Perceptron, Collaborative Filtering, Neural Networks

Assignment Report

Machine Learning

To the

The University Of Texas at Dallas

Submitted by:

Prashant Prakash

(Net id: pxp141730)

The University of Texas at Dallas

Dallas – 75252

March, 2015

Perceptrons

- With Stop words

Learning Rate	Number of Iterations	Naïve Bayes		Logistic Regression		Perceptron	
		Spam Accuracy	Ham Accuracy	Spam Accuracy	Ham Accuracy	Spam Accuracy	Ham Accuracy
.03	20	84.61%	97.98%	94.61%	77.58%	70.0%	95.68%
.03	40	84.61%	97.98%	83.07%	94.54%	83.84%	93.67%
.03	60	84.61%	97.98%	83.07%	95.11%	83.84%	93.67%
.03	80	84.61%	97.98%	83.07%	94.82%	83.84%	93.67%
.03	100	84.61%	97.98%	83.07%	94.54%	83.84%	93.67%
.09	20	84.61%	97.98%	94.61%	76.72%	70.0%	97.12%
.09	40	84.61%	97.98%	85.38%	93.96%	83.84%	91.66%
.09	60	84.61%	97.98%	83.84%	94.82%	83.84%	91.66%
.09	80	84.61%	97.98%	83.07%	95.11%	83.84%	91.66%
.09	100	84.61%	97.98%	83.07%	94.82%	83.84%	91.66%
0.27	20	84.61%	97.98%	94.61%	77.29%	70.0%	96.55%
0.27	40	84.61%	97.98%	83.07%	95.11%	86.15%	92.52%
0.27	60	84.61%	97.98%	83.07%	95.40%	86.15%	92.52%
0.27	80	84.61%	97.98%	83.07%	95.11%	86.15%	92.52%
0.27	100	84.61%	97.98%	83.07%	94.82%	86.15%	92.52%
0.81	20	84.61%	97.98%	94.61%	76.72%	72.30%	95.68%
0.81	40	84.61%	97.98%	82.30%	95.11%	89.23%	91.09%
0.81	60	84.61%	97.98%	83.07%	94.82%	89.23%	91.09%
0.81	80	84.61%	97.98%	82.30%	95.11%	89.23%	91.09%
0.81	100	84.61%	97.98%	82.30%	94.82%	89.23%	91.09%
2.43	20	84.61%	97.98%	94.61%	76.72%	72.30%	95.11%
2.43	40	84.61%	97.98%	82.30%	95.11%	85.38%	93.67%
2.43	60	84.61%	97.98%	83.07%	94.82%	85.38%	93.67%
2.43	80	84.61%	97.98%	83.07%	94.82%	85.38%	93.67%
2.43	100	84.61%	97.98%	82.30%	94.82%	85.38%	93.67%

Conclusion: The above results for LR has been measured by keeping lambda value as 0 that is there is no Regularization. Over all LR behaves almost same to Perceptrons in text classification for our dataset. Since Perceptrons classification is based on multiplication of weight vector and attributes vector and there is no probability calculation involved it converges faster than LR. This can be seen by the readings that for learning rate 0.3 and number of iterations 40 it got an accuracy of 83.84% for spam and 93.67% for ham. After that even increasing number of iterations does not lead to any change in accuracy. We can also see the accuracy improvement with learning rate 0.81 which helped in convergence and reaching to minima. Behaves better with learning rate close to 0.8.

- Without Stop words

Number of Iterations	Learning Rate	Naïve Bayes		Logistic Regression		Perceptron	
		Spam Accuracy	Ham Accuracy	Spam Accuracy	Ham Accuracy	Spam Accuracy	Ham Accuracy
.03	20	86.92%	96.55%	95.38%	60.05%	30.0%	95.68%
.03	40	86.92%	96.55%	85.38%	97.70%	53.07%	95.11%
.03	60	86.92%	96.55%	84.61%	97.70%	77.69%	92.81%
.03	80	86.92%	96.55%	84.61%	97.70%	80.0%	93.1%
.03	100	86.92%	96.55%	84.61%	97.70%	77.69%	93.96%
.09	20	86.92%	96.55%	95.38%	65.51%	31.53%	96.55%
.09	40	86.92%	96.55%	86.15%	97.70%	70.76%	91.95%
.09	60	86.92%	96.55%	84.61%	97.70%	80.76%	92.24%
.09	80	86.92%	96.55%	84.61%	97.70%	80.76%	92.81%
.09	100	86.92%	96.55%	84.61%	97.70%	81.53%	92.52%
0.27	20	86.92%	96.55%	95.38%	65.51%	40.0%	95.68%
0.27	40	86.92%	96.55%	86.92%	98.27%	74.61%	92.52%
0.27	60	86.92%	96.55%	84.61%	98.27%	83.07%	92.24%
0.27	80	86.92%	96.55%	84.61%	98.27%	83.07%	92.81%
0.27	100	86.92%	96.55%	84.61%	98.27%	83.07%	93.56%
0.81	20	86.92%	96.55%	95.38%	63.50%	37.69%	95.97%
0.81	40	86.92%	96.55%	86.92%	98.27%	76.92%	92.52%
0.81	60	86.92%	96.55%	84.61%	98.27%	80.76%	93.39%
0.81	80	86.92%	96.55%	84.61%	98.27%	82.30%	93.10%
0.81	100	86.92%	96.55%	84.61%	98.27%	62.30%	95.97%
2.43	20	86.92%	96.55%	95.38%	67.81%	39.23%	94.82%
2.43	40	86.92%	96.55%	86.15%	98.27%	60.76%	94.54%
2.43	60	86.92%	96.55%	84.61%	98.27%	80.0%	93.10%
2.43	80	86.92%	96.55%	84.61%	98.27%	81.53%	93.39%
2.43	100	86.92%	96.55%	84.61%	98.27%	83.07%	92.81%

Conclusion: After removal of stopwords perceptron gives accuracy got reduced as we can see from readings. The removal of stopwords reduces the number of features. So we can see from readings that for getting better accuracy we have to increase the number of iterations and learning rate to converge to minima. Reduction in number of features might require more number of iterations to learn the model correctly. Behaves better with learning rate close to 0.3 and number of iterations around 200.

Neural Networks

- Data set 1

Learning Rate	Momentum	Iteration	Hidden Layers	Hidden Units	Accuracy
0.03	0.1	2	1	2	72.8033 %
0.03	0.1	10	2	8	73.4286 %
0.03	0.2	20	3	12	76.3256 %
0.1	0.4	2	1	5	83.2636 %
0.1	0.1	20	1	10	93.7238 %
0.1	0.2	10	2	15	92.0502 %
0.1	0.1	50	1	10	94.7699 %
0.1	0.2	50	3	30	95.3975 %
0.1	0.2	100	1	10	94.9791 %
0.1	0.2	100	2	20	94.5607 %
0.1	0.4	100	3	17	95.1883 %
0.3	0.1	2	1	4	91.4226 %
0.3	0.1	20	1	10	95.3975 %
0.3	0.1	50	2	20	95.3975 %
0.3	0.2	50	3	30	96.2343 %
0.3	0.4	50	3	17	93.3054 %
0.3	0.4	100	1	10	94.7699 %
0.3	0.4	100	2	20	95.1883 %
0.9	0.1	2	1	4	93.9331 %
0.9	0.2	10	1	10	94.5607 %
0.9	0.2	20	2	20	94.9791 %
0.9	0.4	50	3	30	89.749 %
0.9	0.6	50	3	17	81.59 %
0.9	0.6	100	2	20	93.7238 %
0.9	0.6	100	3	12	77.6151 %

- Data Set 2

Learning Rate	Momentum	Iteration	Hidden Layers	Hidden Units	Accuracy
0.03	0.1	2	1	2	67.3246 %
0.03	0.1	10	2	8	67.3246 %
0.03	0.2	20	3	12	67.3246 %
0.1	0.4	2	1	5	82.6754 %
0.1	0.1	20	1	10	94.2982 %
0.1	0.2	10	2	15	87.2807 %
0.1	0.1	50	1	10	93.8596 %
0.1	0.2	50	3	30	67.3246 %
0.1	0.2	100	1	10	94.0789 %
0.1	0.2	100	2	20	92.7632 %
0.1	0.4	100	3	17	95.8333 %
0.3	0.1	2	1	4	77.6316 %
0.3	0.1	20	1	10	94.5175 %
0.3	0.1	50	2	20	95.614 %
0.3	0.2	50	3	30	93.8596 %
0.3	0.4	50	3	17	93.8596 %
0.3	0.4	100	1	10	94.2982 %
0.3	0.4	100	2	20	94.5175 %
0.9	0.1	2	1	4	71.2719 %
0.9	0.2	10	1	10	78.5088 %
0.9	0.2	20	2	20	70.3947 %
0.9	0.4	50	3	30	73.6842 %
0.9	0.6	50	3	17	90.5702 %
0.9	0.6	100	2	20	93.6404 %
0.9	0.6	100	3	12	71.2719 %

- Data Set 3

Learning Rate	Momentum	Iteration	Hidden Layers	Hidden Units	Accuracy
0.03	0.1	2	1	2	72.0074 %
0.03	0.1	10	2	8	72.0074 %
0.03	0.2	20	3	12	72.0074 %
0.1	0.4	2	1	5	93.0018 %
0.1	0.1	20	1	10	97.4217 %
0.1	0.2	10	2	15	96.8692 %
0.1	0.1	50	1	10	97.6059 %
0.1	0.2	50	3	30	97.7901 %
0.1	0.2	100	1	10	97.9742 %
0.1	0.2	100	2	20	98.1584 %
0.1	0.4	100	3	17	97.2376 %
0.3	0.1	2	1	4	94.4751 %
0.3	0.1	20	1	10	96.6851 %
0.3	0.1	50	2	20	97.0534 %
0.3	0.2	50	3	30	97.4217 %
0.3	0.4	50	3	17	96.8692 %
0.3	0.4	100	1	10	96.3168 %
0.3	0.4	100	2	20	95.9484 %
0.9	0.1	2	1	4	97.0534 %
0.9	0.2	10	1	10	95.3959 %
0.9	0.2	20	2	20	97.4217 %
0.9	0.4	50	3	30	95.5801 %
0.9	0.6	50	3	17	95.2118 %
0.9	0.6	100	2	20	95.3959 %
0.9	0.6	100	3	12	95.7643 %

Conclusion: After various experiments performed on all the three datasets. The accuracy is better if we use 10 hidden units either with one layer or with the 2 layers. Also learning rate with 0.1 and momentum as 0.2 are fine for our dataset. If we pay attention to the readings we find that sometimes suddenly increasing the learning rate or number of iterations leads to reduction in the accuracy. This is happening because of the divergence.

Collaborative Filtering

Number of Test Instances	RMSE	MAE
1000	0.90	0.54
2000	0.97	0.59
3000	0.95	0.60
4000	0.95	0.60
5000	0.95	0.60
6000	0.94	0.58
7000	0.93	0.57
8000	0.92	0.56
9000	0.98	0.55
10000	0.98	0.55

Final RMSE: 0.92 Final MAE: 0.51

Explanation: RMSE and MAE are recorded at every 50 rows in test set. Like 50, 100, 150… and so on. I captured RMSE with different test data set as provided in the table given above.