Pokémon Assignment Part 1

So, part 1 was to assess what Pokémon were legendary based on the data found and to build a machine learning classifier to determine this.

I used sequential modelling for this and used followed the model life cycle shown in the previous class notes to fully understand the correct procedure and to understand what I was doing.

My main points and reasons for the values that I had chosen were as follows:

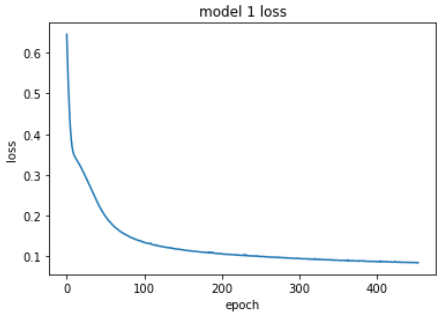
- The two main activation functions I used were relu and sigmoid. From what I learned relu is the most desired and used activation function and sigmoid was once popular but as fallen out of favour. The reason for this is because Relu is easier to use and is much faster which means a lot in neural networks.
- My learning rates were 1e-3 and did mess with these values for a bit, but the result was very mixed and found this to be the most optimal or favourable with determining legendary Pokémon.
- I had my epoch set to 100 initially and it did have the 0.85 range accuracy on four legendries at first and 1 out of the 3 were legendries and had goudra as the fourth and I believe it thought goudra was a legendary due to its high stats.

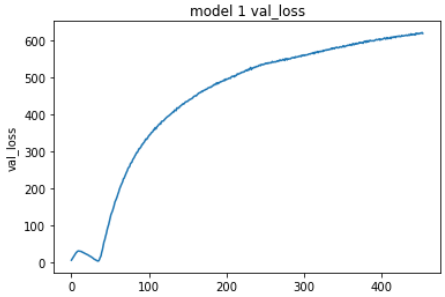
My best Model was Model 1 with high accuracy scores:

```
In [156]: #
ds = np.array(testresults)

ds1 = ds.reshape(400,1)

Gen8Pokemon = np.append(ds1, df2[['Name']],axis=1)
print_stat_greater_than = Gen8Pokemon[Gen8Pokemon[:,0]>0.95 ]# Print pokemon greater than 95% with  names
print(print_stat_greater_than)
```

```
[[0.9698882102966309 'Zacian']
 [0.9698882102966309 'Zamazenta']
 [0.9878839254379272 'Eternatus']]
```

I think it was overfitted a bit and had a high value of epoch of 500 and 32 dense layers.





Part 2

Summary

When I started part 2, I used the link supplied in the project document to find out the best way to load the data and read it.

I separated the data into to sets, one was the training data set and the other I called validation data set, and, in the set, I changed the values such as the seed to my student number, the image size to 128 x 128 and my subset to validation and training.

I set the validation to 0.2 and take 20% of random the images for training.

Why I made the decisions I made

When I built the models, my aim was to try to underfit them on purpose so the predict/accuracy values would not be increasingly high, but the results were not what I expected.

I chose to use relu because it's the fastest to run and was helpful to speed up the process for running the model.

I did the evaluation manually with the predictions towards the evalimages folder to compare and see if it was guessing the correct Pokémon and it was incorrect most of the time, but the image below shows model 2 as my best model was largely due to having lower values like epoch =10 and layer dense= 64

Because in model one I had double the values here.

I was unable to figure out how to implement confusion matrix or precision call but I suspect the reason is because I needed to do a train test split like in part one or I was not using normalising the data either.

```
This image most likely belongs to Pikachu with a 99.97 p
```