Machine Learning: Special Tasks

Shashi Nagarajan

2021-01-03

1: Variable selection with randomized LASSO

Please find below the table showing the requested probabilities $\hat{\Pi}_k^{\lambda}$:

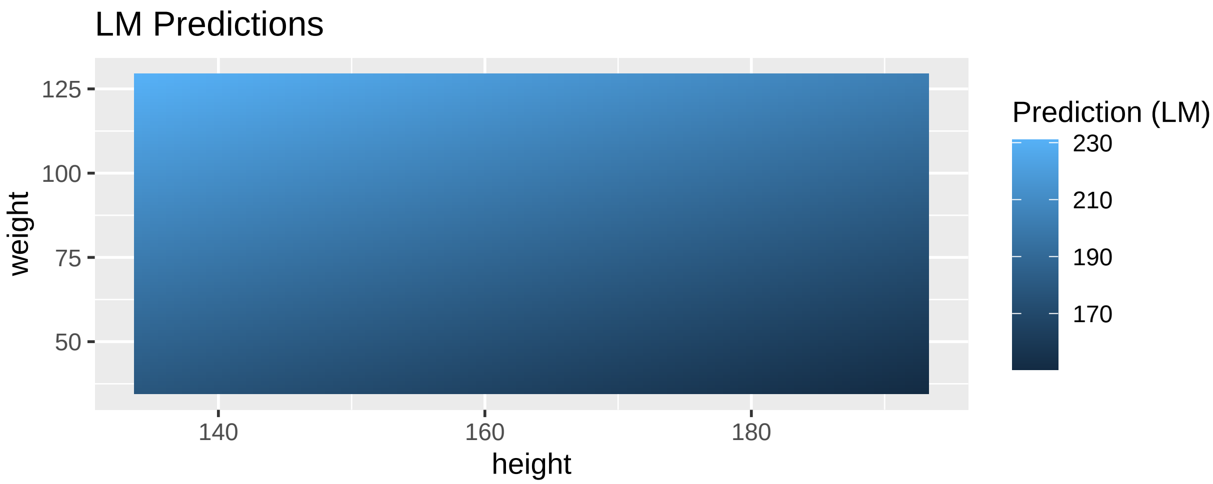
	lambda = 1	lambda = 0.9	lambda = 0.8	lambda = 0.7	lambda = 0.6	lambda = 0.5	lambda = 0.4	lambda = 0.3	lambda = 0.2	lambda = 0.1
subject#	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.57	1.00	1.00
age	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00
sex	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11
test_time	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.19
Jitter(%)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Jitter(Abs)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Jitter:RAP	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Jitter:PPQ5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Jitter:DDP	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Shimmer	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Shimmer(dB)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Shimmer:APQ3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Shimmer:APQ5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Shimmer:APQ11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Shimmer:DDA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
NHR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
HNR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RPDE	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DFA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75
PPE	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.15	1.00

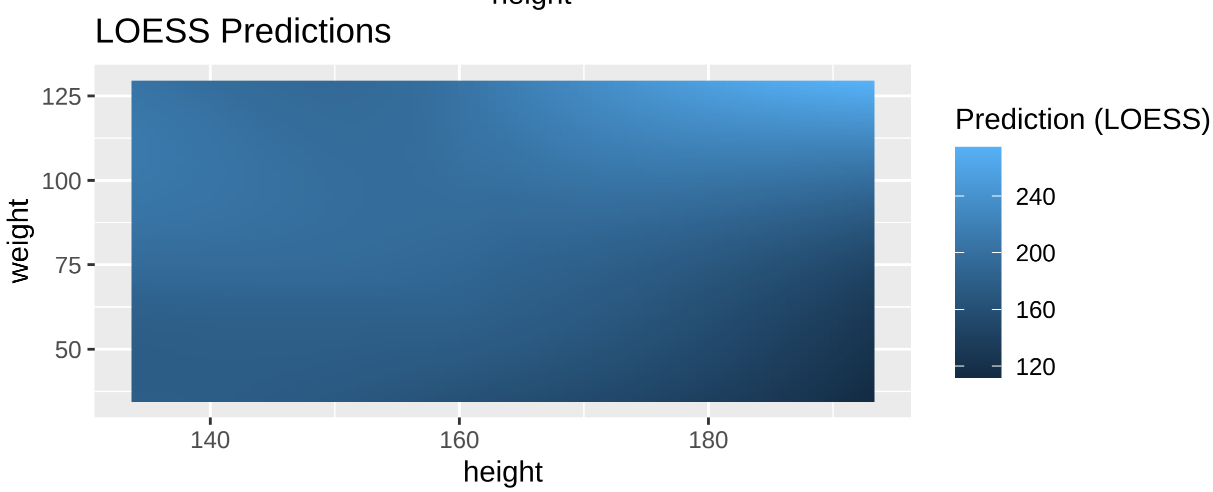
Please find below the requested stable variable set, $\hat{S}^{stable}$:

```
## [1] "subject#" "age"      "DFA"      "PPE"
```

2: Model-Based decision trees

Please find below the requested raster plot (based on 5000 to be the smallest possible number of observations in a tree node, as indicated in the assignment PDF, which results in only one leaf being generated):





3: Neural networks

The following errors (rounded to two decimal places) were observed at the beginning of each 1000th network training iteration:

```
## Iteration: 1 Sigmoid error: 38.15 Sigmoid error_va: 47.93 Tanh error: 105.44 Tanh error_va: 122.99
## Iteration: 1000 Sigmoid error: 4.57 Sigmoid error_va: 4.61 Tanh error: 3.75 Tanh error_va: 3.99
## Iteration: 2000 Sigmoid error: 4.34 Sigmoid error_va: 4.61 Tanh error: 2.51 Tanh error_va: 2.58
## Iteration: 3000 Sigmoid error: 4.16 Sigmoid error_va: 4.57 Tanh error: 1.93 Tanh error_va: 1.82
## Iteration: 4000 Sigmoid error: 3.99 Sigmoid error_va: 4.47 Tanh error: 1.74 Tanh error_va: 1.58
## Iteration: 5000 Sigmoid error: 3.8 Sigmoid error_va: 4.33 Tanh error: 1.64 Tanh error_va: 1.48
## Iteration: 6000 Sigmoid error: 3.6 Sigmoid error_va: 4.12 Tanh error: 1.49 Tanh error_va: 1.35
## Iteration: 7000 Sigmoid error: 3.38 Sigmoid error_va: 3.86 Tanh error: 1.33 Tanh error_va: 1.21
## Iteration: 8000 Sigmoid error: 3.16 Sigmoid error_va: 3.56 Tanh error: 1.05 Tanh error_va: 0.99
## Iteration: 9000 Sigmoid error: 2.95 Sigmoid error_va: 3.24 Tanh error: 0.7 Tanh error_va: 0.72
## Iteration: 10000 Sigmoid error: 2.76 Sigmoid error_va: 2.95 Tanh error: 0.46 Tanh error_va: 0.51
## Iteration: 11000 Sigmoid error: 2.6 Sigmoid error_va: 2.69 Tanh error: 0.35 Tanh error_va: 0.4
## Iteration: 12000 Sigmoid error: 2.46 Sigmoid error_va: 2.46 Tanh error: 0.29 Tanh error_va: 0.32
## Iteration: 13000 Sigmoid error: 2.33 Sigmoid error_va: 2.28 Tanh error: 0.25 Tanh error_va: 0.26
## Iteration: 14000 Sigmoid error: 2.17 Sigmoid error_va: 2.09 Tanh error: 0.21 Tanh error_va: 0.21
## Iteration: 15000 Sigmoid error: 2.05 Sigmoid error_va: 1.95 Tanh error: 0.18 Tanh error_va: 0.17
## Iteration: 16000 Sigmoid error: 1.94 Sigmoid error_va: 1.83 Tanh error: 0.14 Tanh error_va: 0.14
## Iteration: 17000 Sigmoid error: 1.84 Sigmoid error_va: 1.73 Tanh error: 0.12 Tanh error_va: 0.11
```

```
## Iteration: 18000 Sigmoid error: 1.73 Sigmoid error_va: 1.62 Tanh error: 0.1 Tanh error_va: 0.09
## Iteration: 19000 Sigmoid error: 1.61 Sigmoid error_va: 1.51 Tanh error: 0.08 Tanh error_va: 0.07
## Iteration: 20000 Sigmoid error: 1.47 Sigmoid error_va: 1.39 Tanh error: 0.06 Tanh error_va: 0.06
## Iteration: 21000 Sigmoid error: 1.36 Sigmoid error_va: 1.29 Tanh error: 0.05 Tanh error_va: 0.04
## Iteration: 22000 Sigmoid error: 1.25 Sigmoid error_va: 1.2 Tanh error: 0.04 Tanh error_va: 0.04
## Iteration: 23000 Sigmoid error: 1.15 Sigmoid error_va: 1.11 Tanh error: 0.03 Tanh error_va: 0.03
## Iteration: 24000 Sigmoid error: 1.04 Sigmoid error_va: 1.02 Tanh error: 0.02 Tanh error_va: 0.02
## Iteration: 25000 Sigmoid error: 0.94 Sigmoid error_va: 0.93 Tanh error: 0.02 Tanh error_va: 0.02
```

As can be observed from the log of errors, the tanh activation function produces smaller training and validation errors at the end of the last (25000th) iteration as compared to the sigmoid activation function. The rate of decline in error is also faster in the former case.

The weights and biases learned in each case are available below:

Hidden_Layer_Weights_Sigmoid	Hidden_Layer_Biases_Sigmoid	Output_Layer_Weights_Sigmoid
0.6292520	0.6264243	0.3252505
1.8145461	-5.8657822	-4.3464496
-0.1810731	0.3065285	1.3099571
1.1635631	-1.5609544	-1.3011269
0.1878194	-0.4227426	-1.5128705
-1.1832940	5.7394194	-3.6782364
1.7991347	-0.9978777	2.4645232
0.8868820	-2.6366828	2.4440563
0.5975020	0.7483549	0.2505412
-0.2288472	1.0281002	2.4528094

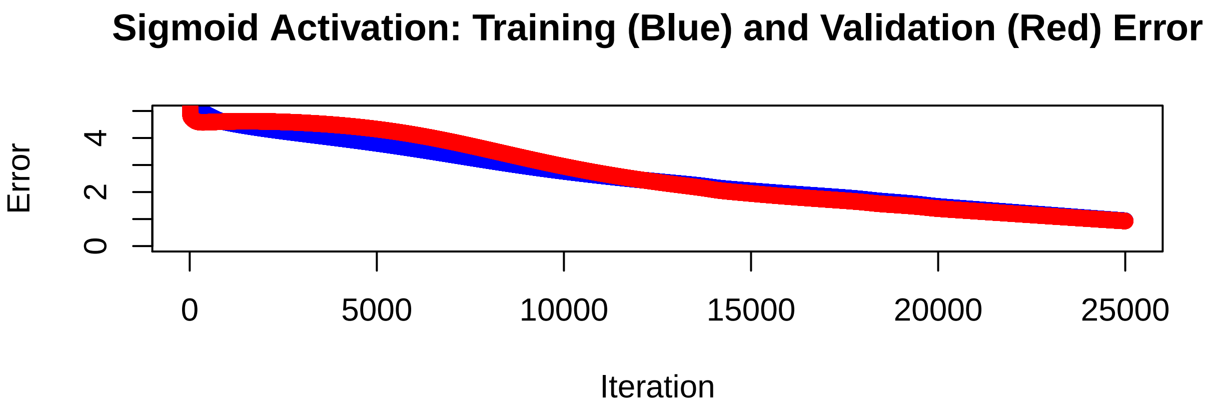
Output_Layer_Bias_Sigmoid
0.8377446

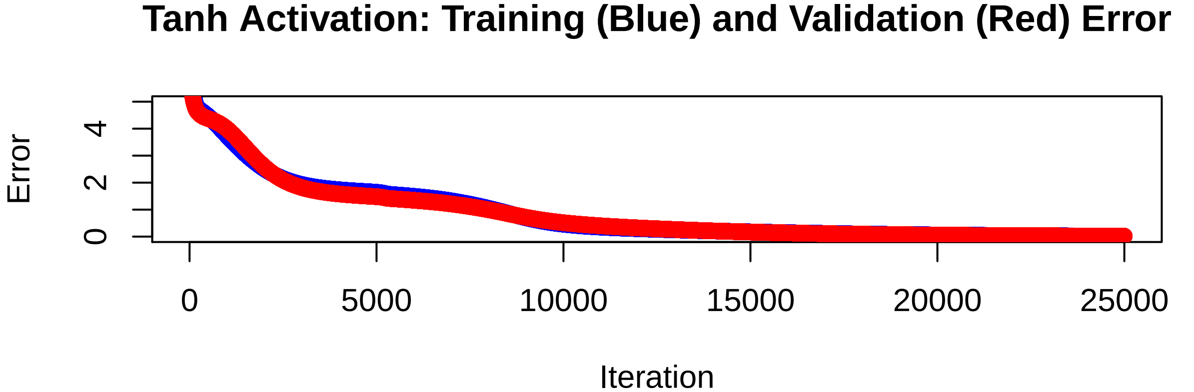
Hidden_Layer_Weights_Tanh	Hidden_Layer_Biases_Tanh	Output_Layer_Weights_Tanh
-0.6670021	-0.6365603	-0.0522287
-1.0265301	-0.2672872	-0.1966515
0.6137629	-3.9739696	2.5413744
-1.0236528	0.0724375	-0.7489867
0.3617864	-3.3003825	-2.8598112
0.5247947	0.5300643	-0.5518730
0.0611107	0.7607050	-0.8205157
0.6203363	0.8023127	-0.4205964
-0.7498886	2.5193126	1.3694597
-1.0957708	0.3588505	-0.5207967

Output_Layer_Bias_Tanh
-0.4797727

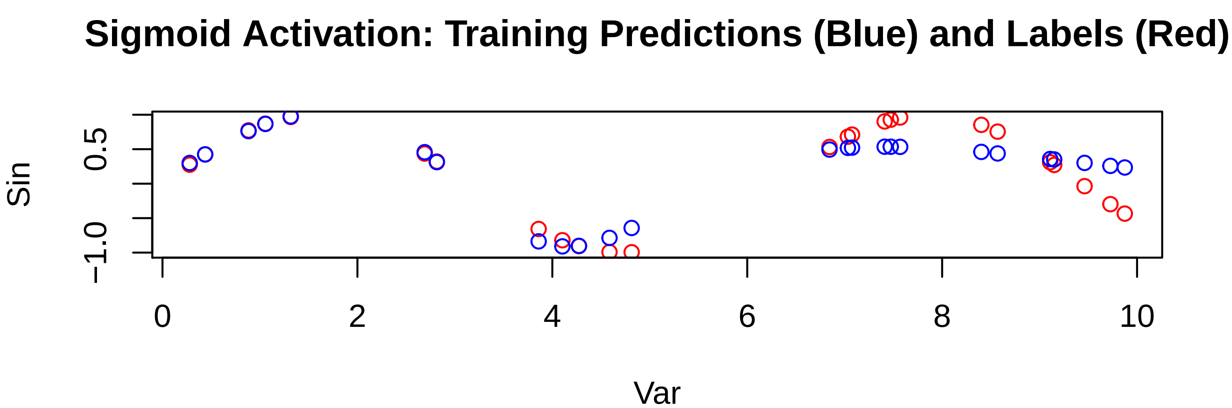
We can see significant differences between the weights/biases learned using the sigmoid and tanh activation functions respectively.

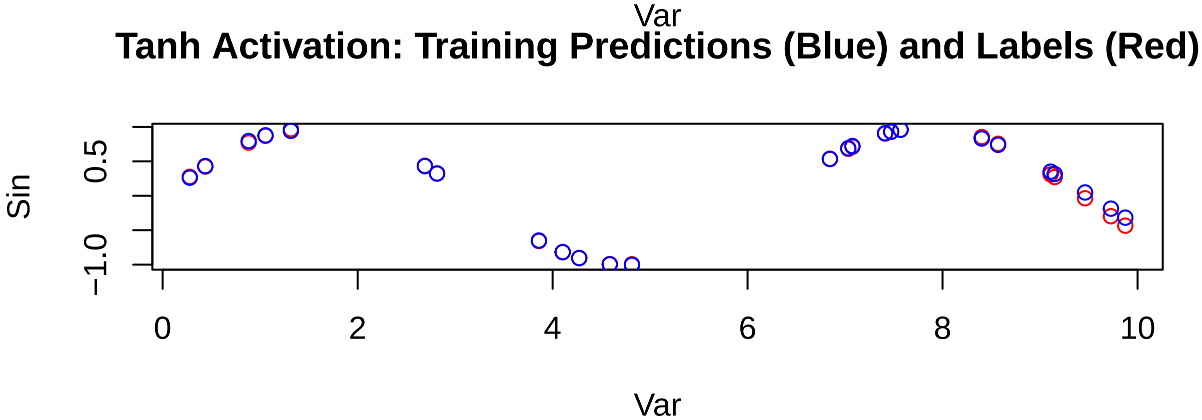
Below are graphs of the errors observed at the beginning of each network training iteration. By way of comparison, the above-mentioned comment on errors holds.

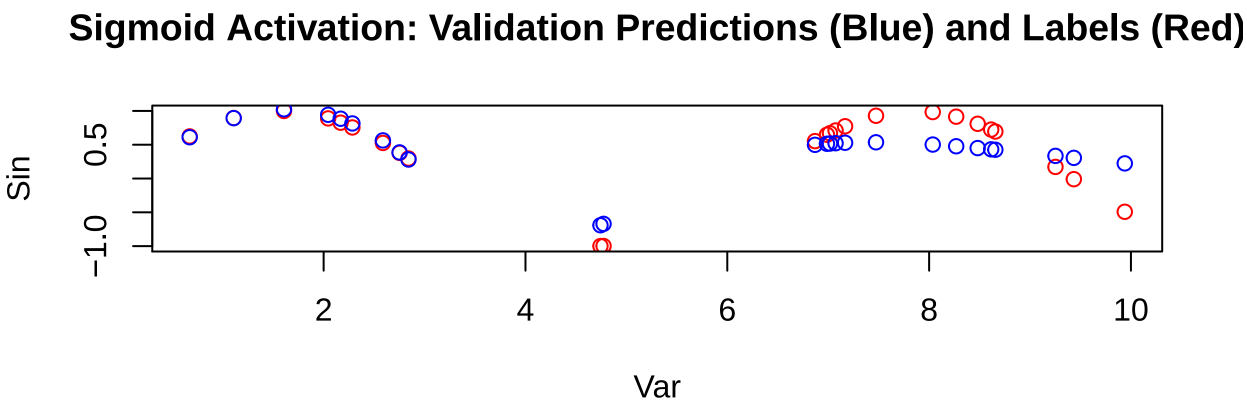


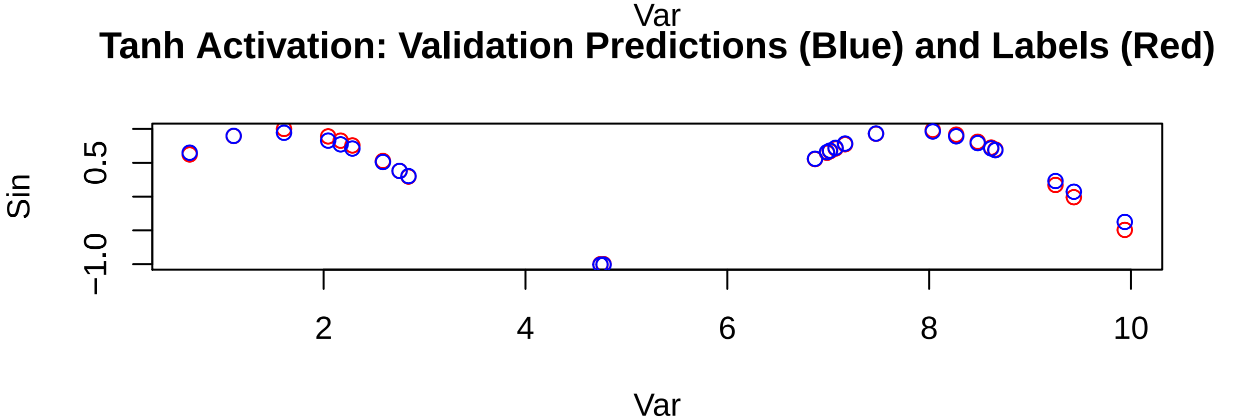


Plotted below are graphs of predictions vs. labels in the training and validation datasets, as generated by the sigmoid and tanh activated networks. Here again, we can see that the tanh activated network performs better.









A brief description of why tanh activation function performs better than the sigmoid activation function for this task is as follows: the tanh function has range [-1, 1] and is centered at 0, whereas the sigmoid function has range[0, 1] and is centered at 0.5; literature suggests that activation functions centered at 0 train networks faster than others (see LeCunn et al)