Тема работы:

Использование OpenMP для ускорения процесса обучения нейронных сетей

Состав коллектива:

1. Пазников Алексей Александрович, к.т.н., с.н.с. СПбГЭТУ «ЛЭТИ», руководитель
2. Мохаммед Омар Таха Мохаммед, аспирант. СПбГЭТУ «ЛЭТИ», исполнитель

Информация о гранте:

РНФ, проект № 22-21-00686 «Алгоритмы и программные средства оптимизации выполнения параллельных программ в модели удаленного доступа к памяти», руководитель – Пазников А.А., 2022-2023

Научное содержание работы:

1. **Постановка задачи:**

Цель данного исследования - улучшить эффективность процесса обучения нейронных сетей с использованием OpenMP для параллельных вычислений. Наш подход заключается в переходе от традиционных последовательных алгоритмов обучения к параллельному выполнению задач, используя общую память. Объективы исследования включают:

1. Изучение существующих алгоритмов обучения нейронных сетей, подходящих для параллелизации на основе OpenMP.
2. Разработка и выполнение параллельных алгоритмов обучения с учетом общей памяти.
3. Разработка эффективных стратегий разделения данных и распределения задач между процессорами.
4. Оценка эффективности предложенных методологий по сравнению с традиционными последовательными подходами.
5. Интерпретация результатов на различных наборах данных и сравнение с существующими методами.

Ожидаемый результат - создание высокоэффективных параллельных алгоритмов обучения в рамках фреймворка OpenMP, что приведет к снижению внутренних накладных расходов и уменьшению времени выполнения программ.

2. **Современное состояние проблемы:**

Существующие исследования по ускорению обучения нейронных сетей показали положительные результаты, особенно при использовании графических ускорителей (GPU) [1]. Однако, с увеличением числа параметров в нейросетевых моделях, обучение становится сложной задачей [3]. Некоторые исследователи, такие как J. Nickolls и его коллеги [4], а также S. Che и его коллеги [5], предложили методы распараллеливания с использованием стандартов MPI, PThreads и OpenMP, что позволяет сократить время вычислений. В отличие от этих работ, наша исследование сфокусирована на обобщенном подходе, применимом к различным наборам данных, и предоставляет подробную экспериментальную оценку.

Другие исследователи, такие как Z. Meng и его коллеги [6], исследовали распараллеливание и оптимизацию алгоритмов классификации, основанных на графах и методах решения дифференциальных уравнений в частных производных, с использованием OpenMP. Они оптимизировали размещение данных для повышения эффективности доступа к кэш-памяти и применили OpenMP для распараллеливания наиболее трудоемких частей своего

подхода. Однако, проведенный анализ выявил основные проблемы и слабые места их реализации.

В существующих подходах каждый обучающий пример обрабатывается по одному набору на каждой итерации обучения, что приводит к увеличению времени обучения нейронных сетей [7]. Одним из распространенных решений является использование мини-пакетного обучения (mini-batch training) [8], при котором нейронная сеть обрабатывает подмножество обучающих примеров на каждой итерации и затем агрегирует обновления обучения. Однако, это может вызывать дополнительные накладные расходы в процессе обучения для приложений, требующих больших подмножеств обучающих примеров [9]. J. Duchi и его коллеги [10] предложили параллельный подход к обучению, который минимизирует эту задержку.

1. O. Yadan, K. Adams, Y. Taigman, M. Ranzato. Multi-GPU training of convnets //2013. arXiv:1312.5853.
2. C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A. Y. Ng, K. Olukotun Map-Reduce for machine learning on multicore //Proceedings of the 19th International Conference on Neural Information Processing Systems, NIPS'06, MIT Press, Cambridge, MA, USA, 2006, p. 281–288o.
3. L. Mai, A. Koliousis, G. Li, A.-O. Brabete, P. Pietzuch. Taming hyper-parameters in deep learning systems //SIGOPS Oper. Syst. Rev. 53 (2019) 52–58. https://doi.org/10.1145/3352020.3352029.
4. J. Nickolls, I. Buck, M. Garland, K. Skadron. Scalable parallel programming with CUDA: Is CUDA the parallel programming model that application developers have been waiting for? //Queue 6 (2008) 40–53. doi:10.1145/1365490.1365500.
5. S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, K. Skadron. A performance study of generalpurpose applications on graphics processors using CUDA //Journal of Parallel and Distributed Computing 68 (2008) 1370–1380.https://doi.org/10.1016/j.jpdc.2008.05.014.
6. Z. Meng, A. Koniges, Y. H. He, S. Williams, T. Kurth, B. Cook, J. Deslippe, A. L. Bertozzi. Openmp parallelization and optimization of graph-based machine learning algorithms //Lawrence Berkeley National Laboratory 9903 (2016).
7. V. Md, S. Misra, G. Ma, R. Mohanty, E. Georganas,A. Heinecke, D. Kalamkar, N. K. Ahmed, S. Avancha, Distgnn. Scalable distributed training for large-scale graph neural networks //Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '21, Association for Computing Machinery, New York, NY, USA, 2021. doi:10.1145/3458817.3480856.
8. M. Li, T. Zhang, Y. Chen, A. J. Smola. Efficient Mini-batch Training for Stochastic Optimization //In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2014 https://www.cs.cmu.edu/~muli/fi le/minibatch_sgd.pdf
9. L. Ziyin, K. Liu, T. Mori, M. Ueda. On minibatch noise: Discrete-time sgd, overparametrization, and bayes //ArXiv abs/2102.05375 (2021).
10. S. Chaturapruek, J. C. Duchi, C. Ré. Asynchronous stochastic convex optimization: the noise is in the noise and sgd don't care. //C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 28, Curran Associates, Inc.,2015.

3. Подробное описание работы, включая используемые алгоритмы:

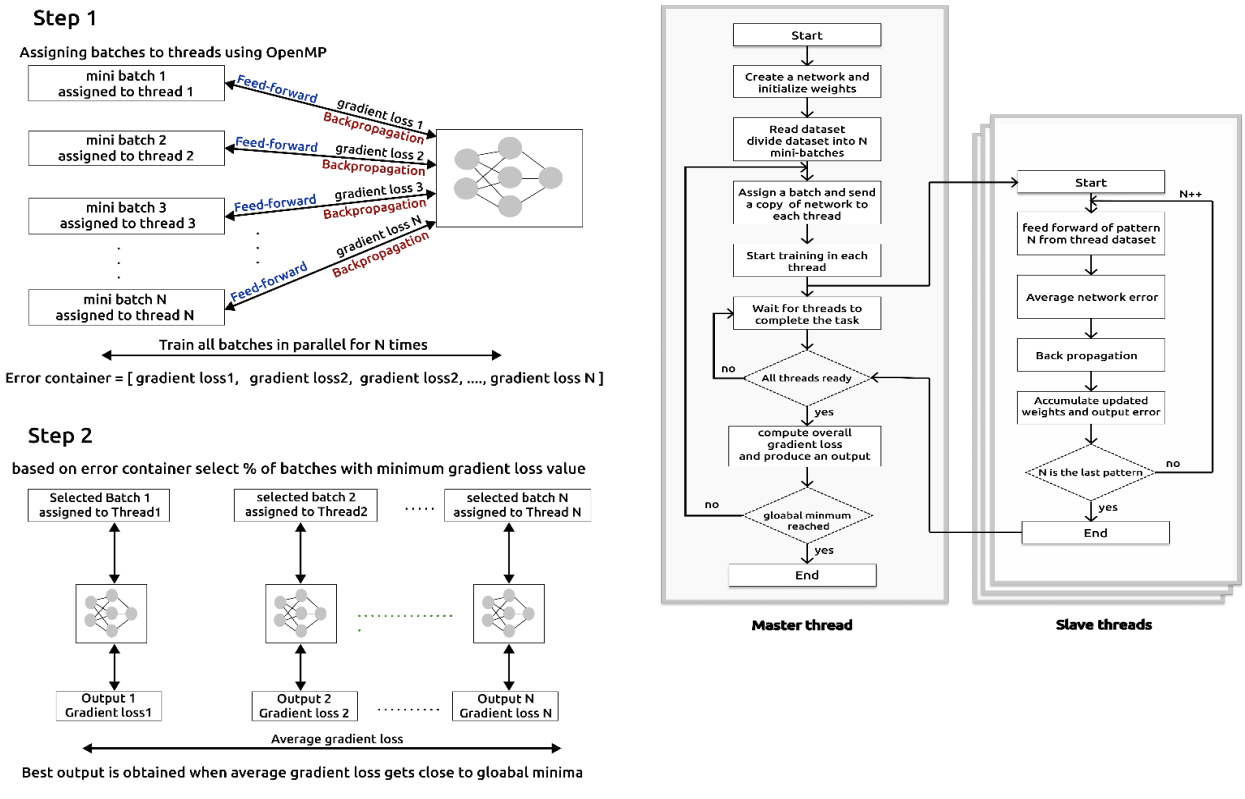
Предложенный подход (рис. 1a) заключается в параллельном распараллеливании обучения мини-пакетов нейронных сетей. Вместо того, чтобы усреднять градиентные потери всех мини-пакетов после нескольких итераций, мы исключаем некоторые мини-пакеты на основе их градиентных потерь. Обучение выполняется в два этапа: на первом

этапе каждый пакет обрабатывается в отдельном потоке в параллельных потоках. Затем, на втором этапе, выбирается подмножество мини-пакетов с наименьшими градиентными потерями для дальнейшего обучения, и остальные исключаются. Для выбора подмножества используется гиперпараметр *k*, который определяет процент выбранных мини-пакетов.

На втором шаге значения градиентных потерь сохраняются в контейнере ошибок. Затем первый шаг повторяется в течение m итераций, и градиентные потери усредняются, пока среднее значение не перестанет уменьшаться. Гиперпараметр m определяет максимальное количество итераций.

Реализация в OpenMP (рис. 1б) включает создание новой модели нейронной сети в главном потоке, разбиение обучающих примеров на мини-пакеты и запуск параллельных рабочих потоков. Каждый рабочий поток обрабатывает свой мини-пакет, выполняет обучение и обновляет значения весов. Затем обновленные значения весов объединяются в главном потоке. Процесс повторяется до достижения заданного количества итераций или пока потери градиента не перестанут уменьшаться.

Использование гиперпараметров *n, c, z, m, k* и *q* подразумевается в соответствии с описанными экспериментами и эмпирическими наблюдениями.



а *б*

Рис. 1. Схема предлагаемого подхода: (а) двухэтапная схема распараллеливания, (б) реализация параллельного алгоритма обучения методом обратного распространения

4. Полученные результаты

Для экспериментов был использован набор данных BanknoteAuthenticationDataset, который представляет задачу двоичной классификации. Набор содержит d = 1372 образца банкнот с 4 предикторными переменными. Цель состоит в предсказании подлинности

банкноты на основе измерений, сделанных по фотографии. В данной задаче выходной слой нейронной сети состоит из одного нейрона, выдающего вероятность в диапазоне от 0 до 1.

Была выбрана задача бинарной классификации для оценки предложенного подхода из-за её низкой вычислительной стоимости и меньшего количества гиперпараметров по сравнению с другими классификаторами, а также из-за конкурентоспособных показателей точности. Данный набор данных выбран для более доступной визуализации и изучения аспектов производительности.

Структуры нейронных сетей были выбраны на основе предварительных экспериментов. Первая сеть состоит из 3 скрытых слоев с 18 нейронами каждый и 5 слоев в целом, включая входной и выходной; вторая сеть имеет 4 скрытых слоя, каждый из которых состоит из 24 нейронов, и 6 слоев в целом. Сложность нейронных сетей, используемых в экспериментах, сопоставима с сетями, применяемыми на практике.

Экспериментальное исследование проводилось на кластере Информационно-вычислительного центра Новосибирского государственного университета.

Предложенный подход позволяет сократить время обучения, однако может иметь немного меньшую точность по сравнению с оригинальным мини-пакетным подходом. Вероятная причина заключается в том, что предложенный подход исключает некоторые обучающие наборы в процессе обучения.

На рис. 2 приведено сравнение ускорения двух подходов к распараллеливанию: существующего мини-пакетного обучения и подхода, описанного в данной работе.

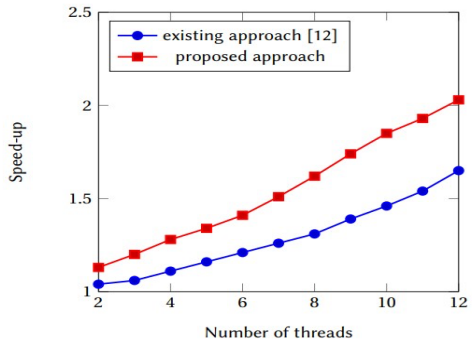


Рис. 2. Ускорение предложенного распараллеливания, по сравнению с существующим мини-пакетным распараллеливанием

Предложенный подход достигает наилучшего ускорения с использованием среднего размера пакета и числа потоков, соответствующего числу процессорных ядер. На рис. 2 показано, что наибольшая скорость обучения достигается при использовании 12 потоков.

Сублинейное ускорение (меньше числа потоков) объясняется ограничениями контролируемого обучения на основе мини-пакетного подхода. В этом подходе возникают временные затраты при объединении обновлений градиентных потерь и синхронизации потоков. Кроме того, обновления внутри пакета становятся устаревшими, так как модель обновляется только после обработки всего пакета данных. Предполагается, что эти ограничения будут устранены в последующих исследованиях..

5. Эффект от использования кластера в достижении целей работы:

Использование кластера Информационно-вычислительного центра Новосибирского государственного университета позволило достичь следующих эффектов в рамках работы:

1. Ускорение обучения: За счет распараллеливания процесса обучения и использования нескольких вычислительных ресурсов, удалось значительно сократить время обучения нейронных сетей, что повысило эффективность и производительность исследования.

2. Улучшение точности: Использование более мощных вычислительных ресурсов, предоставляемых кластером, позволило проводить более глубокие и сложные эксперименты, что привело к повышению точности и качества моделей нейронных сетей.

3. Более масштабируемые эксперименты: Кластер предоставил возможность проводить эксперименты на больших объемах данных, что способствовало более полному и всестороннему исследованию предложенного подхода и его применимости в различных сценариях.

4. Расширение возможностей исследования: Использование кластера позволило рассмотреть и оценить различные варианты архитектур нейронных сетей, подходов к параллелизации и оптимизации обучения, что дало глубокое понимание и инсайты в области нейронных сетей и их применения.

Таким образом, использование кластера в достижении целей работы существенно повысило эффективность, точность и масштабируемость проводимых экспериментов, а также расширило возможности исследования в области нейронных сетей.

Перечень публикаций, содержащих результаты работы

1. O. T. Mohammed, A. A. Paznikov and S. Gorlatch, "Accelerating Neural Network Training Process on Multi-Core Machine Using OpenMP," 2022 III International Conference on Neural Networks and Neurotechnologies (NeuroNT), Saint Petersburg, Russian Federation, 2022, pp. 7-11, doi: 10.1109/NeuroNT55429.2022.9805549.