Chaithanya Chikkannaswamy

SUID : 387781563

Evolutionary Machine Learning – Mid-Term Exam

Modify HW1 or HW2 or HW3 using any one of the following three ideas:

A. There are many small demes (sub-populations), connected using a wrap-around grid topology; periodically, each deme exports two members to its four neighbors.

B. The mutation rate (or step size or similar parameter) depends on fitness, being higher for "worse" individuals in the current population.

C. Most individuals in the population remain dormant, not participating in reproduction nor being killed off. They "wake up" and participate only when the best fitness ceases to improve; they sleep again when best fitness begins to improve.

Instead of accuracy, maximize the fitness function:*(N1 + N2)/(1+ a N2 + b N1),* where

N1=number of data points from class1,

N2=number of data points from class 2,

a=number of data points from class1 that are placed in class2 by the model, and

b=number of data points from class2 that are placed in class1 by the model.

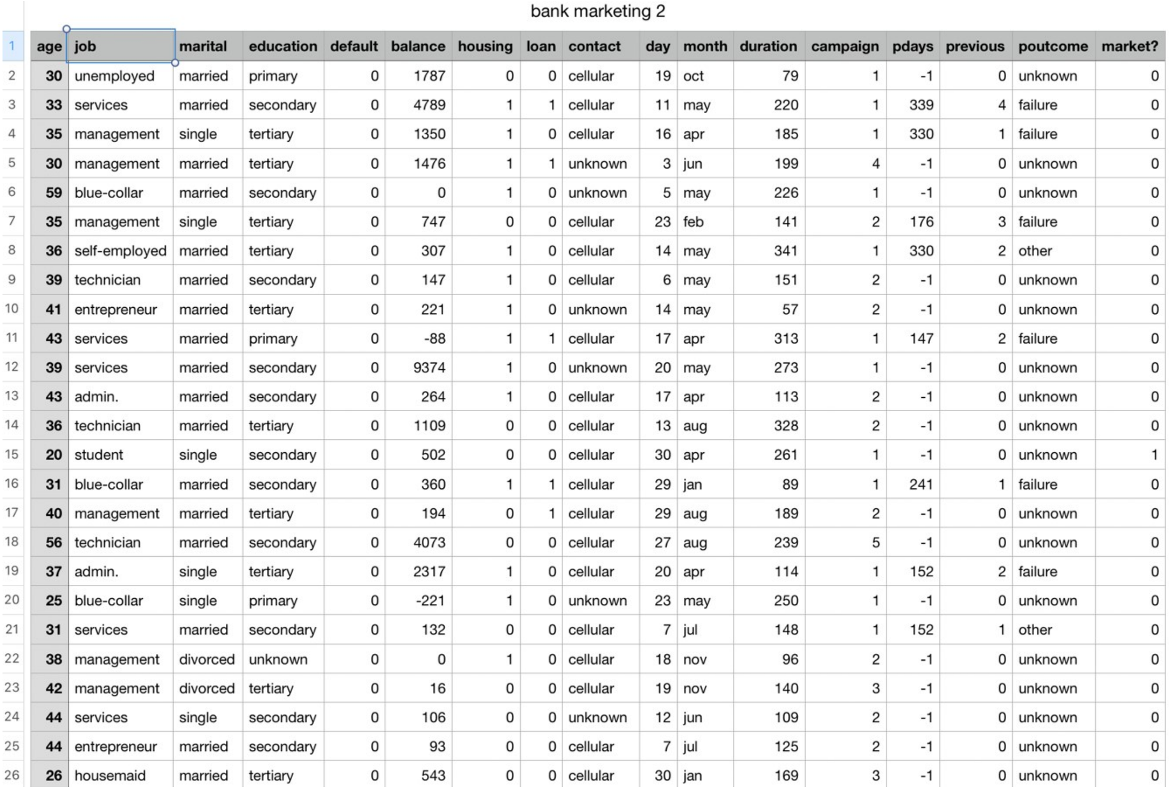
Link:
https://www.kaggle.com/chaithanya96/bankmarketing

The bank marketing dataset is the csv file with users details used to predict the marketing decision. The decision yes or no is represented with binary numbers 1 and 0. The sample columns included are:

1. 'age',
2. 'job',
3. 'marital',
4. 'education',
5. 'default',
6. 'balance',
7. 'housing',
8. 'loan',
9. 'contact',
10. 'day',
11. 'month',
12. 'duration',
13. 'campaign',

14. 'pdays',
15. 'previous',
16. 'poutcome',
17. 'market?'

The screenshot of the sample dataset is attached below:



bank marketing 2

1	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	market?
2	30	unemployed	married	primary	0	1787	0	0	cellular	19	oct	79	1	-1	0	unknown	0
3	33	services	married	secondary	0	4789	1	1	cellular	11	may	220	1	339	4	failure	0
4	35	management	single	tertiary	0	1350	1	0	cellular	16	apr	185	1	330	1	failure	0
5	30	management	married	tertiary	0	1476	1	1	unknown	3	jun	199	4	-1	0	unknown	0
6	59	blue-collar	married	secondary	0	0	1	0	unknown	5	may	226	1	-1	0	unknown	0
7	35	management	single	tertiary	0	747	0	0	cellular	23	feb	141	2	176	3	failure	0
8	36	self-employed	married	tertiary	0	307	1	0	cellular	14	may	341	1	330	2	other	0
9	39	technician	married	secondary	0	147	1	0	cellular	6	may	151	2	-1	0	unknown	0
10	41	entrepreneur	married	tertiary	0	221	1	0	unknown	14	may	57	2	-1	0	unknown	0
11	43	services	married	primary	0	-88	1	1	cellular	17	apr	313	1	147	2	failure	0
12	39	services	married	secondary	0	9374	1	0	unknown	20	may	273	1	-1	0	unknown	0
13	43	admin.	married	secondary	0	264	1	0	cellular	17	apr	113	2	-1	0	unknown	0
14	36	technician	married	tertiary	0	1109	0	0	cellular	13	aug	328	2	-1	0	unknown	0
15	20	student	single	secondary	0	502	0	0	cellular	30	apr	261	1	-1	0	unknown	1
16	31	blue-collar	married	secondary	0	360	1	1	cellular	29	jan	89	1	241	1	failure	0
17	40	management	married	tertiary	0	194	0	1	cellular	29	aug	189	2	-1	0	unknown	0
18	56	technician	married	secondary	0	4073	0	0	cellular	27	aug	239	5	-1	0	unknown	0
19	37	admin.	single	tertiary	0	2317	1	0	cellular	20	apr	114	1	152	2	failure	0
20	25	blue-collar	single	primary	0	-221	1	0	unknown	23	may	250	1	-1	0	unknown	0
21	31	services	married	secondary	0	132	0	0	cellular	7	jul	148	1	152	1	other	0
22	38	management	divorced	unknown	0	0	1	0	cellular	18	nov	96	2	-1	0	unknown	0
23	42	management	divorced	tertiary	0	16	0	0	cellular	19	nov	140	3	-1	0	unknown	0
24	44	services	single	secondary	0	106	0	0	unknown	12	jun	109	2	-1	0	unknown	0
25	44	entrepreneur	married	secondary	0	93	0	0	cellular	7	jul	125	2	-1	0	unknown	0
26	26	housemaid	married	tertiary	0	543	0	0	cellular	30	jan	169	3	-1	0	unknown	0

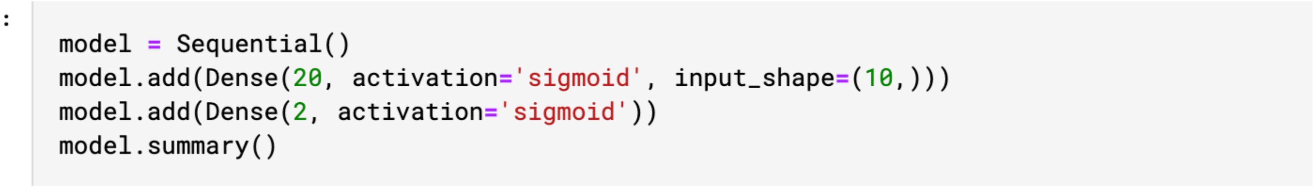
Code execution steps:

Genetic Algorithm

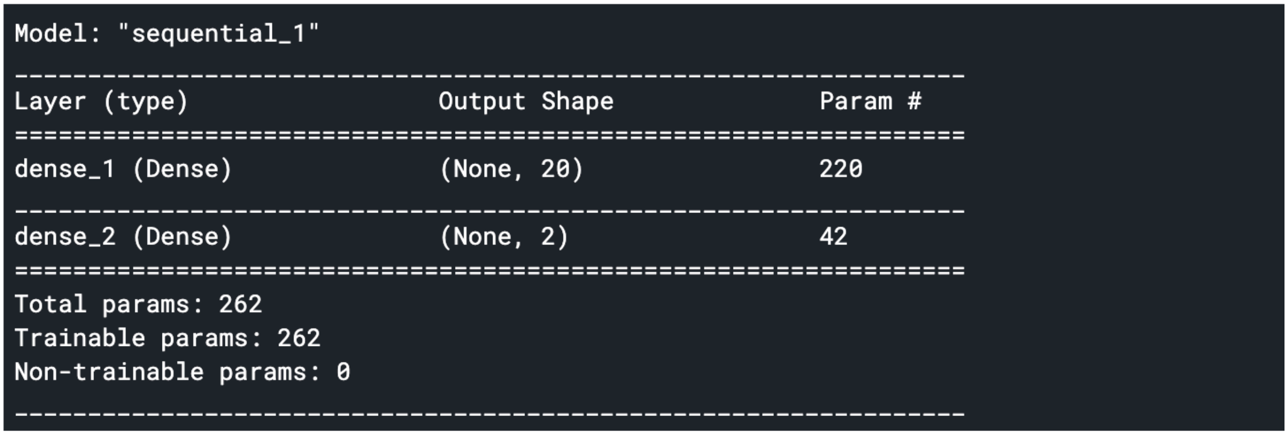
1. GeneticNeuralNetwork class is defined by specifying the sequential model as the argument. If there are no weights specified, then one hidden layer and one output layer will be created by taking the sigmoid activation. The Genetic Neural Network is being created in such manner.
2. Different methods have been defined for mutation and forward propagation.
3. Maximum fitness is being set as 0.9 and for the number of generations as 250 till which

   the program will be executed.

4. The obtained Max fitness is 0.88.



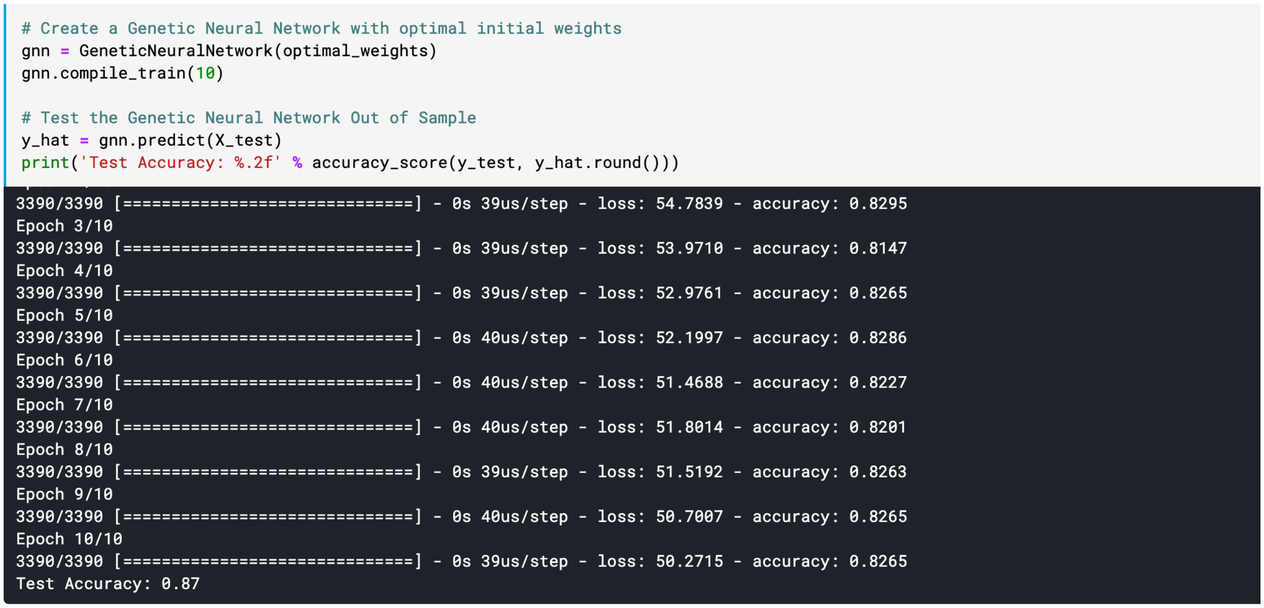
```
:
model = Sequential()
model.add(Dense(20, activation='sigmoid', input_shape=(10,)))
model.add(Dense(2, activation='sigmoid'))
model.summary()
```



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 20)                220
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 42
=================================================================
Total params: 262
Trainable params: 262
Non-trainable params: 0
_________________________________________________________________
```

In this project I have tried to implement the 2nd option in the question i.e. The mutation rate (or step size or similar parameter) depends on fitness, being higher for “worse” individuals in the current population.

The mutation function in the project has been modified according to the given condition to obtain the accurate result. The screenshot is attached below:



```
# Create a Genetic Neural Network with optimal initial weights
gnn = GeneticNeuralNetwork(optimal_weights)
gnn.compile_train(10)

# Test the Genetic Neural Network Out of Sample
y_hat = gnn.predict(X_test)
print('Test Accuracy: %.2f' % accuracy_score(y_test, y_hat.round()))
```

```
3390/3390 [==============================] - 0s 39us/step - loss: 54.7839 - accuracy: 0.8295
Epoch 3/10
3390/3390 [==============================] - 0s 39us/step - loss: 53.9710 - accuracy: 0.8147
Epoch 4/10
3390/3390 [==============================] - 0s 39us/step - loss: 52.9761 - accuracy: 0.8265
Epoch 5/10
3390/3390 [==============================] - 0s 40us/step - loss: 52.1997 - accuracy: 0.8286
Epoch 6/10
3390/3390 [==============================] - 0s 40us/step - loss: 51.4688 - accuracy: 0.8227
Epoch 7/10
3390/3390 [==============================] - 0s 40us/step - loss: 51.8014 - accuracy: 0.8201
Epoch 8/10
3390/3390 [==============================] - 0s 39us/step - loss: 51.5192 - accuracy: 0.8263
Epoch 9/10
3390/3390 [==============================] - 0s 40us/step - loss: 50.7007 - accuracy: 0.8265
Epoch 10/10
3390/3390 [==============================] - 0s 39us/step - loss: 50.2715 - accuracy: 0.8265
Test Accuracy: 0.87
```

Conclusion:

Genetic Neural Network resulted in a test accuracy of **0.89** and the accuracy after maximizing the fitness function and implementing the changes for mutation function, the accuracy is given as **0.87**

References

https://towardsdatascience.com/neural-net-from-scratch-using-numpy-71a31f6e3675
https://machinelearningmastery.com/implement-backpropagation-algorithm-scratch-python/