

ЭФФЕКТИВНОЕ ОБУЧЕНИЕ НЕЙРОННЫХ СЕТЕЙ НА ОСНОВЕ ТЕХНОЛОГИИ РАСПАРАЛЛЕЛИВАНИЯ

В. В. Мацкевич

Белорусский государственный университет, г. Минск;

anatoly_mak@yahoo.com;

науч. рук. – В.В.Краснопрошин, д-р техн. наук, проф.

В данной работе были проведены экспериментальные исследования эффективности использования технологии распараллеливания в обучении без учителя НС прямого распространения в пакетном режиме. Был разработан и реализован комбинированный алгоритм обучения.

Ключевые слова: нейронные сети; технология распараллеливания; комбинированный алгоритм обучения.

1. ПОСТАНОВКА ЗАДАЧИ

Задано ограниченное количество изображений различных объектов. Необходимо организовать эффективное обучение без учителя нейронной сети прямого распространения в пакетном режиме на стандартном компьютере, содержащем многоядерный процессор и видеокарту с использованием технологии распараллеливания.

2. ТЕОРЕТИЧЕСКИЕ ИССЛЕДОВАНИЯ

Для решения поставленной задачи построим комбинированный алгоритм обучения нейронной сети прямого распространения. Необходимо учитывать, что устройства могут обладать разной вычислительной мощностью за счет различных комбинаций процессора и видеокарты, и нужно перераспределить нагрузку между устройствами так, чтобы достигалась максимальная производительность всей системы. Для достижения максимальной производительности необходимо, чтобы многоядерный процессор и видеокарта работали одновременно и время вычислений на обоих устройствах совпадали.

Пусть P – необходимый объем вычислений для обучения НС на одном пакете входных сигналов. Пусть P_1 - объем вычислений на видеокарте, P_2 – объем вычислений на многоядерном процессоре. Обозначим вычислительную мощность видеокарты как gpu , многоядерного процессора – как cpu . Получим следующую систему уравнений:

$$\begin{cases} \frac{P_1}{gpu} = \frac{P_2}{cpu} \\ P_1 + P_2 = P \end{cases}, \quad (1)$$

Второе уравнение указывает на то, что видеокарта и процессор должны в сумме обеспечить обучение НС на одном пакете входных сигналов.

Решение системы (1):

$$\begin{cases} P_1 = \frac{P}{1 + \frac{cpu}{gpu}}, \\ P_2 = P - P_1 \end{cases} \quad (2)$$

В данном решении есть серьезная проблема – неизвестно соотношение мощности многоядерного процессора к мощности видеокарты.

Решение проблемы следующее. Пусть x – объем вычислений, необходимый для прогона одного пакета входных сигналов при обучении без учителя. Обозначим время прогона на видеокарте как gpu_time , на многоядерном процессоре как cpu_time . Тогда имеет место система:

$$\begin{cases} gpu_time = \frac{x}{gpu} \\ cpu_time = \frac{x}{cpu} \end{cases} \quad (3)$$

Поделив первое равенство системы (3) на второе получим искомое соотношение:

$$\frac{cpu}{gpu} = \frac{gpu_time}{cpu_time} \quad (4)$$

Таким образом, комбинированный алгоритм обучения работает следующим образом. При прогоне первого пакета входных сигналов в обучении без учителя первая половина пакета обрабатывается на видеокарте, вторая – на процессоре. Замеряется время прогона на каждом из устройств. После этого, используя формулу (4) и систему (2), определяется количество входных сигналов, которое будет обрабатываться на видеокарте и на процессоре. В качестве P , P_1 и P_2 берется количество входных сигналов, а не вычислений, т.к. для каждого входного сигнала количество операций примерно одинаковое. Дальнейшая обработка осуществляется с учетом системы (2) и время прогонов не замеряется.

Сама обработка на видеокарте и многоядерном процессоре осуществляется с помощью гибридного алгоритма распараллеливания. Он заключается в следующем. Параллельно обрабатывается несколько входных сигналов, при этом каждый поступающий на обработку входной сигнал «обслуживается» несколькими вычислительными потоками.

Проведем теоретический анализ эффективности гибридного алгоритма. Пусть пакет содержит m входных сигналов, обучается один слой нейронной сети и его размер – l_2 , размер слоя перед обучаемым – l_1 . Пусть одновременно обрабатывается x_1 входных сигналов, каждый из которых обслуживается x_2 вычислительными потоками (каждый поток производит вычисления для определенных нейронов обучаемого слоя).

Тогда количество последовательно выполненных операций для одного пакета входных сигналов будет равно величине.

$$\begin{cases} f(x_1, x_2) = \left\lceil \frac{m}{x_1} \right\rceil l_1 \left\lceil \frac{l_2}{x_2} \right\rceil \\ x = x_1 x_2 \end{cases} \quad (5)$$

Где, $f(x_1, x_2)$ - количество последовательно выполненных операций, x – количество вычислительных потоков. Следовательно оптимальные значения параметров

$$\begin{cases} x_1 = m, x_2 = l_2 \\ x = ml_2 \\ f(m, l_2) = l_1 \end{cases} \quad (6)$$

Для процессора использовался гибридный алгоритм распараллеливания с одной особенностью: x_2 равен единице.

3. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ

Для решения поставленной задачи была разработана методика организации экспериментов. Была скачана выборка «STL-10» из репозитория Стенфордского университета [1]. В ней содержится сто тысяч немаркированных изображений для обучения без учителя. Каждое отдельное изображение является цветным и имеет размер 96x96 пикселей. Изображение описывается в следующем формате. Сначала 96x96 чисел описывают содержание красного цвета в пикселях, затем 96x96 чисел – зеленого, затем 96x96 чисел – синего. Все числа целые от 0 до 255. Каждое немаркированное изображение может содержать произвольный объект [2].

Для экспериментов изображения были сжаты в следующем порядке. Разбиваем изображение на блоки по $A \times B$ пикселей и заменяем каждый блок одним пикселем. У результирующего пикселя содержание каждого из цветов является средним арифметическим содержания каждого из цветов этих AB пикселей. В экспериментах A и B были равны двум.

Конфигурация ЭВМ: nvidia 1050, intel i5-4670k 3,8 GHz, ОЗУ 2x8 Gb 1600 MHz, samsung 850 pro 256 Gb, ОС Ubuntu 16.04.3. Все НС реализованы в Nsight Eclipse Editon, компилятор nvcc 7.5 (использовались библиотеки CUDA, OpenMP). Замер времени - функция «gettimeofday».

Структура любой реализации следующая:

1. Считывание входных данных
2. Сжатие изображений
3. Инициализация весов НС, цикл по слоям:
 - «Бесконечный» цикл обучения по пакетам из 6561 изображения

- Выполняется исследуемый алгоритм обучения без учителя
 - Замеры времени обучения по 15 пакетам и проверка условия выхода из цикла
 - Перевод изображений на следующий слой (при необходимости)
- Условие выхода из цикла – «суммарная среднеквадратическая ошибка по 15 пакетам изображений увеличилась».

4. РЕЗУЛЬТАТЫ

Таблица 1

Эффективность комбинированного алгоритма

Название алгоритма	Комбинированный (видеок. + проц.)	Комбинированный (видеок.)	Комбинированный (проц.)	Последоват. (проц.)
Время прогона, с	63	104	134	289

Время прогона указано для пятнадцати пакетов по 6561 изображений в каждом.

Из результатов экспериментов видно, что комбинированный алгоритм распараллеливания наиболее эффективно использует вычислительные ресурсы стандартного компьютера, содержащего многоядерный процессор и видеокарту. Комбинированный алгоритм обучения на двух устройствах оказался на 65 % быстрее, чем его реализация отдельно на видеокарте, и на 113 % быстрее, чем его реализация отдельно на процессоре. Кроме того, реализация построенного алгоритма на двух устройствах оказалась на 359 % быстрее последовательного алгоритма обучения на многоядерном процессоре. Вышеуказанные результаты говорят о возможном широком практическом применении комбинированного алгоритма обучения.

5. ВЫВОДЫ

В данной работе были проведены экспериментальные исследования эффективности использования технологии распараллеливания в обучении без учителя НС прямого распространения в пакетном режиме. Был разработан и реализован комбинированный алгоритм обучения, который эффективно использует вычислительные ресурсы стандартного компьютера, содержащего многоядерный процессор и видеокарту. Разработанный алгоритм способен использовать любое количество вычислительных устройств любой мощности [3].

Библиографические ссылки

1. URL: <http://academictorrents.com/details/a799a2845ac29a66c07cf74e2a2838b6c5698a6a>
2. URL: <https://web.archive.org/web/20110803194852/http://www.stanford.edu/~acoates/stl10/>
3. *Мацкевич В. В.* Нейросетевая технология обработки данных // Дипломная работа 2018.

АВТОМАТИЗАЦИЯ ПРОЦЕССОВ МОНИТОРИНГА ДЛЯ ВЫРАЩИВАНИЯ РАСТЕНИЙ В ИСКУССТВЕННЫХ УСЛОВИЯХ

А. А. Михальченко

*Белорусский государственный университет, г. Минск;
strrife@gmail.com;*

науч. рук. – В. В. Краснопрошин, д-р техн. наук, проф.

В рамках исследования были спроектированы и внедрены в лаборатории Биологического факультета БГУ алгоритмы по определению и анализу состояния растений. Рассмотрены основные подходы к решению сходных задач, выделены основные правила.

Ключевые слова: феномика; обработка изображений; распознавание образов; фенотипирование растений.

ВВЕДЕНИЕ

В результате процесса глобализации в современном мире наблюдается значительные изменения во всех сферах человеческой деятельности. Возникают новые вызовы и проблемы, для решения которых отсутствуют методы и программные средства, поэтому в настоящее время появляются новые научные дисциплины, которые формируются в соответствии с этими вызовами и направлены на решение этих проблем.

ФЕНОМИКА

Поскольку феномика растений является относительно новой областью биоинформатики, мы считаем необходимым наряду с основной проблемой и подходами к ее решению привести небольшое описание исследуемой области.

Феномика – это область биоинформатики, связанная с измерением феномов – физических и биохимических черт организмов, отражающих динамику изменений в ответ на генетические мутации и влияния окружающей среды. Не только в феномике, но в экологии и ботанике в целом, одной из главных целей исследования является изучение фенотипов организмов в различных экосистемах, будь то естественные экосистемы