DESIGN AND PRELIMINARY EVALUATION OF TEAM BASED COMPETITIONS IN VIDEO FORECASTING

Florin Popescu
Fraunhofer-Gesellschaft zur Forderung und Forschung
florin.popescu@fokus.fraunhofer.de

Isabelle Guyon
Chalearn
guyon@chalearn.org

Lisheng Sun
UPSud Paris-Saclay
lsun@lri.fr

Diviyan Kalainathan
dkalainathan@gmail.com

Sébastien Treguer
La Paillasse
streguer@gmail.com

Cecile Capponi
Aix Marseille Université, LIF-CNRS
Cecile.Capponi@lif.univ-mrs.fr

Stephane Ayache
Aix Marseille Université, LIF-CNRS
stephane.ayache@lif.univ-mrs.fr

Xavier Baró
Universitat Oberta de Catalunya
Computer Vision Center, UAB
xbaro@uoc.edu

Julio C. Jacques Junior
University of Barcelona
Computer Vision Center, UAB
julio.silveira@ub.eu

Sergio Escalera
University of Barcelona
Computer Vision Center, UAB
sergio.escalera.guerrero@gmail.com

ABSTRACT

The article describes the design of a series of competitions the aim of which is the evaluation and the further development of the state-of-the art in spatio-temporal forecasting. The means of doing so is to provide novel test data incrementally, while evaluating work of competing teams that submit algorithms in terms of performance criteria which include accuracy of predictions and time of computation. Initial results are presented hereing, whereas final results of the ongoing challenge will be presented at ICLR.

1 BACKGROUND OF THE SEE4C PROJECT

We have undertaken a novel competition design to address current challenges in spatio-temporal forecasting, within the realm of an ongoing projected named SEE.4C (see-foresee) sponsored by the European commission (H2020). Spatio-temporal data (such as video) is rich and complex and it still remains an open question what type of learning algorithms (deep learning, kernel based approaches, traditional economics and engineering based approaches) provide superior performance. The challenges we address have both a long history as forecasting competitions (e.g. "M" series of challenges, the WxChallenge), in which the impetus is on repeated out-of-sample, causal regression and as ML competitions (AutoML challenge) in which the evaluation of submissions by challenge participants is not based on predictions or labels but actually on code submitted, which must run efficiently: in the case of forecasting real-time operation is necessary. The current work describes the process of evaluating algorithms for difficult learning tasks, allowing for collaborative effort and rapid progress, the goal of which is to establish a procedure and platform for evaluating algorithms that predict unseen samples adaptively, future work involves the expansion of the base of the code-base to allow for more complex state-of-the art methods applicable not only to forecasting of video frames but also to other related spatio-temporal forecasting taks involving big data, such as energy flows.

2 HACKATHON SCHEDULE

During a hackathon, time is a rare resource under strong constraints and has to be managed with discipline and flexibility, with the objective to offer each participant, speaker and organizer the opportunity to give the best of what they have to share, and create the proper conditions for good interactions, synergies and contributions. This hackathon was especially time constrained because it lasted only one day, combining inspiring talks with a short 3 hours hands-on phase, during which participants could implement and submit their solutions with code.

To help participants focus directly on the task, a “starting kit” ipython notebook is released before the hackathon day. The idea of this “starting kit” is to guide participants towards a valid submission, we provide in it:

- sample data that participants can use to train / adapt their model
- sample codes that includes: 1) functions to read and visualize data, 2) scoring program to compute the performance of model during development, 3) a ‘model’ module with clear structure to help participants organize their own solution.
- other functions called by the server to process the submission.

As such, participants only need to modify the ‘model’ design inside the “starting kit”. This greatly improves the productivity of the hackathon.

In order to manage time efficiently and make the most of the hackathon day, participants were asked few days before to make a first submission, meaning they had to read the description of the challenge, get familiar with the data, the objective, the rules, and the architecture/code of the “starting kit”, and get their system all set with the required libraries to start implementing new ideas.

- Inspiring presentations of speakers from different fields.
- Ice breaking over breakfast and coffee breaks and teams formation at lunch break.
- Team members are carefully selected to create teams composed with people from various background and equipped with complementary skills and mindset.
- 3 hours hacking time in teams.
- A coach is assigned to each team to animate and motivate the team work, and help to fix any technical problem during the submission and result processing.
- Limited number of submissions per team to avoid overloading the server.

3 HACKATHON DATA

In order to build the dataset used in the hackathon, we first obtained 149 talking-to-camera videos from different sources (quality: 720p HD, 25 FPS), from which about 48.000 non-overlapped video clips of 5 seconds each were generated. Videos present single person facing the camera (teleconference scenario), with different inner face (expressions) and head movements. They also present a great variety in terms of illumination conditions, partial occlusion, ethnics and background clutter. In a second stage, videos were post-processed in order to focus on the peoples face. In a nutshell, a face detector was used to detect the peoples face at each frame in order to select and crop the region of interest as well as to remove video clips that contains cuts and camera movements. Moreover, visual checking were performed to guarantee that the face remains the 100% of the time in the video (although may present partial occlusions). Finally, videos are converted to grayscale and 32x32 pixels resolution. Figure 1 illustrates some images presented in our dataset.

Figure 1: Image samples of the dataset.

Table 1: Leaderboard of the Hackathon.

Rank	Team Name	RMSE	Duration
1	Gamma	0.0559	141.58s
2	Delta	0.0565	61.27s
3	Omega	0.0567	0.90s
4	Beta	0.0567	58.64s
5	Alpha	0.0689	59.09s
6	Epsilon	-	-

4 Hackathon platform

We built a new[1] instance of CodaLab[2], specifically developed for the purpose we need. CodaLab is a powerful (open source) framework for running competitions that involve code submission (with fast feedback). Participants had first to register to the platform and then to the proposed competition/hackathon in order to participate. We also introduced a new feature in our CodaLab instance, which allow us to create and manage teams. Thus, participants are now able to form multiple teams, composed by a team leader and members (without overlap) and make individual submissions that will count for their respective teams.

5 Hackathon results

Table 1 shows the leaderboard of the hackathon. Note: a) the simple baseline method (persistence, RMSE error of 0.0567), is not trivial to outperform, and due to the complexities of the underlying data and the out-of-sample validation method, can result in lower-ranking submissions which overfit. Also note b) the wide variation in running time (the total for 600 prediction steps is shown). More complex methods can be considerably slower, although they can also thereby improve accuracy. While methods outperforming persistence run in slower than real time (24 seconds) these are prototype methods which can benefit from further tuning and runtime optimization. The winning method, for example, constructed correlation matrices for each pixel (independently) and rescaled the resulting linear kernels, applying them uniformly over frames: correlation matrix construction can be optimized for a given data regime. Future and continuing work, the results of which are to be presented in further detail concurrently with ICLR (satellite workshop) will allow comparison to a wider array of state-of-the art methods, including recent deep-learning approaches Ranzato et al. (2014); Mathieu et al. (2015); Lotter et al. (2016) using more computational power in a feedback stage (in which submissions are scored sequentially on unseen data and the performance is fed back to participating teams)and a final validation stage (on further unseen data). In the longer term, we envisage the expansion of the data domains to larger and more complex content: wider content of video and prediction of grid energy flows using geo-spatial datasets as predictors.

Acknowledgments

This work was supported by the European Commission under framework H2020, ICT-16-2015, grant number 688356, “See.4C”. This work has been partially supported by the Spanish project TIN2016-74946-P (MINECO/FEDER, UE) and CERCA Programme / Generalitat de Catalunya. We kindly thank the participants of the hackathon and Igor Carron for support in organizing.

[1] `https://demo.see4c.eu`
[2] `http://codalab.org`

REFERENCES

William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *CoRR*, abs/1605.08104, 2016.

Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2015.

Marc'Aurelio Ranzato, Arthur Szlam, Joan Bruna, Michaël Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *CoRR*, abs/1412.6604, 2014.