

Project: Human Detection

- Niranjan Ketkar

a.

File names:

- Source code: HumanDetector.py
- Feature descriptor files for all test images are saved with their original names and .txt extension
- Normalized gradient magnitude images for all test images are saved with their original names and with .png extension

b.

How to compile and run the program:

Program can be run using following command on terminal

Terminal\$ python HumanDetector.py data

data is command line argument, which is path to the data folder where 4 subdirectories are present containing training and testing positive and negative data respectively.

Dependencies:

i. Matplotlib 2.2.3 with PIL 5.3.0

ii. numpy 1.14.2

iii. Python 2.7

Test Image	Output Value	Classification
crop_000010b	0.89735413	Human
crop001008b	0.91184144	Human
crop001028a	0.98806161	Human
crop001045b	0.5311662	Human
crop001047b	0.96414898	Human
00000053a_cut	0.23096039	Not Human
00000062a_cut	0.29727027	Not Human
00000093a_cut	0.02106758	Not Human
no_person__no_bike_213_cut	0.26802477	Not Human
no_person__no_bike_247_cut	0.16605472	Not Human

Any other comments:

- As we increase the hidden layer size, neural network trains very well, but for normal computers this computation is very heavy and I was not able to set its size beyond 1000(dual core cpu!)

- Also, learning rate decay code is there in source file, if you want to implement that functionality, just comment out 2 lines and learning rate will be halved every 500 epochs.
- I tried to make code as generic as possible and arguments are there for every function which directly set the required hyperparameters or any other properties of program

e.g.

```
def __init__(self, architecture = (7524, 250, 1), epoch_count = 200, learning_rate = 0.01)
```

here we can easily change architecture of our network

- Output I got with 1000 hidden layer size: (was able to try this only once, as it is computationally heavy)

Training started ...hold your breath!!!

```
('epoch No: 0/1000', 'Error Average for this epoch', 0.12536862878744753)
('epoch No: 1/1000', 'Error Average for this epoch', 0.12485203863236811)
('epoch No: 2/1000', 'Error Average for this epoch', 0.12435909149091189)
('epoch No: 3/1000', 'Error Average for this epoch', 0.1238746687661835)
('epoch No: 4/1000', 'Error Average for this epoch', 0.12337391699706765)
('epoch No: 5/1000', 'Error Average for this epoch', 0.12286852978964824)
('epoch No: 6/1000', 'Error Average for this epoch', 0.12234071195248734)
('epoch No: 7/1000', 'Error Average for this epoch', 0.12179274467424099)
('epoch No: 8/1000', 'Error Average for this epoch', 0.12119715839713763)
('epoch No: 9/1000', 'Error Average for this epoch', 0.12057650769112768)
```

.....

.....

.....

.....

.....

```
('epoch No: 988/1000', 'Error Average for this epoch', 8.663215340465777e-05)
('epoch No: 989/1000', 'Error Average for this epoch', 8.652354701427308e-05)
('epoch No: 990/1000', 'Error Average for this epoch', 8.641440033678529e-05)
('epoch No: 991/1000', 'Error Average for this epoch', 8.630349723804485e-05)
('epoch No: 992/1000', 'Error Average for this epoch', 8.619870370397605e-05)
('epoch No: 993/1000', 'Error Average for this epoch', 8.608866748204168e-05)
('epoch No: 994/1000', 'Error Average for this epoch', 8.598243872022442e-05)
('epoch No: 995/1000', 'Error Average for this epoch', 8.587392246213398e-05)
('epoch No: 996/1000', 'Error Average for this epoch', 8.576608255538604e-05)
('epoch No: 997/1000', 'Error Average for this epoch', 8.566034754013362e-05)
('epoch No: 998/1000', 'Error Average for this epoch', 8.555398030130949e-05)
('epoch No: 999/1000', 'Error Average for this epoch', 8.544651911967948e-05)
('Predicted Human Probability: [[0.23096039]]', 'Actual Probability: [0]')
('Predicted Human Probability: [[0.29727027]]', 'Actual Probability: [0]')
```

('Predicted Human Probability: [[0.02106758]], 'Actual Probability: [0]')
 ('Predicted Human Probability: [[0.26802477]], 'Actual Probability: [0]')
 ('Predicted Human Probability: [[0.16605472]], 'Actual Probability: [0]')
 ('Predicted Human Probability: [[0.91184144]], 'Actual Probability: [1]')
 ('Predicted Human Probability: [[0.98806161]], 'Actual Probability: [1]')
 ('Predicted Human Probability: [[0.5311662]], 'Actual Probability: [1]')
 ('Predicted Human Probability: [[0.96414898]], 'Actual Probability: [1]')
 ('Predicted Human Probability: [[0.89735413]], 'Actual Probability: [1]')
 100.0% prediction accuracy of the model on test data.

With 250 hidden layer size I got following results:

('epoch No: 993/1000', 'Error Average for this epoch', 8.59835797452388e-05)
 ('epoch No: 994/1000', 'Error Average for this epoch', 8.587804504025026e-05)
 ('epoch No: 995/1000', 'Error Average for this epoch', 8.576708534319634e-05)
 ('epoch No: 996/1000', 'Error Average for this epoch', 8.565888614403863e-05)
 ('epoch No: 997/1000', 'Error Average for this epoch', 8.554925972704237e-05)
 ('epoch No: 998/1000', 'Error Average for this epoch', 8.544272419664292e-05)
 ('epoch No: 999/1000', 'Error Average for this epoch', 8.533518268507353e-05)
 ('data/test_negative/00000053a_cut.bmpPredicted Human Probability: [[0.21076931]], 'Actual Probability: [0]')
 ('data/test_negative/00000062a_cut.bmpPredicted Human Probability: [[0.28868618]], 'Actual Probability: [0]')
 ('data/test_negative/00000093a_cut.bmpPredicted Human Probability: [[0.02212532]], 'Actual Probability: [0]')
 ('data/test_negative/no_person__no_bike_213_cut.bmpPredicted Human Probability: [[0.25249639]], 'Actual Probability: [0]')
 ('data/test_negative/no_person__no_bike_247_cut.bmpPredicted Human Probability: [[0.16560308]], 'Actual Probability: [0]')
 ('data/test_positive/crop001008b.bmpPredicted Human Probability: [[0.91465841]], 'Actual Probability: [1]')
 ('data/test_positive/crop001028a.bmpPredicted Human Probability: [[0.98951439]], 'Actual Probability: [1]')
 ('data/test_positive/crop001045b.bmpPredicted Human Probability: [[0.50376211]], 'Actual Probability: [1]')
 ('data/test_positive/crop001047b.bmpPredicted Human Probability: [[0.96819798]], 'Actual Probability: [1]')
 ('data/test_positive/crop_000010b.bmpPredicted Human Probability: [[0.89976787]], 'Actual Probability: [1]')
 100.0% prediction accuracy of the model on test data.

- So we can see that even though our results improve after increasing hidden layer size, that improvement is not as significant if we compare it with computational cost.

- Graph of training for 250 hidden layer size for 1000 epochs:
loss vs epochs graph

(loss values are rescaled on graph), but we can see that there is no significant use after 400 epochs of training.

•

