Southern Horizons in Time-Domain Astronomy
Proceedings IAU Symposium No. 339, 2019
E. Griffin, ed.

doi:10.1017/S1743921318002491

Deep-Learning the Time Domain

A. Mahabal[1,2], K. Sheth[3], F. Gieseke[4], A. Drake[2], G. Djorgovski[1,2] and M. J. Graham[1,2]

[1]Department of Astronomy, California Institute of Technology, Pasadena, CA, USA
email: aam@astro.caltech.edu

[2]Center for Data Driven Discovery, California Institute of Technology, Pasadena, CA, USA

[3]Indian Institute of Technology Gandhinagar, India,

[4]Department of Computer Science, University of Copenhagen, Denmark

Abstract. "Deep learning" is finding more and more applications everywhere, and astronomy is not an exception. This talk described the application of convolutional neural networks to time-domain astronomy, specifically to light-curves of sources. The work that is discussed is based on a published paper to which reference can be made for more detail. The talk finished with a note cautioning new practitioners about the pitfalls lurking in out-of-the-box use of deep-learning techniques.

Keywords. Surveys, methods: data analysis, Techniques: image processing

1. Introduction

Astronomy has been a "Big Data" science for a long time. With the advent of multibillion-row sky surveys with tens to hundreds of pointings over large areas of sky, real-time and archival studies are receiving a big boost. The bigger surveys and corresponding data transfers are served by better CCDs, faster electronics, and higher internet speeds. Data analyses are helped by the availability of GPUs, larger clusters and better libraries.

This talk highlighted an area of time-domain astronomy that is crucial for source classification, whether from images or from time series. It described the problems, and showed how convolutional neural networks are useful for the classification of light-curves. The work is based on Mahabal *et al.* (2017), where the reader will find more details than could be included here.

2. Deep Learning

Convolutional neural networks (CNNs) (LeCun *et al.* 2015) are a special type of *artificial neural networks*, ANN (Hastie *et al.* 2009, Murphy 2012). A CNN consists of different types of layers: an input layer, an output layer, and a few between them. The last layers before the output layer are often standard fully connected layers, called *dense* layers. The layers before these dense layers are of three kinds (a) *Convolutional layers* consist of a small set of filters (e.g., 3×3, 5×5, . . .), called *kernels*; convolutional layers convolve every input image with each of the kernels. Several such kernels are used typically as filters in a given convolutional layer to match desired shapes in the input images. That gives rise, for each kernel, to a new representation of the input image. Those representations are called *feature maps*. The convolutional layers are used with rectifiers to introduce non-linearity. (b) *Pooling layers* decrease the number of parameters of the network by aggregating values of spatially-adjacent pixels. One prominent type of pooling layers is

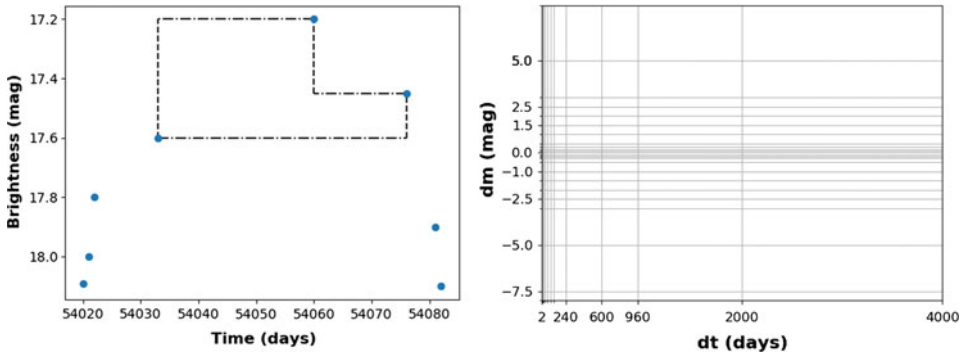

Figure 1. *Left:* Schematic light-curve, without error-bars, to demonstrate *dm* (dashed lines) and *dt* (dotted lines) values. Each pair of points in the light-curve leads to one *dmdt* pair. Four pairs are shown. *Right:* Corresponding *dmdt* grid, populated by the pairs shown on the left. Each unequal-area rectangle/bin translates to one of the equal-area pixels in the 23×24-pixel images in our experiments (examples are in Fig. 2).

max-pooling, which replaces patches of an input feature map by the maximum value within each patch. While that reduces the sizes of the feature maps, it also makes the network more robust to small changes in the input data. (c) *Dropout layers* omit hidden units randomly by setting their values to zero. The network therefore cannot rely on them fully, and this helps prevent overfitting. The depth afforded by the multiple layers, and the extensive mappings provided by them, is what gave rise to the name *deep learning*.

Although dropout layers, maxpooling, etc., guard against overlearning, the large number of adjustable weights in the multiple layers offers a control factor. It is easy, for instance, to have a deep network and have individual examples learned by a subset of the weights. Cross-validation is thus essential during network testing. In our project we started with a three-layer network, and were able to tune it to a single-layer one.

Deep learning has been applied in astronomy to many areas: galaxy classification (Hoyle 2016), supernova classification (Cabrera-Vives *et al.* 2016), light-curve classification (Mahabal *et al.* 2017, Charnock & Moss 2017), identifying bars in galaxies (Abraham *et al.* 2018), separating Near Earth Asteroids from artifacts in images (Morii *et al.* 2016), Gravitational Wave transient classification (Mukund *et al.* 2017), and even classifying noise characteristics (Zevin *et al.* 2017, Abbott 2017, George *et al.* 2017). The main distinction of our work is to convert light-curves to images and use existing CNN machinery with the images for classification.

3. Light-Curve Classification

We create image representations of light-curve data in order to use them with CNNs (LeCun *et al.* 2015) that work really well with images. In early experiments we have used data from the Catalina Realtime Transient Survey (CRTS; Drake *et al.* 2009, Djorgovski *et al.* 2011, Mahabal *et al.* 2011, Mahabal *et al.* 2012, Djorgovski *et al.* 2016) and the Palomar Transient Factory (PTF; Bellm, page 160) surveys. In particular we have used a set of periodic variables (seven classes) from CRTS (Drake *et al.* 2014). In the representation we create, the image pixels are bins with delta-time (dt) as one dimension, and delta-magnitude (dm) as another (Mahabal *et al.* 2011, 2017). The (normalised) density of points in each bin is the intensity in the corresponding image pixel (see Fig. 1). The *dt* and *dm* bins are semi-logarithmic in order to capture the cadence of the survey as well as possible time-scales for variability (e.g., the *dt* bin boundaries we have used are [1/145, 2/145, 3/145, 4/145, 1/25, 2/25, 3/25, 1.5, 2.5, 3.5, 4.5, 5.5, 7, 10, 20, 30, 60, 90, 120,

https://doi.org/10.1017/S1743921318002491 Published online by Cambridge University Press

Table 1. Number of objects in the seven periodic variable classes with at least 500 members. The variable types include EW (contact binaries), EA (detached binaries), three types of RR Lyrae stars, and Mira and semi-regulars lumped into LPV. RS CVn stars are rotating variables. Thus – broadly speaking – we have three classes: binaries, pulsating, and rotating. *Class* refers to the numeric labels used in Drake *et al.* (2014)

Type	EW	EA	RRab	RRc	RRd	RSCVn	LPV
Class	1	2	4	5	6	8	13
Num	30743	4683	2420	5469	502	1522	512

$240, 600, 960, 2000, 4000]$ days to accommodate the four images in the thirty-minute cadence of CRTS and the long time-baseline.

Learning based on such images has several consequences: (1) Features based on the light-curves that are typically computed for classification (Richards *et al.* 2011, Donalek *et al.* 2013, Graham *et al.* 2014) are not required in our method. This is good, since the features are subjective and often bias classifiers. The CNNs create from hundreds to millions of features based on the configuration and carry out classification within their framework. (2) Operating on short and partial light-curves become possible - a handy plus for real-time analysis for surveys. (3) It becomes possible to train the CNN on one survey, and – with small modifications – to use the trained models on other surveys. In this instance we handled PTF data with models trained on CRTS data.

There are multiple ways in which experiments can be carried out to improve performance: (1) Changing the layers of the CNN based on the size of the training sample, image sizes, unbalancedness between the classes, (2) Changing the *dmdt* bins based on classes and surveys, and (3) modifying the light-curve to *dmdt*-image mapping to highlight differences between classes.

We used 32×32 kernels, two *dense* layer of 128, and two *dropout* layers. More details are in Mahabal *et al.* (2017). We used the theano framework (http://deeplearning.net/software/theano/) with lasagne (https://lasagne.readthedocs.io/en/latest/) on NVIDIA GeForce GTX 560 GPU for our runs.

Binary classification. We trained the network with pairs of classes as well as with all seven periodic classes together (see Table. 1). When it was used in binary mode, we noticed poor performance when class 1 was involved (it contains two-thirds of all periodic objects in our sample), as it overwhelmed every other class individually. Class 13 reached the highest accuracy (89%) against class 1. Those are the Long-period variables (LPVs); the long-term structure was probably getting picked up. In general the separation of all other classes from class 13 was similarly far better than other binary comparisons. Except in a few cases, the binary *dmdt*-classifier was comparable or better than the corresponding feature-based random forests (RF) classifier.

Multi-class classification. When used in the 7-class mode, the *dmdt*-images produced an average accuracy of 83%. Without fine tuning, or accounting for the lack of balance in the data, that result is remarkable. The performance is comparable to a feature-based RF-classifier. Class 1 still dominates to an extent but not as blatantly as in the binary cases. In order for this to be useable in real-time for light-curves containing far fewer points, binary classification may be somewhat preferable.

Varying *dmdt* binning. The input image size of 23×24 pixels is small for CNNs and the training is relatively quick. As a result one can consider finer binning in both dm and dt. On the other hand, the discriminating structure probably resides in a few smaller areas, and one could use more granular binning. While it is desirable to determine the binning for a given survey and classes under consideration, such a systematic approach will need more extensive work. Experiments with our data set suggest that finer dm bins improves performance a little, and fewer dt bins do not seem to affect the performance

https://doi.org/10.1017/S1743921318002491 Published online by Cambridge University Press

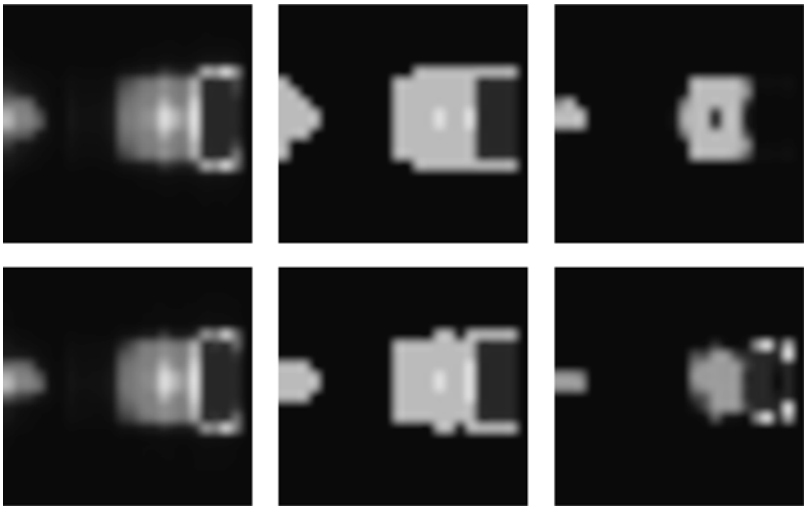

Figure 2. Example of composite *dmdt* images for RRc (top row) and RRd (bottom row) classes. In the first column all individual images are stacked together. Median *dmdt*-images are shown in the second column. The *max* class background as determined using the robust PCA method is shown in the third column. A more complex version of this Figure can be seen in Mahabal *et al.* (2017).

adversely. Since we do not take in to consideration the error bars explicitly, and the error bars are equivalent to dithering to the adjacent *dm* bin, it is not surprising that we do not see large variations.

Background Subtraction. The survey cadence provides a particular *dt* spacing for all sources irrespective of class. Despite adjusting *dt* bins to match the cadence, residual commanalities in *dmdt*-images of all sources do contain a common background (b). In addition, individual classes contain their own patterns at characteristic times and corresponding magnitude differences (c_i for the i^{th} class): *dmdt*–image $= b + c_i + s$. Estimating these backgrounds, subtracting them, and then using CNNs can reveal individual signatures of objects.

We determine backgrounds to subtract before training in a few different ways (see Fig. 2): (1) for individual class backgrounds we consider *dmdt*-images of just that class. It needs to be seen how non-uniform lengths of time-spans for individual light-curves affect individual *dmdt*-images. (2) For a pseudo-cadence background we consider all our objects together. We call this the pseudo-cadence background because all our objects in the current set are periodic variables. (3) As another possibility we ignored the class imbalance, took all training samples, and used the *max* from the background for subtraction from all training and testing samples.

Fig. 2 shows how this plays out for subclasses RRc and RRd of RR Lyrae stars. We singled those out because these are particulary tough to separate. The light-curves look quite similar, and the number of RRds in a sample is far smaller than the number of RRcs. Visually, one can make a case for being able to separate them. The robust PCA based backgrounds look rather similar; so do the median *dmdt*s. The max-values do look different and we will want to see how those can be used for separating these two difficult sub-classes.

Transfer Learning. Part of the real power of techniques like the *dmdt* is its cross-survey applicability. We used models trained with the CRTS-N *dmdt*-images and tested them on CRTS-S *dmdt*-images with the same classes (Drake *et al.* 2017), but no overlapping objects, and with PTF *dmdt*-images with a subset of the same objects as in the

https://doi.org/10.1017/S1743921318002491 Published online by Cambridge University Press

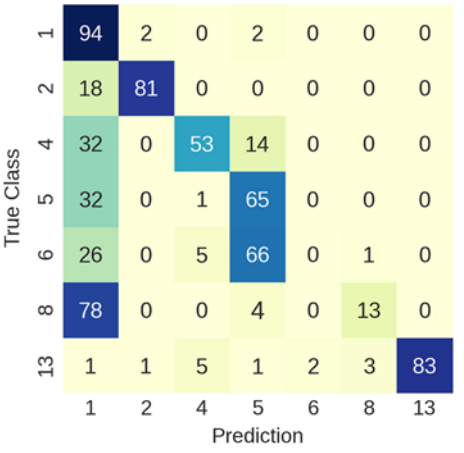

Figure 3. Confusion matrix for the CNN with fiducial *dmdt*-images. Note the high misclassification between classes 5 and 6, both RR Lyrae (RRc and RRd). Class labels are as in Table 1. A more detailed version of this Figure is given by Mahabal *et al.* (2017).

CRTS-N sample. CRTS-S uses the same asteroid-finding cadence as CRTS-N and also has an open filter. PTF used a more mixed cadence with a greater emphasis on looking for explosive events, including a repeat cadence of 1, 3, and 5 nights. We used PTF data taken with the r filter. The results, using both the shallow and deep CNNs, are not as good as with CRTS-N, but that is not unexpected. In fact, for many classes, especially for CRTS-S, they are better than one would naively expect. In the case of PTF the survey cadence is very different, in addition to diferences in aperture and wavelength range, and the results were somewhat worse. But the very fact that they are still useable, and definitely a good starting point, indicates the merit of using such a technique. With proper survey-based background subtraction the results should improve further. The implications for domain adaptation are obvious, especially with applicability to forthcoming surveys like ZTF and LSST.

4. Discussion

We have shown how to transform light-curves to simple *dmdt*-images for use with CNNs for classification of objects with performance comparable to random forests, and without having to resort to designing or extracting features. The internal features the CNN uses need to be explored further using tools like deconvolutional networks. That will make the results interpretable, and provide insights. Many explorations are possible to improve the results further e.g. background subtraction and varying the *dmdt* bins. We have further demonstrated the application of the technique to transfer learning and thereby classifying objects from a completely different survey. Figure 3 shows the confusion matrix for the periodic variables considered.

Misclassified sources. Objects belonging to some classes were more frequently misclassified than others. We investigated the light-curves for some of these sources in order to identify the source of errors. In some cases it was a genuine error (wrong label) indicating that the network was working well. In some other cases the misclassification was due to a sparse light-curve, indicating that in a handful of cases a smaller number of features may be tilting the classifications one way or another. In still other cases, the subclasses were just too close for the technique to discern them apart just from the *dmdt*-images based upon the light-curves (e.g. RR Lyrae of different types).

https://doi.org/10.1017/S1743921318002491 Published online by Cambridge University Press

5. Future Work

We continue to explore various possibilities related to varying CNN hyperparameters, improving background subtraction for more reliable classification, expanding to more classes and surveys, as well as identifying the misclassified sources. We will also experiment to make the technique more useful in real-time cases with far fewer data points. A couple of tests using error-bars to augment smaller classes did not work well; that needs to be explored further for reducing the unbalance of the different classes. There is also the possibility of using Generative Networks to create large simulated examples for different classes to understand the features that really separate different classes. We are already exploring extensions to initial work on finding transients from difference images using CNNS (Sedaghat & Mahabal 2017).

6. Cautionary note

It is tempting to use techniques used elsewhere without fully understanding the implications. For example, large networks with settled weights are often used by training just the last later. Often these are initially trained on datasets like Imagenet which has terrestrial images rather than astronomy. The results have to be investigated properly keeping in mind issues like label pollution and imbalance. In particular, the networks have to be tested thoroughly using adversarial examples. At the same time more experimentation in this field for astronomy data sources is welcome as we look to classify sources from more surveys in near real-time.

Acknowledgements

This work, and CRTS survey, was supported in part by NSF grants AST-0909182, AST-1313422, AST-1413600, and AST-1518308, and by the Ajax Foundation. KS thanks the Caltech SURF programme and IIT-Gandhinagar.

References

Abbott, B.P. *et al.* 2017, *Phys. Rev. Lett.*, 118, 221101
Abraham, S., Aniyan, A. K., Kembhavi, A. K., Philip, N. S., & Vaghmare, K. 2018, *MNRAS*, 477, 894
Cabrera-Vives, G., Reyes, I., Förster, F., Estvez, P., & Maureira, J.-C. 2016, *IJCNN*, p. 251
Charnock, T., & Moss, A. 2017, *ApJ*, 837, L28
Dieleman, S., Willett, K. W., & Dambre, J. 2015, *MNRAS*, 450, 1441
Djorgovski, S. G., Graham, M. J., Donalek, C., *et al.* 2016, *Future Gener. Comput. Syst.*, 59, 95
Djorgovski, S. G., Donalek, C., Mahabal, A., *et al.* 2011, in: A. N. Srivastava, N. V. Chawla, & A. Shehan Perera (eds.), *Proc., Conference on Intelligent Data Understanding* (CIDU 2011), p. 174
Donalek, C., Arun Kumar, A., Djorgovski, S. G., *et al.* 2013, in: *IEEE International Conference on Big Data*, 35
Drake, A. J., Djorgovski, S. G., Mahabal, A., *et al.* 2009, *ApJ*, 696, 870
Drake, A. J., Graham, M. J., Djorgovski, S. G., *et al.* 2014, *ApJS*, 213, 9
Drake, A. J., Djorgovski, S. G., Catelan, M., *et al.* 2017, *MNRAS*, 469, 3688
Dubath, P., Rimoldini, L., Süveges, M., *et al.* 2011, *MRAS*, 414, 2602
George, D., Shen, H., & Huerta, E. A. 2017, arXiv:1706.07446
Graham, M. J., Djorgovski, S. G., Drake, A. J., *et al.* 2014, *MNRAS*, 439, 703
Hastie, T., Tibshirani, R., & Friedman, J. 2009, *The Elements of Statistical Learning*, 2nd ed. (Springer, New York)
Hoyle, B. 2016, *Astron. Comput.*, 16, 34
Law, N. M., Kulkarni, S. R., Dekany, R. G., *et al.* 2009, *PASP*, 121, 1395
LeCun, Y., Bengio, Y., & Hinton, G. 2015, *Nature*, 521, 436
Mahabal, A. A., Djorgovski, S. G., Drake, A. J., *et al.* 2011, *BASI*, 39, 387

Mahabal, A. A., Donalek, C., Djorgovski, S. G., *et al.* 2012, in: E. Griffin, R. Hanisch & R. Seaman (eds.), *New Horizons in Time-Domain Astronomy*, *Proc. IAUS 285* (CUP: Cambridge, UK), p. 355

Mahabal, A., Sheth, K., Gieseke, F., *et al.* 2017, in: IEEE Symposium Series (SSCI), p. 2757

Morii, M., Ikeda, S., Tominaga, N., *et al.* 2016, *PASJ,*, 68, 104

Mukund, N., Abraham, S., Kandhasamy, S., Mitra, S., & Philip, N. S. 2017, *Phys. Rev. D*, 95, 104059

Murphy, K. P. 2012, *Machine Learning: A Probabilistic Perspective* (The MIT Press)

Netrapalli, P., Niranjan, U. N., Sanghavi, S., Anandkumar, A., & Jain, P. 2014, *CoRR*, arXiv:1410.7660

Richards, J. W., Starr, D. L., Butler, N. R., *et al.* 2011, *ApJ*, 733, 10

Sedaghat, N, Mahabal, A. 2017, *MNRAS*, 476, 5365

Zevin, M., et al. 2017, *Class. Quantum Grav.*, 34, No. 6

https://doi.org/10.1017/S1743921318002491 Published online by Cambridge University Press