
Periodic Stepsize Adaptation

Chun-Nan Hsu, Han-Shen Huang and Yu-Ming Chang

Institute of Information Science

Academia Sinica, Taipei, 115, Taiwan

{chunnan, hanshen, porter}@iis.sinica.edu.tw

Previously, Bottou and LeCun [1] established that the second-order stochastic gradient descent (SGD) method can potentially achieve generalization performance as well as empirical optimum in a single pass through the training examples. However, second-order SGD requires computing the inverse of the Hessian matrix of the loss function, which is usually prohibitively expensive. Recently, we invented a new second-order SGD method, called *Periodic Stepsize Adaptation* (PSA). PSA explores a simple linear relation between the Hessian matrix and the Jacobian matrix of the mapping function. Instead of approximating Hessian, PSA approximates the Jacobian matrix which is proved to be simpler and more effective than approximating Hessian in an on-line setting. Experimental results for conditional random fields (CRF) and neural networks (NN) show that single-pass performance of PSA is very close to empirical optimum.

PSA can be derived as follows. Many machine learning methods are aimed to obtain an optimal parameter vector Θ^* that minimizes an empirical loss function. SGD solves this problem by the update rule $\Theta^{(t+1)} = \Theta^{(t)} - \eta \bullet G$, where η is a vector of the step sizes and G is the gradient of the loss function given a current small batch of training examples. Now assigning RHS as a mapping function M and the problem becomes to solve $\Theta^* = M(\Theta^*)$ by fixed-point iteration. Let the Jacobian matrix $\mathbf{J} := M'(\Theta^*)$. Previously, we [2] found an effective method to estimate the i -th eigenvalue of \mathbf{J} by

$$\gamma_i^{(t)} := \frac{|[M(\Theta^{(t)})]_i - \theta_i^{(t)}|}{|\theta_i^{(t)} - \theta_i^{(t-1)}|}. \quad (1)$$

Newton's method solves Θ^* by $\Theta^{(t+1)} = \Theta^{(t)} - \mathbf{H}^{-1}G$, where \mathbf{H} is the Hessian matrix of the loss function. However, it is prohibitively expensive to compute \mathbf{H}^{-1} . Instead, we can approximate \mathbf{H}^{-1} with its eigenvalues. Since $\text{eig}(\mathbf{I} - \eta\mathbf{H}) = \text{eig}(\mathbf{M}') = \text{eig}(\mathbf{J}) \approx \gamma$, $\text{eig}(\mathbf{H}^{-1}) \approx \frac{\eta}{1-\gamma}$, which implies an update rule for the step size by $\eta_i^{(t+1)} = \frac{\eta_i^{(t)}}{1-\gamma_i^{(t)}}$.

However, due to stochasticity of the mapping, estimating $\gamma_i^{(t)}$ with Equation (1) by consecutive $\Theta^{(t)}$ may yield inaccurate estimations. To make the mapping more stationary, we apply SGD with a fixed step size η for $2b$ iterations to obtain $\Theta^{(t)}$, $\Theta^{(t+b)}$, and $\Theta^{(t+2b)}$, then use them in Equation (1) to obtain $\gamma_i^{(t)}$ and update η every $2b$ SGD iterations. $b = 10$ works effectively in our experiments. For numerical stability, we constrain the range of $\frac{1}{1-\gamma}$ when updating η . Clearly, the time complexity of Equation (1) is $O(d)$, where d is the dimension of Θ , and the per iteration cost of PSA is $O(\frac{d}{b})$. Optimization of the implementation that only updates those affected dimensions can drastically reduce the per iteration cost further.

We implemented PSA to train CRF and compared PSA with state-of-the-art algorithms for three large-scale entity-recognition tasks: CoNLL-2000 chunking task, BioNLP/NLPBA-2004¹ bio-entity recognition task, and BioCreative 2 gene mention tagging task. These tasks have been used in competitions and the performance was measured by F-scores for hold-out sets. The best performing CRF models for these tasks used millions of parameters estimated from training corpora containing tens thousands of sentences.

¹The $>85\%$ F-score and learning curve results of SMD for this task reported in [5] is due to a bug that includes true labels as a feature, according to the author.

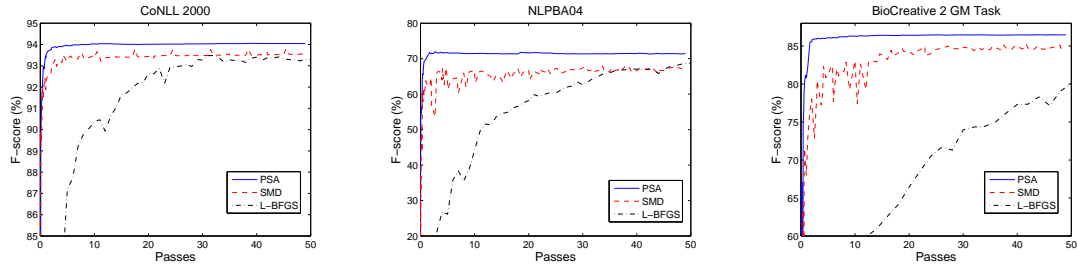


Figure 1: Learning curves of PSA, SMD and L-BFGS on CoNLL 2000 (left), BioNLP/NLPBA 2004 (center) and BioCreative II (right) data sets.

Method	CoNLL 2000	BIONLP/NLPBA04	BioCreative 2
L-BFGS	93.5% in 50 passes converge at 94%	67% in 40 passes 70% after 70 passes	85% in 156 passes 86% after 200 passes
SMD	93.6% in 7.7 passes converge at \sim 93.6%	67% in 13 passes vibrate between 66 and 68%	84% in 16.67 passes vibrate between 84 and 85%
PSA	93.6% in 1.12 passes 94% in 8 passes converge at 94.05%	67% in 0.54 passes 70% in 1.01 passes 71.4% after 1.68 passes	85% in 1.66 passes 86% in 3 passes 86.46% after 4 passes

Table 1: Comparison of F-scores and passes required by different CRF training methods for three named-entity recognition tasks. Results of L-BFGS were used to serve as empirical optima.

Figure 1 shows the comparison of the learning curves of two second-order SGD methods, PSA and SMD [5], and a batch algorithm L-BFGS, the *de facto* standard for training CRF. Results of L-BFGS were included to serve as empirical optima. The learning curves are defined as the function of F-score given the number of passes (i.e., epochs) through the entire training sets. Table 1 summarizes their performance. The results show that for all three tasks, PSA outperforms SMD by an order of magnitude in terms of the number of passes. Their new method, oL-BFGS [4], fared even worse, taking 30 passes for CoNLL 2000 and requiring a very large batch size. By contrast, PSA achieved F-scores about as good as L-BFGS in a single pass with the batch size equal to one for all tasks.

To test if PSA works for non-convex problems, we implemented PSA to train a 124-80-1 feed-forward neural network with ADULT data set (a9a, 32.5K training examples from LIBSVM). In this preliminary study, PSA reduced the error rate to 17.83% in 0.55 passes and converged at 17.65% in 1.1 passes, while SGD with momentum 0.1 vibrated between 16.08 and 26.52% after 5 passes.

PSA provides a practical solution to accomplish near-optimal performance of second-order SGD predicted theoretically in [1]. Our future work is to test and tune PSA implementations for more models and compare with other SGD methods such as TONGA. For details please refer to [3].

References

- [1] Léon Bottou and Yann LeCun. Large scale online learning. In *NIPS-04*.
- [2] Chun-Nan Hsu, Han-Shen Huang, Bo-Hou Yang, and Yu-Ming Chang. Global and componentwise extrapolations for accelerating training of Bayesian networks and conditional random fields. Technical Report TR-IIS-07-013, Institute of Information Science, Academia Sinica, Taiwan, 2007. Submitted to *JMLR*.
- [3] Han-Shen Huang, Yu-Ming Chang, and Chun-Nan Hsu. Training conditional random fields by periodic step size adaptation for large-scale text mining. In *IEEE ICDM-07*.
- [4] Nicol N. Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-Newton method for online convex optimization. In *AISat-07*.
- [5] S.V.N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *ICML-06*.