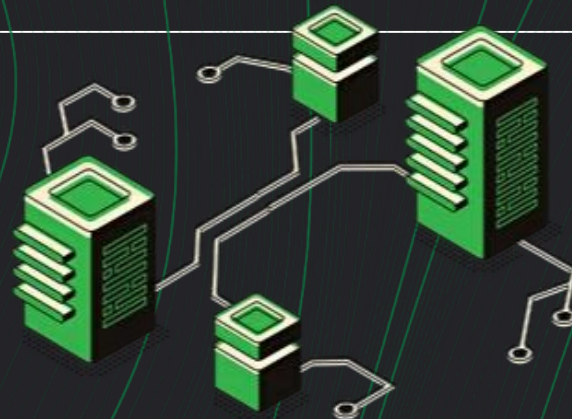


Технологический стек машинного обучения



Шпилевский Яромир

Ведущий разработчик First Line Software

Agenda

- Вычислительная модель нейрона.
- CPU и архитектура Фон Неймана.
- Классификация Флинна. SIMD подход. GPU и TPU.
- Нейропроцессоры.

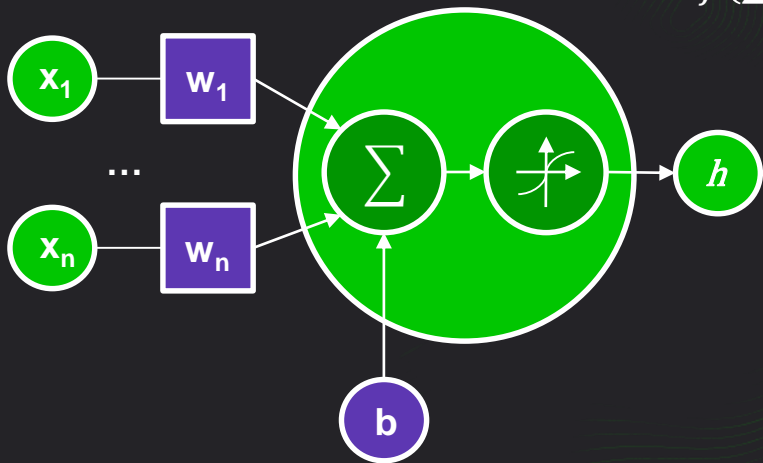
Зачем

- Всегда нужно понимать программно-аппаратный стек технологии, с которой работаете.
 - Для отладки.
 - **Для анализа производительности.**
 - С машинным обучением особенно актуально — ресурсоёмкие задачи, приходится думать об эффективности использования ресурсов.
 - ...

Не совсем нейрон

- Следует помнить, что мы оперируем не совсем нейроном, а его вычислительной моделью.

$$h = f(\sum_i^n w_i x_i + b)$$

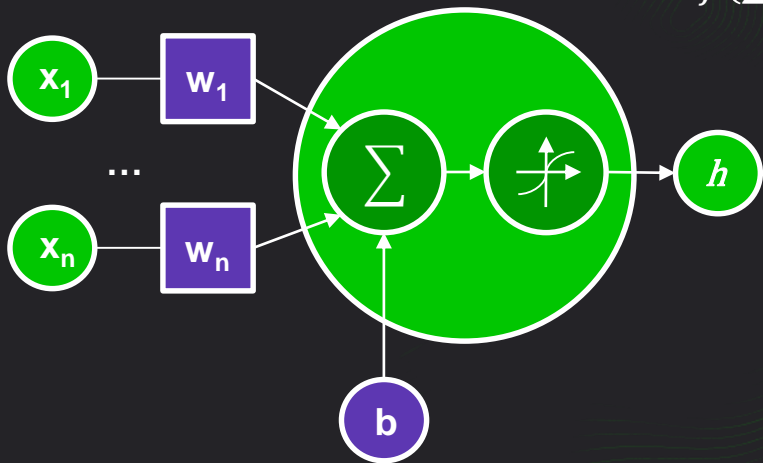


- По сути происходит моделирование работы одной вычислительной архитектуры (нейронных сетей) с помощью другой (компьютера).
 - Который, по большому счету, является всё той же архитектурой фон Неймана [\[1\]](#).

Почему важна параллельность

- В вычислительной модели нейрона многие операции можно производить параллельно.

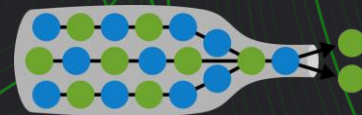
$$h = f(\sum_i^n w_i x_i + b)$$



- Благодаря этому можно получить большой прирост производительности.

Central Processing Unit (CPU)

- Есть несколько архитектурных принципов машины фон Неймана [1].
 - <...>
 - **Программное управление.** <...> Команды выполняются последовательно друг за другом. <...>
 - Single Instruction Single Data (SISD) по классификации Флинна. [2]
 - <...>
- Попытка распараллелить расчет модели на современных многоядерных процессорах даст относительно небольшой выигрыш (Multiple Instruction Multiple Data – MIMD).
 - x2?, x4?, x8? (в идеале, в случае отсутствия взаимного негативного влияния)
 - Реально – меньше, из-за конфликтов между ядрами.
 - Взаимное положительное влияние – тем более вряд ли. ☺
- Нужна ли нам полная независимость вычислительных ядер?

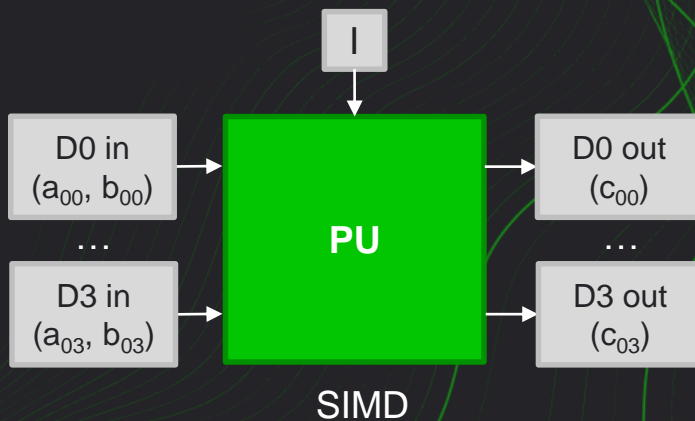
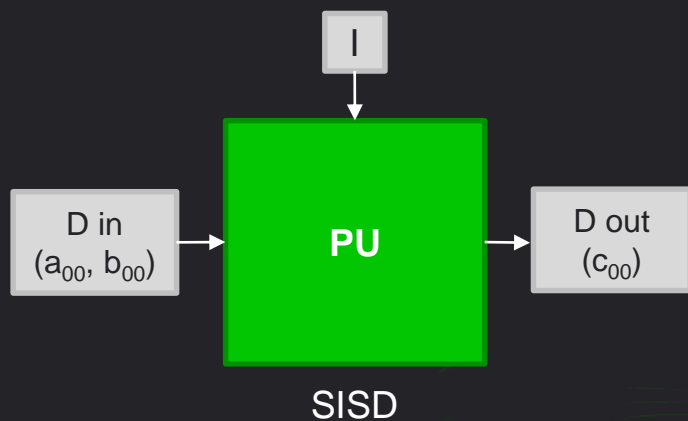


Graphics Processing Unit (GPU) в ML. Предпосылки

- В задачах компьютерной графики есть потребность в операциях над большими массивами данных.
- При этом это **одна** операция, которая оперирует большим массивом данных.
 - Например, сложить две матрицы.
- Почему бы не реализовать сопроцессор с SIMD архитектурой?
 - В дополнение к архитектуре фон Неймана.
- SIMD обработка очень выручает и в задачах машинного обучения.

GPU. SIMD обработка

- Одна инструкция, обрабатывающая множественные данные.
 - За **один** процессорный цикл.



$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 6 & 8 \\ 10 & 12 & 14 & 16 \\ 18 & 20 & 22 & 24 \\ 26 & 28 & 30 & 32 \end{bmatrix}$$

GPU. Технологии (1)

- NVIDIA CUDA (Compute Unified Device Architecture) [3]
 - Проприетарная технология NVIDIA.
 - Наиболее популярная технология.
- OpenCL (Open Computing Language) [4]
 - Открытый стандарт. Разрабатывается Khronos Group (организация, разрабатывающая стандарты OpenGL и Vulkan).
 - Получил меньшее распространение.
- AMD ROCm (Radeon Open Compute Module) [5]
 - Открытый стандарт. При этом вкладываются ресурсы коммерческой компании.
 - Пока небольшое распространение, но активно набирает большую долю рынка.
 - Интересный SDK. Компиляторы, как под CPU, так и под GPU, реализованы, как бекенд к LLVM.

GPU. Технологии (2)

- Intel Habana
 - Проприетарная технология Intel.
 - Пока небольшое распространение, но активно набирает большую долю рынка.
 - SynapseAI Software – собственный стек системного ПО. [6]
- HTC «Модуль»
 - Проприетарная технология HTC «Модуль».
 - Пока небольшое распространение, но активно набирает большую долю рынка.
 - Система-на-кристалле (System-on-Chip): ARM процессоры + ядра тензорных сопроцессоров.
 - Neuromatrix Deep Learning SDK. [7]

GPU. Технологии (3)

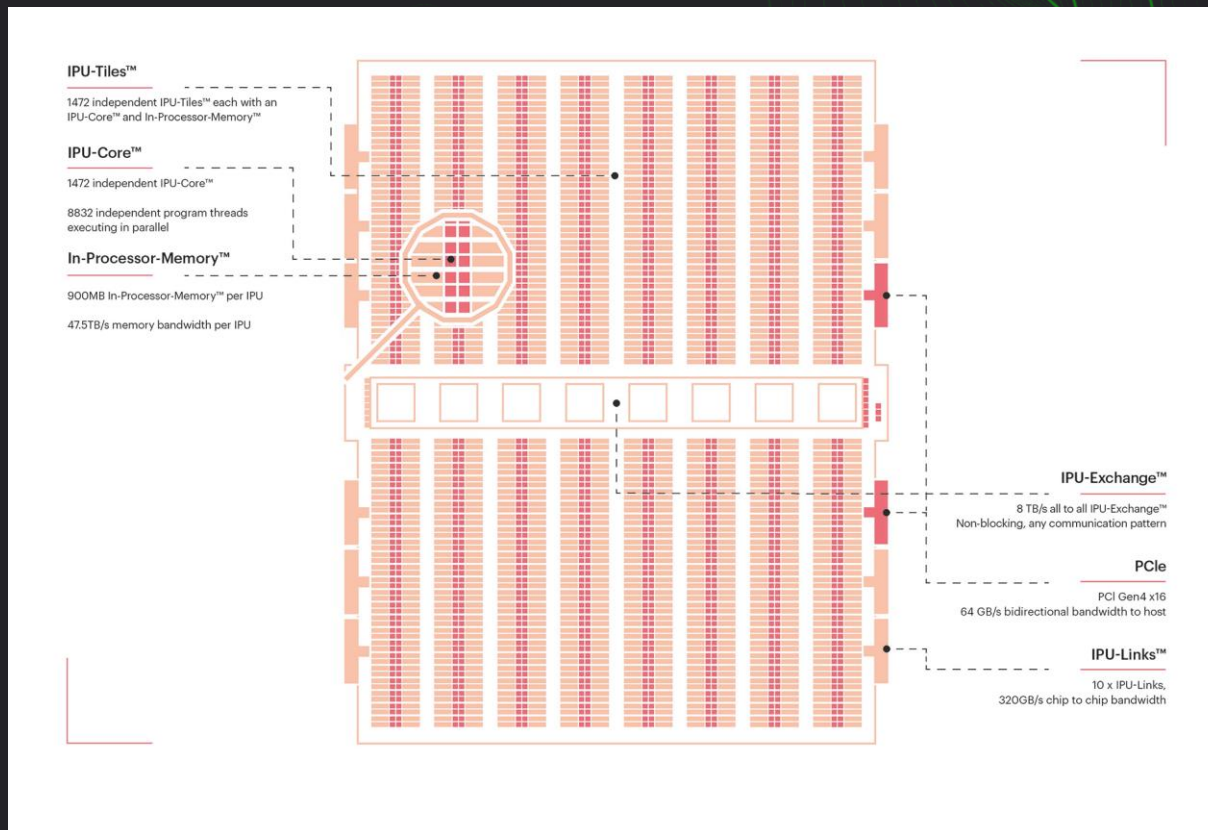
- Tenstorrent. [\[8\]](#)
 - Проприетарная технология.
 - Пока небольшое распространение.
 - Система-на-кристалле (System-on-Chip): RISC-V процессоры + ядра тензорных сопроцессоров.
 - RISC-V: архитектура набора инструкций (Instruction Set Architecture (ISA)) со свободной спецификацией (в отличие от ARM).

GPU. Технологии (4)

- Graphcore. [\[9\]](#)
 - Проприетарная технология.
 - Пока небольшое распространение.
 - Poplar SDK. [\[10\]](#)
 - «Плитки» ядер, соединённых между собой – по структуре распространения сигналов подход ближе к нейронным сетям «в железе».

GPU. Технологии (5)

- Graphcore. [9]



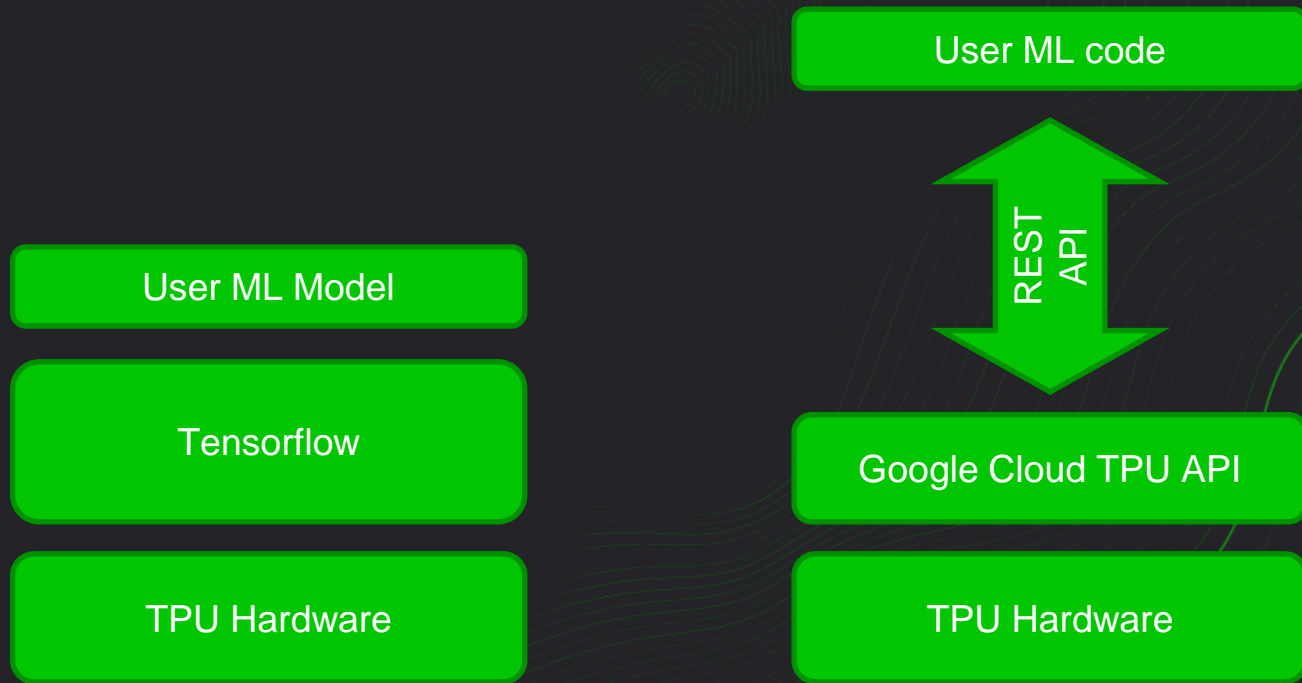
GPU. Технологии. О рынке

- Большой рост применений AI. Рынок растёт.
- Рынок молодой и ещё формирующийся:
 - Возможно возникновение новых игроков.
 - Лидерство существующих игроков – не догма.
 - Возможно, кто-нибудь найдёт технологическую особенность, которая изменит всё.

Tensor Processing Unit (TPU)

- Специализированный сопроцессор для обработки тензоров.
- В большинстве случаев быстрее GPU.
- TPU – сам по себе общий термин, но чаще всего подразумевается конкретная реализация от Google.
- TPU специально проектировались, как аппаратные ускорители для платформы Tensorflow.
- Проприетарная технология Google.
 - TPU доступен, как аппаратный ускоритель для Tensorflow при работе в Colab.
 - TPU доступен в виде Compute as a Service с REST API [\[11\]](#) в облаке Google.
 - Нет открытого TPU API, аналога CUDA или OpenCL.

Tensor Processing Unit (TPU). Use Cases



Нейропроцессоры, они же нейроморфные процессоры

- Даже TPU – это всё ещё аппаратный ускоритель для расчета модели нейрона, но не нейросетевая структура, реализованная непосредственно «в железе».
- Противоречие:
 - Обычно аппаратно реализуют «строительные кубики» той или иной степени абстракции.
 - Насколько оправдано аппаратно реализовывать всю нейронную сеть, решающую специфичную задачу?
 - Цикл разработки микросхемы от логического дизайна до фотолитографии может достигать 5 лет.
- Довольно давно предпринимаются попытки реализовать именно нейропроцессор / нейроморфный (в форме нейронов) процессор.
- Работы всё ещё на ранних исследовательских стадиях, готовых продуктов нет.

Нейропроцессоры. Ключевые слова

- Ключевые слова:
 - Neural Processor Unit (NPU)
 - Neural Network Processor (NNP)
 - Intelligence Processing Unit (IPU)
 - Vision Processing Unit (VPU)
 - Graph Processing Unit (GPU) (не тот же самый GPU 😊)

Нейроморфный процессор Алтай (AltAI)

- Разработчик – Мотив Нейроморфные Технологии. [\[12\]](#) [\[13\]](#)
- Физически воссозданы нейроны и синапсы.



Сравнение

- CPU

- + Можно подключить большое количество RAM. Может быть хорошим вариантом для моделей с большим количеством данных и относительно небольшой сложностью обучения.
- - Низкая параллельность.

- GPU

- + Высокая SIMD параллельность.
- - Часто требуется, чтобы модель уместилась в GRAM, а её меньше, чем RAM.

- TPU

- + Ещё большая SIMD параллельность, «GPU на стероидах». 😊
- - Подробности аппаратной архитектуры часто скрыты. Не понятно, с какими характеристиками модели имеет смысл поэкспериментировать.

Программно-аппаратный стек

- На протяжении курса будем периодически возвращаться к этой картине.

ML Model

Frameworks

CUDA API

OpenCL

ROCm API

SynapseAI

NeuroMatrix
API

CUDA Capability

CPU
(x86_64: SSE, AVX)
(ARM: NEON, NVDLA)

NVIDIA GPU

AMD GPU

Intel Habana
Gaudi

Модуль
SoC
(CPU + TPU)

Ссылки (1)

1) Архитектура фон Неймана

- <https://inf1.info/machineneumann>

2) Классификация Флинна

- <https://sites.google.com/site/exemsenko/4-klassifikacia-vycislitelnyh-setej-parallelnoj-obrabotki-sisd-simd-mimd-misd-konvejery-kes-pamat>

3) NVIDIA CUDA

- <https://developer.nvidia.com/cuda-toolkit>

4) OpenCL

- <https://www.khronos.org/opencl/>

5) AMD ROCm

- <https://rocmdocs.amd.com/en/latest/>

Ссылки (2)

6) SynapseAI Software

- https://docs.habana.ai/en/latest/Gaudi_Overview/SynapseAI_Software_Suite.html

7) Neuromatrix Deep Learning SDK

- <https://www.module.ru/directions/iskusstvennyj-intellekt/neuromatrix-deep-learning>

8) Tenstorrent

- <https://tenstorrent.com/grayskull/>

9) Graphcore

- <https://www.graphcore.ai/products/poplar>

10) Graphcore Poplar SDK

- <https://www.graphcore.ai/>

Ссылки (3)

11) Google Cloud TPU API

- <https://cloud.google.com/tpu/docs/reference/rest>

12) Нейроморфный процессор «AltAI»

- <https://motivnt.ru/neurochip-altai/>

13) Нейроморфный процессор «AltAI». Выступление Валерия Канглера

- <https://www.youtube.com/watch?v=GpdAzK3rRvw>

Резюме

- Рассмотрели вычислительную модель нейрона.
- Рассмотрели её вычисление на CPU.
- Рассмотрели, как SIMD подход позволяет добавить производительности. Рассмотрели GPU и TPU.
- Рассмотрели нейропроцессоры, их перспективность и проблемы.

Вопросы для самоконтроля

- В чём сложность выполнения вычислительной модели нейрона?
- За счёт чего достигается прирост производительности при SIMD подходе?
- В чём принципиальное отличие нейропроцессоров?
- Какое противоречие есть у этого подхода?

Спасибо!

