Goal
Reduce the communication cost in distributed Stochastic Gradient MCMC with the master-worker framework

Background
\[ \theta_{t+1} \leftarrow \theta_t + \eta_t \hat{f}_t + \sqrt{\frac{2}{N}} \xi_t \text{ where } \xi_t \sim \mathcal{N}(0, 1) \]
\[ \hat{f}_t = \nabla \tilde{U}(\theta_t), \text{ where } \widetilde{U}(\theta_t) = - \log p(\theta_t) - N \sum_{m=1}^{M} \log p(D_m | \theta_t) \]

SG-MCMC with a master-work framework
Traditional SG-MCMC (serial) can be seen as evaluating gradients on a single worker, while the master waits until the worker has completed its task

Communication Protocols

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
</table>
| Downpour GSLD | Scheme A in Table 1 is extended to accumulate the update during the \( \tau \) steps in \( \nu \) (Dean and others 2012), which denotes the space that the worker has explored since its last communication. Besides \( \theta(\nu) \), the \( p \)-th worker also maintains \( \nu(\nu) \). When the next communication happens, the master absorbs \( \nu(\nu) \) and sends new center parameters back to the \( p \)-th worker to replace (update) \( \theta(\nu) \).
| Elastic SGHMC | The weighted average in Scheme B results in a difference between historical center parameter and current parameters sent from a worker, as the former are smoothed over previous steps. The master waits until the \( p \)-th worker has sent the requested \( \theta(\nu) \), then computes the elastic difference \( \alpha(\theta(\nu) - \tilde{\theta}) \) (Zhang et al. 2015). Next, this difference is sent back to the worker who then updates \( \theta(\nu) \).

Experimental Results
A. More workers \( P \) can accelerate training
B. Performance remains similar with reasonable communication periods
C. Prevent Overfitting compared with distributed optimization algorithms.
D. Maintain Uncertainty in Prediction
E. Exploration in Asynchronous Advantage Actor-Critic (A3C)

Communication Efficient Stochastic Gradient MCMC for Neural Networks
Chunyun Li, Hangyou Chen, Yunchen Pu, Ricardo Henao, and Lawrence Carin
Microsoft Research, Redmond University at Buffalo Facebook Duke University

Figure 1(d) shows PW(1), After 2000 epochs, the number of trainable parameters for a specific task hard to find a good solution any available solutions can be discovered as the number of updates increasing the redundancy of the solution set. Training on random labels forces the network to set up a base infrastructure.