CS230

Precipitation Nowcasting using Deep Learning Techniques
(Computer Vision)

Yash Gaur
Department of Civil & Environmental Engineering
Stanford University
yashgaur@stanford.edu

Anika Jain
Department of Chemical Engineering
Stanford University
anikaj@stanford.edu

Georgios Sarmonikas
Department of Computer Science
Stanford University
gs7776@stanford.edu

Abstract

Nowcasting Precipitation is a field in meteorology which is about precisely predicting the short-term rainfall intensity of a local region and it plays a crucial role in multiple sectors of society. Due to its importance, the field is gaining traction in the artificial intelligence community which is aiming towards creating more accurate models using the latest algorithms and methods. There are many neural network techniques already tried, therefore this work explores different deep learning architectures by combining CNN and LSTM neural networks, using a radar echo dataset. We have developed three models with different architectures and tested them against the synthetic moving mnist dataset, prior to validating on precipitation nowcasting and concluded that stacked ConvLSTMs with residual input give satisfactory results but could also be improved with by combining other more specialised architectures.

1 Introduction

Our main idea is to predict the most possible sequence of future images given a sequence of input frames. Using as the input sequence radar echoes of precipitation, the predicted images will be the precipitation nowcasting. The radar images used are consecutive 6-minute interval radar echo frames and so according to the definition of nowcasting a single image prediction - 6 minutes in the future - is considered a "nowcast". In this work we attempt to predict 6-10 images which translate to 36 mins - 1 hour in the future. We used Convolutional LSTM and encoding-forecasting architecture to make the predictions inspired by [8].

2 Challenges

Nowcasting is a non-trivial problem and has emerged as a hot topic in meteorology. The problem is challenging because the spatio-temporal sequences have high dimensionality and we try to make multi-step predictions. Another reason is the chaotic nature of atmosphere which make predictions very unreliable and inaccurate. In addition, precipitation if highly influenced by temperature, wind,

CS230: Deep Learning, Winter 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

humidity and other atmospheric factors which cannot be captured by radar echo images, making the nowcasting more complicated than just predicting the next image. Preprocessing the large amounts of (often noisy images due to radar faults or artefacts) high resolution weather data was also a challenge in this project.

3 Related work

Existing methods for precipitation nowcasting usually use complex atmospheric simulation models which are slow. Recently, computer vision techniques and Numerical Weather Prediction (NWP), have proven useful especially Optical Flow based methods [7]. However, NWP models are less suitable for short-term forecasting since are based in complex equations, have high computational cost and can result in prediction errors in nowcasting due to rapid weather changes. Optical flow methods, despite having limitations, can achieve more accurate predictions and are usually preferred in precipitation nowcasting systems.

A new - data driven - approach using neural networks was proposed by Shi et. al who used a Recurrent Neural Network(RNN) [8] in which they proposed encoder-decoder architecture using convolutional long short-term memory (ConvLSTM) layers. ConvLSTM use its recurrent neural network architecture to memorize temporal information in image sequences and extract the spatial feature map by using convolutional operations. This project work is inspired by the paper in [7] and used ConvLSTM layers in our developed models.

4 Experiments on Moving-MNIST dataset

We started to test different deep neural network architectures on Moving-MNIST dataset in which we tried to predict one time step ahead in model0 in our github repository [4] based on the github repository `https://github.com/Linusnie/convLSTM_movingMNIST`. The encoding and forecasting architecture is used to make predictions and an image showing the comparison between ground truth and prediction is shown below.

(a) Example 1 (b) Example 2

Figure 1: *One time step prediction using encoder-forecasting network and mean squared error.*

We also tried to make multi-step predictions using model 1 and model 2 in our github repository [4]). We used binary cross-entropy loss function and Adam optimization but the predictions got worse as the number of predicted frames increased like shown below

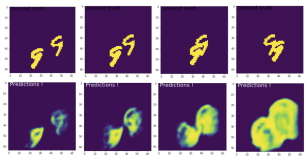


Figure 2: *Model 1, four-timestep predictions using binary cross entropy. Top row is ground truth and bottom row are predictions. Time flows from left to right.*

Model 2 is inspired by [6]. It captures the spatial layout of an image and the corresponding temporal dynamics independently. We used content encoder and motion encoder and combined the two to send to the decoder layers of architecture. We used mean squared error loss function and we show the multi-step prediction using this architecture below

Figure 3: *Model 2, eight-timestep predictions using mean squared error. Top row show ground truth. Bottom row, first eight images are input sequence to the model and the next eight are predictions.*

5 Dataset and Features

The dataset we used is publicly available on Harvard Dataverse [3]. It is from radar echos captured from the Weather Surveillance Radar-1988 Doppler Radar (WSR-88D, NEXRAD) located in Pudong, Shanghai. It contain 170,000 weather radar intensity frames collected from October 2015 to July 2018 with a minimal interval of 6 minutes between consecutive images. Each frame is a 501 x 501 pixel grid image, covering almost 501 x 501 square kilometers. The dataset despite the large volume of radar images, contained too many images with no precipitation content leading to no training of our models. In addition many images that had precipitation content had also significant radar noise in non-consecutive frames due to radar faults. That resulted in bad predictions in our baseline model. To address these challenges, for training and testing our models we picked a day and chose 80 sequenced images that had a good representation of precipitation content and less or no radar noise. Image preprocessing was applied to the images, in particular re-scaling from 501x501 to 64x64 pixels (mainly to increase the performance), zero-centering and cropping. The processed images were sampled and generated two array sequences shifted by a frame in order to have a sliding window necessary for the predictions. The sliding window length is 16 images long which translates to 96 minutes. Lastly we chose a validation set to be 10% of the total set.

6 Methods & Algorithms

To tackle the problem of precipitation nowcasting, we used convolutional LSTM network architectures, similar to [8]. Convolutional networks are good at pattern recognition and extracting temporal image features while LSTM networks have "memory" properties. The combination of these two is expected to find patterns in radar images sequences and predict the expected precipitation. Our model takes as input not just the current input example but also the historical perceived input. The model will preserve information from sequence images that has passed through it using the hidden states. LSTM has an advantage over traditional feed-forward neural networks and RNNs because they selectively "remember" patterns for longer periods.

The architecture in model 0 uses Conv2DLSTMCell which are stacked using MultiRNNCell function and dynamicRNN is used to get all encoder states and final encoder states which gives predictions. 2D convolution is performed on predictions to decode the final predictions and mean squared error is used as loss function.

The architecture in model 1 uses four ConvLSTM2D layers and one final Conv3D layer in sequence. The ConvLSTM layers have 64,64,64,64,1 filters in respectively. Each ConvLSTM layer use filters of size (5,5) with same padding and stride one. Activation function is tanh for all hidden layers and sigmoid for the last layer. We used logcosh loss and Adam optimization.

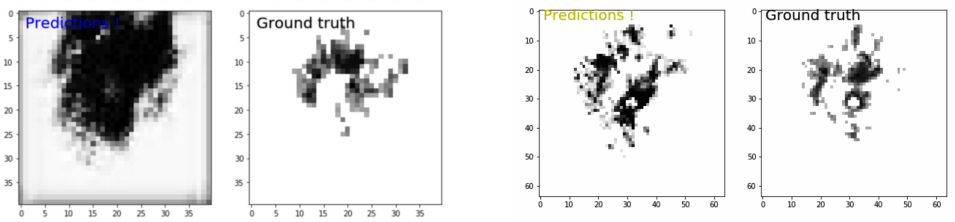
We use Conv2D and MaxPooling layers in model 2 to encode content and Conv2D and MaxPooling layers to encode motion and then we concatenate content encoding and motion encoding layers. Then we use decoding layers to obtain predictions. Here again, we used mean squared error and Adam optimizer.

We used model 0, model 1 and model 2 to make predictions on moving-MNIST data. We also used model 1 and model 2 to make predictions on precipitation data but the predictions do not have great spatial resolution using these architectures, mainly due to rescaling to 64x64 resolution to improve the performance. So, we also tried another architecture (Model1, test-model) in which we stacked multiple ConvLSTM2D layers and used batch normalization after every ConvLSTM2D layers. We used 64 filters of size (3,3) in all the ConvLSTM2D layers and a last layer in which we performed 3D convolution using 1 filter of size (3,3,3). We used linear activation in the last layer and used Mean

Squared Error as the loss function. We show the prediction results on precipitation problem using these architectures in results section.

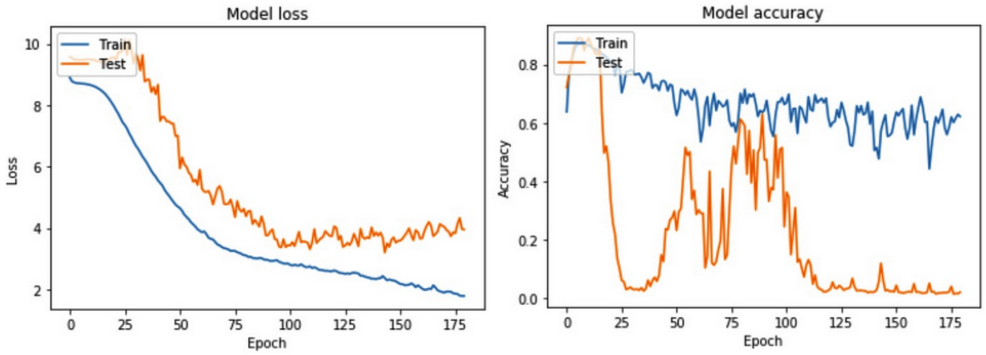
7 Results & Evaluation

We tested our code using different cost metric like squared (MSE) loss and binary cross-entropy (BCE) loss but those gave bad predictions. So we decided to use logcosh function, as proposed by [2] and that helped with both training loss and accelerated training. We also tried DICE and IOU as loss functions but without getting satisfactory predictions. In all models we used "Adam" optimization and “He Normal" initialisation. Our results are shown below for eight-timestep predictions.

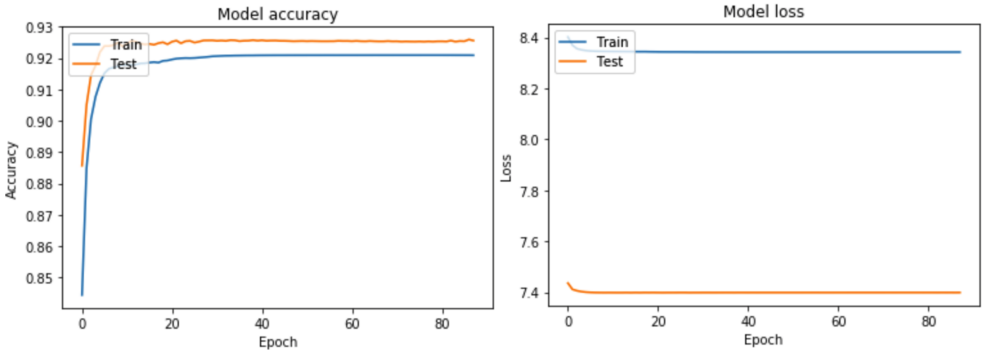


(a) MSE loss (b) Logcosh loss

Figure 4: *Model-1v2 - Prediction of a precipitation image. We show only the first time step prediction.*



(a) MSE loss



(b) Logcosh loss

Figure 5: *Model-1v2 - Loss and Accuracy plots*

Several metrics beyond loss and accuracy were used to evaluate the models:
SSIM (Structural Similarity Index) measures the perceptual difference of two similar images x and y,

FAR (False Alarm Ratio) = False Positive / (True Positives + False Positives + smooth factor), give a measure of reliability,
POD (Probability of Detection, or Hit Rate) = (True Positives + smooth factor) / (True Positives + False Negatives + smooth factor) give a measure of discrimination.
A smooth factor with value=0.0001 is also used to smooth the results and avoid any potential division by zero. The evaluation of the models against these metrics is shown in the table below:

Precipitation Models Evaluation:

		Metrics / Skills										
Model Name	**Type**	**Accuracy**	**SSIM**	**POD**	**FAR**	**Epochs**	**Learning Rate**	**Batch Size**	**Loss Function**	**Optimizer**	**beta1**	**beta2**
Model1_v1	Stacked ConvLSTM	80%	40.94%	0.97	0.22	80	0.001	2	MSE	Adam	0.9	0.999
Model1_v2	Stacked ConvLSTM	92.00%	42.71%	0.9789	0.003	80	0.001	2	Logcosh	Adam	0.9	0.999
Model2_v1	Encoder-Decoder	91.95%	43.62%	0.2	0.822	25	0.1	2	mse	Adam	0.9	0.999

Figure 6: *Evaluation metrics and skills for Model 1 and Model 2 for precipitation nowcasting*

From the evaluation metrics above, we conclude that accuracy is not always the only indicative metric to how well a model performs, but subject to the model and the use case additional metrics should be considered. The blurriness and lack of perceptual accuracy of the predicted images is justified from the low SSIM values in all models developed. While we tested all models at various hyperparameters, the above chosen parameters were giving the best results both qualitative and quantitative. Accuracy in all tests did not improve beyond 92%. Since we have not used the full dataset due to reasons mentioned in Section 5, we believe our models are not trained enough to generalize well and produce better results. Despite all these factors, ConvLSTM-based models produce better results than approaches based on fully connected neural networks, but the predictions exhibit blurriness, especially for many time-step predictions. Among all our model implementations, Model1-v2 produced the best perceived long time-step predictions able to produce eight time-step predictions which translates to 48 minutes of nowcasting.

8 Future Work

Precipitation nowcasting using data-driven or deep learning techniques is currently an active research topic. In this project we focused mainly in exploring and developing several models based on ConvLSTM architectures. For future work there are several directions e.g use transfer learning and change the last few layers to test our model for predicting values in the synthetic Moving-MNIST dataset [5], increase the number of training data and incorporation of additional input dimensions such as 3D radar images. Motion feature learning strategies could be investigated to increase the accuracy of the motion trajectory.

In addition, we have initiated work using a U-NET architecture (model-3 in [4]), as inspired by [1]. U-Nets can address better the localisation of precipitation with pixel-level accuracy, addressing the blurriness of ConvLSTMs. Also U-Nets are typically scale invariant and have the benefit to work with less input data. A novel architecture which will combine the benefits of U-Net and ConvLSTM could potentially give more accurate predictions.

9 Contributions

Georgios Sarmonikas: Worked on the methodology, algorithms and development of models1,2,3. Responsible for preparing the group report.

Anika Jain: Tested existing architectures on moving-MNIST dataset. Development of Model 0, and finetuning of Models 1 and 2. Responsible for preparing the group report.

Yash Gaur: Worked on initial data processing, helped in development of model 1 and 2 and testing different evaluation metrics. Responsible for preparing the group poster.

References

[1] Shreya Agrawal et al. “Machine Learning for Precipitation Nowcasting from Radar Images”. In: (2019). arXiv: `1912.12132 [cs.CV]`.

[2] G. Ayzel et al. "All convolutional neural networks for radar-based precipitation nowcasting". In: *Procedia Computer Science* 150 (2019). Proceedings of the 13th International Symposium âIntelligent Systems 2018â (INTELSâ18), 22-24 October, 2018, St. Petersburg, Russia, pp. 186–192.

[3] Chen Lei. "A Deep Learning Based Methodology for Precipitation1 Nowcasting with Radar". Version V1. In: (2019). DOI: 10.7910/DVN/2GKMQJ. URL: https://doi.org/10.7910/DVN/2GKMQJ.

[4] *Precipitation Nowcasting using DeepLearning architectures*. https://github.com/gsarm78/cs230Project.

[5] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. "Unsupervised Learning of Video Representations using LSTMs". In: *CoRR* abs/1502.04681 (2015). arXiv: 1502.04681. URL: http://arxiv.org/abs/1502.04681.

[6] Ruben Villegas et al. "Decomposing Motion and Content for Natural Video Sequence Prediction". In: *CoRR* abs/1706.08033 (2017). arXiv: 1706.08033. URL: http://arxiv.org/abs/1706.08033.

[7] Wang-chun WOO and Wai Kin Wong. "Operational Application of Optical Flow Techniques to Radar-Based Rainfall Nowcasting". In: *Atmosphere* 2017 (Feb. 2017), p. 48. DOI: 10.3390/atmos8030048.

[8] SHI Xingjian et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting". In: *Advances in neural information processing systems*. 2015, pp. 802–810.