Deep Learning

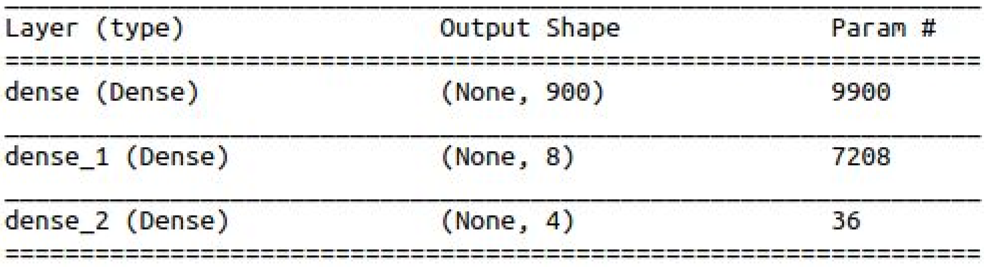
Project 1

- In my first attempt at this project, I used the input as decimal numbers but with that, I was getting accuracy near to 50% in the test set.
- Therefore, I converted the input to binary numbers and with this, I am getting the following accuracies in the train and test data

ACCURACY

epochs	Train	Test
1	0.5172	0.5300
40	0.7386	0.6700
50	0.9399	0.7900
80	0.9989	0.9700
100	1	0.98

MODEL STRUCTURE



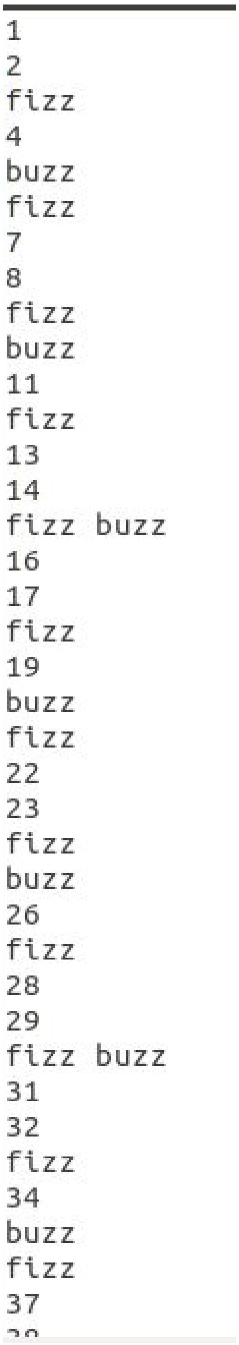
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 900)               9900
_________________________________________________________________
dense_1 (Dense)              (None, 8)                 7208
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 36
=================================================================
```

In the neural network model, I used only 1 hidden layer. Although I experimented by increasing the number of layers but, that was not helping to improve test accuracy further. Also when I increased the number of Epochs. By increasing the number of epochs we can get accuracy equal to 100%. One hidden layer is sufficient in this network because the fizzbuzz problem is very easy and one hidden layer is sufficient to learn the solution.

The reason that we have to convert the input to binary format is that binary representation of a number makes more sense in saying that a number will be divisible by some number or not as it follows a periodic fashion.

Software1.py will generate the output file output1.py
Software2.py will generate the output file output2.py
Tensorflow Version: 2.0.0
Keras version: 2.2.4-tf

SNAPSHOT OF OUTPUT



```
1
2
fizz
4
buzz
fizz
7
8
fizz
buzz
11
fizz
13
14
fizz buzz
16
17
fizz
19
buzz
fizz
22
23
fizz
buzz
26
fizz
28
29
fizz buzz
31
32
fizz
34
buzz
fizz
37
38
```