Кодовые распределенные вычисления для задач машинного обучения

Мария Копылова

Проект №13

Сочи, Сириус
2020

Введение:

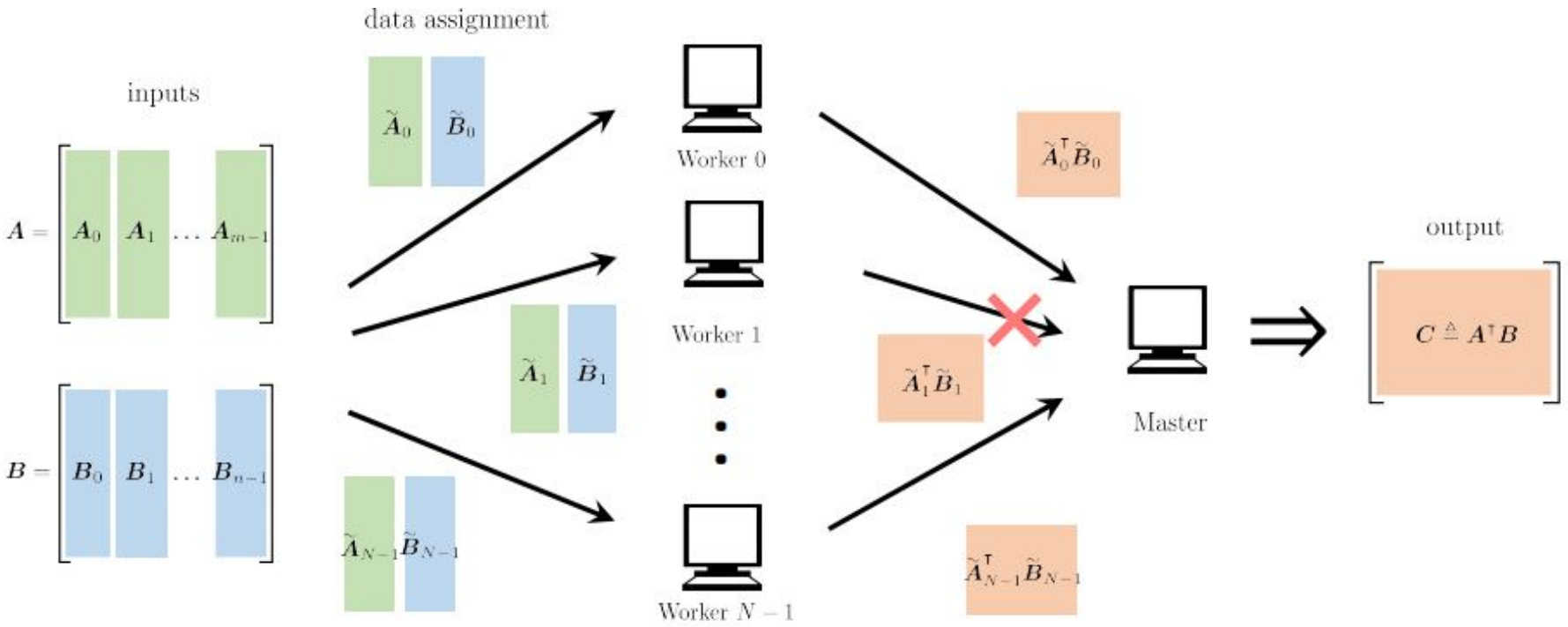
Использование распределенных вычислений позволит, например, ускорить процедуру обучения нейронной сети. Цель заключается в том, что задача делится на подзадачи, а подзадачи независимо выполняются на отдельных узлах, и далее результаты собираются вместе для получения решения изначальной задачи. Специфика распределенных систем такова, что некоторые узлы могут отвечать с задержками или вовсе не возвращать результат. Процедуры распределенных вычислений должны быть устойчивы к такого рода событиям.

Постановка задачи

Цели:

1. Рассмотреть задачу распределенного перемножения матриц $C = A\dot{B}$ над вещественным полем.
2. Реализовать подход на основе полиномиальных кодов. Параметры таковы: каждая матрица делится на 7 подматриц, т.е. число промежуточных произведений (подзадач) $K = 49$.
3. Варьировать число не ответивших узлов S, полагая общее число узлов $N = K + S$, и измерять среднюю ошибку $e = ||C - C'||/||C||$, где $||.||$ - норма Фробениуса, C - правильный ответ, C' - результат распределенного вычисления. Построить e от S.
4. Объяснить полученный результат, сделать вывод об использовании матриц Вандермонда в распределенных вычислениях.

Схема работы



Вычисление выполняется с использованием распределенной системы с главным узлом и рабочими узлами, каждый из которых может хранить $\frac{1}{m}$ долю A и $\frac{1}{n}$ долю B для некоторых параметров $m, n \in N_+$. Подматрицы обозначаются $\tilde{A}_i$ и $\tilde{B}_i$. Каждый из $N-1$ рабочих затем вычисляет произведение $\tilde{A}_i{}^T \tilde{B}_i$ и возвращает результат на главный узел.

Кодирования:

Определим полиномиальный код $\tilde{A}_i = \sum_{j=0}^{m-1} A_j x_i^j$, $\tilde{B}_i = \sum_{k=0}^{n-1} B_k x_i^{km}$.

Каждый узел i вернет $\tilde{C}_i = \tilde{A}_i{}^T \tilde{B}_i = \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} A_j B_k x_i^{j+km}$

Полином $h(x_i) = h_0 + h_1 x_i + +h_(K-1)x_i^(K-1)$

Чтобы декодировать на главном узле необходимо восствновить $\hat{y} = h^T G^T$

$$[y_1 ... y_i ... y_N] = [h_1 ... h_j ... h_k] \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ x_1^{K-1} & \cdots & x_N^{K-1} \end{bmatrix} \tag{1}$$

Описание:

Чтобы проверить эффективность предлагаемого нами полиномиального кода, реализуем алгоритм на Python, используя библиотеку mpi4py. Входные матрицы случайным образом генерируются как две матрицы numpy размером 8000 на 8000, а затем кодируются и назначаются рабочим на этапе предварительной обработки. Каждый рабочий хранит 14 частей каждой входной матрицы. На этапе вычислений каждый рабочий вычисляет произведение назначенных им матриц, а затем возвращает результат с помощью MPI.Comm.Isend(). Мастер активно слушает ответы от 17 рабочих узлов через MPI.Comm.Irecv() и использует MPI.Request.Waitany() для продолжения опроса самого раннего выполненного запроса. Получив 16 ответов, мастер прекращает прослушивание и начинает расшифровывать результат. Для достижения максимальной производительности мы реализуем алгоритм на основе декодирования Рида-Соломона.

References

1. *Q. Yu, M. Maddah-Ali, A. Avestimehr* APolynomial Codes: an Optimal Design for High-Dimensional Coded Matrix Multiplication, arxiv:1705.10464
2. *Li Tang , Konstantinos Konstantinidis, and Aditya Ramamoorthy* Erasure Coding for Distributed Matrix Multiplication for Matrices With Bounded Entries
3. Repository of Erasure Coding for Distributed Matrix Multiplication for Matrices With Bounded Entries. Accessed: 2018. [Online]. Available: `https://bitbucket.org/kkonstantinidis/stragglermitmm/src/master`

Спасибо

Спасибо за внимание!