Adaptive Step sizes weekly report

Anubhav Mittal
Project Supervisors : Martin Jaggi, Aymeric Dieuleveut

November 16, 2019

1 Week of December 17th and December 10th, 2018

- Changed approach from random reshuffling to SGD.
- Tried SGD1/2 + averaging
- Plots for SGD1/2 with/without burnin, initial conditions.
- Used excess loss instead of loss for Logistic case. MLE loss was calculated using Newton's method.

1.1 Least squares

Loss.png

Figure 1: Training Loss

Loss log-log.png

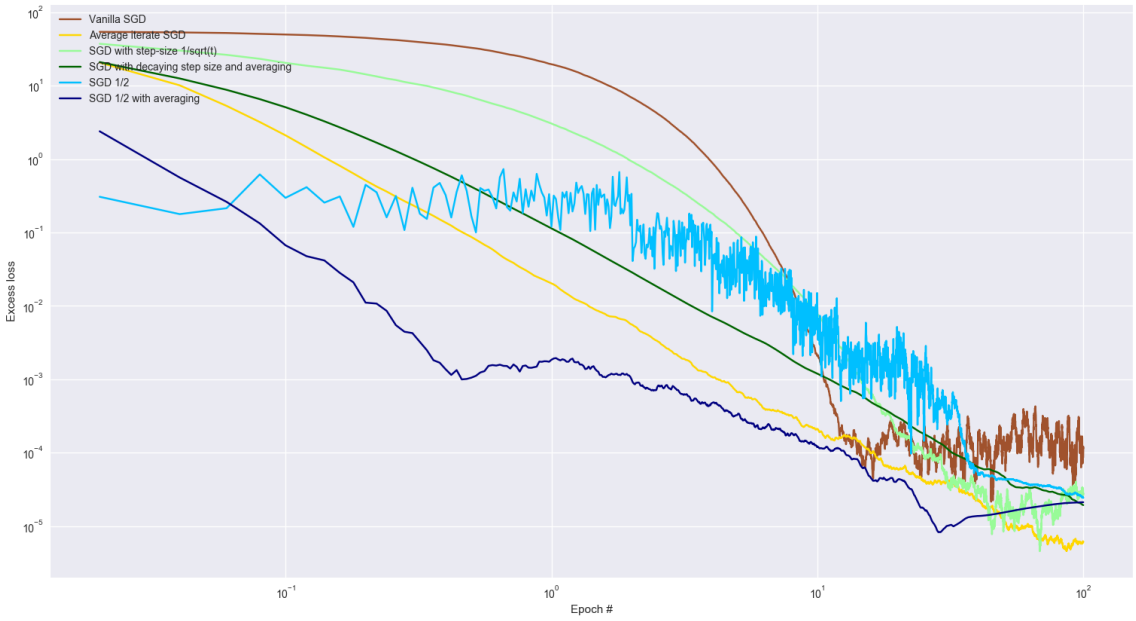

Figure 2: Training Loss log-log

loss for different mini variations of SGDH.png

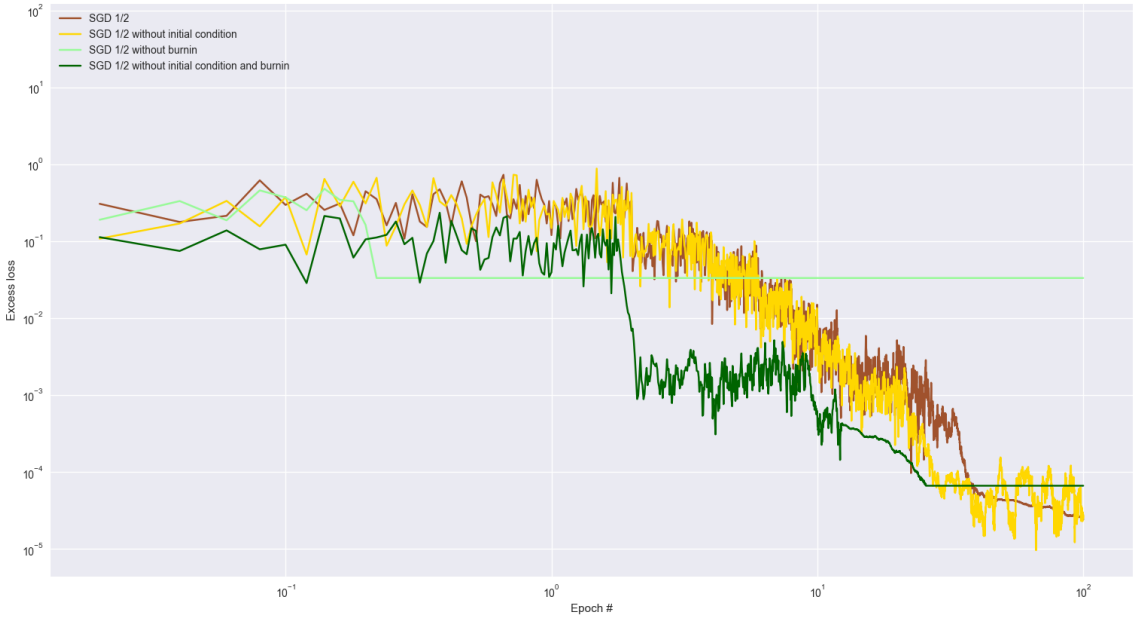

Figure 3: Training loss for different mini variations of SGDH

size variation.png

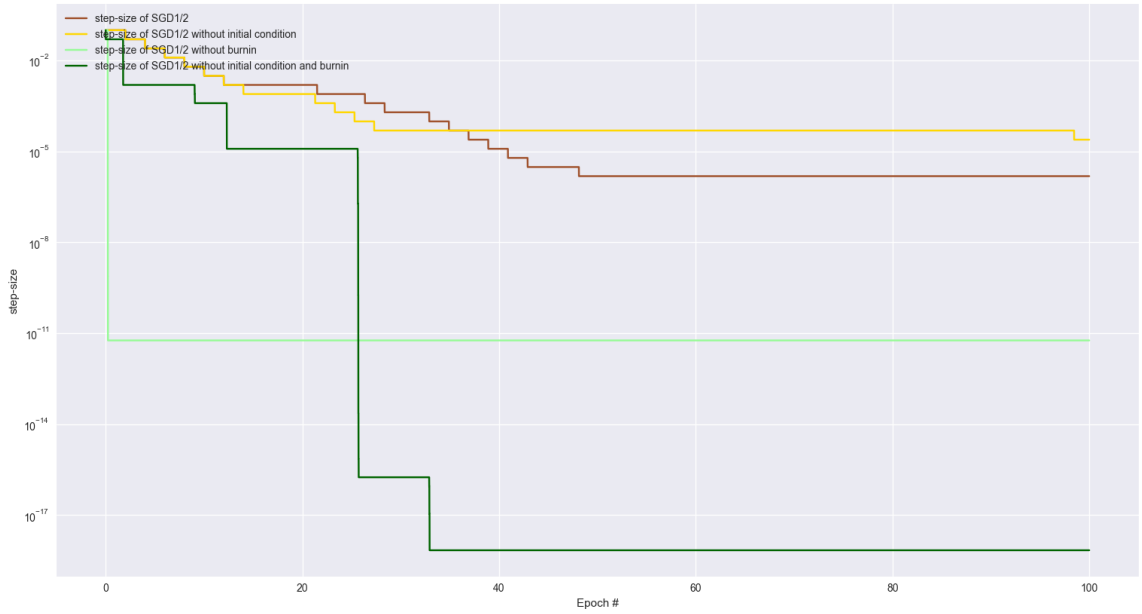

Figure 4: Step size variation

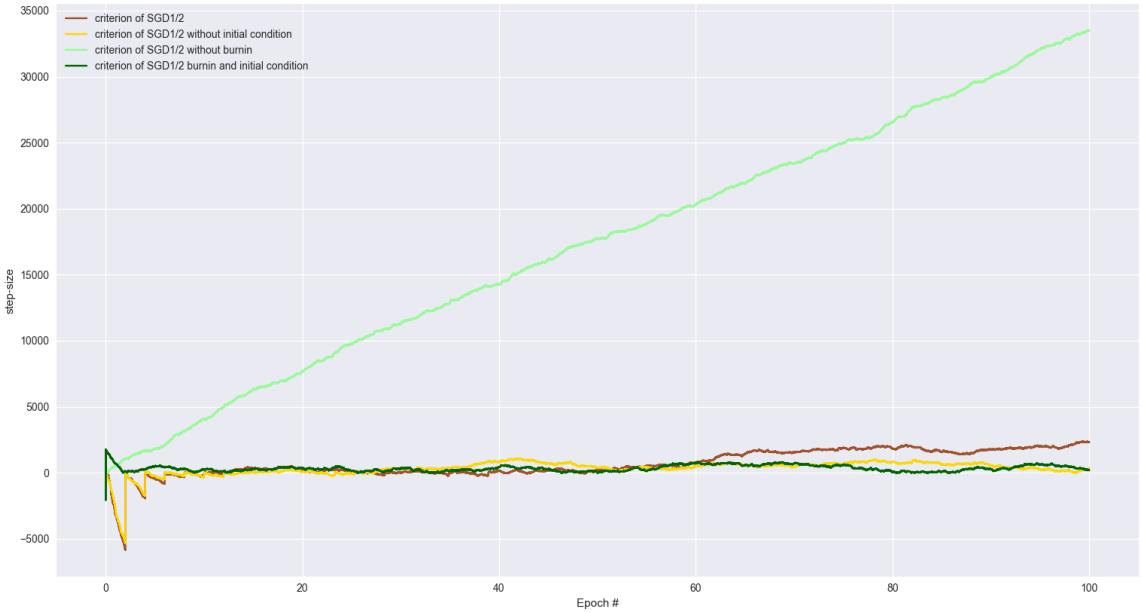

Figure 5: Criterion

first iter.png

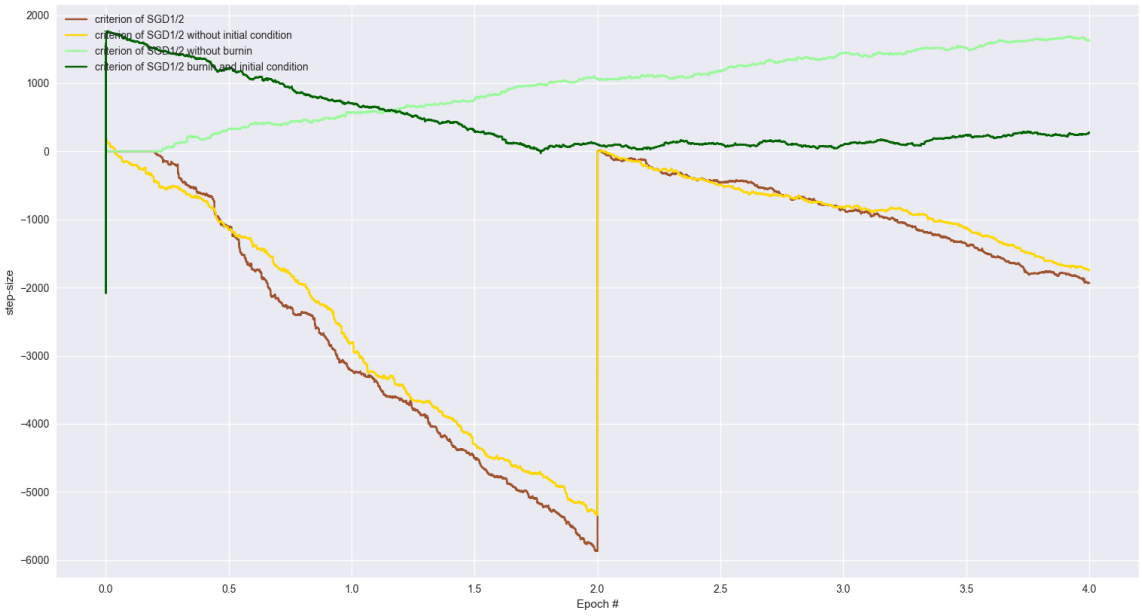

Figure 6: Criterion first iter

1.2 Logistic regression

Loss Logistic.png

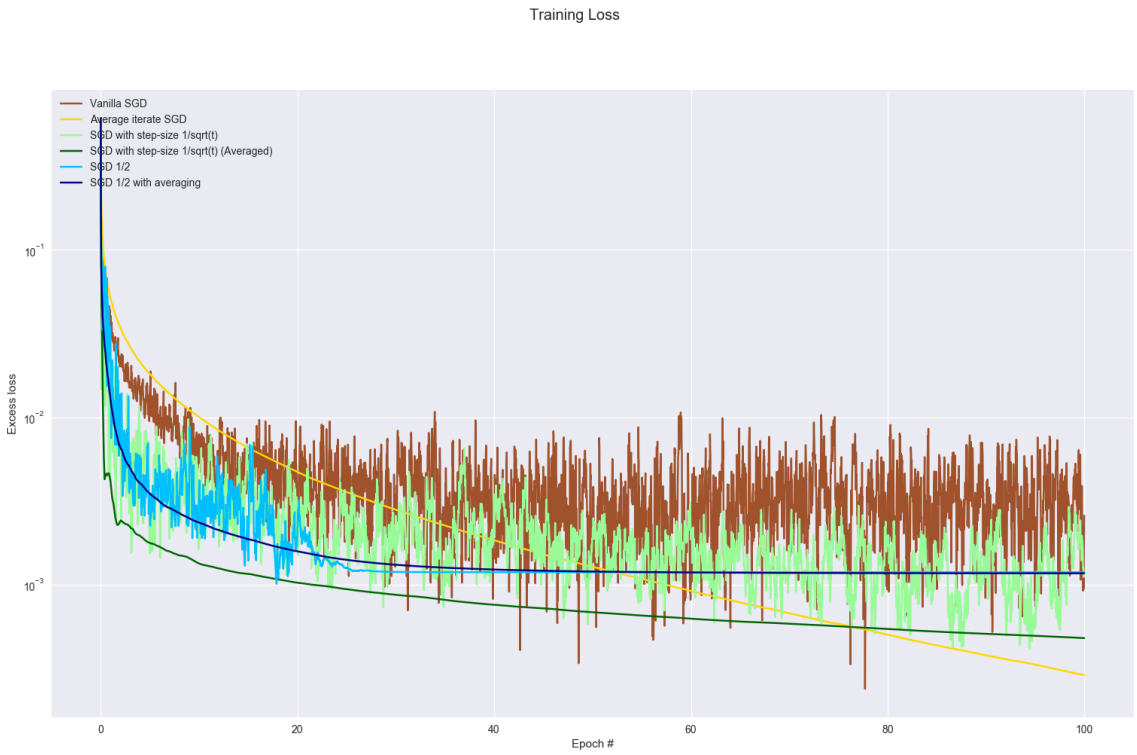

Figure 7: Training Loss

Loss Log-Log Logistic.png

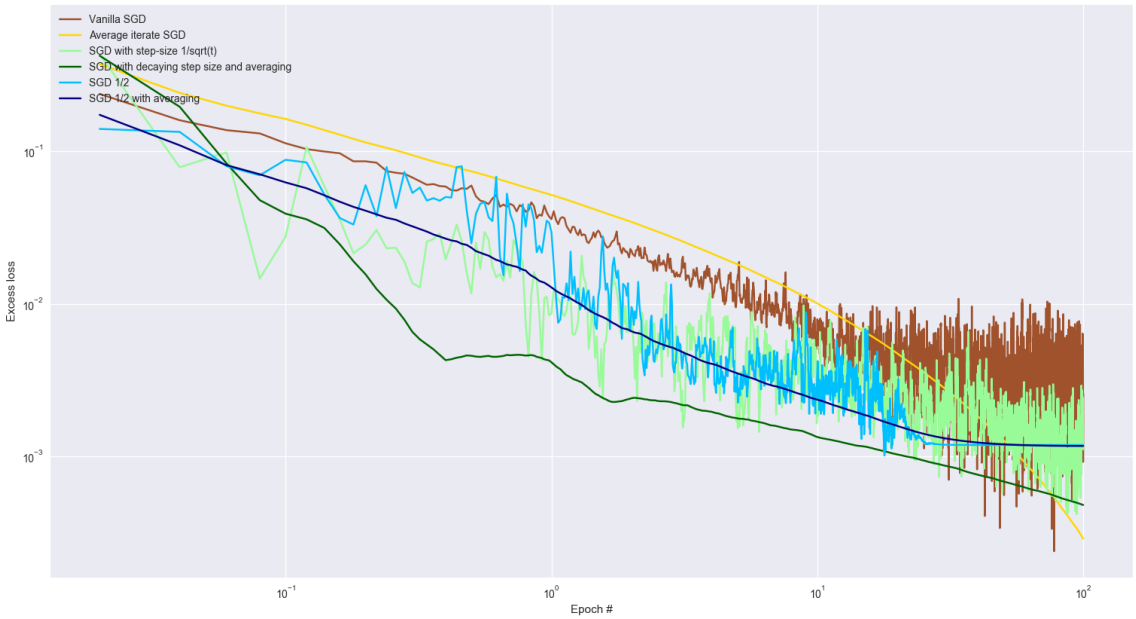

Figure 8: Training Loss log-log

of Training loss for different SGDH Logistic.png

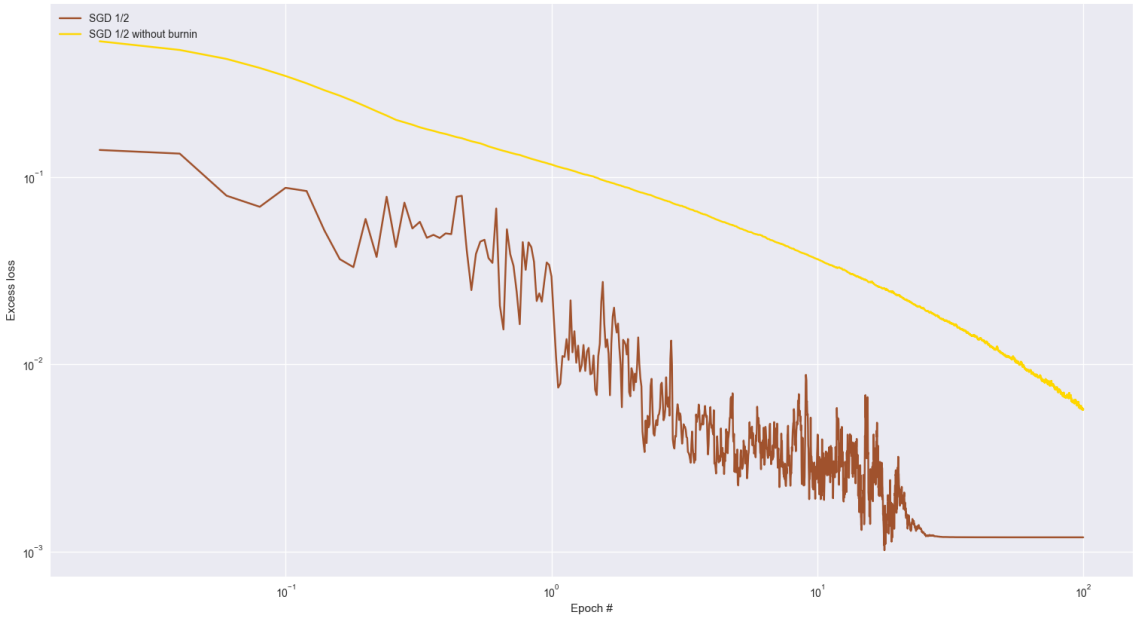

Figure 9: Training loss for different mini variations of SGDH

of step size for different SGDH Logistic.png

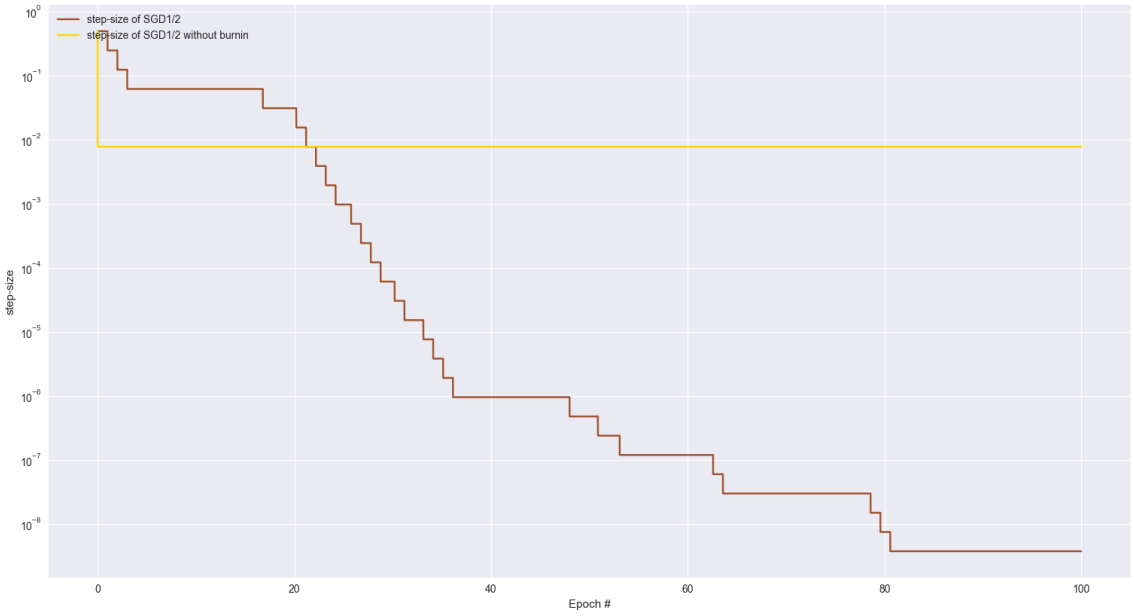

Figure 10: Step size variation

of criterion for different SGDH Logistic.png

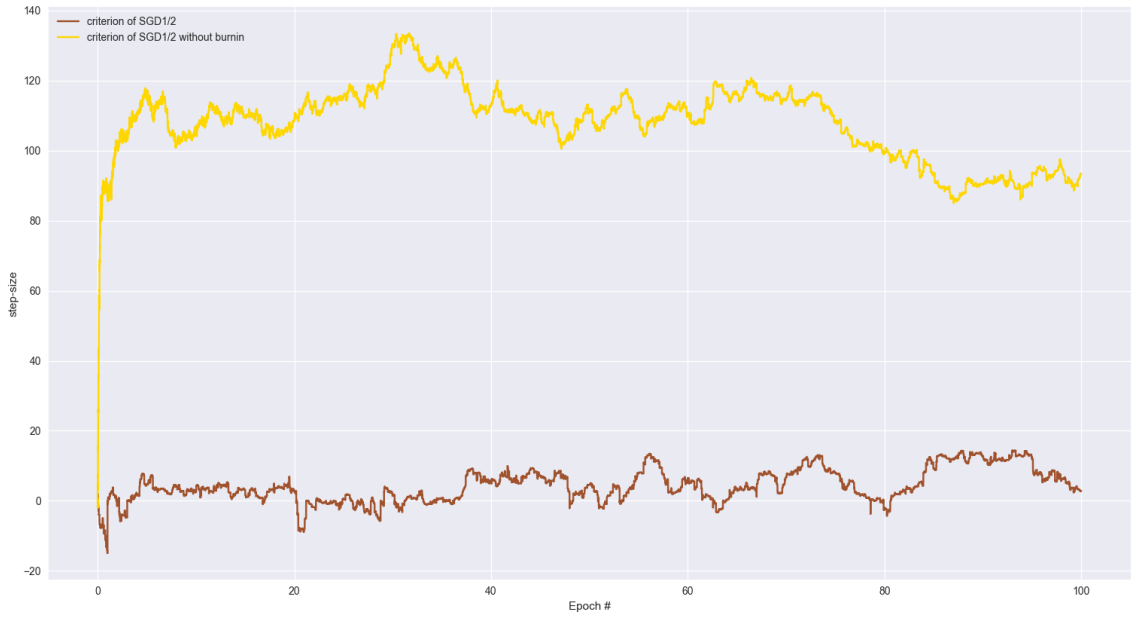

Figure 11: Criterion

2 Week of November 19th and November 12th, 2018

2.1 Comparing performances of the algorithms

2.1.1 Least Squares Regression

When the step-sizes/gamma for the algorithms was properly tuned, the epoch-wise periodicity remains. The algorithm SGD1/2 performs equivalent to SGD with Averaging, but is worse to SGD with decaying step-size and averaging.

Also, the performance of SGD1/2 was pretty unreliable, with very different performance on two runs with same hyperparameters. This may suggest for refinement of the restart condition.

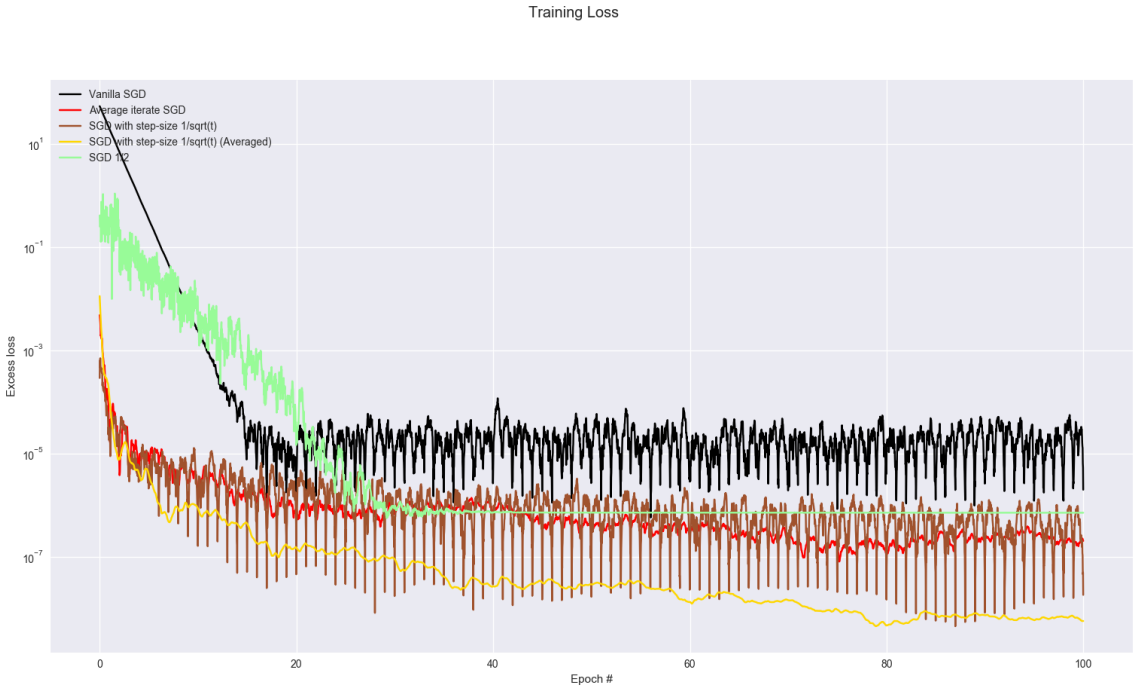

Figure 12: Performance of algorithms on Least squares (SemiLogy)

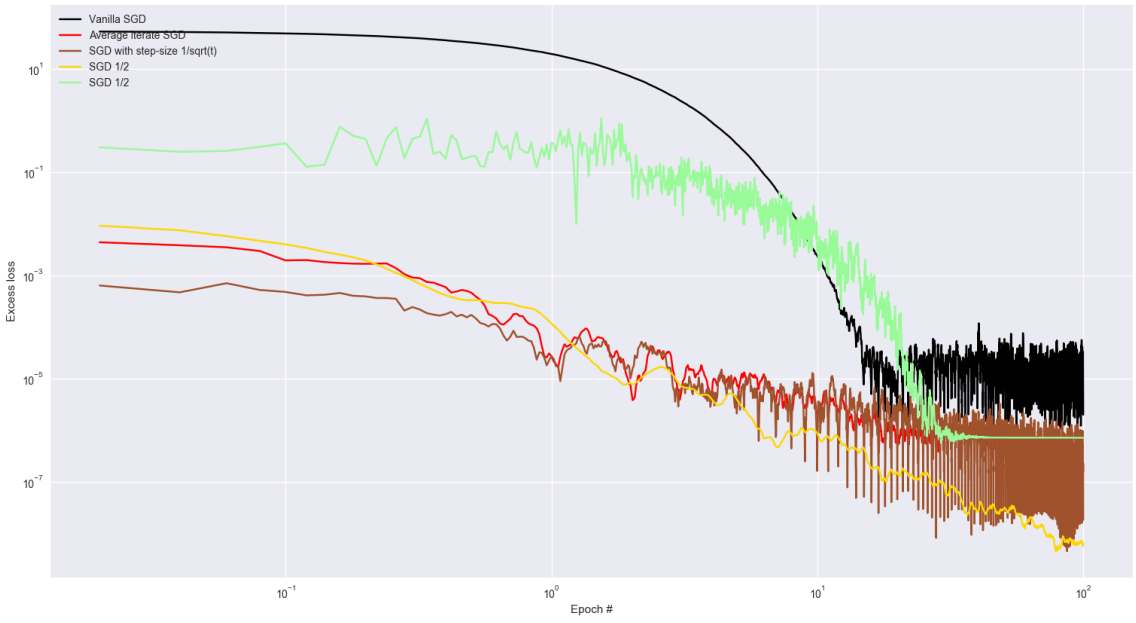

Figure 13: Performance of algorithms on Least squares (LogLog)

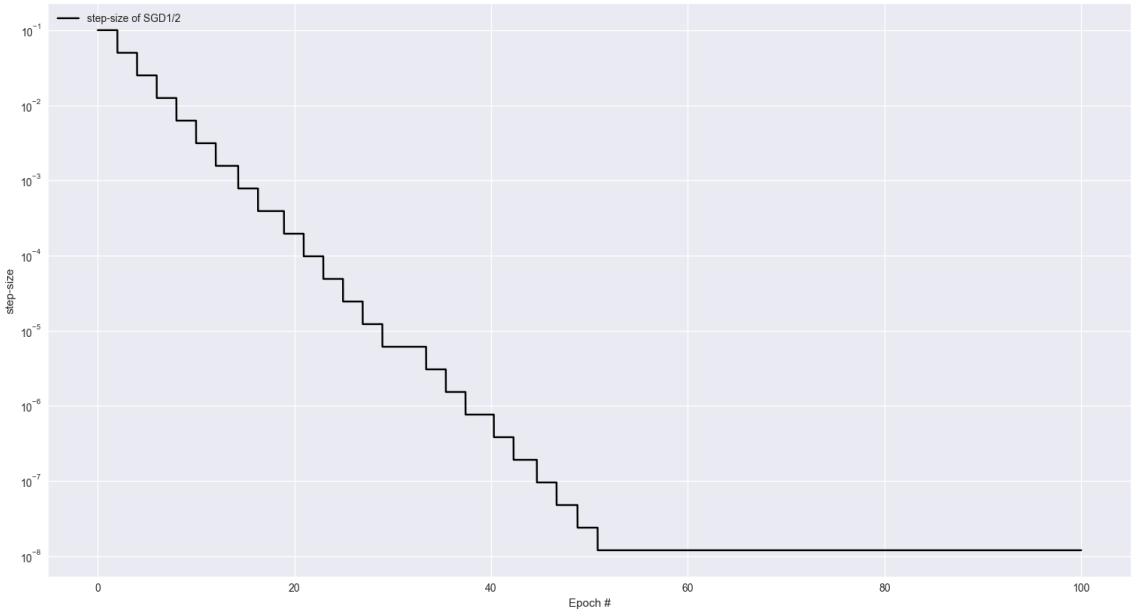

Figure 14: Step size change in SGD1/2

2.1.2 Logistic Regression

Again, the performance of SGD1/2 was too inconsistent across multiple runs. Sometimes its performance was same as that of Sgd with decaying step-size (averaged), while other times it was much worse.

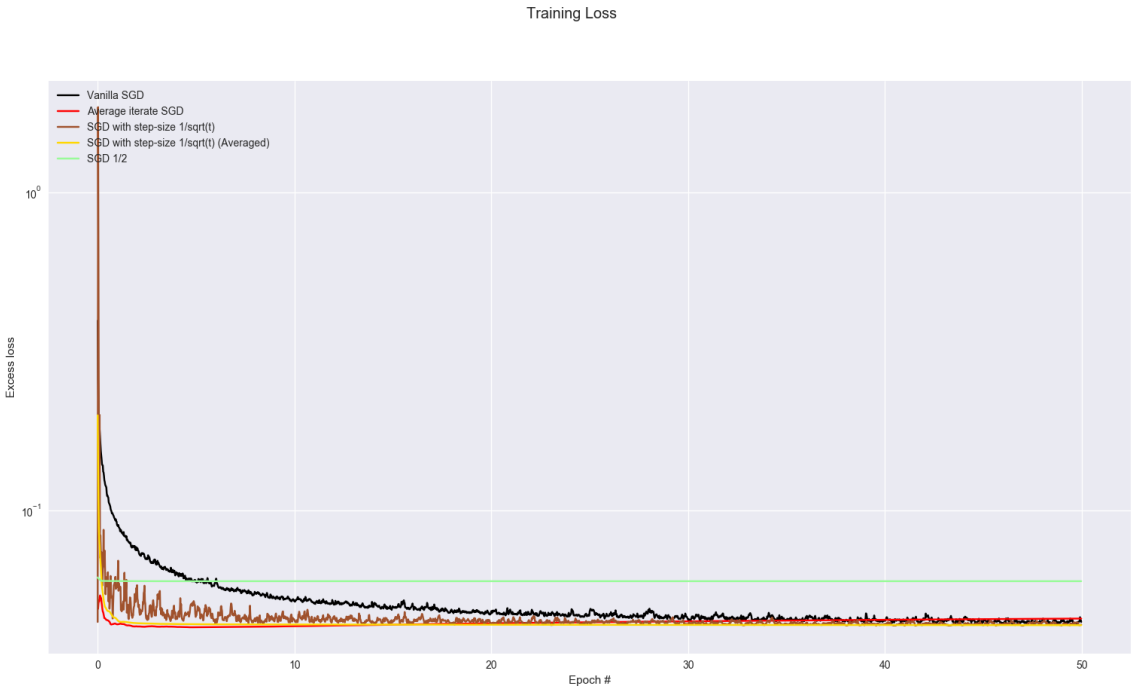

Figure 15: Performance of algorithms on Least squares (SemiLogy)

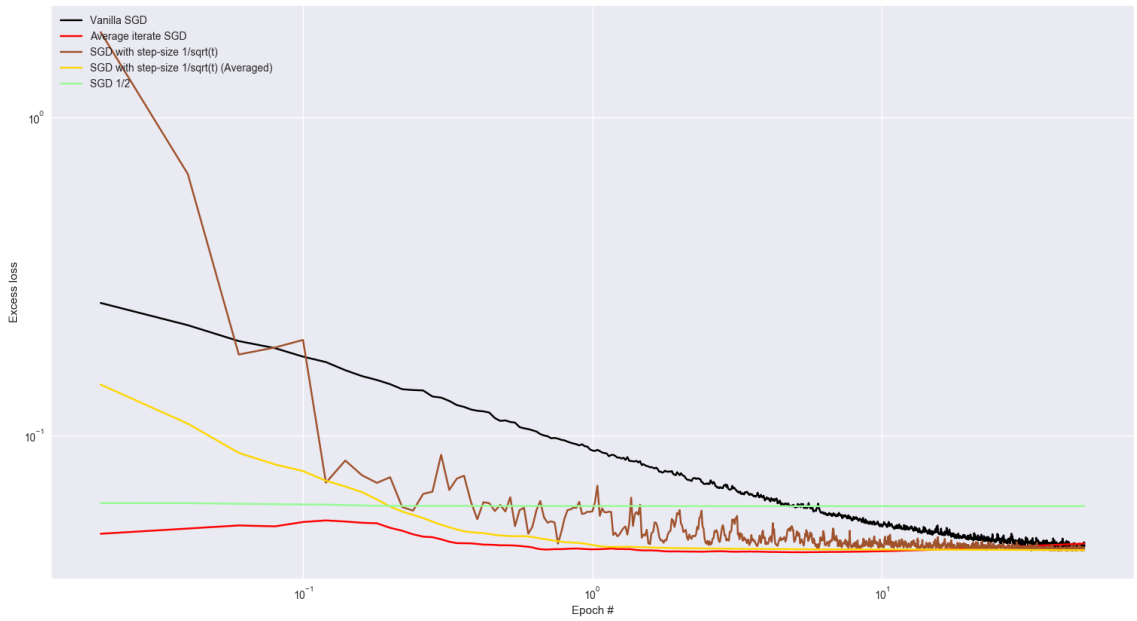

Figure 16: Performance of algorithms on Least squares (LogLog)

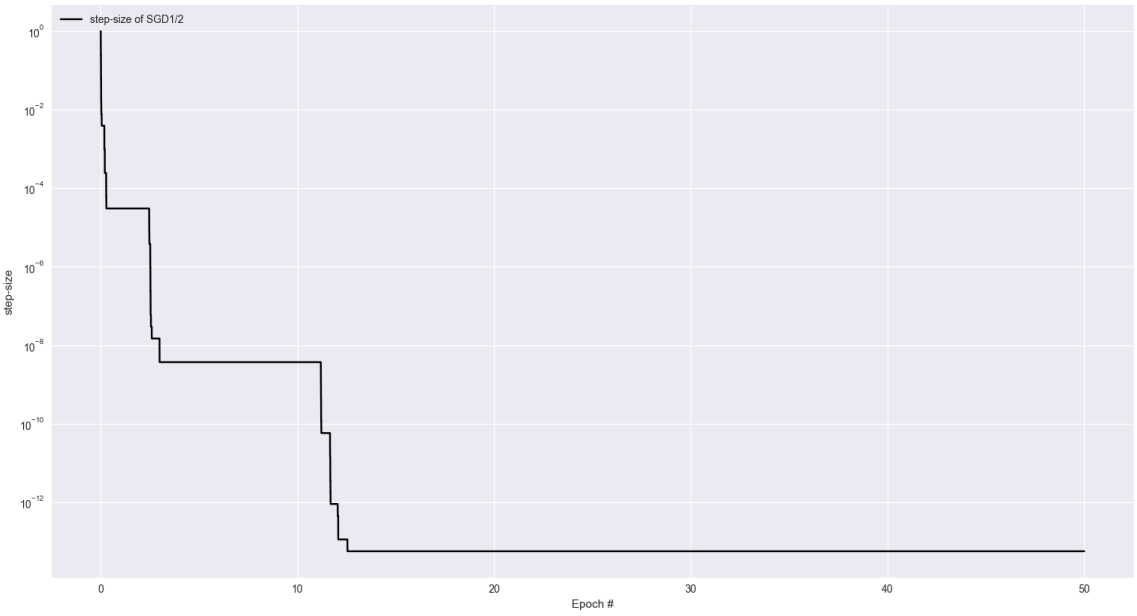

Figure 17: Step size change in SGD1/2

2.2 Improving step sizes

For all the algorithms, I tried to include the best step size for better comparison (All the codes are up at git). This is all for the synthetic dataset we generate, as described in the Convergence diagnostics paper.

2.2.1 Least Squares Regression

1. For standard SGD in LSR, step-size chosen=0.0001

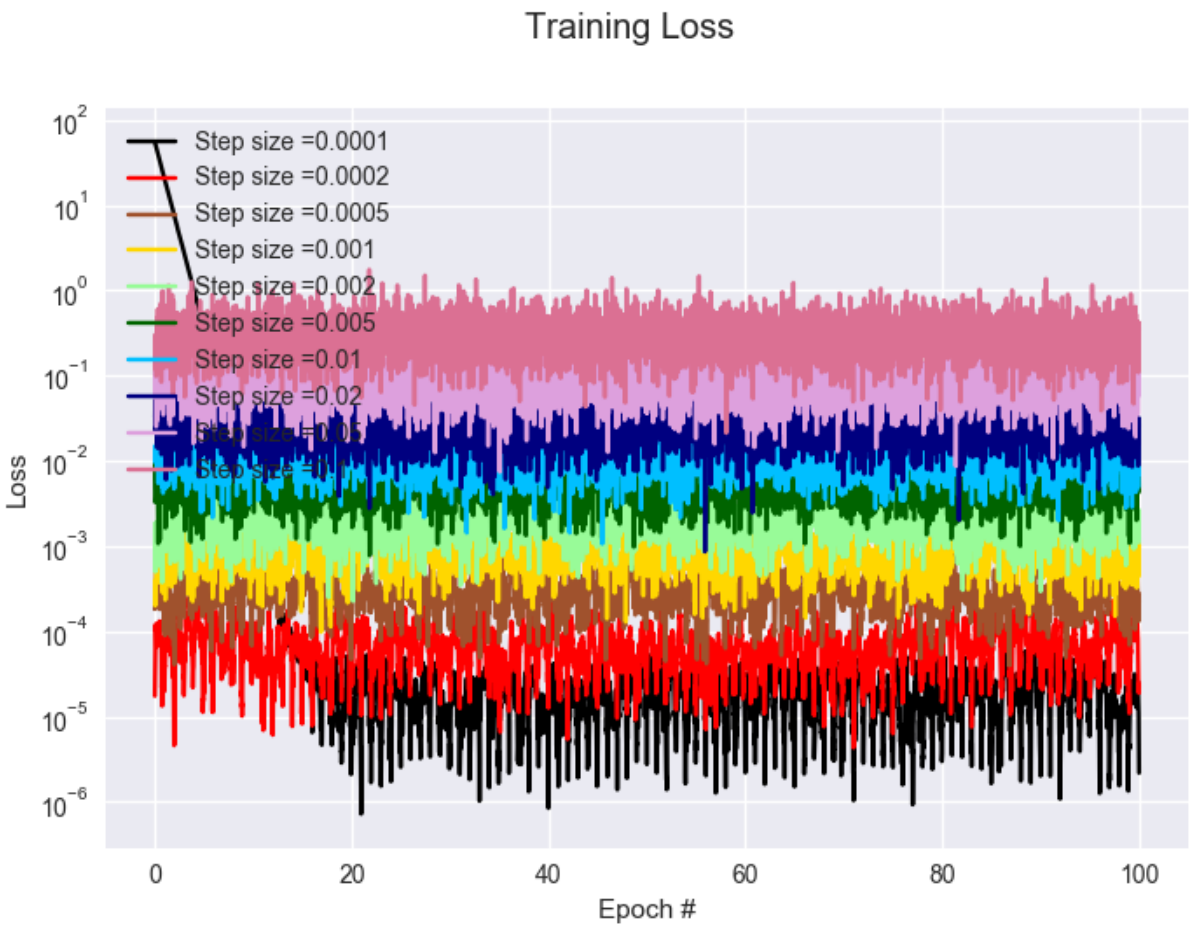

Figure 18: Vanilla SGD for Least squares

2. For average SGD in LSR, step-size chosen=0.01

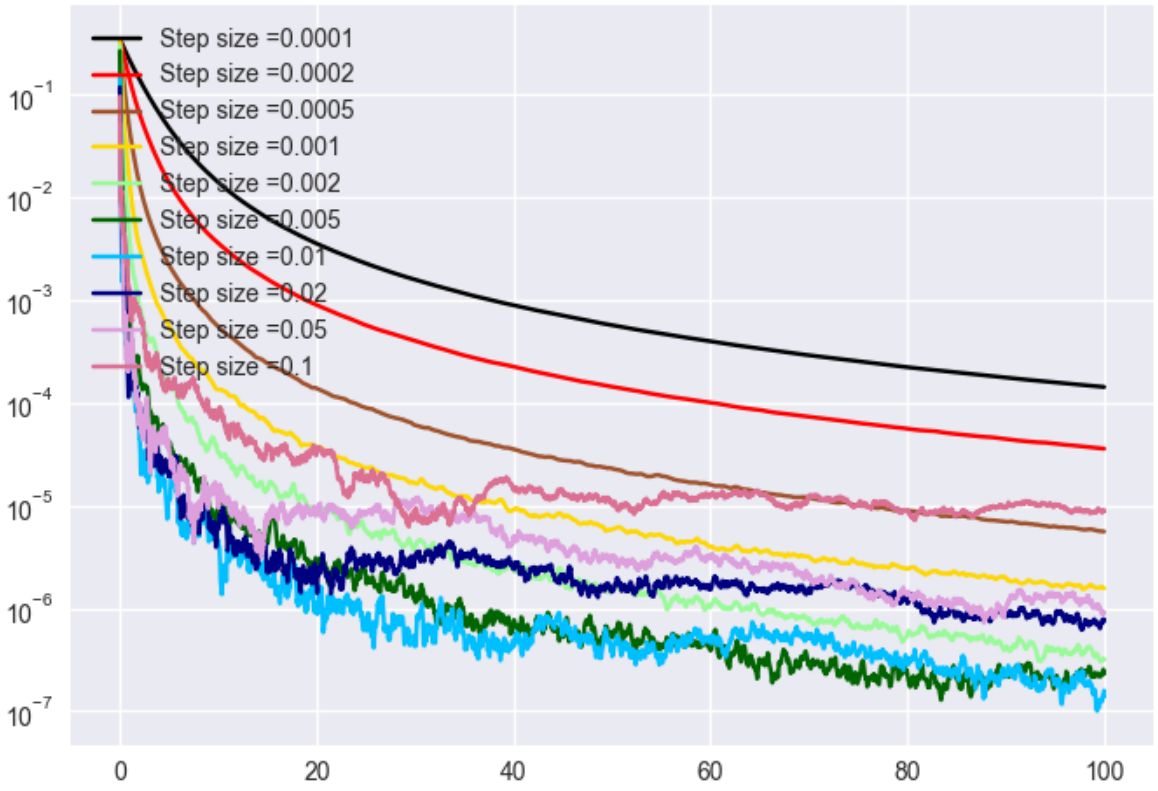

Figure 19: Average SGD for Least squares

3. For SGD with $\gamma/\sqrt{t}$ decaying stepsize, in LSR, chosen γ=0.01

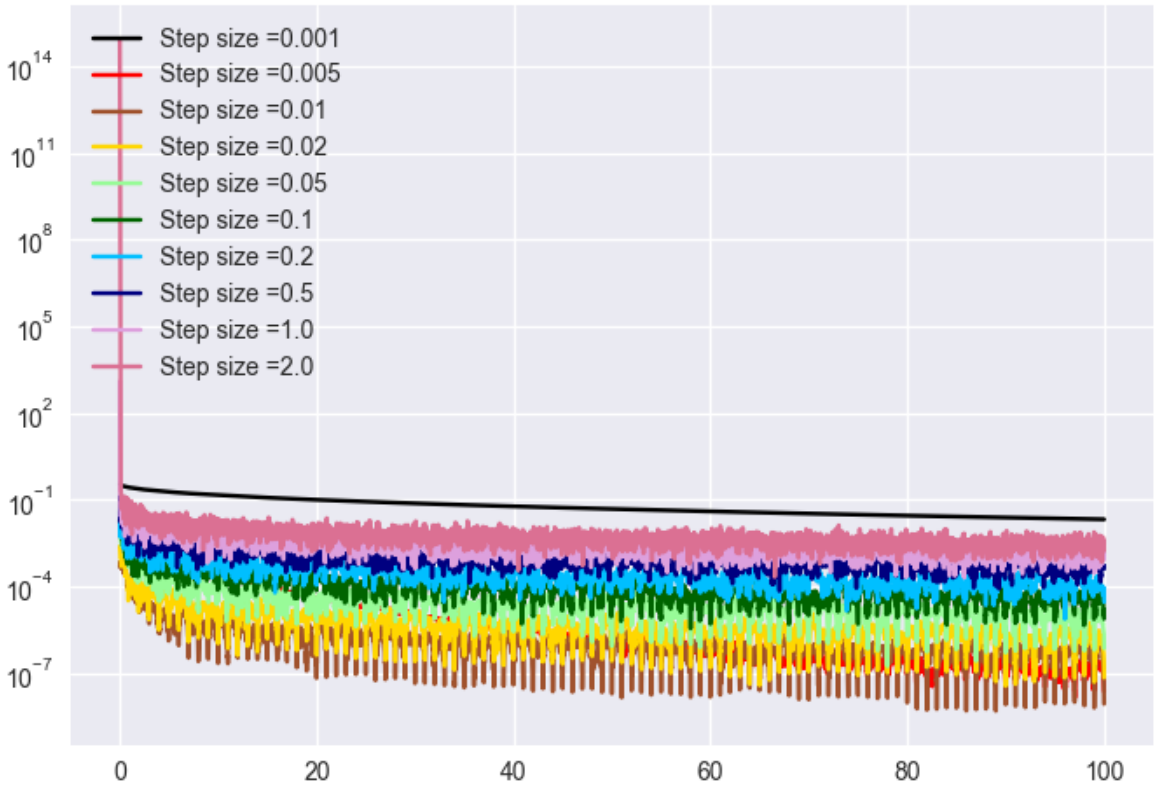

Figure 20: SGD with $\gamma/\sqrt{t}$ decaying stepsize, for Least squares

4. For SGD with $\gamma/\sqrt{t}$ decaying stepsize (Averaged), in LSR, chosen γ=0.05

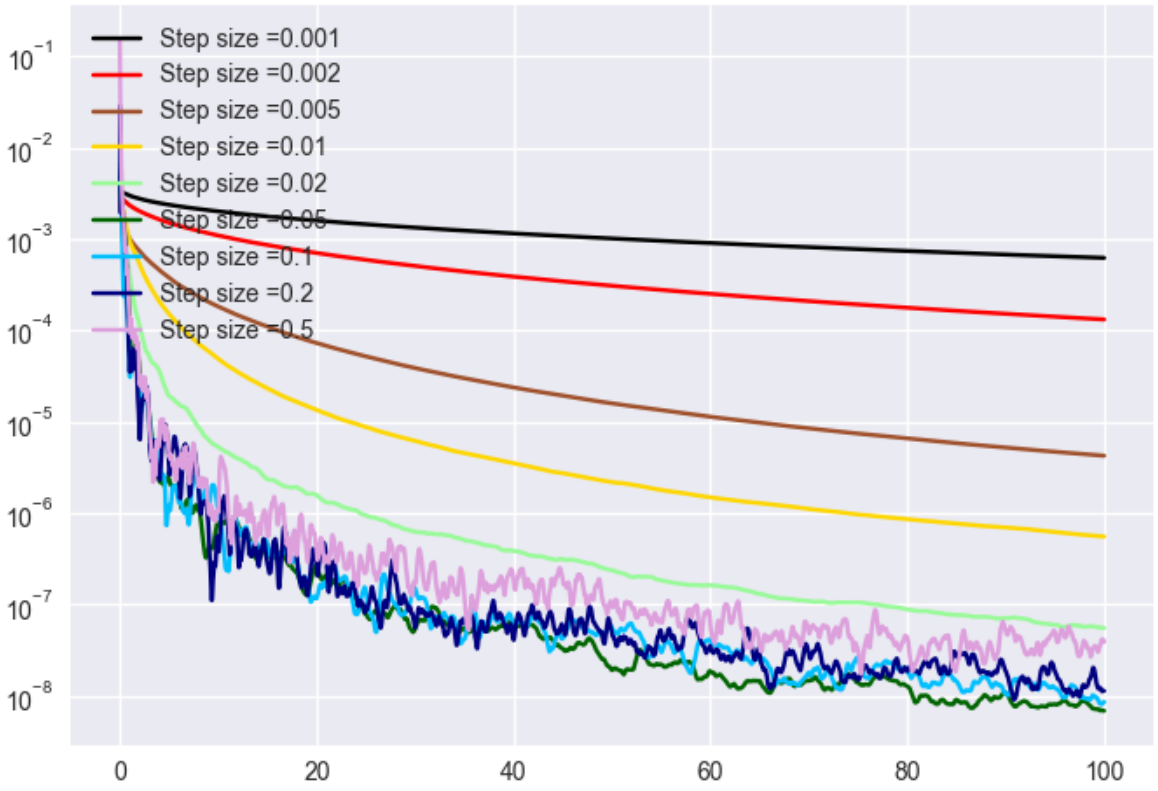

Figure 21: SGD with $\gamma/\sqrt{t}$ decaying stepsize (averaged), for Least squares

5. For SGD1/2 in LSR, burnin = 10000 iter (or 2 epochs)

Figure 22: SGD1/2 for Least squares

2.2.2 Logistic Regression

1. For standard SGD in Logistic Regression, step-size chosen=0.05

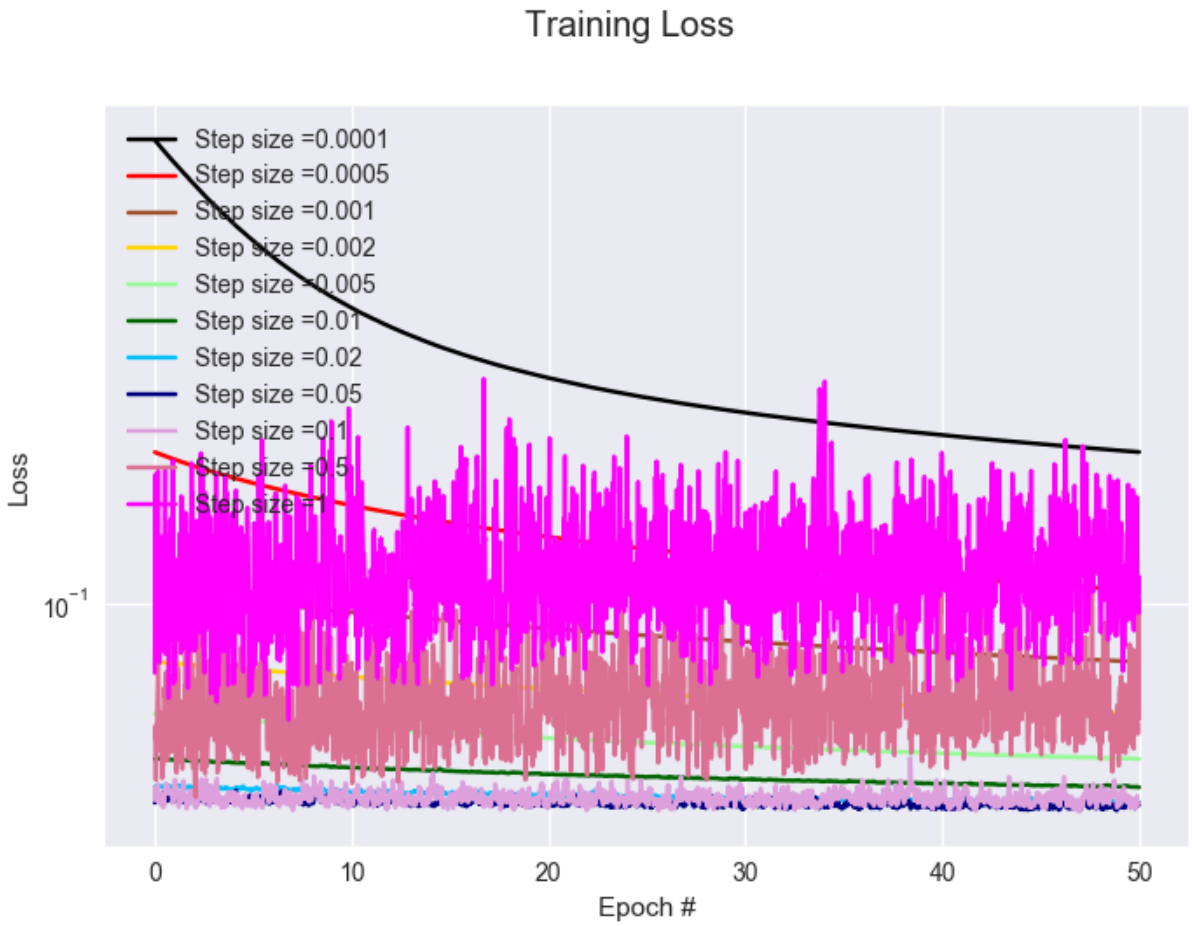

Figure 23: Vanilla SGD for Logistic Regression

2. For average SGD in Logistic Regression, step-size chosen=0.5

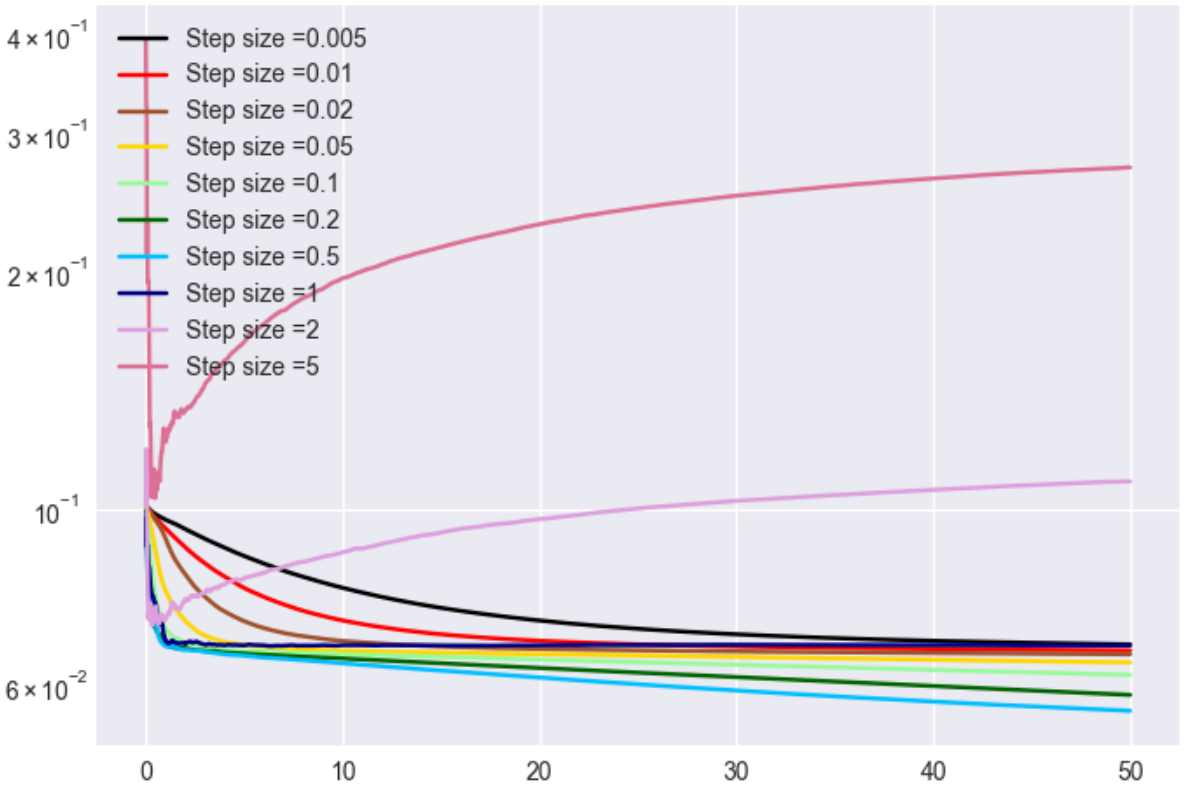

Figure 24: Average SGD for Logistic Regression

3. For SGD with $\gamma/\sqrt{t}$ decaying stepsize, in Logistic Regression, chosen γ=20

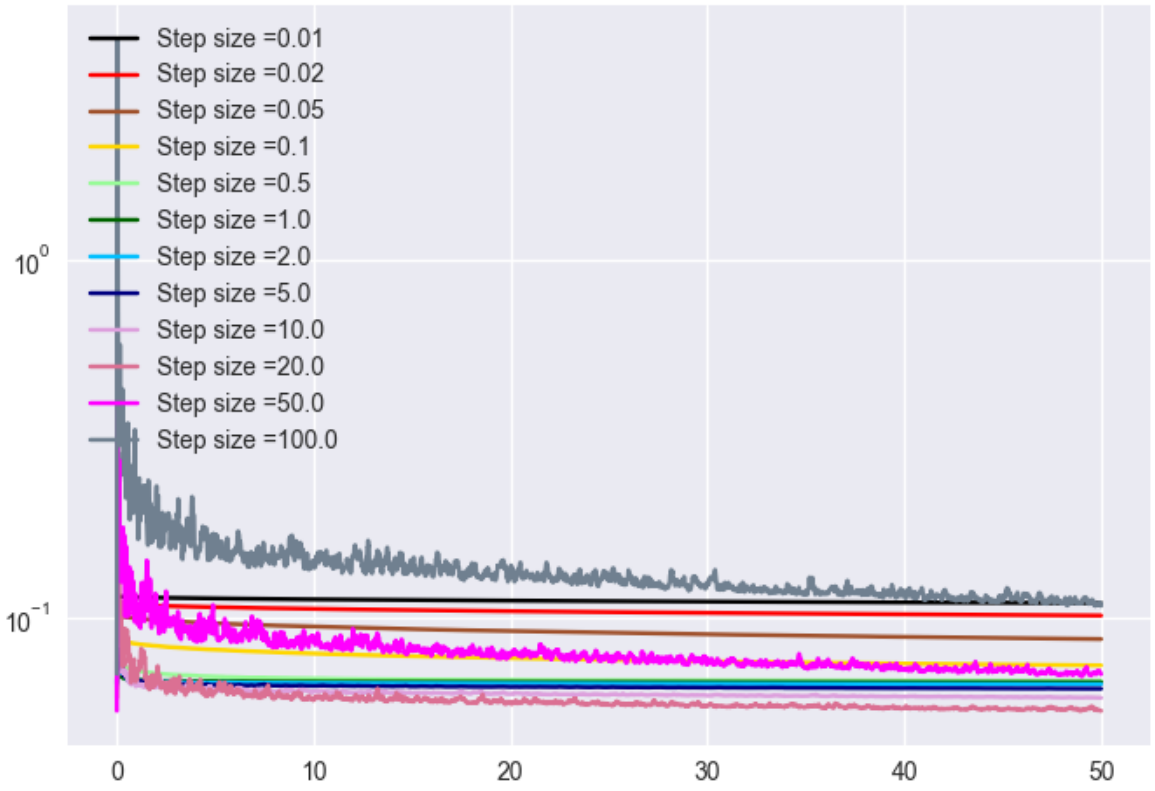

Figure 25: SGD with $\gamma/\sqrt{t}$ decaying stepsize, for Logistic Regression

4. For SGD with $\gamma/\sqrt{t}$ decaying stepsize (Averaged), in Logistic Regression, chosen γ=10

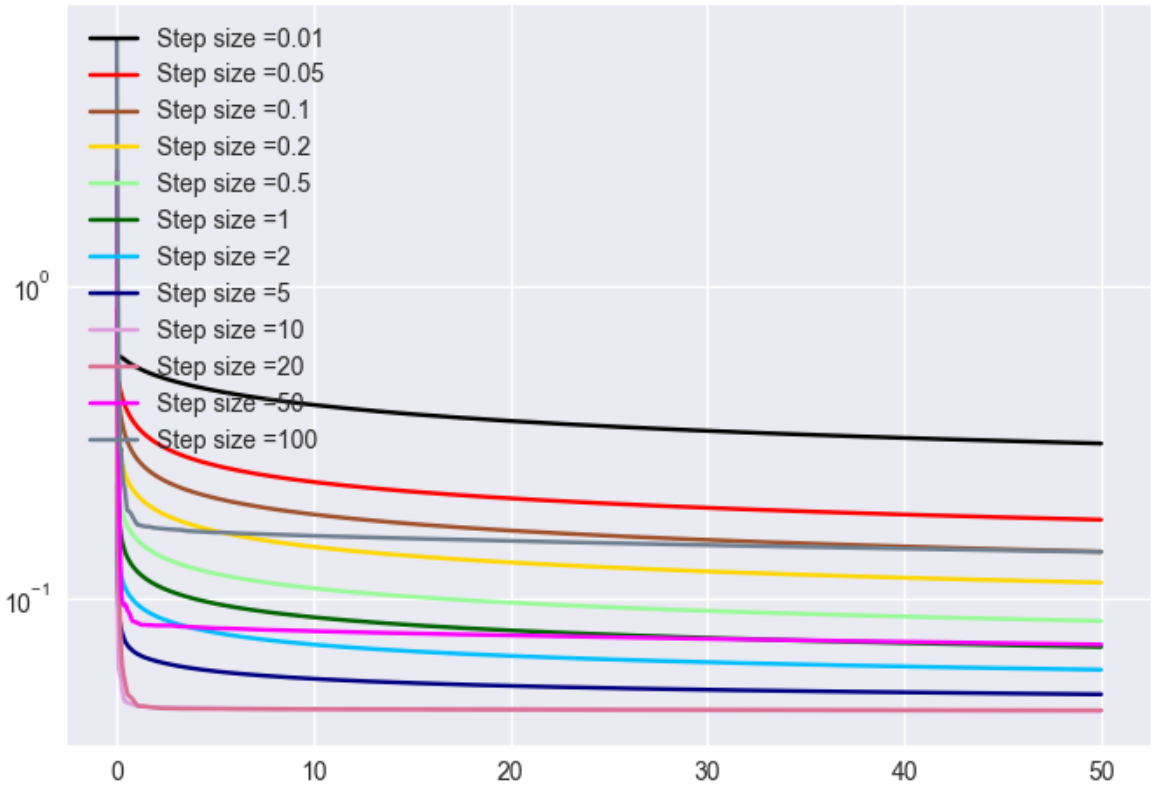

Figure 26: SGD with $\gamma/\sqrt{t}$ decaying stepsize (averaged), for Logistic Regression

5. For SGD1/2 in Logistic Regression, burnin = 20 iter

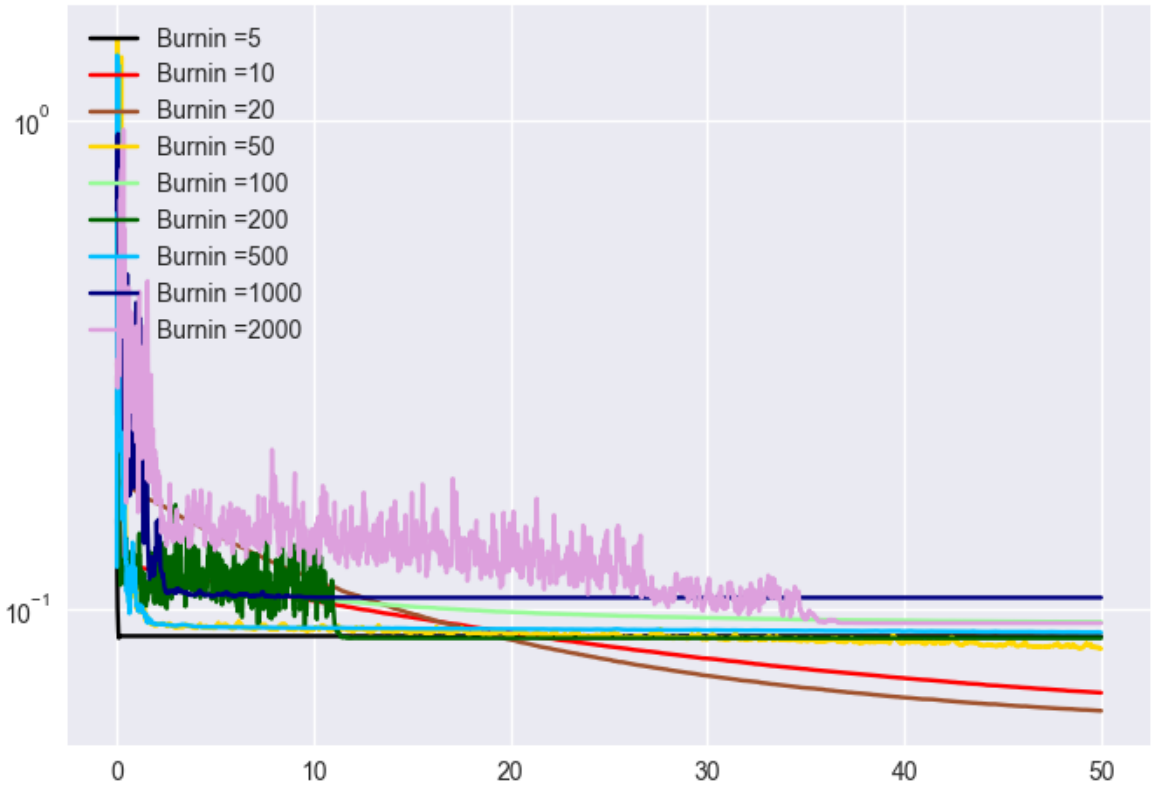

Figure 27: SGD1/2 for Logistic Regression

3 Week of October 29th, 2018

I implemented and compared the performance of different types of SGD, on artificially generated dataset with the following parameters:

- Feature dimension $p = 10$, $SNR = 2$ where SNR=var(x)/(p*var(y given x)).
- weight vector w is fixed as $w_j = 10 * exp(-0.75j)$.
- number of data points N=5000, with each $x_i \sim N(0, I), y_i \sim N(w^T x_i, \sigma^2)$

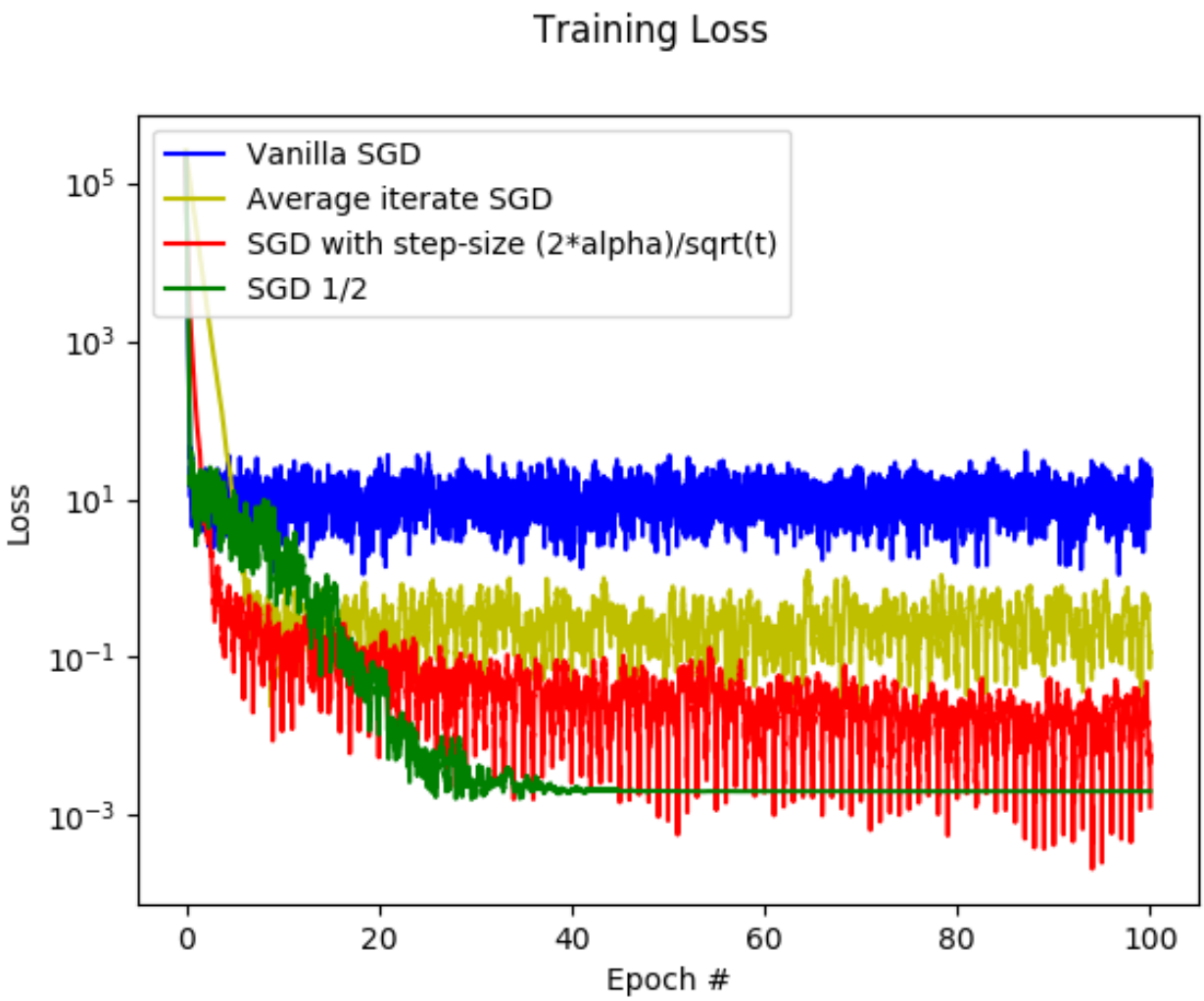

Figure 28:

As clear from the figure, performance of SGD 1/2 is comparable to SGD with $1/\sqrt{t}$ step size.
Issues:

- We know that the sum of dot product of gradients will eventually get a negative value, but in the initial epochs, the value of the dot product of the gradients is very large, and it so happens that if I start adding the dot product of gradients from the first epoch, the value becomes so large that it does not go negative for even 200 epochs. I chose to start summing from epoch 3 for this reason in the code.
- The loss values increase rapidly at the start of each epoch, then go down through the epoch. It may be a bug in the data generation, or in the training, has to be cleared up.