David Stokes 801056264

Homework 1

GIT LINK: https://github.com/Fallt0earth/RTML_HW1

1.

1.c. For each image tensor, use the .mean() method to get a sense of how bright the image is.

The average brightness is easily calculated by getting the mean value of the pixels. The green and blue image was brighter than the red image by a factor of almost 33%.

1.d Take the mean of each channel of your images. Can you identify the red, green, and blue items from only the channel averages?

Its relatively easy to determine which color is which as the value that the primary color is significantly larger than the others. I did this by isolating each channels mean and displaying it. It was interesting that the red image made the least usage of the blue and green channels. This is opposed to the green image which used about 66% of the total green channel as the red channel. This behavior was duplicated with the blue image where the secondary color was the green channel again with approximately 66% of what the blue values are.

2.

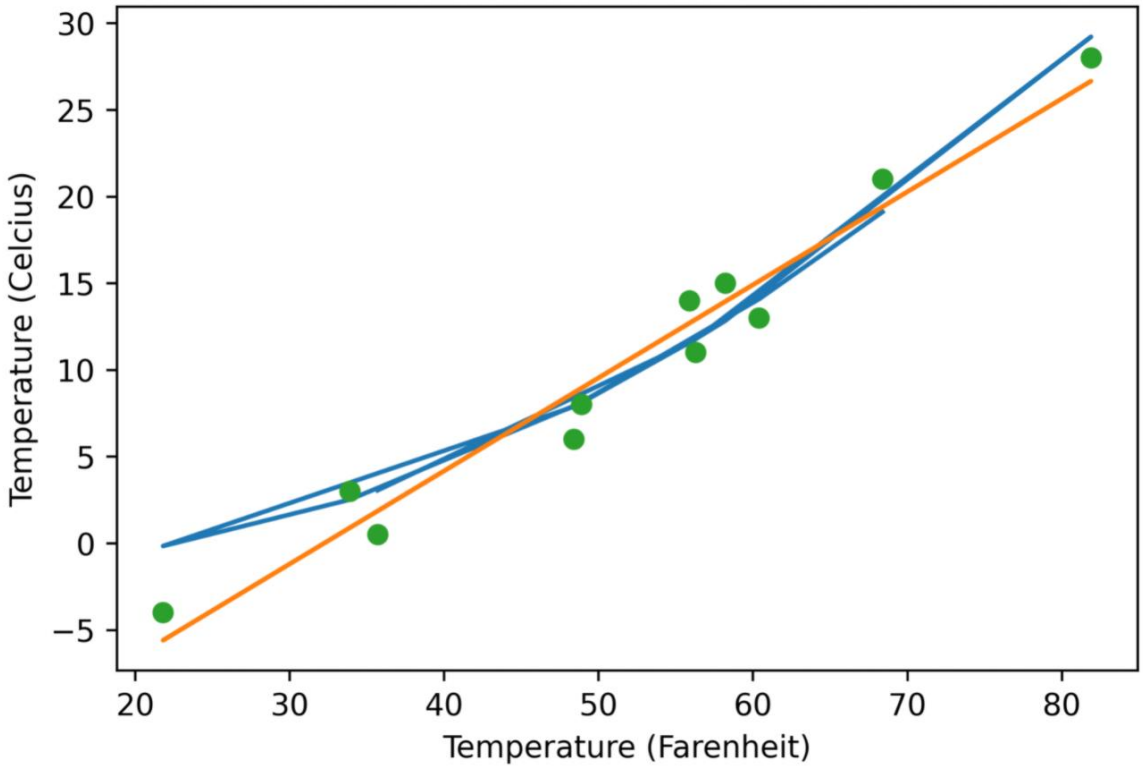
2.b Use 5000 epochs for your training. Explore different learning rates from 0.1 to 0.0001 (you need four separate trainings). Report your loss for every 500 epochs per training.

I found that the only learning rate that converged was 1e-4. This is likely due to the learning rate being too high to catch the local minimums and instead the loss went to infinity.

2.c Pick the best non-linear model and compare your final best loss against the linear model that we did during the lecture. For this, visualize the non-linear model against the linear model over the input dataset, as we did during the lecture. Is the actual result better or worse than our baseline linear model?

The results of using a polynomial model are far less accuracy. The model has a worse ability to generalize based on the graph. This is likely due to the inherent linear model that exists to describe the temperature conversion. We are trying to map a polynomial function to a linear function. This is obviously less than ideal compared to attempting to map a linear function to a linear function. This is supported by the higher loss values for polynomial approach.

David Stokes 801056264



3.

3.c Pick the best linear model (the one with the smaller final loss) and visualize it over the input dataset, as we did during the lecture.

The best linear model was one with a learning rate of 1e-3. This provided the fastest convergence and resulted in incredibly low loss values. It was interesting that the lower 1e-4 learning rate seemed to get stuck sooner. The accuracy of the model to generalize is somewhat lower than what I would like with the model either being highly accurate or highly inaccurate. Its somewhat at odds with the loss values I saw which raises questions about if the loss is an accurate representation of the model's accuracy.

David Stokes 801056264