Stefan Grulović (20150280)
Computer Science 340
Dr. Alexander Astaras
Spring II 2019

Part C: Project Report

1. Design Requirements

The project is divided in two parts or programs, written in Python, for F1 racing results (Part A) and Training an Artificial Neural network (Part B). **Part A** is to build a program which can read a file that contains Formula 1 racing results and then filter, search and or calculate statistics from the data. **Part B** was to build a Deterministic Artificial Neural Network consisting of binary state neurons, analogue state weights and a variable topology in order to classify a unary encoded number whether it is even or odd and whether is larger or equal to 4.
The design requirements could be summed up in the form of menu which they present to user and they are:

Part A (F1):

Menu with **5** options and Quit:

1. Read and Display contents from "partA_input_data.txt"
    - *needs "partA_input_data.txt
2. Search contents by limit of laps
    - *Displayed ascending
3. Calculate the average lap time per race
    - *Adds 7th column to new file "partA_output_data.txt"
4. Sort by field, ascending or descending
    - *Needs file from Choice 3
5. Calculate the total average lap time per driver across all races
    - *Displayed as pop-up windows (x-axis: driver names. y-axis: average lap time in minutes)
6. QUIT

Part B (ANN):

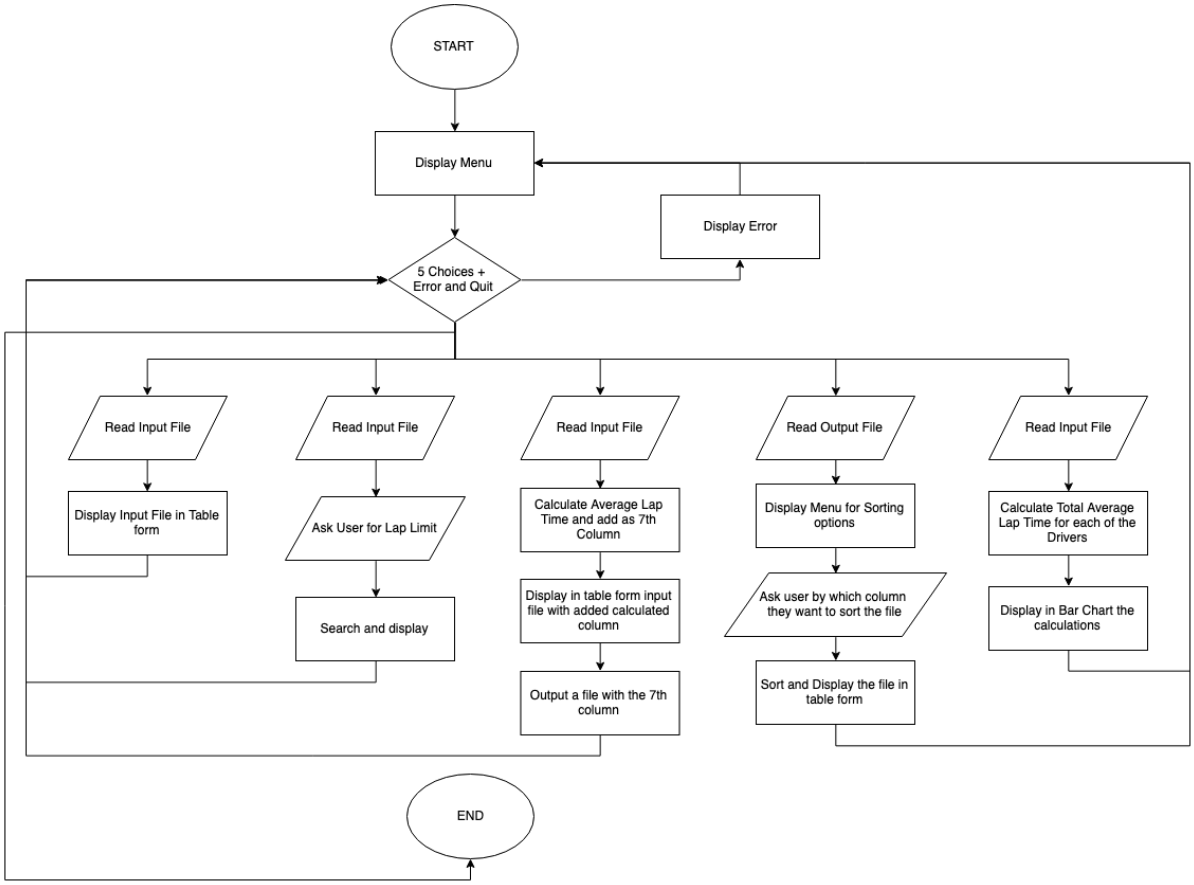
Menu with **4** options and Quit:

1. Enter size of the hidden layer
2. Initiate training pass
3. Classify test data and display results
4. Display training results graphics
5. QUIT

The code should be able to perform all the tasks which it presents itself in the menu. However, the functions **tanh** and **relu** are not used in the code as they don't produce the correct output.

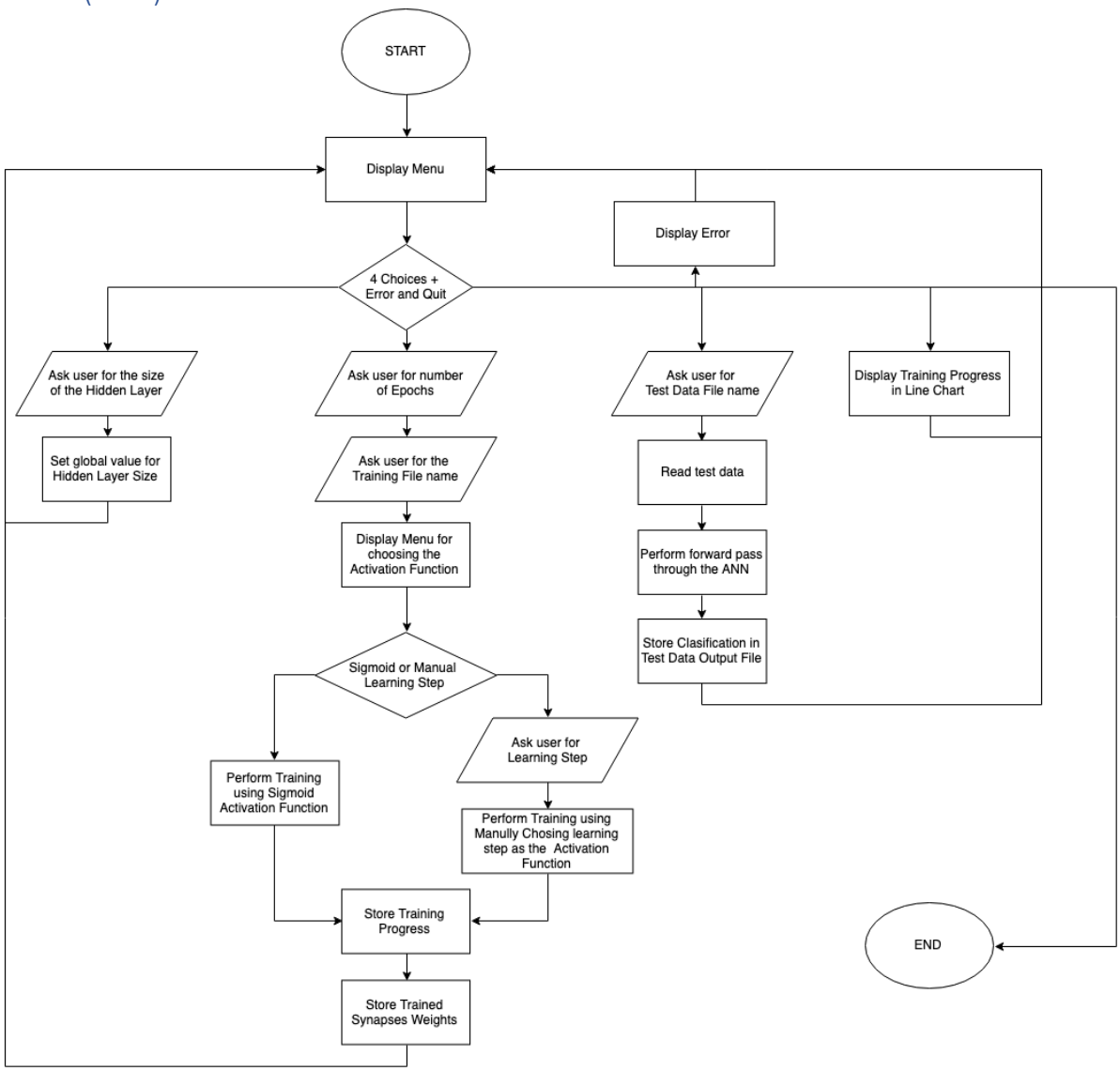
2. Top-level flowchart diagrams

Part A (F1):



Part A top-level flowchart diagram.

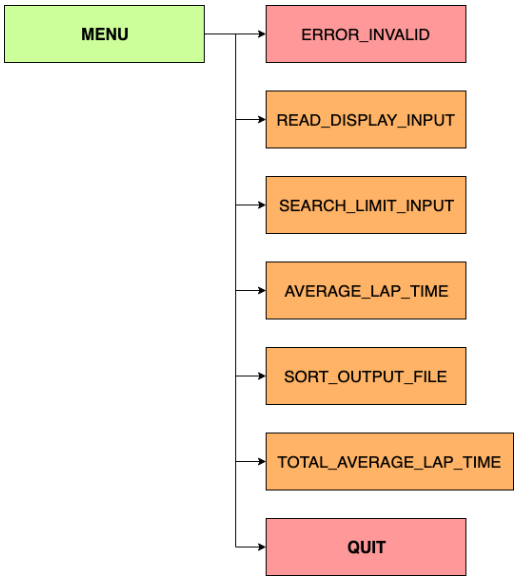
Part B (ANN):



Part B top-level flowchart diagram.

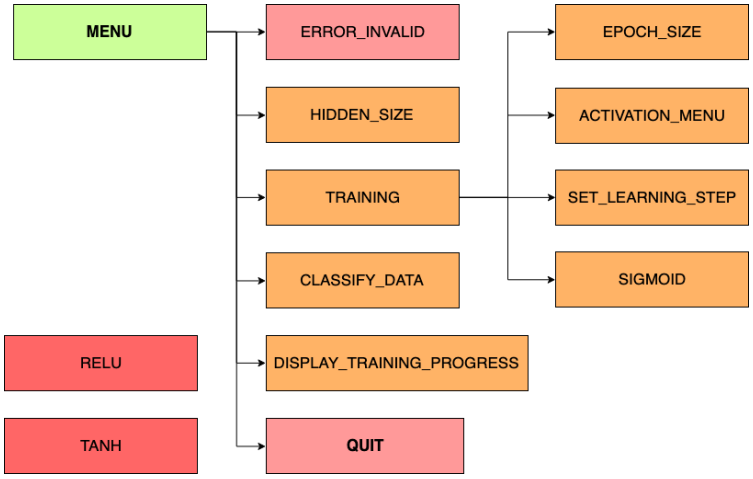
3. Function interdependency diagrams

Part A (F1):



Part A function interdependency diagram.

Part B (ANN):



Part B function interdependency diagram

4. Input and output data samples

Raw files are included alongside this report.

Part A (F1):

Input (partA_input_data.txt)

This is the format of the data table; it is in CSV form and is needed in order for the program to work.

```
GRAND_PRIX,DATE,WINNER,CAR,LAPS,TIME
Australia,25-Mar-18,Sebastian_Vettel,FERRARI,58,01:29:33.000
Bahrain,08-Apr-18,Sebastian_Vettel,FERRARI,57,01:32:01.940
China,15-Apr-18,Daniel_Ricciardo,RED_BULL,56,01:35:36.380
Azerbaijan,29-Apr-18,Lewis_Hamilton,MERCEDES,51,01:43:44.291
Spain,13-May-18,Lewis_Hamilton,MERCEDES,66,01:35:29.972
Monaco,27-May-18,Daniel_Ricciardo,RED_BULL,78,01:42:54.807
Canada,10-Jun-18,Sebastian_Vettel,FERRARI,68,01:28:31.377
France,24-Jun-18,Lewis_Hamilton,MERCEDES,53,01:30:11.385
Austria,01-Jul-18,Max_Verstappen,RED_BULL,71,01:21:56.024
Great_Britain,08-Jul-18,Sebastian_Vettel,FERRARI,52,01:27:29.784
Germany,22-Jul-18,Lewis_Hamilton,MERCEDES,67,01:32:29.845
Hungary,29-Jul-18,Lewis_Hamilton,MERCEDES,70,01:37:16.427
Belgium,26-Aug-18,Sebastian_Vettel,FERRARI,44,01:23:34.476
Italy,02-Sep-18,Lewis_Hamilton,MERCEDES,53,01:16:54.484
Singapore,16-Sep-18,Lewis_Hamilton,MERCEDES,61,01:51:11.611
Russia,30-Sep-18,Lewis_Hamilton,MERCEDES,53,01:27:25.181
Japan,07-Oct-18,Lewis_Hamilton,MERCEDES,53,01:27:17.062
United_States,21-Oct-18,Kimi_Raikkonen,FERRARI,56,01:34:18.643
Mexico,28-Oct-18,Max_Verstappen,RED_BULL,71,01:38:28.851
Brazil,11-Nov-18,Lewis_Hamilton,MERCEDES,71,01:27:09.066
Abu_Dhabi,25-Nov-18,Lewis_Hamilton,MERCEDES,55,01:39:40.382
```

Output (partB_output_data.txt)

This is the output file which the program produces, and it is the input file with another column added called AVG_LAP_TIME which contains the average lap time of the given row and its data.

```
GRAND_PRIX,DATE,WINNER,CAR,LAPS,TIME,AVG_LAP_TIME
Australia,25-Mar-18,Sebastian_Vettel,FERRARI,58,01:29:33.000,0:01:32.637931
Bahrain,08-Apr-18,Sebastian_Vettel,FERRARI,57,01:32:01.940,0:01:36.859649
China,15-Apr-18,Daniel_Ricciardo,RED_BULL,56,01:35:36.380,0:01:42.428571
Azerbaijan,29-Apr-18,Lewis_Hamilton,MERCEDES,51,01:43:44.291,0:02:02.039216
Spain,13-May-18,Lewis_Hamilton,MERCEDES,66,01:35:29.972,0:01:26.803030
Monaco,27-May-18,Daniel_Ricciardo,RED_BULL,78,01:42:54.807,0:01:19.153846
Canada,10-Jun-18,Sebastian_Vettel,FERRARI,68,01:28:31.377,0:01:18.102941
France,24-Jun-18,Lewis_Hamilton,MERCEDES,53,01:30:11.385,0:01:42.094340
Austria,01-Jul-18,Max_Verstappen,RED_BULL,71,01:21:56.024,0:01:09.239437
Great_Britain,08-Jul-18,Sebastian_Vettel,FERRARI,52,01:27:29.784,0:01:40.942308
Germany,22-Jul-18,Lewis_Hamilton,MERCEDES,67,01:32:29.845,0:01:22.820896
Hungary,29-Jul-18,Lewis_Hamilton,MERCEDES,70,01:37:16.427,0:01:23.371429
Belgium,26-Aug-18,Sebastian_Vettel,FERRARI,44,01:23:34.476,0:01:53.954545
Italy,02-Sep-18,Lewis_Hamilton,MERCEDES,53,01:16:54.484,0:01:27.056604
Singapore,16-Sep-18,Lewis_Hamilton,MERCEDES,61,01:51:11.611,0:01:49.360656
Russia,30-Sep-18,Lewis_Hamilton,MERCEDES,53,01:27:25.181,0:01:38.962264
Japan,07-Oct-18,Lewis_Hamilton,MERCEDES,53,01:27:17.062,0:01:38.811321
United_States,21-Oct-18,Kimi_Raikkonen,FERRARI,56,01:34:18.643,0:01:41.035714
Mexico,28-Oct-18,Max_Verstappen,RED_BULL,71,01:38:28.851,0:01:23.211268
Brazil,11-Nov-18,Lewis_Hamilton,MERCEDES,71,01:27:09.066,0:01:13.647887
Abu_Dhabi,25-Nov-18,Lewis_Hamilton,MERCEDES,55,01:39:40.382,0:01:48.727273
```

Part B (ANN):

Input

training_data.txt

This is the data which the program uses and needs to be able to train. Each row of the data is divided in three parts the Input layer consisting of different Unary Encoded Numbers, whether the number is even (0) or odd (1) and whether the number is smaller or equal to 4 (0) or bigger than 4 (1).

```
1000000000,1,0
1100000000,0,0
1110000000,1,0
1111000000,0,0
1111100000,1,1
1111110000,0,1
1111111000,1,1
1111111100,0,1
1111111110,1,1
```

test_data.txt

This is the test data which the program uses in order to check whether the ANN has learned to classify properly the unary encoded numbers.

```
1 0 0 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0
1 1 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0
1 1 1 1 1 1 0 0 0 0
1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 0 0 0 0
1 1 1 0 0 0 0 0 0 0
1 1 1 1 1 0 0 0 0 0
…
```

Output

training_progress.txt

This notes down the epoch and its mean error of training total of 100 times evenly distributed across total epochs that the program runs.

```
0 , 0.48009345225003625
50 , 0.2158637114706782
100 , 0.06981098975283308
150 , 0.04660600786079344
200 , 0.03674654289135713
…
```

synapse_matrix_0.txt or synapse_matrix_1.txt

These are the layer 0 (input) and layer 1 (hidden) synapse weights which are used to go through the test data with a single forward pass in order to classify the test data based on what it learned previously. The files vary in size this is due to the user choosing the size of the hidden layer.

```
0.13965516168660796 0.6871007567190435 -0.836028508289862 0.6682221965904329
-2.7398948774400345 -1.111005459113073 -0.8859926312906192 -0.8352370083572894
-2.5963234619853255 1.3411670486555705 0.16920892982294092 0.08259535147922543
```

test_data_output.txt

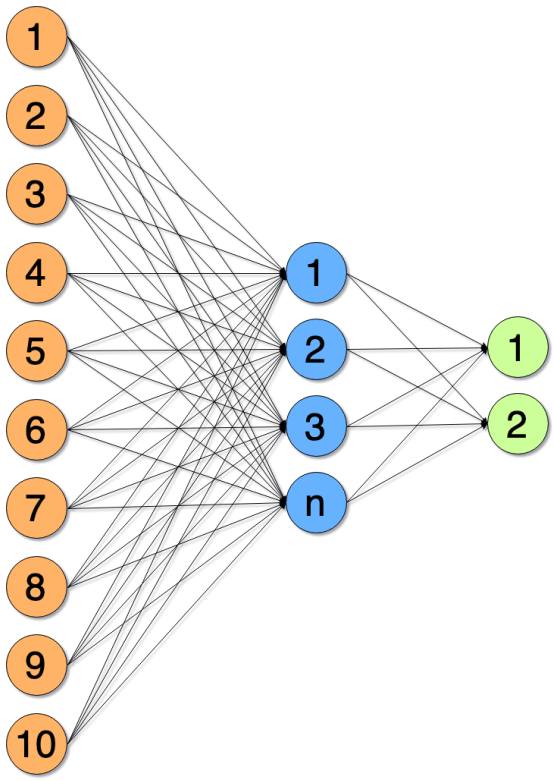
This is the data which is produced from classifying the test data from what the ANN has learned and stored.

```
1 0 0 0 0 0 0 0 0 0 , 1 , 0
1 1 0 0 0 0 0 0 0 0 , 0 , 0
1 0 0 0 0 0 0 0 0 0 , 1 , 0
1 1 1 1 1 0 0 0 0 0 , 1 , 1
1 1 1 0 0 0 0 0 0 0 , 1 , 0
1 1 1 1 1 1 0 0 0 0 , 0 , 1
1 1 1 1 0 0 0 0 0 0 , 0 , 0
1 1 1 1 1 1 0 0 0 0 , 0 , 1
1 1 1 0 0 0 0 0 0 0 , 1 , 0
1 1 1 1 1 0 0 0 0 0 , 1 , 1
1 1 1 1 1 0 0 0 0 0 , 1 , 1
…
```

5. ANN diagram

Part B (ANN):

The design of the Artificial Neural Network diagram consists out of three layers the **input layer**, **hidden layer** and **output layer**. The input layer has a fixed size of 10 and they are the 10 binary numbers of the unary encoded numbers from 0-9. The hidden layer varies in sizes this is due to the user choosing its size. Lastly, the output layer has a size of two and they are the two outputs the first being whether the number is even or odd and second whether the number is bigger or smaller than 4.



The ANN diagram.

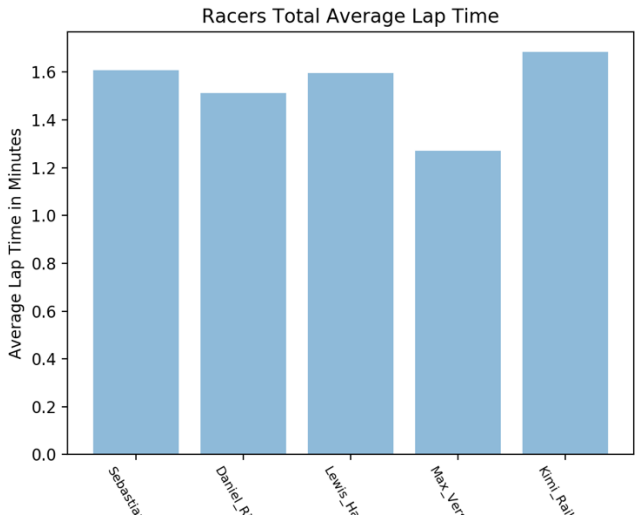
6. Program Part B Experimentation

The program allows various combination of the neural network by allowing the choosing of the hidden layer size, number of epochs as well as the activation function for the learning. This way there are various ways of setting up the artificial neural network and the most optimum is the hidden layer of size **3**, number of epochs of **100** and the **sigmoid** as the choice for the activation function.

The program could be expanded by making the function **tanh** and **relu** work in the code. Where tanh is Hyperbolic Tangent function and relu is Rectified Linear units. Besides those two the program can be expanded by adding other activation functions which would provide a more efficient learning as the adjusting of synapses weights is better. [1][2][3]

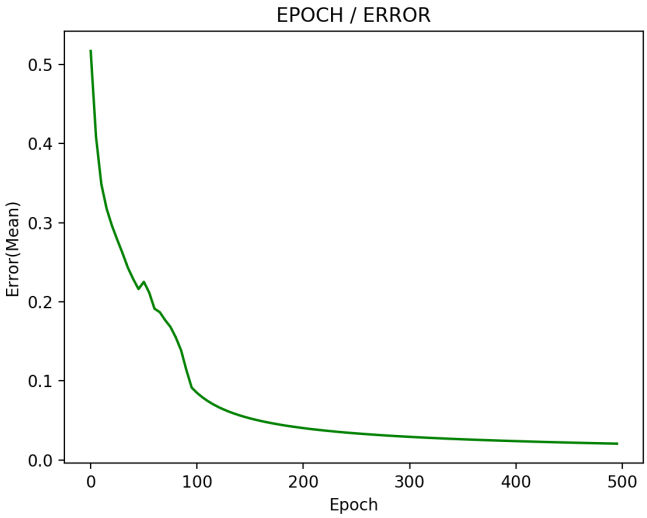
7. Graphs displaying processed data results

Part A (F1):



Bar chart which shows the total average lap time of each of the racers from the data.

Part B (ANN):



Line chart which shows the mean error of every 10th epoch.

8. Conclusion

The skills learning during the making of the project was coding and making a small deterministic artificial neural network in Python.

9. Bibliography

[1] geva, “7 Types of Neural Network Activation Functions: How to Choose? -.” [Online]. Available: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/. [Accessed: 10-Jun-2019].

[2] A. S. V, “Understanding Activation Functions in Neural Networks,” *Medium*, 30-Mar-2017. .

[3] A. S. Walia, “Activation functions and it’s types-Which is better?,” *Towards Data Science*, 29-May-2017. [Online]. Available: https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f. [Accessed: 10-Jun-2019].