Project 2

March 3, 2020

0.1 Base model

- accuracy (3 epoch) = 58.46%
- accuracy (10 epoch) = 61.21%
- accuracy (25 epoch) = 67.19%

0.2 Model 2 (base model + binary crossentropy + some layers)

- accuracy (3 epoch) = 91.24%
- accuracy (10 epoch) = 92.14%
- accuracy (25 epoch) = 93.08%

0.3 Model 3 (base model + much more layers)

- accuracy (3 epoch) = 47.90%
- accuracy (10 epoch) = 66.62%
- accuracy (25 epoch) = 78.82%

0.4 Report:

For this project I used the CIFAR-10 dataset, which is accesed through Keras. I loaded the data in through Keras, and then normalized the inputs. To do this, I took each value and divided it by 255 (which is the max observation.) I had to convert them to floats because they get imported as integers and we wouldn't get the correct data if we kept them as such. Then I converted them into a binary matrix of width 10 (because there are 10 classes.) In the base model, I have a few layers and then evaluate the model. The base model classifies alright (67% for 25 epochs), but my second model classifies much better. I got up to an accuracy of 93% by just adding a few layers, as well as using binary crossentropy instead of categorical crossentropy. I don't think it was ever mentioned that we couldn't use this, but it greatly improved my accuracy. My third model I bascially just threw in a ton of layers and tested to see if it made a difference. It actually helped a lot, getting me to 78% accuracy by just adding some more convolutional layers, dropout layers, and connected layers within the network. I chose my loss functions based on how the data is modeled. In the second one, I am inferring that the prediction does so well because it is a binary loss function, and I have inputs/target values of 0 and 1. Besides that, it was mostly just trial and error.

0.4.1 Used for help:

- https://www.tensorflow.org
- https://keras.io
- https://machinelearningmastey.com