DOI 10.37882/2223-2966.2023.08.25

ВЫБОР АППАРАТНОЙ ПЛАТФОРМЫ ДЛЯ МАШИННОГО ОБУЧЕНИЯ

CHOOSING A HARDWARE PLATFORM FOR MACHINE LEARNING

V. Perepelkin

Summary. Despite the fact that the basic principles of neural networks were formulated by Warren McCulloch and Walter Pitts back in 1947, today's time can be called as the era of the development of machine learning and artificial intelligence without exaggeration. It is over the past few years that there has been a breakthrough in the development of neural network technologies and their industrial applications. Neural networks have been successfully used in computer vision, natural language processing, building autopilot systems, robotics, etc. To a large extent, all this became possible due to the development of computer technology and the appearance of new hardware platforms that show high performance when handle mathematical operations, which also facilitated to create and train deep neural networks with a large number of layers and neurons. The problem areas of training neural networks and the efficiency of using various hardware platforms by an example of training LeNet-5 neural network with MNIST dataset are considered at the research work.

Keywords: machine learning, neural networks, hardware platforms, processors.

Перепелкин Вадим Юрьевич
аспирант, Московский государственный университет технологий и управления (Первый казачий университет), vup@rambler.ru

Аннотация. Не смотря на то что основные принципы работы нейросетей были сформулированы Уорреном Мак-Каллоком и Уолтером Питтсом еще в 1947 году, именно сегодняшнее время без преувеличения можно назвать эрой развития машинного обучения и искусственного интеллекта. Именно за последние несколько лет произошел прорыв в развитии нейросетевых технологий и их промышленного применения. Нейросети стали успешно использоваться в задачах компьютерного зрения, обработки естественного языка, построения автопилотируемых систем, роботостроении и т.д. Во многом все это стало возможным благодаря развитию вычислительной техники и появлению новых аппаратных платформ, которые показывают высокую производительность при выполнении математических операций, что в свою очередь позволило создавать и обучать глубокие нейросети с большим количеством слоев и нейронов. В исследуемой работе рассмотрены проблемные аспекты обучения нейросетей и эффективность использования различных аппаратных платформ на примере обучения нейросети LeNet-5 на датасете MNIST.

Ключевые слова: машинное обучение, нейросети, аппаратные платформы, процессоры.

Нейронные сети состоят из множества искусственных нейронов, которые объединяются в слои и взаимодействуют друг с другом. Организация и структура нейросетей могут варьироваться в зависимости от типа решаемой задачи, но существуют базовые компоненты и принципы устройства нейросетей. Основные строительные блоки нейросети — это искусственные нейроны, которые имитируют работу биологических нейронов и обрабатывают информацию. Каждый нейрон принимает входные сигналы, умножает их на соответствующие веса и производит выходной сигнал, который проходит через активационную функцию. Схематично принцип работы искусственного нейрона показан на Рисунке 1.

Нейроны в свою очередь объединяются в слои. Входной слой принимает входные данные, выходной слой выдает результат, а скрытые слои выполняют промежуточные вычисления между входом и выходом. Одним из распространенных типов нейросетей являются полносвязные нейросети. Они состоят из нескольких слоев нейронов, где каждый нейрон в предыдущем слое соединен с каждым нейроном в следующем слое. На Рисунке 2 показан пример полносвязной нейросети с несколькими слоями.

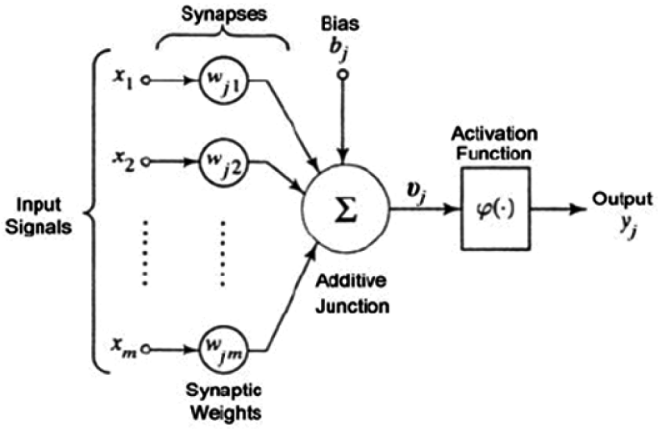

Рис. 1. Принцип работы искусственного нейрона

$$sum = \vec{X}^T\vec{W} + \vec{B} = \sum_{i=1}^{n} x_i w_i + b_i$$

$$out = \varphi(sum)$$

Каждая связь между нейронами имеет ассоциированный вес, который определяет важность входного сигнала для выходного нейрона. Веса являются параметрами, которые модель обучает в процессе обучения нейросети. До начала обучения веса инициируются слу-

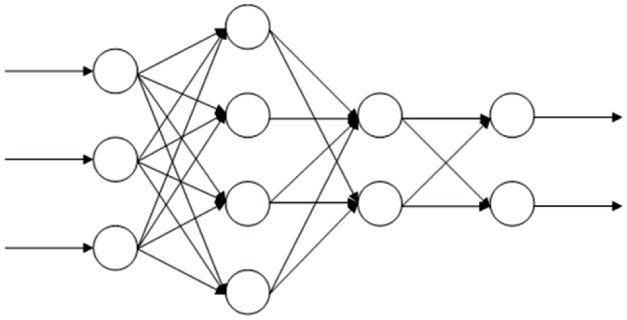

Рис. 2. Полносвязная нейросеть с несколькими слоями

чайными значениями. Для проведения обучения нужна обучающая выборка, которая представляет собой датасет из входных и ожидаемых выходных значений. На каждой итерации обучения на вход сети подаются входные данные, а нейросеть генерирует выходные данные ($y_{predicted}$).

$$y_{predicted} = \sum_{i=1}^{n} x_i w_i = \vec{X}^T \vec{W} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}^T \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} =$$

$$= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

В начале обучения генерируемые нейросетью выходные значения будут отличаться от ожидаемых. Для оценки расхождения полученных от ожидаемых значений используется функция ошибки (L). Существует несколько наиболее часто используемых функций ошибки, для примера возьмем среднеквадратичную ошибку.

$$L = \frac{1}{N} \sum_{i=1}^{N} \left(y_{predicted(i)} - y_{expected(i)} \right)^2$$

Задача обучения сводится к минимизации функции ошибки путем корректировки весов нейросети. Для минимизации функции ошибки используется градиентный спуск, который позволяет на каждой итерации обучения корректировать веса в направлении минимума функции ошибки.

$$\vec{w}^{(k+1)} = \vec{w}^k - \mu \nabla L\left(w^k \right)$$

где: k — итерация обучения сети; μ — шаг обучения; ∇L — градиент функции ошибки.

Для нахождения градиента, используются частные производные:

$$\nabla L\left(\vec{w} \right) = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_N} \end{bmatrix}$$

Корректировка весов нейросети в процессе обучения также называется обратным распространением ошибки. Вычисление градиента осуществляется на каждой итерации обучения. Кроме того, обычно обучение на одной и той обучающей выборке повторяется в течение нескольких эпох для повышения качества работы нейросети.

Как видно из формул, параметры нейросети представлены в виде векторов и матриц. Процесс обратного распространения ошибки и корректировки весов нейросети предполагает выполнение большего количества математических вычислений над матрицами, которые требуют большого количества вычислительных ресурсов аппаратной платформы, на которой выполняется обучение. При этом математические операции, выполняемые над матрицами, могут быть легко распараллелены для оптимизации скорости обучения. С учетом постоянного усложнения архитектур и количества обучаемых параметров обучение нейросетей может занимать значительное время. Рассмотрим аппаратные платформы, которые могут быть использованы для обучения нейросетей.

Центральный процессор (CPU) является основным компонентом любого компьютера и предназначен для выполнения самых разнообразных задач, и считается устройством общего назначения. Процессоры состоят из нескольких ядер, которые могут одновременно выполнять несколько инструкций. В общем виде CPU состоит из следующих компонентов:

- Арифметико-логическое устройство (ALU), которое выполняет арифметические (сложение, вычитание, умножение и деление) и логические операции над данными.
- Устройство управления, координирующего работу всех компонентов CPU. Оно извлекает инструкции из памяти, декодирует их и запускает соответствующие операции в ALU и других функциональных блоках.
- Регистры или кэш-память — это быстрая и небольшая память, которая хранит наиболее часто используемые данные и инструкции. Она находится непосредственно на процессоре и предназначена для ускорения доступа к данным, уменьшая задержки чтения из оперативной памяти.
- Шина представляет собой коммуникационный канал, который связывает различные компоненты CPU и позволяет им обмениваться данными и сигналами. Также CPU взаимодействует с оперативной памятью, где хранятся данные и инструкции программы и различными устройствами ввода — вывода.

Основным преимуществом CPU является его гибкость, которая позволяет использовать его для решения

широкого спектра задач, включая вычисления, управление памятью, операции ввода-вывода и выполнение инструкций программы. CPU идеально для подходит для выполнения на нем большинства приложений, которые запускаются на ПК. Платой за универсальность является низкая эффективность CPU для некоторых частных задач. Применительно к задаче машинного обучения узким местом в CPU является взаимодействие с регистрами для чтения и записи данных на каждой итерации выполнения математических вычислений, что отнимает много времени.

Альтернативой использования CPU является использование графических процессоров (GPU). С развитием компьютерных игр и трехмерной графики возникла потребность в быстрой и эффективной обработке графических данных. Такие операции, как рендеринг трехмерных моделей, текстурирование и освещение, требовали выполнения множества параллельных вычислений. Традиционные центральные процессоры (CPU) не могли обеспечить достаточную производительность для этих задач, поэтому появилась необходимость в специализированном устройстве — GPU.

На Рисунке 3 представлено отличие архитектуры CPU от GPU. GPU содержит намного больше ALU, что позволяет более эффективно распараллеливать выполнение математических и логических операций. Подобная архитектура позволяет GPU эффективно справляться со специфическими задачами, связанные с компьютерной графикой, майнингом криптовалют (т.к. предполагает расчет большого количества хешей), но при этом лишена той гибкости, которой обладает CPU. С точки зрения машинного обучения использование GPU позволяет распараллеливать математические операции, что значительно повышает скорость обучения. Но при этом GPU в рамках одного ALU все еще имеет туже проблему, что и CPU, чтение и запись данных в регистры все также является узким местом.

Еще одной альтернативой является использование тензорных процессоров (TPU). Тензор (термин применяется в сфере машинного обучения) — это математических объект, при помощи которого могут быть представлены векторы и матрицы различной размерности. Разработчиком TPU является компания Google, которая создала аппаратную платформу, адаптированную для решения задач машинного обучения. Основания идея подобных устройств состоит в том, что они, как и GPU, состоят из большого количества ALU, что позволяет распараллеливать математические вычисления. Но в тоже время TPU отличаются от GPU тем, что ALU в TPU соединены напрямую между собой, что позволяет одним ALU получать на вход результаты вычислений других, минуя операции чтения и записи регистров. Подобное технологическое решение устраняет узкое место, связанное с низкой скоростью обмена с регистрами, и с одной стороны позволяет ускорить математические вычисления, а с другой сократить расходы энергии. Также стоит отметить, что компания Google является не единственным разработчиком такого типа устройств, другие ИТ гиганты также имею своим аппаратные разработки в данной сфере.

Проведем эксперимент по обучению нейросети LeNet-5 на разных аппаратных платформах и замерим время, затрачиваемое на обучение. LeNet-5 является одной из первых сверточных нейронных сетей, была разработана Йаном Лекуном в 1998 году. Эта нейронная сеть была предназначена для распознавания рукописных цифр и стала одним из прорывов в области компьютерного зрения и глубокого обучения. Сеть состоит из 7-ми слоев, 3 сверточных, 2 пулинга и 2 полносвязных. Для обучения используется датасет MNIST, который состоит из 60 000 обучающих изображений. Каждое изображение представляет собой рукописную цифру от 0 до 9 размера 28x28 пикселей в оттенках серого. Сеть LeNet-5 является довольно простой и содержит сравнительно не большое количество обучаемых параметров, что позволяет провести обучение за короткое время.

Обучение проводилось на двух аппаратных платформах:

— CPU: Intel Xeon E5-1620v2 3.7GHz. Стоимость на момент написания статьи порядка 70$.

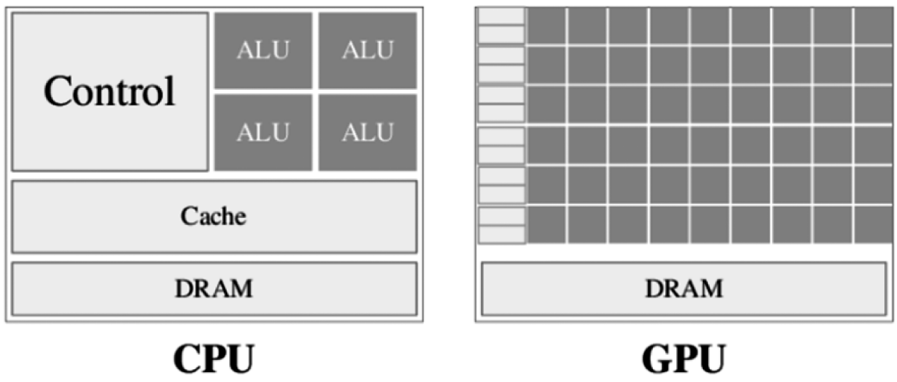

Рис. 3. Сравнение архитектур CPU и GPU

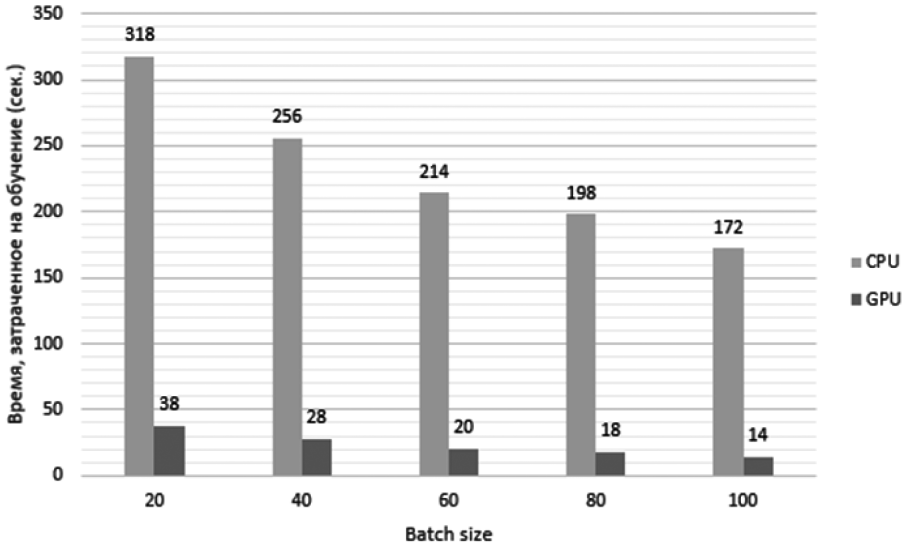

Рис. 4. Время, затраченное на обучение

— GPU: NVidia GeForce RTX 3060 12G. Стоимость на момент написания статьи порядка 230$.

Обучение проводилось на протяжении 10 эпох с learning rate равным 0.2. Batch size варьировался от 20 до 100 изображений.

На Рисунке 4 показаны результаты эксперимента. По результатам видно, что использование GPU позволило достичь примерно 10 кратного ускорения обучения по сравнению с использованием CPU. Но стоит также отметить, что стоимость двух аппаратных платформ отличается, используемый в эксперименте GPU был где-то в 3 раза дороже чем CPU. Но если аппроксимировать результаты эксперимента с точки зрения затраченного времени и стоимости оборудования, то получим, что использование GPU обеспечивает более низкую стоимость обучения. При этом использование TPU или других специфических аппаратных платформ для задач машинного обучения позволяет достичь еще большего снижения стоимости обучения за счет увеличения скорости обучения и меньшего потребления энергии. Несмотря на то, что эксперимент проводился на простой LeNet-5, обучение которой занимает мало времени и фактически может проводится на любом оборудовании, нужно не забывать, что обучение современных нейросетей, которые содержат кратно большее количество обучаемых параметров, может занимать несколько дней и недель. В этом случае использование более эффективной аппаратной платформы позволит значительно снизить время и стоимость, затрачиваемые на обучение. Таким образом выбор аппаратной платформы становится чрезвычайно важным аспектом машинного обучения.

ЛИТЕРАТУРА

1. Омельяненко, Я. Эволюционные нейросети на языке Python: практическое руководство / Я. Омельяненко; пер. с англ. В.С. Яценкова. — Москва: ДМК Пресс, 2020. — 310 с. — ISBN 978-5-97060-854-8.
2. Raschka, Sebastian Python Machine Learning: Third Edition / Sebastian Raschka, Vahid Mirjalili — Birmingham: Packt Publishing Ltd., 2015. — 741 с.
3. Боресков, А.В. Параллельные вычисления на GPU. Архитектура и программная модель CUDA: Учебное пособие. 2-е издание. / А.В. Боресков и др. Предисл.: В.А. Садовничий. — М.: Издательство Московского университета, 2015. — 336 с., — ISBN 978-5-19-011058-6.
4. Хеннесси, Д.Л. Компьютерная архитектура. Количественный подход. Издание 5-е. / Д.Л. Хеннесси, Д.А. Паттерсон — Москва: ТЕХНОСФЕРА, 2016. — 936 с. — ISBN 978-5-94836-413-1.
5. Жерон, Орельен Прикладное машинное обучение с помощью Scikit-Learn, Keras и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем. 2-е издание. / Орельен Жерон; пер. с англ. Ю.Н. Артеменко — СПб: ООО «Диалектика», 2020. — 1040 с. — ISBN 978-5-907203-33-4.

Журнал «Современная наука: актуальные проблемы теории и практики»