TRAINING SAMPLE SELECTION FOR DEEP LEARNING OF DISTRIBUTED DATA

Zheng Jiang, Xiaoqing Zhu, Wai-tian Tan, and Rob Liston

Chief Technology and Architecture Office
Cisco Systems, San Jose, CA, USA

ABSTRACT

The success of deep learning — in the form of multi-layer neural networks — depends critically on the volume and variety of training data. Its potential is greatly compromised when training data originate in a geographically distributed manner and are subject to bandwidth constraints. This paper presents a data sampling approach to deep learning, by carefully discriminating locally available training samples based on their relative importance. Towards this end, we propose two metrics for prioritizing candidate training samples as functions of their test trial outcome: *correctness and confidence*. Bandwidth-constrained simulations show significant performance gain of our proposed training sample selection schemes over convention uniform sampling: up to 15× bandwidth reduction for the MNIST dataset and 25% reduction in learning time for the CIFAR-10 dataset.

Index Terms— Deep neural networks, training sample selection, bandwidth-constrained learning

1. INTRODUCTION

Since its breakthrough advances in speech and visual recognition in the last decade, deep learning has seen many successful applications, ranging from self-driving cars to business intelligence. Its success, nevertheless, depends critically on the volume and variety of training data. In many practical settings such as multi-site video surveillance, data naturally originate from geographically distributed sites. They need to be transmitted over a bandwidth-constrained connection to a central location for training. This may severely cripple the applicability of deep learning.

One possibility is to move learning to where the data are, and to train the deep neural network locally at each site. Such an approach inevitably imposes heavy computational burden on local sites for training. It is also unclear how learning can be infrequently aggregated across multiple sites. A practical alternative is to leave the training of a common neural network at a central site, but use only a subset of the most valuable training samples from each local site. This paper takes this second approach of distributed data sampling. As shown in Fig. 1, the central site hosts a common deep neural network model and continuously updates it using training samples from remote sites. Each remote site periodically obtains a refreshed copy of the neural network model to "trial test" local training samples. The outcome of these trail tests are used to determine relative importance of each training sample. Only a subset of the most informative ones are transmitted under a given bandwidth constraint.

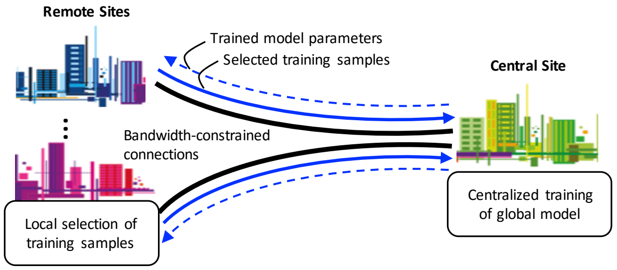

Remote Sites
Trained model parameters
Selected training samples
Central Site
Bandwidth-constrained connections
Centralized training of global model
Local selection of training samples

Fig. 1. Proposed framework for continuous training of deep neural networks with geographically distributed data.

Focusing on classification tasks based on deep neural networks, we consider two intuitive metrics of a candidate training sample: *correctness* and *confidence*. The former is a binary indicator of whether the label predicted by the current neural-network-based classifier already agrees with ground-truth label. Incorrect samples constitutes surprises, hence are considered as more valuable than correct ones. The latter denotes how confident the current classifier is about its prediction, and naturally maps to the softmax layer output of the neural network. Using two typical image classification tasks as examples, we examine in this paper various strategies for training data sampling, using correctness and confidence as selection criteria. The proposed training sample selection schemes are evaluated in simulations with bandwidth constraints. Their performance gains over conventional uniform sampling are discussed in terms of bandwidth savings and learning time reductions.

The rest of the paper is organized as follows. The next section reviews related work in deep learning. Section 3 presents an optimization formulation of the training sample selection problem and presents a heuristic solution based on proxy functions. Section 4 addresses the construction of the proxy function, and evaluation under bandwidth constraints are discussed in Sec. 5.

978-1-5090-2175-8/17/$31.00 ©2017 IEEE

2. RELATED WORK

In literature, distributed machine learning typically refers to distributed computation of centrally stored data [1] rather than learning of distributed data. Traditional algorithms for parallel stochastic gradient descent are vastly communication bound [2]. Nevertheless, it has been shown in [3] that careful selection and arrangement of data can improve learning speed and communication cost for parallel stochastic gradient descent. In contrast, the focus of our paper is distributed selection of training data for centralized learning.

In [4, 5], the authors study the problem of efficient estimation of mean values for distributed data, but did not consider the impact of bandwidth constraint for machine learning. In [6], the authors study machine learning of distributed data, but the approach is only valid for convex optimization problems and does not apply to general deep neural networks with rectified linear and max pooling layers. The notion of "federated learning" is mentioned in [7]. However, the work investigates various model merging techniques instead of a data sampling approach. Hard example mining [8][9] addresses the complementary problem of rare "negative" examples, which are reused multiple times to bootstrap the training of object detectors [10][11][12].

3. DISTRIBUTED DATA SAMPLING

In this section, we first formulate distributed data sampling as a constrained optimization problem. We then discuss several approximations that lead to practical algorithms.

3.1. Problem Formulation

In supervised learning tasks such as image classification, a training sample $s_i = (x_i, y_i)$ is represented by observed data x_i (e.g., an image) and its corresponding ground-truth label y_i (e.g., a dog). The process of training can be captured by the stochastic function of the classification accuracy on a given test dataset, at each step of the training: $A_n = f(s_{0:n})$, where $s_{0:n} = s_0, s_1, \ldots, s_n$ denotes the *sequence* of training samples submitted to the model.

In practice, training data is often grouped into batches and are randomly shuffled during training. The dependence on the order of training samples is thus negligible assuming the model has gone through sufficient number of epochs of training – even when the trainings data are available sequentially, as our setup in Fig. 1. In other words, the input $s_{0:n}$ to the accuracy function $f(\cdot)$ can be alternatively represented by a *set* $\mathbf{s}_n = \{s_0, s_1, \cdots, s_n\}$ rather than a sequence.

We consider the scenario where training data originate from multiple remote sites and are transmitted to a common neural network model hosted at a central site. Periodically, the remote site can send a subset of its observed training samples to the central site to assist its continuous training. The total number of samples are limited to $\rho|\mathbf{s}|$, where $|\mathbf{s}|$ is the number of all local training samples and $0 < \rho < 1$ is the fraction imposed by the bandwidth constraint.

Given the existing set of trained data $\mathbf{s}_0$ at the central site, the marginal accuracy improvement for the global model from a *subset* of the candidate training data $\tilde{\mathbf{s}} \subseteq \mathbf{s}$ is given by:

$$\Delta_{\tilde{\mathbf{s}}} = f(\tilde{\mathbf{s}} \cup \mathbf{s}_0) - f(\mathbf{s}_0).$$

At each site, the optimization procedure is then to compute the optimal binary selection vector $\mathbf{z}$, given by:

$$\begin{aligned} \max_{\mathbf{z}} \quad & \Delta_{\tilde{\mathbf{s}}} && (1) \\ \text{s.t.} \quad & \tilde{\mathbf{s}} = \{s_i | s_i \in \mathbf{s}, z_i = 1\} && (2) \\ & z_i \in \{0, 1\} \ \ \forall i \in \{1, \cdots, n\} && (3) \\ & \sum_{i=1}^{n} z_i < \rho|\mathbf{s}| = \rho n. && (4) \end{aligned}$$

3.2. A greedy heuristic solution

It is impractical to solve (1) - (4) directly, since it requires not only an accurate representation of f, but the evaluation of $\Delta_{\tilde{\mathbf{s}}}$ for combinatorially large number of candidate subsets from $\mathbf{s}$. Furthermore, the remote site lacks the knowledge of $\mathbf{s}_0$. One practical heuristic to reduce search space significantly for candidate set $\mathbf{s}$ is to consider only first order effects and to treat as additive the contribution of individual training samples s_i to the improvement of classification performance:

$$\Delta_{\tilde{\mathbf{s}}} \approx \sum_{s_i \in \tilde{\mathbf{s}}} \Delta_{s_i}, \qquad (5)$$

so that the optimization reduces to computation of individual contribution Δ_{s_i} followed by greedy packing.

Nevertheless, computation of (5) still requires evaluation of f, which is unavailable at the remote site. Instead, we note that with greedy packing, the solution of (1) - (4) depends only on sorting order of Δ_{s_i} for different s_i's, but not on their actual values. Therefore, it suffices to compute a conveniently computable proxy function that can preserve the relative order of incremental accuracy improvement.

3.3. Design of proxy function

Back-propagation [13], the primary process for training neural network models, takes as input the difference between predicted and actual labels. It is thus clear that incorrectly classified samples are more valuable than correctly classified ones. The proxy function should therefore depend on *correctness* of test-trial outcome of each training sample.

As we will show in Sec. 4, it is nevertheless not true that all correctly labelled samples are equally useful. Neural network models are constantly evolving during training, and correct samples that are "borderline" are more likely to contribute to learning via back-propagation. Therefore, it is also important to include *confidence* as input to the proxy function. In this paper, confidence $C(\mathbf{p})$ is computed from the probability distribution $\mathbf{p}$ of softmax layer output for the different possible labels. A few variants are considered as below and further evaluated in Sec. 4:

- **Cross-Entropy:** $C(\mathbf{p}) = -\log(p_l)$ where l is label of the training sample. This corresponds to the loss function used during training of neural network.
- **Entropy:** $C(\mathbf{p}) = -\sum_i p_i \log(p_i)$ is a classical measure for randomness.
- **Gini-Simpson index:** $C(\mathbf{p}) = 1 - \sum_i p_i^2$ measures the degree of concentration across different classes [14].
- **Max-Likelihood:** $C(\mathbf{p}) = 1 - \max_i p_i$ is derived from probability of the most likely class.

4. EXPERIMENTS ON PROXY FUNCTION DESIGN

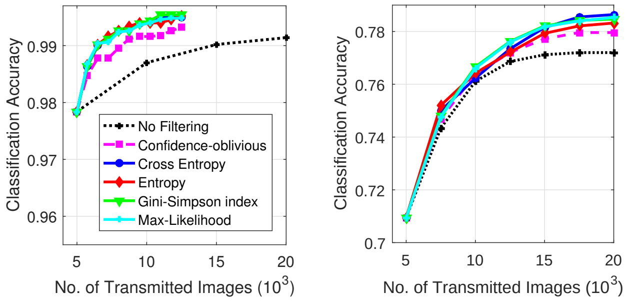
In this section, we establish experimentally the proxy function from correctness and confidence measures using two well known datasets: MNIST [15] for handwritten digits and CIFAR-10 for tiny images [16]. We employ deep neural network using well known models described in [15] for MNIST and in [16] for CIFAR-10.

The MNIST dataset contains 5k images for validation and 55k images for training. We use 5k from the training set to initialize the common model, and organize the remaining into 10 folds of 5k images each. For CIFAR-10, we use 5k images for validation, 10k for the initial model, and organize the remainder into 8 folds of 5k images each. The folds are then presented sequentially in multiple rounds. The training samples undergo various data selection algorithms as outlined below; the model is retrained at the end of each round.

4.1. Evaluation of confidence metrics

We first compare the different confidence metrics as listed in Sec. 3.3 in terms of their effect on learning speed. To remove the effect of correctness, we perform an experiment in which all mislabelled data are kept, and we further keep a varying amount of correctly labelled data with the *lowest* confidence according to various metric to maintain a fixed selection ratio ρ. The results are shown in Figure 2, where different color lines correspond to different choice of confidence metrics. For comparison, we also show in black the scheme of uniformly choosing samples to fill budget ρ, and in red a *confidence-oblivious* scheme which forwards all incorrect samples first and fills rest of the transmission budget by uniformly sampling from all remaining correct ones.

It is quite obvious from Fig. 2 that for both datasets, our proposed strategy of sorting training samples based on both correctness and confidence outperforms the reference scheme which sorts by correctness along, which in turn significantly outperforms the scheme without filtering in terms of trade-off between accuracy and bandwidth. On the other hand, results from using different confidence metrics are comparable with each other. For the rest of this paper, we employ the simplest, *cross-entropy* as confidence metric.

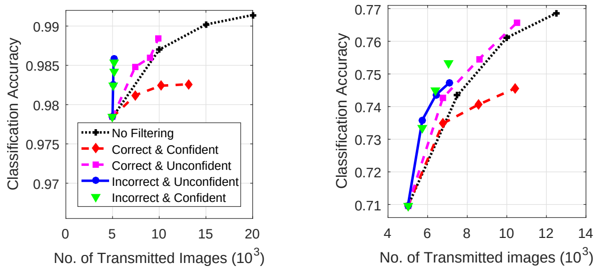


(a) MNIST, $\rho = 0.15$ (b) CIFAR-10, $\rho = 0.5$

Fig. 2. Impact of employing different confidence metrics in selective forwarding of training samples under a bandwidth constraint.

4.2. Relative importance of correctness and confidence

We next compare relative importance of *correctness* and *confidence*. We first apply a confidence threshold to the correctly labelled data so that we have same number of sample in the "correct & confident" group as "correct & unconfident". We similarly divide the incorrectly labelled data to arrive at four groups. We then perform an experiment in which each fold of data is classified into the four groups, and only data from one group is used for training. Figure 3 shows the learning speed of the different groups. It is quite obvious that the two schemes which prioritize transmission of incorrect samples significantly outperform their counterparts. It is also interesting to note the *correct & unconfident* group has meaningfully higher learning speed than the *correct & confident* group.



(a) MNIST Dataset (b) CIFAR-10 Dataset

Fig. 3. Comparison of four categories of training samples in terms of their incremental contribution to the classification accuracy of the global neural network model.

Based on the observations from both sets of experiments, we introduce two proxy functions with different complexity. The *confidence-oblivious* proxy sorts all samples by correctness only, thus corresponds to a binary indicator function $g_i(o_i, y_i) = \mathbf{1}_{y_i \neq o_i}$, where o_i is the test-trial output for sample i with ground-truth label y_i. It favors the transmission of all incorrect samples, and does not differentiate between correct samples. The more elaborate *correctness & confidence* proxy accounts for both metrics, and can be mathematically expressed as:

$$g_i(o_i, y_i, \mathbf{p}_i) = \begin{cases} H(\mathbf{p}_i), & \text{if } o_i \neq y_i, \\ -\frac{1}{H(\mathbf{p}_i)+\epsilon}, & \text{otherwise.} \end{cases} \tag{6}$$

where ϵ is a small constant to ensure numerical stability. Such a proxy also favors the transmission of incorrect samples first. Samples are further sorted according to confidence within each subcategory.

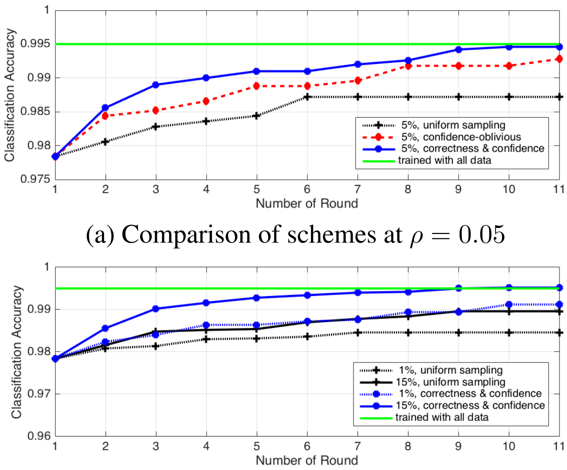
5. EVALUATION RESULTS

We now evaluate proposed bandwidth-constrained training sample selection schemes using the same setup and dataset as in Sec. 4. We focus on training sample selection at a single site, and reserve for future work the investigation of coordination across multiple remote sites.

Figure 4 (a) shows evolution of the classification accuracy after retraining at each round. Given bandwidth constraint of $\rho = 0.05$, it takes only 2 rounds for the proposed sample selection scheme employing *correctness & confidence* (in blue) to achieve a classification accuracy of 98.9%. In comparison, the *confidence-oblivious* selection scheme (in red) requires 4 rounds and the *uniform sampling* scheme (in black) needs all 10 rounds to reach the same accuracy level. This translates into speed up in learning time by $2\times$ over confidence-oblivious selection and by $5\times$ over uniform sampling. Relative performance of the three schemes remains the same over a range of bandwidth constraints, from $\rho = 0.01$ to $\rho = 0.30$.

Figure 4 (b) further examines the impact of different bandwidth constraints. It can be observed that sampling 1% of data based on correctness & confidence yields comparable performance as uniformly sampling 15% of training data at each round. This is equivalent to a $15\times$ reduction in bandwidth requirement. For MNIST, the performance gain is most significant at $\rho = 0.01$, since a tight bandwidth constraint underscores the importance of identifying and transmitting only the most valuable training samples.

Finally, Fig. 5 shows the classification accuracy at each round with $\rho = 0.30$, for the CIFAR-10 dataset. In this case, the initial accuracy is fairly low, at around 74%. The subset of selected training samples is therefore dominated by incorrect ones, resulting in comparable performances between confidence-oblivious and correctness & confidence schemes. Both outperform uniform sampling and reach a final classification accuracy of 77.4% instead of 76.5%. Note that both schemes exceed the final accuracy of the uniform sampling scheme by the 6-th round, effectively speed up the training time by 25%. Intuitively, the CIFAR-10 dataset contains more diverse image samples than the MNIST dataset, therefore it sees less bandwidth reduction gain via selective transmission of training samples. We believe that this observation holds more generally: the level of redundancy in the original training data will influence the amount of potential savings a training sample selection scheme can achieve.



(a) Comparison of schemes at $\rho = 0.05$

(b) Impact of different bandwidth constraints.

Fig. 4. Bandwidth-constrained transmission of training samples from the MNIST dataset, using different forwarding strategies.

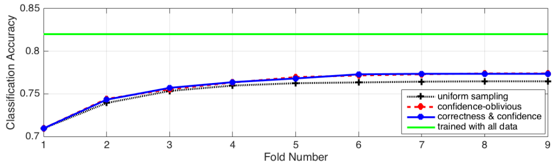


Fig. 5. Bandwidth-constrained transmission of training samples from the CIFAR-10 dataset, using different forwarding strategies. The bandwidth constraint corresponds to 30% of total training data.

6. CONCLUSIONS AND FUTURE WORK

This paper presents a data sampling approach to training deep neural networks with geographically distributed data under bandwidth constraints. Via analysis and experiments using two popular datasets (MNIST and CIFAR-10), we show that training samples vary vastly in terms of their relative importance and that correctness and confidence from local test trials can serve as effective proxies for prioritizing their transmissions given bandwidth constraint. Experimental evaluations of our proposed intelligent sampling schemes show that performance gains largely depend on the nature and redundancy of the original full training dataset, ranging from $15\times$ bandwidth savings on MNIST to around 25% reduction of training time for CIFAR-10.

The idea of selecting most relevant samples should also apply to training acceleration. A larger scale study is needed to determine whether omitting unimportant samples in our method leads to long-term performance degradation. It will also be interesting to characterize the impact of dataset redundancy on performance of various sample selection schemes.

7. REFERENCES

[1] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al., "Large scale distributed deep networks," in *Advances in neural information processing systems*, 2012, pp. 1223–1231.

[2] Janis Keuper and Franz-Josef Pfreundt, "Distributed training of deep neural networks: theoretical and practical limits of parallel scalability," in *Proc. Workshop on Machine Learning in High Performance Computing Environments (MLHPC'16)*, Salt Lake City, UT, USA, Sept. 2016, pp. 19–26.

[3] Guillaume Alain, Alex Lamb, Chinnadhurai Sankar, Aaron Courville, and Yoshua Bengio, "Variance reduction in SGD by distributed importance sampling," *arXiv preprint arXiv:1511.06481*, Apr. 2016.

[4] Ananda Theertha Suresh, Felix X Yu, H Brendan McMahan, and Sanjiv Kumar, "Distributed mean estimation with limited communication," *arXiv preprint arXiv:1611.00429*, 2016.

[5] Jakub Konečnỳ and Peter Richtárik, "Randomized distributed mean estimation: Accuracy vs communication," *arXiv preprint arXiv:1611.07555*, 2016.

[6] Paolo Di Lorenzo and Simone Scardapane, "Parallel and distributed training of neural networks via successive convex approximation," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP'16)*, Salerno, Italy, Sept. 2016, pp. 1–6.

[7] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, Oct. 2016.

[8] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 761–769.

[9] Ilya Loshchilov and Frank Hutter, "Online batch selection for faster training of neural networks," *arXiv preprint arXiv:1511.06343*, 2015.

[10] Christopher Z Mooney, Robert D Duval, and Robert Duvall, *Bootstrapping: A nonparametric approach to statistical inference*, Number 94-95. Sage, 1993.

[11] Bradley Efron and Robert J Tibshirani, *An introduction to the bootstrap*, CRC press, 1994.

[12] Christian Léger, Dimitris N Politis, and oseph P Romano, "Bootstrap technology and applications," *Technometrics*, vol. 34, no. 4, pp. 378–398, 1992.

[13] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, pp. 1, 1988.

[14] Lou Jost, "Entropy and diversity," *Oikos*, vol. 113, no. 2, pp. 363–375, Feb. 2006.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998.

[16] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," 2009.