|  |
| --- |
|  |

|  |
| --- |
| Optimization Techniques |

|  |  |
| --- | --- |
| Presented by:  Omar Ihab El-Hariry 5360  Ahmed Tharwat Magdy 5336  Aly Mahmoud 5304  Mostafa Ehab ElZwawy 5311  Malak Kassem 5979  Omar Hisham Tawfik 5845  Youssef Amr ElEssawi 4284  Ayman Ahmed Abdelaziz 5529 |  |



Final Project

Contents

Summary 2

Overview3

Dataset…………………….………….…………………………………………………………………………………..4

Data Analysis…………....…………………………………………………………………………………………….,5

Surrogate Model………………………………………………………………………………………………………6

Multi-Objective Optimization..………………………………………………………………………………...8

Code…………………………………………………………………………………………………………………………9

|  |
| --- |
| Summary |

Problem Statement:

It was required to find the resolution of the scanned image of the national ID, that achieves the best compromise between transfer time from the mobile and the computer using bluetooth interface.

Solution approach:

The solution approach was done in several steps:

1. Data acquisition using simulators
2. Creating the dataset
3. Analysing the data
4. Creating the surrogate model and training it
5. Testing the surrogate model and fine tune the hyperparameters
6. Get the pareto fronts and find the optimal, using multi-objective optimization methods

Language used:

* Python3

Framework used:

* Tensorflow 2.0

Libraries used:

1. Numpy
2. Matplotlib
3. Keras
4. Seaborn
5. Pandas

|  |
| --- |
| Overview |

The problem is to get the optimal resolution, that is the width and the height of the scanned image, that achieves the best compromise between minimizing file transfer time using bluetooth and maximizing quality.

It was found that what affects the transfer time of the image, is the size of the image and the pixels of the image are what affects the size of the image, more pixels means bigger size, means longer transfer time, and what defines the quality of the image is how the pixels of the image are distributed throughout the screen, so the quality of the image can be measured using what is known by PPI or Pixels-Per-Inch, so that means the higher the resolution of the image, the more pixels per inch, so here we can see the conflict of interests between maximizing the quality and minimzing the transfer time.

Note:

Resizing the images causes the image to make an interpolation of the resized image using interpolation algorithms like KNN and bilinear that either fills the gaps if resizing to a bigger image or averaging the pixels if resizing to a smaller image, and these algorithms yields great results in interpolating the images, so the quality pretty much increases while increasing the size of the original image.

|  |
| --- |
| Dataset |

Data acquisition and dataset creation:

Data acquisition was done by simulating the transfer process by trying different resolutions with different aspect ratios, starting all the way from a 16x16 image to a 16K image, and then calculating the size of the image taking into consideration no compression used to the image, after calculating the size the transfer time was calculated in seconds using Bluetooth 1.0 interface with latency up to 4 seconds applied, and the PPI was calculated by the reciever’s screen dimensions.

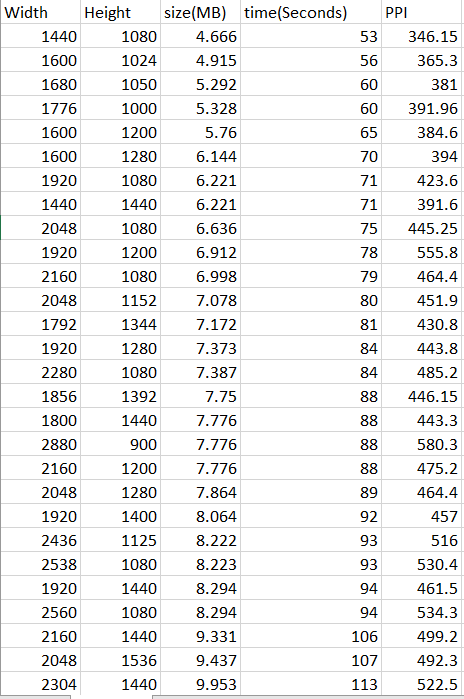
After aquiring the data, the dataset was created in the form of a spreadsheet (csv file)

The dataset created has 194 enteries, and has 5 columns: Width, Height, size in Megabytes, transfer time in Seconds and quality in PPI

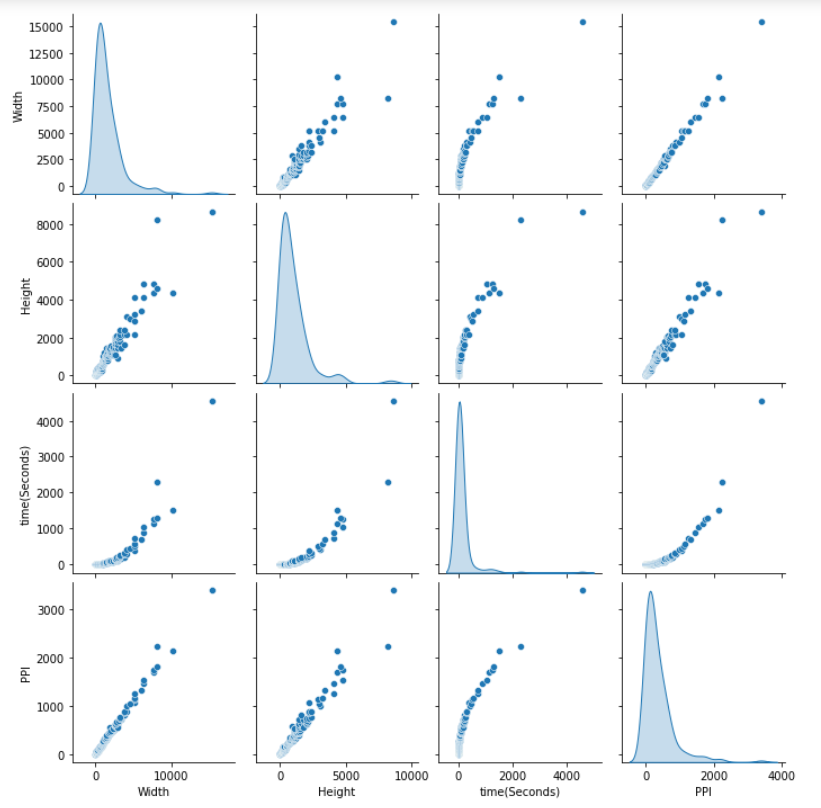
The dataset was split to 60% of its enteries to training and 40% to validation, this split was chosen because the dataset isn’t quite large, and to make get a rough sense if our model is overfitting the training data or not.

Assumptions:

1. Bluetooth 1.0 interface
2. No compression of the image
3. Egyptian National ID image
4. PPI was calculated using the dimensions of 1 screen
5. The image was scanned using the camera of of 1 phone



|  |
| --- |
| Data Analysis |

To have some knowledge about the data before creating the model is very important, so simple data analysis was done by plotting the features of the dataset against each other to have some sense about how the model would look like

|  |
| --- |
| Surrogate Model |

A Surrogate model is a model used if the adressed problem is hard to get solutions using analytic methods, like here in the case of this problem that we cannot find an equation that defines time and quality given width and height, so the Surrogate model used was an artificial neural network regressor, a simple linear regression model didn’t yield great results because as we can see in the plots above the time and PPI are not linear functions of width and time, so a more complicated model was required.

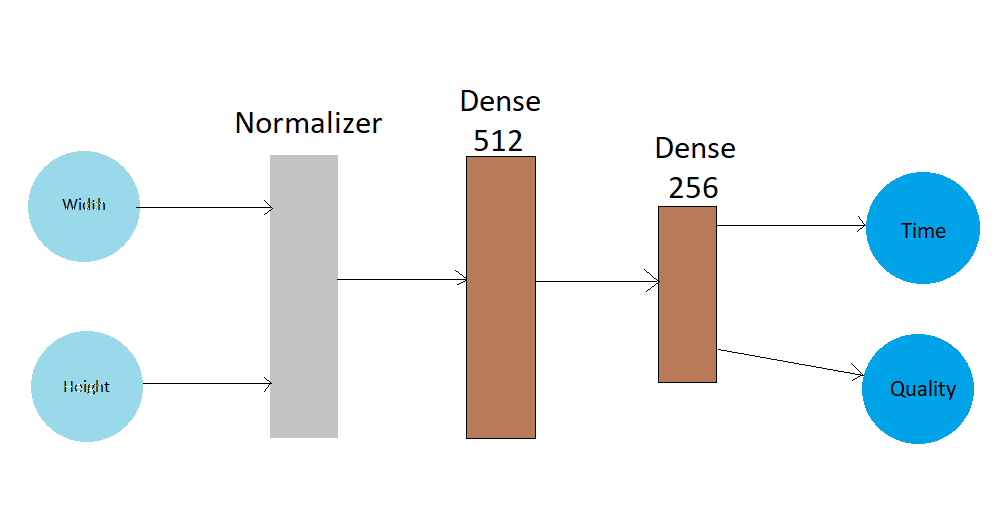
After exhaustive model researching to get the best neural network that fits the data best, the model chosen was a neural network with 2 input features, width and height and several hidden layers and then 2 outputs, transfer time in seconds and quality in PPI

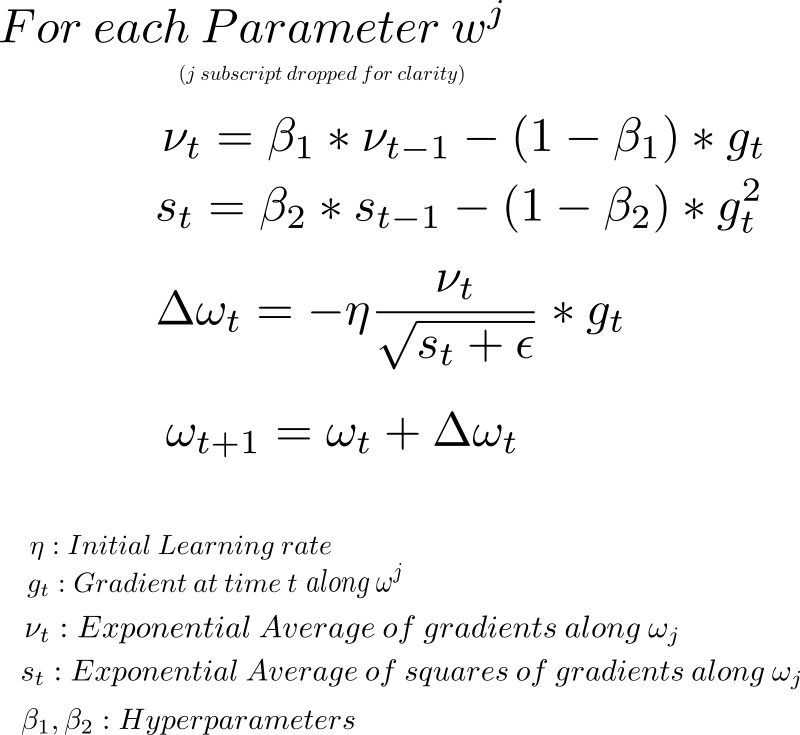
Model description:

To get the best results, a normalization to the features feeding into the model is required, so a normalizing layer was placed right after the input layer that takes the features and normalize them, to make the scale of the training features between 0 and 1, the normalization is adapted to the training features only, without the test features because the test data should be for testing only and the model shouldn’t have any estimate of how the test data look like.

After exhaustive research of the hyperparameters, the hyperparameters choses were:

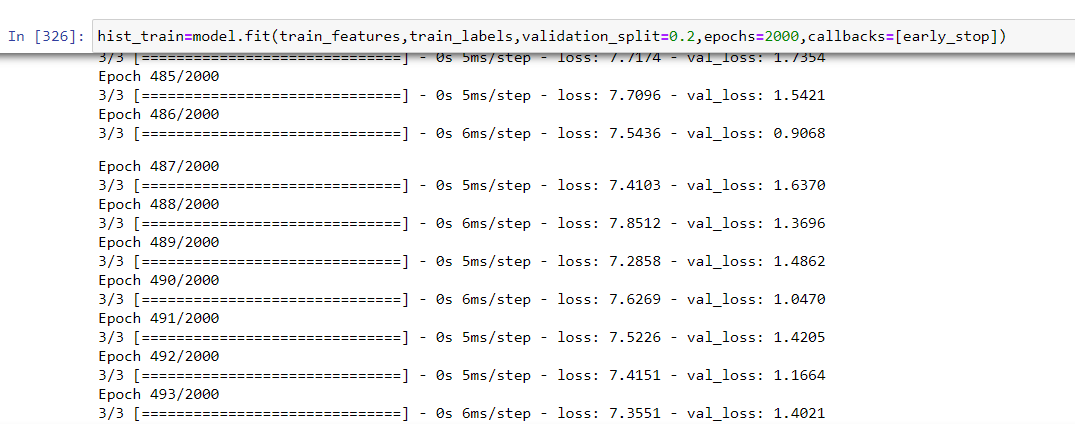
* Adam optimizer with learning rate of 0.001, beta1=0.9,beta2=0.999 and epsilon=10-8
* Mean Absolute Error loss function
* 2 hidden layers with 512 and 256 neurons respectively with activation function ReLU on both

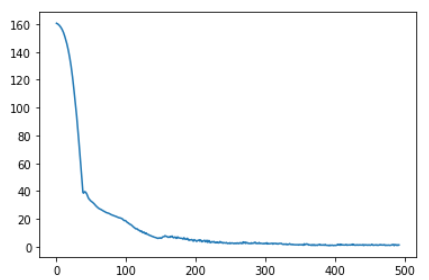
So the blueprint of out model looks like this:

The optimization function used for this model was Adam which uses exponential weighted averages of gradients and exponential weighted averages of the square of the gradients to compute the new gradient, and it’s a form of stochastic gradient descent, the new weights is caluclated using the below formula:

|  |
| --- |
| Surrogate Model |

The model training was set to a maximum of 2000 epochs over the training data with a cross-validation split of 20% of the training data, to make get a rough sense of how the model would perform on the validation set.

An Early Stop callback function was used to ensure that the model doesn’t overfit the data, with a patience of 100 iterations that if the cross-validation loss didn’t decrease in 100 iterations the model stops training and it returns the weights that resulted in the minimum cross-validation loss, this plot shows how the loss function decreases over the iterations.



After the model finished training, the validation set is used to evaluate the model, this procedure was done many times to ensure that the final model design performed the best out of all the models tried and the loss of the final model is acceptable, and then when the final model is chosen, then we train the model furthermore on the validation data to further improve the model, but this time since the validation data is less than the training data we define make a new callback that has patience over 50 iterations only.

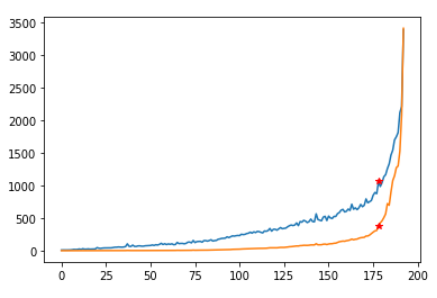
|  |
| --- |
| Multi-Objective Optimization |

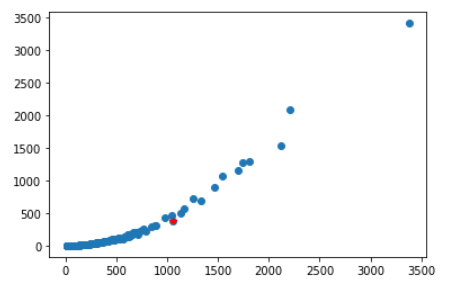
There exists a conflict of interests between transfer time and quality as we explained before, so we need to find the best trade-off point, that compromises between transfer time and quality, assuming that there are no boundaries to neither transfer time nor quality.

The multiobjective optimization solution used was weighted sum of objective functions, and the weight was chosen by exhaustive brute force to find the point that was least changed by trying different weights.

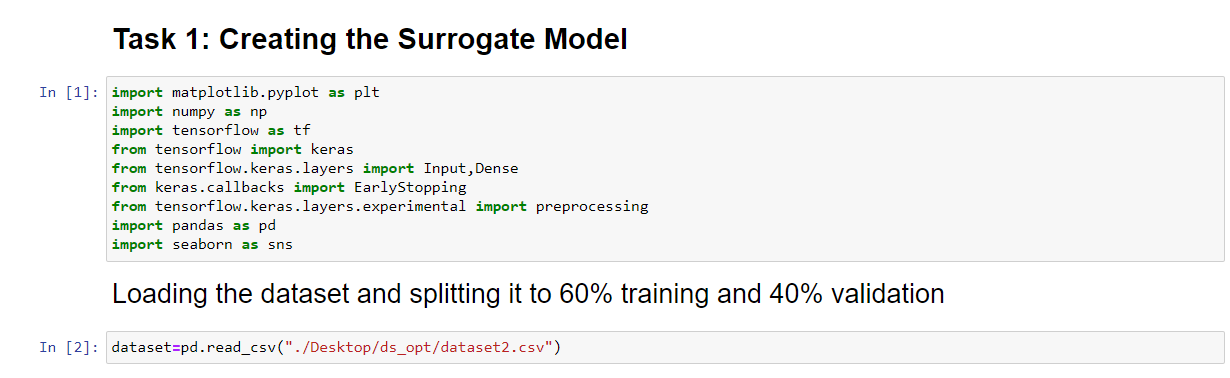
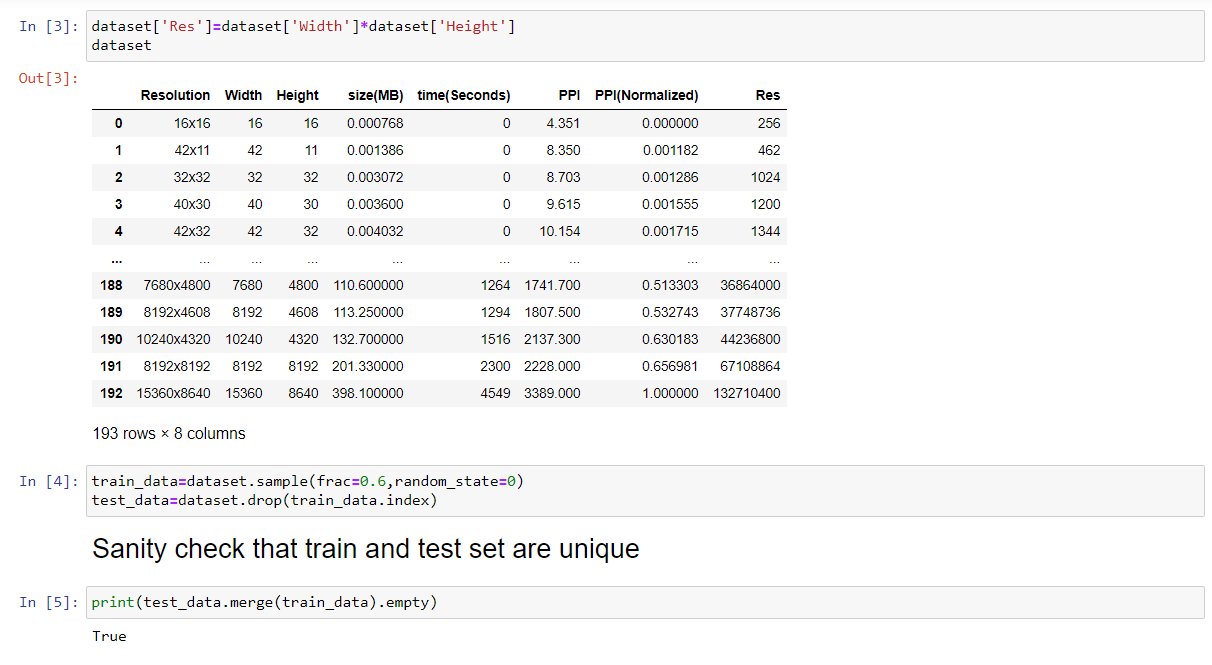
To maximize a function is to minimize its inverse, so for the sake of simplicity our 2 objectives are minimizing the transfer time and minimizing the inverse of the quality, so a weight was placed to the quality objective starting by -10000, because prior to that weight, it was found that the results were not pleasing, and down to a weight of -1500000, and it was found that the best trade-off point in the dataset is 5120x2160, as this point was stable from the weight value of -400000 to -1500000 and that is 110 iterations with a steo of -10000

Scatter plot of quality on x-axis and Plot of quality (Blue) and time (orange)

 time on the y axis through the iterations

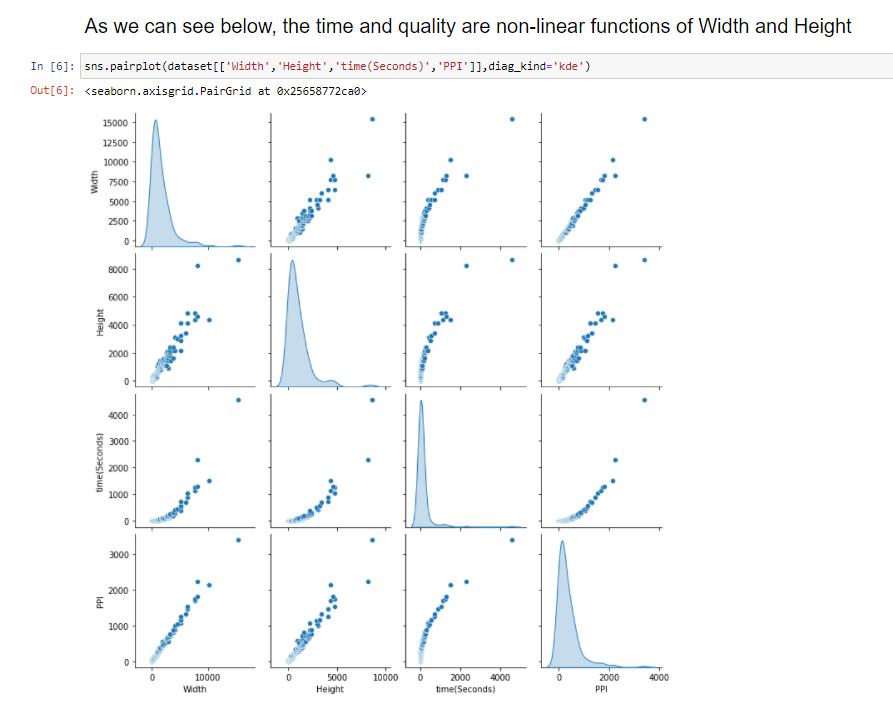


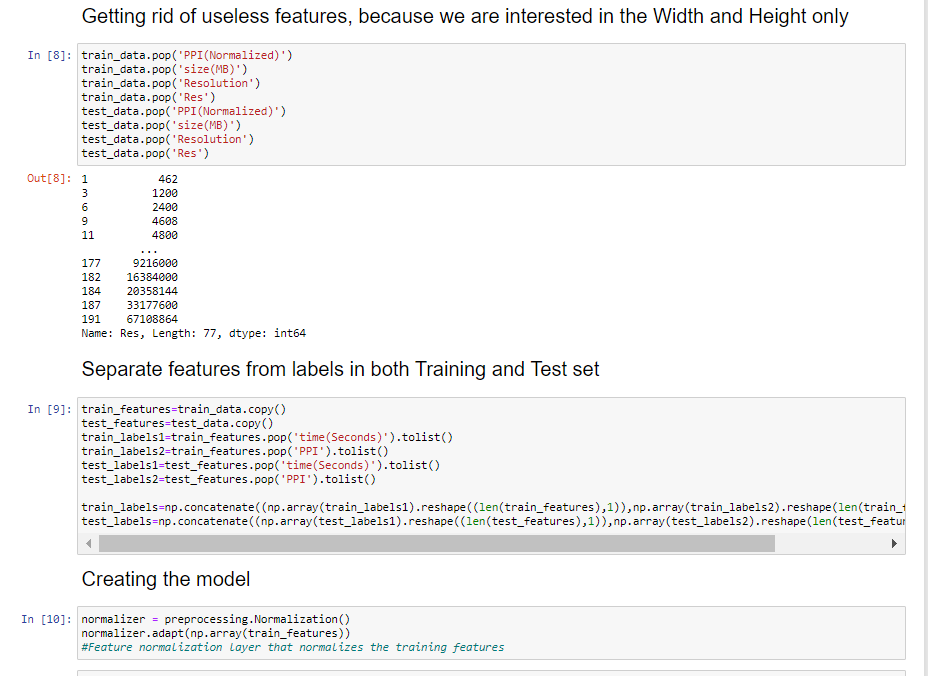
The red dots mark the optimal point’s quality and transfer time, and the plot on the right shows that the point chosen, truly compromises between quality and time with no boundaries placed as by increasing the quality more, it will lead to significally greater time (orange plot) and by decreasing the time more it will lead to significally less quality (blue plot).



|  |
| --- |
| Code |

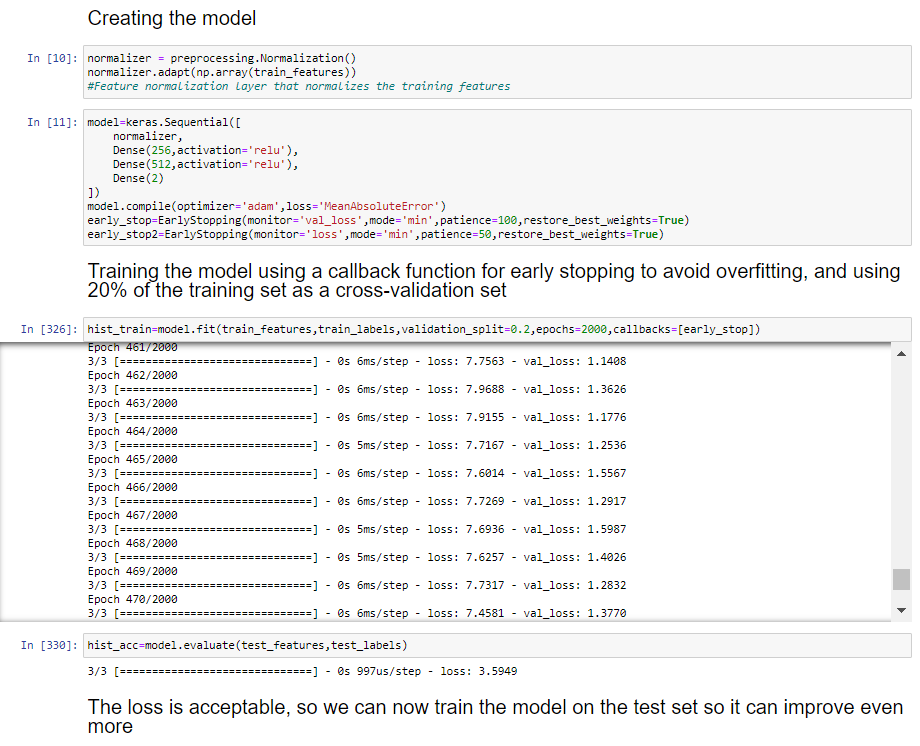
|  |
| --- |
| Code |



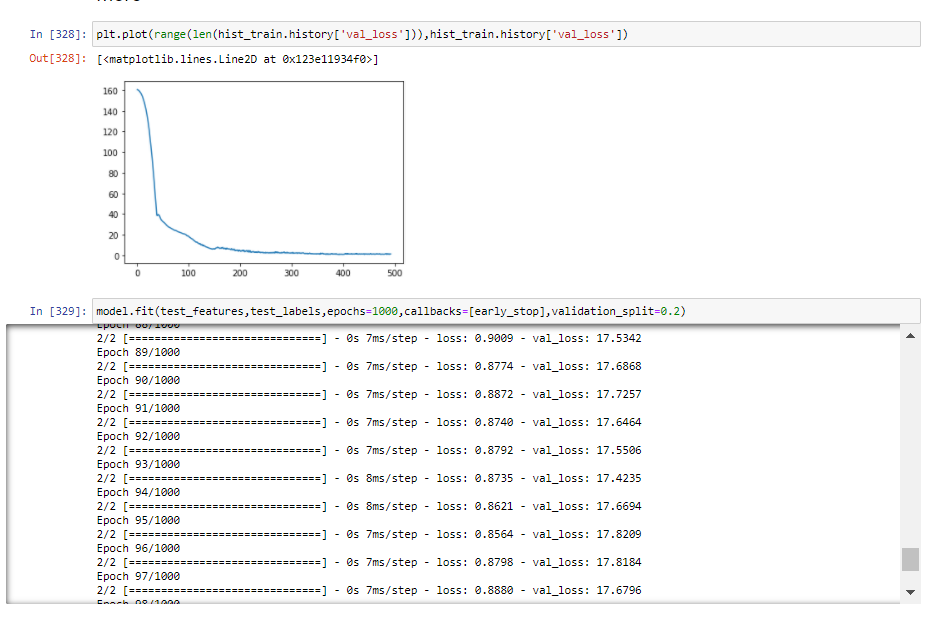


|  |
| --- |
| Code |

|  |
| --- |
| Code |



|  |
| --- |
| Code |



|  |
| --- |
| Code |

