It's a bird... It's a plane... It's a Type Ia supernova!

Astronomical Object Classification from Photometric Time-Series Data

Ryan Gao (rgao@stanford.edu)

Stanford University - CS230: Deep Learning (Video Presentation: https://youtu.be/CCIPoV5xoYc)

Introduction

The amount of observational astronomical data collected is increasing exponentially, as we build larger and more powerful telescopes. For example, the Large Synoptic Survey Telescope (LSST) is expected to generate up to 40 TB of data per night of observation after going live in 2022.

Figure 1: Artist's rendering of the completed LSST.

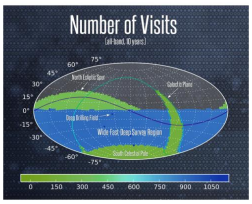

Figure 2: LSST sky coverage map.

To deal with this high data volume, there is a need for efficient and accurate automated data processing techniques. In this project, we address the problem of automated object classification based on photometric time-series data (measurements of brightness), especially for variable and transient sources (objects whose brightnesses change over time).

Dataset & Features

Our dataset comes from the PLAsTiCC (Photometric LSST Astronomical Time-Series Classification Challenge) Kaggle competition. We use the labeled training dataset containing 7848 simulated objects, and each object has the following features:

- For a given Modified Julian Date `mjd`, and color channel `passband`,
  - brightness `flux`, and its estimated error `flux_err`,
  - whether a flux change was detected relative to the background "template" image `detected`.
- Sky location: `ra`, `decl`, `gal_l`, `gal_b`
- Distance from Earth: `hostgal_photoz`, `hostgal_specz`, `distmod`
- Error associated with photometric redshift: `hostgal_photoz_err`
- "Wide-Fast-Deep" (WFD) or "Deep Drilling Fields" (DDF) survey: `ddf`
- Extinction due to galactic dust: `mwebv`
- The target class, obfuscated, with 14 possible classes: `target`

Data Processing

1. Observations are performed in only one passband at a time, so we linearly interpolate `flux` for the other 5 missing passbands at every measurement time.
2. Ignoring irregular intervals, we map data onto regular time steps t.
3. We resize the data to length $T = 352$, using linear interpolation.

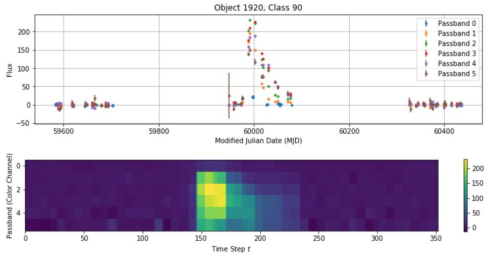

Figure 3: `flux` data transformation: a) Original data, with irregular intervals and length. b) Final input data: interpolated, pivoted by channel, and normalized to length $T = 352$.

In addition to the 10 provided, we derive 6 more metadata features from the `flux` time series: min, max, mean, stdev, median, and skew. We finally scale all 16 metadata features to zero mean and unit variance.

Models

We trained two NN architectures, 1D convolutional and recurrent LSTM, to learn the time-series data. We then concatenated its output and the 16 metadata features, before a fully connected output layer with 14 nodes and softmax activation.

- 1D CNN: Inspired by the VGG architecture, every CONV layer has filter size 3 and same padding, and is followed by BatchNorm, ReLU, and MaxPool (filter size 2, stride 2).
  INPUT $(352 \times 6) \rightarrow$ CONV16 $(176 \times 16) \rightarrow$ CONV32 $(88 \times 32) \rightarrow$ CONV32 $(44 \times 32) \rightarrow$ CONV32 $(22 \times 32) \rightarrow$ CONV32 $(11 \times 32) \rightarrow$ FLATTEN (352)
- LSTM: single-layer uni-directional LSTM with 32 hidden nodes. The output is from the final time-step $T = 352$.

Results & Discussion

Our evaluation metric is accuracy, weighted so that all classes are equally important. Our loss function is weighted cross-entropy loss, optimized using Adam. We trained each model on 5,886 examples for 46 epochs with a mini-batch size of 128. The test set consists of 1,962 examples.

Model	# Trainable Params	Train Accuracy	Test Accuracy
Metadata only	238	55.07%	48.40%
1D CNN only	16,414	66.26%	55.11%
LSTM only	5,582	48.77%	35.25%
CNN + Metadata	16,638	78.61%	70.09%
LSTM + Metadata	5,806	72.29%	51.58%

Overall, our models performed quite well, far exceeding the metadata-only logistic regression baseline of 48.4% accuracy. There is significant information in the time-series data, as evinced by the accuracy of the CNN-only model. That information differs from what the metadata encodes, as CNN-only performs better than metadata-only on some classes (15, 53, 67) and worse on other classes (64, 95).

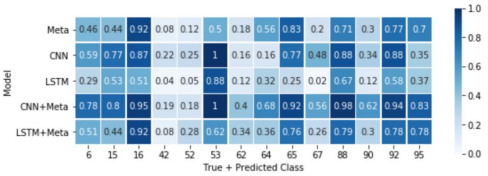

Figure 4: Recall matrix, showing the fraction of objects of class i correctly labeled

Future Work

- Improve model regularization, especially via data augmentation based on the measurement errors `flux_err` and `hostgal_photoz_err`.
- Create more hand-engineered metadata features based on the `flux` data, especially features that represent periodicity, which is lost during step 2 of data processing.
- Explore other neural network architectures, such as ResNets, Inception, and multi-layer RNNs.

[1] The PLAsTiCC team et al., "The Photometric LSST Astronomical Time-series Classification Challenge (PLAsTiCC): Data set," 2018 (arXiv:1810.00001).