



курс: управление научными проектами

## лекция 6

# от чего зависит и что такое эффективность цифровых вычислений

(процессор, память, язык программирования,  
цели вычислений)

5/12 ноября  
2024



**ПОЛИТЕХ**

Санкт-Петербургский  
политехнический университет  
Петра Великого

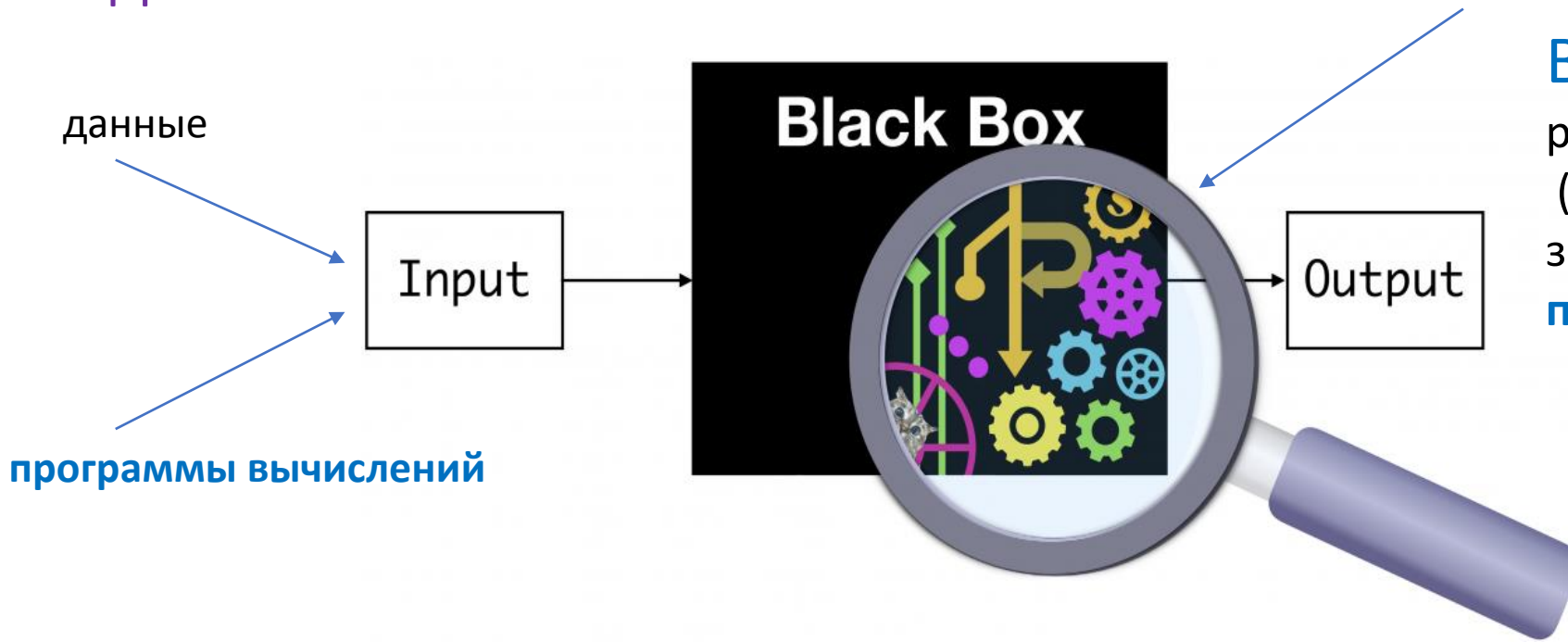
# Введение: программировать vs «обучать» компьютер – «черный ящик»

Как описать и понять «внутренний» процесс **трансформации** текста программы в результат вычислений внутри «черного ящика»

Вопрос: что такое программа с точки зрения :

- прикладной задачи ?
- физики протекающих процессов ?

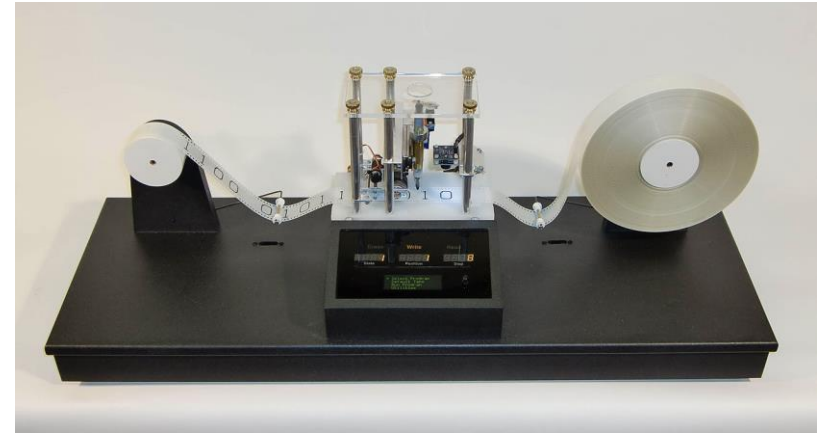
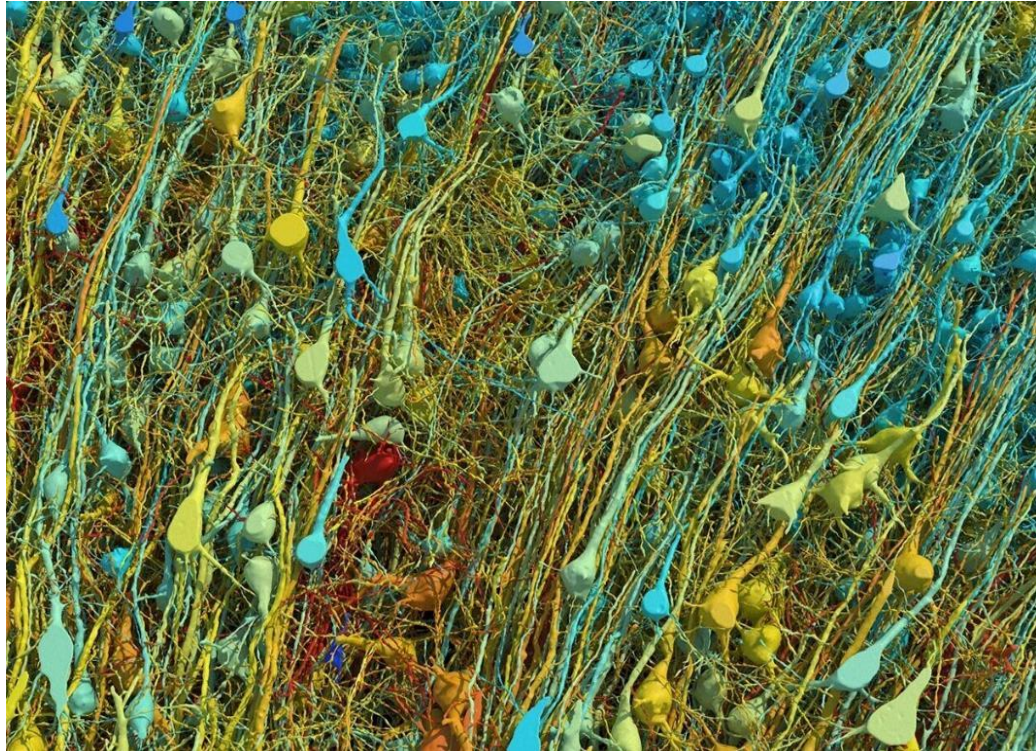
Вход:



**Выход** – числа кодирующие результат и статус решения (успешно или не успешно) завершилась программа вычислений

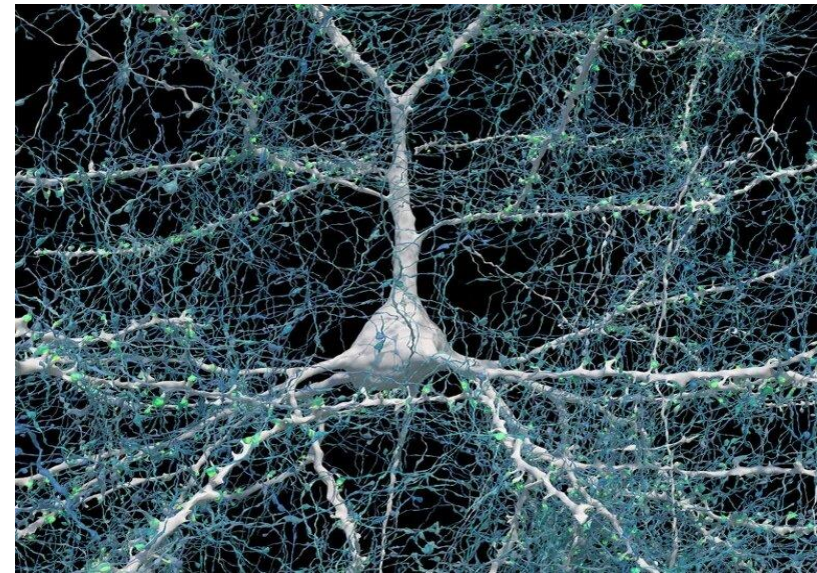


# Мозг-трансформер: «чего»? во «что»? и «как»?



Машина Тьюринга

изображение одного кубического миллиметра мозга с  
точностью до каждой клетки



нейрон (белый) с 5 600 аксонами (синий). Синапсы, обеспечивающие  
эти связи, показаны зеленым цветом.

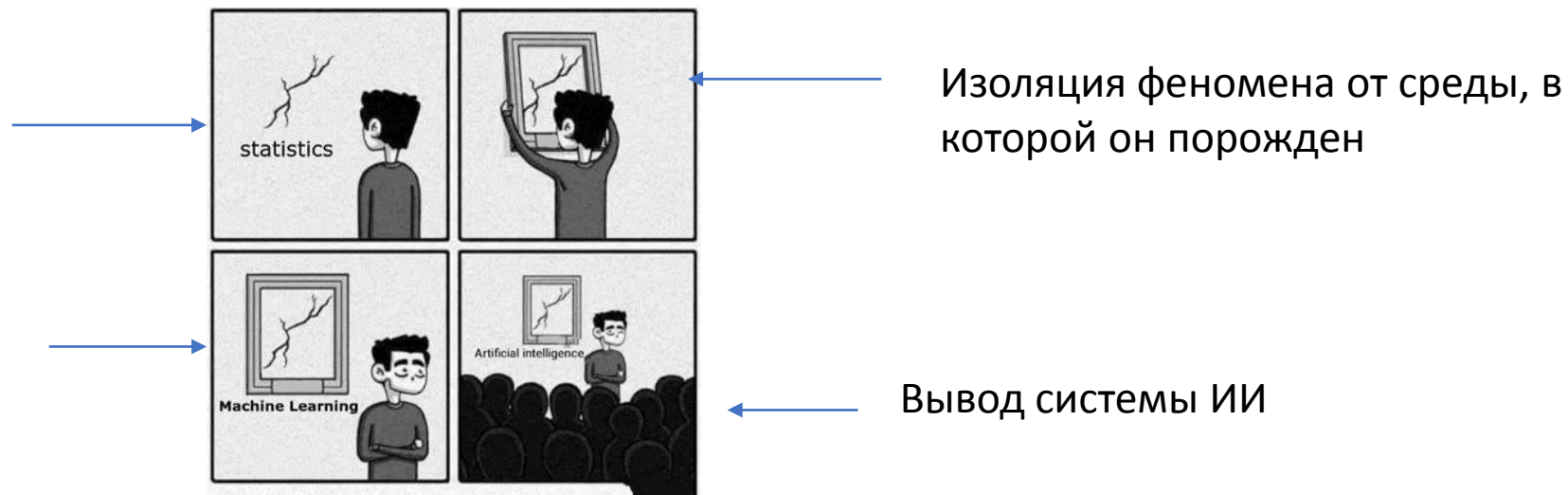
# Суперкомпьютерные технологии (СК): Проблема №1

«Сложность программирования и машинного обучения»:

для пользователей СК «**черный ящик**», а для «самого СК» – прикладные задачи пользователей - каждый раз **совершенно «новые»**.

Наблюдаемый феномен

Разметка и классификация данных – машинное обучение ИНС



Основной вопрос пользователя - как повысить скорость обучения и точность решения своих прикладных задач? **Решение проблемы:** сделать так, чтобы на вопрос пользователей мог ответить «сам СК» : какая часть используемого ПО и АО «ему» мешает работать



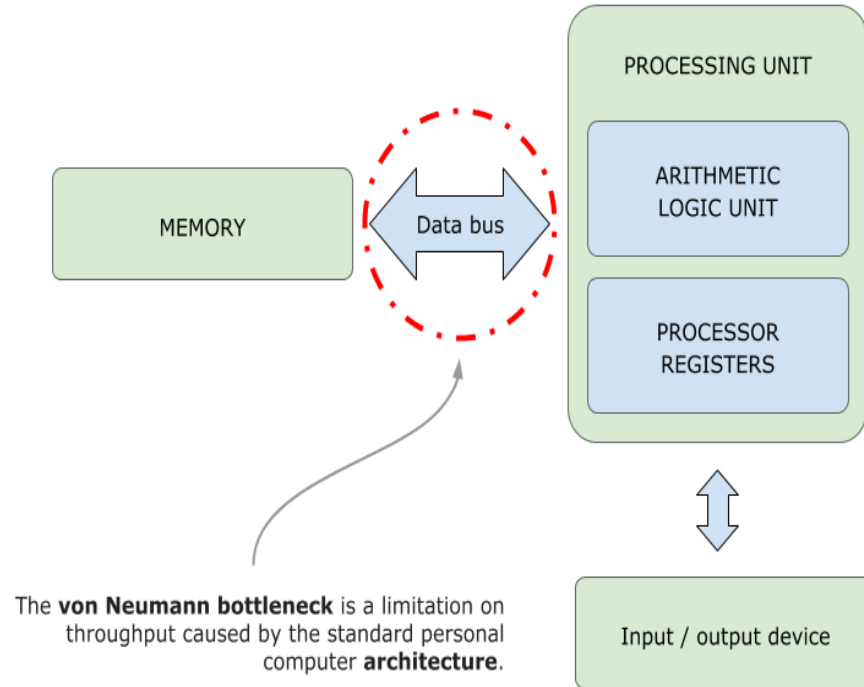
# Архитектура фон Неймана: является ли человек ее частью ?

Решение как последовательность чисел

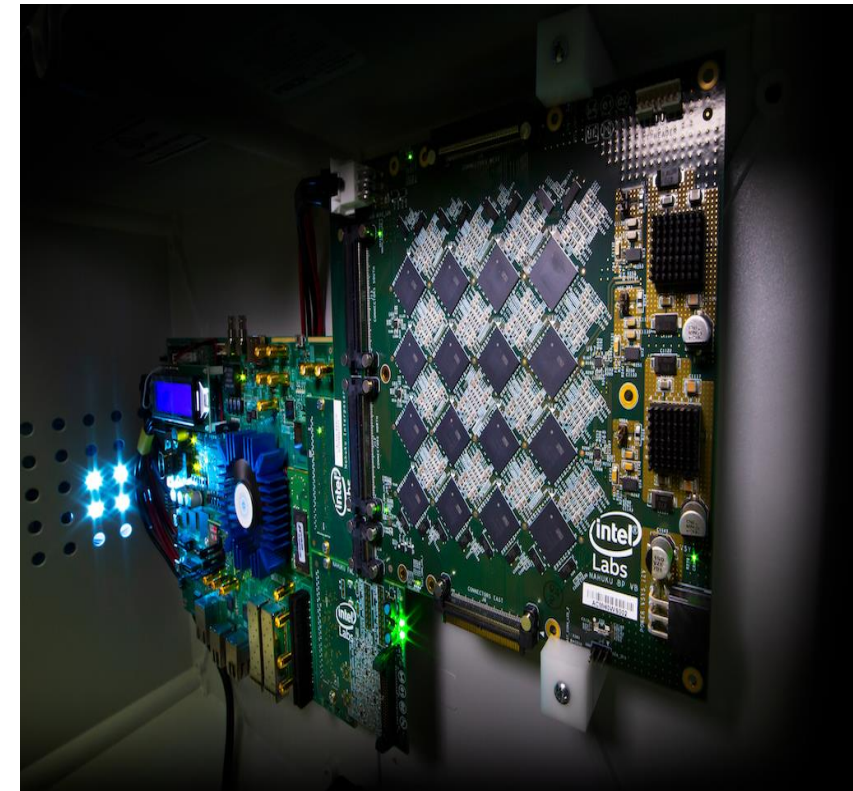
1, 56. 1003, 0, 1.....



В поиск смысла



von Neumann architecture



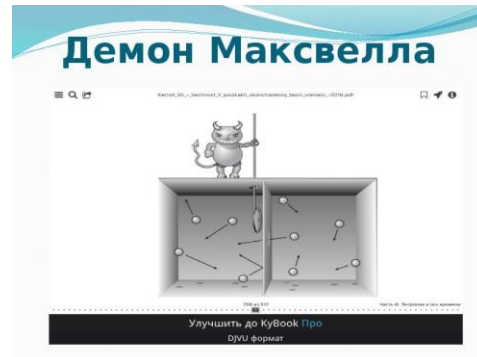
задача параллелизма вычислений и синхронизации результатов может занять большую часть самых вычислений

# Есть ли место для «демона Максвелла» в машине Тьюринга ?

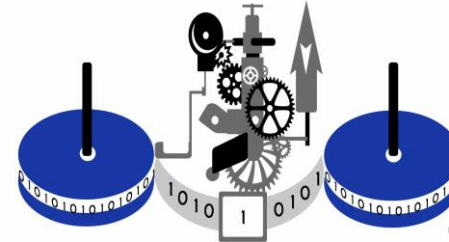
## Машины



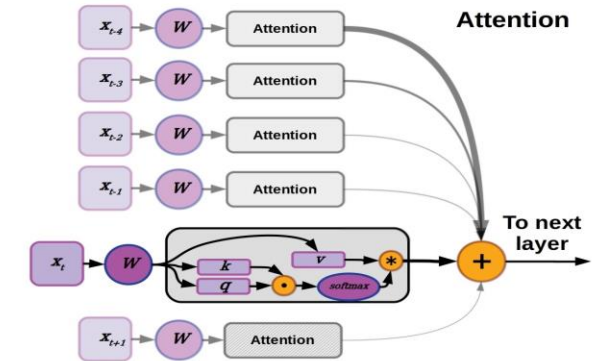
паровоз Черепанова



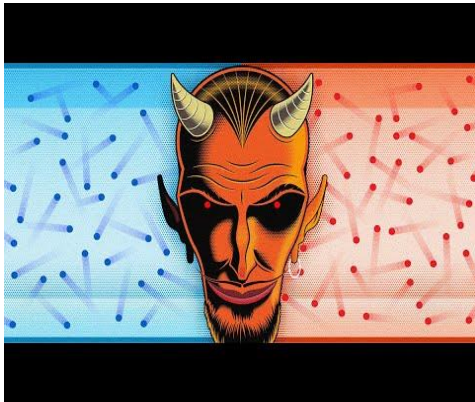
Сциларда



Тьюринга



GPT трансформер



- **Машина Сциларда – МС (1929 г)** «позволяет» извлекать с помощью гипотетического существа – демона Максвелла, информацию, содержащуюся в окружающей среде и **трансформировать ее в свободную энергию**
- А. Тьюринг (1936 г.) переформулировал теорему Гёделя (1932 г.) о неполноте, заменив универсальный формальный арифметический язык на **простые гипотетические устройства - «машины Тьюринга»**.

Итак, компьютерные науки ждет технологий передел : «объединение» машины Тьюринга (МТ) с механизмом self-attention, чтобы извлекать «нужную» информацию из внешней среды и «предупредить» МТ, когда ей пора остановится 😊

# «Искушения» компьютерных наук (КН)

**Искушение** - это побуждение нарушить (объективные) законы

90-е годы 20 века: **задачу решим на компьютере** («действие» - программы код, **КТО** напишет ???)

20-е годы 21 века: **задачу решит ИИ** («действие» - обучение, **чему и Как** надо «научить» ???)

Практическая интерпретация проблемы «искушения»

- всё **«хорошее»** (например, правильно написанная программа) более «хрупко» (в смысле **вероятности сохранить нужное свойство при любых изменениях**), чем любое **«плохое»** (антоним слова «хорошее», «плохое» – значить устойчивое к возможным изменениям или «робостное»)

В компьютерных науках **«хрупкое» понятие** – это **«хорошая программы»**. Это понятие было заменено **условно** более **«плохими»** понятиями как : **«вычислимость, перечислимость, разрешимость»**,

Эти понятия носят условный характер, которые задают **«границы»** понимания полученного результата пользователем –

- результат должен быть формально **точным** или
- результат надо получить **быстро**



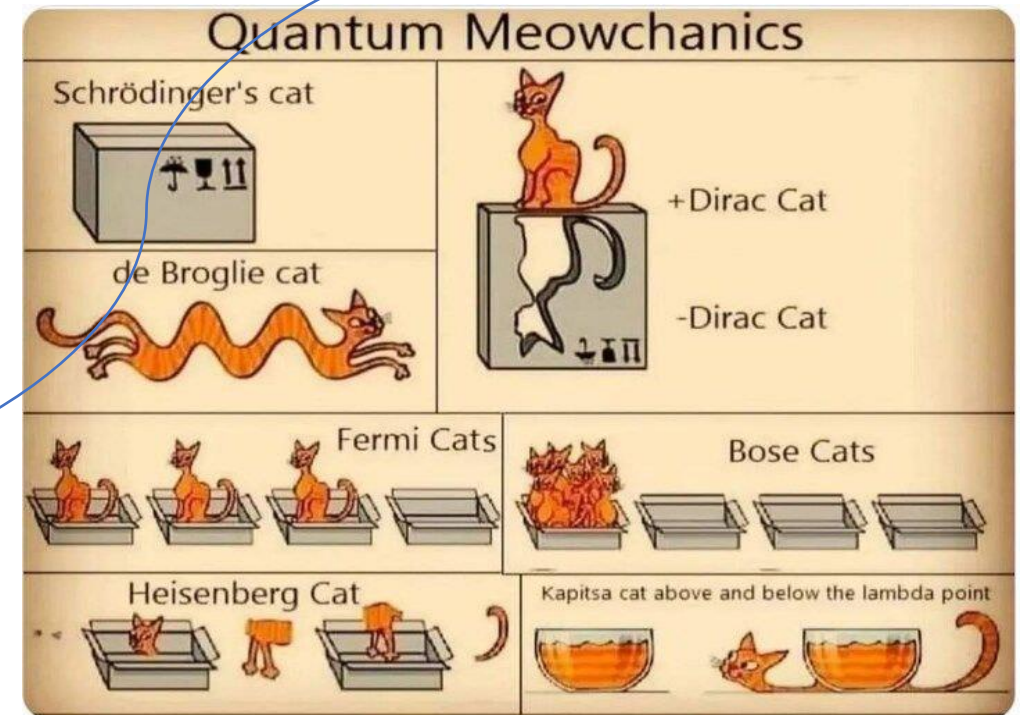
# Роль «кота Шредингера» в компьютерных науках



Суть проблемы суперпозиции состояний  
быстро или точно/ успешно или не успешно

считается, что прямых аналогов квантово-механических систем в макромире не существует...но

Смыслы результата вычислений – это метафора «кота Шредингера» находящаяся в суперпозиции логически несовместных состояний





# Программирование как обучение «машины-автомата»

- **Машинное обучение** -
  - термин из расширенной спецификации **алгоритмических возможностей ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ**,
    - связанный с использованием декларативных («знать, что») и процедурных («знать, как») знаний, полученных ранее в процессе решения прикладных задач,
      - для решения обратных задач цифрового моделирования (построением алгоритмов) и объяснением результатов вычислений
  - **Информация** – *informatio* (лат.): **формирование**, то есть процесс, в котором «форма» (структура) получает воплощение:
    - Цицероном *informatio* использовал для таких понятий как идея
      - св. Августин использовал словосочетание *informatio materiae* в смысле процесса «формирование материи».
        - ISO 5127:2017 *information* - сообщение, которые порождают значение (смысл)
          - в теории передачи информации Шеннона - **значение  $I = -\log P$**
    - Будем исходить из следующих аксиом (Блум, 1967 г.):
      - Каждая вычислимая функция имеет бесконечное число программных представлений
      - Существуют алгоритмически неразрешимые задачи (самоприменимость машины Тьюринга, задача «останова» алгоритма, задача синтеза алгоритмов на основе анализа данных и пр.

Уточнение: Алгоритмическая неразрешимость понимается в том смысле, что **для алгоритма**, решающего данную задачу, существует набор **входных данных**, для которого алгоритм **дает неверный результат** (результат взаимодействие алгоритма с внешней средой в общем случае не вычислим)

# Какой из ответов имеет правильный (имеет смысл)

## C++

- $1+2 = 3$
- $1.0 + 2 = 3.0$
- $1.000000 + 2.000000 = 3$
- $1.00000000000000000001 + 2.0 = 3.0$
- $1 + 2.0000000000000000 = 3.0000000000000000$
- $1.001 + 2.0001 = 3.0011$

LLM: один полюс два равно 3

## JavaScript

$$5 + \text{«3»} = 8$$

$$5 + \text{«3»} = 53$$



## Online learning and continuous-flow

- Современный компьютер хорошо справляется с дискретными данными, желательно целочисленных данных,
- . когда речь заходит о последовательностях, непрерывности, бесконечно малых или бесконечно больших значениях , найти некоторую похожесть или приближение есть проблемы
- Пока мы интерпретируем данные в последовательность дискретных кадров, дробим, разделяем и обрабатываем каждый фрейм как нечто статическое и конечное.
- подходы bi-directional soft attention (см. BERT) позволяют связывать эти самые кадры в работе с языковыми моделями
- современные подходы машинного обучения лишены возможности обучаться непосредственно в процессе решения поставленной задачи.

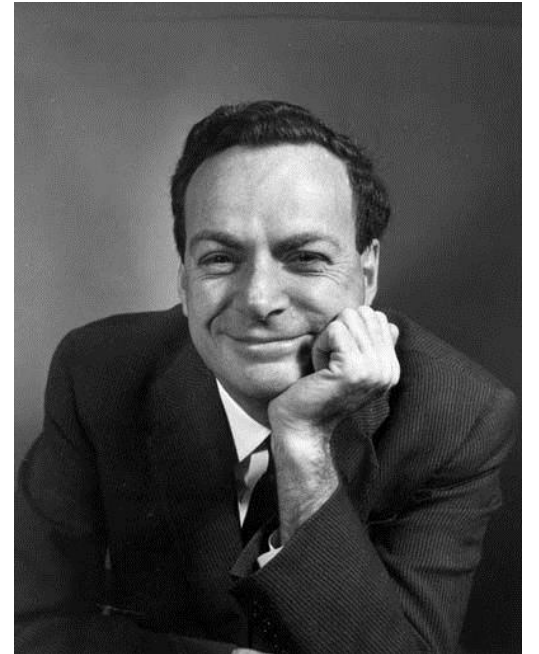
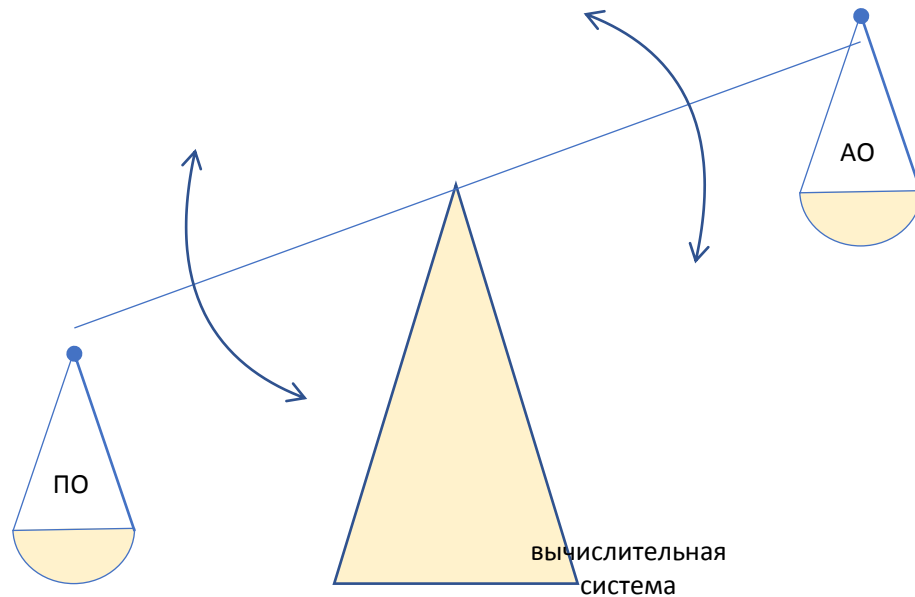


Are there some questions that computer could never answer for us,  
however beautifully made it might be ?

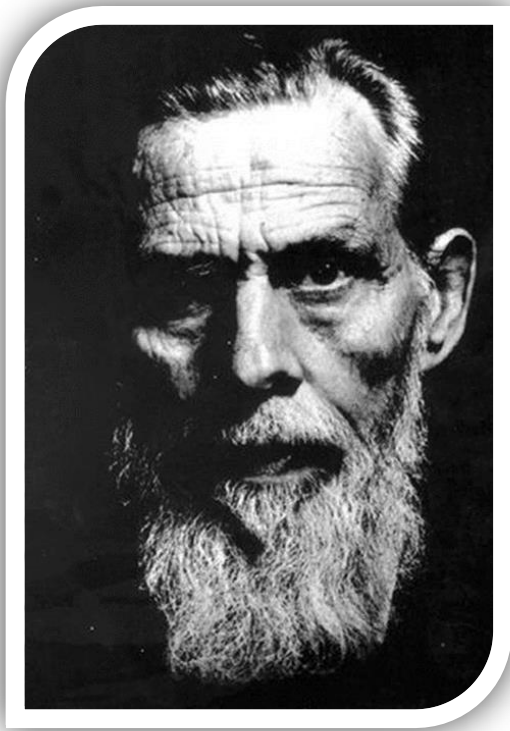
Richard Feynman

Есть ли какие-то вопросы, на которые **компьютер никогда не смог бы ответить** за нас, каким бы прекрасным он ни был?

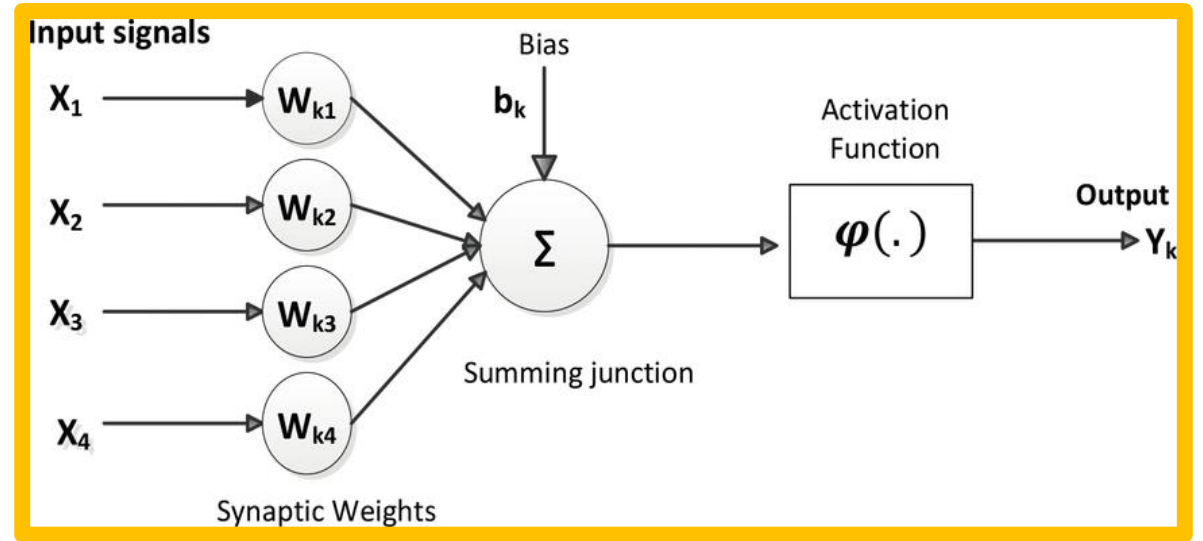
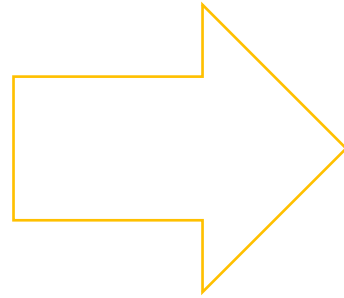
Ричард Фейнман



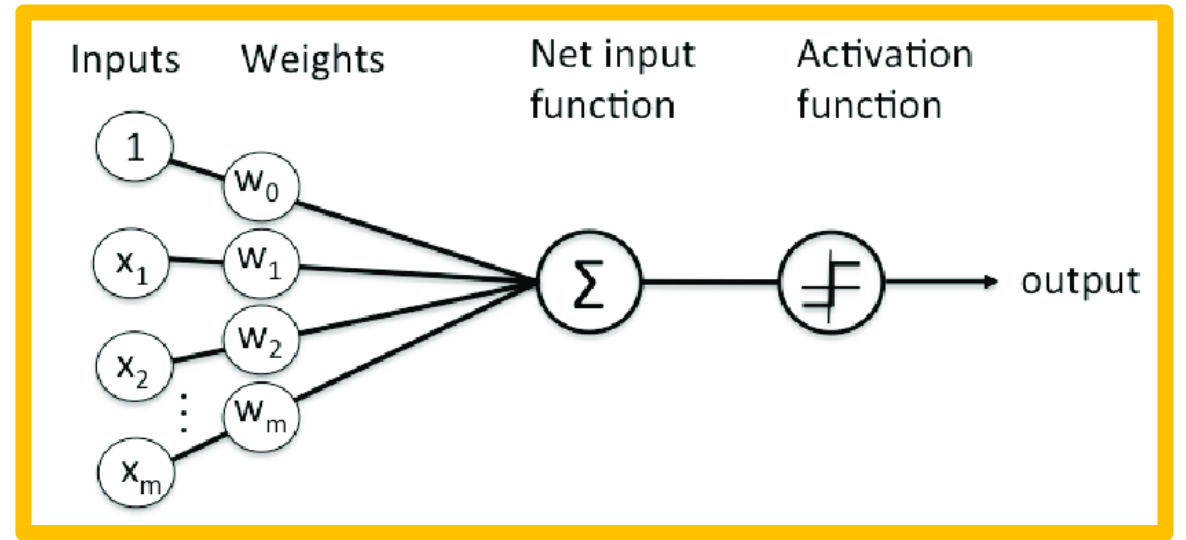
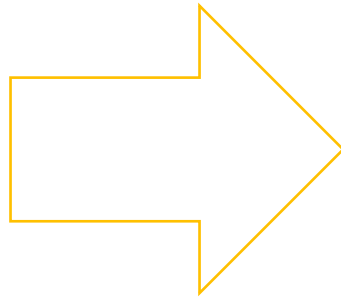
Что современный компьютер не может вычислить в принципе и почему ?



Уоррен Мак-Каллок  
(1898 —1969)



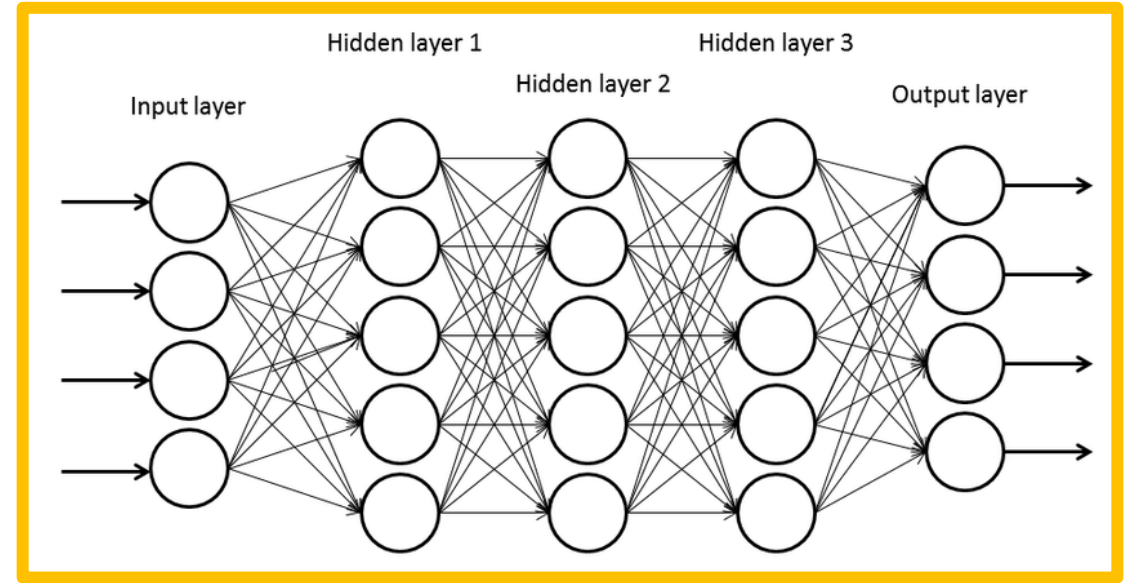
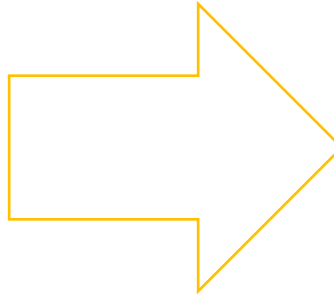
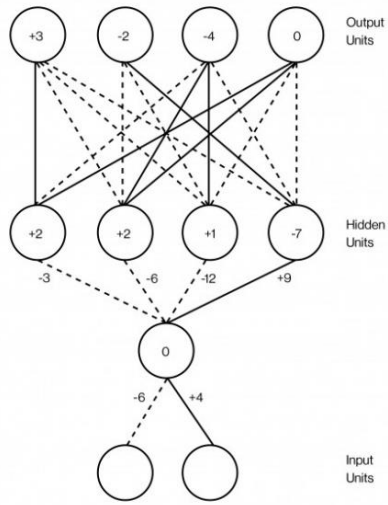
"Nervous Activity" (1943)  
(Логическое исчисление идей, имманентных нервной деятельности)



**Фрэнк Розенблатт** ([англ. Frank Rosenblatt](#)) : «надо понять фундаментальные законы обработки информации, включая как машины, так и человеческий разум».

Ф. Розенблатт читал курс лекций, который назывался «Теория механизмов мозга», а в 1962 году на основе материалов курса он опубликовал книгу «Принципы нейродинамики: Перцептроны и теория механизмов мозга»





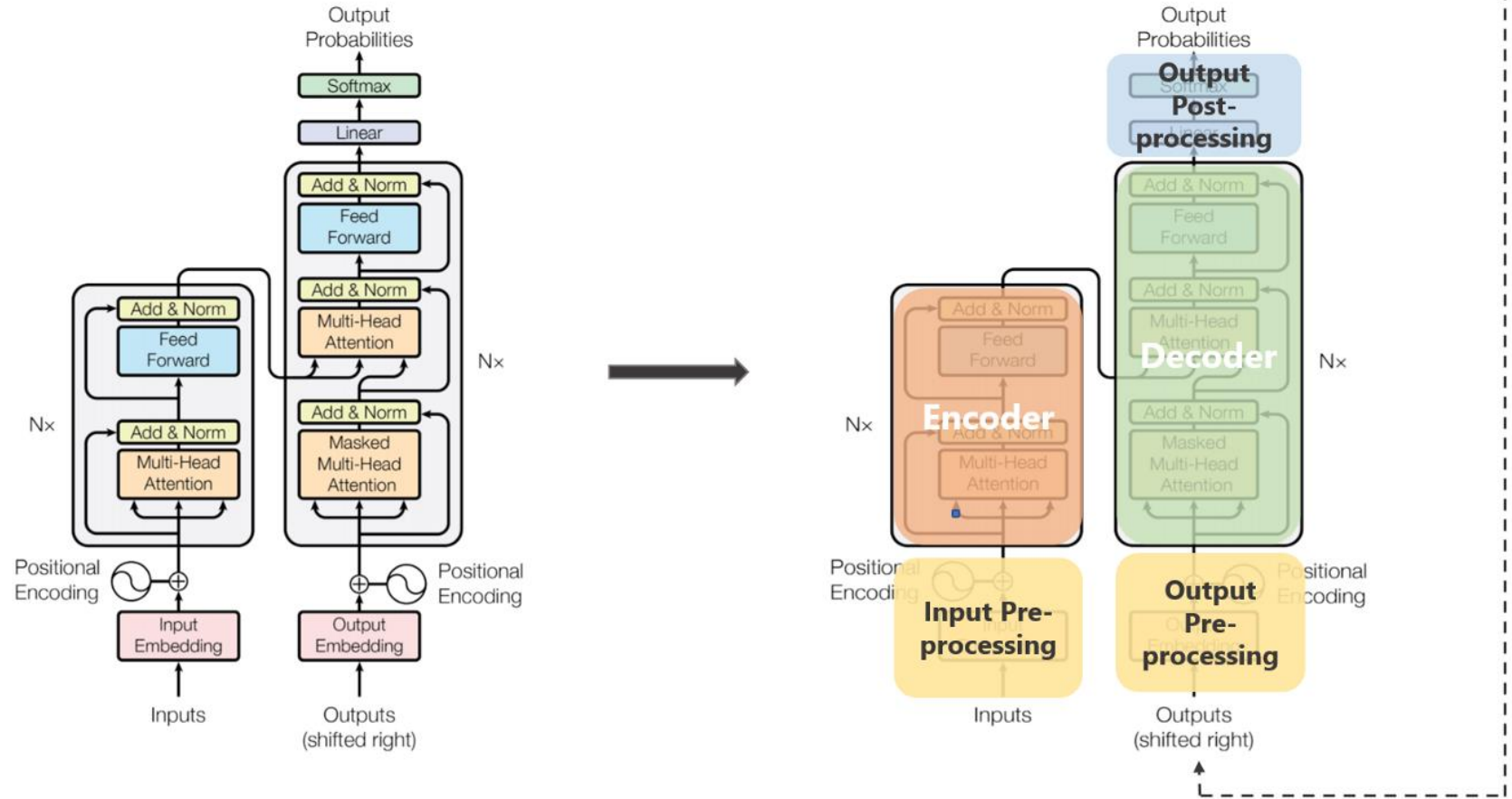
Джеффри Хинтон (1947) - Нобелевская премия 2024 года по **физике** (???) вручена за:

«за основополагающие открытия и изобретения, которые позволяют осуществлять **машинное обучение** искусственных нейронных сетей».

Дж. Хинтон (2020): **Глубокое обучение может многое, но .... я думаю, что должно быть еще много концептуальных прорывов:** Есть два подхода к тому, как мозг воспринимает изображение: одни считают, что мозг обрабатывает изображение («пиксели»), другие — что символы. Оба подхода неверны: мозг оперирует не пикселями и не символами, а векторами нейронной активности (т.е. наборами активаций тех или иных нейронов)



[Ашиш Шарма](#)



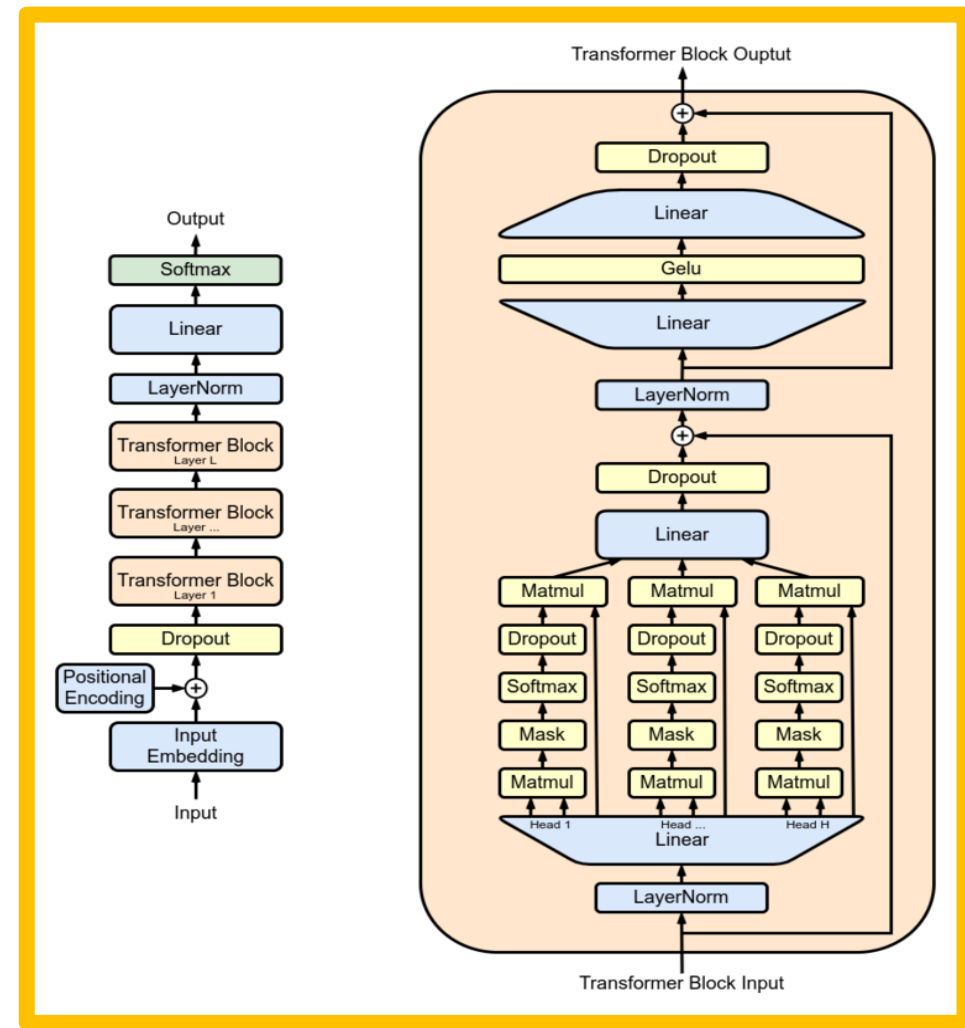
# Attention Is All You Need



Ilya Sutskever



AI is nothing **but digital brains** inside large computer.



### Feedback (RLHF):

**Программированию** как процесс поиска «багов» в программном коде и пояснений по работе кода на человеческом языке. Реальных супер разработчиков ПО просили описывать словами, как они подходят к задаче, как «борются» с ошибками и как они эти ошибки исправляют. **Ответ: пытаемся исполнить написанный код в своем сознании**

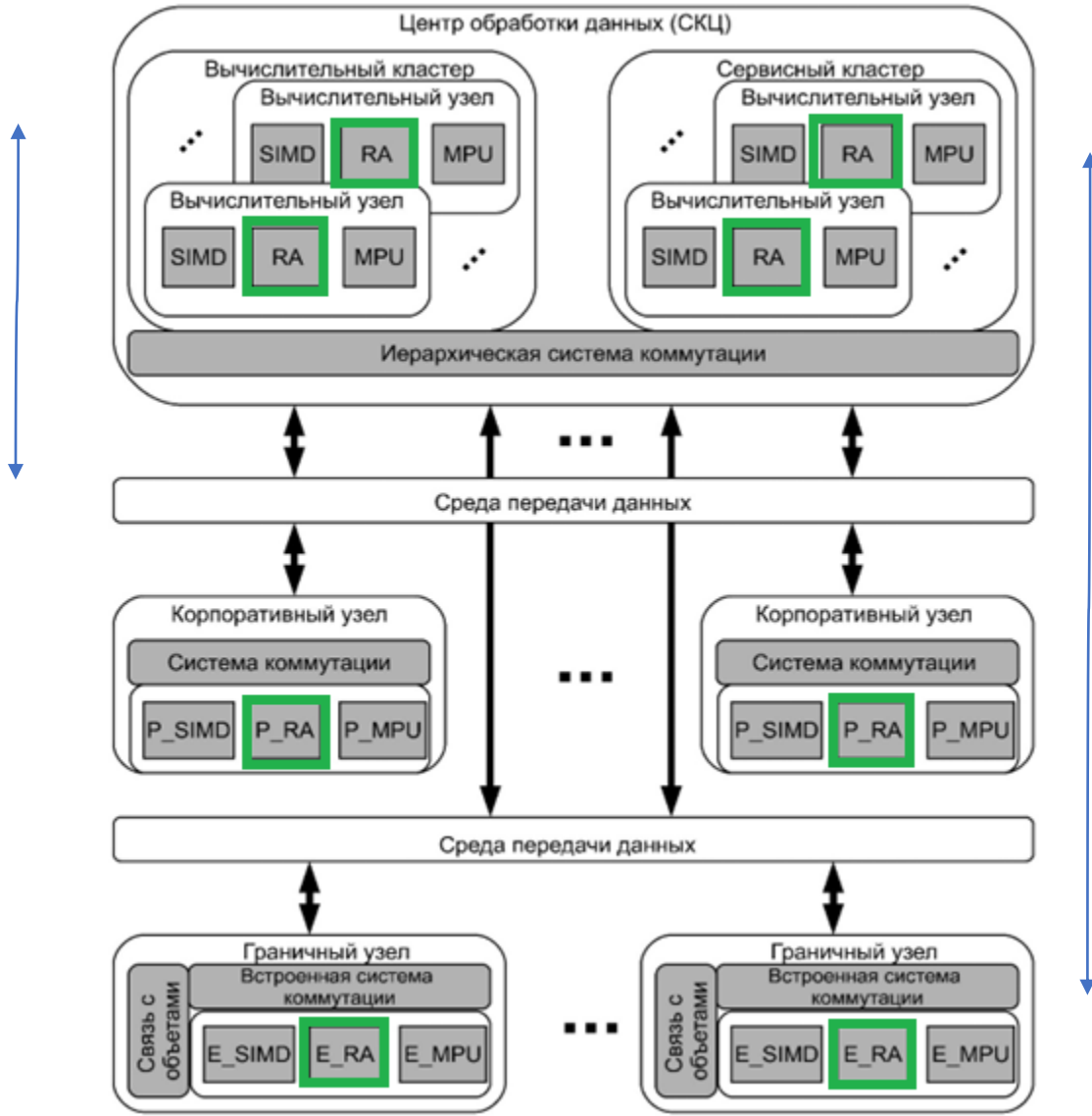


# интеллектуальное пополнение «машины Тьюринга» : «машина Геделя» способная к обучению и само-программированию

Уровень «объяснения» и «моделирования»

Уровень «обобщения» и обучения  
>

Уровень доступа и предобработки «больших данных»  
>20 Гфлопс/Вт



## От математических к языковым моделям

- Трансформеры это один из новых архитектур цифровых систем вычислений. Ряд таких больших языковых моделей, как ChatGPT, GPT-4 и Llama, созданы на основе трансформерной архитектуры (Дж. Хинтон: мозг оперирует не пикселями и не символами, а векторами нейронной активности (т.е. наборами активаций тех или иных нейронов)).
- Формально трансформер
  - это вид простой (не обязательно "глубокой") нейросетевой архитектуры – «черный ящик», который используется для обработки последовательностей данных. Типичным примером таких последовательностей могут быть предложения из слов на естественном языке, которые имеют некоторый смысл (здесь смысл понимается как «значение» или дескриптор элементов языка, на котором написано входное предложение)
  - «оператор» преобразующий входную последовательность данных в некоторое цифровое представление, которое инкапсулирует информацию
    - как об самом элементе так и окружающем этот элемент контексте, для которого «значение» смысла и было определено

## Преобразование «слово в числовой вектор»

- Полученное на выходе трансформера «цифровое» представление входной последовательности данных **можно передать** в другие системы, которые используют эту информацию, чтобы решать такие задачи как :
  - **генерация** новых предложений (последовательности слов), связанных по смыслу с исходной последовательностью (перевод текстов)
  - **классификация** входных последовательности данных за счет выявления паттернов и скрытых в входной последовательности взаимосвязей
- Входные последовательности для трансформеров могут проходить «интеллектуальную» предобработку, например классифицироваться с помощью «леса случайных деревьев вывода»

# компьютер - это ... трансформер возможных счетных состояний

## ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ В конечное множество правильных ответов

Деревья классификации возможных состояний системы

Имеется 1)

**обучающая выборка**  $(x_j, y_j)$ , где  $x$  – вектор признаков (давление, частота с/с),  $y$  – **реакция организма** (вычисленные индексы Робинсона, Старра, Кердо....),

2) индексы  $j \in J_k(x)$ , попадают совместно с вектором признаков  $x$  в один и тот же  $p$ -й лист  $k$ -го дерева классификации.

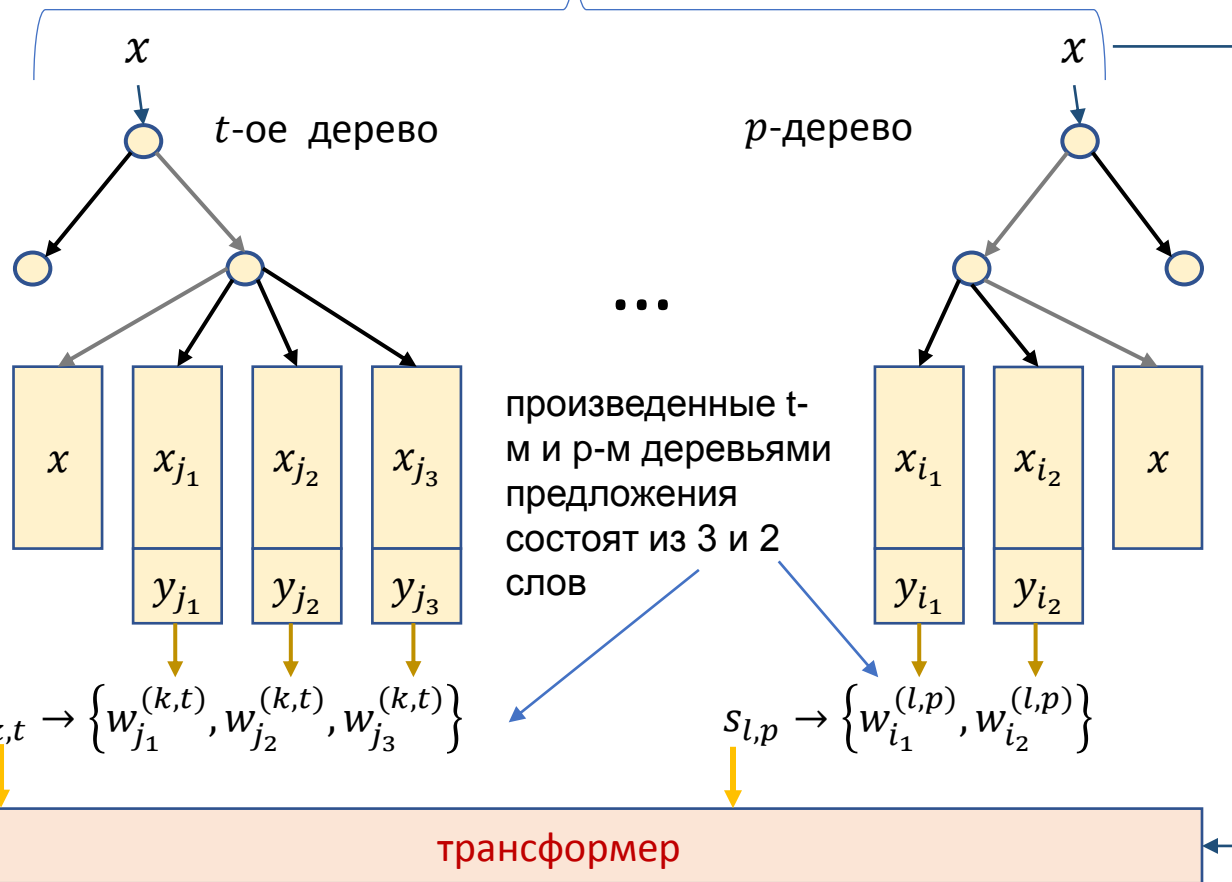
$(x_j, y_j)$ , представим в виде **предложения**  $s$ , состоящего из слов, обозначаемых как  $w_i^{(k,p)} \in J_k(x)$ , из этих слов «собрать» диагноз и на основе клинического опыта «выбрать» корректирующее воздействие, которое «подать на организм пациента.

$i \in J_k(x)$ ,

$$s_{k,p} = \{nw_{i1}^{(k,p)}, \dots, w_{ip}^{(k,p)}\}_{1ir}$$

**«Предложение»**  $s: s_{k,t} \rightarrow \{w_{j_1}^{(k,t)}, w_{j_2}^{(k,t)}, w_{j_3}^{(k,t)}\}$

$k$  - номер дерева;  $p$  - номер листа;  $i$  - индекс примера (индекс слова в  $p$ -м предложении), который попадает в  $p$ -й лист.



ИЗ КАЖДОГО «ЛИСТА» ДИРЕКТЕВ МОЖЕТ ВЫХОДИТЬ РАЗНОЕ КОЛИЧЕСТВО «ВЕТВЕЙ»:

- 1) из листа  $t$ -ого дерева выходит три «слова» описания реакции, организма, а из листа  $p$ -дерева выходит описание, состоящее из двух «слов»
- 2) На вход системы подается текущий вектор « $x$ » и соответствующие ему значения индексов Робинсона, Старра, Кердо.

произведенные  $t$ -м и  $p$ -м деревьями предложения состоят из 3 и 2 слов

### Задача ИИ системы

- 1) вычислить оценку  $[x^*]$
- 2) По этой оценке рассчитать значения индексов Робинсона, Старра, Кердо, то есть  $\hat{y}$  и выбрать вид и форму воздействия  $z = q(\hat{y}, d1, d2, d3)$ , где коэффициент  $d_i$  получаются также в процессе обучения **для каждого пациента отдельно**



# реализация

- Данные, которые подаются в трансформер, нужно преобразовать в последовательность **токенов** то есть набор целых чисел, например

[101, 7592, 2045, 1010, 3475, 1521, 1056, 1996, 4633, 3835, 2651, 1999, 2852, 2891, 10175, 1029, 102]

- Такое преобразование текста в целочисленный вектор проще всего можно сделать с помощью **словаря** – таблицы, в которой **всем словам сопоставляются некоторое число**. Если конкретного слова нет в словаре, то для «токенизации» этого слова используется дополнительный набор чисел.
- Существуют разные способы **токенизации предложений**, при которых слова перед индексированием разбиваются на фрагменты или используются специальные токены для обозначения начала и конца предложения, чтобы в цифровой модели обрабатываемого текста можно было учесть «больше» информации о контексте
- Ясно, что **трансформеры могут обрабатывать не только текст**. Эта архитектура может использоваться для различных задачах, если
  - изображение «**разрезать**» на не пересекающиеся фрагменты размером, например, 16 на 16 пикселей, а затем конкатенировать их в вектор.
  - на вход трансформерной модели подать идентификаторы последних N объектов, которые были просмотрены пользователем
- Итак, если в некоторой тематической области **можно создать осмысленное представление входных токенов, то их можно передать трансформеру для обработки**

# От слов предложения к их цифровым «эмбедингам»

- Получив **из слов** предложения вектор целых чисел, который в том или ином виде представляет входные слова/данные, можно трансформировать полученное представление в так называемые **эмбединги** — представления информации о входной последовательности, которые облегчают её дальнейшую обработку цифровыми **алгоритмами машинного обучения**.
- Эмбединги в определенном формате кодируют «смысл» токенов, представляя информацию о «смысле» в виде последовательности чисел. Так, на первом этапе обучения трансформеров эмбединги синтезируются как **случайная последовательность чисел**, а значимое **смысловое представление слов формируется уже во время обучения**. Однако, у эмбедингов есть **два наследственных «заболевания»**:

## «Наследственные заболевания»:

- **Первое** - они не учитывают контекст, в котором синтезировались токены входного предложения, что может привести к эффекту известному как «мешок слов». Поэтому для «излечения» нужно сохранить порядок токенов, для чего следует сформировать еще один набор эмбедингов, кодирующих положение каждого токена во входной последовательности. Этот второй набор комбинируется с эмбедингами токенов
- в итоге проблема порядка решается с помощью использования нескольких векторов-эмбедингов: вектор токенов + вектор позиций токенов => вектор позиционно закодированных токенов
- **Второе**, у токенов могут быть разные значения в зависимости от значения соседних токенов. В современных трансформерах эта проблема решается с помощью механизма **внимания**

# Механизм внимания (1)

- Основа трансформенной архитектуры лежит механизм внимания. Этот механизм позволяет трансформеру (нейросети) «понять», какая часть обрабатываемой входной последовательности релевантна (актуальна) контексту обрабатываемого **запроса**.
- Механизм внимания определяет для каждого **«токена входной» последовательности**, какие другие токены необходимы для его «понимания» в данном контексте запроса.
  - Чего же «добивается» механизм внимания ?



## Механизм внимания (2)

- заменяет каждый эмбединг токена входной последовательности на эмбединг, содержащий информацию о соседних токенах этого же текста
- В итоге вместо использования одинакового эмбединга для каждого токена вне зависимости от контекста **используется несколько эмбедингов**.
- Если бы мы знали, какие токены релевантны текущему контексту запроса, то узнать этот контекст можно с помощью
  - **средневзвешенного значения токенов**или,
  - **линейной комбинации эмбедингов**.

## В литературе линейную комбинацию эмбеддингов обычно представляют так:

$$w_1 \text{ Embedding Vector}_1 + w_2 \text{ Embedding Vector}_2 + w_3 \text{ Embedding Vector}_3 = \text{Combination Embedding Vector}$$

до применения механизма внимания у эмбеддингов слов входной последовательности нет контекста их соседних слов.

Для того, чтобы применить механизм внимания, нужно узнать матрицу весов  $\{w_i\}$ , которая позволит выразить комбинированный вектор эмбеддинга для всех слов входной последовательности.

Эмбеддингам, соответствующим наиболее релевантным для выбранного токена частям последовательности, должны быть присвоены веса большего размера. Это должно обеспечить размещение в новом **комбинированном векторе эмбеддинга** самого важного контекста. Эмбеддинги, содержащие информацию об их текущем контексте, иногда называют *контекстуализированными*, и именно такие эмбеддинги создает трансформер.

# Как формально можно представить «Механизм ВНИМАНИЯ»

Дана непараметрическая модель

«черного ящика

«вход»  $\{v_i\}$  - значение

$$y = \sum_{i=1}^N \alpha(q, k_i) v_i$$

$$\alpha(q, k_i) = \text{softmax}_i \left( \text{score}(q, k_j) \right)_{j=1}^N$$

$q$  – query -

$k_i$  – key -

$v_i$  – value –

запрос

ключ

значение

«механизм внимание» в данном случае это преобразование «ядра» непараметрической модели  $\alpha(q, k_i)$  в оценку «вероятности» некоторого действия

- Такое преобразование характерно для задач обработки сложных последовательностей, например предложений естественного языка и позволяет работать с произвольным числом пар (ключ, значение)
- Обычно функция  $\text{score}(q, k_i) \propto q^T k_i$ , либо линейная функция, либо матричный оператор  $(W_Q q)^T (W_K k_i) = q^T W_Q^T W_K k_i = q^T W k_i$

# Многокомпонентная архитектура системы трансформации входных данных в оценки состояния объекта

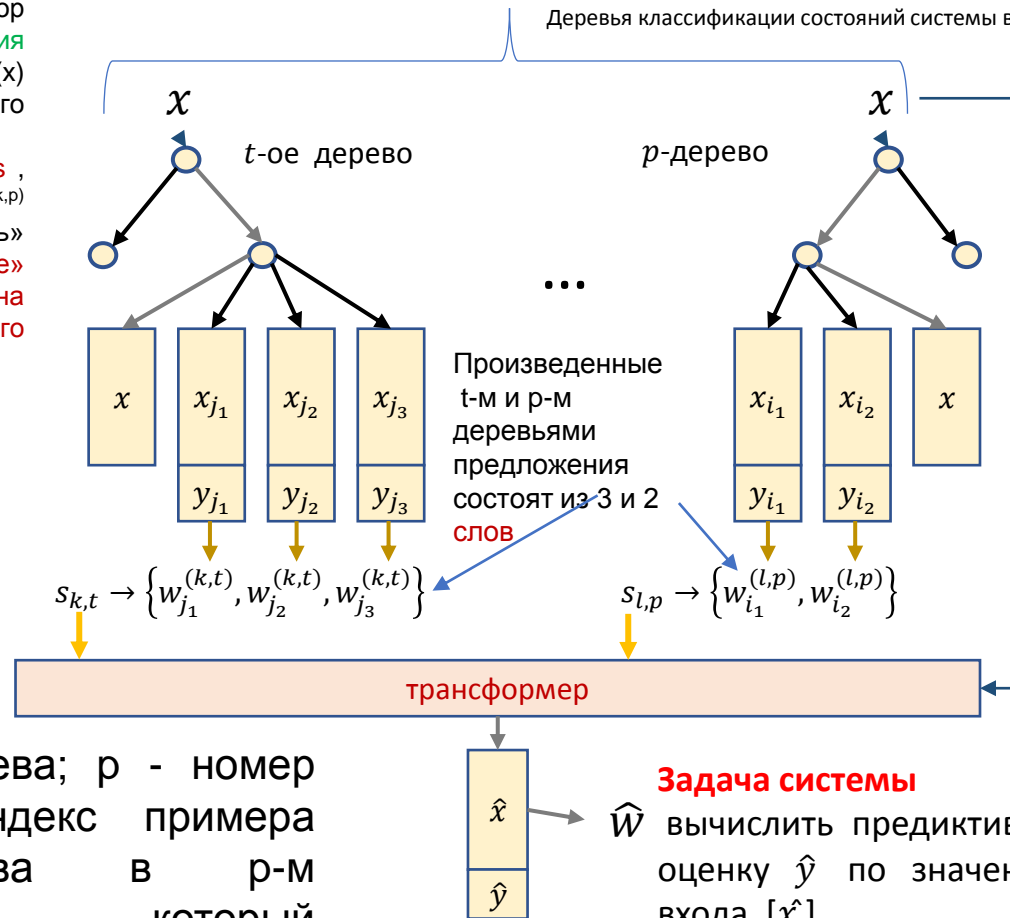
Имеется система описание которой задается текстом. Для системы сформирована **обучающая выборка**  $(x_j, y_j)$ , где  $x$  – вектор признаков (давление, частота..),  $y$  – **реакция организма/системы** и индексы  $j \in J_k(x)$  признаков  $x$  в один и тот же  $p$ -й лист  $k$ -го дерева классификации.  $(x_j, y_j)$ , представим в виде **предложения**  $s$ , состоящего из слов, обозначаемых как  $w_i^{(k,p)} \in J_k(x)$ , из которых можно «сформировать» актуальное **«лингвистическое описание»** текущего состояния в контексте (на основании) значения текущего входного сигнала  $x$

$$i \in J_k(x),$$

$$s_{k,p} = \{nw_{i_1}^{(k,p)}, \dots, w_{i_p}^{(k,p)}\}_{1r}$$

«предложение»  $s$ :

$k$  - номер дерева;  $p$  - номер листа;  $i$  - индекс примера (индекс слова в  $p$ -м предложении), который попадает в  $p$ -й лист.



- ИЗ КАЖДОГО «ЛИСТА» ДЕРЕВЬЕВ МОЖЕТ ВЫХОДИТЬ РАЗНОЕ КОЛИЧЕСТВО «ВЕТВЕЙ»:
- 1) из листа  $t$ -ого дерева выходит три «слова» описания реакции системы, а из листа  $p$ -дерева выходит описание, состоящее из двух «слов»
  - 2) На вход системы подается текущий вектор « $x$ » и соответствующие ему значения выхода системы



## Итак: архитектуры СК могут и должны быть модифицированы

классические подходы к организации программно-управляемых вычислений, известные как принципы фон Неймана, могут быть модифицированы, чтобы наделить компьютеры способностями к индуктивным выводам, а именно:

- **принцип однородности памяти** –
  - **трансформируется** в принцип организации вычислений в нейроморфном поле памяти, где хранятся ранее приобретенные ассоциативные знания,
- **принцип адресности**
  - **дополняется** принципом гиперконвергентности и ассоциативной маршрутизации,
- **принцип программного управления**
  - **расширяется** за счет возможности изменения порядка выполнения команд на основе результатов анализа предшествующих вычислений,
- **принцип двоичного кодирования**
  - **дополняется** кодированием категориальных признаков с использованием процедуры, которая представляет собой преобразование дескрипторов признаков в численное представление по заданным правилам.

## Варианты пути развития

- **Имеем:** **Сейчас** «машинное обучение» – синоним извлечение **ИМПЛИЦИТНЫХ ЗНАНИЙ** (**НЕЯВНЫХ**, скрытых, латентных, неcodифицированных ) из «больших данных».
- Без заранее размеченных данных современные алгоритмы машинного обучения не способны **извлечь знания из данных**, которые затем можно было бы перенести на **новые данные** (знания)
- **Вопрос:** как СК получит **максимум «знаний» из минимума данных**, то есть выявить суть явления или процесса анализируя результаты наблюдений и используя априорные знания?
- **Ответ:** нужно **встроить в СК механизм самовнимания**, чтобы ... эффективно
  - использовать и сохранять в памяти всю имеющуюся информацию,
  - выявлять релевантные и контекстные связи между данными,
  - строить гипотезы и обобщать полученные данные, чтобы **оценивать вероятности** соответствия выбранных моделей вычислений и результатов экспериментальных наблюдений, извлекая из «сырых» данных **объяснительные абстракции**, которые можно переносить на другие ситуации .....
- **задача «умного» программирования – принудительное «обучение» компьютера находить «наилучшее из возможного».**

# Обучение как программирование, а научение как самоприменимость машины Тьюринга

- *Различие между прошлым, настоящим и будущим – лишь необычайно устойчивая иллюзия.*

А. Эйнштейн

Наблюдаемая реальность включает :

- 1) уже состоявшейся (прошлое),
- 2) текущее-настоящее и
- 3) потенциально возможное (будущее), в котором то может **воплотиться процесс «обучения»**.

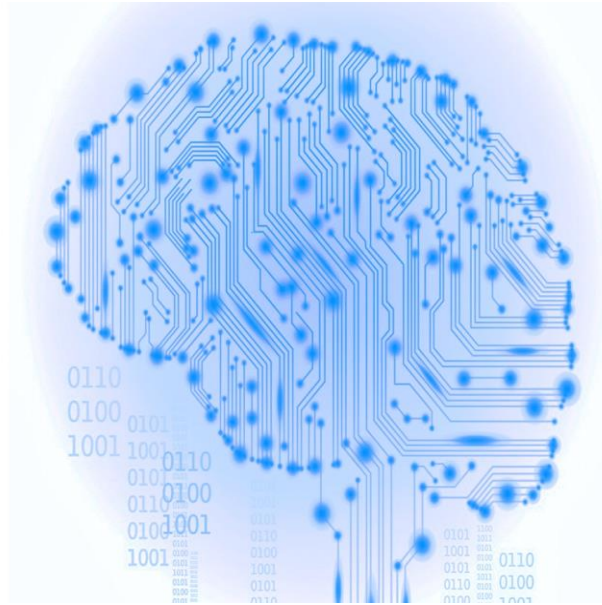
Время в мире, где «правит» информационное взаимодействие», – трехмерно  $T^3$

- Мышление и сознание «самоприменимы» и являются феноменом, который может проявляться в случае, когда «вещество» находится (переводится) **в особом «фазовом состоянии»**, в котором способно накапливать и обрабатывать информацию с использованием «процессов» вычислений - обработки символьных потоков данных.

## Часть 2 как модифицировать супер-вычислитель

### Человеческий мозг—самоприменимая сложная адаптивная система

По аналогии с мозгом—самоприменимой адаптивной системы пополнить архитектуру «машины Тьюринга» процессами нейроморфных вычислений. Частью нейроморфных вычислений являются спайковые нейронные сети, позволяющие преодолеть ограничения нейронных вычислений и эффективно использовать алгоритмы машинного обучения в реальных приложениях.



*Spiking neural network*, SNN) — третье поколение искусственных нейронных сетей (ИНС), которое отличается от бинарных (первое поколение) и частотных/скоростных (второе поколение) ИНС тем, что в нем нейроны обмениваются короткими (у биологических нейронов — около 1—2 мс) импульсами одинаковой амплитуды (у биологических нейронов — около 100 мВ). Является самой реалистичной, с точки зрения физиологии, моделью ИНС

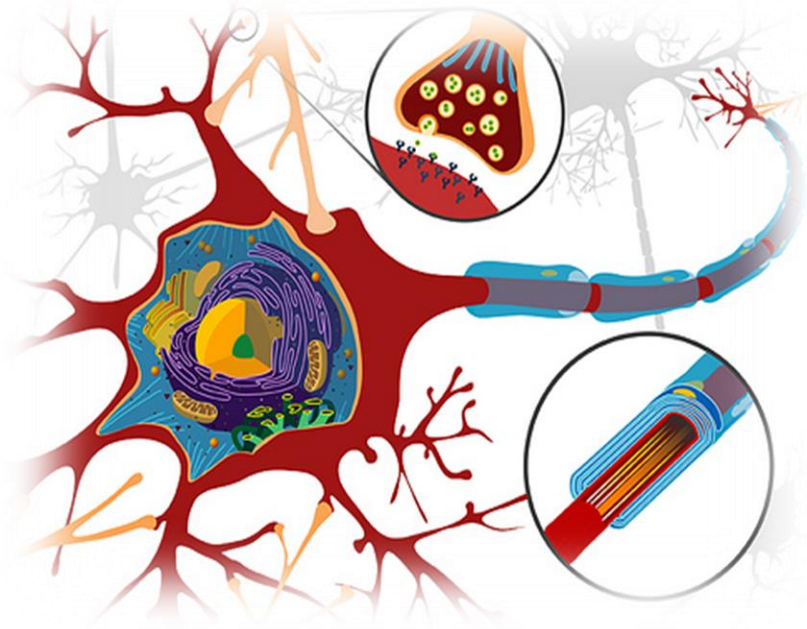


# Специфические свойства биологической памяти по сравнению с компьютерной



- ❖ **Нерепрезентативность** - она не является точным отражением событий внешнего мира.
- ❖ **Реконструктивность** - ее воспроизведение является активным процессом самосборки нейронной системы.
- ❖ **Нерепликативность** - каждое ее следующее воспроизведение отличается от предыдущего, вовлекая перекрывающуюся, но отличающуюся популяцию нейронов и синапсов.
- ❖ **Рекатегориальность** - каждая ее новая реконструкция при воспроизведении проходит оценку идентичности на весах других, связанных с ней систем.
- ❖ **Реконсолидируемость** - каждая новая реконструкция подвергается запоминанию, сходному по своим механизмам с процессами исходного запоминания.

# Специфические свойства биологической памяти по сравнению с компьютерной

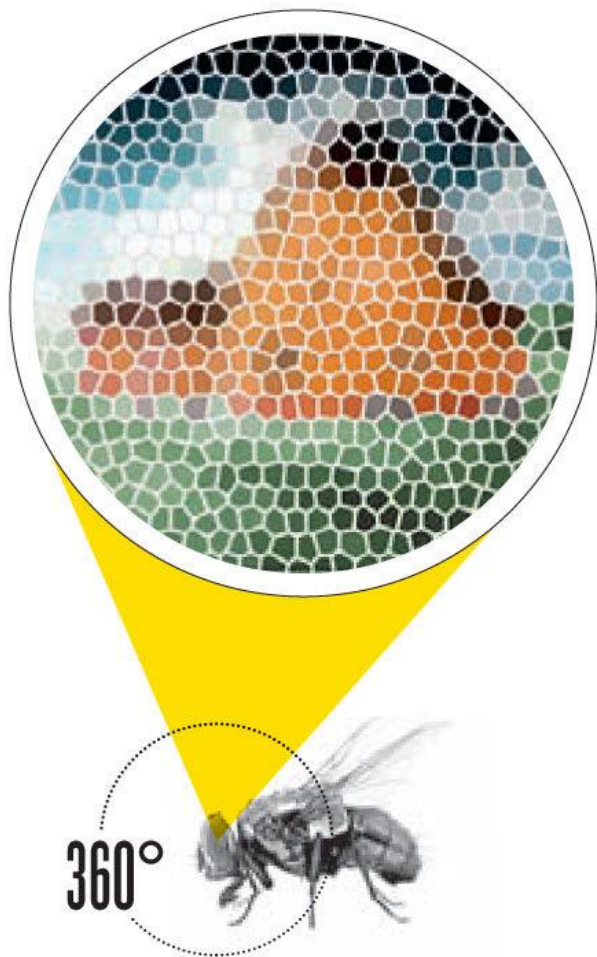


- ❖ **вырожденность** - одно и то же событие хранится в виде множественных неидентичных копий функциональной системы,
- ❖ **автоассоциативность** - разные копии одной и той же функциональной системы имеют связи с разнообразными другими системами за счет вырожденности набора входящих в эти копии нейронов,
- ❖ **реинтегративность** – целая система может быть извлечена из памяти по возбуждению небольшой части ее элементов,
- ❖ **репаративность** – система может восстанавливаться при повреждении части из ее элементов или даже части из ее копий.

## Бесмысленность цифрового моделирования «сложных систем»

- Сложные системы , например, крыло птицы, смысла эмулировать при наличии реактивных самолетов, превосходящих птиц по всем (почти) характеристикам.
- Нет смысла эмулировать мозг при наличии искусственных нейронных сетей, которые работают быстрее и точнее, в распознавании лиц
- Например, AI-планер-трансформер в стратосфере может летать бесконечно долго трансформируя энергию солнечного света с помощью солнечных батарей, расположенных в очень длинных крыльях.
- А вот крылья птиц не имеют такой прочности и и длины и, поэтому, летное качество их в целом ниже

## Качество "видеокамеры" мухи не всем понравится

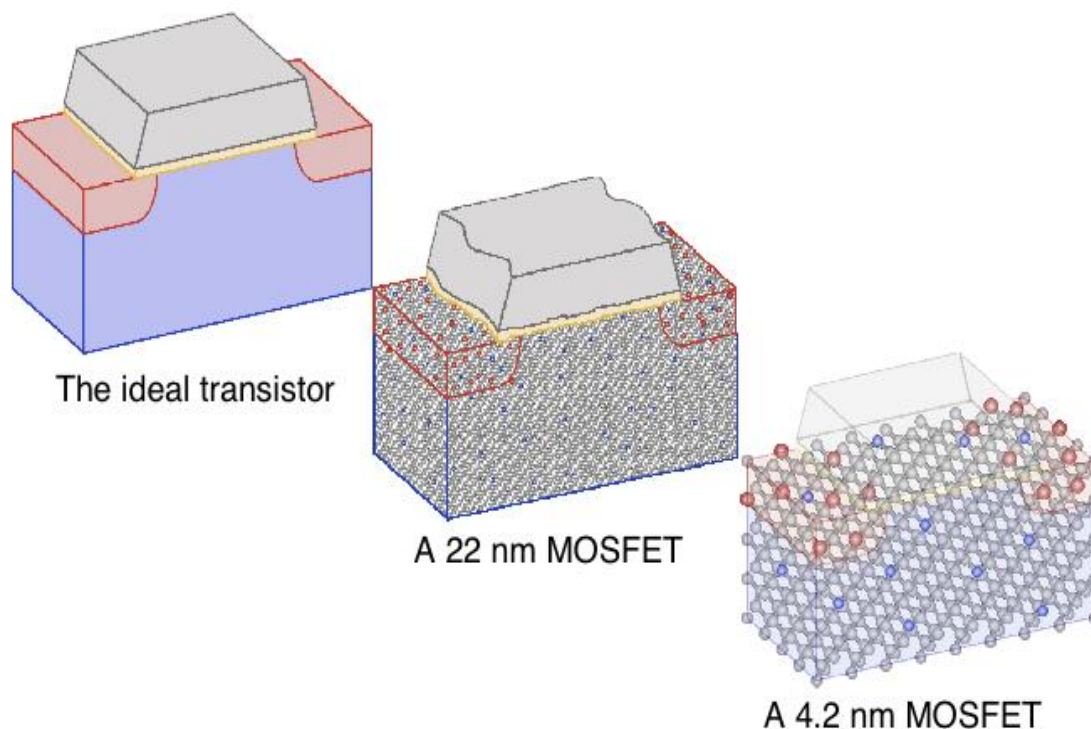


В утверждении "мозг — это компьютер" смысла столько же, сколько во фразах "руки — это экскаватор", "ноги — это автомобиль".

Компьютер — это инструмент для автоматизации некоторых элементарных мыслительных операций, которые отдельно от «контекста» и сознательно сам человек никогда не использует.



# Почему современный СК еще не «цифровой» мозг



законь физики микромира не само применимы .  
отказ 0.000000001% транзисторов в МП  
приводит к потере 10% и более  
производительности

*Уменьшение размеров транзистора приводит к ошибкам в его производстве*

Разрабатывается «максимально мощный чип» как некая повторяющаяся архитектура. Практически все элементы дублируются, как и в авиации.

Это сделано для того, чтоб в том случае, если в каком-то блоке выйдет из строя большое количество транзисторов из-за брака во время производства, этот блок можно отключить, а чип (МП) целиком остался в рабочем состоянии.

Так: процент брака кристаллов для 28-ядерных процессоров Intel Xeon составляет до 65 %. Если у конечного изделия не работает один из блоков памяти или одно из ядер не проходит тест, его **отключают**, а процессор

# Потребление электроэнергии и размер суперкомпьютеров

В современном мире, мобильный телефон обладает вычислительными способностями компьютера пятилетней давности и при этом работает от аккумулятора. Но, если мы сравним вычислительные способности суперкомпьютера IBM Summit, его размеры и потребляемые им объемы энергии с мозгом мыши, окажется, что он **неимоверно большой и очень неэффективный**.



*BM Power System AC922, IBM POWER9 22C  
3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox  
EDR InfiniBand, 2.41 million cores, 148.6 petaflops*  
Пиковая потребляемая мощность: 13 000 000 W  
Размеры: 4,608 узлов \* 0.2 м<sup>3</sup> = 920 м<sup>3</sup>

Мозг мыши способен обрабатывать более сложные задачи при потреблении всего **1-5 ватт**.  
**Итак, когда речь заходит о последовательностях, непрерывности, бесконечно малых или бесконечно больших значениях** цифровой СК  
...просто интерпретирует эти данные как последовательность дискретных кадров, обрабатывая их **как нечто статическое и конечное**.

# Мозг – это он/лайн само-программируемый вычислитель



А вот современные подходы машинного обучения лишены возможности обучаться непосредственно в процессе решения поставленной задачи.



# Многокомпонентная архитектура трансформации данных мониторинга в объяснимые оценки состояния самого объекта

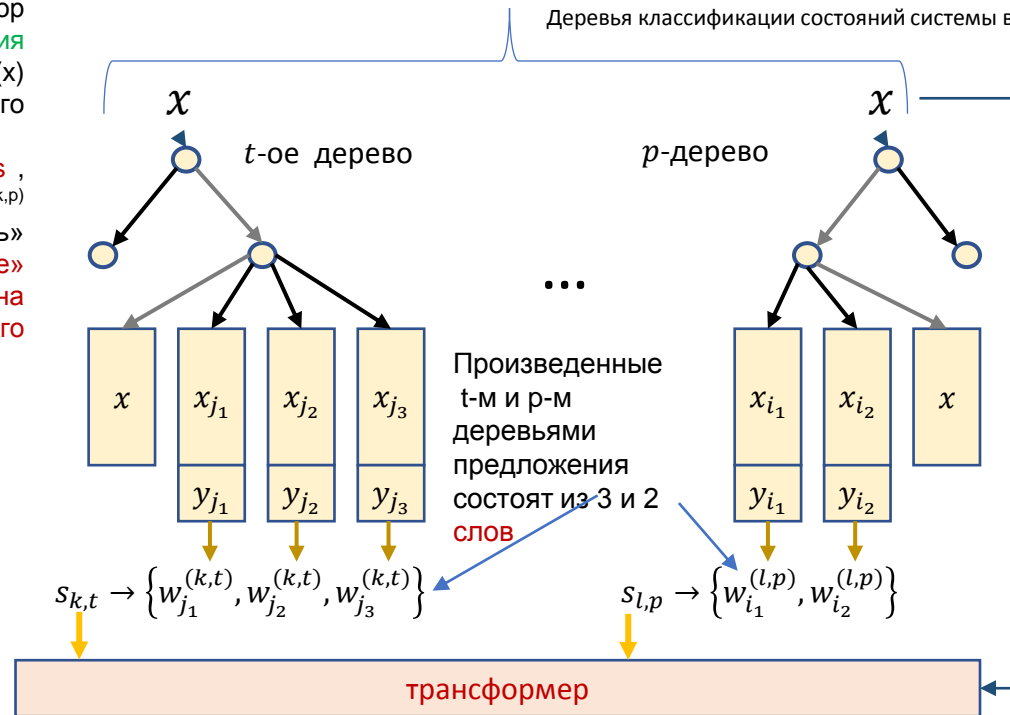
Имеется система описание которой задается текстом. Для системы сформирована **обучающая выборка**  $(x_j, y_j)$ , где  $x$  – вектор признаков (давление, частота..),  $y$  – **реакция организма/системы** и индексы  $j \in J_k(x)$  признаков  $x$  в один и тот же  $p$ -й лист  $k$ -го дерева классификации.  $(x_j, y_j)$ , представим в виде **предложения**  $s$ , состоящего из слов, обозначаемых как  $w_i^{(k,p)} \in J_k(x)$ , из которых можно «сформировать» актуальное **«лингвистическое описание»** текущего состояния в контексте (на основании) значения текущего входного сигнала  $x$

$$i \in J_k(x),$$

$$s_{k,p} = \{nw_{i_1}^{(k,p)}, \dots, w_{i_p}^{(k,p)}\}_{1r}$$

«предложение»  $s$ :

$k$  - номер дерева;  $p$  - номер листа;  $i$  - индекс примера (индекс слова в  $p$ -м предложении), который попадает в  $p$ -й лист.



- ИЗ КАЖДОГО «ЛИСТА» ДЕРЕВЬЕВ МОЖЕТ ВЫХОДИТЬ РАЗНОЕ КОЛИЧЕСТВО «ВЕТВЕЙ»:
- 1) из листа  $t$ -ого дерева выходит три «слова» описания реакции системы, а из листа  $p$ -дерева выходит описание, состоящее из двух «слов»
  - 2) На вход системы подается текущий вектор « $x$ » и соответствующие ему значения выхода системы

## Задача системы

вычислить предиктивную оценку  $\hat{y}$  по значениям входа  $[x]$



# Выводы

В настоящее время компьютеры и системы на базе ИНС не реализуют все известные особенности функционирования мозга. Современные ИНС построены на **выявлении корреляций между входными данными** при их фиксированном распределении и соответствующих состояний нейронов. Но эти корреляции не отражают причинно-следственную связь (**супервентность**) между паттернами данных и состоянием ИНС.

Сформированные **на основе корреляций объяснения** не выявляют причин, которые вызывает наблюдаемые на «механистическом уровне» процессы, что может привести к неправильным обобщениям и искаженному пониманию происходящего. Предсказать редкое или нетипичное поведение модели возможно, но для этого **понадобится объединение корреляционных, информационных и механистических (физических) моделей.**

Почему же мозг – супер вычислитель, а современный суперкомпьютер не цифровой мозг? Это потому, что «Машина Тьюринга» как формальная алгоритмическая система согласно теоремам Геделя - не самоприменима»