Report of master 2 research intership in computer science

Fused deep learning for hurricane track forecast from reanalysis data

Author :
Mo Yang

Internship supervisor :
Claire Monteleoni
Guillaume Charpiat

Host organisation :
Laboratoire de l'Accélérateur Linéaire
Inria

Secretariat - tel : 01 69 15 81 58
e-mail : alexandre.verrecchia@u-psud.fr

Contents

Abstract

Hurricanes is one of the most severe natural diaster that cost tremendous damage and death in each year. The forecast of incoming hurricanes trajectory can be very important for protecting people who live in shoreline areas and reducing their economic loss. In this report, we propose deep learning method for hurricanes trajectory forecasting. We introduce a fusion architecture. The proposed fusion architecture of neural network allows for processing different sources of data in parallel, including past hurricanes movements and reanalysis atmospheric wind fields and geopotential fields images. This fused network is trained to estimate the longitude and latitude 24h-forecast of hurricanes and depressions from a large database from both hemispheres (more than 3000 storms since 1979). It is demostrated that the fused network which integrate all sources of data have significantly lower error distance than network that use any single source of data. (add compare with state-of-art)

Mots clefs

Hurricanes, Trajectory forecasting, Deep learning, Fusion architecture, Neural networks.

Chapter 1

Introduction

1.1 Introduction To Laboratory

1.2 Introduction To Groups

1.3 Structure

The intership focuses on forecast of hurricane trajectory using deep learning method.

Chapter 2

Description of Internship Subject

2.1 Hurricane Trajectory Forecasting

Cyclones, hurricanes or typhoons are words desigining for the same phenomena: a rare and complex event characterized by strong winds surronding a low pressure area. Hurricanes is one of the most severe natural diaster that cost tremendous damage and death in each year. An average hurrican can release as much energy in a day as explosion of half a million small atomic bombs. In 20th century, over 45000 people were killed by hurricanes. In 2005, hurricane Katrina has costed as much as 125 billions USD worth of damage. That is why the forecast of hurricanes trajectory is so important, if we know when and where the hurricanes is going to make landfall, people live in that area will be alarmed and get some time to protect their goods and evacuate. However this is difficult, hurricane movements can be very unpredictable. Their evolution depends on many factors at different scales and altitudes, which leads to difficulties in their modelling. Also, since the 1990s, storms have been more numerous, leading to both more representative and more consistent error statistics.

State-of-art

Today, the forecasts (track and intensity) are provided by a numerous number of guidance models [1]. The original best model was statistical model based on historical relationships between storm behavior and various other parameters and it was the major forecasting model until 1980's [4]. Today it is mostly used for testing and comparing new models. Dynamical models solve the physical equations governing motions in the atmosphere. If they can have quite precise results, they are computationally demanding. If the computation last for days, it wouldn't make sense to make prediction since the prediction will already be out of date. The statistical model, in contrast, does not demand high computational cost, but are less accurate than dynamic models. Current national forecasts are typically driven by consensus methods able to combine different dynamical models.

Relevent Studies

The hurricane statistical forecasting models perform poorly with respect to dynamical models, even if the database made of past hurricane is constantly growing. The machine learning methods, able for example to capture non-linearity and complex relations, have only been scarcely tested. However, they have recently shown their efficacy in a various number of forecasting tasks. Particularly, convolutional neural networks (CNNs) have raised attention as they are suited for large imaging data.A convolutional LSTM model has been used for precipitation forecast in a promising study [10]. Another recent study predicts the evolution of sea surface temperature maps by combining CNNs with physical knowledge [2]. The CNNs have also been used for the detection of extreme weather like hurricanes from meteorological variables patches. [8]. To our knowledge, only two preliminary studies have tackled the hurricane forecast tracking using machine learning: the first one uses random forests on local reanalysis histograms [6], however the mean error of 6h-forecasts seem to indicate poor results (more than 60km). The second uses a sparse recurrent neural network from trajectory data [7], but it was tested on only 4 hurricanes and seem aslo to give large distance errors (mean 6h-forecast error is 72km).

2.2 Internship Subject

In this work, we propose a neural network architecture taking into account past trajectory data and reanalysis atmospheric wind fields and geopotential fields images. This fused network estimates the longitude and laritude 24h-forecast of hurricanes and depressions from both hemispheres and different basins (more than 3000 storms since 1979)....

CHAPTER 3

Theoratical insights

3.1 Domaine Knowledge

some domaine knowledge about hurricanes and forecasts...

3.2 Tensor Data

somewords on its temporal and spatial dependencies...

3.3 Convolutional Neural Network

Convolutional neural network(CNN) has been widely adopted as a powerful tool for pattern recognition tasks with images or image-like data in various previous studies.(cite) In the architecture of CNN, the basic components are convolutional layers, maxpooling layers, activation layers and fully connected layers. The convolutional layers perform convolutional operations between kernels and input images(for intermidiate feature map). A convolutional layer contains K kernels(filters) with the size (i,j,c), where i, j represent the width and height of the kernel and c represent the number of input channels, after the convlutional operation K feature maps serve as ouput and will be inputs of the next layers, deal with spatial-correlations...

With the evolution of CNN, researchers keep looking for better CNN architecture which has better precision and more computential efficiency. Since Alex net(2012), more and more attempts...

A typical CNN architecture known as VGG applies alternatively convolutional layers (use only 3x3 size kernel) and maxpooling layers (use only 2x2 size kernel) through the whole network, and put several fully connected layers at the end of the network to generate output. VGG often serves as a guideline to design architecture of the nerual network. The advantages of adopting VGG is that the most important hyper-parameters for designing CNN becomes the depth of the network without worrying about other hyper-parameters like kernel size. In this paper we will use VGG as a guideline to design our CNN for the task of storm track forecasting.

a throughout introduction to CNN and pattern recognition with CNN

3.4 The Fused Deep Neural Network

Fusion networks....

Chapter 4

Method

4.1 Data Description

The raw storm track data used in this study is composed of more than 3000 extra-tropical and tropical storm tracks since 1979 extracted from the NOAA database IBTrACS[5], see Figure 2. The tracks are defined by the 6-hourly center locations (latitude and longitude). They come from both hemispheres and the number of records per storm varies from 2 to 120 time steps. In total, the database counts more than 90 000 time steps.

The trajectory of a storm depends on large scale atmospheric flows. Thus, we extracted the wind fields of the neighborhood of the storm at every time step t from the ERA-interim reanalysis database [3]. Specifically, we extracted the u-wind and v-wind fields on a $25x25$ degree grid centered on the current storm location, at 3 atmospheric pressure levels (700 hPa, 500 hPa and 225 hPa). Because we wanted to capture the dynamics, we also extracted the wind fields measured at $t - 6h$ at the same locations.

The choice of the 3 pressure levels was driven by statistical forecast models [4]. The reason why we focused on the wind parameter is that we applied a sparse feature selection technique (Automatic Relevance Determination, based on linear regression) over all available reanalysis fields, which highlighted the usefullness of wind.

4.2 CNN Configuration for Wind Fields and Geopotential Fields

Convolutional neural networks (CNN) are suited for non-linear learning with image-like data. They have already shown their efficiency in the climate informatics field [10, 2, 8]. The centered wind fields at different pressure levels at t and $t - 6h$ can be seen as 12 images of size 25x25. We used as a guideline a typical CNN architecture alternating convolutional layers and maxpooling layers and added several fully connected layers at the end of the network [9]. To measure the improvements brought by increasing the CNN depth, we have designed 4 CNNs with the same number of neurons and varying depths. We observed very unobvious improvement on the result, so we chose the most shallow CNN with only 1 convolutional layer for computational reasons.

4.3 Neural Network for Past Tracks

Another important source of information is the previous displacements (latitude and longitude for $t - 12h$ and $t - 6h$). We designed a small neural network (two small fully connected layers) able to learn the future track from this past track.

4.4 Fusion Architecture for All Sources of Data

Because of the different nature of the wind field image and of the past track data, it is not straightforward to mix them as a common input to a bigger network. Instead, we first train separately the wind field CNN and the small past track neural network (NN) previously mentioned, and then we fuse their two last layers, and re-train them together (see Figure **??**).

Table 4.1: **ConvNet configurations**(shown in columns). The depth of the configurations increases from left(A) to the right(D), as more layers are added. The convolutional layer parametres are denoted as "conv(kernel size)-(number of output channels)". The ReLU activation layer and Batch normalization layers are not shown in figure

ConvNet Configurations			
A	B	C	D
7 layers	8 layers	9 layers	10 layers
input (25*25, 12 channels image)			
conv4-32 maxpool	conv4-32 conv3-32 maxpool	conv4-64 conv3-64 maxpool conv3-256	conv4-64 conv3-64 maxpool conv3-128 conv3-256 maxpool
FC-576			
FC-128			
FC-64			
FC-8			
FC-8			
FC-2			

Table 4.2: Number of parameters (in millions)

Network	A	B	C	D
Number of parameters	2.27	2.33	2.75	2.67

CHAPTER 5

Experiments and Result Analysis

5.1 Data and Evaluation Settings

5.2 Training Details and Results

The storms were randomly separated in 3 sets as follows: train (60%) / valid (20%) / test (20%). Then, within each set, all time instants were treated independently. As a loss function (quantity to optimise), we used the mean square error (MSE) in kilometers between the forecast and the true storm location at $t + 6h$. We added an L2 penalty on the weights of the model (*coef.* = 0.01). The training was performed by the Adam optimizer.

Our implementation uses PyTorch 4.0. The training and testing took less than 1 hour on 4 TitanX GPUs with data parallelism [**?**].

Figure 5.1 shows the 6h-forecast results on the test set in absolute distance error. We define the baseline prediction as equal to the last displacement (from $t - 6h$ to t). We can see the improvement of fusing networks (mean error $\bar{e} = 32.9km$) with respect to the wind field CNN alone ($\bar{e} = 40.7km$) or the track neural network alone ($\bar{e} = 35km$). We have plotted in Figure 5.2 an example of 6h-forecasts on one storm track for the baseline and for our prediction (fusion networks). Our forecast predicts well, even in the case of change of direction or speed.

If these results are promising, some more long-term predictions are needed for a practical use. Moreover, current forecast models do not provide less than 24h-forecasts, which prevents us from comparing the results. With respect to the existing machine learning studies predicting 6h-forecasts [6, 7], we tend to perform better (error larger than 60km for both studies) and on a larger/more diverse dataset. Moreover, if we only look at hurricane time steps (without depressions), our mean prediction error drops to 25.8km. Depressions seem to be more difficult to predict: an explanation can be that they are smaller and more subject to local perturbations.

5.3 Result Analysis

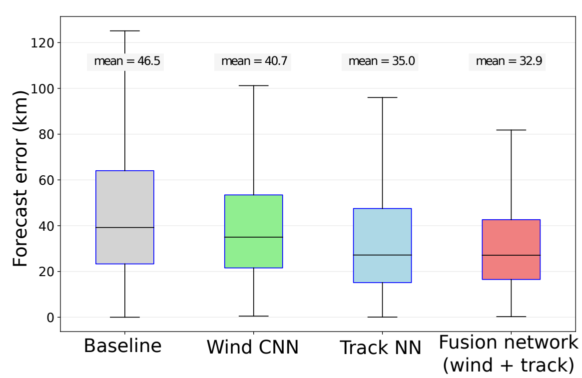

Figure 5.1: 6h-forecast results on the test set (storms coming from all oceanic basins), in distance between predicted and real location. Baseline = previous displacement (going straight).

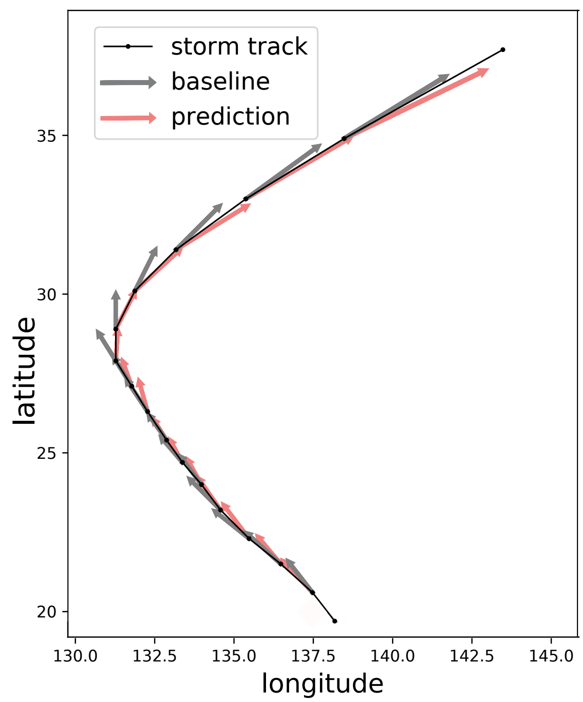

Figure 5.2: Example of 6h-forecasts on one storm track. The baseline prediction is equal to the last 6h-displacement (going straight).

CHAPTER 6

Conclusion and Discussion

We showed a promising deep learning framework for storm track forecasting. We demonstrated the benefit of coupling two types of data (past tracks and wind fields) in an efficient fusion model. Our results on a large database (90 000 time steps from 3000 storms) from different oceanic basins are showing 6h predictions with less than 33km error. Moreover, the error on only hurricane data points (without depressions) drops to 25.8km. We think that the use of such deep learning methods can help the current forecast modellers by providing a complementary prediction that could be integrated in some consensus methods.

Remerciements

Saraceni tamen nec amici nobis umquam nec hostes optandi, ultro citroque discursantes quicquid inveniri poterat momento temporis parvi vastabant milvorum rapacium similes, qui si praedam dispexerint celsius, volatu rapiunt celeri, aut nisi impetraverint, non inmorantur.

Bibliography

[1] Nhc track and intensity models. https://www.nhc.noaa.gov/modelsummary.shtml. Accessed: 2018-07-04.

[2] Manu de Bezenac, Arthur Pajot, and Patrick Gallinari. Deep learning for physical processes: Incorporating prior scientific knowledge. *arXiv preprint arXiv:1711.07970*, 2017.

[3] Dick P Dee, S M Uppala, AJ Simmons, Paul Berrisford, P Poli, S Kobayashi, U Andrae, MA Balmaseda, G Balsamo, d P Bauer, et al. The era-interim reanalysis: Configuration and performance of the data assimilation system. *Quarterly Journal of the royal meteorological society*, 137(656):553–597, 2011.

[4] Mark DeMaria, Michelle Mainelli, Lynn K Shay, John A Knaff, and John Kaplan. Further improvements to the statistical hurricane intensity prediction scheme (ships). *Weather and Forecasting*, 20(4):531–543, 2005.

[5] Kenneth R Knapp, Michael C Kruk, David H Levinson, Howard J Diamond, and Charles J Neumann. The international best track archive for climate stewardship (ibtracs) unifying tropical cyclone data. *Bulletin of the American Meteorological Society*, 91(3):363–376, 2010.

[6] Sterenn Marie Liberge, Sileye Ba, Philippe Lenca, and Ronan Fablet. Prévision de trajectoires de cyclones à l'aide de forêts aléatoires avec arbres de régression. In *Conférence internationale francophone sur l'extraction et la gestion des connaissances*, pages 623–634. Hermann, 2011.

[7] Mina Moradi Kordmahalleh, Mohammad Gorji Sefidmazgi, and Abdollah Homaifar. A sparse recurrent neural network for trajectory prediction of atlantic hurricanes. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 957–964. ACM, 2016.

[8] Evan Racah, Christopher Beckham, Tegan Maharaj, Samira Ebrahimi Kahou, Mr Prabhat, and Chris Pal. Extremeweather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. In *Advances in Neural Information Processing Systems*, pages 3402–3413, 2017.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[10] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.