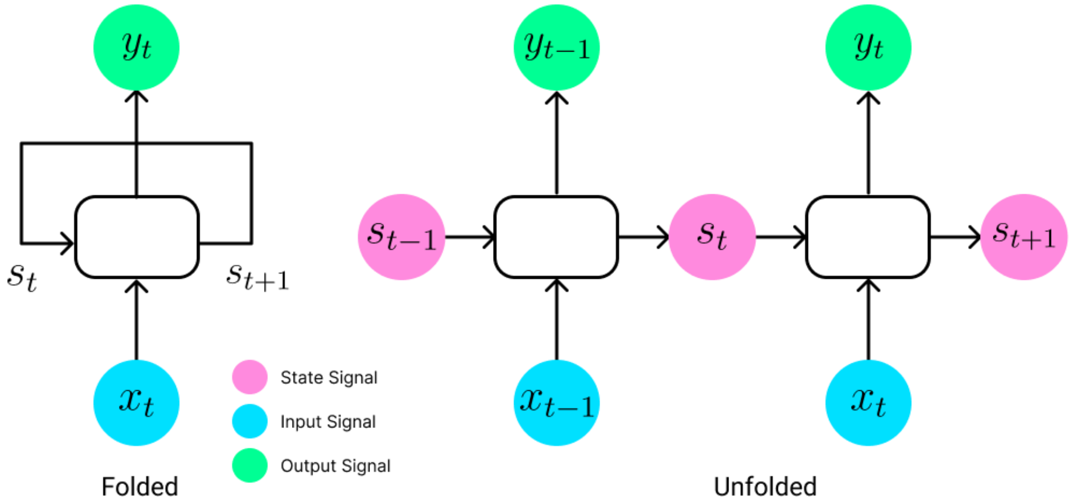
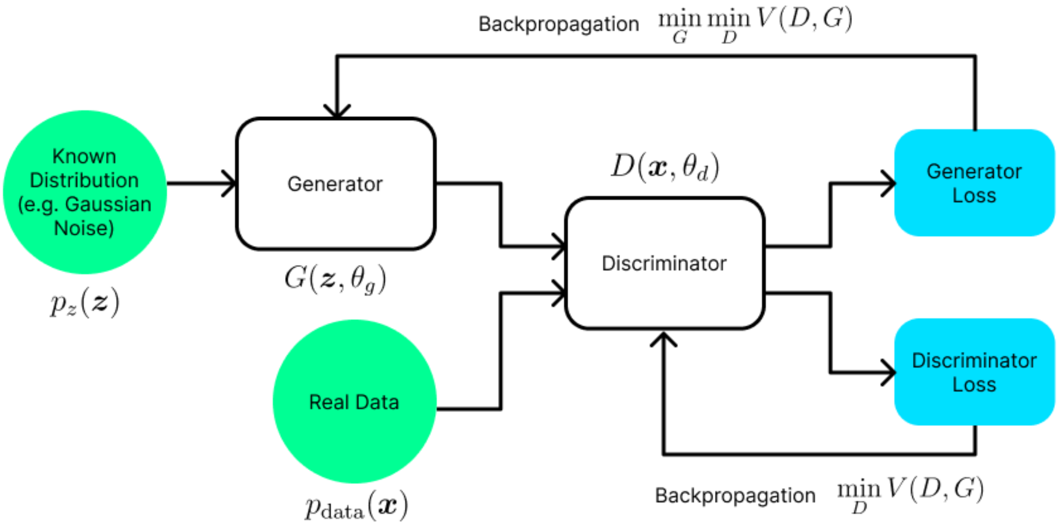


Figure 1: RNN Architecture

Recurrent neural networks (RNNs) consist of a network of nodes that stretches over some sequential information, typically in the form of a time sequence (Which is the case in video prediction). Each node will perform another learning technique on the data in sequence (This could be a convolution operation, a linear layer, or a combination of several, for example) and output its own activation. Commonly, this is implemented not with a network of individual nodes but rather in the form of a feedback loop over a single node, which passes some data embedding forward through the network to itself in a loop (a hidden state, or variable), only taking in new data from the original input sequence at each step. In this way, an RNN equipped with convolution is capable of learning from time-varying information while preserving spatio-temporal relations (That is, relationships in the data that exist over space, such as the shape of a person's leg and hip, as well as relationships in the data that exist over time, such as the motion of a person walking, will be preserved in the final activations of the network).

The outputs of each node are then used for other purposes, depending on the task, and the result is then backpropagated against a loss function. In machine learning terminology, this is referred to as backpropagation through time (BPTT) [9], since a gradient must be computed in the input sequence's reverse order, i.e., backwards through time, and in the case of an RNN implemented with feedback, this gradient must be computed with respect to the input data at each time step and then added to the weights of the single node in sum.

3.2 Generative Models



Backpropagation $\min_G \min_D V(D,G)$
Known Distribution (e.g. Gaussian Noise)
$p_z(\boldsymbol{z})$
Generator
$G(\boldsymbol{z},\theta_g)$
$D(\boldsymbol{x},\theta_d)$
Discriminator
Generator Loss
Discriminator Loss
Real Data
$p_{\text{data}}(\boldsymbol{x})$
Backpropagation $\min_D V(D,G)$

Figure 2: General GAN Architecture

Generative neural networks (for example, GANs) consist of mostly the same architecture as discriminative neural nets, such as the ones which are classically used for image classification. While discriminative nets seek to learn the conditional probability $p(y \mid x)$ of an input x belonging to a particular class y, generative nets seek to learn the conditional probability distribution $p(x \mid y)$ of an input data given the output, allowing them to make inferences in the form of “imagined” data that might belong to the same distribution as x [4]. In short, discriminative models would look at many Van Gogh paintings and fakes in order to learn to differentiate between them, and generative models would look at many Van Gogh paintings in order to learn how to paint like Van Gogh.

In practice, this is done by passing a low-dimensional embedding vector from a known latent distribution (such as the kind that may come as an activation from a discriminative net) through a typical network in reverse, generating an upscaled output in the same shape as the targeted learning data.

4 Video Prediction Models

4.1 Convolutional LSTM

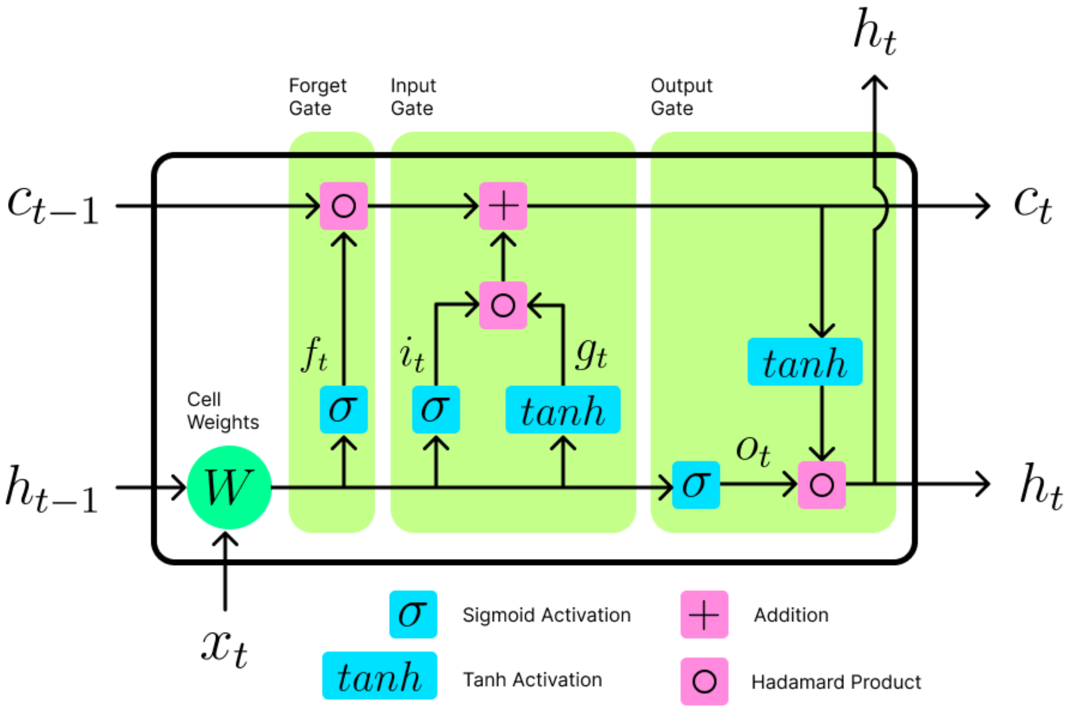


Figure 3: LSTM Cell Architecture

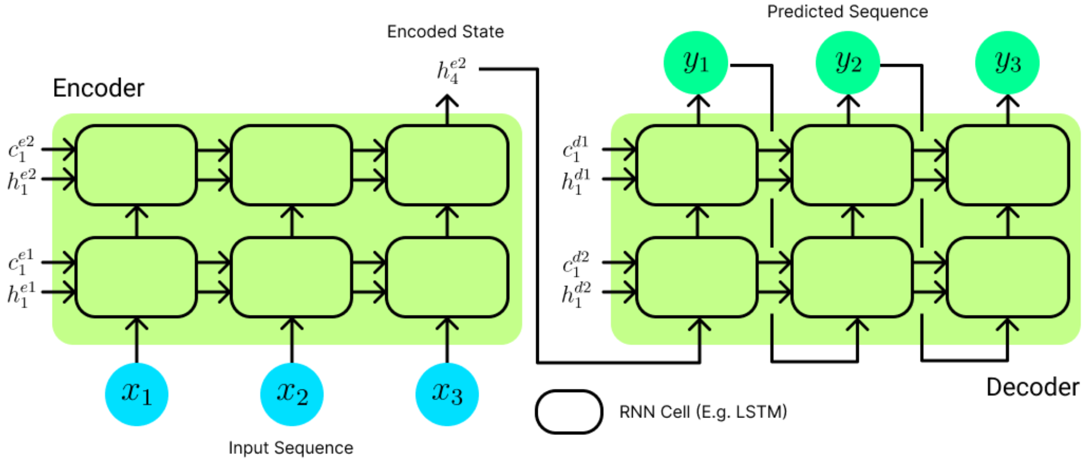


Figure 4: Seq2Seq Architecture

4.2 FutureGAN

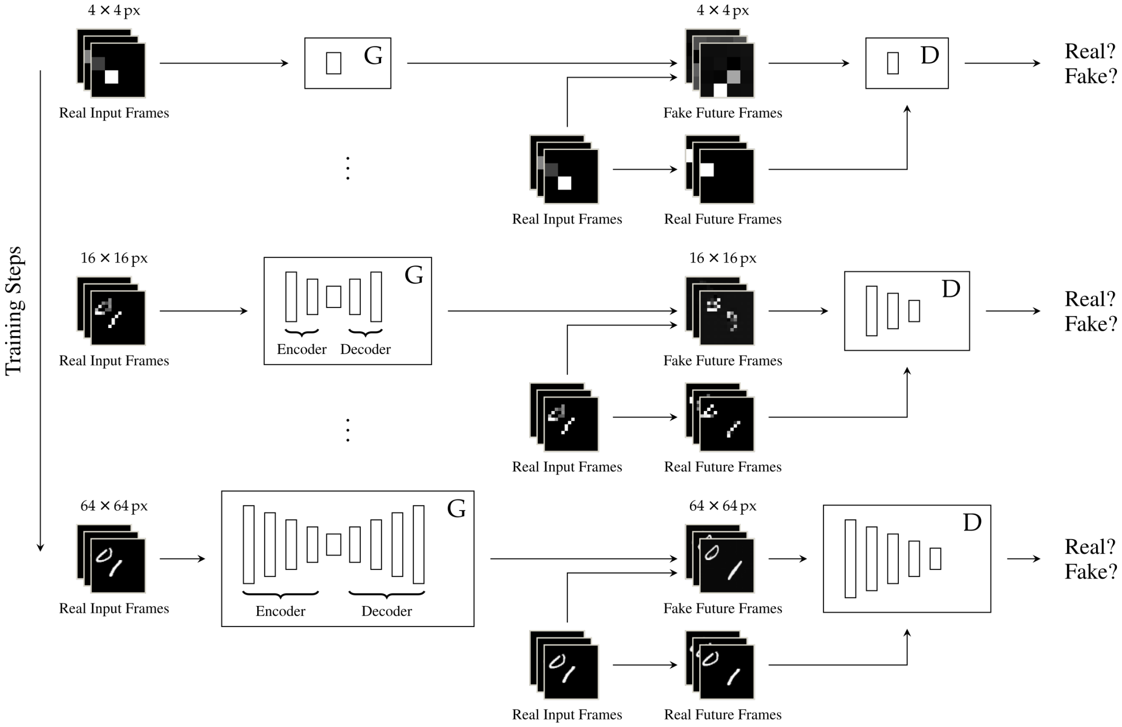


Figure 5: FutureGAN Architecture

4.3 SAVP

5 Datasets

5.1 MovingMNIST

5.2 KTH

5.3 BAIR

6 Experiments

The main experimental procedure carried out in this report is the training and testing of the re-implemented Convolutional LSTM model on the MovingMNIST, KTH, and BAIR datasets, however, in order to more gain a more robust understanding of LSTM features and limitations, as well as to debug the training mechanisms in a much simpler setting, it was useful to first test a Linear LSTM model on one-dimensional sequential data before moving fully to video prediction.

This Linear LSTM was implemented with nearly the same exact architecture as the Convolutional LSTM, however with Linear layers in place of convolutions, and as a result, a 2D convolution layer as the final step as opposed to 3D.

6.1 Sequence Prediction

Two generated datasets of one-dimensional sequences were implemented, one which was composed of sin waves with random frequency and phase offset, and another which represented random points chosen at several steps within the sequence and simply interpolated using a sinusoidal interpolation function. Both datasets were normalized to the range $(0, 1)$, with 20 total data points for each seqeuence. The model was then given the first 10 points and tasked to predict the last 10.

6.1.1 Generated Sinusoids

Figure **??** shows the model's inferences on a held-out set of the generated sinusoidal data as the model trains over one epoch.

6.2 Video Prediction

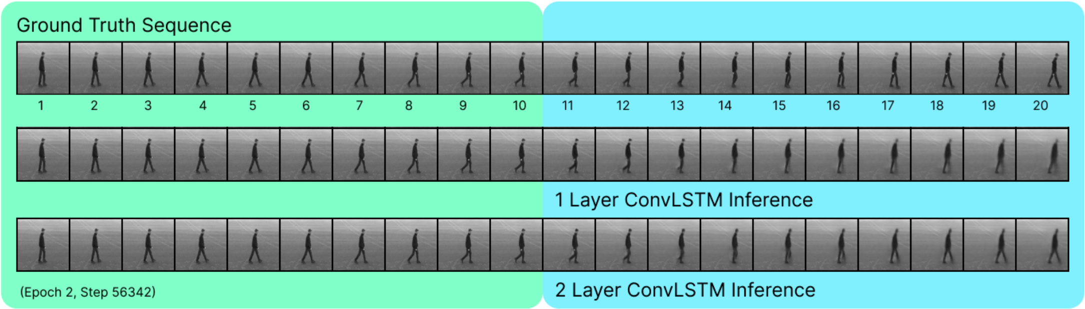


Figure 6: Convolutional LSTM Inference on KTH dataset

7 Conclusion

References

[1] S. Aigner and M. Körner. Futuregan: Anticipating the future frames of video sequences using spatio-temporal 3d convolutions in progressively growing gans, 2018. URL https://arxiv.org/abs/1810.01325.

[2] A. Cleeremans and J. Mcclelland. Learning the structure of event sequences. *Journal of experimental psychology. General*, 120:235–53, 10 1991. doi: 10.1037//0096-3445.120.3.235.

[3] C. Finn and S. Levine. Deep visual foresight for planning robot motion, 2016. URL https://arxiv.org/abs/1610.00696.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

[5] A. Hu, F. Cotter, N. Mohan, C. Gurau, and A. Kendall. Probabilistic future prediction for video scene understanding, 2020. URL https://arxiv.org/abs/2003.06409.

[6] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. *CoRR*, abs/1804.01523, 2018. URL http://arxiv.org/abs/1804.01523.

[7] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow, 2017. URL https://arxiv.org/abs/1702.02463.

[8] S. Oprea, P. Martinez-Gonzalez, A. Garcia-Garcia, J. A. Castro-Vargas, S. Orts-Escolano, J. Garcia-Rodriguez, and A. Argyros. A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2020. ISSN 1939-3539. doi: 10.1109/tpami.2020.3045007. URL http://dx.doi.org/10.1109/TPAMI.2020.3045007.

[9] A. Sherstinsky. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *CoRR*, abs/1808.03314, 2018. URL http://arxiv.org/abs/1808.03314.