Fluid Surface Height Estimation

UNIVERSITY OF MICHIGAN

Julien Faro, Max Froelich, Chun-Wei Kong, Michael Wolf, Samuel Worsham

{jfaro, mfroe, chunwei, mjwolf, shworsh} @umich.edu

EECS 545: Machine Learning
Winter 2022

Overview

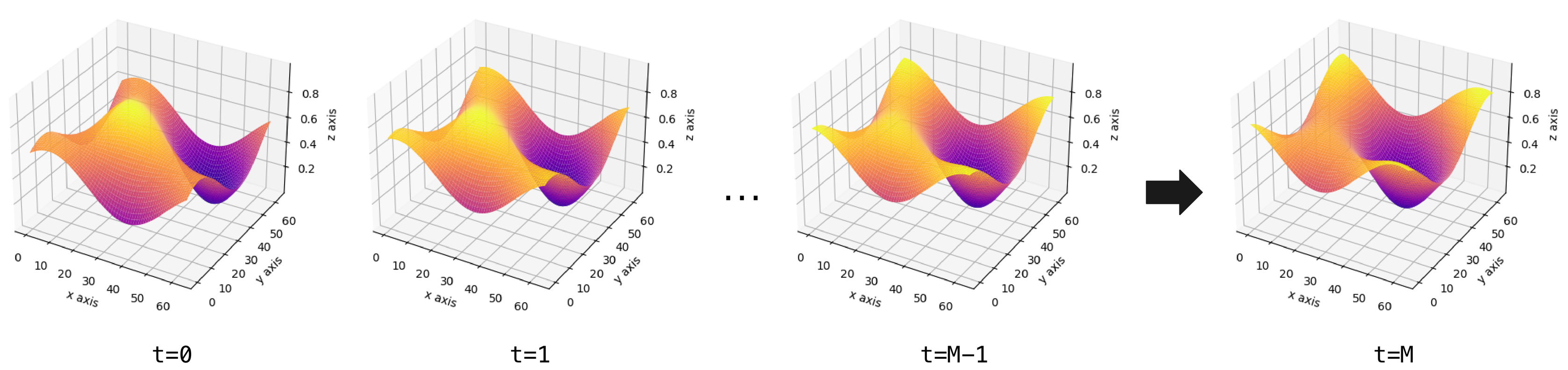
Computational fluid dynamics (CFD) simulations allow engineers to simulate the behavior of fluids, which is essential in many fields such as aerospace engineering and computer graphics. Efficiently simulating fluid behavior is challenging, as it is highly nonlinear, involving complex interactions between many equations.

Our research considers a different approach to fluid simulation; instead of explicitly defining a simulation's equations, we aim to develop a machine learning architecture that **learns to simulate fluid behavior.**

We describe, implement, and evaluate the results from two different learning model architectures for this task: Convolutional LSTM (ConvLSTM) and Reinforcement Learning CNN (RL-CNN).

Data

Our data is **spatio-temporal**, meaning that it consists of a time-sequence of spatial frames. Specifically, the data **represents the motion of fluid surfaces over time** through a sequence of `WxL` grids of height values. For training, `M` frames from a sequence are used as input, and the model is expected to predict the following frame.

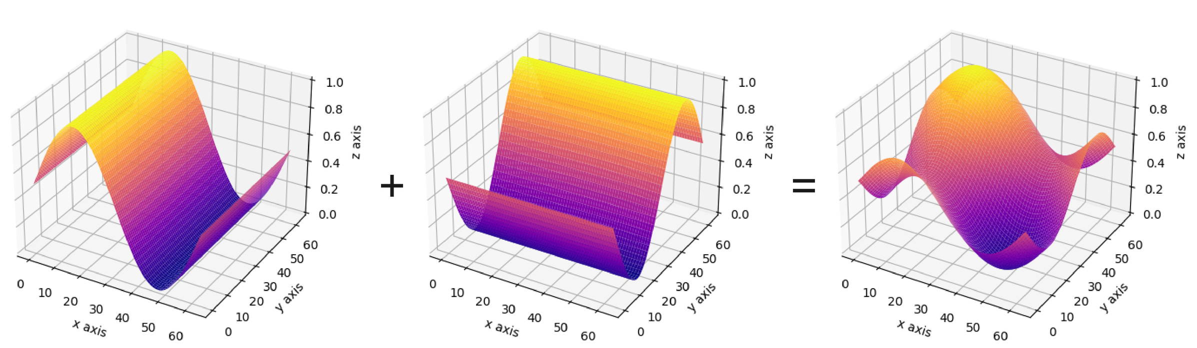


The model is trained to minimize the difference between each predicted frame and its corresponding target frame, using mean squared error (MSE) as the loss function.

$$L = \frac{1}{N}\sum_{i=1}^{N} ||\hat{Y}^{(i)} - Y^{(i)}||^2$$

Data Generation and Augmentation

We generate a large set of examples by augmenting a simple travelling sine wave. Augmentation is achieved through **rotations**, **varying speeds and frequencies**, and **combinations of examples** with all of the previously mentioned augmentations applied. Height values are normalized to the range `[0,1]`.



Additionally, we use a CFD fluid simulation program to generate many **realistic examples** with more complex dynamics.

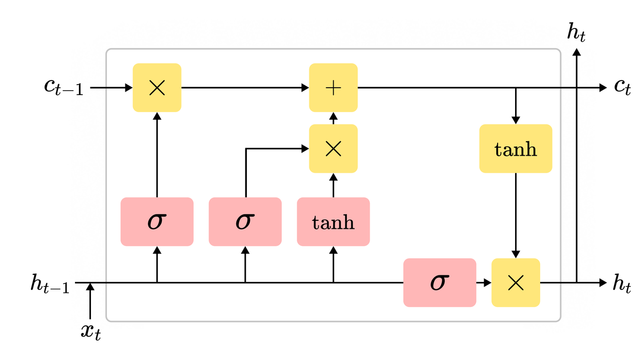
Neural Network Design

We implemented several different model architectures and evaluated their performance to determine the best design for the context of this problem.

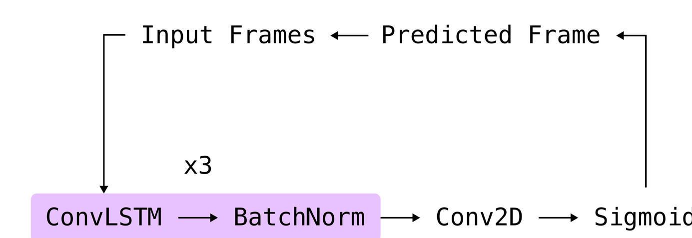
Convolutional LSTM

ConvLSTM is a variation of the long short-term memory (LSTM) architecture. Our model predicts a single frame from 10 input frames. Training involves a supervised learning approach, where weights are adjusted after each prediction to improve the error for future predictions. To generate a **sequence of predictions**, each predicted frame is simply appended to the sequence of inputs and the 10 most recent frames are used at each iteration. The model is designed as follows:

- 3 ConvLSTM layers with batch normalization in between
- Convolutional 2D layer to compute a prediction frame from the final output state
- Sigmoid layer to adjust pixel values to be between `0` and `1`



Structure of a single standard LSTM cell



ConvLSTM model architecture

Reinforcement Learning

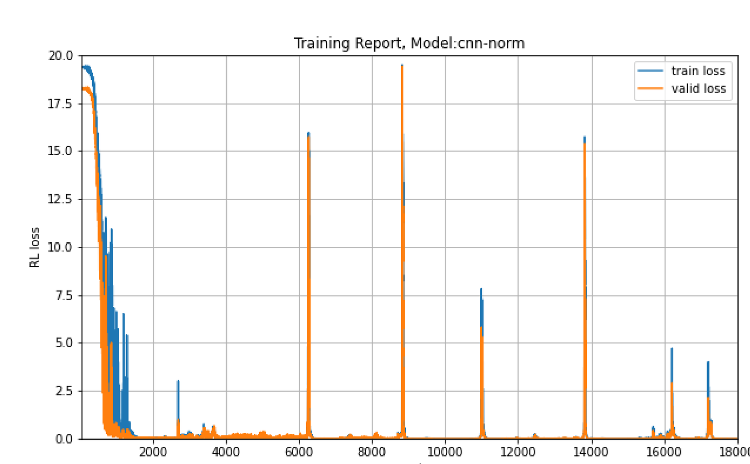
The ConvLSTM approach presented several issues. In addition to taking very long to train, it was only able to consider a single frame at a time, leading to high error in predictions after the first in a sequence. To address this, we re-formulated the problem through the lens of reinforcement learning (RL).

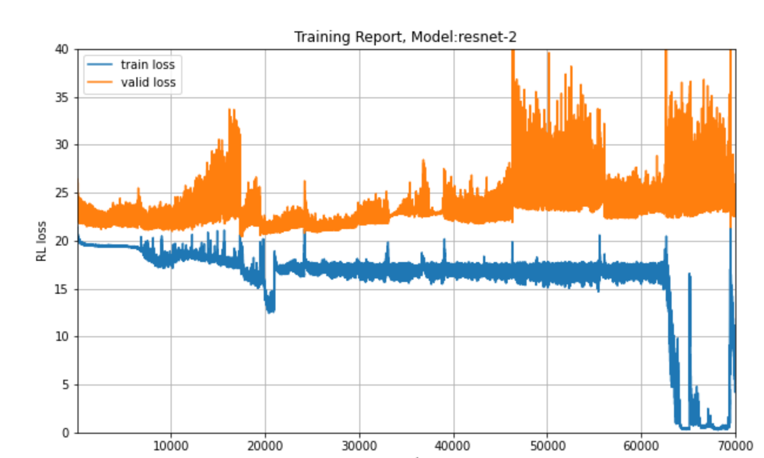
We attempted training a few different CNN architectures, including one with two convolutional layers and three fully-connected layers, and Resnet-18.

Pseudocode

1: Start epoch
2: Initialize *obs* from the training dataset; $step$=0
3: **while** epoch not terminated do
4: action = policy(obs)
5: $out = obs(end) + action$
6: $obs_{new}[0, 1, \ldots, 8] = obs[1, 2, \ldots, 9]$
7: $obs_{new}[9] = out$
8: $error$ += MSE(out, $target$)
9: $step$ += 1
10: **end while**
11: loss = R($error$, $step$)
12: Update policy from training RL loss
13: Save best policy from validation RL loss
14: End epoch

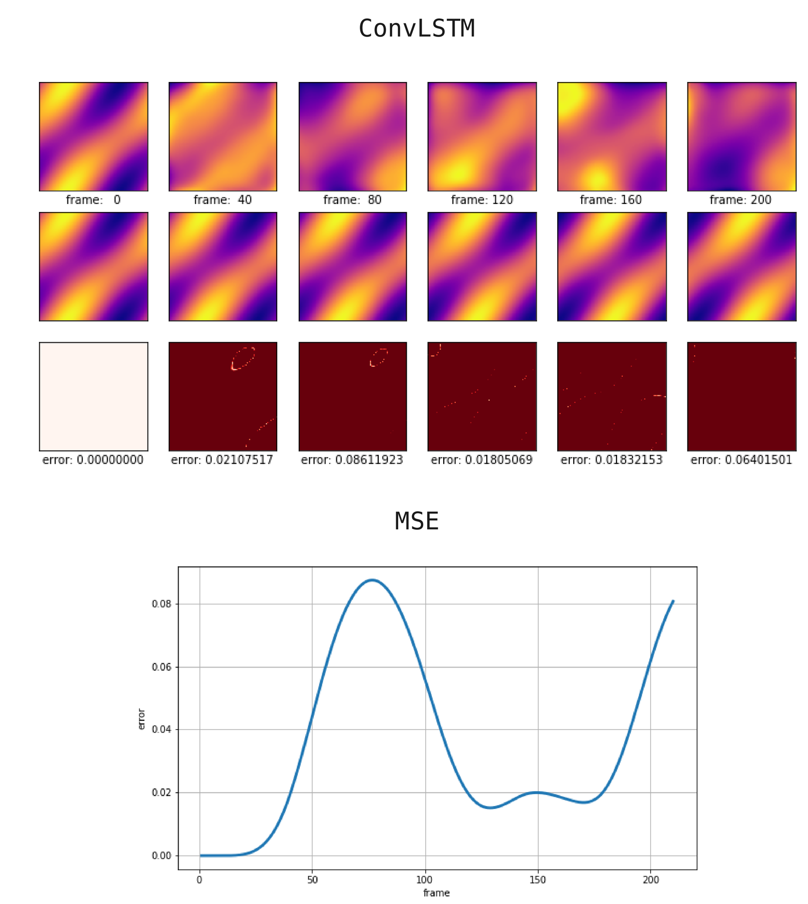
Training History

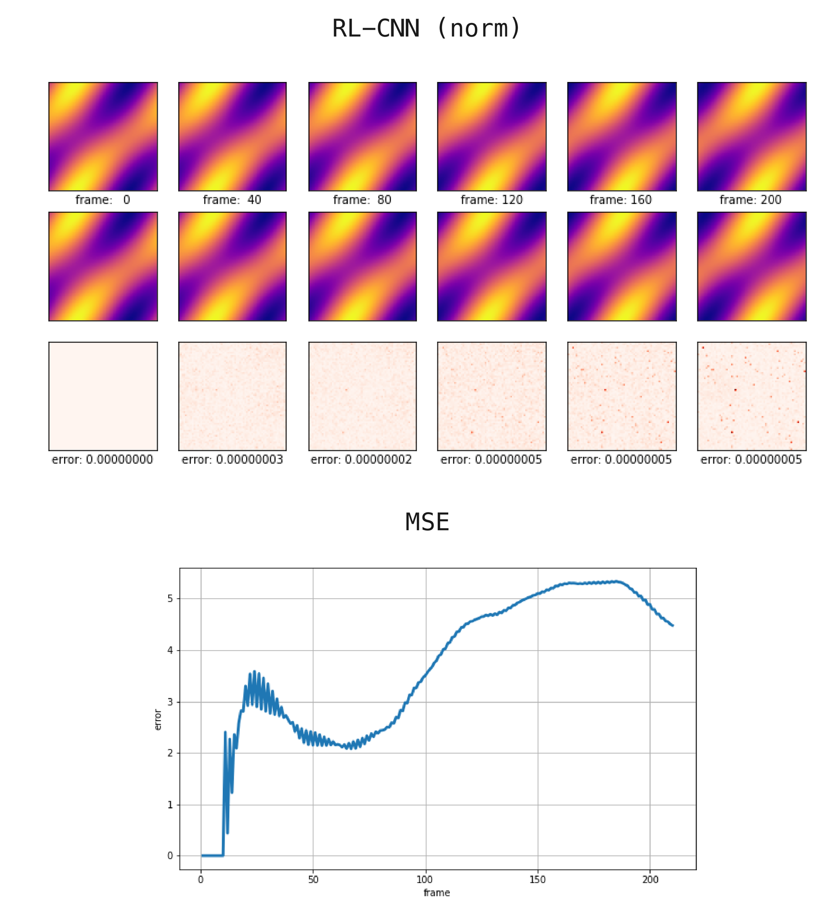




Results

Our **RL method trains faster and outputs more accurate predictions** than the ConvLSTM approach. Here, we compare ConvLSTM to RL-CNN in the task of predicting 200 frames when given 10 frames as input from an unseen sequence. In both examples below, the top row visualizes **predicted frames,** the middle row is the **ground-truth**, and the bottom row is **error**.

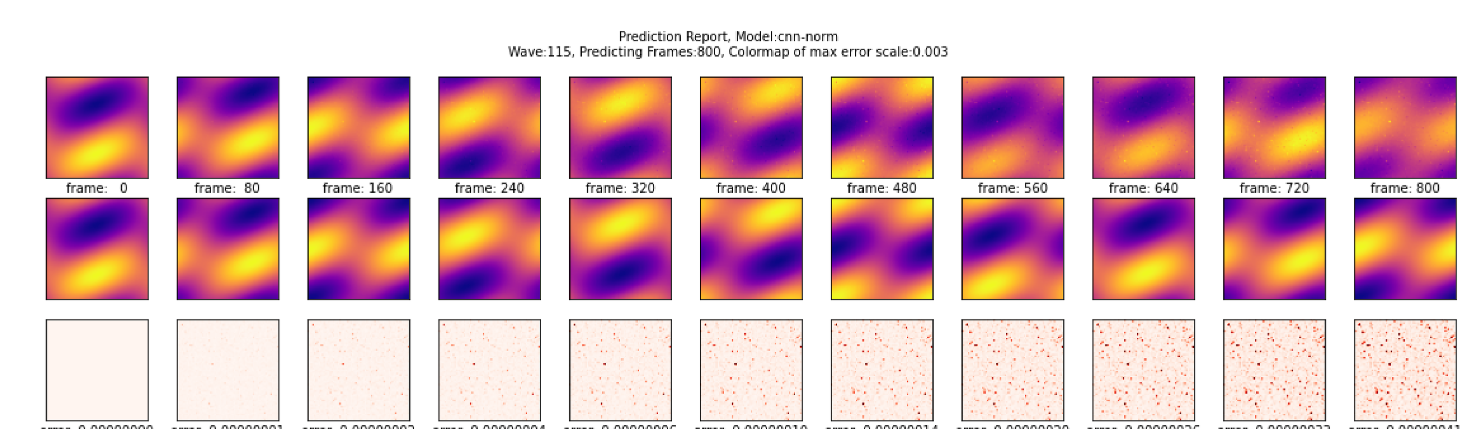




From the models we experimented with for the RL policy, we found that a simple normalized CNN performed best at the task of predicting 200 frames.

Policy	Final frame error	Validation loss	Training loss
resnet	1177.0111	20.8657	19.47
cnn	1.5983	2.2974	1.5983
cnn (norm)	1.0236	0.0009	0.0010

Conclusion + Discussion



- The RL method is able to **accurately predict >200 frames**, given 10 frames as input
- The **ConvLSTM approach cannot observe its own prediction error** during training, resulting in poor outputs for long sequences of predictions
- The RL approach allows us to specify how long we want to predict (specified by the target frame during training) and **how much cumulative error should be tolerated** during the entire prediction process (specified by the stop criteria), which is very powerful for the task of predicting long sequences of frames

References

[1] D. Grattarola, L. Livi, and C. Alippi. Learning graph cellular automata. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, Advances in Neural Information Processing Systems, 2021.
[2] Z. X. Ning P, Zhang C and J. X. Short- to medium-term sea surface height prediction in the bohai sea using an optimized simple recurrent unit deep network. Frontiers in Marine Science, September, 2021.
[3] H. N. Z. Y. e. a. Song, T. Application of deep learning technique to the sea surface height prediction in the south china sea. Acta Oceanologica Sinica, 40:68–76, 2021.
[4] Panda and Rohit. "Video Frame Prediction using ConvLSTM Network in PyTorch". Medium article. 2021. url: https://sladewinter.medium.com/video-frame-prediction-using-convlstm-network-in-pytorch-b5210a6ce582.
[5] Shi, Xingjian et al. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. CoPR, August 2018.