Supplementary Material for "A Progressive Batching L-BFGS Method for Machine Learning"

Raghu Bollapragada [1] **Dheevatsa Mudigere** [2] **Jorge Nocedal** [1] **Hao-Jun Michael Shi** [1] **Ping Tak Peter Tang** [3]

A. Initial Step Length Derivation

To establish our results, recall that the stochastic quasi-Newton method is defined as

$$x_{k+1} = x_k - \alpha_k H_k g_k^{S_k}, \tag{1}$$

where the batch (or subsampled) gradient is given by

$$g_k^{S_k} = \nabla F_{S_k}(x_k) = \frac{1}{|S_k|} \sum_{i \in S_k} \nabla F_i(x_k), \tag{2}$$

and the set $S_k \subset \{1, 2, \cdots\}$ indexes data points (y^i, z^i). The algorithm selects the Hessian approximation H_k through quasi-Newton updating prior to selecting the new sample S_k to define the search direction p_k. We will use $\mathbb{E}_k$ to denote the conditional expectation at x_k and use $\mathbb{E}$ to denote the total expectation.

The primary theoretical mechanism for determining batch sizes is the exact variance inner product quasi-Newton (IPQN) test, which is defined as

$$\frac{\mathbb{E}_k\left[\left((H_k \nabla F(x_k))^T (H_k g_k^i) - \|H_k \nabla F(x_k)\|^2\right)^2\right]}{|S_k|} \leq \theta^2 \|H_k \nabla F(x_k)\|^4. \tag{3}$$

We establish the inequality used to determine the initial steplength α_k for the stochastic line search.

Lemma A.1. *Assume that F is continuously differentiable with Lipschitz continuous gradient with Lipschitz constant L. Then*

$$\mathbb{E}_k\left[F(x_{k+1})\right] \leq F(x_k) - \alpha_k \nabla F(x_k)^T H_k^{1/2} W_k H_k^{1/2} \nabla F(x_k),$$

where

$$W_k = \left(I - \frac{L\alpha_k}{2}\left(1 + \frac{\mathrm{Var}\{H_k g_k^i\}}{|S_k| \|H_k \nabla F(x_k)\|^2}\right) H_k\right),$$

and $\mathrm{Var}\{H_k g_k^i\} = \mathbb{E}_k\left[\|H_k g_k^i - H_k \nabla F(x_k)\|^2\right]$.

Proof. By Lipschitz continuity of the gradient, we have that

$$\begin{aligned}
\mathbb{E}_k\left[F(x_{k+1})\right] &\leq F(x_k) - \alpha_k \nabla F(x_k)^T H_k \mathbb{E}_k\left[g_k^{S_k}\right] + \frac{L\alpha_k^2}{2} \mathbb{E}_k\left[\|H_k g_k^{S_k}\|^2\right] \\
&= F(x_k) - \alpha_k \nabla F(x_k)^T H_k \nabla F(x_k) + \frac{L\alpha_k^2}{2}\left(\|H_k \nabla F(x_k)\|^2 + \mathbb{E}_k\left[\|H_k g_k^{S_k} - H_k \nabla F(x_k)\|^2\right]\right) \\
&\leq F(x_k) - \alpha_k \nabla F(x_k)^T H_k \nabla F(x_k) + \frac{L\alpha_k^2}{2}\left(\|H_k \nabla F(x_k)\|^2 + \frac{\mathrm{Var}\{H_k g_k^i\}}{|S_k| \|H_k \nabla F(x_k)\|^2} \|H_k \nabla F(x_k)\|^2\right) \\
&= F(x_k) - \alpha_k \nabla F(x_k)^T H_k^{1/2}\left(I - \frac{L\alpha_k}{2}\left(1 + \frac{\mathrm{Var}\{H_k g_k^i\}}{|S_k| \|H_k \nabla F(x_k)\|^2}\right) H_k\right) H_k^{1/2} \nabla F(x_k) \\
&= F(x_k) - \alpha_k \nabla F(x_k)^T H_k^{1/2} W_k H_k^{1/2} \nabla F(x_k).
\end{aligned}$$

□

B. Convergence Analysis

For the rest of our analysis, we make the following two assumptions.

Assumptions B.1. *The orthogonality condition is satisfied for all k, i.e.,*

$$\frac{\mathbb{E}_k\left[\left\|H_k g_k^i - \frac{(H_k g_k^i)^T (H_k \nabla F(x_k))}{\|H_k \nabla F(x_k)\|^2} H_k \nabla F(x_k)\right\|^2\right]}{|S_k|} \leq \nu^2 \|H_k \nabla F(x_k)\|^2, \tag{4}$$

for some large $\nu > 0$.

Assumptions B.2. *The eigenvalues of H_k are contained in an interval in $\mathbb{R}^+$, i.e., for all k there exist constants $\Lambda_2 \geq \Lambda_1 > 0$ such that*

$$\Lambda_1 I \preceq H_k \preceq \Lambda_2 I. \tag{5}$$

Condition (4) ensures that the stochastic quasi-Newton direction is bounded away from orthogonality to $-H_k \nabla F(x_k)$, with high probability, and prevents the variance in the individual quasi-Newton directions to be too large relative to the variance in the individual quasi-Newton directions along $-H_k \nabla F(x_k)$. Assumption B.2 holds, for example, when F is convex and a regularization parameter is included so that any subsampled Hessian $\nabla^2 F_S(x)$ is positive definite. It can also be shown to hold in the non-convex case by applying cautious BFGS updating; e.g. by updating H_k only when $y_k^T s_k \geq \epsilon \|s_k\|_2^2$ where $\epsilon > 0$ is a predetermined constant (Berahas et al., 2016).

We begin by establishing a technical descent lemma.

Lemma B.3. *Suppose that F is twice continuously differentiable and that there exists a constant $L > 0$ such that*

$$\nabla^2 F(x) \preceq LI, \quad \forall x \in \mathbb{R}^d. \tag{6}$$

Let $\{x_k\}$ be generated by iteration (1) for any x_0, where $|S_k|$ is chosen by the (exact variance) inner product quasi-Newton test (3) for given constant $\theta > 0$ and suppose that assumptions (B.1) and (B.2) hold. Then, for any k,

$$\mathbb{E}_k\left[\|H_k g_k^{S_k}\|^2\right] \leq (1 + \theta^2 + \nu^2) \|H_k \nabla F(x_k)\|^2. \tag{7}$$

Moreover, if α_k satisfies

$$\alpha_k = \alpha \leq \frac{1}{(1 + \theta^2 + \nu^2) L \Lambda_2}, \tag{8}$$

we have that

$$\mathbb{E}_k[F(x_{k+1})] \leq F(x_k) - \frac{\alpha}{2} \|H_k^{1/2} \nabla F(x_k)\|^2. \tag{9}$$

Proof. By Assumption (B.1), the orthogonality condition, we have that

$$\begin{aligned}\mathbb{E}_k\left[\left\|H_k g_k^{S_k} - \frac{(H_k g_k^{S_k})^T (H_k \nabla F(x_k))}{\|H_k \nabla F(x_k)\|^2} H_k \nabla F(x_k)\right\|^2\right] &\leq \frac{\mathbb{E}_k\left[\left\|H_k g_k^i - \frac{(H_k g_k^i)^T (H_k \nabla F(x_k))}{\|H_k \nabla F(x_k)\|^2} H_k \nabla F(x_k)\right\|^2\right]}{|S_k|} \\ &\leq \nu^2 \|H_k \nabla F(x_k)\|^2.\end{aligned} \tag{10}$$

Now, expanding the left hand side of inequality (10), we get

$$\begin{aligned}&\mathbb{E}_k\left[\left\|H_k g_k^{S_k} - \frac{(H_k g_k^{S_k})^T (H_k \nabla F(x_k))}{\|H_k \nabla F(x_k)\|^2} H_k \nabla F(x_k)\right\|^2\right] \\ &= \mathbb{E}_k\left[\|H_k g_k^{S_k}\|^2\right] - \frac{2\mathbb{E}_k\left[\left((H_k g_k^{S_k})^T (H_k \nabla F(x_k))\right)^2\right]}{\|H_k \nabla F(x_k)\|^2} + \frac{\mathbb{E}_k\left[\left((H_k g_k^{S_k})^T (H_k \nabla F(x_k))\right)^2\right]}{\|H_k \nabla F(x_k)\|^2} \\ &= \mathbb{E}_k\left[\|H_k g_k^{S_k}\|^2\right] - \frac{\mathbb{E}_k\left[\left((H_k g_k^{S_k})^T (H_k \nabla F(x_k))\right)^2\right]}{\|H_k \nabla F(x_k)\|^2} \\ &\leq \nu^2 \|H_k \nabla F(x_k)\|^2.\end{aligned}$$

Therefore, rearranging gives the inequality

$$\mathbb{E}_k\left[\|H_k g_k^{S_k}\|^2\right] \leq \frac{\mathbb{E}_k\left[\left((H_k g_k^{S_k})^T(H_k \nabla F(x_k))\right)^2\right]}{\|H_k \nabla F(x_k)\|^2} + \nu^2 \|H_k \nabla F(x_k)\|^2. \tag{11}$$

To bound the first term on the right side of this inequality, we use the inner product quasi-Newton test; in particular, $|S_k|$ satisfies

$$\begin{aligned}\mathbb{E}_k\left[\left((H_k \nabla F(x_k))^T(H_k g_k^{S_k})) - \|H_k \nabla F(x_k)\|^2\right)^2\right] &\leq \frac{\mathbb{E}_k\left[\left((H_k \nabla F(x_k))^T(H_k g_k^i) - \|H_k \nabla F(x_k)\|^2\right)^2\right]}{|S_k|} \\ &\leq \theta^2 \|H_k \nabla F(x_k)\|^4, \end{aligned} \tag{12}$$

where the second inequality holds by the IPQN test. Since

$$\mathbb{E}_k\left[\left((H_k \nabla F(x_k))^T(H_k g_k^{S_k}) - \|H_k \nabla F(x_k)\|^2\right)^2\right] = \mathbb{E}_k\left[\left((H_k \nabla F(x_k))^T(H_k g_k^{S_k})\right)^2\right] - \|H_k \nabla F(x_k)\|^4, \tag{13}$$

we have

$$\begin{aligned}\mathbb{E}_k\left[\left((H_k g_k^{S_k})^T(H_k \nabla F(x_k))\right)^2\right] &\leq \|H_k \nabla F(x_k)\|^4 + \theta^2 \|H_k \nabla F(x_k)\|^4 \\ &= (1+\theta^2)\|H_k \nabla F(x_k)\|^4, \end{aligned} \tag{14}$$

by (12) and (13). Substituting (14) into (11), we get the following bound on the length of the search direction:

$$\mathbb{E}_k\left[\|H_k g_k^{S_k}\|^2\right] \leq (1+\theta^2+\nu^2)\|H_k \nabla F(x_k)\|^2,$$

which proves (7). Using this inequality, Assumption B.2, and bounds on the Hessian and steplength (6) and (8), we have

$$\begin{aligned}\mathbb{E}_k[F(x_{k+1})] &\leq F(x_k) - \mathbb{E}_k\left[\alpha (H_k g_k^{S_k})^T \nabla F(x_k)\right] + \mathbb{E}_k\left[\frac{L\alpha^2}{2}\|H_k g_k^{S_k}\|^2\right] \\ &= F(x_k) - \alpha \nabla F(x_k)^T H_k \nabla F(x_k) + \frac{L\alpha^2}{2}\mathbb{E}_k[\|H_k g_k^{S_k}\|^2] \\ &\leq F(x_k) - \alpha \nabla F(x_k)^T H_k \nabla F(x_k) + \frac{L\alpha^2}{2}(1+\theta^2+\nu^2)\|H_k \nabla F(x_k)\|^2 \\ &= F(x_k) - \alpha (H_k^{1/2}\nabla F(x_k))^T \left(I - \frac{L\alpha(1+\theta^2+\nu^2)}{2} H_k\right) H_k^{1/2} \nabla F(x_k) \\ &\leq F(x_k) - \alpha \left(1 - \frac{L\Lambda_2 \alpha(1+\theta^2+\nu^2)}{2}\right) \|H_k^{1/2}\nabla F(x_k)\|^2 \\ &\leq F(x_k) - \frac{\alpha}{2}\|H_k^{1/2}\nabla F(x_k)\|^2. \end{aligned}$$

□

We now show that the stochastic quasi-Newton iteration (1) with a fixed steplength α is linearly convergent when F is strongly convex. In the following discussion, x^* denotes the minimizer of F.

Theorem B.4. *Suppose that F is twice continuously differentiable and that there exist constants $0 < \mu \leq L$ such that*

$$\mu I \preceq \nabla^2 F(x) \preceq LI, \quad \forall x \in \mathbb{R}^d. \tag{15}$$

Let $\{x_k\}$ be generated by iteration (1), for any x_0, where $|S_k|$ is chosen by the (exact variance) inner product quasi-Newton test (3) and suppose that the assumptions (B.1) and (B.2) hold. Then, if α_k satisfies (8) we have that

$$\mathbb{E}[F(x_k) - F(x^*)] \leq \rho^k (F(x_0) - F(x^*)), \tag{16}$$

where x^* denotes the minimizer of F, and $\rho = 1 - \mu\Lambda_1\alpha$.

Proof. It is well-known (Bertsekas et al., 2003) that for strongly convex functions,

$$\|\nabla F(x_k)\|^2 \geq 2\mu[F(x_k) - F(x^*)].$$

Substituting this into (9) and subtracting $F(x^*)$ from both sides and using Assumption B.2, we obtain

$$\begin{aligned}\mathbb{E}_k[F(x_{k+1}) - F(x^*)] &\leq F(x_k) - F(x^*) - \frac{\alpha}{2}\|H_k^{1/2}\nabla F(x_k)\|^2 \\ &\leq F(x_k) - F(x^*) - \frac{\alpha}{2}\Lambda_1\|\nabla F(x_k)\|^2 \\ &\leq (1 - \mu\Lambda_1\alpha)(F(x_k) - F(x^*)).\end{aligned}$$

The theorem follows from taking total expectation.

□

We now consider the case when F is nonconvex and bounded below.

Theorem B.5. *Suppose that F is twice continuously differentiable and bounded below, and that there exists a constant $L > 0$ such that*

$$\nabla^2 F(x) \preceq LI, \quad \forall x \in \mathbb{R}^d. \tag{17}$$

Let $\{x_k\}$ be generated by iteration (1), for any x_0, where $|S_k|$ is chosen by the (exact variance) inner product quasi-Newton test (3) and suppose that the assumptions (B.1) and (B.2) hold. Then, if α_k satisfies (8), we have

$$\lim_{k\to\infty} \mathbb{E}[\|\nabla F(x_k)\|^2] \to 0. \tag{18}$$

Moreover, for any positive integer T we have that

$$\min_{0\leq k\leq T-1} \mathbb{E}[\|\nabla F(x_k)\|^2] \leq \frac{2}{\alpha T\Lambda_1}(F(x_0) - F_{min}),$$

where F_{min} is a lower bound on F in $\mathbb{R}^d$.

Proof. From Lemma B.3 and by taking total expectation, we have

$$\mathbb{E}[F(x_{k+1})] \leq \mathbb{E}[F(x_k)] - \frac{\alpha}{2}\mathbb{E}[\|H_k^{1/2}\nabla F(x_k)\|^2],$$

and hence

$$\mathbb{E}[\|H_k^{1/2}\nabla F(x_k)\|^2] \leq \frac{2}{\alpha}\mathbb{E}[F(x_k) - F(x_{k+1})].$$

Summing both sides of this inequality from $k = 0$ to $T - 1$, and since F is bounded below by F_{min}, we get

$$\sum_{k=0}^{T-1} \mathbb{E}[\|H_k^{1/2}\nabla F(x_k)\|^2] \leq \frac{2}{\alpha}\mathbb{E}[F(x_0) - F(x_{\mathrm{T}})] \leq \frac{2}{\alpha}[F(x_0) - F_{min}].$$

Using the bound on the eigenvalues of H_k and taking limits, we obtain

$$\Lambda_1 \lim_{T\to\infty} \sum_{k=0}^{T-1} \mathbb{E}[\|\nabla F(x_k)\|^2] \leq \lim_{T\to\infty} \sum_{k=0}^{T-1} \mathbb{E}[\|H_k^{1/2}\nabla F(x_k)\|^2] < \infty,$$

which implies (18). We can also conclude that

$$\min_{0\leq k\leq T-1} \mathbb{E}[\|\nabla F(x_k)\|^2] \leq \frac{1}{T}\sum_{k=0}^{T} \mathbb{E}[\|\nabla F(x_k)\|^2] \leq \frac{2}{\alpha T\Lambda_1}(F(x_0) - F_{min}).$$

□

Table 1. Characteristics of all datasets used in the experiments.

Dataset	# Data Points (train; test)	# Features	# Classes	Source
`gisette`	(6,000; 1,000)	5,000	2	(Chang & Lin, 2011)
`mushrooms`	(7,311; 813)	112	2	(Chang & Lin, 2011)
`sido`	(11,410; 1,268)	4,932	2	(Guyon et al., 2008)
`ijcnn`	(35,000; 91701)	22	2	(Chang & Lin, 2011)
`spam`	(82,970; 9,219)	823,470	2	(Cormack & Lynam, 2005; Carbonetto, 2009)
`alpha`	(450,000; 50,000)	500	2	synthetic
`covertype`	(522,910; 58,102)	54	2	(Chang & Lin, 2011)
`url`	(2,156,517; 239,613)	3,231,961	2	(Chang & Lin, 2011)
`MNIST`	(60,000; 10,000)	28×28	10	(LeCun et al., 1998)
`CIFAR-10`	(50,000; 10,000)	32×32	10	(Krizhevsky, 2009)

C. Additional Numerical Experiments

C.1. Datasets

Table 1 summarizes the datasets used for the experiments. Some of these datasets divide the data into training and testing sets; for the rest, we randomly divide the data so that the training set constitutes 90% of the total.

The alpha dataset is a synthetic dataset that is available at `ftp://largescale.ml.tu-berlin.de`.

C.2. Logistic Regression Experiments

We report the numerical results on binary classification logistic regression problems on the 8 datasets given in Table 1. We plot the performance measured in terms of training error, test loss and test accuracy against gradient evaluations. We also report the behavior of the batch sizes and steplengths for both variants of the PBQN method.

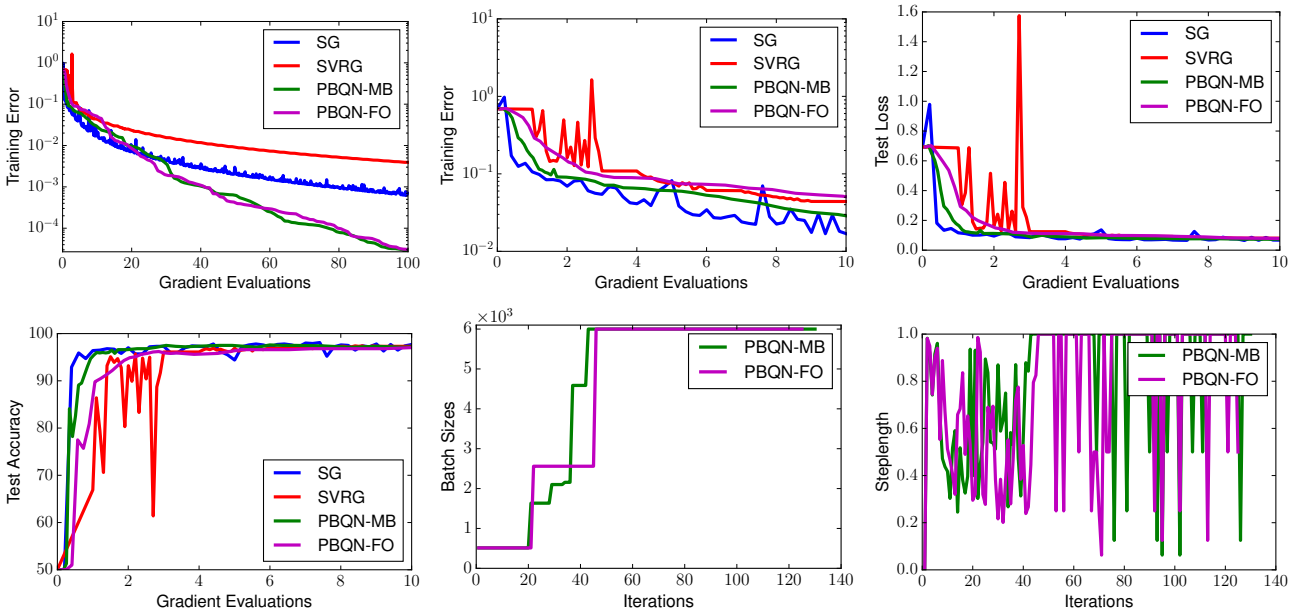

Figure 1. **gisette dataset:** Performance of the progressive batching L-BFGS methods, with multi-batch (MB) (25% overlap) and full-overlap (FO) approaches, and the SG and SVRG methods.

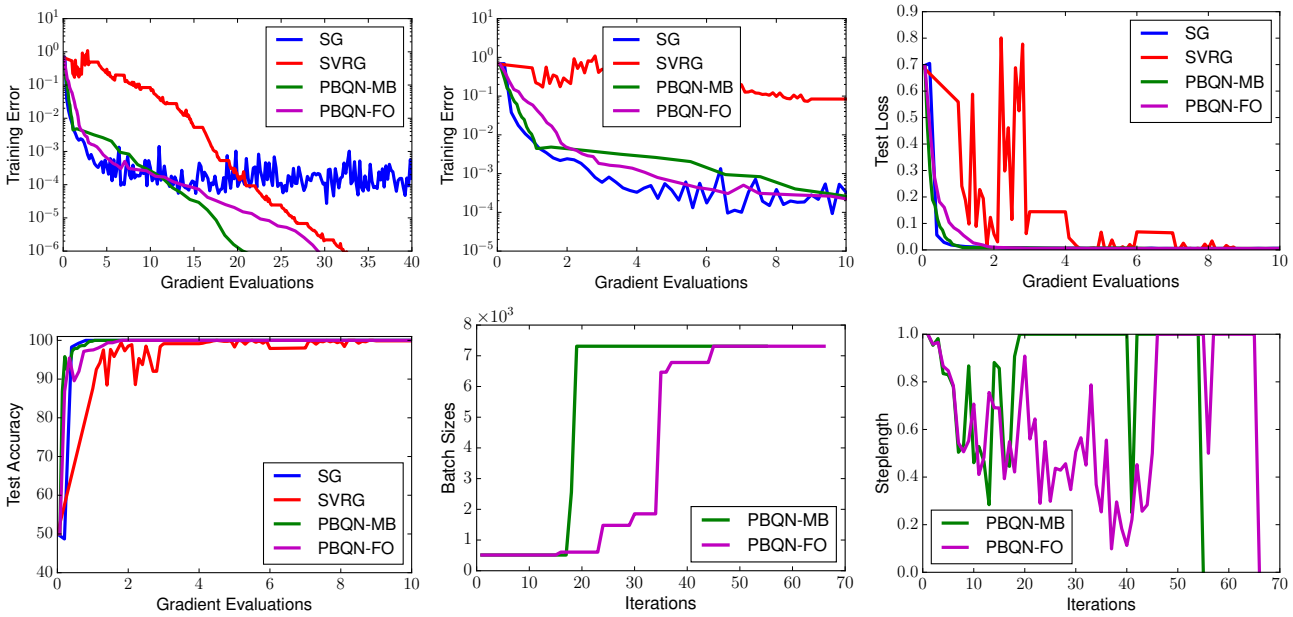

Figure 2. **mushrooms dataset:** Performance of the progressive batching L-BFGS methods, with multi-batch (MB) (25% overlap) and full-overlap (FO) approaches, and the SG and SVRG methods.

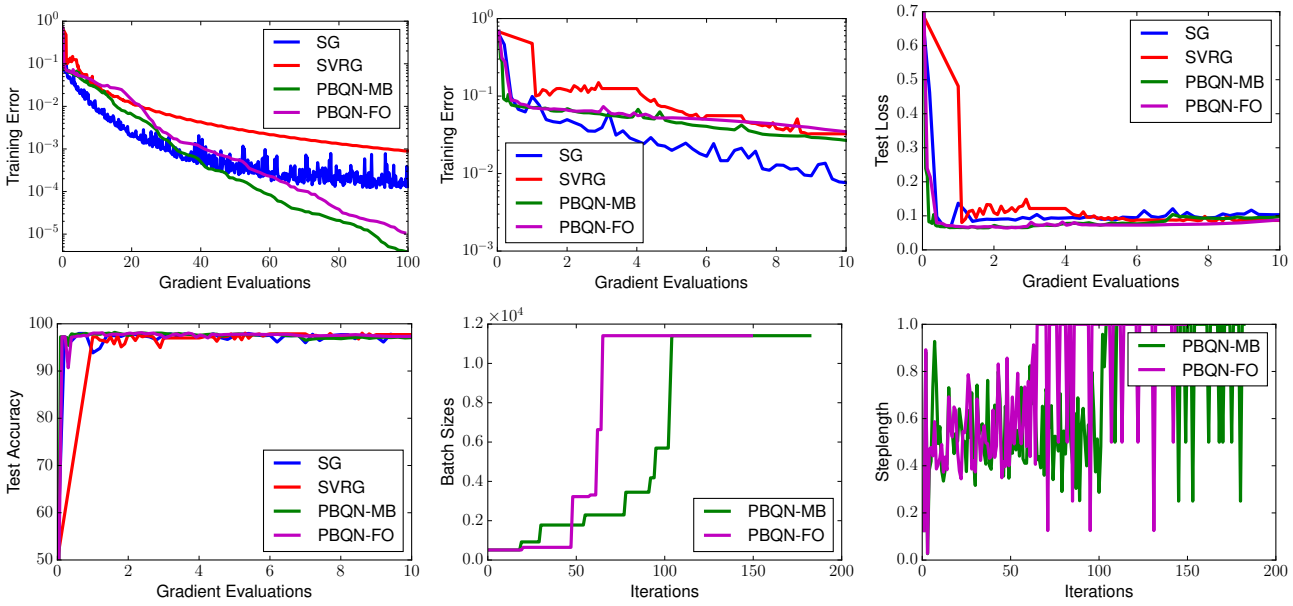

Figure 3. **sido dataset:** Performance of the progressive batching L-BFGS methods, with multi-batch (MB) (25% overlap) and full-overlap (FO) approaches, and the SG and SVRG methods.

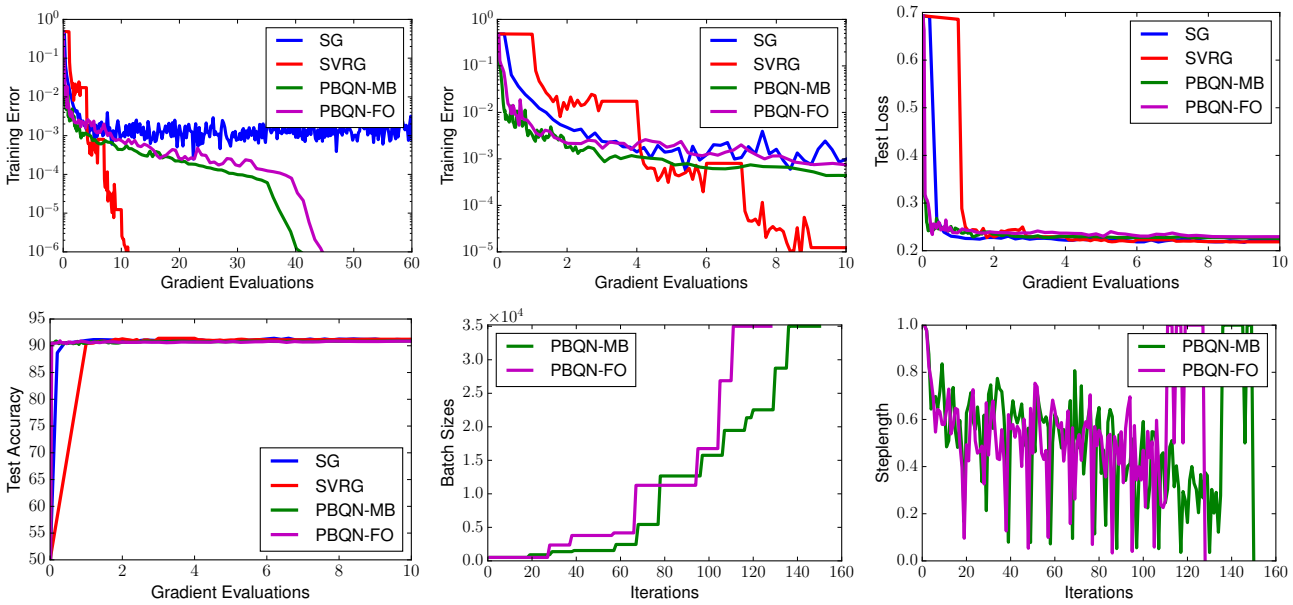

Figure 4. **ijcnn dataset:** Performance of the progressive batching L-BFGS methods, with multi-batch (MB) (25% overlap) and full-overlap(FO) approaches, and the SG and SVRG methods.

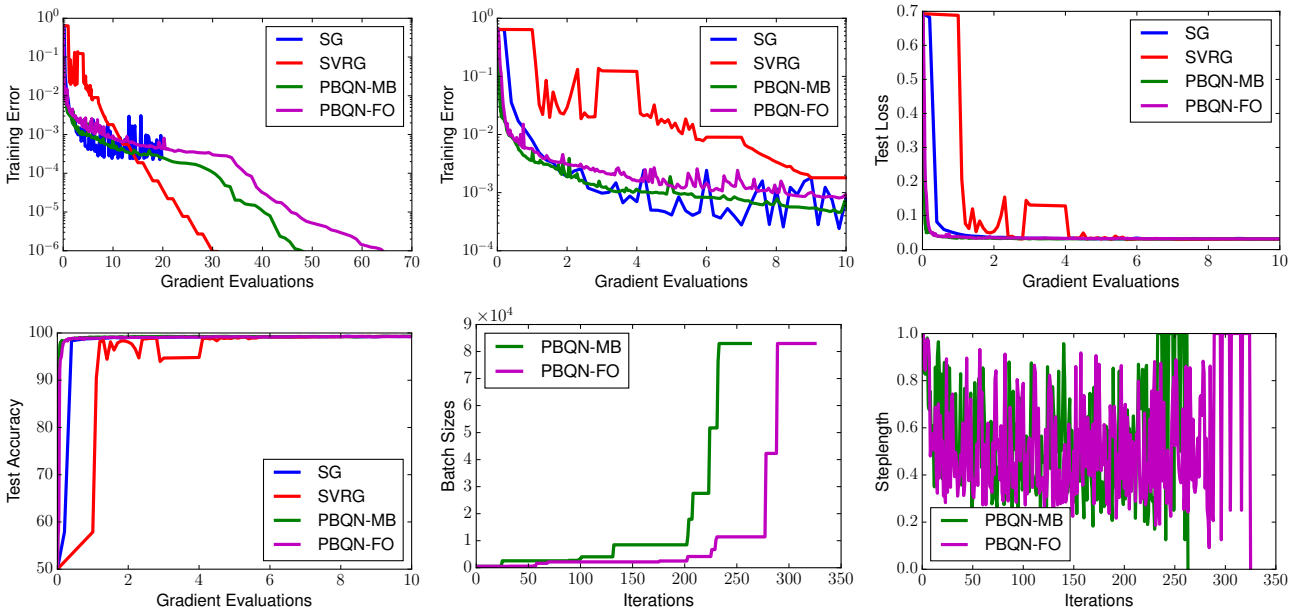

Figure 5. **spam dataset:** Performance of the progressive batching L-BFGS methods, with multi-batch (MB) (25% overlap) and full-overlap (FO) approaches, and the SG and SVRG methods.

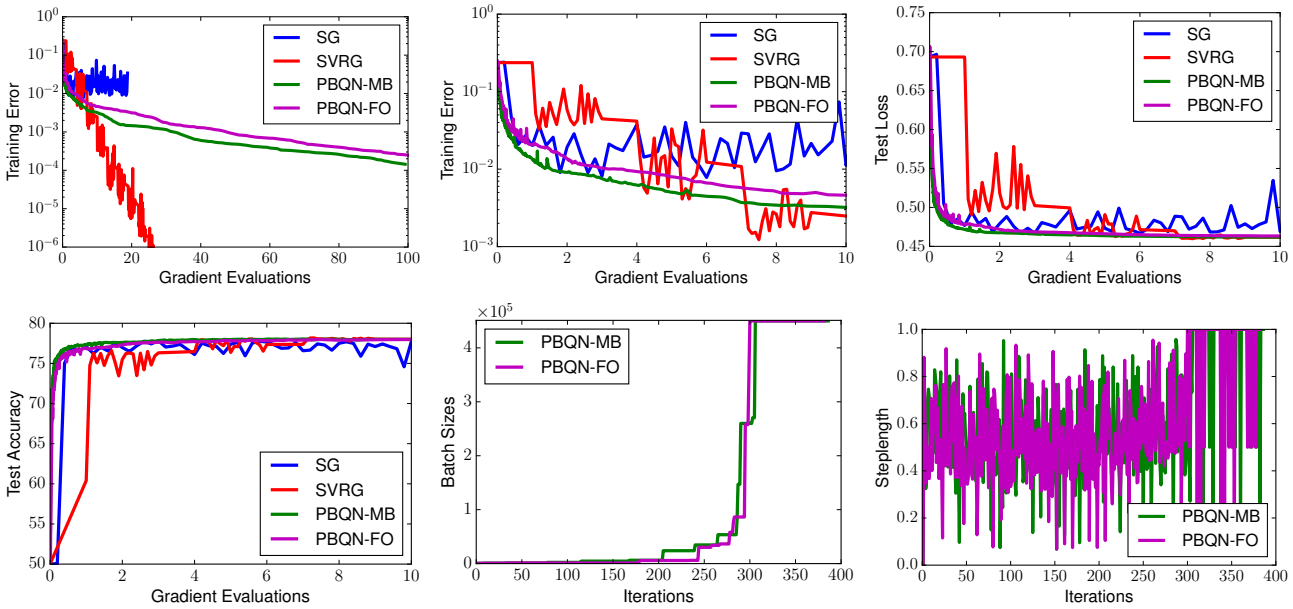

Figure 6. **alpha dataset:** Performance of the progressive batching L-BFGS methods, with multi-batch (MB) (25% overlap) and full-overlap (FO) approaches, and the SG and SVRG methods.

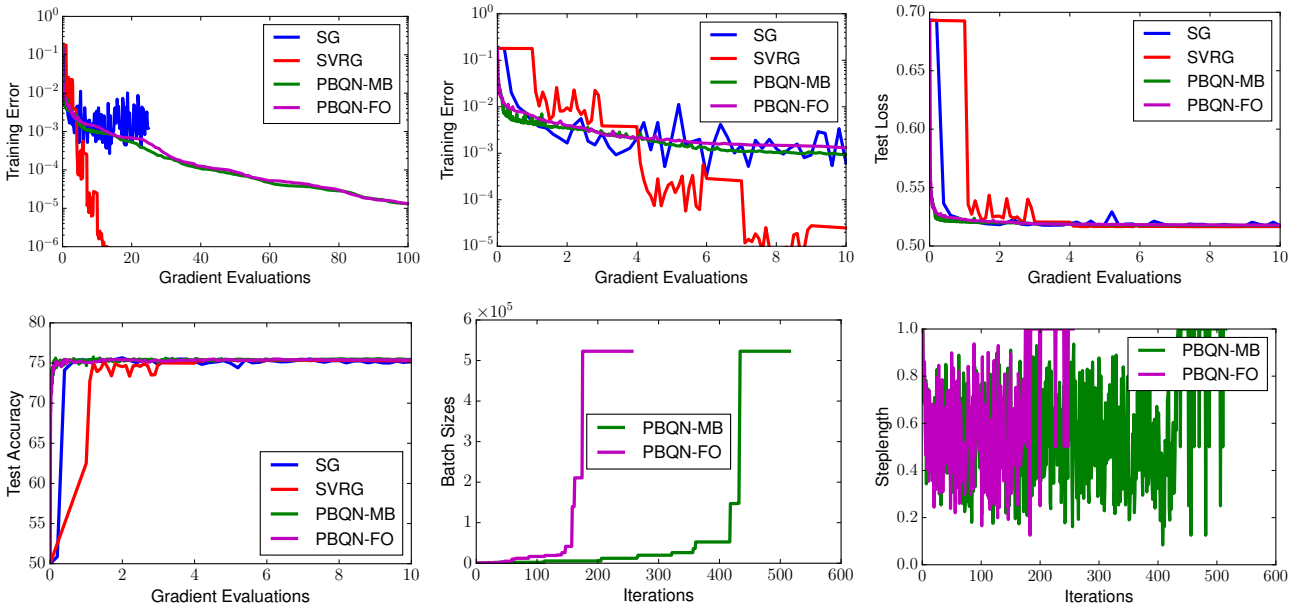

Figure 7. **covertype dataset:** Performance of the progressive batching L-BFGS methods, with multi-batch (MB) (25% overlap) and full-overlap (FO) approaches, and the SG and SVRG methods.

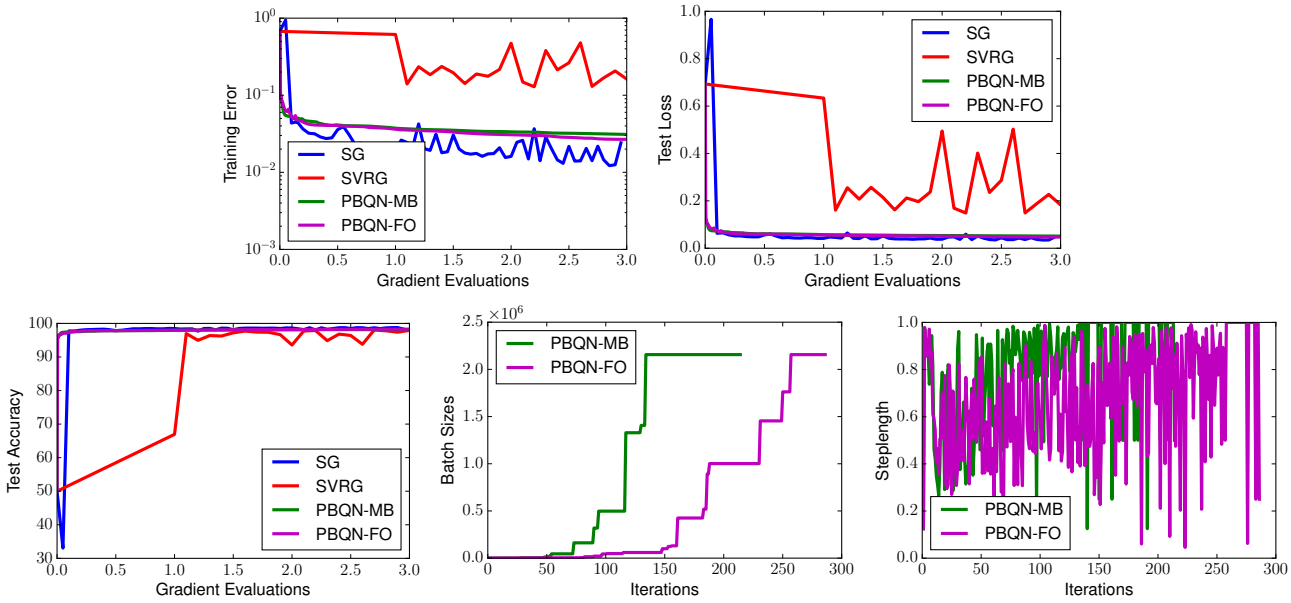

Figure 8. **url dataset:** Performance of the progressive batching L-BFGS methods, with multi-batch (MB) (25% overlap) and full-overlap (FO) approaches, and the SG and SVRG methods. Note that we only ran the SG and SVRG algorithms for 3 gradient evaluations since the equivalent number of iterations already reached of order of magnitude 10^7.

C.3. Neural Network Experiments

We describe each neural network architecture below. We plot the training loss, test loss and test accuracy against the total number of iterations and gradient evaluations. We also report the behavior of the batch sizes and steplengths for both variants of the PBQN method.

C.3.1. CIFAR-10 Convolutional Network ($\mathcal{C}$) Architecture

The small convolutional neural network (ConvNet) is a 2-layer convolutional network with two alternating stages of 5×5 kernels and 2×2 max pooling followed by a fully connected layer with 1000 ReLU units. The first convolutional layer yields 6 output channels and the second convolutional layer yields 16 output channels.

C.3.2. CIFAR-10 and MNIST AlexNet-like Network ($\mathcal{A}_1, \mathcal{A}_2$) Architecture

The larger convolutional network (AlexNet) is an adaptation of the AlexNet architecture (Krizhevsky et al., 2012) for CIFAR-10 and MNIST. The CIFAR-10 version consists of three convolutional layers with max pooling followed by two fully-connected layers. The first convolutional layer uses a 5×5 kernel with a stride of 2 and 64 output channels. The second and third convolutional layers use a 3×3 kernel with a stride of 1 and 64 output channels. Following each convolutional layer is a set of ReLU activations and 3×3 max poolings with strides of 2. This is all followed by two fully-connected layers with 384 and 192 neurons with ReLU activations, respectively. The MNIST version of this network modifies this by only using a 2×2 max pooling layer after the last convolutional layer.

C.3.3. CIFAR-10 Residual Network ($\mathcal{R}$) Architecture

The residual network (ResNet18) is a slight modification of the ImageNet ResNet18 architecture for CIFAR-10 (He et al., 2016). It follows the same architecture as ResNet18 for ImageNet but removes the global average pooling layer before the 1000 neuron fully-connected layer. ReLU activations and max poolings are included appropriately.

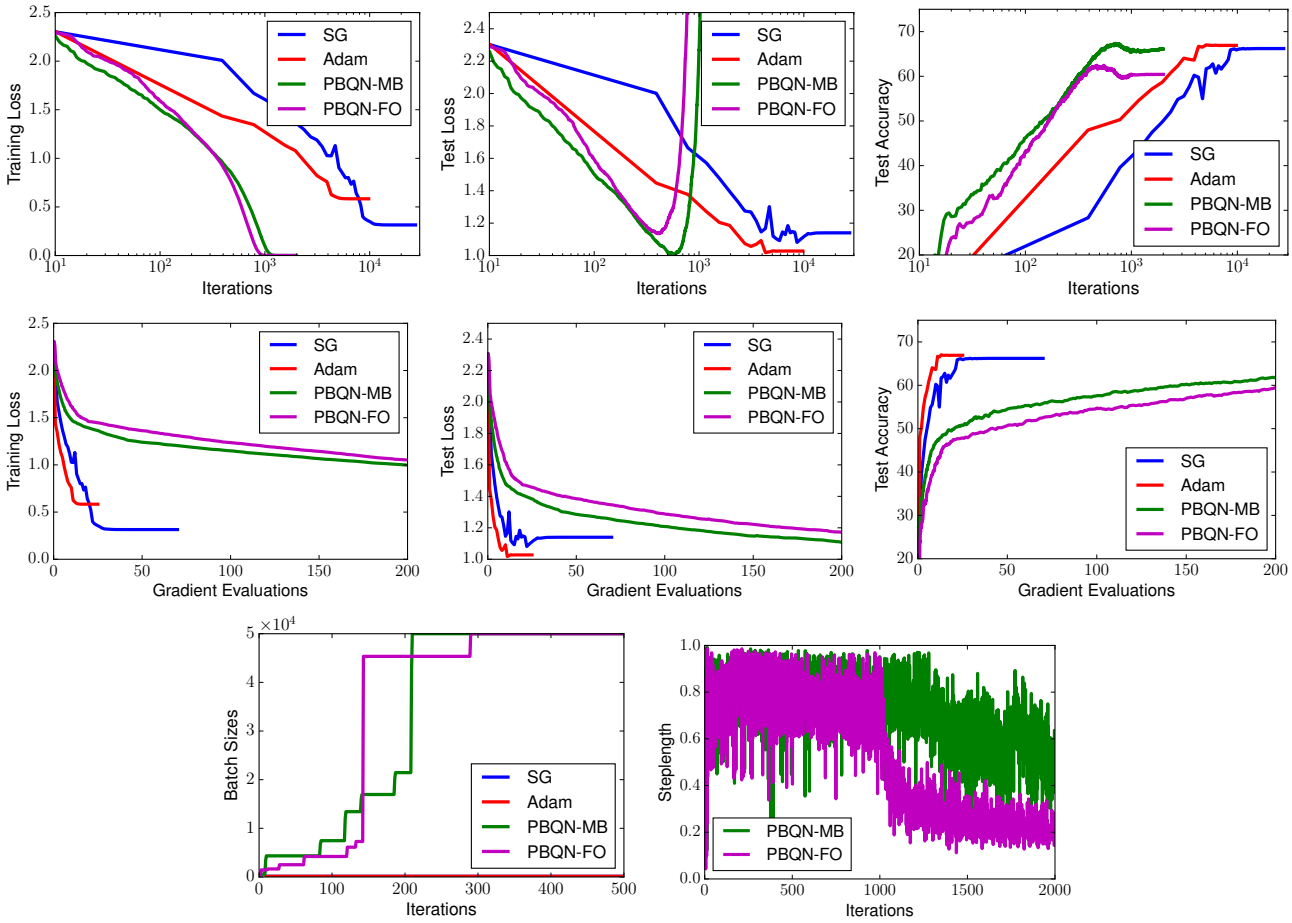

Figure 9. **CIFAR-10 ConvNet** ($\mathcal{C}$)**:** Performance of the progressive batching L-BFGS methods, with multi-batch (MB) (25% overlap) and full-overlap (FO) approaches, and the SG and Adam methods. The best results for L-BFGS are achieved with $\theta = 0.9$.

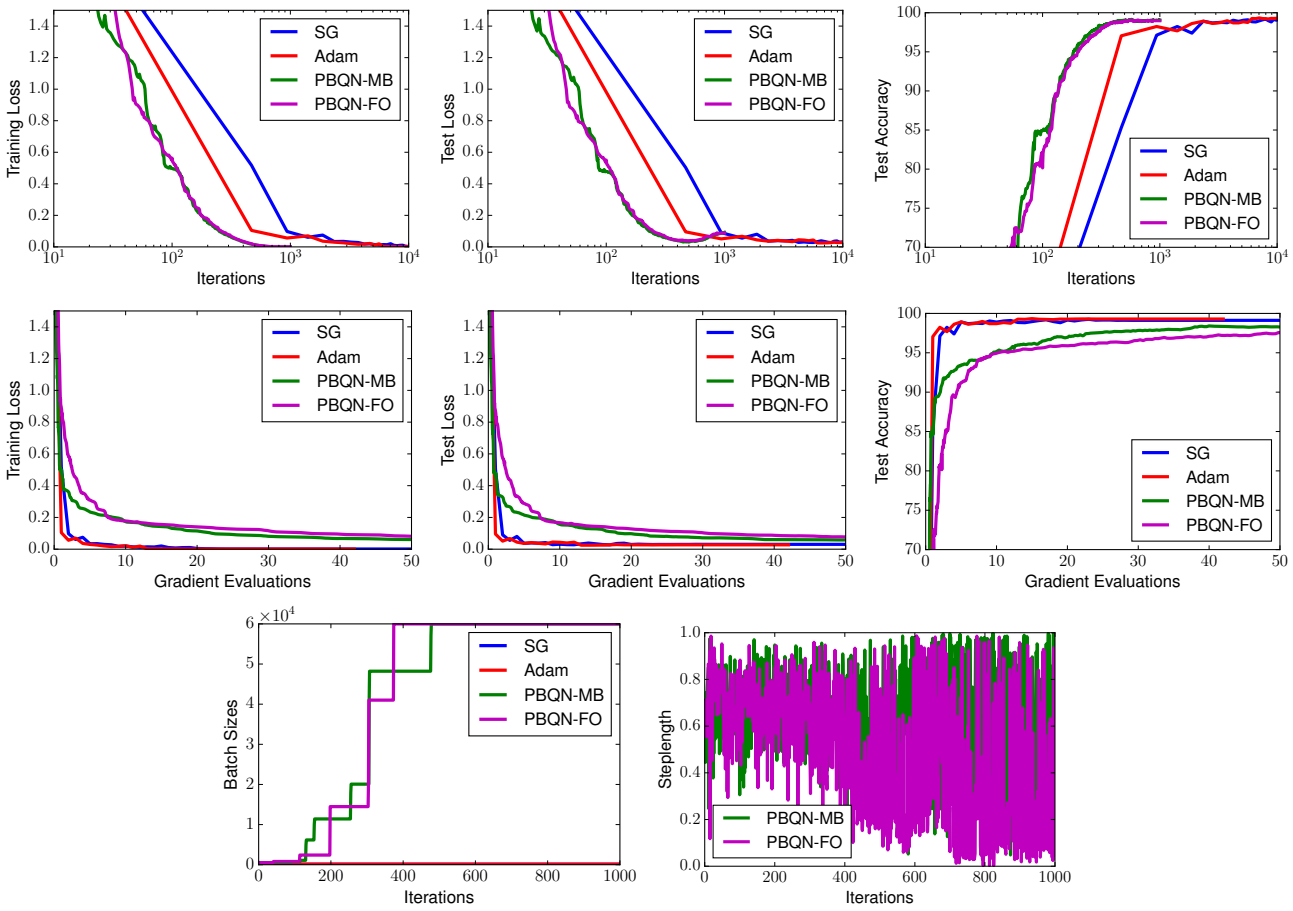

Figure 10. **MNIST AlexNet** ($\mathcal{A}_1$)**:** Performance of the progressive batching L-BFGS methods, with multi-batch (MB) (25% overlap) and full-overlap (FO) approaches, and the SG and Adam methods. The best results for L-BFGS are achieved with $\theta = 2$.

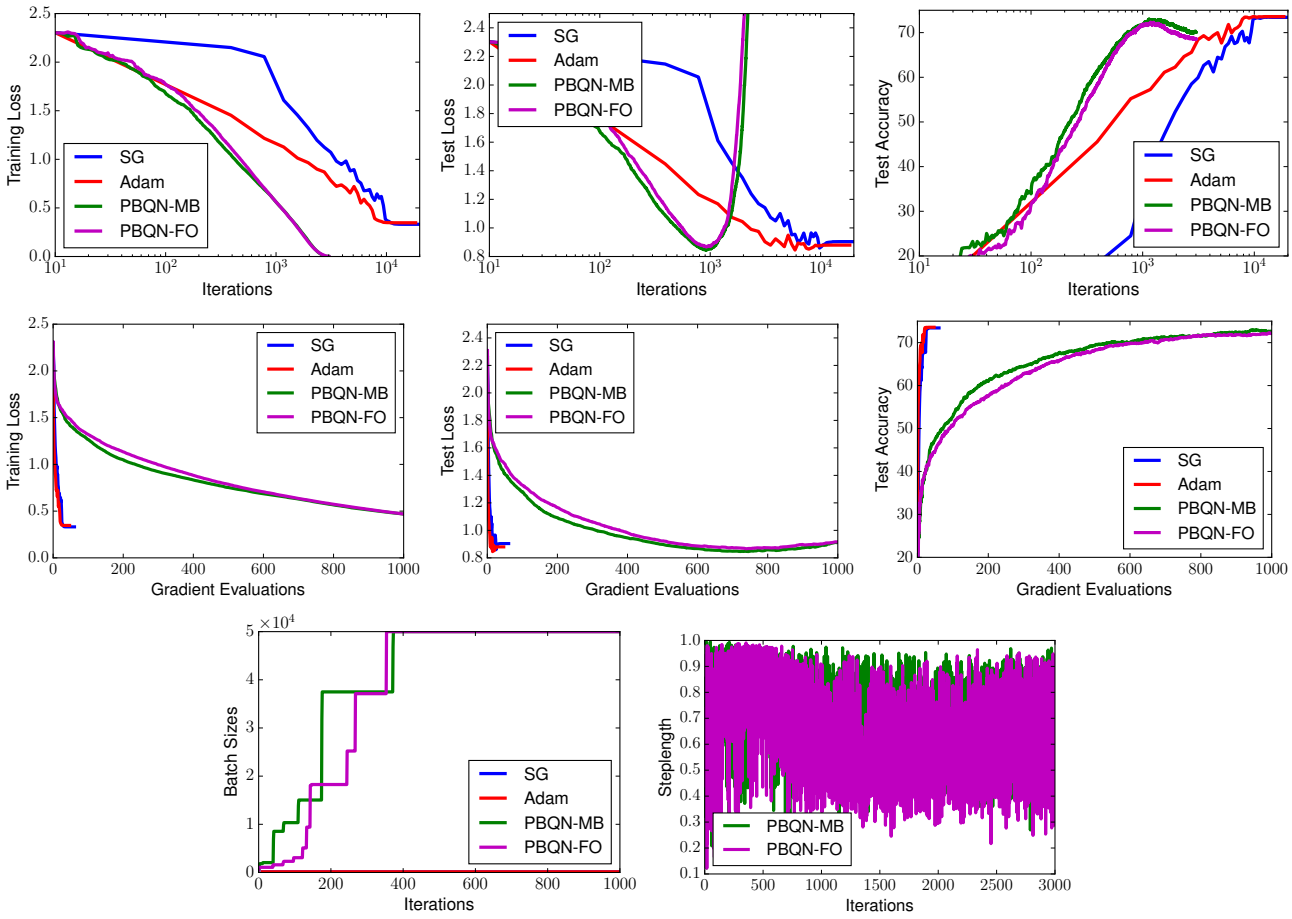

Figure 11. **CIFAR-10 AlexNet** ($\mathcal{A}_2$)**:** Performance of the progressive batching L-BFGS methods, with multi-batch (MB) (25% overlap) and full-overlap (FO) approaches, and the SG and Adam methods. The best results for L-BFGS are achieved with $\theta = 0.9$.

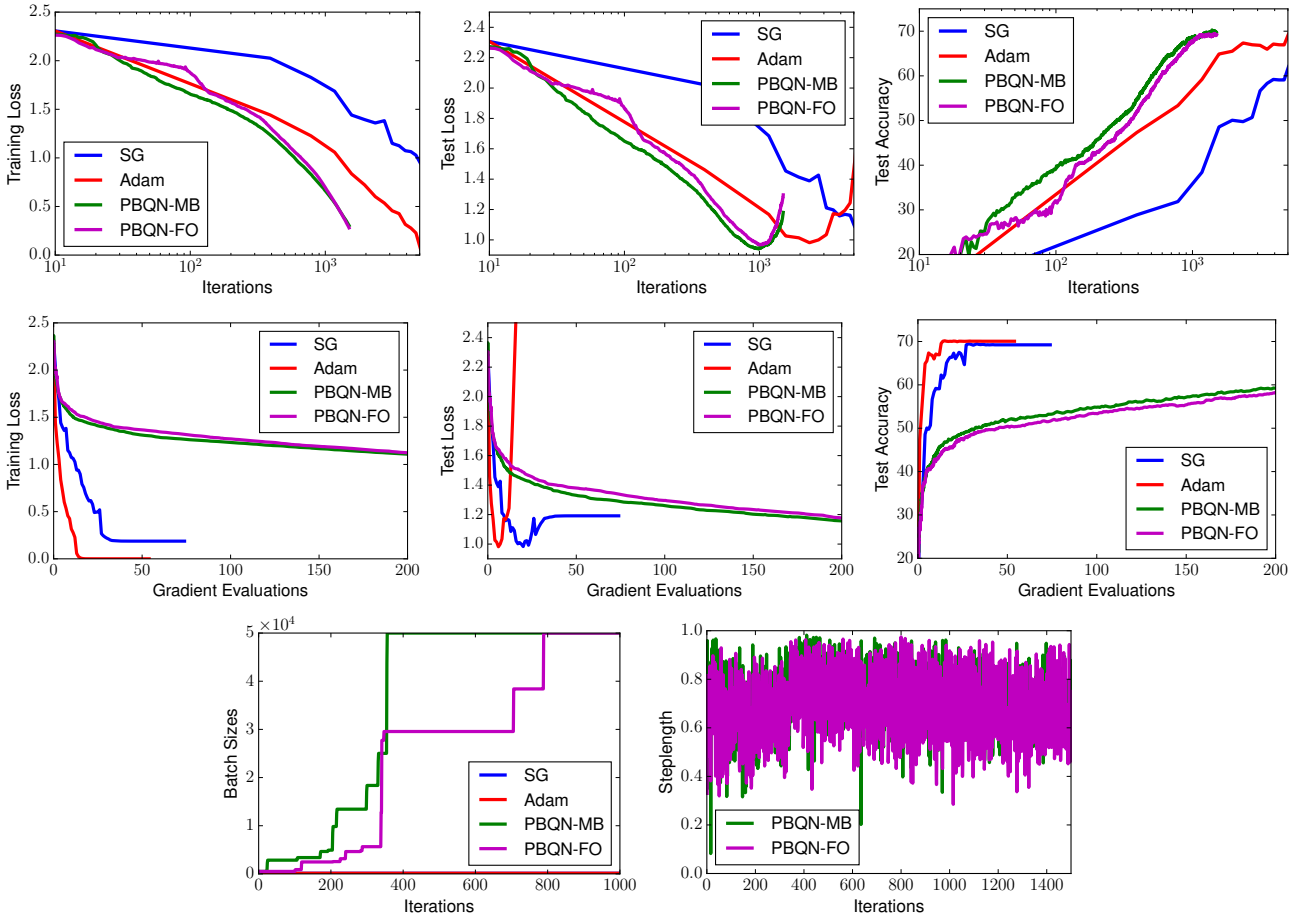

Figure 12. **CIFAR-10 ResNet18** ($\mathcal{R}$)**:** Performance of the progressive batching L-BFGS methods, with multi-batch (MB) (25% overlap) and full-overlap (FO) approaches, and the SG and Adam methods. The best results for L-BFGS are achieved with $\theta = 2$.

D. Performance Model

The use of increasing batch sizes in the PBQN algorithm yields a larger effective batch size than the SG method, allowing PBQN to scale to a larger number of nodes than currently permissible even with large-batch training (Goyal et al., 2017). With improved scalability and richer gradient information, we expect reduction in training time. To demonstrate the potential to reduce training time of a parallelized implementation of PBQN, we extend the idealized performance model from (Keskar et al., 2016) to the PBQN algorithm. For PBQN to be competitive, it must achieve the following: (i) the quality of its solution should match or improve SG's solution (as shown in Table 1 of the main paper); (ii) it should utilize a larger effective batch size; and (iii) it should converge to the solution in a lower number of iterations. We provide an initial analysis for this by establishing the analytic requirements for improved training time; we leave discussion on implementation details, memory requirements, and large-scale experiments for future work.

Let the effective batch size for PBQN and conventional SG batch size be denoted as $\widehat{B_L}$ and B_S, respectively. From Algorithm 1, we observe that the PBQN iteration involves extra computation in addition to the gradient computation as in SG. The additional steps are as follows: the L-BFGS two-loop recursion, which includes several operations over the stored curvature pairs and network parameters (Algorithm 1:6); the stochastic line search for identifying the steplength (Algorithm 1:7-16); and curvature pair updating (Algorithm 1:18-21). However, most of these supplemental operations are performed on the weights of the network, which is orders of magnitude lower than computing the gradient. The two-loop recursion performs $O(10)$ operations over the network parameters and curvature pairs. The cost for variance estimation is negligible since we may use a fixed number of samples throughout the run for its computation which can be parallelized while avoiding becoming a serial bottleneck.

The only exception is the stochastic line search, which requires additional forward propagations over the model for different sets of network parameters. However, this happens only when the step-length is not accepted, which happens infrequently in practice. We make the pessimistic assumption of an addition forward propagation every iteration, amounting to an additional $\frac{1}{3}$ the cost of the gradient computation (forward propagation, back propagation with respect to activations and weights). Hence, the ratio of cost-per-iteration for PBQN C_L to SG's cost-per-iteration C_S is $\frac{4}{3}$. Let I_S and I_L be the number of iterations that it takes SG and PBQN, respectively, to reach similar test accuracy. The target number of nodes to be used for training is N, such that $N < \widehat{B_L}$. For N nodes, the parallel efficiency of SG is assumed to be $P_e(N)$ and we assume that for the target node count, there is no drop in parallel efficiency for PBQN due to the large effective batch size.

For a lower training time with the PBQN method, the following relation should hold:

$$I_L C_L \frac{\widehat{B_L}}{N} < I_S C_S \frac{B_S}{N P_e(N)}. \tag{19}$$

In terms of iterations, we can rewrite this as

$$\frac{I_L}{I_S} < \frac{C_S}{C_L} \frac{B_S}{\widehat{B_L}} \frac{1}{P_e(N)}. \tag{20}$$

Assuming target node count $N = B_S < \hat{B}_L$, the scaling efficiency of SG drops significantly due to the reduced work per single node, giving a parallel efficiency of $P_e(N) = 0.2$; see (Kurth et al., 2017; You et al., 2017). If we additionally assume that effective batch size for PBQN is $4\times$ larger, with SG large batch $\approx$ 8K and PBQN $\approx$ 32K as observed in our experiments (from Section 4), this gives $\widehat{B_L}/B_S = 4$. PBQN must converge with about the same number of iterations as SG in order to achieve lower training time. From Section 4, the results show that PBQN converges in significantly fewer iterations than SG, hence establishing the potential for lower training times. We refer the reader to (Das et al., 2016) for a more detailed model and commentary on the effect of batch size on performance.

References

Berahas, A. S., Nocedal, J., and Takác, M. A multi-batch L-BFGS method for machine learning. In *Advances in Neural Information Processing Systems*, pp. 1055–1063, 2016.

Bertsekas, D. P., Nedić, A., and Ozdaglar, A. E. *Convex analysis and optimization*. Athena Scientific Belmont, 2003.

Carbonetto, P. *New probabilistic inference algorithms that harness the strengths of variational and Monte Carlo methods*. PhD thesis, University of British Columbia, 2009.

Chang, C. and Lin, C. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Cormack, G. and Lynam, T. Spam corpus creation for TREC. In *Proc. 2nd Conference on Email and Anti-Spam*, 2005. http://plg.uwaterloo.ca/gvcormac/treccorpus.

Das, D., Avancha, S., Mudigere, D., Vaidynathan, K., Sridharan, S., Kalamkar, D., Kaul, B., and Dubey, P. Distributed deep learning using synchronous stochastic gradient descent. *arXiv preprint arXiv:1602.06709*, 2016.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Guyon, I., Aliferis, C. F., Cooper, G. F., Elisseeff, A., Pellet, J., Spirtes, P., and Statnikov, A. R. Design and analysis of the causation and prediction challenge. In *WCCI Causation and Prediction Challenge*, pp. 1–33, 2008.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

Kurth, T., Zhang, J., Satish, N., Racah, E., Mitliagkas, I., Patwary, M. M. A., Malas, T., Sundaram, N., Bhimji, W., Smorkalov, M., et al. Deep learning at 15pf: Supervised and semi-supervised classification for scientific data. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 7. ACM, 2017.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

You, Y., Gitman, I., and Ginsburg, B. Scaling SGD batch size to 32k for ImageNet training. *arXiv preprint arXiv:1708.03888*, 2017.