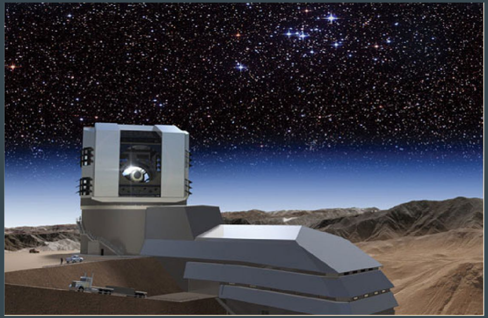
Classifying Astronomical Objects Using Time Series Neural Network

Matt Bundas

Final Project Presentation 11/30/2020

Overview Of Project

- Following a Kaggle Competition - PLAsTiCC Astronomical Classification: https://www.kaggle.com/c/PLAsTiCC-2018/overview
- Goal - Given simulated LSST observation data, create model which can classify the object.
- Motivation - LSST is an enormous telescope, has an enormous camera, and creates an ungodly amount of data, too much to process manually.
- Solution - Deep neural network



Credit: LSST/NSF/AURA/Todd Mason Productions Inc
https://www.kaggle.com/michaelapers/the-plasticc-astronomy-starter-kit

object_id	mjd	passband	flux	flux_err	detected_bool
int64	float64	int64	float64	float64	int64
10	56210.172	1	5.659	3.7	1
10	56210.188	2	21.32	3.245	1
10	56210.203	3	13.28	3.221	1
10	56210.234	4	9.579	3.85	1
10	56218.172	1	49.22	1.654	1
10	56218.191	2	114.6	2.376	1
10	56218.211	3	88.85	2.261	1
10	56219.156	4	75.01	2.394	1

object_id	ra	decl	gall	galb	ddf_bool	hostgal_specz	hostgal_photoz	hostgal_photoz_err
int64	float64	float64	float64	float64	int64	float64	float64	float64
615	349.046051	-61.943836	320.79653	-51.753706	1	0.0	0.0	0.0
713	53.085938	-27.784405	223.525509	-54.460748	1	1.8181	1.6153	0.55
730	33.574219	-6.579593	170.455585	-61.548219	1	0.232	0.2262	0.0157
745	0.189873	-45.586655	328.254458	-68.969298	1	0.3037	0.2813	1.1523

https://www.kaggle.com/michaelapers/the-plasticc-astronomy-starter-kit

Competition

- Most accurate estimate of probability for object to belong to each class.

- Evaluated using a custom log loss algorithm applied to predictions. Lowest log loss on test predictions wins!

- One class of object in test sets but not training data, class_99

- ~8,000 training examples, 3.5 million test examples

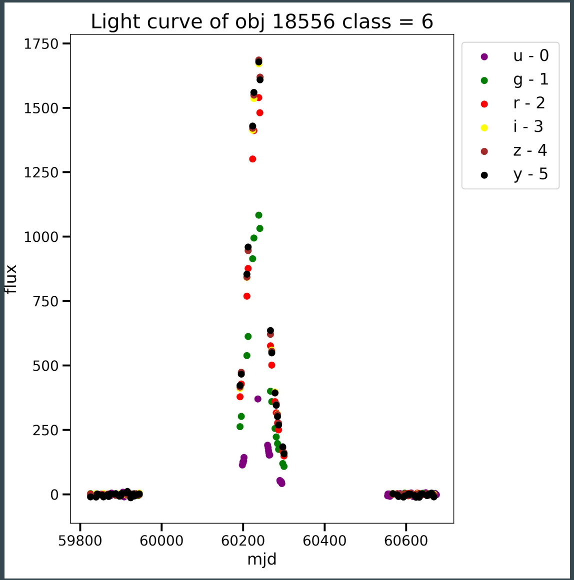
- Lots of collaboration between competitors

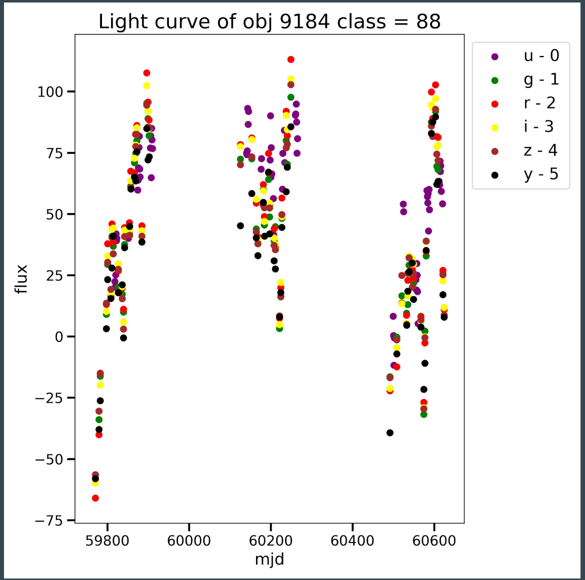
object_id	class_6	class_15	class_16	class_42	class_52	class_53	class_62	class_64	class_65	class_67	class_88	class_90	class_92	class_95	class_99
1000183	0.000299	0.01994	6.07E-06	0.204195	0.068375	1.15E-05	0.244842	0.020661	0.000304	0.184336	6.68E-05	0.230091	1.96E-06	0.026871	0.034977
1000235	0.155702	0.044877	0.001189	0.073496	0.049802	0.091944	0.071423	0.006812	0.031905	0.05321	0.011282	0.044595	0.146766	0.216997	0.031
1000260	0.000119	0.533414	1.48E-07	0.253402	0.047015	3.72E-06	0.038172	0.018308	0.00056	0.002109	0.000156	0.06825	2.33E-05	0.038467	0.076202
1000280	0.001188	0.001725	2.26E-05	0.047454	0.080728	3.69E-05	0.097123	0.008476	0.000484	0.575768	1.91E-05	0.182671	5.05E-06	0.004298	0.082253
1000282	0.008692	0.002962	0.019596	0.047707	0.066193	0.00197	0.072989	0.053307	0.174546	0.339726	0.008346	0.200231	0.000631	0.003104	0.048532
1000345	0.000421	0.045201	1.70E-05	0.195582	0.408577	2.04E-05	0.037273	0.002753	0.000428	0.009936	0.00073	0.289182	9.80E-05	0.009782	0.058368
1000388	0.000563	0.026746	6.79E-05	0.255745	0.057676	5.77E-05	0.323441	0.013377	0.000365	0.127342	0.00027	0.133995	2.42E-05	0.060331	0.046206
1000483	0.001065	0.019256	3.73E-05	0.122919	0.040059	1.59E-05	0.201559	0.123632	0.001595	0.312133	2.16E-05	0.164409	3.95E-06	0.013294	0.04459
1000490	0.000108	0.013401	7.92E-06	0.121894	0.065012	5.95E-06	0.164589	0.183999	0.001988	0.147714	7.61E-05	0.289859	1.66E-06	0.011343	0.041408
1000501	0.000219	0.744714	3.99E-07	0.122151	0.014495	7.39E-06	0.02091	0.039852	0.001584	0.001008	0.000243	0.027646	4.43E-05	0.027126	0.106388
1000587	0.000448	0.002042	0.011868	0.117381	0.435223	0.000436	0.085129	0.002305	0.000738	0.088165	0.006302	0.234909	0.001143	0.01391	0.062175
1000601	0.000255	0.058115	6.48E-07	0.190439	0.331314	1.58E-06	0.052952	0.007134	0.000441	0.011721	7.65E-06	0.33922	1.69E-05	0.008382	0.04846
1000614	0.00024	0.013034	1.45E-05	0.287401	0.098254	6.39E-05	0.272816	0.003669	0.000245	0.069802	0.00034	0.19992	2.92E-05	0.054171	0.041057

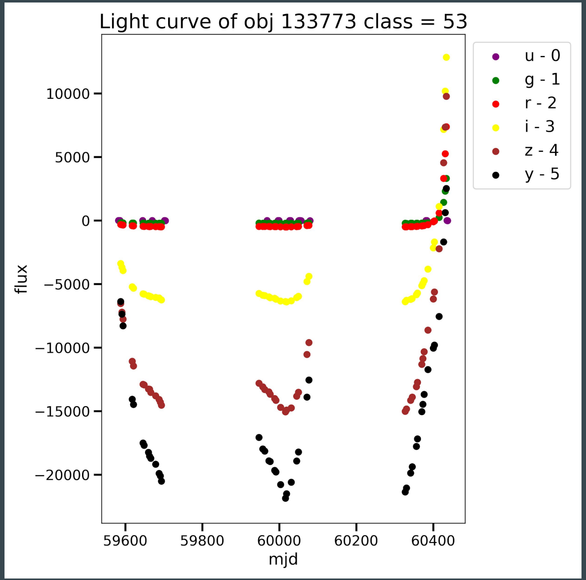
$$\text{Log Loss} = -\left(\frac{\sum_{i=1}^{M} w_i \cdot \sum_{j=1}^{N_i} \frac{y_{ij}}{N_i} \cdot \ln p_{ij}}{\sum_{i=1}^{M} w_i} \right)$$

https://www.kaggle.com/c/PLAsTiCC-2018/overview/evaluation

Light Curves







Engineered Features

- Applied to flux, flux_err, rolling average of flux:

    mean,median, max, standard deviation, skew, kurtosis various percentiles, max slopes etc.

- Applied to passband, mjd, a few flux/error ratios:

    - mean,median,std,skew

- LOTS ~100 features broken down by passband:

    mean, median, max, min, std

    raw differences and more importantly ratios

- Metadata:

    ra, dec, ddf, distmod, redshift etc.

- Difficult to do time and period features, take too long to calculate.

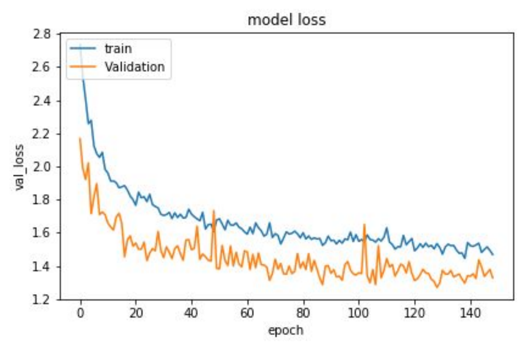
Neural Network

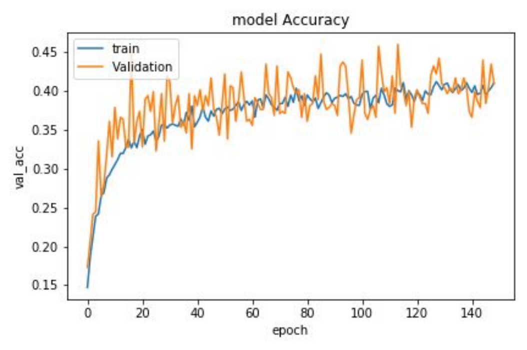
- Layers:
  5 Dense with batch normalization and dropout

- Parameters:
  Neurons - 512
  Dropout Rate - 0.40
  Epochs - 150
  Batch Size - 20

- Loss Function:
  Log Loss similar to competition metric
- Activation:
  Sigmoid
- Input Normalization:
  Built in ts.keras normalization

```
model = tf.keras.models.Sequential([

  tf.keras.layers.Dense(num_neurons, input_dim = num_features, activation='sigmoid'),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Dropout(drop_rate),

  tf.keras.layers.Dense(num_neurons, input_dim = num_features, activation='sigmoid'),
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Dropout(drop_rate),

  tf.keras.layers.Dense(num_neurons//2, input_dim = num_features, activation='sigmoid'
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Dropout(drop_rate),

  tf.keras.layers.Dense(num_neurons//2, input_dim = num_features, activation='sigmoid'
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Dropout(drop_rate),

  tf.keras.layers.Dense(num_neurons//4, input_dim = num_features, activation='sigmoid'
  tf.keras.layers.BatchNormalization(),
  tf.keras.layers.Dropout(drop_rate),

  tf.keras.layers.Dense(num_classes, activation='softmax')

])
```





Results

```
Done 3492890
loss with 99: 1.6591637198978222
Without 99 len 344700
loss without 99: 1.4247021992598299
```

My Score without Class 99: 1.42

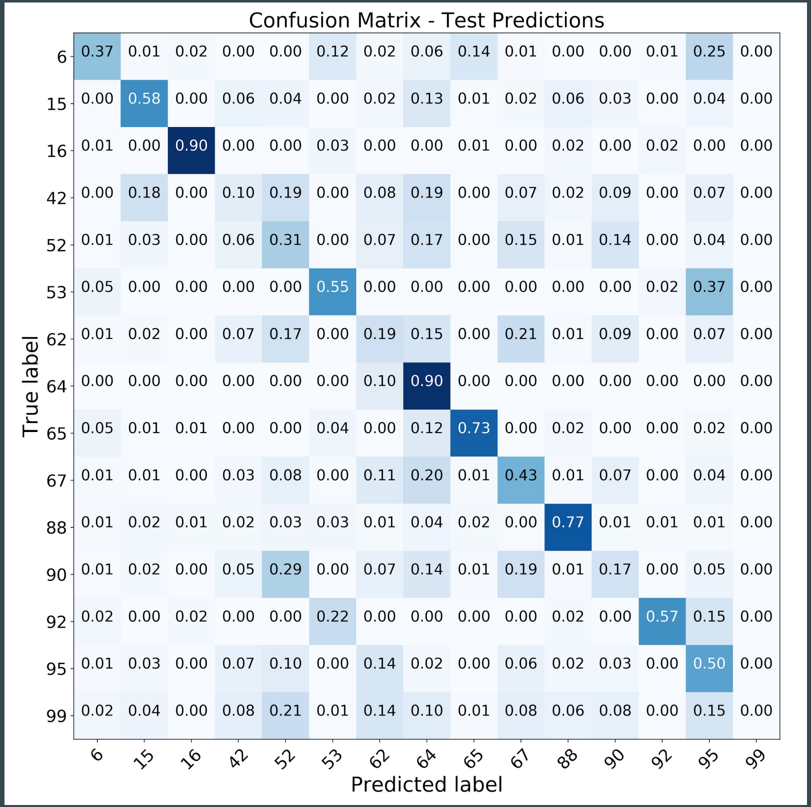
My score without Class 99: 1.66 - Down to 1.53 since the presentation!

Avg Score: 1.75

Winning Score: 0.685

Class 99 probability = (1 - maxP)/14

Struggled with classes 42,52,62, 90 who appear to have very similar light curves.



Lessons Learned

Large scale neural networks are a lot of work, especially engineering features and testing them.

Understanding the domain well is essential in coming up with good features.

Neural Networks might not be the best solution for everything.

Time to process data vs training time