# Software 1.0 Vs Software 2.0 (FizzBuzz implementation)

## INTRODUCTION

- The objective of the report is to analyze the differences in solving problems using the logical approach (1.0) and the machine learning approach (2.0) by looking at the implementation of the FizzBuzz problem.
- The focus is on understanding the Machine Learning approach

## MODEL DESCRIPTION

### TRADITIONAL MODEL:

- The traditional model uses a simple logical approach to solve the problem. The if else block is implemented to classify a given number into one of the classes.
- If a number is divisible only by 3, it is classified as Fizz
- If a number is divisible only by 5, it is classified as Buzz
- If a number is divisible by both 3 and 5, it is classified as FizzBuzz
- If a number isn't any of these, it's classified as "other".

### MACHINE LEARNING MODEL [1],[2],[3],[4],[5]:

- The machine learning approach is based on training the classification model using data known as training data.
- The information gained by learning from the training data is then used by the network to classify new data which the network typically hasn't seen before. This is known as testing data. (Note: The testing data should not be given to the network during training).
- For this model, we take the number range (101 to 1000) as our training data and number range (1 to 100) as our testing range.
- We use the Keras machine learning library and the functionalities it provides to train the model over the training data and then test it.
- Important terminologies:
- Keras: It is a deep learning library for python which allows neural networks to run efficiently.
- Sequential Model API: Keras API that allows us to add layers in the network sequentially.
- Dense layer: A network layer in which every unit is connected to every unit of the next layer.
- Activation: Keras functionality that allows us to apply an activation function to the output
- Softmax: Type of activation function which takes two parameters, x (an input tensor) and axis (along which normalization is to be applied) and returns a tensor(output)
- Relu: Stands for Rectified Linear Unit. Takes three arguments, x (input tensor), alpha (slope of negative component) and max_value (max value for output) and returns a Relu activation as output

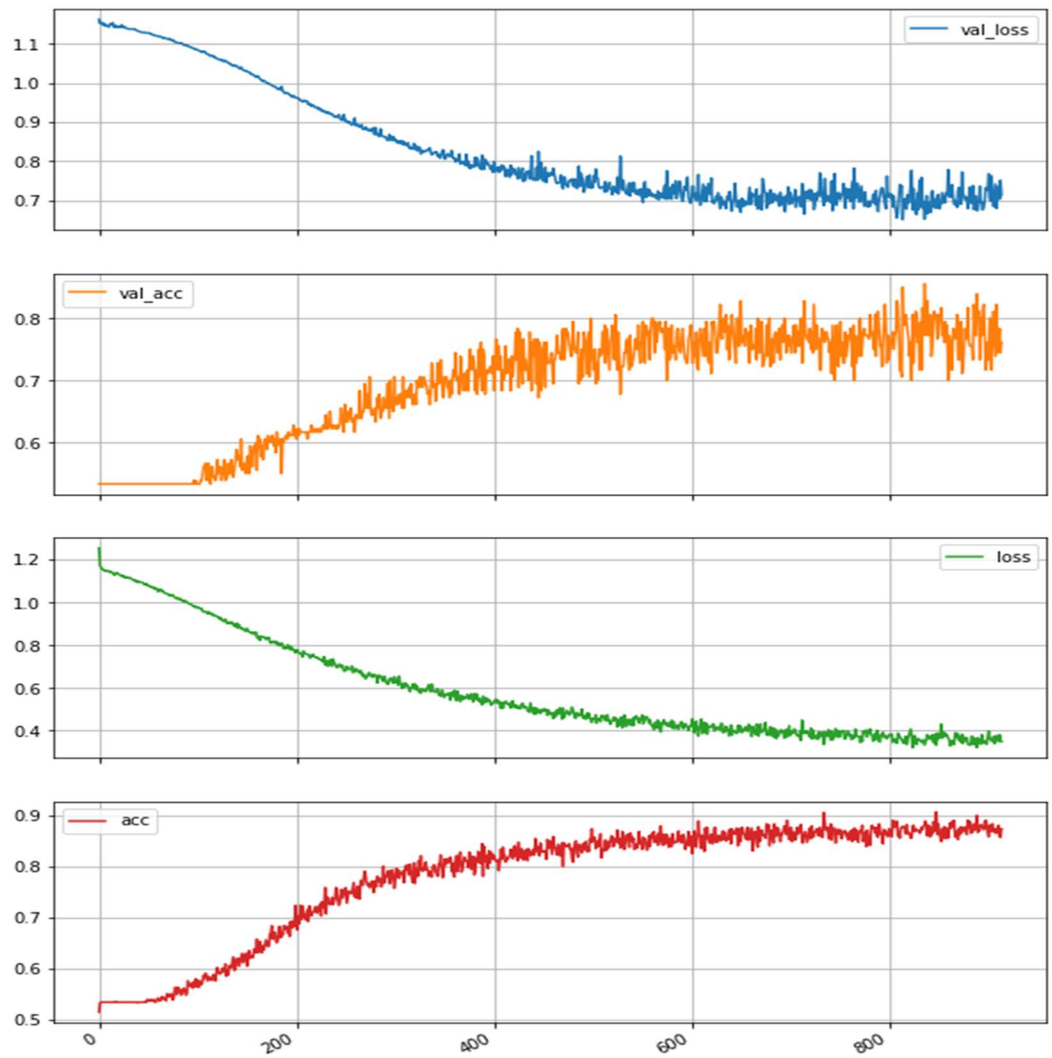
- Dropout: It's a regularization technique that is aimed at reducing overfitting of data by a model. Using dropout, according to the rate set, a fixed number of nodes in a layer are deactivated. Hence the network only relies on the remaining number of nodes for classification.
- Categorical cross entropy: It is used to measure the performance of a model that has an output range from 0 to 1. In this model, it is useful when used with an activation function like softmax.
- Overfitting: It occurs when a model fits the data "too well". This means that for a given training data, the model learns even the noise and fluctuations as actual features. While it may be accurate for the training data, the testing data may not have these features, and hence overfitting leads to lower overall performance.
- Early stopping: It is a callback function which is applied at a particular stage in the process of training to avoid overfitting. It basically determines how many iterations can be run before the model is at the risk of generalization error and overfitting.

### WORKING OF MACHINE LEARNING MODEL:

- Initially, datasets for training and testing are created in the form of lists. These datasets are then filled with the training data as input for the model.
- The dataset obtained is then processed by assigning the input and respective output values from CSV sets to variables "data" and "labels".
- Data is then encoded and stored.
- Model is created which contains the input, output and hidden layers. These layers contain neurons or units which when "fired" perform a computation and pass on the output forward.

## EXPERIMENT

- The results are displayed in the output.csv file in which numbers are classified accordingly as fizz, buzz, Fizzbuzz or other.
- Upon execution of the code blocks, it can be observed that four important graphs are generated (see graphs).
- Validation loss graph which shows the loss (errors) during training.
- Validation accuracy graph which is nothing but the training accuracy.
- Loss graph, which is the losses or errors accumulated while running the testing data. As the number of epochs increase, the overall loss decreases.
- Accuracy graph, which is the depiction of total accuracy for the model per epoch. As the number of epochs increase, the overall accuracy increases.
- After a point of time, the accuracy is more or less constant. This is the point where the callback function is employed to stop training to prevent overfitting.




## RESULTS

Summary of running and testing accuracy after running the code:

Trial 1:

```
Errors: 16  Correct :84
Testing Accuracy: 84.0
```

Trial 2:

```
Errors: 8  Correct :92
Testing Accuracy: 92.0
```

Trial 3:

```
Errors: 12  Correct :88
Testing Accuracy: 88.0
```

Trial 4:

```
Errors: 21  Correct :79
Testing Accuracy: 79.0
```

Trial 5

```
Errors: 22  Correct :78
Testing Accuracy: 78.0
```

Trial 6

```
Errors: 18  Correct :82
Testing Accuracy: 82.0
```

Total number of runs: 6

Mean testing accuracy achieved: (503/6) = 83.833

Variance:

A = $(0.167)^2 + (8.167)^2 + (4.167)^2 + (-4.833)^2 + (-5.833)^2 + (-1.833)^2 = 1164.155$

B = 6

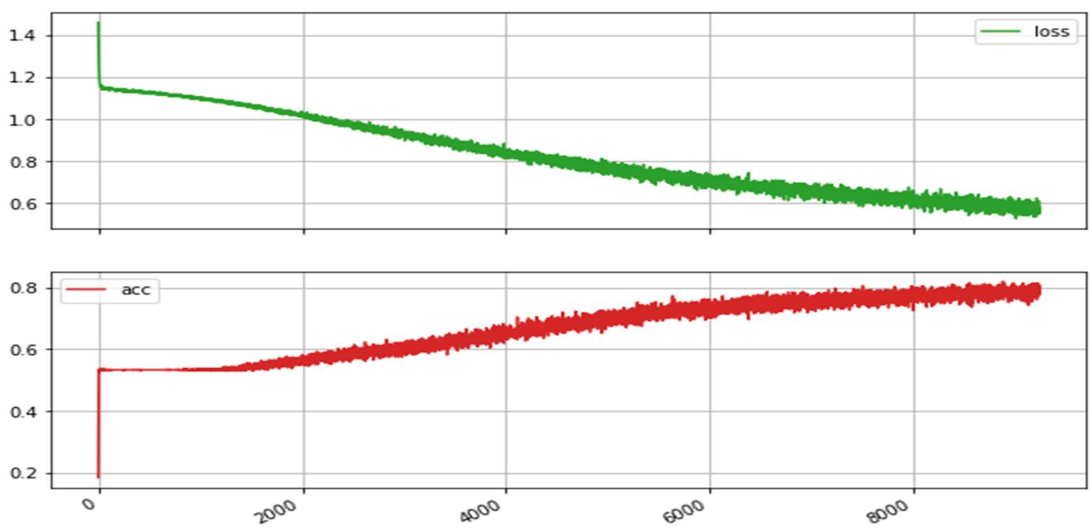
Variance observed (A/B) = 194.025

VARIATION OF PARAMETERS

- Many parameters such as the dropout rate can be changed.
- The optimizer argument can be altered and set to different type. Popular ones include SGD, RMSprop, Adagrad, Adadelta etc.
- When the optimizer argument is changed from RMSprop to SGD (stochastic gradient descent), we observe that:
- The number of epochs for which the model runs has increased. Early stopping occurred much later at the 9229th epoch.
- Testing accuracy was observed to be 76.0 i.e. slightly lower than RMSprop.

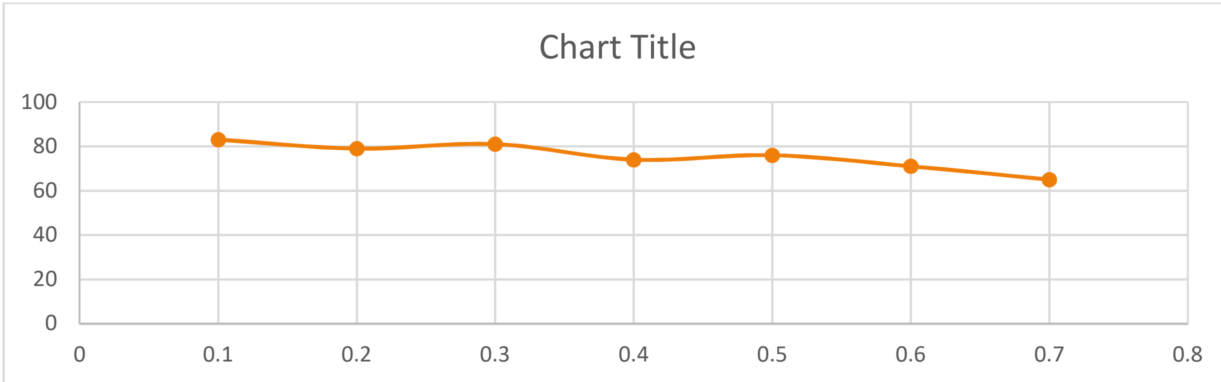
```
Errors: 24  Correct :76
Testing Accuracy: 76.0
```

SGD graph

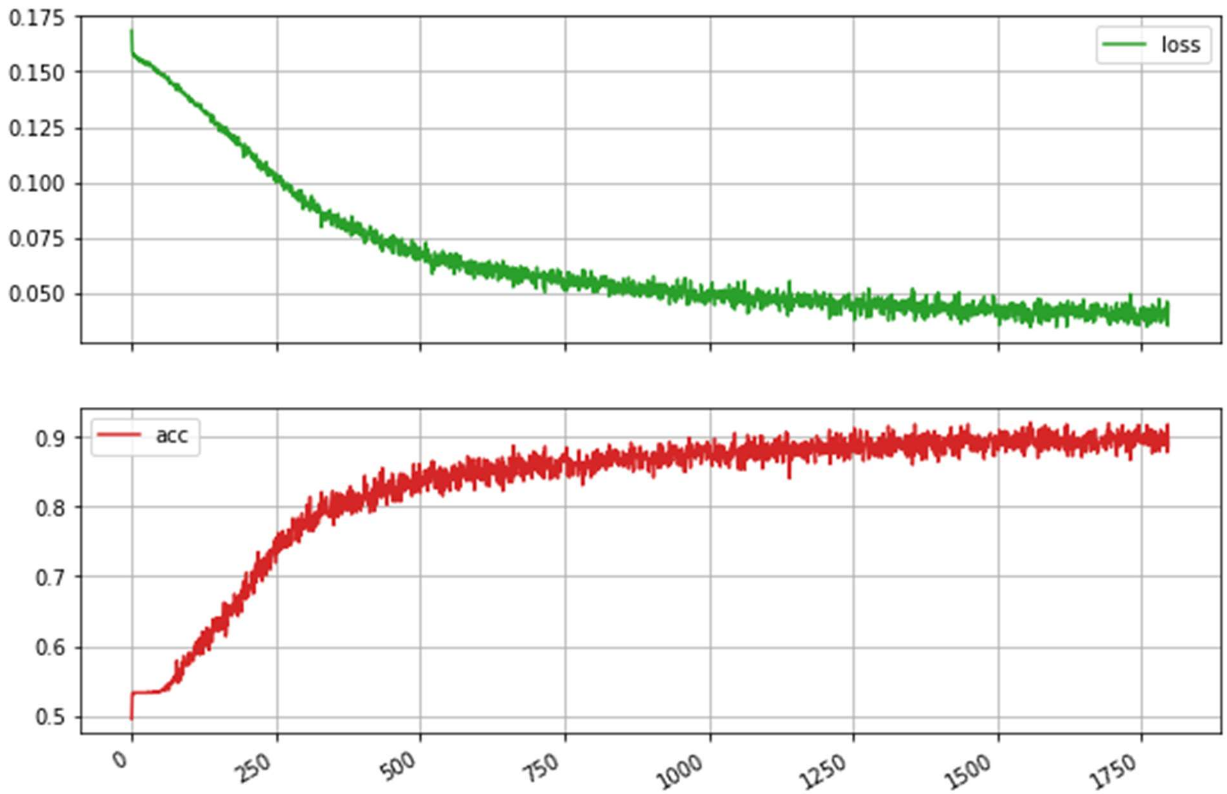



- When the dropout rate was altered between 0.1 and 0.8 in increments of 0.1, the following graph was observed (a).
- When the loss function was changed from categorical cross entropy to “mean squared error”, the following graph was observed (b) along with testing accuracy (c).

(a)




(b)




(c)

```
Errors: 12  Correct :88
Testing Accuracy: 88.0
```

## CONCLUSION

- Software 1.0 approach is the logical approach of solving the classification problem by providing the instructions to the machine.
- On the other hand, Software 2.0 (ML approach) is a different approach to solving the classification problem.
- Rather than providing explicit instructions, the machine is provided with a large set of data for training which it “learns”. It can then be used to classify new data (testing data) which it hasn’t encountered before based on this knowledge.
- Various Hyper parameters such as activation function, loss function, learning rate and dropout rate, examples of which have been demonstrated earlier, provide different results and have a direct impact on the model.
- **A model with an optimal combination of hyperparameters and functions would yield highest accuracy with least errors.**
- In general, functions like Softmax and Relu along with loss functions like mean squared error or cross entropy produce results with high accuracy.
- This approach has potentially unlimited applications in all fields.

## REFERENCES


1. https://keras.io/
2. https://stackoverflow.com/questions/34518656/how-to-interpret-loss-and-accuracy-for-a-machine-learning-model
3. https://www.coursera.org/lecture/ai/sequential-models-in-keras-RBbLP
4. https://www.quora.com/In-Keras-what-is-a-dense-and-a-dropout-layer