TERM PROJECT:
ARTIFICIAL NEURAL NETWORK – NUMBER OF CLASSES PREDICTION

BASALLOTE, Lawrence
CARANDO, Paolo Creas
LORENA, Neil John
ZABAT, Juan Paolo

I. INTRODUCTION

Artificial Neural Network (ANN) is a computational tool modelled on the interconnection of neurons which exchanges information. It is used to estimate functions based on the number of sample data within a network using mathematical functions. ANN will be designed according to its application, such as data prediction through learning process.

The project will be implementing ANN to create a solution for an existing problem in the university. The problem is, sometimes, the number of classes offered during enlistment week is not enough to accommodate the students; resulting to students experiencing difficulty to apply for petition of classes.

The idea of the project is to build a network that can predict the number of sections of the succeeding courses based on the students' grade of the pre-requisite course. By applying the artificial neural network, the machine must come up an output of the needed sections to accommodate the students.

II. PROCEDURES

The input data of the system are the grade of students' in all of the sections of the pre-requisite course. Fig. 1 below is an example of the input data for the system. The first row data are the pre-defined quantity of the input data. The "50" value are the number of samples. The second row will be the inputs. The "100" value are the number of students' grades for all the sections of the pre-requisite course. And the last row is the predicted output. The "4" value is the output; it is the predicted number of classes for the succeeding course. The first index in the output is equivalent to 0 classes, succeeding are 1, 2, 3 classes depends where binary "1" logic is outputted.

```
50  100 4
0.0 1.0 4.0 0.0 2.5 0.0 3.0 0.0 1.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 2.5 0.0
0.0 2.5 0.0 3.5 0.0 0.0 0.0 2.5 0.0
0 1 0 0
```

Fig. 1. Sample of input data

Fig. 2 shows the initialization of the number of input, number of output, number of layers, and the number of hidden neurons. It also initializes the iteration which has the value of 1000. For the number input we assigned the value 100 for every sample on the data. Number of output indicates the number of sections that must be available. The expected output will be 0, 1, 2, or 3. The index with a value of 1 will determine the exact output.

```
fann_type *calc_out;
const unsigned int num_input = 100;
const unsigned int num_output = 4;
const unsigned int num_layers = 3;
const unsigned int num_neurons_hidden = 150;
const float desired_error = (const float) 0;
const unsigned int max_epochs = 1000;
const unsigned int epochs_between_reports = 10;
struct fann *ann;
struct fann_train_data *data;
```

Fig. 2. Initialization of Variables

Fig. 3 shows the setting of parameter for training the program. The new.data file was used to train the program. Here we set the extremes to “1” and sigmoid so that the output that generated is either “0” or “1”. The learning algorithm is also set here to “Quickprop” because it is the most advanced among other algorithm.

```
data = fann_read_train_from_file("new.data");

fann_set_activation_steepness_hidden(ann, 1);
fann_set_activation_steepness_output(ann, 1);

fann_set_activation_function_hidden(ann, FANN_SIGMOID);
fann_set_activation_function_output(ann, FANN_SIGMOID);

fann_set_train_stop_function(ann, FANN_STOPFUNC_BIT);
fann_set_bit_fail_limit(ann, 0.01f);

fann_set_training_algorithm(ann, FANN_TRAIN_QUICKPROP);
```

Fig. 3. Setting of Parameters

Fig. 4 shows the command to execute the training of the input. After training the program, the validation is important to test its accuracy. For validation the file new_validate.data was used.

```
fann_train_on_data(ann, data, max_epochs, epochs_between_reports, desired_error);

    data = fann_read_train_from_file("new_validate.data");

printf("Testing network. %f\n", fann_test_data(ann, data));
```

Fig. 4. Training & Validation

The calc_out is used to compute for the probabilities of each output. Also the highest probability from the predicted and actual is stored to variables “max_output” and “max_expected”.

```
calc_out = fann_run(ann, data->input[i]);

max_output = 0;
for(j = 1; j < 4; j++)
{
    if(calc_out[j] > calc_out[j-1])
        max_output = j;
}

max_expected = 0;
for(k = 1; k < 4; k++)
{
    if(data->output[i][k] > data->output[i][k-1])
        max_expected = k;
}
```

Fig. 5. Probability Computation

The computed probabilities from the predicted and actual are used to obtain the accuracy of the network.

```
if(max_expected == max_output)
    correct++;
else
    wrong++;

printf("\nCorrect: %d\nWrong: %d\n", correct, wrong);
printf("Accuracy: %f\n", (float) correct/(correct+wrong)*100);
```

Fig. 6. Accuracy: Computation & Display

After the accuracy of the network has been determined, the network used is then saved into the directory. This is so that there is no more need to generate a new network for testing. The program only has to call the saved network.

```
fann_save(ann, "new_float.net");

decimal_point = fann_save_to_fixed(ann, "new_fixed.net");
fann_save_train_to_fixed(data, "new_fixed.data", decimal_point);
```

Fig. 7. Saving of Neural Network

In new_test.c, the neural network saved in the training data is accessed using the code below. Therefore, the neural network used for testing is the same as the neural network used in the training before.

```
#ifdef FIXEDFANN
    ann = fann_create_from_file("new_fixed.net");
#else
    ann = fann_create_from_file("new_float.net");
#endif
```

Fig. 8. Loading of Neural Network

After loading the neural network, the testing data is read into the program. Since we are using floating point, we are using the second code in Figure 9. After reading the testing data, accuracy is computed using the same algorithm as the training code.

```
#ifdef FIXEDFANN
    data = fann_read_train_from_file("new_fixed.data");
#else
    data = fann_read_train_from_file("new_test.data");
#endif
```

Fig. 9. Reading of Testing Data

III. RESULTS & DISCUSSION

Epoch	Error	Epoch	Error
1	7.50E+09	510	87959450
10	2.04E+09	520	64996108
20	2.50E+09	530	26537243
30	1.47E+09	540	11645131
40	1.02E+09	550	1.57E+08
50	1.96E+09	560	1.73E+09
60	1.25E+09	570	4.16E+08
70	1.23E+09	580	5.89E+08
80	8.65E+08	590	1.26E+09
90	1.23E+09	600	1.48E+09
100	7.78E+08	610	3.88E+08
110	1.22E+09	620	3.10E+08
120	8.66E+08	630	8.54E+08
130	9.56E+08	640	4.19E+08
140	6.83E+08	650	5.12E+08
150	6.99E+08	660	1.00E+09
160	7.39E+08	670	9.45E+08
170	5.62E+08	680	1.09E+09
180	1.08E+09	690	3.76E+08
190	1.02E+09	700	1.51E+09
200	9.08E+08	710	4.82E+08
210	4.23E+08	720	1.94E+09
220	7.35E+08	730	4.22E+08
230	9.21E+08	740	8.96E+08
240	5.89E+08	750	2.70E+08
250	4.82E+08	760	2.51E+08
260	4.29E+08	770	1.38E+08
270	8.01E+08	780	1.72E+08
280	1.69E+09	790	58183349
290	4.17E+08	800	28438098
300	6.31E+08	810	10924669
310	1.80E+09	820	3454640
320	7.68E+08	830	2491807
330	4.89E+08	840	2328226
340	1.57E+09	850	2240801
350	1.16E+09	860	2175665
360	1.59E+09	870	2122682
370	1.28E+09	880	2077306
380	4.26E+08	890	2036876
390	8.48E+08	900	1999821
400	4.60E+08	910	1965622
410	6.29E+08	920	1933399
420	8.59E+08	930	1902509
430	5.86E+08	940	1872447
440	3.96E+08	950	1842801
450	3.02E+08	960	1813245
460	1.28E+09	970	1783553
470	1.20E+09	980	1753650
480	3.59E+08	990	1723650
490	1.68E+08	1000	1693875
500	1.13E+08		

Training Results:
Test: 0
P0: 0.000420
P1: 0.999530
P2: 0.000581
P3: 0.000000
Expected no. of sections: 1
Output no. of sections: 1

Test: 1
P0: 0.001364
P1: 0.000105
P2: 0.999751
P3: 0.000000
Expected no. of sections: 2
Output no. of sections: 2

Test: 2
P0: 0.000725
P1: 0.996536
P2: 0.005544
P3: 0.000000
Expected no. of sections: 1
Output no. of sections: 1

Test: 3
P0: 0.001682
P1: 0.000175
P2: 0.970303
P3: 0.001525
Expected no. of sections: 2
Output no. of sections: 2

Test: 4
P0: 0.001646
P1: 0.000132
P2: 0.993295
P3: 0.000102
Expected no. of sections: 2
Output no. of sections: 2

Test: 5
P0: 0.001514
P1: 0.000395
P2: 0.986510
P3: 0.000228
Expected no. of sections: 2
Output no. of sections: 2

Test: 6
P0: 0.001335
P1: 0.000446
P2: 0.977217
P3: 0.000010
Expected no. of sections: 2
Output no. of sections: 2

Test: 7
P0: 0.001373
P1: 0.001402
P2: 0.990222
P3: 0.000001
Expected no. of sections: 2
Output no. of sections: 2

Test: 8
P0: 0.001471
P1: 0.000132
P2: 0.993355
P3: 0.000015
Expected no. of sections: 2
Output no. of sections: 2

Test: 9
P0: 0.000640
P1: 1.000000
P2: 0.059647
P3: 0.000000
Expected no. of sections: 1
Output no. of sections: 1

Test: 10
P0: 0.001356
P1: 0.000142
P2: 0.999483
P3: 0.000000
Expected no. of sections: 2
Output no. of sections: 2

Test: 11
P0: 0.000771
P1: 0.999837

P2: 0.000589
P3: 0.000000
Expected no. of sections: 1
Output no. of sections: 1

Test: 12
P0: 0.001783
P1: 0.000152
P2: 0.981694
P3: 0.001180
Expected no. of sections: 2
Output no. of sections: 2

Test: 13
P0: 0.001698
P1: 0.000120
P2: 0.985285
P3: 0.000465
Expected no. of sections: 2
Output no. of sections: 2

Test: 14
P0: 0.000682
P1: 0.988468
P2: 0.000725
P3: 0.003893
Expected no. of sections: 1
Output no. of sections: 3
Saving network.

Correct: 14
Wrong: 1
Accuracy: 93.333333

Testing Results:
Test: 0
P0: 0.001362
P1: 0.000104
P2: 0.999761
P3: 0.000000
Expected no. of sections: 2
Output no. of sections: 2

Test: 1
P0: 0.000798
P1: 0.961568
P2: 0.011256
P3: 0.000000
Expected no. of sections: 1
Output no. of sections: 1

Test: 2
P0: 0.001682
P1: 0.000175
P2: 0.970303
P3: 0.001525
Expected no. of sections: 2
Output no. of sections: 2

Test: 3
P0: 0.001344
P1: 0.000434
P2: 0.982525
P3: 0.000006
Expected no. of sections: 2
Output no. of sections: 2

Test: 4
P0: 0.001365
P1: 0.001594
P2: 0.990423
P3: 0.000001
Expected no. of sections: 2
Output no. of sections: 2

Test: 5
P0: 0.001467
P1: 0.000133
P2: 0.991883
P3: 0.000020
Expected no. of sections: 2
Output no. of sections: 2

Test: 6
P0: 0.000640
P1: 1.000000
P2: 0.059647
P3: 0.000000
Expected no. of sections: 1
Output no. of sections: 1

Test: 7

P0: 0.001362
P1: 0.000155
P2: 0.999454
P3: 0.000000
Expected no. of sections: 2
Output no. of sections: 2

Test: 8
P0: 0.000771
P1: 0.999837
P2: 0.000589
P3: 0.000000
Expected no. of sections: 1
Output no. of sections: 1

Test: 9
P0: 0.001798
P1: 0.000163
P2: 0.979560
P3: 0.001605
Expected no. of sections: 2
Output no. of sections: 2

Test: 10
P0: 0.001691
P1: 0.000135
P2: 0.979547
P3: 0.000870
Expected no. of sections: 2
Output no. of sections: 2

Test: 11
P0: 0.000781
P1: 0.915065
P2: 0.003819
P3: 0.005071
Expected no. of sections: 1
Output no. of sections: 3

Test: 12
P0: 0.001637
P1: 0.000363
P2: 0.977191
P3: 0.000423
Expected no. of sections: 2
Output no. of sections: 2

Test: 13
P0: 0.001704
P1: 0.000132
P2: 0.988745
P3: 0.000256
Expected no. of sections: 2
Output no. of sections: 2

Test: 14
P0: 0.001726
P1: 0.000171
P2: 0.988400
P3: 0.000418
Expected no. of sections: 2
Output no. of sections: 2

Test: 15
P0: 0.001704
P1: 0.000110
P2: 0.990021
P3: 0.000238
Expected no. of sections: 2
Output no. of sections: 2

Test: 16
P0: 0.001725
P1: 0.000114
P2: 0.988019
P3: 0.000360
Expected no. of sections: 2
Output no. of sections: 2

Test: 17
P0: 0.001576
P1: 0.000286
P2: 0.993720
P3: 0.000122
Expected no. of sections: 2
Output no. of sections: 2

Test: 18
P0: 0.000674
P1: 0.984810
P2: 0.000004
P3: 0.003137

Expected no. of sections: 1
Output no. of sections: 3

Test: 19
P0: 0.000641
P1: 0.999991
P2: 0.000401
P3: 0.000000
Expected no. of sections: 1
Output no. of sections: 1

Test: 20
P0: 0.000589
P1: 0.999999
P2: 0.000554
P3: 0.000000
Expected no. of sections: 1
Output no. of sections: 1

Test: 21
P0: 0.000646
P1: 1.000000
P2: 0.000131
P3: 0.000000
Expected no. of sections: 1
Output no. of sections: 1

Test: 22
P0: 0.000576
P1: 0.999988
P2: 0.000120
P3: 0.000000
Expected no. of sections: 1
Output no. of sections: 1

Test: 23
P0: 0.001053
P1: 0.000966
P2: 0.636848
P3: 0.031888
Expected no. of sections: 2
Output no. of sections: 2

Test: 24
P0: 0.000992
P1: 0.001886
P2: 0.000534
P3: 0.999949
Expected no. of sections: 2
Output no. of sections: 3

Test: 25
P0: 0.001375
P1: 0.001392
P2: 0.012707
P3: 0.999147
Expected no. of sections: 2
Output no. of sections: 3

Test: 26
P0: 0.000974
P1: 0.001947
P2: 0.000435
P3: 0.999961
Expected no. of sections: 2
Output no. of sections: 3

Test: 27
P0: 0.001653
P1: 0.000120
P2: 0.994802
P3: 0.000069
Expected no. of sections: 2
Output no. of sections: 2

Test: 28
P0: 0.000978
P1: 0.357578
P2: 0.218610
P3: 0.000000
Expected no. of sections: 2
Output no. of sections: 1

Test: 29
P0: 0.001410
P1: 0.000163
P2: 0.999709
P3: 0.000000
Expected no. of sections: 2
Output no. of sections: 2

Test: 30

P0: 0.001641
P1: 0.000931
P2: 0.973764
P3: 0.000190
Expected no. of sections: 2
Output no. of sections: 2

Test: 31
P0: 0.001671
P1: 0.000740
P2: 0.059958
P3: 0.999479
Expected no. of sections: 3
Output no. of sections: 3

Test: 32
P0: 0.001361
P1: 0.001137
P2: 0.027201
P3: 0.998071
Expected no. of sections: 3
Output no. of sections: 3

Test: 33
P0: 0.001189
P1: 0.000362
P2: 0.087036
P3: 0.982344
Expected no. of sections: 3
Output no. of sections: 3

Test: 34
P0: 0.001356
P1: 0.001136
P2: 0.029317
P3: 0.997747
Expected no. of sections: 3
Output no. of sections: 3

Test: 35
P0: 0.001702
P1: 0.000102
P2: 0.990840
P3: 0.000222
Expected no. of sections: 2
Output no. of sections: 2

Test: 36
P0: 0.001706
P1: 0.000115
P2: 0.985396
P3: 0.000513
Expected no. of sections: 2
Output no. of sections: 2

Test: 37
P0: 0.001663
P1: 0.000729
P2: 0.054577
P3: 0.999556
Expected no. of sections: 3
Output no. of sections: 3

Test: 38
P0: 0.001197
P1: 0.001286
P2: 0.005695
P3: 0.999555
Expected no. of sections: 3
Output no. of sections: 3

Test: 39
P0: 0.001336
P1: 0.001130
P2: 0.043348
P3: 0.994450
Expected no. of sections: 3
Output no. of sections: 3

Test: 40
P0: 0.001118
P1: 0.001620
P2: 0.002510
P3: 0.999826
Expected no. of sections: 3
Output no. of sections: 3

Test: 41
P0: 0.001641
P1: 0.000667
P2: 0.059127
P3: 0.999402

Expected no. of sections: 3
Output no. of sections: 3

Test: 42
P0: 0.001708
P1: 0.000107
P2: 0.986314
P3: 0.000498
Expected no. of sections: 2
Output no. of sections: 2

Test: 43
P0: 0.001723
P1: 0.000670
P2: 0.122723
P3: 0.997704
Expected no. of sections: 3
Output no. of sections: 3

Test: 44
P0: 0.001319
P1: 0.001145
P2: 0.054295
P3: 0.991091
Expected no. of sections: 3
Output no. of sections: 3

Test: 45
P0: 0.001638
P1: 0.000658
P2: 0.063699
P3: 0.999281
Expected no. of sections: 3
Output no. of sections: 3

Test: 46
P0: 0.001349
P1: 0.001134
P2: 0.035063
P3: 0.996683
Expected no. of sections: 3
Output no. of sections: 3

Test: 47
P0: 0.001371
P1: 0.000969
P2: 0.039191
P3: 0.996486
Expected no. of sections: 3
Output no. of sections: 3

Test: 48
P0: 0.001123
P1: 0.001995
P2: 0.000734
P3: 0.999943
Expected no. of sections: 3
Output no. of sections: 3

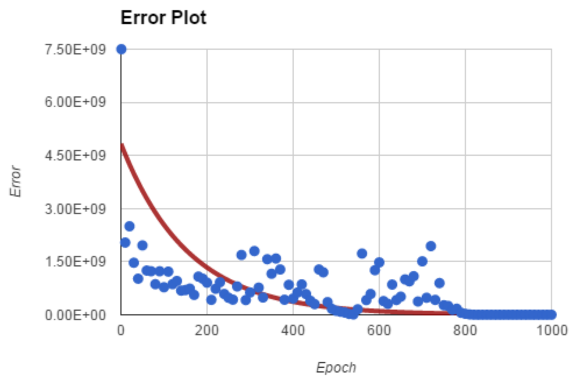
Test: 49
P0: 0.001098
P1: 0.001041
P2: 0.022218
P3: 0.997367
Expected no. of sections: 3
Output no. of sections: 3

Correct: 44
Wrong: 6
Accuracy: 88.000000
Cleaning up.

Confusion Matrix						
Training		Predicted				
		0	1	2	3	T
Actual	0	0	0	0	0	0
	1	0	4	0	1	5
	2	0	0	10	0	10
	3	0	0	0	0	0
	T	0	4	10	1	15

Confusion Matrix						
Testing		Predicted				
		0	1	2	3	T
Actual	0	0	0	0	0	0

	1	0	7	0	2	9
	2	0	1	21	3	25
	3	0	0	0	16	16
	T	0	8	21	21	50



IV. CONCLUSION

Our group concluded that the neural network created was able to obtained 88% accuracy predicting the number of classes needed for the succeeding course. Improvements can be added to the program including the function that will terminate the training operation once the optimum MSE value is obtained. Also, the group recommend enhancing the learning rate by using additional advanced parameters and including more training data samples to the training process to achieve a higher accuracy.

V. BIBLIOGRAPHY

[1] Doc.ic.ac.uk, 'Neural Networks', 2015. [Online]. Available: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#What is a Neural Network.

[2] A. Sampson, 'uwsampa/nntune', GitHub, 2014. [Online]. Available: https://github.com/uwsampa/nntune/tree/master/FANN-2.2.0-Source.