Predicting Asteroid Diameter Using Artificial Neural Networks

Deep Learning

Cain Farnam

University of Central Arkansas

Outline

Section 1

Introduction

Introduction

- The data set contains 27 features on asteroids
- The task is to predict the asteroid's diameter based on the features provided
- To predict the diameter of the asteroids, we will be creating an artificial neural network

Section 2

Data

Raw Data Structure

feature	observations	non-null	data type
full name	839736	non-null	object
a	839734	non-null	float64
e	839736	non-null	float64
G	119	non-null	float64
i	839736	non-null	float64
om	839736	non-null	float64
w	839736	non-null	float64
q	839736	non-null	float64
ad	839730	non-null	float64
per y	839735	non-null	float64
data arc	823947	non-null	float64
condition code	838743	non-null	object
n obs used	839736	non-null	int64
H	837042	non-null	float64
diameter	137681	non-null	object
extent	18	non-null	object
albedo	136452	non-null	float64
rot per	18796	non-null	float64
GM	14	non-null	float64
BV	1021	non-null	float64
UB	979	non-null	float64
IR	1	non-null	float64
spec B	1666	non-null	object
spec T	980	non-null	object
neo	839730	non-null	object
pha	822814	non-null	object
moid	822814	non-null	float64

Data Cleaning

- The data set needs to be cleaned before it can be used
- Take the following steps:
  - Drop all observations that have missing values for diameter
  - Drop arbitrary features (asteroid name and condition code)
  - Drop features containing too many missing values
  - Encode categorical features
  - Fill remaining missing values with feature mean

Clean Data Structure

- The cleaned data set has 137,680 asteroids that can be used for training

feature name	observations	non-null	data type
a	137680	non-null	float64
e	137680	non-null	float64
i	137680	non-null	float64
om	137680	non-null	float64
w	137680	non-null	float64
q	137680	non-null	float64
ad	137680	non-null	float64
per y	137680	non-null	float64
data arc	137680	non-null	float64
n obs used	137680	non-null	int64
H	137680	non-null	float64
diameter	137680	non-null	float64
albedo	137680	non-null	float64
moid	137680	non-null	float64
neo N	137680	non-null	uint8
neo Y	137680	non-null	uint8
pha N	137680	non-null	uint8
pha Y	137680	non-null	uint8

Summary Statistics

- Summary statistics of the numerical features

```
##                a          e          i  ...   diameter     albedo       moid
## count  137680.00  137680.00  137680.00  ...  137680.00  137680.00  137680.00
## mean        2.81       0.15      10.35  ...       5.48       0.13       1.42
## std         1.52       0.08       6.84  ...       9.37       0.11       0.51
## min         0.63       0.00       0.02  ...       0.00       0.00       0.00
## 25%         2.54       0.09       5.12  ...       2.77       0.05       1.08
## 50%         2.75       0.14       9.39  ...       3.96       0.08       1.38
## 75%         3.09       0.19      13.74  ...       5.74       0.19       1.70
## max       389.15       0.98     170.32  ...     939.40       1.00      39.51
##
## [8 rows x 14 columns]
```

Correlation

- Correlation coefficients between diameter and predictors
- The data doesn't appear to be highly correlated

feature	correlation
diameter	1.000000
H	0.568580
data arc	0.493088
n obs used	0.386325
moid	0.333046
q	0.330317
a	0.145027
albedo	0.107493
ad	0.093622
i	0.052827
e	0.049180
per y	0.049053
neo Y	0.036215
neo N	0.036215
pha N	0.019629
pha Y	0.019629
w	0.002909
om	0.001191

Section 3

Pre-Processing

Pre-Processing

- Split the data into a train and test sets (80:20 split) and separate the targets from the predictors
- Normalize the predictors using a standard scaler
- Set aside 25,000 observations for hold-out validation to observe the performance during the training process (approx same as test set)

Section 4

Deep Neural Network

Model Architecture

- The following model structure remains constant:
  - Since this is a regression problem, use:
    - Loss function: mean square error (mse)
    - Optimizer: ADAM
    - Metric: mean absolute error (mae)
  - Layers (activation): dense (rectified linear unit function)
  - Output layer (activation): one unit (none)
- Tune the following:
  - batch size
  - number of hidden layers
  - size of hidden layers
  - dropout layers
  - epochs

Initial Model Parameters

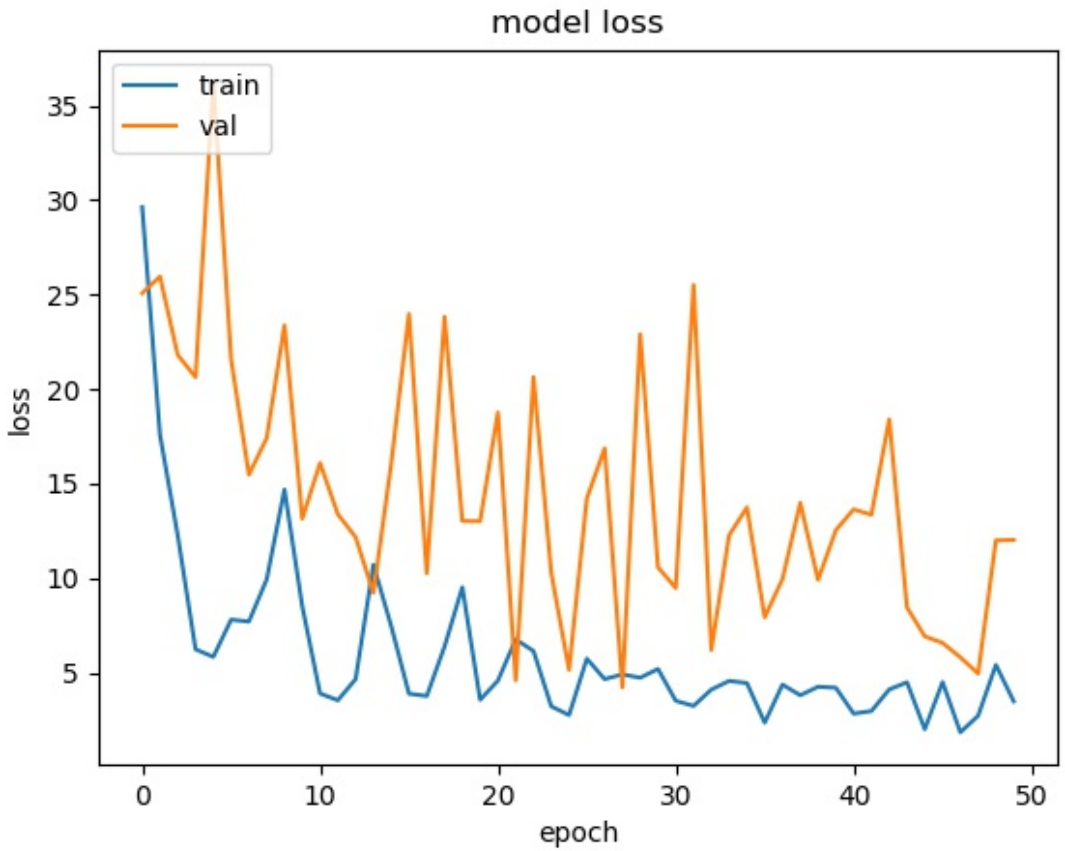
- The first model needs to be large enough that it over-fits the train set
- The goal is to over-fit the data and then scale down until we achieve the optimal model that generalizes the data
- Use the following hyper-parameters:
  - batch size: 64
  - number of hidden layers: 4
  - size of hidden layers: 128, 128, 64, 64
  - dropout layers: none
  - epochs: 50

Model Summary

```
## _________________________________________________________________
## Layer (type)                 Output Shape              Param #
## =================================================================
## dense_1 (Dense)              (None, 128)               2304
## _________________________________________________________________
## dense_2 (Dense)              (None, 128)               16512
## _________________________________________________________________
## dense_3 (Dense)              (None, 64)                8256
## _________________________________________________________________
## dense_4 (Dense)              (None, 64)                4160
## _________________________________________________________________
## dense_5 (Dense)              (None, 1)                 65
## =================================================================
## Total params: 31,297
## Trainable params: 31,297
## Non-trainable params: 0
## _________________________________________________________________
```

Model Loss

- The train and validation loss have quite a bit of noise
- Over-fitting the train set



Section 5

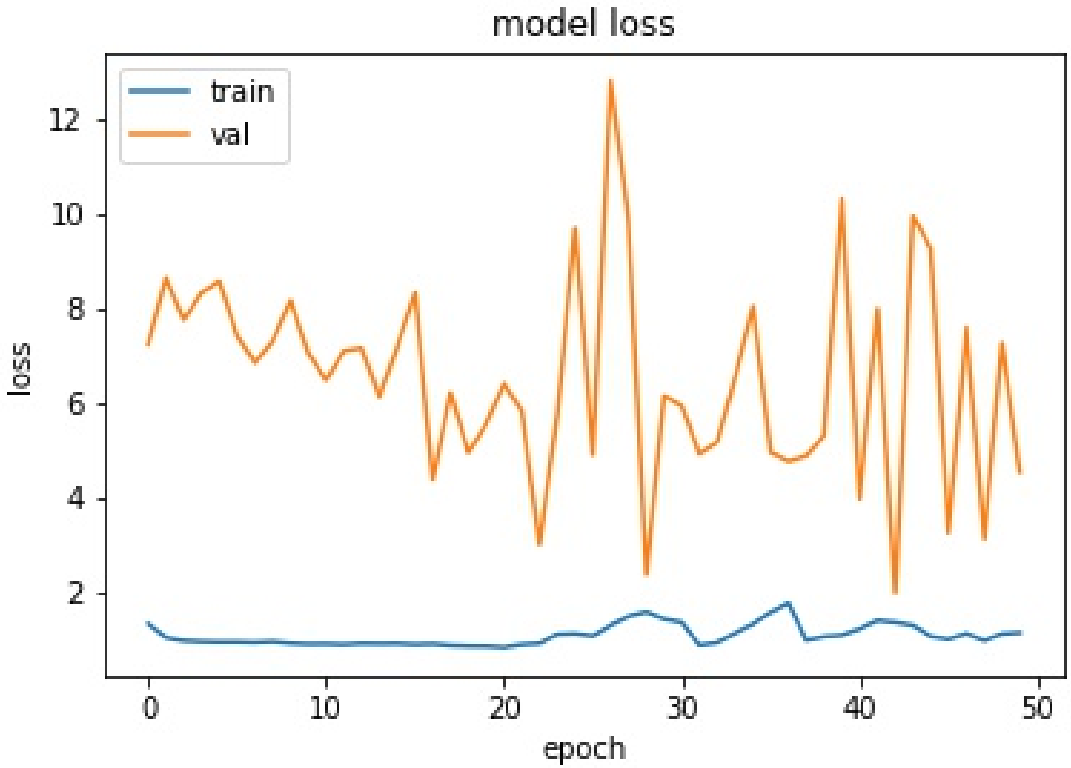
Hyper-parameter Tuning

Tuning Hyper-parameters

- Begin tuning the hyper-parameters of the model
- Eliminate the noise in the train loss
- Likely a result of high variance between batches
- Increase the batch size and re-run the same model (batch size: 512)

Model Loss

- Train loss is now much more smooth
- Still over-fitting the train set



Reducing Model Size

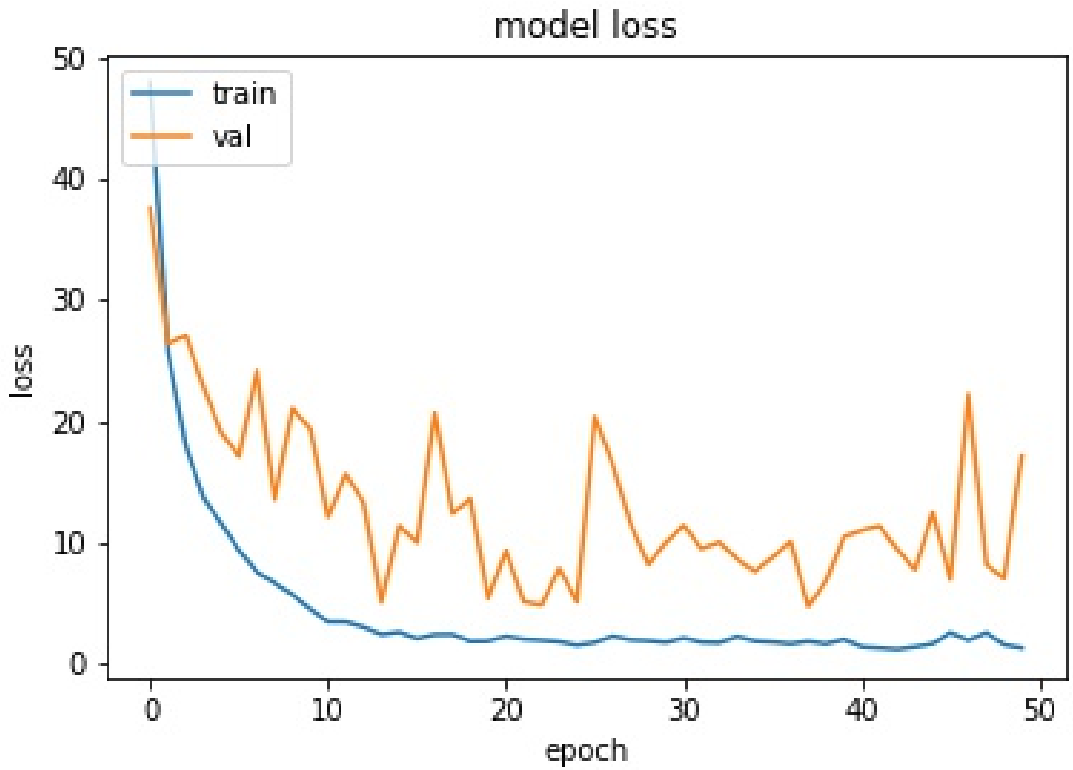
- Reduce the model size by halving the output size for each layer
- Use the following hyper-parameters:
  - batch size: 512
  - number of hidden layers: 4
  - size of hidden layers: 64, 64, 32, 32
  - dropout layers: none
  - epochs: 50

Model Summary

```
## _________________________________________________________________
## Layer (type)                 Output Shape              Param #
## =================================================================
## dense_6 (Dense)              (None, 64)                1152
## _________________________________________________________________
## dense_7 (Dense)              (None, 64)                4160
## _________________________________________________________________
## dense_8 (Dense)              (None, 32)                2080
## _________________________________________________________________
## dense_9 (Dense)              (None, 32)                1056
## _________________________________________________________________
## dense_10 (Dense)             (None, 1)                 33
## =================================================================
## Total params: 8,481
## Trainable params: 8,481
## Non-trainable params: 0
## _________________________________________________________________
```

Model Loss

- Over-fitting is reduced, but still prevalent
- Train loss is converging fast (around 10 epochs)



Reducing Number of Layers

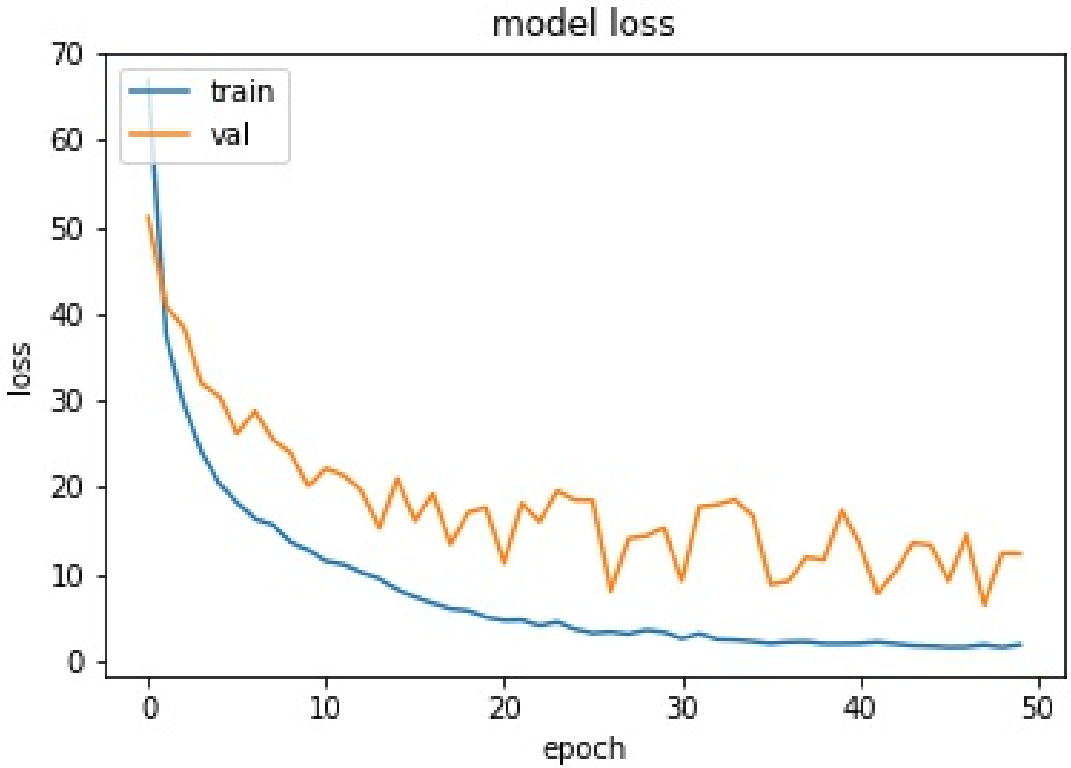
- Remove a large hidden layer and reduce the output for the last hidden layer
- Use the following hyper-parameters:
  - batch size: 512
  - number of hidden layers: 3
  - size of hidden layers: 64, 32, 16
  - dropout layers: none
  - epochs: 50

Model Summary

```
## _________________________________________________________________
## Layer (type)                 Output Shape              Param #
## =================================================================
## dense_11 (Dense)             (None, 64)                1152
## _________________________________________________________________
## dense_12 (Dense)             (None, 32)                2080
## _________________________________________________________________
## dense_13 (Dense)             (None, 16)                528
## _________________________________________________________________
## dense_14 (Dense)             (None, 1)                 17
## =================================================================
## Total params: 3,777
## Trainable params: 3,777
## Non-trainable params: 0
## _________________________________________________________________
```

Model Loss

- Over-fitting reduced
- Train loss is now converging much later, around 30 epochs



Adding Dropout Layers

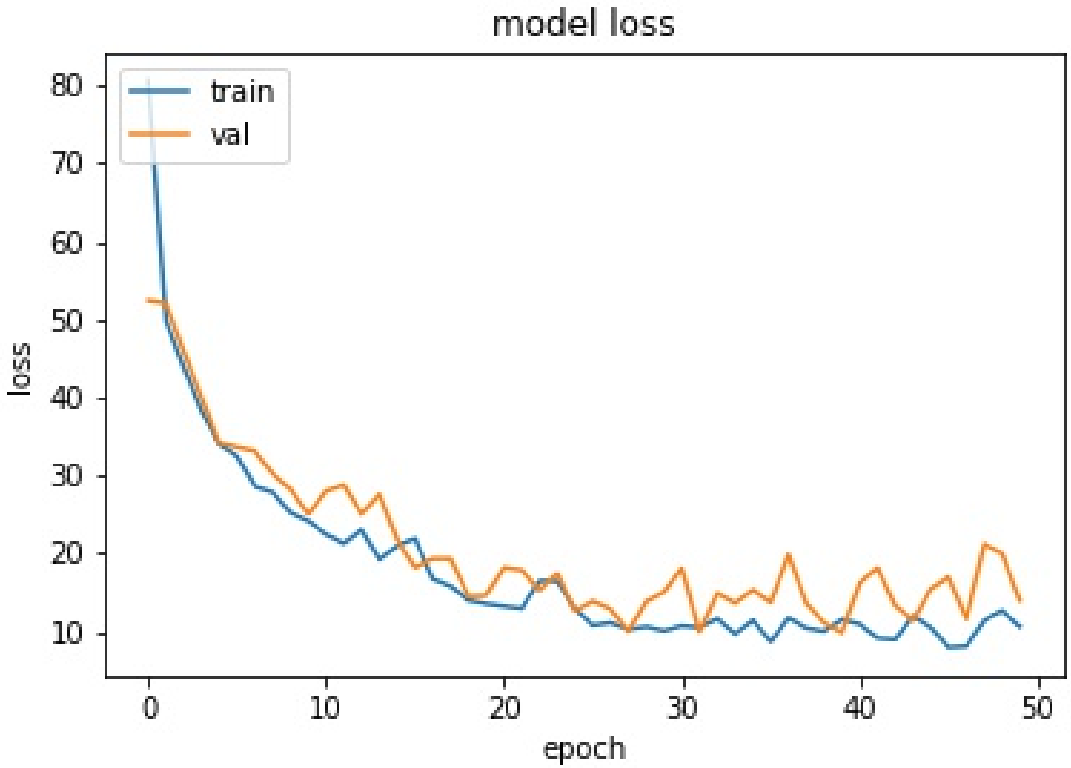
- Rather than reducing the model size, add a dropout layer between each hidden layer
- Use the following hyper-parameters:
  - batch size: 512
  - number of hidden layers: 3
  - size of hidden layers: 64, 32, 16
  - dropout layers: 3 (rate: 0.25)
  - epochs: 50

Model Summary

```
## _________________________________________________________________
## Layer (type)                 Output Shape              Param #
## =================================================================
## dense_15 (Dense)             (None, 64)                1152
## _________________________________________________________________
## dropout_1 (Dropout)          (None, 64)                0
## _________________________________________________________________
## dense_16 (Dense)             (None, 32)                2080
## _________________________________________________________________
## dropout_2 (Dropout)          (None, 32)                0
## _________________________________________________________________
## dense_17 (Dense)             (None, 16)                528
## _________________________________________________________________
## dropout_3 (Dropout)          (None, 16)                0
## _________________________________________________________________
## dense_18 (Dense)             (None, 1)                 17
## =================================================================
## Total params: 3,777
## Trainable params: 3,777
## Non-trainable params: 0
## _________________________________________________________________
```

Model Loss

- Train and validation loss are nearly the same
- The model begins to converge around 30 epochs



Results

- Train the last model again, but only for 30 epochs:
  - batch size: 512
  - number of hidden layers: 3
  - size of hidden layers: 64, 32, 16
  - dropout layers: 3 (rate: 0.25)
  - epochs: 30
- Evaluate the model on the test set
  - $mae = 0.969$
  - $R^2 = 0.895$

Section 6

Feature Engineering

Feature Engineering

- Increase model performance with feature engineering
- Log is a common transformation in astrophysics
- Use the log of diameter as the new target
- Add log transforms for all numeric predictors

Correlation

- Top 10 predictor correlation coefficients with the log of the diameter
- Correlation in the data has increased significantly

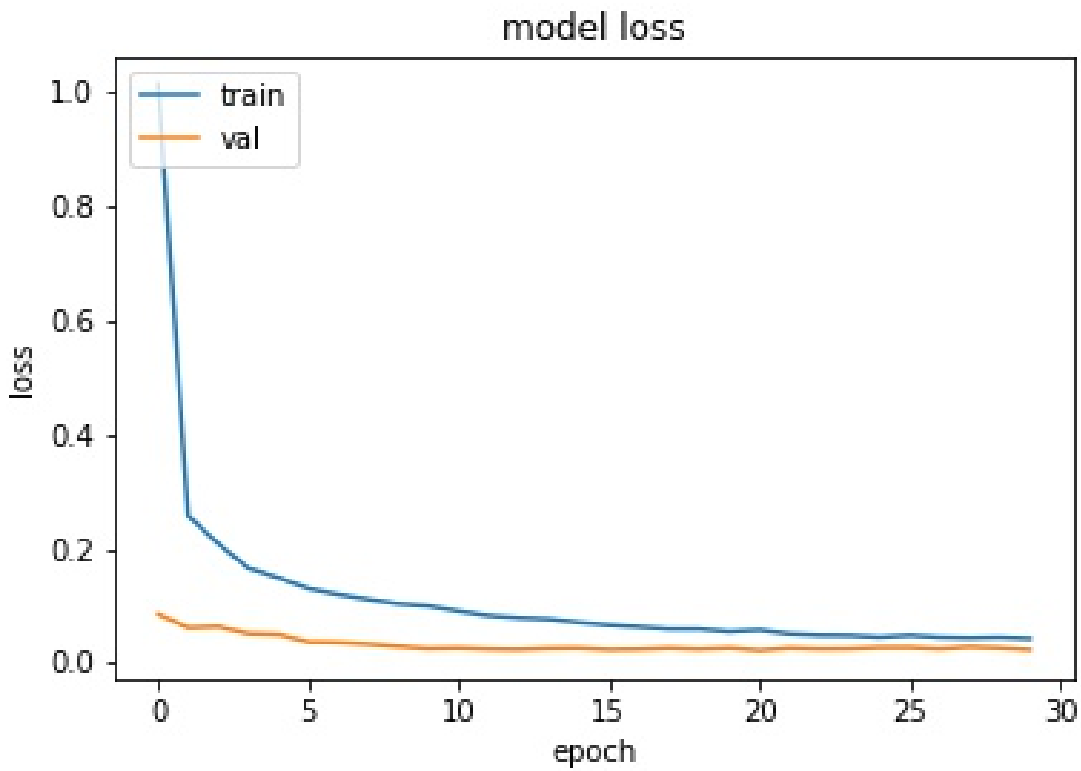
feature	correlation
diameter	1.000000
log(H)	0.835468
H	0.832306
log(a)	0.563592
log(per y)	0.563592
log(q)	0.543708
log(moid)	0.528661
q	0.522391
moid	0.521080
data arc	0.519099
n obs used	0.509806

Previous Model

- Train the model using the tuned hyper-parameters from the previous model:
  - batch size: 512
  - number of hidden layers: 3
  - size of hidden layers: 64, 32, 16
  - dropout layers: 3 (rate: 0.25)
  - epochs: 30

Model Loss

- The model converges within the first couple of epochs



Simpler Model

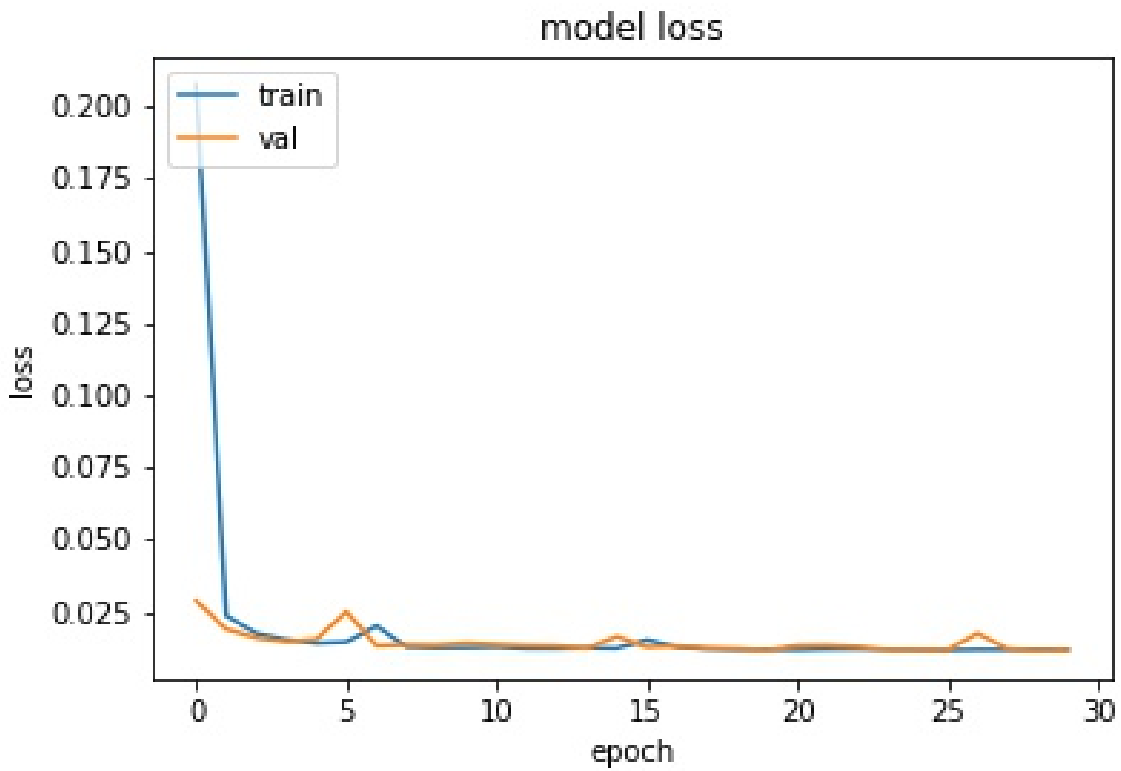
- Use a more basic model
- Reduce the model complexity
- Use the following hyper-parameters:
  - batch size: 64
  - number of hidden layers: 2
  - size of hidden layers: 16, 8
  - dropout layers: none
  - epochs: 10

Model Summary

```
## _________________________________________________________________
## Layer (type)                 Output Shape              Param #
## =================================================================
## dense_19 (Dense)             (None, 16)                496
## _________________________________________________________________
## dense_20 (Dense)             (None, 8)                 136
## _________________________________________________________________
## dense_21 (Dense)             (None, 1)                 9
## =================================================================
## Total params: 641
## Trainable params: 641
## Non-trainable params: 0
## _________________________________________________________________
```

Model Loss

- The loss in converging extremely fast, even with such a small model



Results

- Train the last model again, but only with 4 epochs:
  - batch size: 64
  - number of hidden layers: 3
  - size of hidden layers: 64, 32, 16
  - dropout layers: 3 (rate: 0.25)
  - epochs: 4
- Evaluate the model on the test set:
  - $mae = 0.078$
  - $R^2 = 0.967$

Section 7

Conclusion

Conclusion

- Developed a deep neural network to predict asteroid diameter
- Used hold-out validation
- Tuned the hyper-parameters to generalize the data
- Evaluated the model on the test set and got $R^2 = 0.895$
- Engineered features to enhance model performance
- Developed a much more simple model
- Evaluated the model on the test set and got $R^2 = 0.967$