Solar flare prediction with temporal convolutional networks

Dewald D. Krynauw[1,2][0000−0001−6077−5906], Marelie H. Davel[1,2][0000−0003−3103−5858], and Stefan Lotz[1,3][0000−0002−1037−348X]

[1] Multilingual Speech Technologies (MuST), North-West University, South Africa;
[2] Centre for Artificial Intelligence Research (CAIR), South Africa.
[3] South African National Space Agency (SANSA), Space Science directorate, Hermanus, South Africa.
dewaldkrynauw123@gmail.com, marelie.davel@nwu.ac.za, slotz@sansa.org.za

Sequences are typically modelled with recurrent architectures, but growing research is finding convolutional architectures to also work well for sequence modelling [1]. We explore the performance of Temporal Convolutional Networks (TCNs) when applied to an important sequence modelling task: solar flare prediction. We take this approach, as our future goal is to apply techniques developed for probing and interpreting general convolutional neural networks (CNNs) to solar flare prediction.

Severe space weather events originate near sunspots and are caused by solar flares (broadband bursts of electromagnetic energy) and the accompanying coronal mass ejections (plumes of magnetised gas projected outwards into space). These space weather phenomena can damage spacecraft, communications and electric power systems [2]. We follow Liu et al. [7] in trying to predict future flares from past observations of the sun, and specifically from various images and magnetograms of identified active regions (ARs), and parameters derived from these. This is framed as a binary classification task that asks: will an AR produce a Υ-class flare within the next 24 hours? In the current work we focus on $\geq M5.0$ class flares. These are potentially more harmful, but also easier to predict than lower class flares.

The dataset used in this work is open source and compiled by Liu et al. [7]. It consists of the Space Weather HMI Active Region Patches (SHARP) data produced by the Helioseismic and Magnetic Imager (HMI) on Solar Dynamics Observatory (SDO) and an additional 15 parameters from Jonas et al. [5]; plus another 9 from Nishizuka et al. [8] related to the flaring history. Due to the unbalanced nature of the task, the True Skill Statistic (TSS) metric is most commonly used to determine the effectiveness of a model, as suggested by Bloomfield et al. [3]. As of this writing, Liu et al. has produced the best result on this dataset: a test TSS of 0.858 for a vanilla Long Short-Term Memory (LSTM), and 0.877 for an LSTM extended with additional attention layers and fully connected layers.

We replicate the LSTM only as a sanity check of the results from Liu et al., as our focus is on developing and optimising a TCN [1] to determine whether similar performance is achievable. A vanilla LSTM was trained and evaluated with different numbers of layers (1, 5, 10), and varying batch sizes and learning rates. After optimizing on the training and validation set, TSS was measured on the test set, averaging over 3 seeds. After basic optimisation, a test TSS of 0.850 could be achieved, which is fairly similar to that achieved by Liu et al. No dropout or weight decay was used.

A TCN is in essence a 1D Fully Convolutional Network (FCN) [9] with dilated causal convolutions. This network architecture is not new and is based on the time delay neural network published 30 years ago by Waibel et al. [10], with the addition of zero-padding to ensure all layers are of equal size. There are essentially two ways to increase the receptive field of the TCN: increasing the number of levels (that is, the number of residual blocks [4]), or the kernel size.

We implement the TCN using publicly available source code[4], implemented using Pytorch. All the models are trained with the weighted cross-entropy loss function to combat the unbalanced data and Adam [6] is used as the optimiser. After a set of initial probing runs to determine well-performing network hyper-parameters, a more in-depth optimisation of the TCN was conducted, by searching over a wider range of learning rates. The results are logged and graphed using the "Weights & Biases" API and a full report of the results is available [5].

A grid search over different levels, kernel sizes and hidden dimensions (channels) was performed and the best two networks on the validation set were selected to further refine learning rates. At first, a hidden dimension of 128 was selected, but showed no significant increase and was reduced to the number of input features (20). Initially, the results showed that the more levels the TCN has the better it performs, but after optimising on different learning rates; the same TSS can be achieved on the shallower networks with smaller learning rates that train longer.

The TCN was able to reach average validation TSS of 0.838 and average test TSS of 0.848 with 7 levels and a kernel size of 2 (with dropout and weight decay). A TCN with many levels comes at a large computational cost relative to the LSTM. The best-performing TCN took 1 hour to train, compared to 10 minutes for the LSTM, using the same hardware. Optimising the TCN further (using smaller learning rates, training longer and no regularisation), the shallow networks of 1 level were able to obtain an average validation TSS of 0.711 and test TSS of 0.886, with some individual networks reaching up to 0.910 test TSS. These level 1 TCNs average around 17 minutes of training time, which is more comparable (though still slower) than the vanilla LSTM.

We applied TCNs to solar flare prediction – according to our knowledge an architecture not yet used for this task. Results indicate that TCNs perform on par with the LSTMs used by Liu et al., which are currently considered state of the art. This is important as we are specifically interested in developing models that can be probed and interpreted, and LSTMs are very difficult to analyse. Our work confirms the statement by Bai et al. [1], predicting that TCNs should have similar performance as vanilla LSTMs.

[4] https://github.com/locuslab/TCN

[5] Full report

References

1. Bai, S., Kolter, J.Z., Koltun, V.: An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling (3 2018), http://arxiv.org/abs/1803.01271
2. Baker, D.N., Daly, E., Daglis, I., Kappenman, J.G., Panasyuk, M.: Effects of Space Weather on Technology Infrastructure. Space Weather **2**(2), n/a–n/a (2 2004). https://doi.org/10.1029/2003sw000044, http://doi.wiley.com/10.1029/2003SW000044
3. Bloomfield, D.S., Higgins, P.A., McAteer, R.T., Gallagher, P.T.: Toward reliable benchmarking of solar flare forecasting methods. Astrophysical Journal Letters **747**(2), 41 (2012). https://doi.org/10.1088/2041-8205/747/2/L41, http://www.swpc.noaa.gov/ftpdir/warehouse/
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2016). https://doi.org/10.1109/CVPR.2016.90
5. Jonas, E., Bobra, M., Shankar, V., Todd Hoeksema, J., Recht, B.: Flare Prediction Using Photospheric and Coronal Image Data. Solar Physics **293**(3), 48 (2018). https://doi.org/10.1007/s11207-018-1258-9, https://doi.org/10.1007/s11207-018-1258-9
6. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization (12 2014), http://arxiv.org/abs/1412.6980
7. Liu, H., Liu, C., Wang, J.T.L., Wang, H.: Predicting Solar Flares Using a Long Short-term Memory Network. The Astrophysical Journal **877**(2), 121 (6 2019). https://doi.org/10.3847/1538-4357/ab1b3c, https://iopscience.iop.org/article/10.3847/1538-4357/ab1b3c
8. Nishizuka, N., Sugiura, K., Kubo, Y., Den, M., Watari, S.I., Ishii, M.: Solar Flare Prediction Using Machine Learning with Multiwavelength Observations. Proceedings of the International Astronomical Union **13**(S335), 310–313 (2017). https://doi.org/10.1017/s1743921317007293, https://doi.org/10.1017/S1743921317007293
9. Shelhamer, E., Long, J., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(4), 640–651 (4 2017). https://doi.org/10.1109/TPAMI.2016.2572683, http://ieeexplore.ieee.org/document/7478072/
10. Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K.J.: Phoneme Recognition Using Time-Delay Neural Networks. IEEE Transactions on Acoustics, Speech, and Signal Processing (1989). https://doi.org/10.1109/29.21701