Data Compression using Sparse Stochastic Gradient Descent

SCOTT HOYLAND

STAT 672

MAY 5, 2021

Obstacles to Distributed Learning

Large datasets require algorithms to be optimized for a different trade off than small datasets.

- For small datasets, the tradeoff is approximation error vs estimation error (bias/variance tradeoff)
- For large datasets, the tradeoff is computational complexity of the optimization algorithms versus accuracy (Buttou and Bousquet)

One solution to the large dataset tradeoff is to use distributed algorithms

- Different computer cores run different parts of the algorithm and then share the results with the other cores, referred to as "communication"
- Distributed algorithms have additional communication costs at startup and for each iteration (David, A.)

To be a more efficient distributed system, the communication costs can be reduced

- Communication costs can be reduced by splitting messages into smaller chunks or by reducing the message size
- Sparse Stochastic Gradient Descent allows for both message types to be used.

Stochastic Gradient Descent

Gradient Descent is a method to approximate the solution of a function

- Similar to Newton's Method, Gradient Descent is an iterative algorithm that updates the coefficient vector until convergence

The general form of Gradient Descent is $w_{(t+1)} = w_t - \eta * g_t(w_t)$

- Where w_t is the coefficient vector at time t, η is the learning rate, and $g(w_t)$ is the vector from the gradient, or updating, function evaluated on w_t.

Stochastic Gradient Descent uses a subset of the dataset each iteration to update the coefficient vector.

- This subset can be of size $1 \leq k < n$

Sparse Stochastic Gradient Descent

To lessen the communication cost of distributed Stochastic Gradient Descent, the Sparse Stochastic Gradient Descent reduces the size of the updating vector, $g(w_t)$, by constraining a random subset to zero with probability p (Wangni J., Wang J., Liu J., Zhang T.).

The optimal probability p is approximated by the algorithm:

$$\text{Initialize } p_i^0 = min(\rho d / g_i| / \sum_{i=1}^{d} /g_i|, 1)$$

Repeat: $\text{Identify } I = \{1 < i < d | p_i^j \neq 1\}$

$$\text{Compute } c = \frac{\rho d - d + |I|}{\sum_{i \in I} p_i^j}$$

$$\text{If } c \leq 1, \text{return } p_i^j, \text{otherwise}, p_i^{j+1} = min(cp_i^j, 1)$$

Until Convergence ($c \leq 1$**)**

The remaining, non-zero elements of $g(w_t)$ are scaled by p_i to create a sparse scaled gradient vector (Q(g)) for each worker. Each updated gradient vector in the distributed system is averaged, and each iteration updates the coefficient vector by the equation $w_{(t+1)} = w_t - \eta * Q_t(g_t)$

Theoretical Bounds of Stochastic Gradient Descent

In *Gradient Sparsification for Communication-Efficient Distributed Optimization,* it is shown that:

The expected increase in variance is upper bounded by $(1+\rho)$

- ρ is the expected sparsity of the gradient

While the maximum number of iterations increases by up to a factor of $(1+\rho)$, the number of floating point numbers is reduced by a factor of up to $(1+\rho)^2 s/d$

- s is the subset of the feature space that is included in the sparse gradient vector

With optimal coding, the total communication cost can be reduced by a factor of at least

$$\frac{(1+\rho)\left((s+1)b+\log_2 d\right)}{db}$$

- where b is the bit cost of a floating-point scalar (Wangni J., Wang J., Liu J., Zhang T.).

Missed Loan Payment Example

One of the main considerations in a bank's decision to offer a loan is the probability that the applicant will be able to pay back the loan.

I used a dataset from Kaggle.com containing applicant information from over 300,000 approved loans and information on whether the applicant missed a payment

The factor variables in the dataset were split into leave-one-out indicator variables and the continuous variables in the dataset were normalized

I compared the performance of logistic regression to three algorithms that approximate logistic regression:

- Gradient Descent
- Stochastic Gradient Descent with a subset size of 0.33
- Sparse Stochastic Gradient Descent with a subset size of 0.33 and sparsification factor of 0.5.

Logistic Regression using Gradient Descent

The logistic regression is based on the Sigmoid function

$$\hat{y}_t = \frac{1}{1 - e^{-Z_t}}$$

- where Z is the linear combination of the coefficient vector, w, an intercept, b, and the data, X

$$Z_t = w_t X + b_t$$

The gradient function, $g(w_t)$, is a combination of the coefficient updating and the intercept updating

$$dw_t = \frac{1}{N} X(\hat{y}_t - y)$$

$$db_t = \frac{1}{N} \sum (\hat{y}_t - y)$$

The updating function then becomes

$$w_{(t+1)} = w_t - \eta * dw_t$$

$$b_{(t+1)} = b_t - \eta * db_t$$

Missed Loan Payment Example Results

Regression Type (Gradient Descent ran in parallel using 4 cores)	Misclassification Rate	Number of Iterations to Convergence	Runtime	Percent Improvement Compared to Logistic Regression
Logistic Regression	7.96%	-	62s	-
Gradient Descent	8.11%	17	17.3s	72%
Stochastic Gradient Descent	8.11%	45	45.5s	26%
Sparse Stochastic Gradient Descent	8.11%	2	2.01s	97%

Gradient Descent Type	Runtime of 45 iterations	Percent Improvement
Gradient Descent	43.39	4.5%
Stochastic Gradient Descent	45.45	0%
Sparse Stochastic Gradient Descent	39.53	13.0%

References

Buttou, L. and Bousquet, O. *The Tradeoffs of Large Scale Learning* https://proceedings.neurips.cc/paper/2007/file/0d3180d672e08b4c5312dcdafdf6ef36-Paper.pdf

David, A. *Communication Cost in Parallel Machines*. http://people.cs.aau.dk/~adavid/teaching/MTP-06/03a-MVP06-notes.pdf

Dutta, Gaurav. *Loan Defaulter EDA in a real business scenario.* https://www.kaggle.com/gauravduttakiit/loan-defaulter

Wangni J., Wang J., Liu J., Zhang T. *Gradient Sparsification for Communication-Efficient Distributed Optimization.* https://arxiv.org/pdf/1710.09854.pdf