**1.1 Introduction:**

The project is to compare two problem solving approaches to software development: the logic-based approach (Software 1.0) and the machine learning approach (Software 2.0). It is also designed to quickly gain familiarity with Python and machine learning frameworks.

**1.2 Task:**

We consider the task of FizzBuzz. In this task an integer divisible by 3 is printed as Fizz, and integer divisible by 5 is printed as Buzz. An integer divisible by both 3 and 5 is printed as FizzBuzz. If an integer is not divisible by 3 or 5 or 15, it simply prints the input as output (for this last case, the input number can be classified as Other in Software 2.0, it should then be handled using Software 1.0, which prints the input as output).

**1.3 Software 1.0:**

I have implemented the program in python with a simple if…else structure,

for i in range(0,101):

if(i%15==0):

print("FizzBuzz")

elif(i%5==0):

print("Buzz")

elif(i%3==0):

print("Fizz")

else:

print(i)

The above code loops through the number 0 to 101 and prints Fizz, Buzz, FizzBuzz when the number is divisible 3, 5 and 15 respectively.

**Output:**

PS C:\Users\sassi\Documents\Codes\Python> python sample.py

1,2,Fizz,4,Buzz,Fizz,7,8,Fizz,Buzz,11,Fizz,13,14,FizzBuzz,16,17,Fizz,19,Buzz,Fizz,22,23,Fizz,Buzz,26,Fizz,28,29,FizzBuzz,31,32,Fizz,34,Buzz,Fizz,37,38,Fizz,Buzz,41,Fizz,43,44,FizzBuzz,46,47,Fizz,49,Buzz,Fizz,52,53,Fizz,Buzz,56,Fizz,58,59,FizzBuzz,61,62,Fizz,64,Buzz,Fizz,67,68,Fizz,Buzz,71,Fizz,73,74,FizzBuzz,76,77,Fizz,79,Buzz,Fizz,82,83,Fizz,Buzz,86,Fizz,88,89,FizzBuzz,91,92,Fizz,94,Buzz,Fizz,97,98,Fizz,Buzz

**1.4 Software 2.0:**

In Software 2.0 we want to implement the fizzbuzz problem using machine learning. We have been given two code samples that has the implementation of FizzBuzz. One implementation is in TensorFlow framework and other is in Keras which users TensorFlow to run.

**1.4.1 Observations:**

***Machine Learning:*** It uses computers to predict unknown values through the recognition of patterns in the input data.

***TensorFlow:*** It is an open source, python-friendly library developed by google to implement machine learning faster and easier.

***Keras:*** Keras is a high-level machine learning API that runs on various machine learning framework like TensorFlow.

***Steps implemented in the program:***

Step 1: Create training dataset and testing dataset.

Step 2: Defining a modal that we are going to train.

Step 3: Processing the input data suitable for the modal that we have created.

Step 4: Training the modal that we have created using the input data that has the label for each value (Supervised learning).

Step 5: Testing the modal that we have created using the testing data set and finding the accuracy using the output data generated by the modal

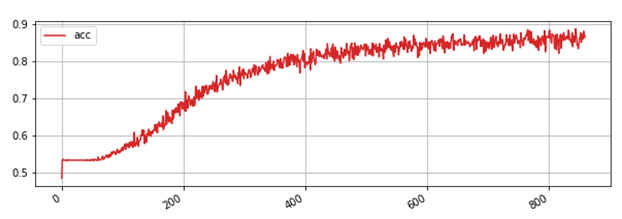
*Test case 1: No changes in the code given*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number of layers** | **Number of nodes in each layer** | **Optimizer** | **Activation function** | **Number of epochs** | **Early stopping** |
| 2 | 256,4 | rmsprop | relu | 10000 | Yes |

Error: 21

Correct: 79

Testing accuracy: 79.0



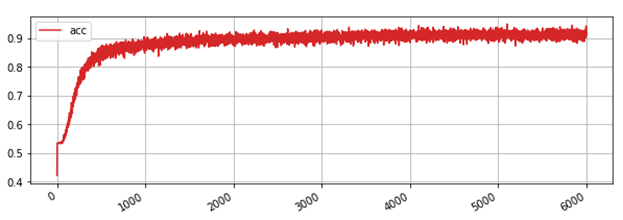
*Test case 2:*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number of layers** | **Number of nodes in each layer** | **Optimizer** | **Activation function** | **Number of epochs** | **Early stopping** |
| 2 | 256,4 | rmsprop | relu | 6000 | No |

Errors: 12

Correct :88

Testing Accuracy: 88.0



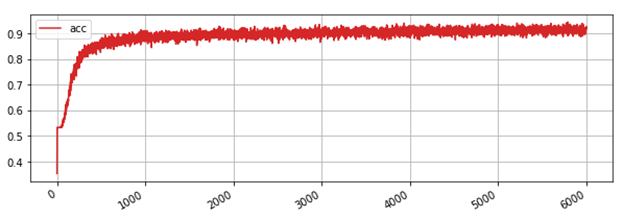
*Test case 3:*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number of layers** | **Number of nodes in each layer** | **Optimizer** | **Activation function** | **Number of epochs** | **Early stopping** |
| 2 | 256,4 | adadelta | relu | 6000 | No |

Errors: 15

Correct :85

Testing Accuracy: 85.0



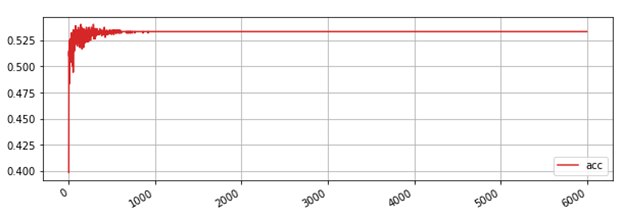
*Test case 4:*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number of layers** | **Number of nodes in each layer** | **Optimizer** | **Activation function** | **Number of epochs** | **Early stopping** |
| 2 | 256,4 | adagrad | sigmoid | 6000 | No |

Errors: 47

Correct :53

Testing Accuracy: 53.0



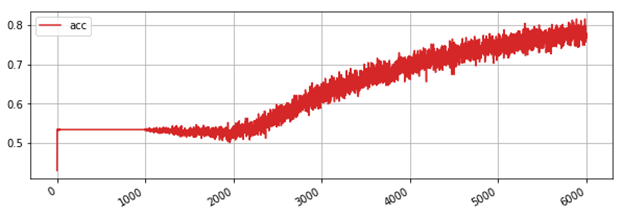
*Test case 5:*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number of layers** | **Number of nodes in each layer** | **Optimizer** | **Activation function** | **Number of epochs** | **Early stopping** |
| 2 | 256,4 | adadelta | tanh | 6000 | No |

Errors: 19

Correct :81

Testing Accuracy: 81.0



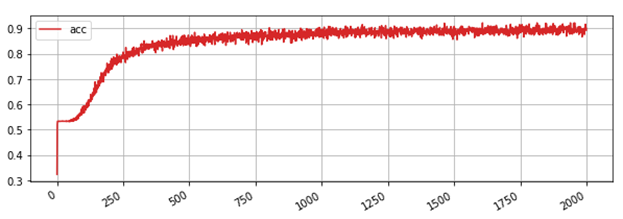
*Test case 6:*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number of layers** | **Number of nodes in each layer** | **Optimizer** | **Activation function** | **Number of epochs** | **Early stopping** |
| 2 | 256,4 | adadelta | relu | 2000 | No |

Errors: 9

Correct :91

Testing Accuracy: 91.0



**1.5 Reference:**

1. <https://towardsdatascience.com>
2. <http://joelgrus.com/2016/05/23/fizz-buzz-in-tensorflow/>
3. <https://www.tensorflow.org/tutorials/>
4. <https://keras.io/>