A. Initial Step Length Derivation

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

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

where the batch (or subsampled) gradient is given by

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

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

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

We establish the inequality used to determine the initial steplength \( \alpha_k \) for the stochastic line search.

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

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

where

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

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

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

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

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

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

$$
\Lambda_1 I \preceq H_k \preceq \Lambda_2 I.
$$

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 $y_k^T s_k \geq \epsilon \| s_k \|^2$ 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

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

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 $k$,

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

Moreover, if $\alpha_k$ satisfies

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

we have that

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

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

$$
\mathbb{E}_k \left[ \left\| H_k g_k^S \right\|^2 \right] \leq \mathbb{E}_k \left[ \left\| H_k g_k^S - \frac{(H_k g_k^S)^T (H_k \nabla F(x_k))}{\| H_k \nabla F(x_k) \|^2} H_k \nabla F(x_k) \right\|^2 \right]
$$

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

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

$$
\mathbb{E}_k \left[ \left\| H_k g_k^S \right\|^2 \right] = \mathbb{E}_k \left[ \left\| H_k \nabla F(x_k) \right\|^2 \right] - 2 \mathbb{E}_k \left[ \frac{(H_k g_k^S)^T (H_k \nabla F(x_k))}{\| H_k \nabla F(x_k) \|^2} H_k \nabla F(x_k) \right] + \mathbb{E}_k \left[ \left\| (H_k g_k^S)^T (H_k \nabla F(x_k)) \right\|^2 \right].
$$

$$
\leq \nu^2 \left\| H_k \nabla F(x_k) \right\|^2.
$$
Therefore, rearranging gives the inequality
\[
\mathbb{E}_k \left[ ||H_k g_k^S||^2 \right] \leq \mathbb{E}_k \left[ \frac{((H_k g_k^S)^T (H_k \nabla F(x_k)))^2}{||H_k \nabla F(x_k)||^2} \right] + \nu^2 ||H_k \nabla F(x_k)||^2. \tag{11}
\]
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
\[
\mathbb{E}_k \left[ \frac{((H_k \nabla F(x_k))^T (H_k g_k^S)) - ||H_k \nabla F(x_k)||^2}{|S_k|} \right] \leq \theta^2 ||H_k \nabla F(x_k)||^2, \tag{12}
\]
where the second inequality holds by the IPQN test. Since
\[
\mathbb{E}_k \left[ \left( (H_k \nabla F(x_k))^T (H_k g_k^S) \right)^2 \right] = \mathbb{E}_k \left[ \left( (H_k \nabla F(x_k))^T (H_k g_k^S) \right)^2 \right] - ||H_k \nabla F(x_k)||^4, \tag{13}
\]
we have
\[
\mathbb{E}_k \left[ \left( (H_k g_k^S)^T (H_k \nabla F(x_k)) \right)^2 \right] \leq ||H_k \nabla F(x_k)||^4 + \theta^2 ||H_k \nabla F(x_k)||^4 \tag{14}
\]
by (12) and (13). Substituting (14) into (11), we get the following bound on the length of the search direction:
\[
\mathbb{E}_k [||H_k g_k^S||^2] \leq (1 + \theta^2 + \nu^2)||H_k \nabla F(x_k)||^2,
\]
which proves (7). Using this inequality, Assumption B.2, and bounds on the Hessian and steplength (6) and (8), we have
\[
\mathbb{E}_k [F(x_{k+1})] \leq F(x_k) - \mathbb{E}_k \left[ \alpha (H_k g_k^S)^T \nabla F(x_k) \right] + \frac{L \alpha^2}{2} ||H_k g_k^S||^2
\]
\[
= F(x_k) - \alpha \nabla F(x_k)^T H_k \nabla F(x_k) + \frac{L \alpha^2}{2} \mathbb{E}_k [||H_k g_k^S||^2]
\]
\[
\leq F(x_k) - \alpha \nabla F(x_k)^T H_k \nabla F(x_k) + \frac{L \alpha^2}{2} \left(1 + \theta^2 + \nu^2\right) ||H_k \nabla F(x_k)||^2
\]
\[
= F(x_k) - \alpha (H_k^{1/2} \nabla F(x_k))^T \left(I - \frac{L \alpha (1 + \theta^2 + \nu^2)}{2} H_k\right) H_k^{1/2} \nabla F(x_k)
\]
\[
\leq F(x_k) - \alpha \left(1 - \frac{L \alpha^2 (1 + \theta^2 + \nu^2)}{2}\right) ||H_k^{1/2} \nabla F(x_k)||^2
\]
\[
\leq F(x_k) - \frac{\alpha}{2} ||H_k^{1/2} \nabla F(x_k)||^2.
\]
\]

We now show that the stochastic quasi-Newton iteration (1) with a fixed steplength \(\alpha\) 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
\[
\mu I \preceq \nabla^2 F(x) \preceq L I, \quad \forall x \in \mathbb{R}^d. \tag{15}
\]
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 \(\alpha_k\) 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 \(\rho = 1 - \mu \Delta_1 \alpha\).
We now consider the case when $F$ is twice continuously differentiable and bounded below, and that there exists a constant $L > 0$ such that
\[ \nabla^2 F(x) \preceq LI, \quad \forall x \in \mathbb{R}^d. \] (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 $\alpha_k$ satisfies (8), we have
\[ \lim_{k \to \infty} \mathbb{E}[\|\nabla F(x_k)\|^2] \to 0. \] (18)

Moreover, for any positive integer $T$ we have that
\[ \min_{0 \leq k \leq T-1} \mathbb{E}[\|\nabla F(x_k)\|^2] \leq \frac{2}{\alpha T \Lambda_1} (F(x_0) - F_{\text{min}}), \]
where $F_{\text{min}}$ 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 $F_{\text{min}}$, we get
\[ \sum_{k=0}^{T-1} \mathbb{E}[\|H_k^{1/2} \nabla F(x_k)\|^2] \leq \frac{2}{\alpha} \mathbb{E}[F(x_0) - F(x_T)] \leq \frac{2}{\alpha} [F(x_0) - F_{\text{min}}]. \]

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
\[ \min_{0 \leq k \leq T-1} \mathbb{E}[\|\nabla F(x_k)\|^2] \leq \frac{1}{T} \sum_{k=0}^{T} \mathbb{E}[\|\nabla F(x_k)\|^2] \leq \frac{2}{\alpha T \Lambda_1} (F(x_0) - F_{\text{min}}). \]
Table 1. Characteristics of all datasets used in the experiments.

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

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.
Supplementary Material for “A Progressive Batching L-BFGS Method for Machine Learning”

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 (C) Architecture

The small convolutional neural network (ConvNet) is a 2-layer convolutional network with two alternating stages of $5 \times 5$ kernels and $2 \times 2$ 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 ($A_1, A_2$) 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 $5 \times 5$ kernel with a stride of 2 and 64 output channels. The second and third convolutional layers use a $3 \times 3$ kernel with a stride of 1 and 64 output channels. Following each convolutional layer is a set of ReLU activations and $3 \times 3$ 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 $2 \times 2$ max pooling layer after the last convolutional layer.

C.3.3. CIFAR-10 Residual Network ($R$) 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 (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 ($A_1$): 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 ($A_2$): 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 (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 \( \hat{B}_L \) 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 \( \frac{1}{4} \) 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 \( \frac{3}{4} \). 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 \( N < \hat{B}_L \). 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:

\[
I_L C_L \frac{\hat{B}_L}{N} < I_S C_S \frac{B_S}{NP_e(N)}.
\]

In terms of iterations, we can rewrite this as

\[
\frac{I_L}{I_S} < \frac{C_S}{C_L} \frac{B_S}{\hat{B}_L} \frac{1}{P_e(N)}.
\]

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 \( \hat{B}_L/B_S = 4 \). 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


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


