Efficient models: Challenges and trends

Egor Shvetsov

Today's plan:

1. Motivation type one:

- Which model is more suitable for a taks?
- Which data is more suitable for a task?
- How long does it take to find the best solution?
- An example with SeqNAS

2. Motivation type two computational effectiveness

- LLM
- Low resource devices
- Trends

3. How do we measure computational effectiveness?

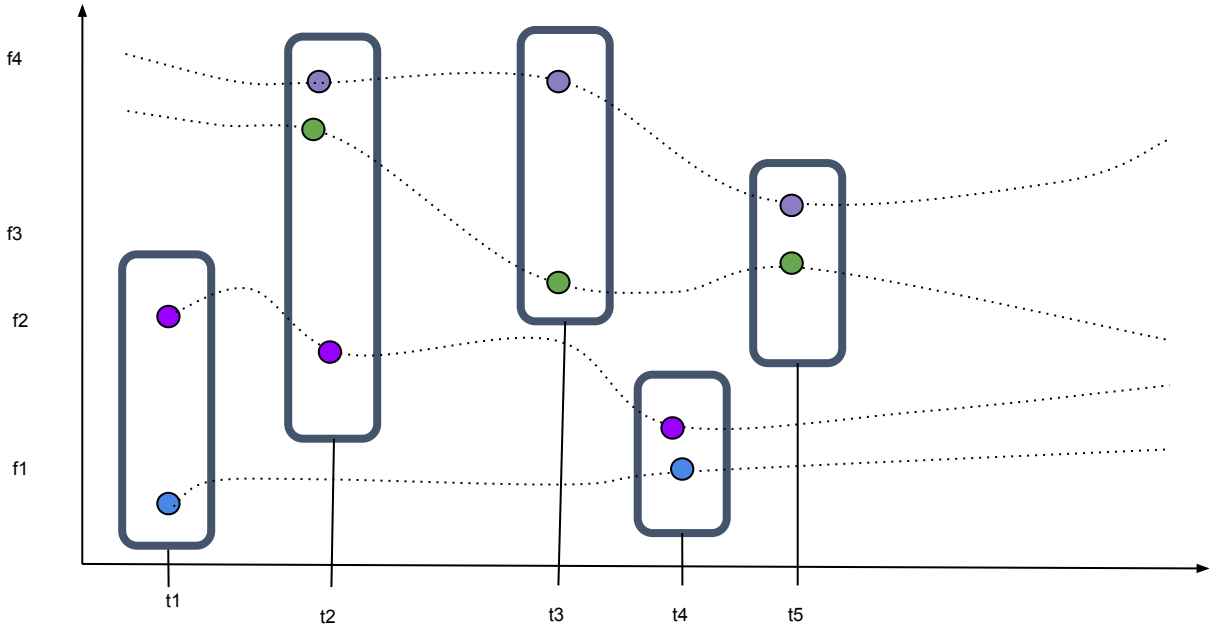
4. Hardware Software and inbetween
- Hardware review
- CPU vs GPU
- How they are connected
    - OpenVINO
    - ONNX
    - TensorRT

5. Course itinerary

>>> Motivation type one

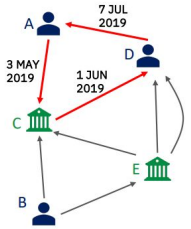
Dataset Example - How to model it?

- f4 Amount
- f3 Product category
- f2 Geo location
- f1 Currency



Dataset Example - How to model it?

Trans. ID	Timestamp	Source bank ID	Source Account	Target bank ID	Target Account	Amount	Currency	Payment type
0	3 MAY 2019 12:45	1	A	1	C	1400	USD	Cheque
1	15 MAY 2019 07:34	2	B	1	C	710	EUR	ACH
2	18 MAY 2019 16:55	3	E	1	C	950	USD	Credit card
3	1 JUN 2019 10:06	1	C	3	D	1200	CHF	Wire
4	27 JUN 2019 13:18	3	E	3	D	2300	EUR	Credit card
5	7 JUL 2019 11:14	3	D	1	A	1100	USD	Credit card
6	14 JUL 2019 09:37	2	B	3	E	650	USD	ACH
7	20 JUL 2019 14:02	3	E	3	D	2500	USD	Wire



Do we model it as:

- Sequences
- Tabular data
- Graph

Does it depend on a problem?

- f4 Amount
- f3 Product category
- f2 Geo location
- f1 Currency

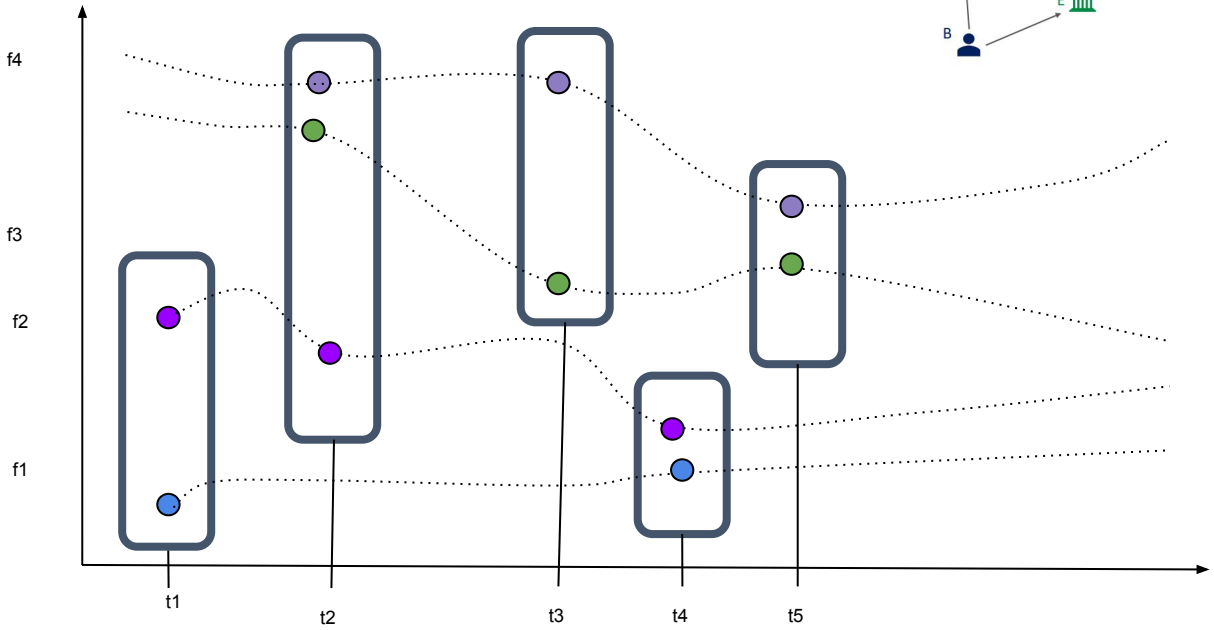


Image source: [Realistic Synthetic Financial Transactions for Anti-Money Laundering Models]

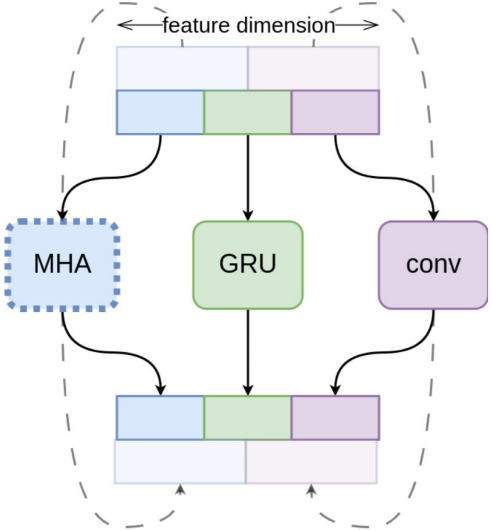
Real life example:

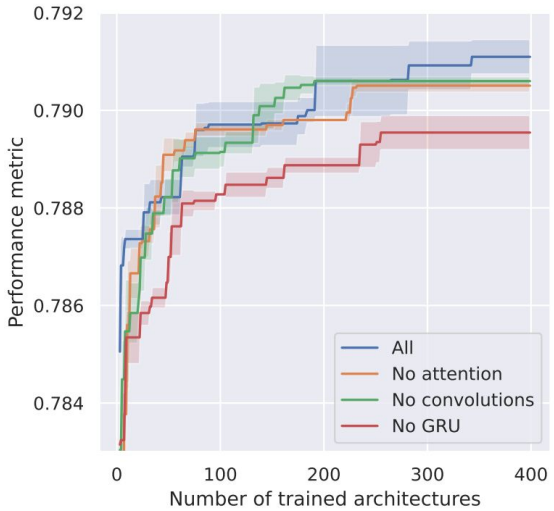
Encoder has both a searchable number of layers and the operations within each layer. Input of each Encoder layer

is divided into one to three blocks along the feature dimension, which can be processed using one of six potential

operations, such as MHA, GRU, or Convolution. The number of heads in MHA is searchable and is chosen from the set {1, 2, 4,8}. In total it provides up to 19 variations for a single Encoder layer. It is worth noting that each layer has a distinct set of operations. The outputs from each block are concatenated and sent to the next layer within the

Encoder. This structure is illustrated in Figure 5. The encoder may have three possible values for the number of layers (1, 2, or 4), resulting in approximately 130×10^3 possible Encoder variations. [1]





[1] SeqNAS: Neural Architecture Search for Event Sequence Classification

Example:

Top-performing banks can potentially save between $1-2$ million USD per year for every 0.5% increase of classification performance, as measured by AUC, in some applied problems, involving classification of users' transactions.

Developing a new model and implementing it into a bank's system could cost approximately $100,000 to $150,000 over a 6-month project period, with half of the effort usually dedicated to model development. [1]

How to find such models?
What efficiency depends on?

Example:

Top-performing banks can potentially save between $1-2$ million USD per year for every 0.5% increase of classification performance, as measured by AUC, in some applied problems, involving classification of users' transactions.

Developing a new model and implementing it into a bank's system could cost approximately $100,000 to $150,000 over a 6-month project period, with half of the effort usually dedicated to model development. [1]

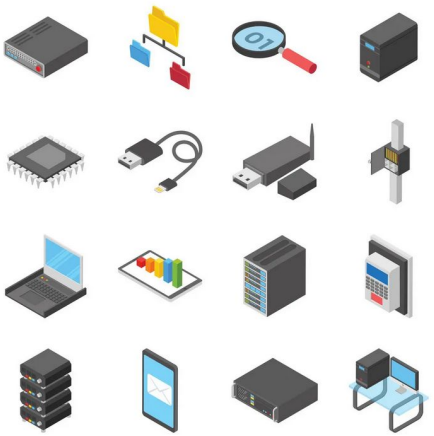
How to find such models?
What efficiency depends on?

- Data pre-processing and transformations
- Expert domain knowledge
- Hyper-parameter optimization
- Architecture design
- Else ... (Your ideas) ?

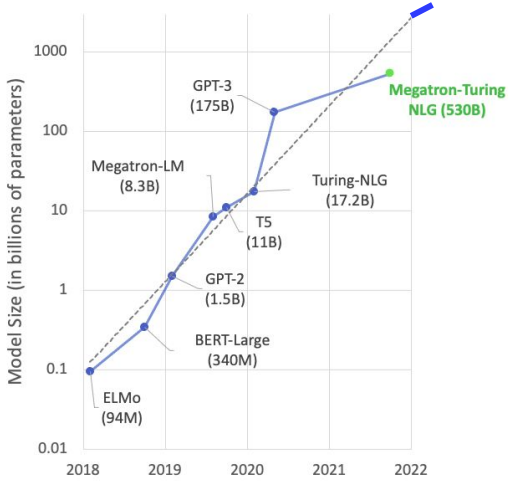
But search space might be too big?

>>> Motivation type two
Computation

Edge devices
Autonomous devices
Robotics



Modern LLMs



Trends

- LLM
- LLM based agents
- Multimodal models
- Scientific

applications

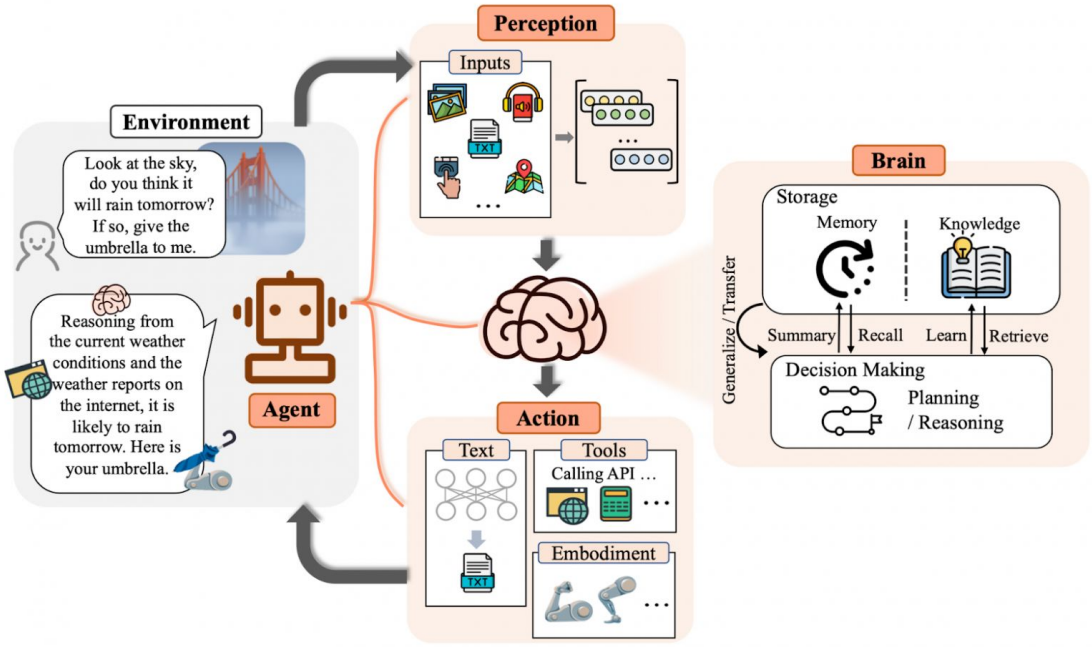


Image source: [Кто такие LLM-агенты и что они умеют?]

Trends

- LLM
- LLM based agents
- Multimodal models
- Scientific applications

Агенты - автономно действуют в некоторой среде без детального участия пользователя.

Agen - Tool

Агенты взаимодействуют со средой, среда здесь это различные инструменты, например, среда разработки кода или рекламаня платформа или, например экономические условия.

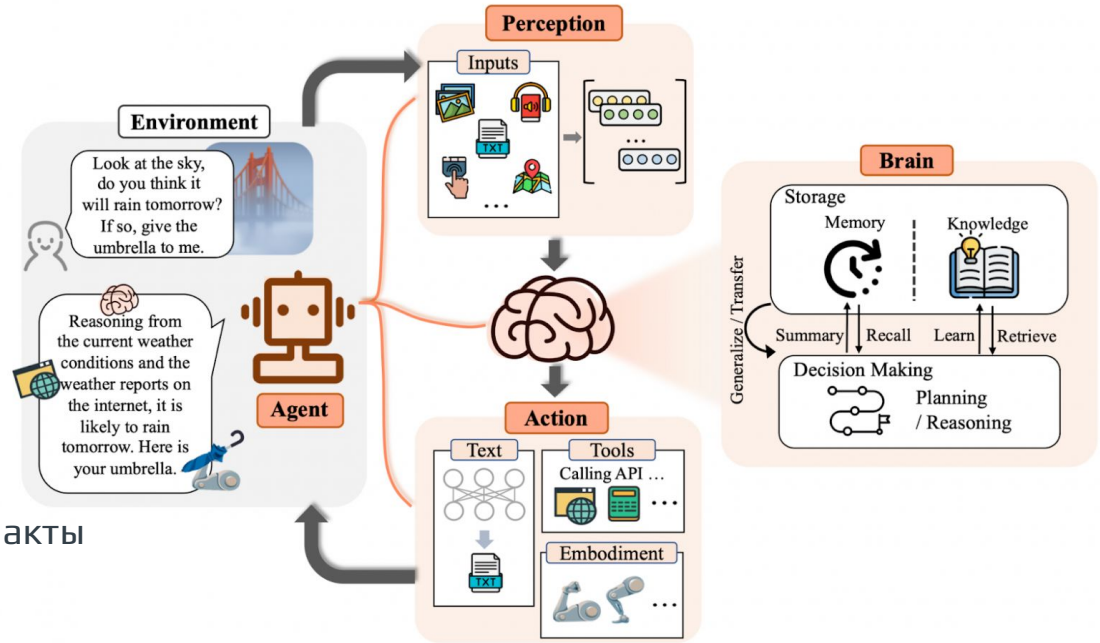
Агент может создавать рекламные компании и анализировать их эффективность.

Agent - Agent

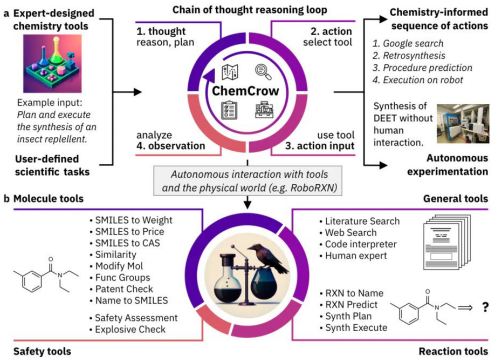
Системы симуляции взаимодействия агентов друг с другом.

Тренды

- LLM
- LLM агенты
- Мультимодальные модели
- Научные применения
- Цифровые двойники
- Специализированные LLM
- Оптимизация моделей
- Безопасность
- Нормативно-правовые акты



Тренды



Примеры Агентов

[1] LLM с личными предпочтениями и качествами для изучения экономического поведения человека в смоделированных сценариях. [ссылка]

[2] Система генеративных агентов, которые имитируют повседневную жизнь человека в виртуальном городе. [ссылка]

[3] Имитация взаимодействия судей между собой, для анализа и моделирования принятия решений в суде. [ссылка]

[4] Агенты для анализа и поиска информации, создание выжимок и различных источников, выделение основных сущностей в тексте. [ссылка]

[5] ChemCrow - агент LLM, который использует базы данных, связанные с химией, для автономного планирования и выполнения синтеза репеллента от насекомых, трех органокатализаторов, а также управляемого открытия нового хромофора. [ссылка]

[6] Математические агенты помогают исследователям открывать, решать и доказывать математические задачи. [ссылка]

[7] EduChat и CodeHelp — два других ярких примера агентов LLM, предназначенных для образования. [ссылка] [ссылка]

[8] Инструмент для взаимодействия архитекторов с агентами. [ссылка]

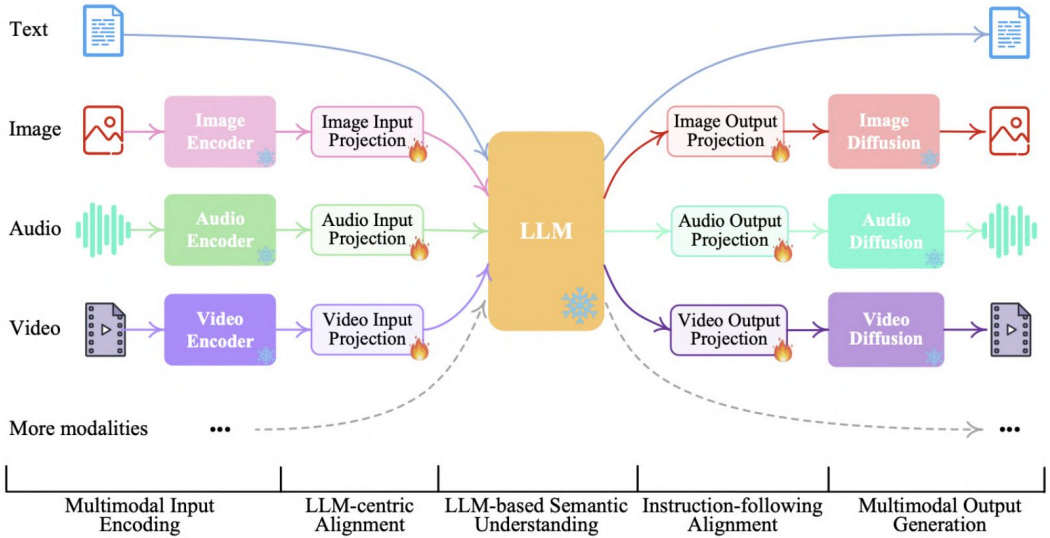
[9] ChatDev, ToolLLM, MetaGPT — примеры того, как агенты ИИ демонстрируют потенциал для автоматизации написания, отладки и тестирования кода, а также помощи в выполнении других задач по разработке программного обеспечения. [ссылка] [ссылка] [ссылка]

[10] D-Bot — администратор баз данных, который постоянно обучается и предоставляет советы по диагностике и оптимизации баз данных. [ссылка]

[11] OS-Copilot — платформа для создания универсальных агентов, способных взаимодействовать со сложными элементами операционной системы (ОС), включая интернет, кодовые терминалы, файлы, мультимедиа и различные сторонние приложения. [ссылка]

Тренды

- LLM
- LLM агенты
- Мультимодальные модели
- Научные применения
- Цифровые двойники
- Специализированные LLM
- Оптимизация моделей
- Безопасность
- Нормативно-правовые акты



Тренды

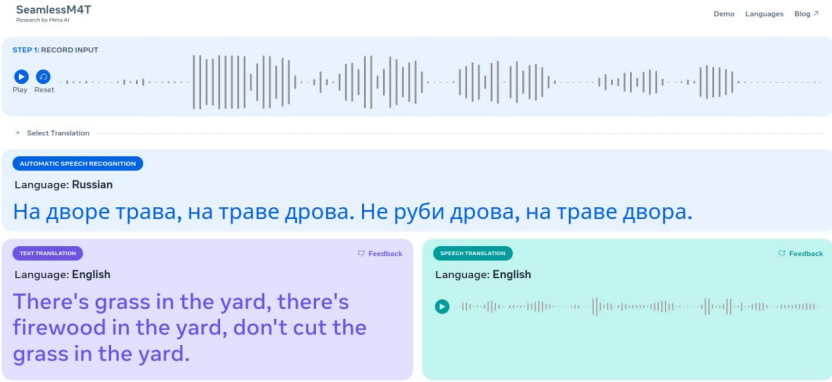
- LLM
- LLM агенты
- Мультимодальные модели
- Научные применения
- Цифровые двойники
- Специализированные LLM
- Оптимизация моделей
- Безопасность
- Нормативно-правовые акты

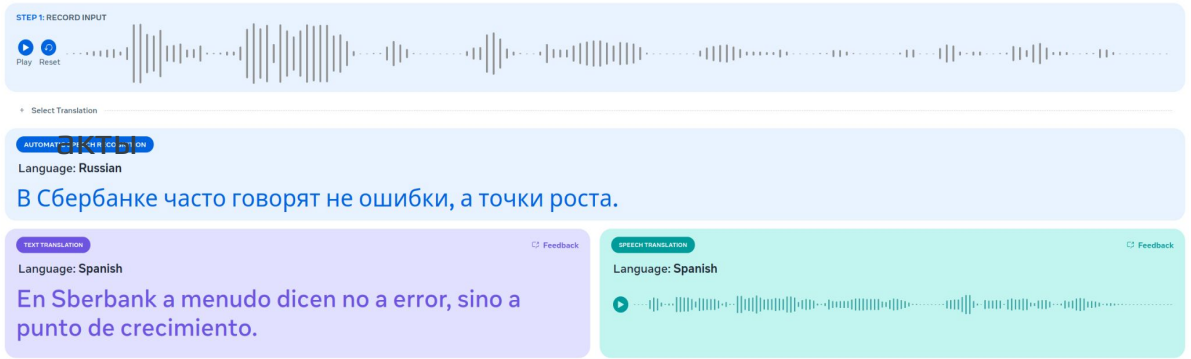
Prompt: A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and reflective, creating a mirror effect of the colorful lights. Many pedestrians walk about.

Промпт: стильная женщина идет по улице Токио, наполненной теплым светящимся неоном и анимированными вывесками города. Она носит черную кожаную куртку, длинное красное платье и черные ботинки, а также носит черную сумочку. Она носит солнцезащитные очки и красную помаду. Она ходит уверенно и непринужденно. Улица влажная и отражающая свет, создавая зеркальный эффект разноцветных огней. Много пешеходов идут.

Тренды

- LLM
- LLM агенты
- Мультимодальные модели
- Научные применения
- Цифровые двойники
- Специализированные LLM
- Оптимизация моделей
- Безопасность
- Нормативно-правовые

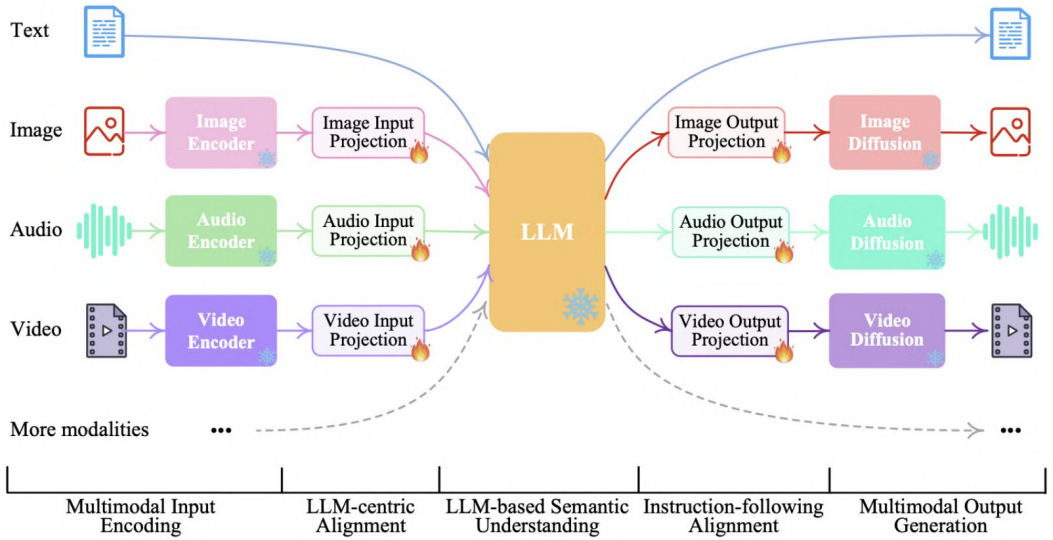




Trends

- LLM
- LLM based agents
- Multimodal models
- Scientific

applications

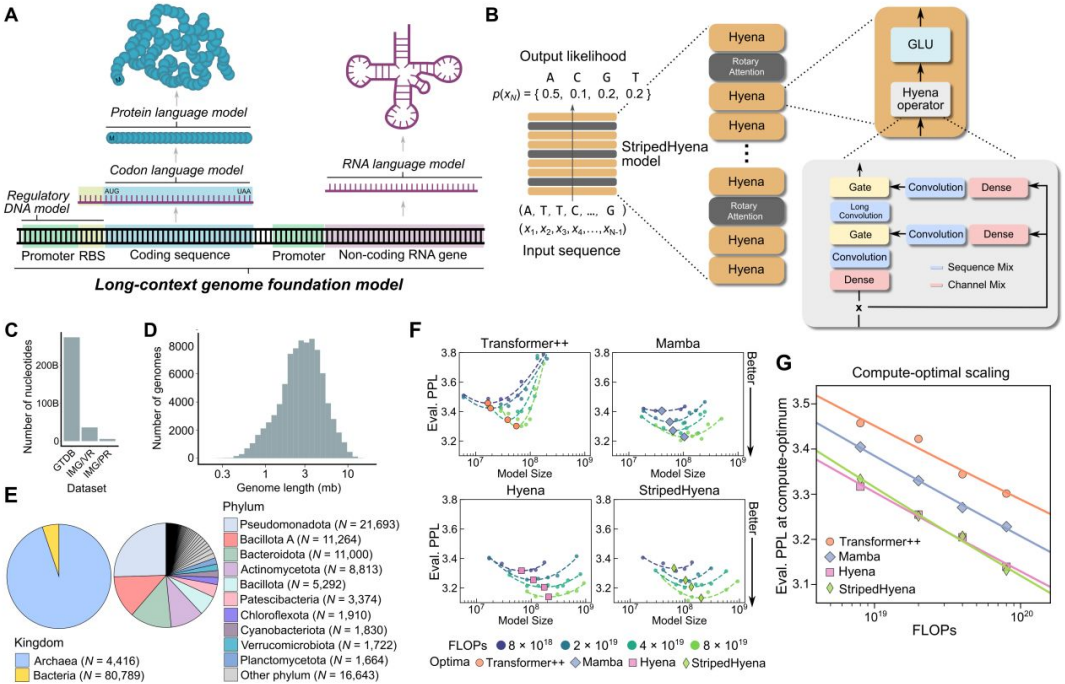


Trends

- LLM
- LLM based agents
- Multimodal models
- Scientific

applicat

Max sequence lengths: 131 000



paper: [Sequence modeling and design from molecular to genome scale with Evo]

Computational efficiency & Money:

- Using NVIDIA A100 GPUs ChatGPT is capable of outputting around 15-20 words per second;
- Given ChatGPT's version 3.5 has over 175 billion parameters, to get an output for a single query, it needs at least five A100 GPUs;
- Running this on AWS with the above suggestion would cost $25/hr;

Q1: What the cost of one query should be to break even assuming uniform requests in time?

Q2: What if we improve latency 10%? How much would we save over one day?

Q3: What if we improve memory 10%? How much would we save over one day?

Computational efficiency & Money:

- Using NVIDIA A100 GPUs ChatGPT is capable of outputting around 15-20 words per second;
- Given ChatGPT's version 3.5 has over 175 billion parameters, to get an output for a single query, it needs at least five A100 GPUs;
- Running this on AWS with the above suggestion would cost $25/hr;

Q1: What the cost of one query should be to break even assuming uniform requests in time?

Q2: What if we improve latency 10%? How much would we save over one day?

Q3: What if we improve memory 10%? How much would we save over one day?

A1: ?
A2: We would serve about 8640 requests more!
A2: ?

How do we measure computational efficiency?

- Memory
  FLOPs
- Latency
  Energy

Other examples ... ?

- Memory
  FLOPs
- Latency
  Energy

FLOPs - number of floating point operations?

How much is it for a matrix multiplication?

$$F[(n \times p) \times (p \times m)] = ?$$

- Memory
- FLOPs
- Latency
  Energy

FLOPs - number of floating point operations?

How much is it for a matrix multiplication?

$$F[(n \times p) \times (p \times m)] = 2nmp$$

- Memory
- FLOPs
- Latency
  Energy

What limits the latency?

Your ideas?

- Memory
  FLOPs
- Latency
  Energy

In general, if a computation re-uses data, it will require less memory bandwidth. Re-use can be accomplished by:

- sending more inputs to be processed by the same weights
- sending more weights to process the same inputs

If there is no input or weight data re-use, then the **bandwidth** is at a maximum for a given application.

Some examples:

Linear layers: here a weight matrix of M by M is used to process a vector of M values with b bits. Total data transferred is: $b(M+M^2)$ or $\sim bM^2$

If the linear layer is used only for one vector, it will require to send the entire M^2 matrix of weights as computation occurs.

So, is it about Memory or Number of parameters?

- Memory
- FLOPs
- Latency
  Energy

In general, if a computation re-uses data, it will require less memory bandwidth. Re-use can be accomplished by:

- sending more inputs to be processed by the same weights
- sending more weights to process the same inputs

If there is no input or weight data re-use, then the **bandwidth** is at a maximum for a given application.

Some examples:

Linear layers: here a weight matrix of M by M is used to process a vector of M values with b bits. Total data transferred is: $b(M+M^2)$ or $\sim bM^2$

If the linear layer is used only for one vector, it will require to send the entire M^2 matrix of weights as computation occurs.

So, is it about Memory or Number of parameters?

- **It is different for inference and training**

- Memory
  FLOPs
- Latency
  Energy

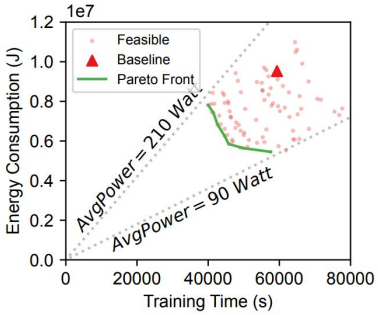
Input Type	Accumulation Type	Relative math throughput	Bandwidth
FP16	FP16	8x	2x
INT8	INT32	16x	4x
INT4	INT32	32x	8x
INT1	INT32	128x	32x

*Relative to FP32

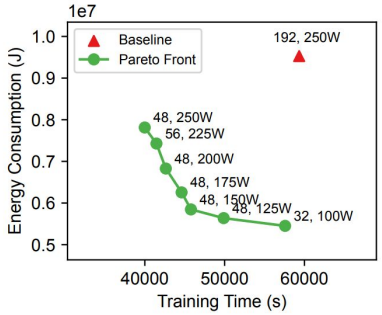
- Memory
  FLOPs
- Latency
- Energy

Some facts:

- AlphaGo: 1920 CPUs and 280 GPUs, $3000 electric bill per game
- At extreme scales, training the GPT-3 model just once consumes 1,287 MWh, which is enough to supply an average US household for 120 years.



(a) *Energy-Time Tradeoff*



(b) *Pareto Front Zoom-in*

paper: [Zeus: Understanding and Optimizing GPU Energy Consumption of DNN Training]

model	gpu	task	energy	throughput	response_length	latency	arc	hellaswag	truthfulqa	parameters
MetaAI/Llama-7B	A100	chat	335.26	27.47	60.65	1.99	51.11	77.74	34.08	7
MetaAI/Llama-13B	A100	chat	631.57	23.2	76.47	2.97	56.31	80.86	39.9	13
tatsu-lab/alpaca-7B	A100	chat	684.17	28	132.85	4.63	52.65	76.91	39.55	7
RWKV/rwkv-raven-7b	A100	chat	735.77	61.52	214.78	3.08	39.42	66.45	38.54	7
Neutralzz/BiLLa-7B-SFT	A100	chat	881.44	33.57	161.81	4.79	27.73	26.04	49.05	7
H2OAI/H2OGPT-oasst1-7B	A100	chat	951.08	33.2	212.46	6.39	36.86	61.55	37.94	7
FreedomIntelligence/phoenix-inst-chat-7b	A100	chat	1030.89	55.84	240.34	4.12	44.97	63.22	47.08	7
StabilityAI/stablelm-tuned-alpha-7b	A100	chat	1124.4	45.42	243.88	5.21	31.91	53.59	40.22	7
databricks/dolly-v2-12B	A100	chat	1242.92	22.11	153.76	7.94	42.15	71.83	33.37	12
Salesforce/xgen-7b-8k-inst	A100	chat	1246.09	49.54	275.27	5.6	46.67	74.85	41.89	7
BAIR/koala-7b	A100	chat	1345.55	26.56	261.07	9.62	47.1	73.7	46	7
togethercomputer/RedPajama-INCITE-7B-Chat	A100	chat	1457.86	27.49	274.39	9.54	42.15	70.84	36.1	7
LMSys/vicuna-7B	A100	chat	1531.7	27.26	284.92	10.3	53.5	77.53	49	7
project-baize/baize-v2-7B	A100	chat	1588.27	31.59	326.3	10.17	48.46	75	41.66	7
LMSys/fastchat-t5-3b-v1.0	A100	chat	1802.98	19.29	314.3	20.68	35.92	46.36	48.79	3
nomic-ai/gpt4all-13b-snoozy	A100	chat	2004.73	26.57	247.8	9.28	56.06	78.69	48.36	13
OpenAssistant/oasst-sft-1-pythia-12b	A100	chat	2060.03	25.4	259.59	9.93	45.56	69.93	39.19	12
BAIR/koala-13b	A100	chat	2144.83	21.21	265.57	12.14	52.9	77.54	50.09	13
openaccess-ai-collective/manticore-13b-ch	A100	chat	2189.25	26.28	288.82	10.95	58.7	81.96	48.86	13
Camel-AI/CAMEL-13B-Combined-Data	A100	chat	2526.46	24.5	291.92	13.17	55.55	79.3	47.33	13
LMSys/vicuna-13B	A100	chat	2600.84	21.61	280.74	12.74	52.9	80.12	51.82	13

How much energy consumes LLama-7B in terms of 60W a light bulb:

- 3 seconds of work
- 5 seconds of work
- 60 seconds on work

source: ml.energy

model	gpu	task	energy	throughput	response_length	latency	arc	hellaswag	truthfulqa	parameters
MetaAI/Llama-7B	A100	chat	335.26	27.47	60.65	1.99	51.11	77.74	34.08	7
MetaAI/Llama-13B	A100	chat	631.57	23.2	76.47	2.97	56.31	80.86	39.9	13
tatsu-lab/alpaca-7B	A100	chat	684.17	28	132.85	4.63	52.65	76.91	39.55	7
RWKV/rwkv-raven-7b	A100	chat	735.77	61.52	214.78	3.08	39.42	66.45	38.54	7
Neutralzz/BiLLa-7B-SFT	A100	chat	881.44	33.57	161.81	4.79	27.73	26.04	49.05	7
H2OAI/H2OGPT-oasst1-7B	A100	chat	951.08	33.2	212.46	6.39	36.86	61.55	37.94	7
FreedomIntelligence/phoenix-inst-chat-7b	A100	chat	1030.89	55.84	240.34	4.12	44.97	63.22	47.08	7
StabilityAI/stablelm-tuned-alpha-7b	A100	chat	1124.4	45.42	243.88	5.21	31.91	53.59	40.22	7
databricks/dolly-v2-12B	A100	chat	1242.92	22.11	153.76	7.94	42.15	71.83	33.37	12
Salesforce/xgen-7b-8k-inst	A100	chat	1246.09	49.54	275.27	5.6	46.67	74.85	41.89	7
BAIR/koala-7b	A100	chat	1345.55	26.56	261.07	9.62	47.1	73.7	46	7
togethercomputer/RedPajama-INCITE-7B-Chat	A100	chat	1457.86	27.49	274.39	9.54	42.15	70.84	36.1	7
LMSys/vicuna-7B	A100	chat	1531.7	27.26	284.92	10.3	53.5	77.53	49	7
project-baize/baize-v2-7B	A100	chat	1588.27	31.59	326.3	10.17	48.46	75	41.66	7
LMSys/fastchat-t5-3b-v1.0	A100	chat	1802.98	19.29	314.3	20.68	35.92	46.36	48.79	3
nomic-ai/gpt4all-13b-snoozy	A100	chat	2004.73	26.57	247.8	9.28	56.06	78.69	48.36	13
OpenAssistant/oasst-sft-1-pythia-12b	A100	chat	2060.03	25.4	259.59	9.93	45.56	69.93	39.19	12
BAIR/koala-13b	A100	chat	2144.83	21.21	265.57	12.14	52.9	77.54	50.09	13
openaccess-ai-collective/manticore-13b-ch	A100	chat	2189.25	26.28	288.82	10.95	58.7	81.96	48.86	13
Camel-AI/CAMEL-13B-Combined-Data	A100	chat	2526.46	24.5	291.92	13.17	55.55	79.3	47.33	13
LMSys/vicuna-13B	A100	chat	2600.84	21.61	280.74	12.74	52.9	80.12	51.82	13

How much energy consumes LLama-7B in terms of 60W a light bulb:

- 3 seconds of work
- 5 seconds of work
- 60 seconds on work

Brain consumes 20 joules energy per second.

source: ml.energy

- Memory
- FLOPs
- Latency
- Energy

Arithmetic operation	ADD	MUL
8-bit Integer	0.03 pJ	0.2 pJ
16-bit Floating point	0.4 pJ	1.1 pJ
32-bit Integer	0.1 pJ	3.1 pJ
32-bit Floating point	0.9 pJ	3.7 pJ

Memory size	64-bit Memory access
8KB	10 pJ
32KB	20 pJ
1 MB	100 pJ
DRAM	1.3-2.6 nJ

paper: Energy-Efficient Model Compression and Splitting for Collaborative Inference Over Time-Varying Channels

- Memory
- FLOPs
- Latency
- Energy

Memory limited layers:

- Normalization (scaling)
- Batch Normalization
- Activations
- Pooling
- Else ... (Your ideas) ?

source: memory-Limited Layers

- Memory
- FLOPs
- Latency
- Energy

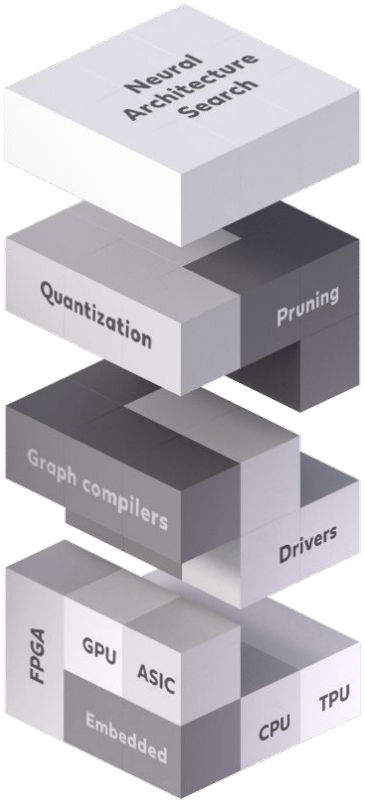
How to understand if an operation is memory of arithmetic bound?

$$Arithmetic \quad Intensity = \frac{FLOPs}{byte \quad access} = \frac{2npm}{2(np+pm+np)}$$

Let's consider a M x N x K = 8192 x 128 x 8192 GEMM. For this specific case, the arithmetic intensity is 124.1 FLOPS/B, lower than V100's 138.9 FLOPS:B, thus this operation would be memory limited.

source: Nvidia Matrix Multiplication Background User's Guide

Software meets Hardware



HARDWARE	SOFTWARE	ALGORITHMIC
• GPU • TPU • CPU • FPGA • Ascend • etc.	• CUDA • Graph Compilers • MKL - DNN • etc.	• PRUNING • QUANTIZATION • NAS • etc.

A majority of machine learning architectures lack explicit consideration for software or hardware properties during development, resulting in computational inefficiency.

Furthermore, many models concentrate exclusively on achieving optimal results rather than accounting for practical implementation issues.

Image source: deci.ai

HARDWARE

- GPU
- TPU
- CPU
- FPGA
- Ascend

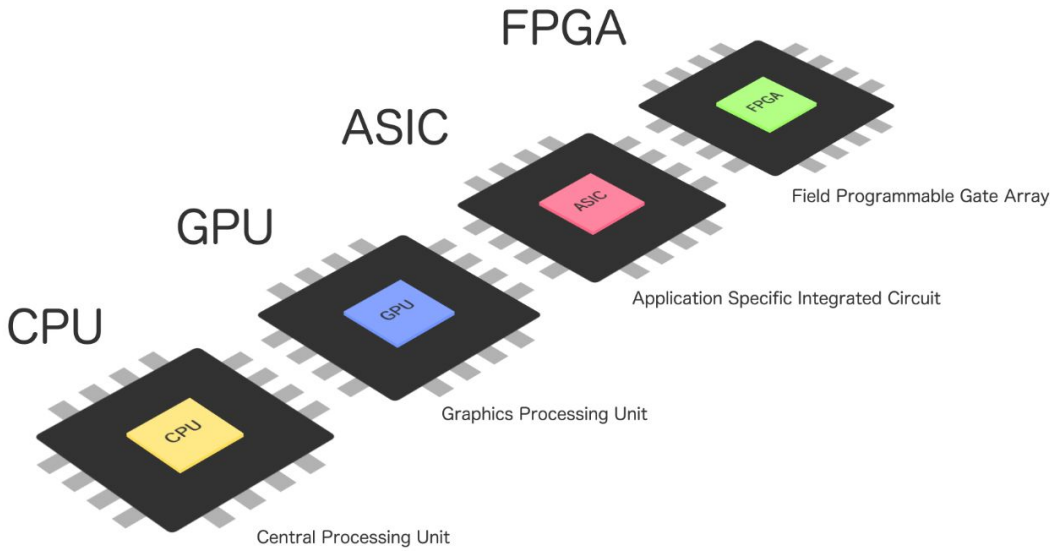
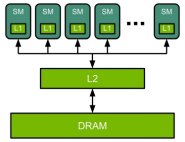


Image source: deci.ai

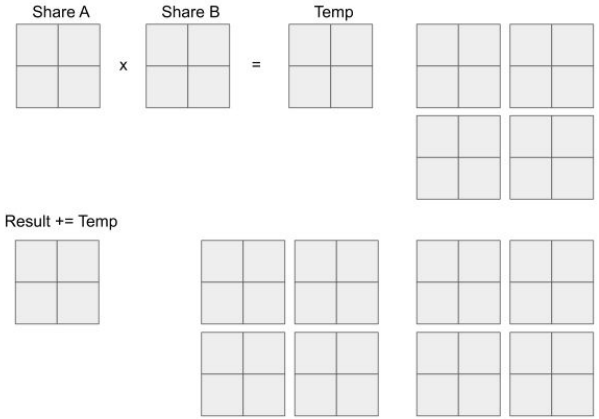
CPU	The central processing unit (CPU) is a general-purpose processor typically containing between 4 and 16 cores.
GPU	Graphics Processing Units (GPUs) are highly parallel cores (100 or 1000) for high-speed graphics rendering. They provide high-performance processing and typically take up more space and consume more power than CPUs. Because of their large number of small cores, GPUs are well suited for AI workloads, facilitating both neural network training and AI operation.
FPGA	Field Programmable Gate Array (FPGA), a technology that creates a chip with a set of logic elements, flip-flops, sometimes RAM and programmable electrical connections between them, consumes less power than CPUs and GPUs. Can be reprogrammed on site by engineers with programming experience.
ASICS	Application-specific integrated circuits (ASICs) are custom logic designed using manufacturer circuit libraries and offer the benefits of low power consumption, speed and small footprint. However, they are time-consuming to develop and more expensive than other options, so ASICs are recommended for products that will operate in very high volumes. ASICS Types: - Image processing units (VPUs), image and image processors, and coprocessors - Tensor processing units (TPUs), such as the first TPU developed by Google for its machine learning platform, TensorFlow. - Neural Compute Units (NCU), including from ARM
QPU	A quantum processing unit (QPU) is the central component of a quantum computer or quantum simulator. It is a physical or simulated processor that contains a series of interconnected qubits that can be manipulated to compute quantum algorithms. A QPU uses the behavior of particles such as electrons or photons to perform certain types of calculations much faster than the processors in modern computers. QPUs rely on behavior called superposition, the ability of a particle to be in many states at once, described in a relatively new branch of physics called quantum mechanics. In contrast, CPUs, GPUs, and DPUs apply the principles of classical physics to electrical currents. This is why modern systems are called classical computers.
Neuromorphic chips	Neuromorphic computing—brain-inspired computing—has emerged as a new technology that can process information at very low energy costs using electronic devices that mimic the electrical behavior of (biological) neural networks.
Photonic computing	Оптические вычисления или фотонные вычисления используют световые волны, создаваемые лазерами или некогерентными источниками, для обработки данных, хранения данных или передачи данных для вычислений. На протяжении десятилетий фотоны обещают обеспечить более высокую пропускную способность, чем электроны, используемые в обычных компьютерах (см. оптические волокна).

GPU vs CPU

CPU	GPU
CPUs are latency optimized	GPUs are bandwidth optimized
	 
- The CPU (racing car) can fetch some memory (packages) in your RAM quickly - pack new passengers - Racing car needs to go back in force - Can adjust to a new route relatively quickly, small capacity - ~ 50GB/s memory bandwidth - The CPU L1 cache only operates at about 5TB/s which is quite slow and has the size of roughly 1MB	- GPU (big truck) is slower in fetching memory (loading packages) - much higher latency - Big truck does not need to go back on force once so many times - Not very flexible in adjusting to a new road - need to wait long time to pack - ~ 750GB/s memory bandwidth - GPUs can have in total a lot of register memory, which is very small and thus very fast. - GPU registers size being more than 30 times larger compared to CPUs and still twice as fast which translates to up to 14MB register memory that operates at a whopping 80TB/s.

GPU vs CPU

A good way to understand the difference is to compare Matrix Multiplication on GPU and CPU with Tiling



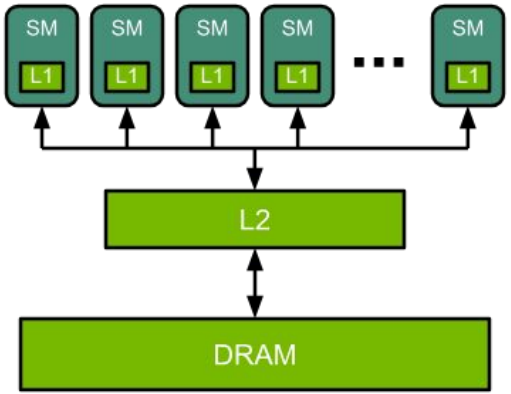


Image source: Tiled Matrix Multiplication
resources: Optimizing Matrix Multiplication: Unveiling the Power of Tiles

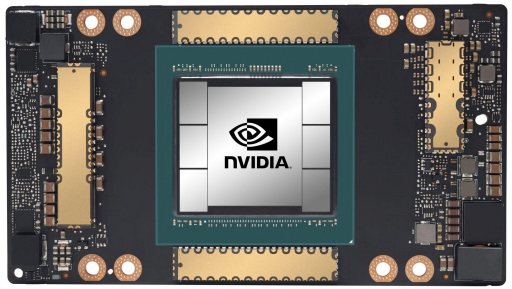
Support of low precision computations in GPU

- Float16
- Float32
- Bfloat16

Will be discussed in future series …

Supported by hardware and software for most modern GPU

Not supported by hardware or software



Supported by some hardware only

Problems with the current processors (CPU/GPU) are:

Energy efficiency:

AlphaGo, GPT and other examples

Architecture:

good for matrix multiplication (still the essence of DL) but not well-suitable for brain-like computations

source: Hardware for Deep Learning

Cerebras Systems

California-based Cerebras Systems has unveiled the Wafer Scale Engine (WSE-3), its latest artificial intelligence (AI) chip with a whopping four **trillion transistors**. It delivers twice the performance of its predecessor, the Cerebras WSE-2, which previously held the record for the fastest chip.

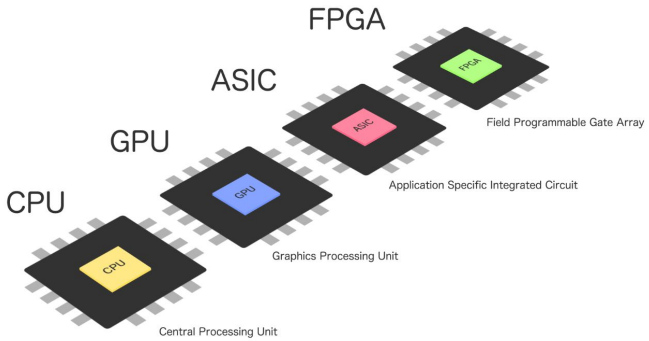
Systems built using WSE-3 will be able to accurately train models with 70 billion parameters in just one day.

The chip is almost size of iPad.

Fast but still has a huge energy consumption.

FPGA

- Amazon has FPGA F1 instances in the cloud
- Alibaba has FGPA F3 instances in the cloud
- Yandex uses FPGA for its own DL output
- Microsoft launched (in 2015) Project Catapult, which uses FPGA clusters
- Microsoft Project Brainwave: FPGA-Based AI Inference
- Microsoft Azure allows you to deploy pre-trained models on FPGA(!)
- Baidu has FPGA instances.



source: Hardware for Deep Learning

ASIC

An ASIC (Application Specific Integrated Circuit) is an integrated circuit that is customized for a specific use rather than intended for general use.

- Google has Tensor Processing Units (TPU v2/v3/v4) in the cloud
- Intel has acquired Habana, Mobileye, Movidius, Nervana and has processors for training and inference
- Graphcore has a second generation IPU
-
- AWS has its own features for training and inference with ASICS
- Alibaba Hanguang 800
- Huawei Ascend 310, 910
- Bitmain Sophon, Cerebras, Groq and many, many others
- Many ASICs are designed for multi-chip and supercomputer configurations!

source: Hardware for Deep Learning

ASIC/FPGA

Problems with FPGA/ASIC and edge devices

Energy efficiency:
- Better than CPU/GPU, but still far from 20 watts used by the human brain

Architecture:
- Even more specialized for ML/DL computations, but...

Still far from brain-like computations

source: Hardware for Deep Learning

Neuromorphic computing

Neuromorphic computing - brain-inspired computing - has emerged as a new technology to enable information processing at very low energy cost using electronic devices that emulate the electrical behaviour of (biological) neural networks.

- Usually uses spiking neural networks (SNN)
- Neuromorphic chips attempt to model in silicon the massively parallel way the brain processes information as billions of neurons and trillions of synapses respond to sensory inputs such as visual and auditory stimuli.
- DARPA SyNAPSE program (Systems of Neuromorphic Adaptive Plastic Scalable Electronics) Intel Loihi 2, IBM TrueNorth, Tianjic; Stanford Neurogrid; HRL neuromorphic chip; Human Brain Project BrainScaleS-2, SpiNNaker and HICANN.

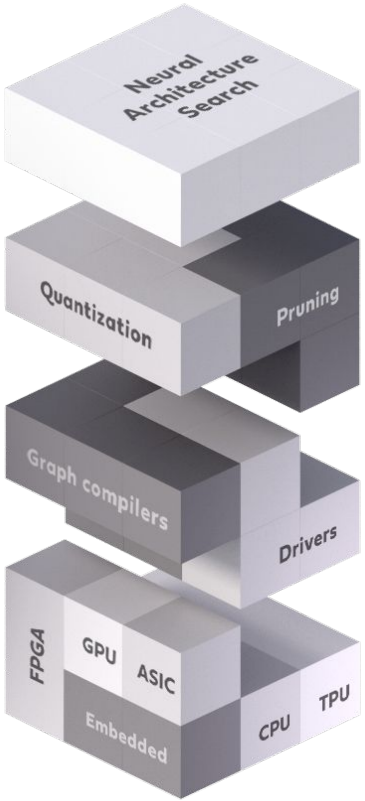
source: Hardware for Deep Learning

Other

- DNA computing
- Unconventional computing: cellular automata, reservoir computing, using biological cells/neurons, chemical computation, membrane computing, slime mold computing and much more

source: Hardware for Deep Learning

CPU	The central processing unit (CPU) is a general-purpose processor typically containing between 4 and 16 cores.
GPU	Graphics Processing Units (GPUs) are highly parallel cores (100 or 1000) for high-speed graphics rendering. They provide high-performance processing and typically take up more space and consume more power than CPUs. Because of their large number of small cores, GPUs are well suited for AI workloads, facilitating both neural network training and AI operation.
FPGA	Field Programmable Gate Array (FPGA), a technology that creates a chip with a set of logic elements, flip-flops, sometimes RAM and programmable electrical connections between them, consumes less power than CPUs and GPUs. Can be reprogrammed on site by engineers with programming experience.
ASICS	Application-specific integrated circuits (ASICs) are custom logic designed using manufacturer circuit libraries and offer the benefits of low power consumption, speed and small footprint. However, they are time-consuming to develop and more expensive than other options, so ASICs are recommended for products that will operate in very high volumes. ASICS Types: - Image processing units (VPUs), image and image processors, and coprocessors - Tensor processing units (TPUs), such as the first TPU developed by Google for its machine learning platform, TensorFlow. - Neural Compute Units (NCU), including from ARM
QPU	A quantum processing unit (QPU) is the central component of a quantum computer or quantum simulator. It is a physical or simulated processor that contains a series of interconnected qubits that can be manipulated to compute quantum algorithms. A QPU uses the behavior of particles such as electrons or photons to perform certain types of calculations much faster than the processors in modern computers. QPUs rely on behavior called superposition, the ability of a particle to be in many states at once, described in a relatively new branch of physics called quantum mechanics. In contrast, CPUs, GPUs, and DPUs apply the principles of classical physics to electrical currents. This is why modern systems are called classical computers.
Neuromorphic chips	Neuromorphic computing—brain-inspired computing—has emerged as a new technology that can process information at very low energy costs using electronic devices that mimic the electrical behavior of (biological) neural networks.
Photonic computing	Оптические вычисления или фотонные вычисления используют световые волны, создаваемые лазерами или некогерентными источниками, для обработки данных, хранения данных или передачи данных для вычислений. На протяжении десятилетий фотоны обещают обеспечить более высокую пропускную способность, чем электроны, используемые в обычных компьютерах (см. оптические волокна).



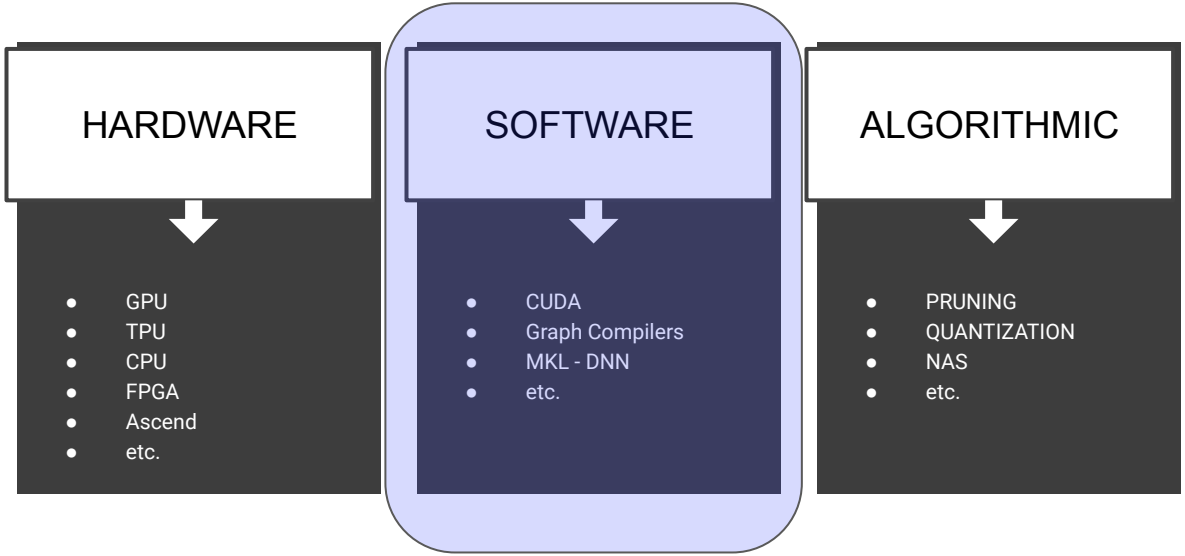
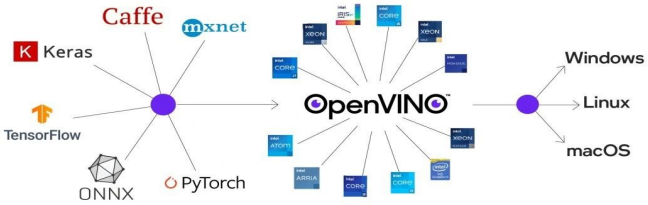
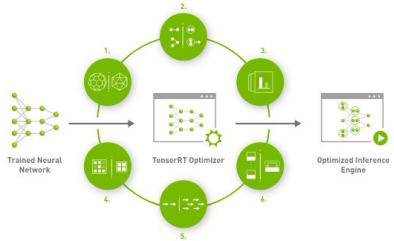
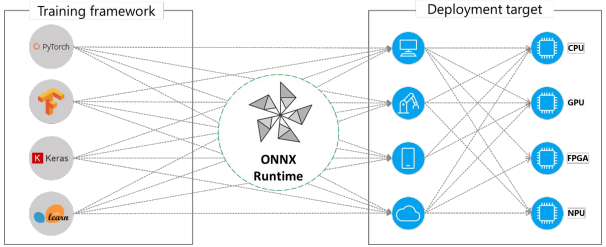


Image source: deci.ai



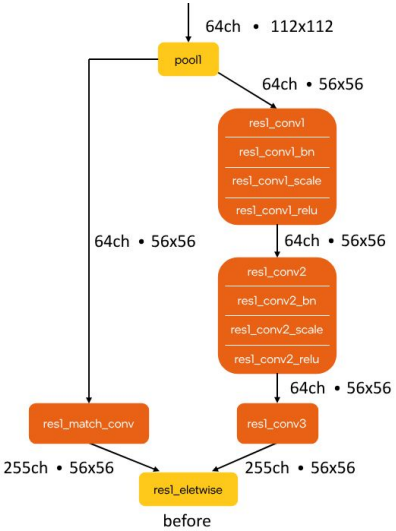


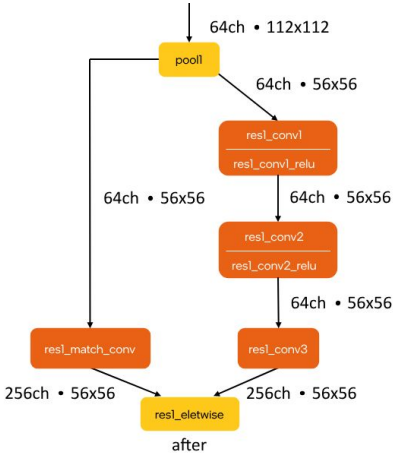
1. **Weight & Activation Precision Calibration**
   Maximizes throughput by quantizing models to INT8 while preserving accuracy
2. **Layer & Tensor Fusion**
   Optimizes use of GPU memory and bandwidth by fusing nodes in a kernel
3. **Kernel Auto-Tuning**
   Selects best data layers and algorithms based on target GPU platform
4. **Dynamic Tensor Memory**
   Minimizes memory footprint and re-uses memory for tensors efficiently
5. **Multi-Stream Execution**
   Scalable design to process multiple input streams in parallel
6. **Time Fusion**
   Optimizes recurrent neural networks over time steps with dynamically generated kernels

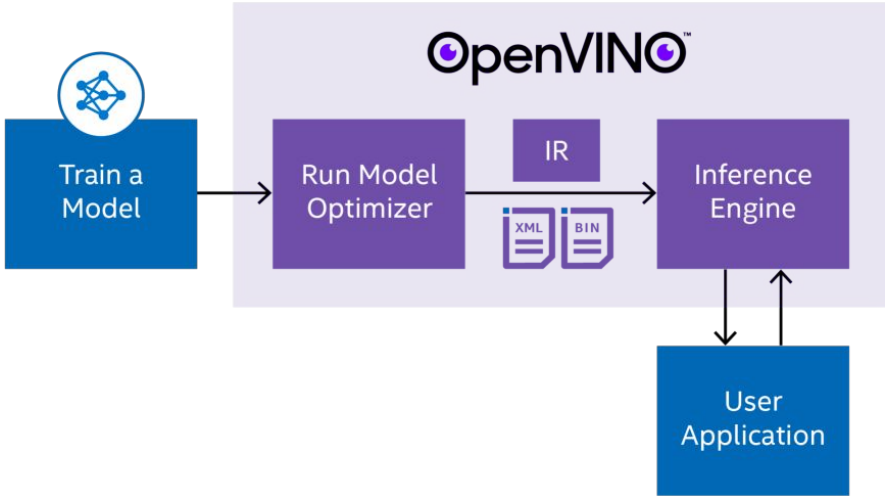


ONNX: This tool provides a standardized format for deep learning models, enabling them to be easily shared across different frameworks and platforms. In our project, ONNX used to ensure that our ResNet-50 model can be seamlessly transitioned between various environments.

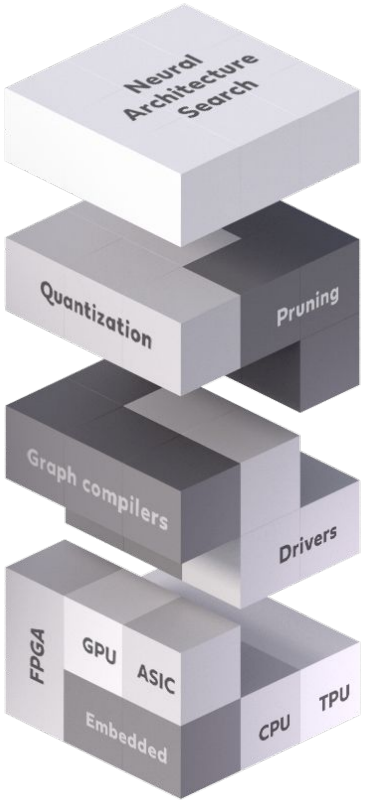
ONNX Runtime	Open source originally created by Microsoft and Facebook	Can be used as high level interface for TensorRT and OpenVino
TensorRT	Nvidia	This tool is specifically designed for NVIDIA GPUs, focusing on maximizing throughput and efficiency. It optimizes neural network models by fusing layers, selecting the most efficient data formats, and leveraging reduced precision (FP16) arithmetic where possible.
OpenVino	Intel	Developed by Intel, OpenVINO specializes in optimizing deep learning models for Intel hardware, particularly CPUs. It is a crucial tool for enhancing the performance of models where GPU resources are not available or limited.

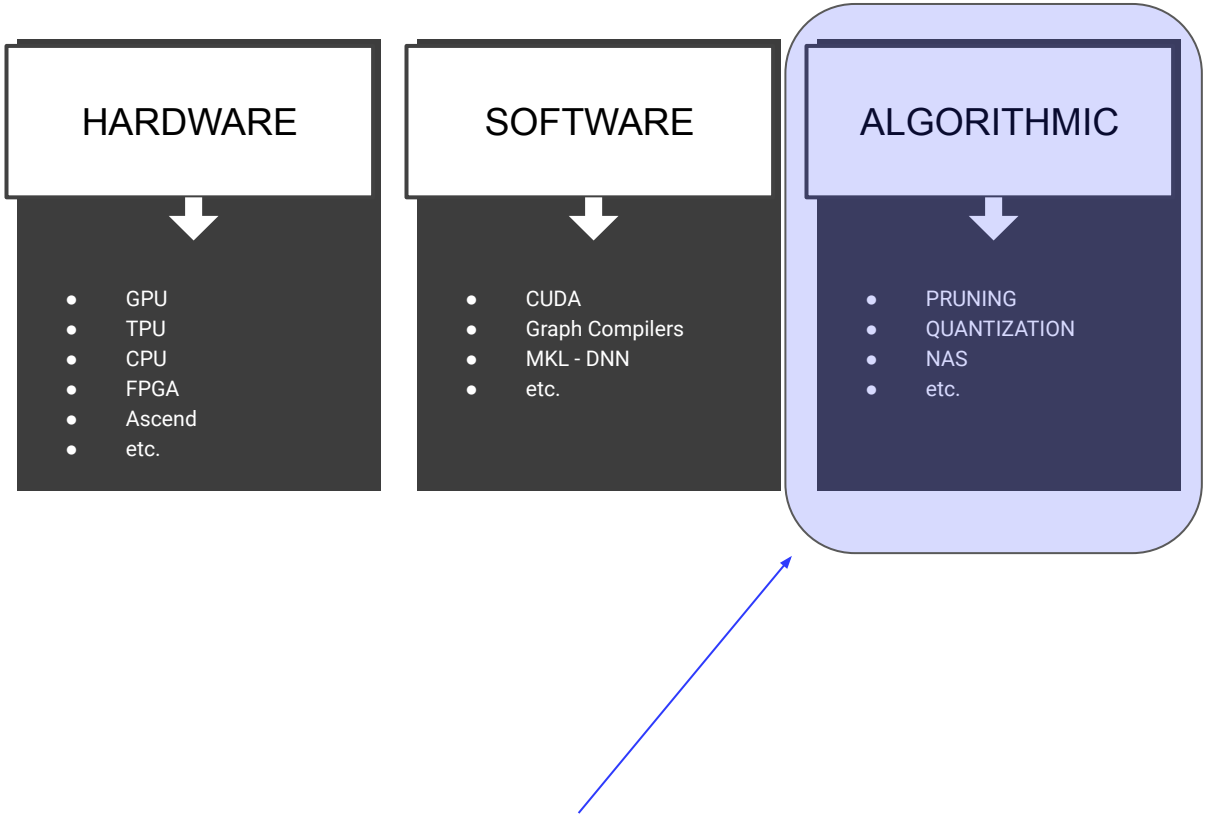






To use quantization, some types of specification and various data formats we need specialized frameworks or we need to write low level solutions ourselves.

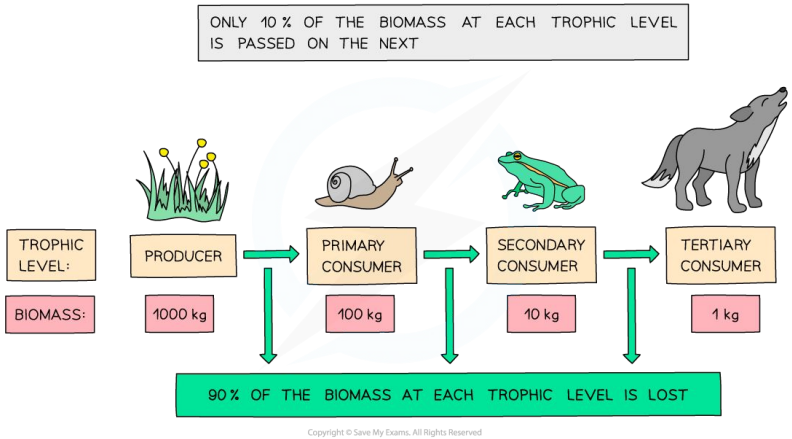




This part will be covered in the future lectures

Image source: deci.ai

Adapted together



Brain consumes 20 joules energy per second

LLama-7B consumes 335 joules per request

Efficiency (Energy) loss at each level - each component was developed independently

Hardware → Software → Model

Image source: save my exams

>>> Course description

We covered some of the problems - let's recall them

What are possible solutions?

Problems Addressed:

- Lack of knowledge about the most suitable models and pipelines for specific tasks within a vast search space.
- Current models not being designed with computational efficiency in mind.
- Increasing computational demands of modern models.
- Independent design of software, hardware, and models.

Solutions:

- Hardware-Software-Model Co-Design: Implementing integrated design approaches similar to those in existing solutions. [1]
- Automated Optimization of Pipelines and Architectures: Utilizing automated techniques to optimize pipelines and architectures for specific tasks.
- Model Optimization and Compression: Optimizing and compressing existing models to improve computational efficiency.

[1]: Codesign-NAS: Automatic FPGA/CNN Codesign Using Neural Architecture Search

Black Box optimization	Black Box optimization is a set of instruments which can be used for hyper-parameter tuning, neural architecture search, in automated machine learning and for Hardware-Software-Model
Automated Machine Learning	We will discuss existing solutions and future directions of how algorithms and models are created and optimized automatically, allowing data scientists and analysts to focus on higher-level tasks. We will review the latest State-of-the-Art methods in Automated Machine Learning and understand their advancements and implications.
Neural Architecture Search	Discussing the core ideas of NAS, modern approaches, limitations, and applications
Sparsification, Knowledge Distillation and other	Exploring various sparsification techniques, their limitations, applications, and recent approaches, including their application to LLMs.
Quantization	Discussing the principles, limitations, and applications of quantization techniques, including recent approaches and their application to LLMs.

>>> Thank you