



**ПОЛИТЕХ**

Санкт-Петербургский  
политехнический университет  
Петра Великого

COMPUTO ERGO SUM

КУРС: РЕШЕНИЕ ПРИКЛАДНЫХ ЗАДАЧ МЕТОДАМИ МАШИННОГО ОБУЧЕНИЯ

## ЛЕКЦИЯ 7

*Интерпретация результатов моделирования  
с использованием методов машинного обучения...*

26 октября/2 ноября  
2022

**Cognito ergo sum**

Р. Декарт

В общем случае задача компьютерного моделирования это **прямая задача отображения** «цифровые **данные** – **цифровые данные**». Такое отображение основано на моделях научной картине мира, которая возникает в результате процесса **факторизации** (разделения) непрерывной физической реальности, данной субъекту через множество данных, на конечное множество понятий, т.е. отображение «**данные-субъект-понятия**».

**Задача машинного обучения** есть **моделирование процесса факторизации** средствами компьютерных технологий, Эта задача в общем случае **не имеет единственно решения и не непрерывно** зависит от входных данных (фактически зависит и от данных, которые в явном виде в условиях задачи отсутствуют ) **Этапы решения задач машинного обучения:**

**понимание** задачи и анализ имеющихся данных

«**изобретение**» признаков в данных, которые супервентны предметным понятиям

**построение** модели «данные-понятия» сведение обучения к оптимизации

оценивание качества

«**эксплуатация**» модели на различных наборах входных данных

# ОСНОВНОЙ ТЕЗИС КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ

ИСТИННО ЛИШЬ ТО, ЧТО МОЖНО

- вычислить на компьютере (? теоремы Геделя)  
а полученный результат можно
- понять и объяснить (интерпретировать)

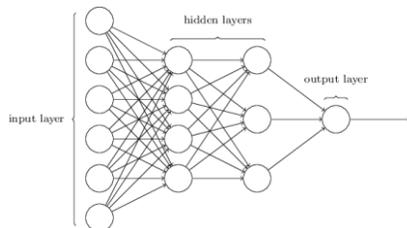
1) **Вычислимость** – описание решения **«прямых»** задач с помощью программ-алгоритмов, реализующих счетный набор частично рекурсивных операций ?

2) **Интеллектуальность** - решение **«обратных»** задач путем построения АЛГОРИТМОВ вычислений и объяснения (интерпретации) результатов

**Интерпретируемость (объяснение)** результатов - это степень уверенности ... в том, что вычисленный результат можно **понять**, а используемый алгоритм вычислений **уточнить** так, чтобы повысить степень уверенности и снизить «коэффициент незнания»

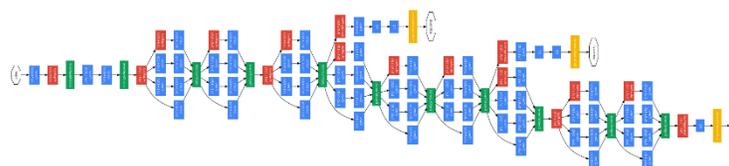


Нужны такие **эффективные** (точные) модели, которые обладают свойством **самообъяснения** результатов, что повышает **уровень доверия к ним**



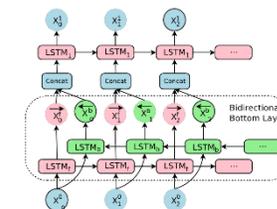
## Многоуровневый перцептрон

- **классификация**
- функциональная аппроксимация
- автокодер



## Сверточные нейронные сети

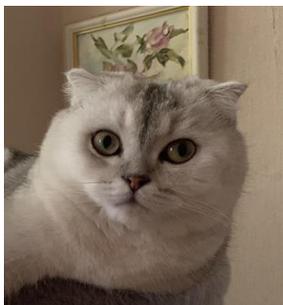
- выделение свойств
- **детектирование объектов**
- сегментация изображений



## Рекуррентные нейронные сети

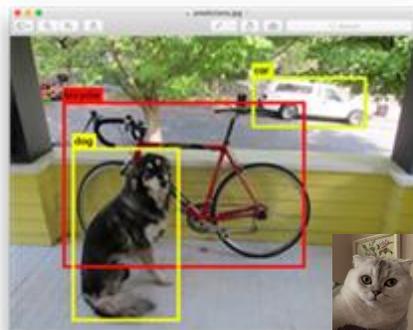
- Сегментация временных потоков
- Преобразования данных в текст
- **Перевод текстов**

## классификация



“КОТ”

## Обнаружение объектов

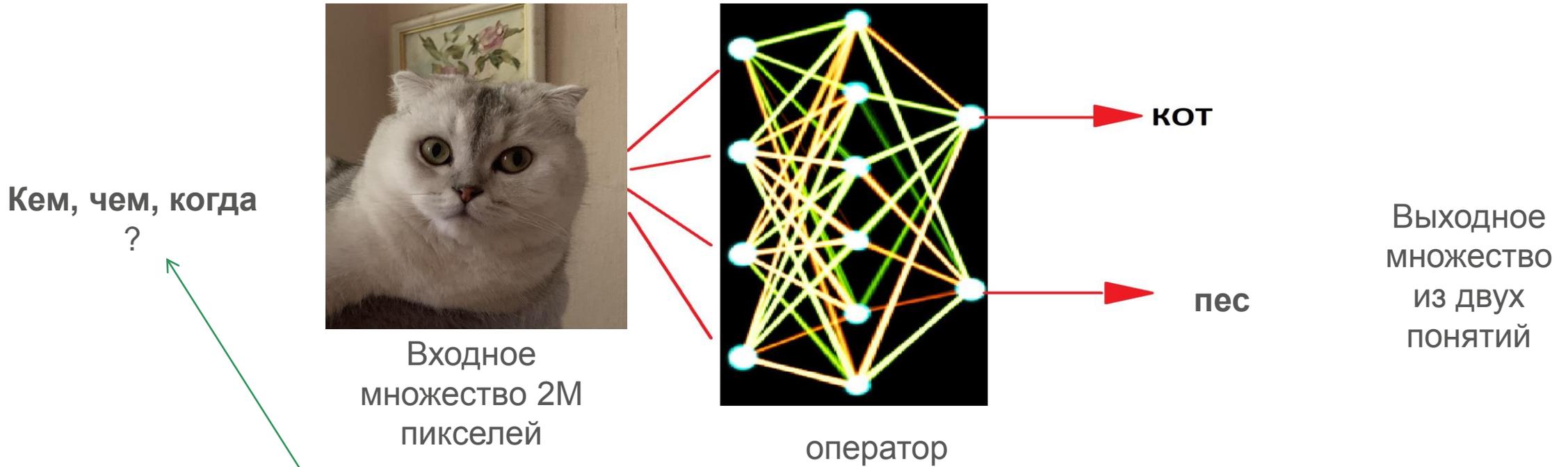


## Сегментация данных





# ИТАК: МОДЕЛЬ МО С ТОЧКИ ЗРЕНИЯ РАЗРАБОТЧИКА

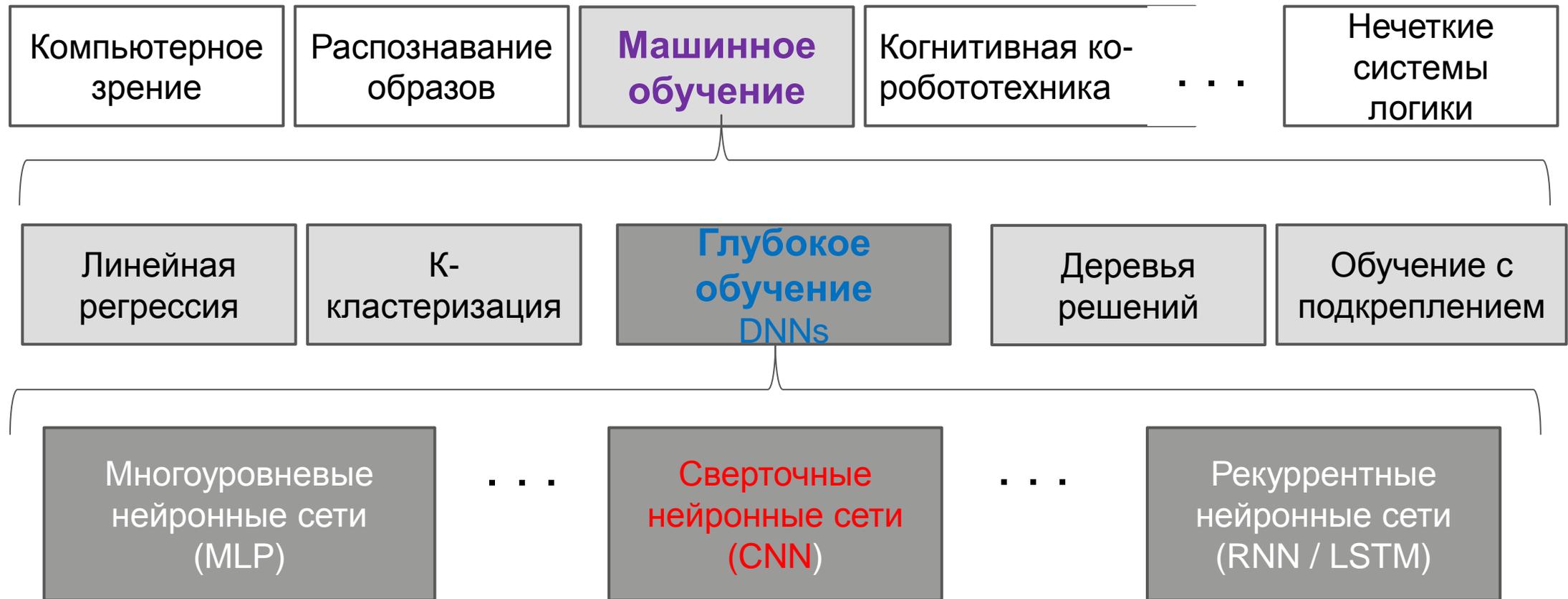


"заранее **обученная** модель **предсказывает**, что **2 М пикселей** входного изображения **кодируют** с помощью hidden **сигнатуры** в выходном embedding vector позицию «кот» с вероятностью **0.98**", а слово «пес» с вероятностью **2%**.



# ПОДХОДЫ К РЕШЕНИЮ ЗАДАЧИ МАШИННОГО ОБУЧЕНИЯ

Технологии: Используя GPU (векторные) платформы отдельные компоненты программ можно распределить между сотнями (или тысячам) идентичных процессоров, и, таким образом, **время обучения можно сократить**



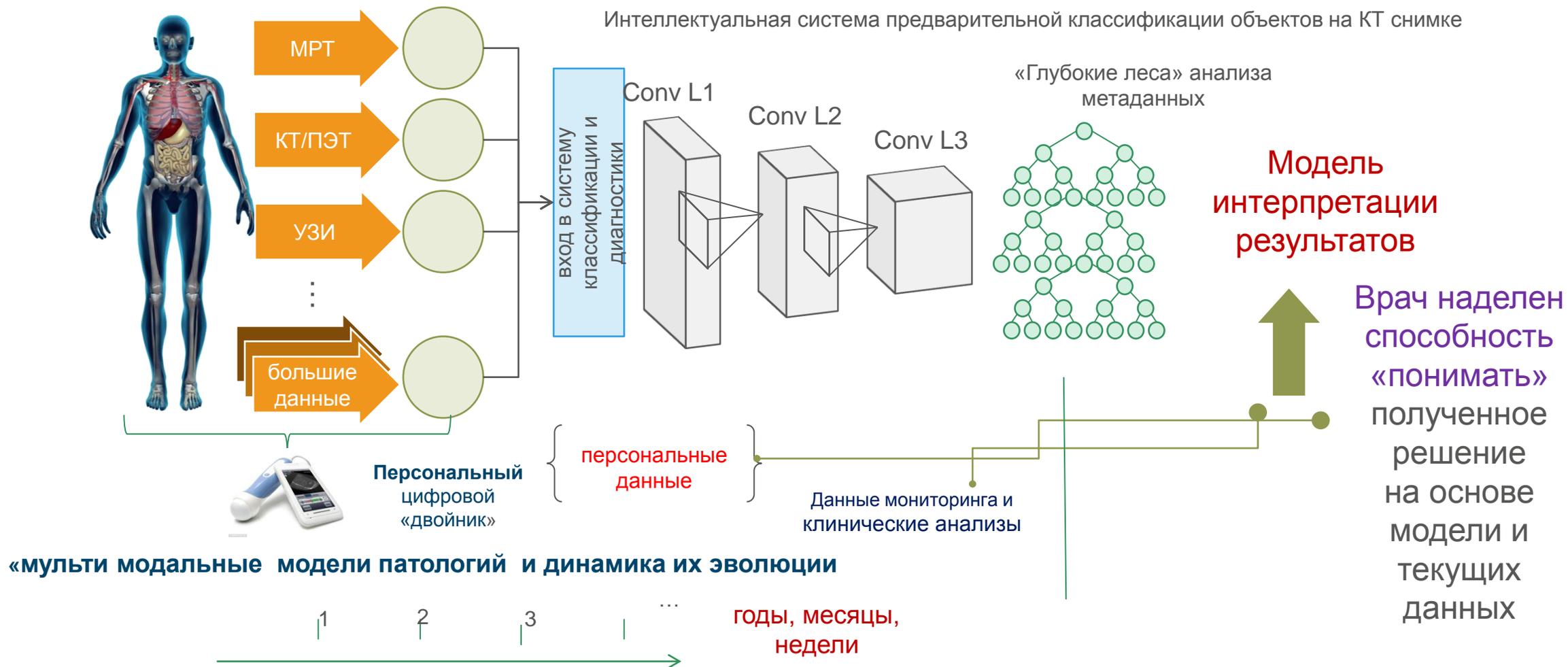
Проблемы: трудоемкость обучения высокая, а робастность результатов низкая

# ПРИМЕР: АЛГОРИТМИЧЕСКАЯ МОДЕЛЬ + ТЕКУЩИЕ ДАННЫЕ

Потоки первичных данных, характеризующих состояние биологических тканей и процессов

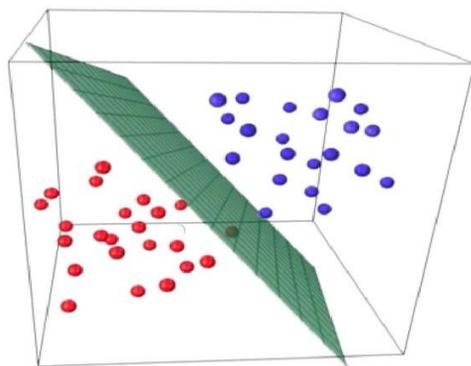
центр хранения, анализа и предсказательной диагностики

Интеллектуальная система предварительной классификации объектов на КТ снимке

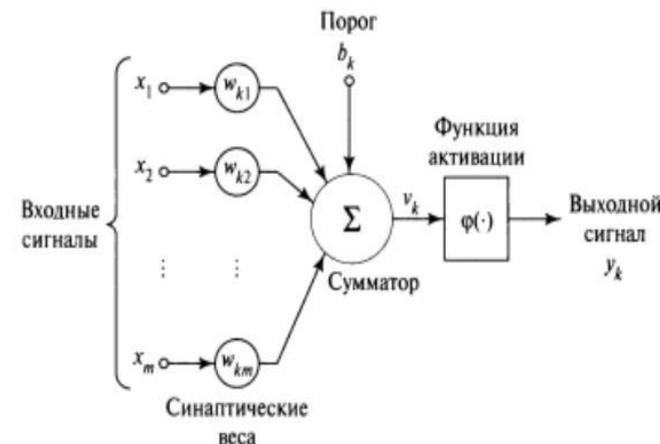


- Основа сверточных ИНС - искусственный нейрон – линейный классификатор, к котором входные элементы напрямую связаны с выходными с помощью системы весов и выполняют функцию линейной классификации (**однослойный нейрон не может** моделировать например «исключающее ИЛИ ( XOR – сумма по модулю 2)

Линейная модель классификации

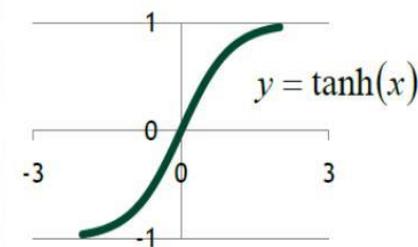
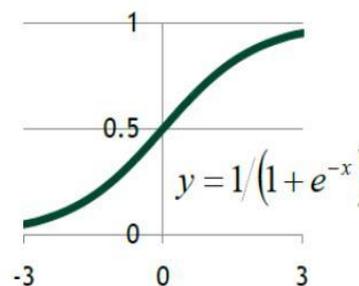


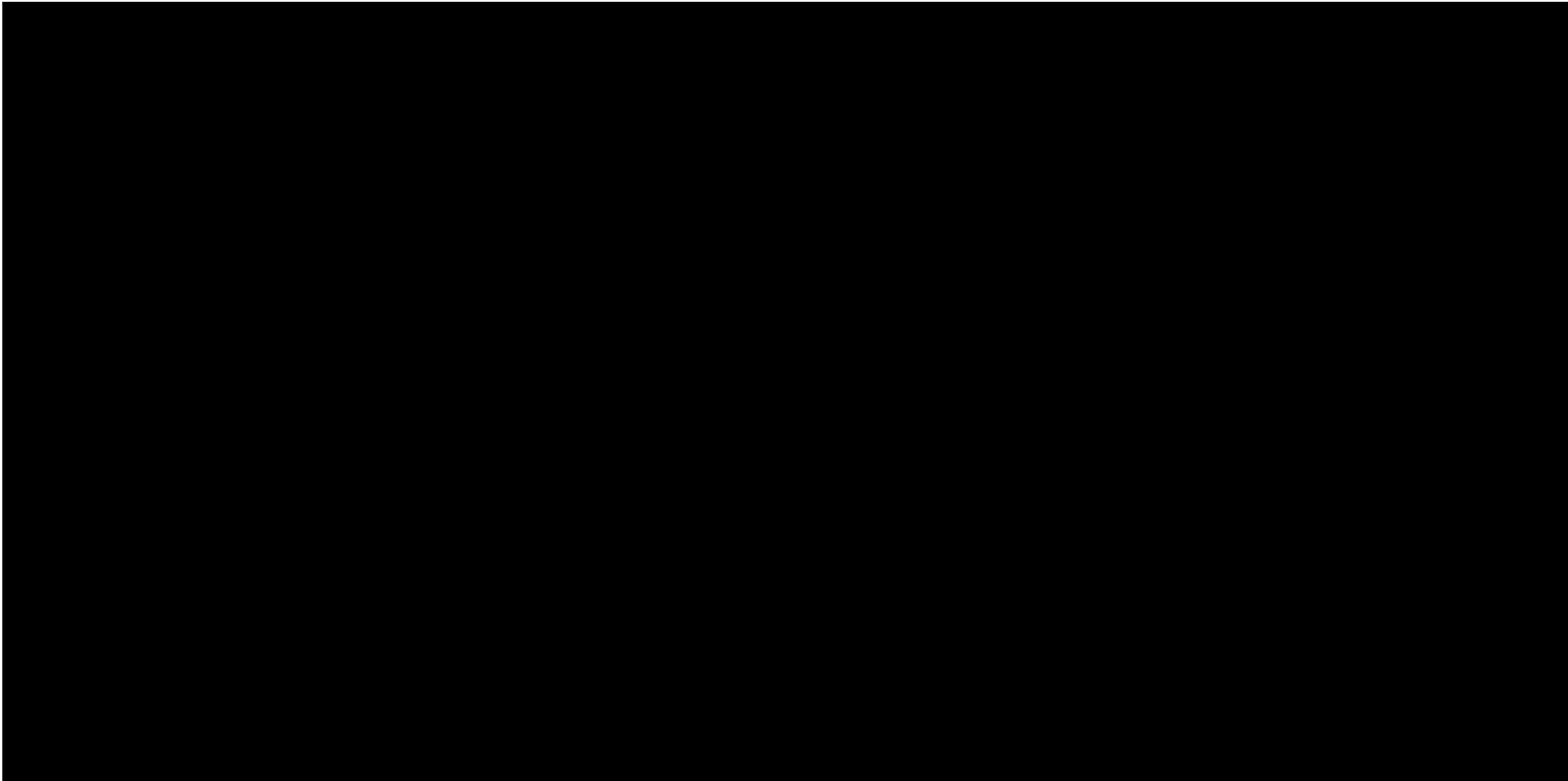
## Модель искусственного нейрона



$$u_k = \sum_{j=1}^m w_{kj} x_j,$$

$$y_k = \varphi(u_k + b_k)$$





задача обучения ГЛУБОКОЙ ИНС распознавать 3D изображения компьютерной томографии (КТ)



СК: 1Пфлопс  
=  $1 \times 10^{15}$  Флопс

Для обучения н/с с точностью классификации **90%** надо:

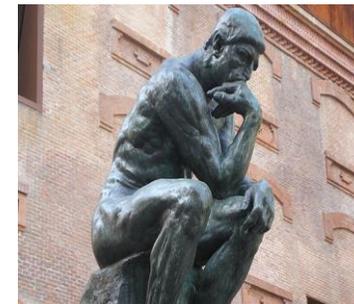
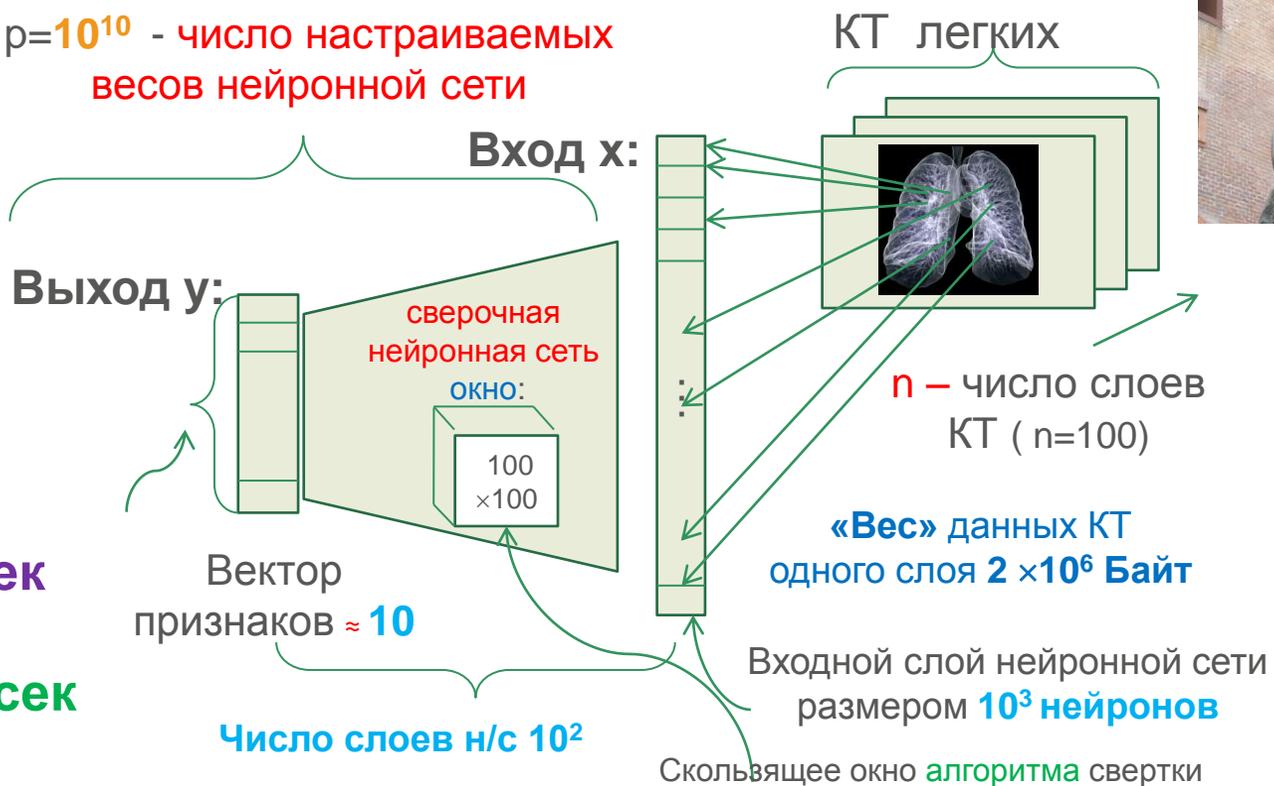
- Число операций  $\approx 10^{17}$
- Время обучения ИНС на СК  $\approx 10^2$  сек
- t обучения на ПК (200 ГФлопс)  $\approx 2 \times 10^6$  сек

**Градиентный алгоритм** обучения **глубокой сверточной нейронной сети**:

- функция ошибки  $F = \|y^* - y\|^2$ ,  $y^*$  - эталонный вектор признаков
- Алгоритм использует  $10^{10}$  частных производных  $F$  по всем настраиваемым параметрам;
- Число операций численного дифференцирования на одну итерацию  $Q = 10^{15}$

Обучающая выборка  **$10^4$  КТ снимков**

$p = 10^{10}$  - число настраиваемых весов нейронной сети



обучение ИНС даже среднего размера может занимать **недели** или **месяцы**

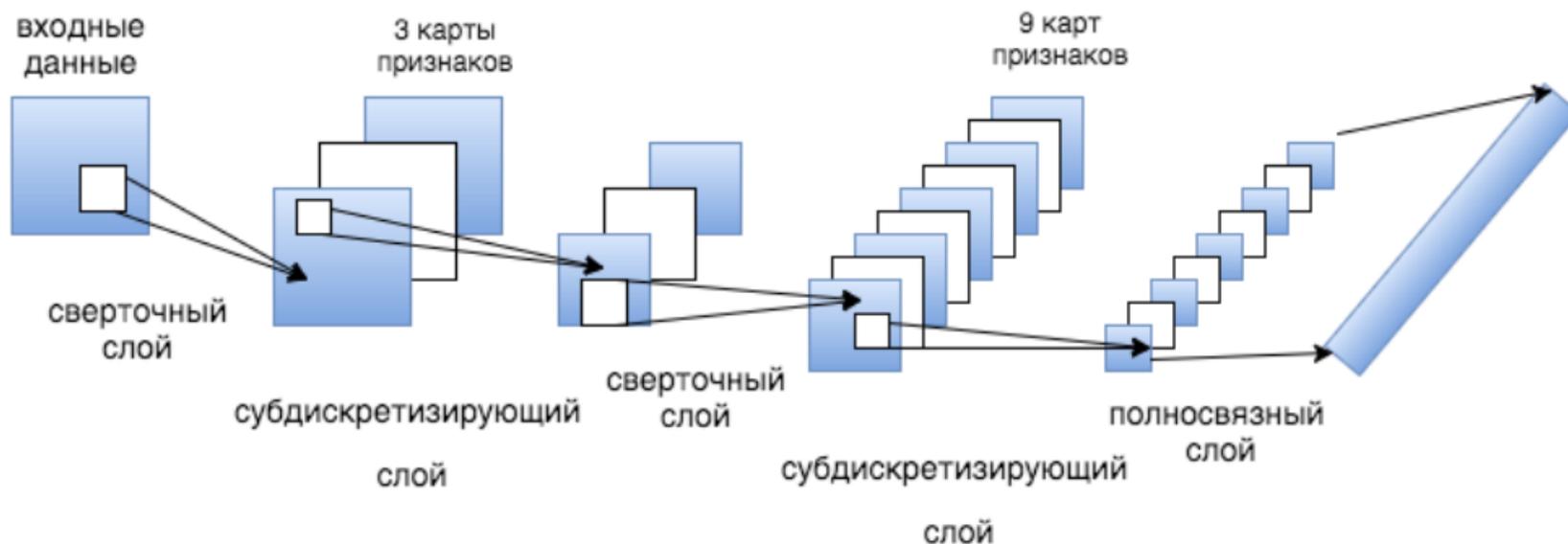
Сверточная н/с состоит из 3-х различных слоев:

- Сверточный слой, к которой каждый нейрон выполняет свертку (операцию конволюции) некоторой области предыдущего слоя, которая задается множеством нейронов, выход которых связан со входом данного нейрона.
- субдискретизационный слой, нейроны которого выполняют функции уменьшения пространства карт признаков
- полносвязные слои (обычно выходной слой).

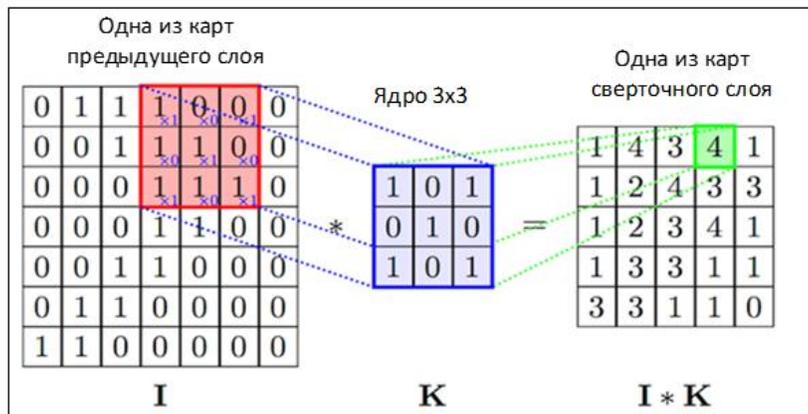
Ключевым моментом является наличие так называемых «разделяемых» весов – которые состоят из нейронов с одинаковыми весовыми коэффициентами.. Такие нейроны объединяются в «карты признаков» - feature maps.

При этом все три вида слоев могут чередоваться в произвольном порядке, что позволяет составлять карты признаков из карт признаков, а это на практике означает способность распознавания сложных иерархий признаков

Идея механизма «свёрточной нейронной сетм (*convolutional neural network*, CNN) основана на чередовании сверточных слоев - англ. *convolution layers* и субдискретизирующих слоев - *subsampling layers*, уменьшающих пространство признаков ( CNN - прямая аналогия работы зрительной коры головного мозга, где используются «простые клетки» – обрабатывающие прямые линии, пересекающиеся под разными углами, а также «сложные клетки», которые реагируют на определенный набор простых клеток).

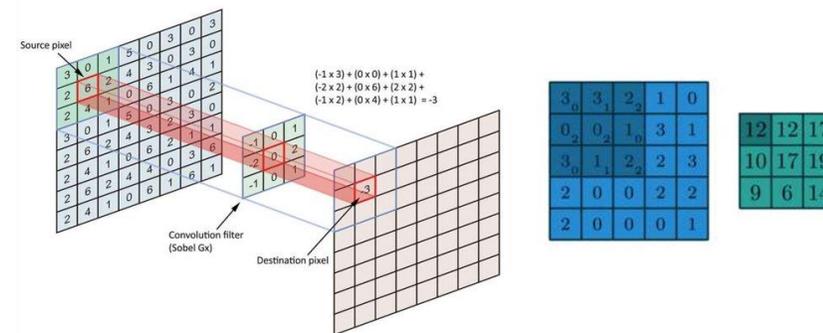


# ОПЕРАЦИЯ СВЕРТКИ И ПОЛУЧЕНИЕ ЗНАЧЕНИЙ КАРТЫ ПРИЗНАКОВ

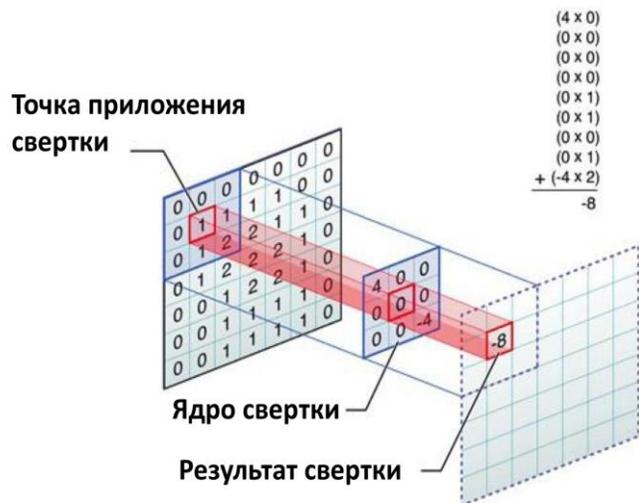


$$4 = 1 + 1 + 1 + 1$$

## Двумерная свертка



## Свертка изображения



Исходные данные

1	0	2	3	1
2	2	4	1	3
0	2	1	1	5
1	3	0	2	6
1	1	1	3	3
0	1	2	3	0
5	2	3	5	2

$$1 + 4 + 2 = 7$$

Ядро свертки

1	0	1
0	2	0
3	0	2

7	19	18
13	18	21
12	12	19
7	16	18
25	24	23

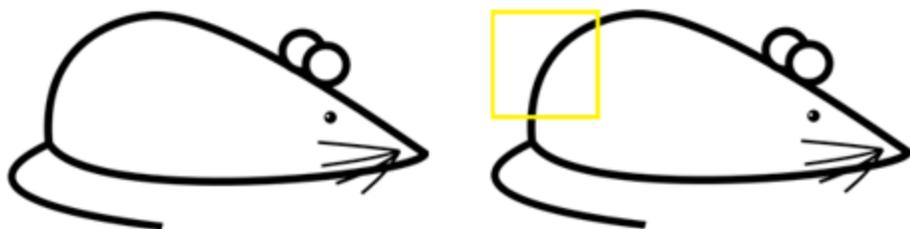
Полученные данные

## Пример ядра с обученным признаком

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0



## Входное изображение



В этом месте, где желтое окно, будет большой отклик (сигнал), что говорит о наличии этого признака на изображении.

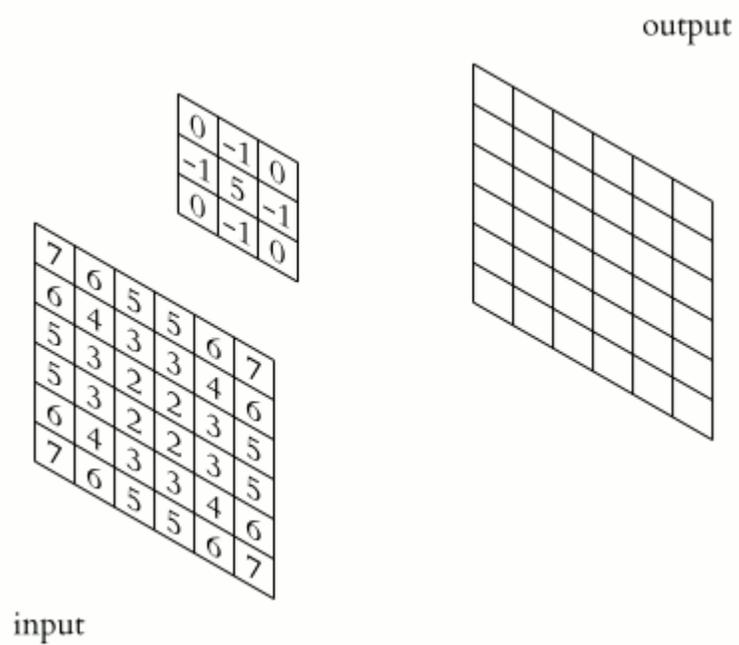
Изначально значения каждой карты сверточного слоя равны 0. **Значения весов ядер** задаются случайным образом в области от -0.5 до 0.5. Ядро скользит по предыдущей карте и производит операцию свертка, которая часто используется для обработки изображений, формула:

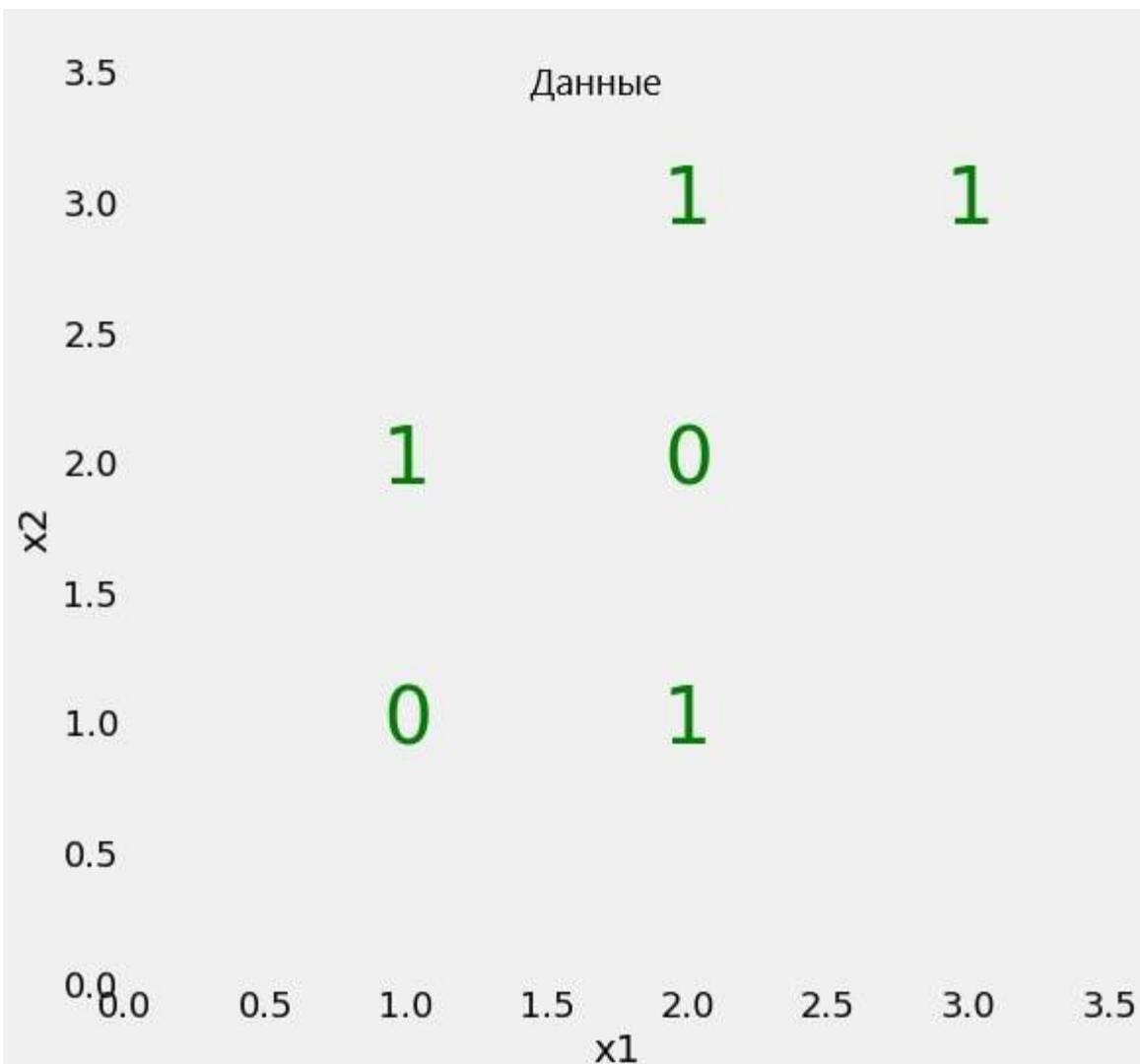
$$(f * g)[m, n] = \sum_{k, l} f[m - k, n - l] * g[k, l],$$

где  $f$  – исходная матрица изображения;  
 $g$  – ядро свертки.

Неформально эту операцию можно описать следующим образом — **окном размера ядра  $g$  проходим с заданным шагом** (обычно 1) все изображение  $f$ , на каждом шаге **поэлементно умножаем содержимое окна на ядро  $g$** , результат суммируется и записывается в матрицу результата

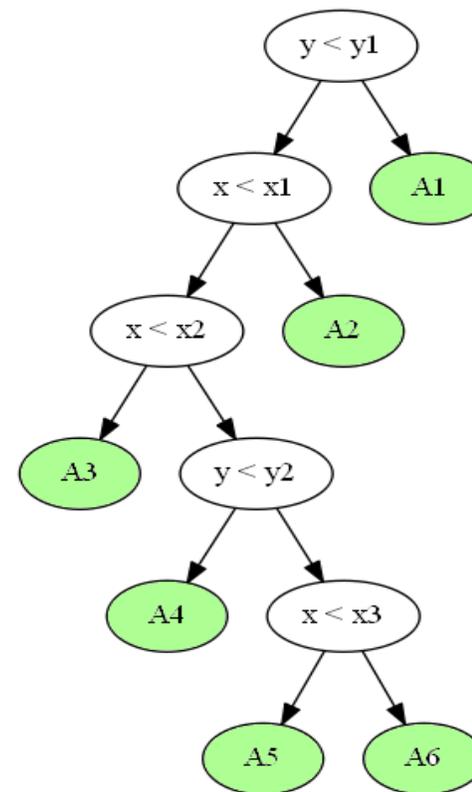
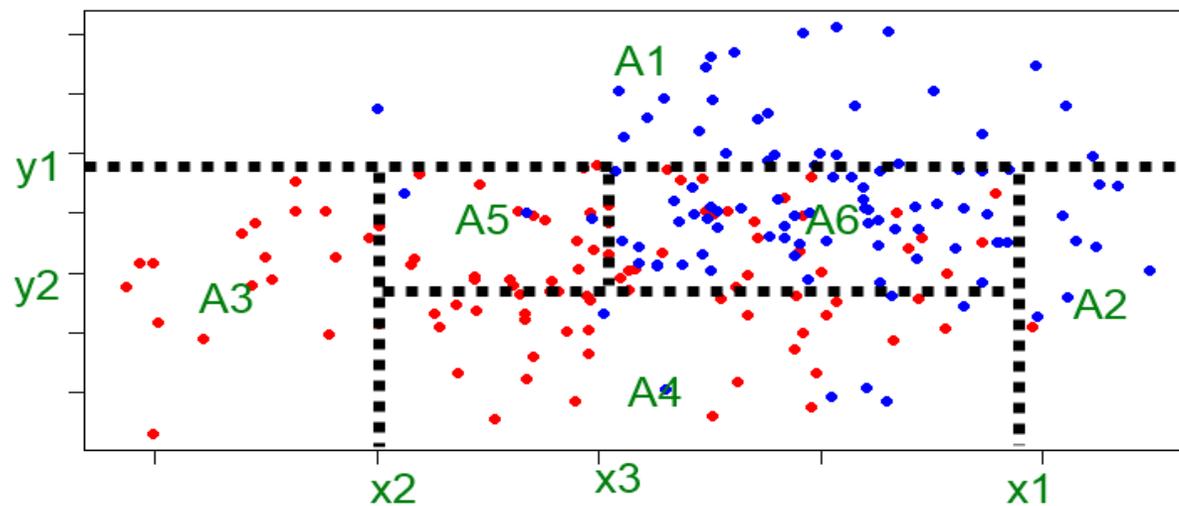
# ПРИМЕР ВЫЧИСЛЕНИЯ КАРТЫ В «ДИНАМИКЕ»





набор данных имеет всего два параметра (две заданные переменные),  $x_1$  и  $x_2$ , а также 6 образцов, несущих эти параметры. Образцы разделены метками на два класса. Хотя это простая задача, линейно классы разделить невозможно. Это означает, нет возможности нарисовать на предложенной плоскости **прямую линию, которая отделит один класс от другого.**

По сути дерево решений — нелинейная модель, создаваемая с помощью множества линейных ограничителей.



## Пример дерева решений



Дерево решений — интуитивно понятная базовая единица алгоритма случайный лес. Мы можем рассматривать его как серию вопросов да/нет о входных данных

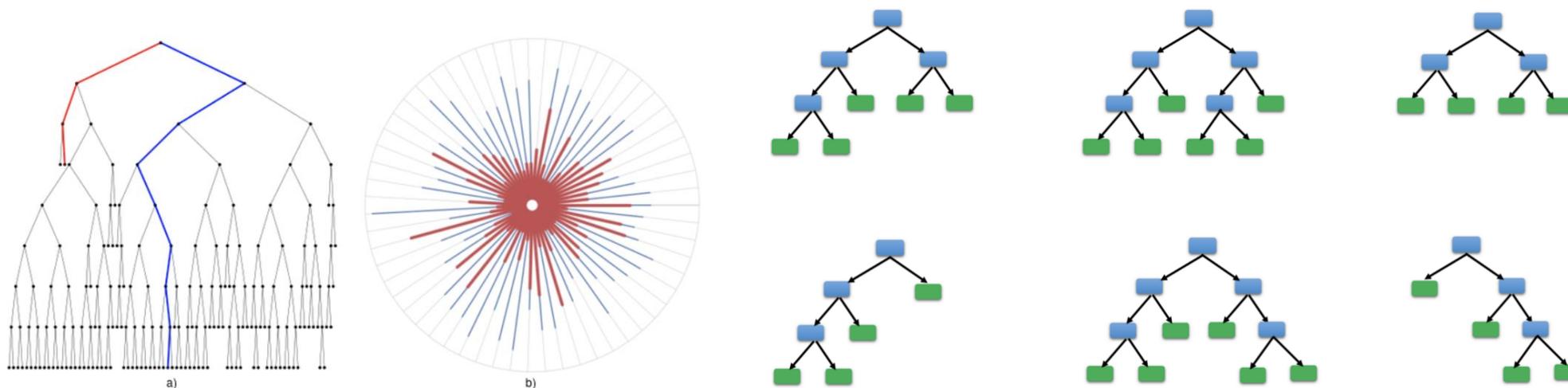
Корень дерева: «солнечно ?»

Внутренний узел дерева:

- температура воздуха высокая ?
- идет ли дождь

Лист — конечный узел дерева (узел решения) - играть , не играть.

Точность с использованием деревьев решений определяется «глубиной»



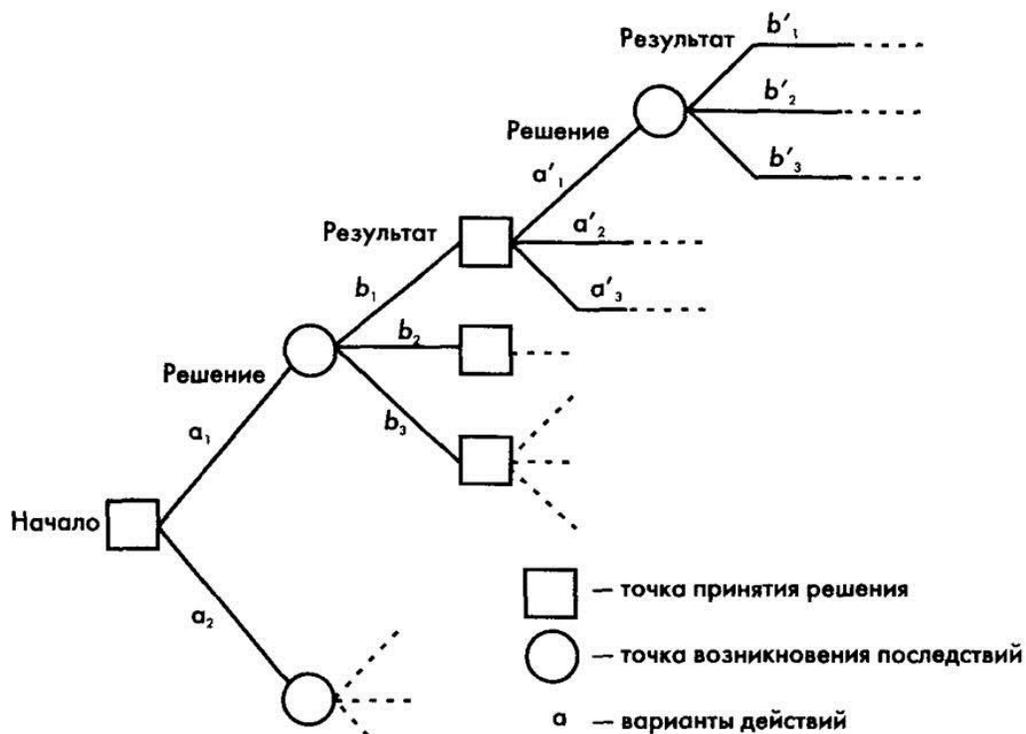
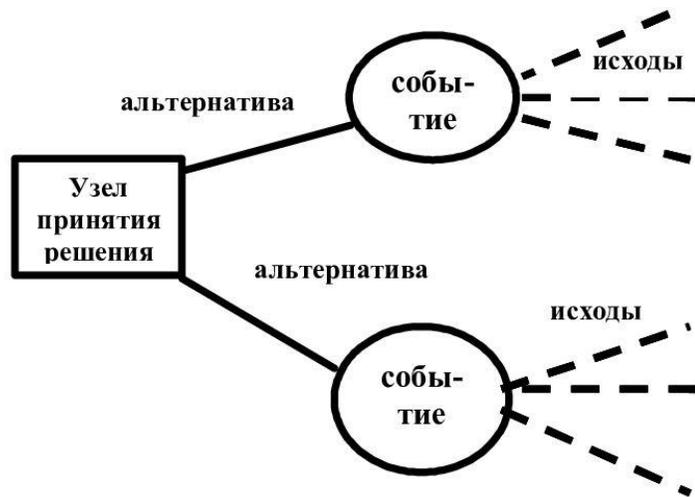
В качестве альтернативы ограничению глубины, которое ведёт к уменьшению вариативности (хорошо) и увеличению погрешности (плохо), можно собрать множество деревьев в единую модель или просто «случайный лес». Случайный лес — модель, состоящая из множества деревьев решений. Вместо того, чтобы просто усреднять прогнозы разных деревьев (такая концепция называется просто «лес»), эта модель использует две ключевые концепции, которые и делают этот лес случайным.

# «ДЕРЕВО РЕШЕНИЙ» В ЗАДАЧЕ МАШИННОГО ОБУЧЕНИЯ

Цель машинного обучения состоит в том, чтобы научить алгоритм обобщать полученную информацию и верно обрабатывать новые, ранее не встречавшиеся данные.

Если алгоритм обучения не ограничивает **глубину дерева (количество слоёв)**, то происходит **переобучение (модель с высокой «вместимостью»)**, которая запоминает обучающий набор данных, Проблема в том, что такая модель выявляет не только закономерности в данных, но и любой присутствующий в них шум В результате структура дерева решений будут значительно варьироваться в зависимости от обучающего набора данных.

5

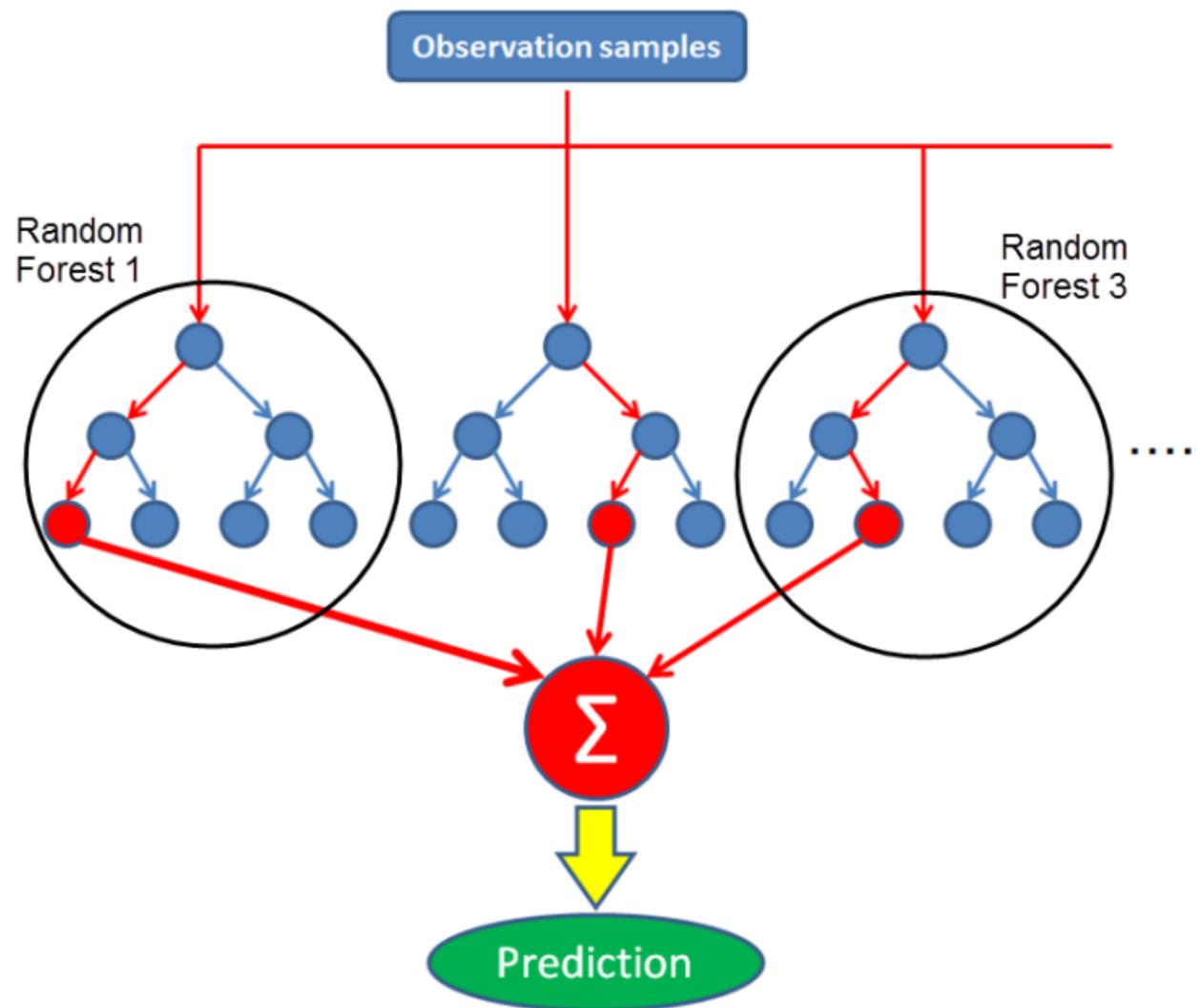


Метод машинного обучения, где несколько моделей обучаются для решения одной и той же проблемы и объединяются для получения лучших результатов называется ансамблевым методом. Основная предпосылка заключается в том, что результат работы нескольких моделей будет более точен, чем результат только одной модели.

**Наиболее популярными ансамблевыми методами являются:**

- Используется несколько **разнородных «слабых» моделей**. Их обучают и объединяют для построения прогноза, основанного на результатах различных слабых моделей. **(Стекинг)**.
- В этом случае однородные модели **обучают на разных наборах данных** и объединяют. Получают прогноз путём усреднения. Если использовать в качестве слабого ученика дерева решений, то получится случайный **(Бэггинг)**. лес [RandomForestClassifier](#) / [RandomForestRegressor](#).
- . При использовании данного метода несколько однородных моделей последовательно обучаются, **исправляя ошибки друг друга. (Бустинг)**

это мета-оценщик, который соответствует ряду деревьев решений. классификаторы на различных подвыборках набора данных и использует усреднение в повысить точность прогнозирования и контроль переобучения



Базовая концепция **случайного леса** заключается в использовании определённой выборки параметров образца для разделения каждого узла в каждом отдельном дереве. Размер выборки равен квадратному корню из общего числа параметров. То есть, если каждый образец набора данных содержит 16 параметров, то в каждом отдельном узле будет использовано 4.

Случайный лес может сочесть **сотни или тысячи деревьев принятия решений**, обучая каждое на **отдельной выборке данных**, разделяя узлы в каждом дереве с использованием ограниченного набора параметров. Итоговый прогноз делается **путём усреднения прогнозов** от всех деревьев.

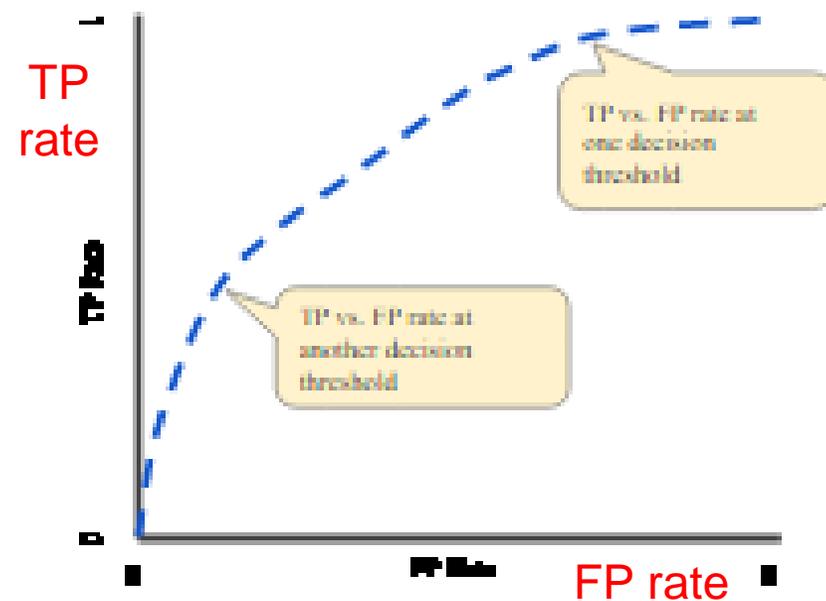
При использовании случайного леса, каждому дереву можно предоставить доступ только к определённым подмножествам обучающих данным, чтобы эффекты шумов были нейтрализованы разнообразием выборки.

**Интуитивное правило обучения: никогда не доверяйте единственному источнику данных**

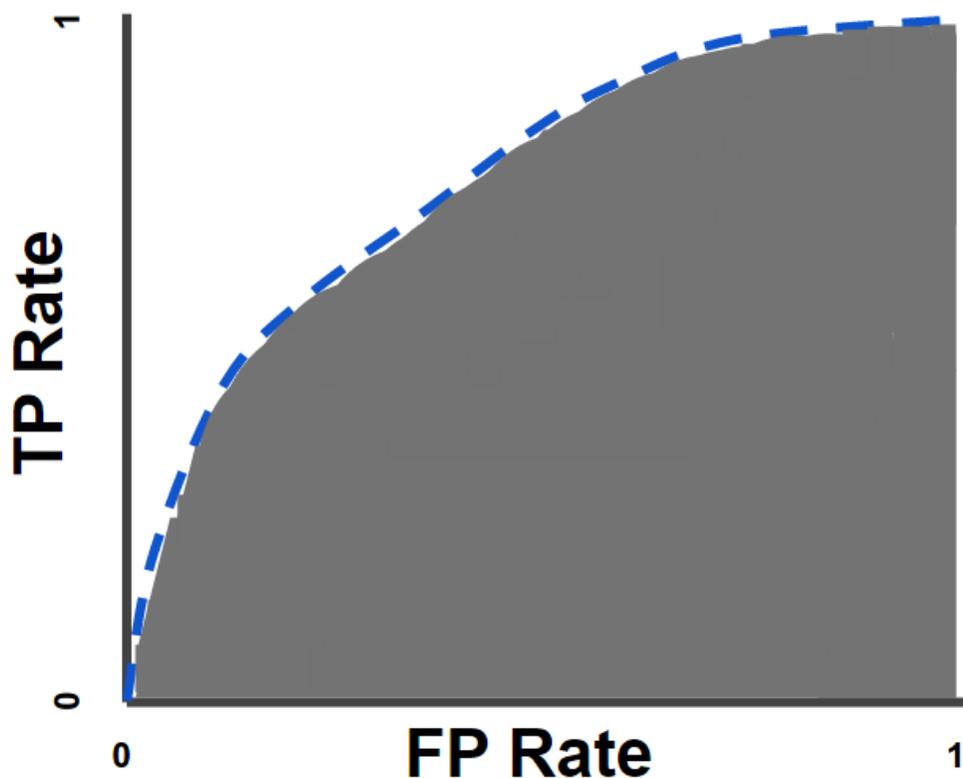
Т.Н. кривая рабочих характеристик (ROC) представляет собой график, показывающий эффективность модели классификации при всех пороговых значениях классификации. Эта кривая отображает два параметра:

- Истинная положительная скорость или True Positive Rate ( TPR ) является синонимом отзыва и поэтому определяется следующим образом  $TPR = TP / ( TP + FN )$
- Ложноположительный показатель ( FPR ) определяется следующим образом:  $FPR = FP / ( FP + TN )$

Кривая ROC отображает соотношение TPR и FPR при различных порогах классификации. Снижение порога классификации позволяет **классифицировать больше элементов как положительные**, тем самым увеличивая количество ложных срабатываний и истинных срабатываний.



Коэффициент TP и FP при различных порогах классификации.



AUC обеспечивает совокупный показатель производительности по всем возможным пороговым значениям классификации. Один из способов интерпретации AUC — это вероятность того, что модель ранжирует случайный положительный пример выше, чем случайный отрицательный пример/ AUC представляет собой вероятность того, что случайный положительный (зеленый) пример расположен справа от случайного отрицательного (красный) примера. Значение AUC варьируется от 0 до 1. **Модель, чьи прогнозы на 100 % неверны, имеет значение AUC, равное 0,0;** тот, чьи предсказания верны на 100%, имеет AUC 1,0.

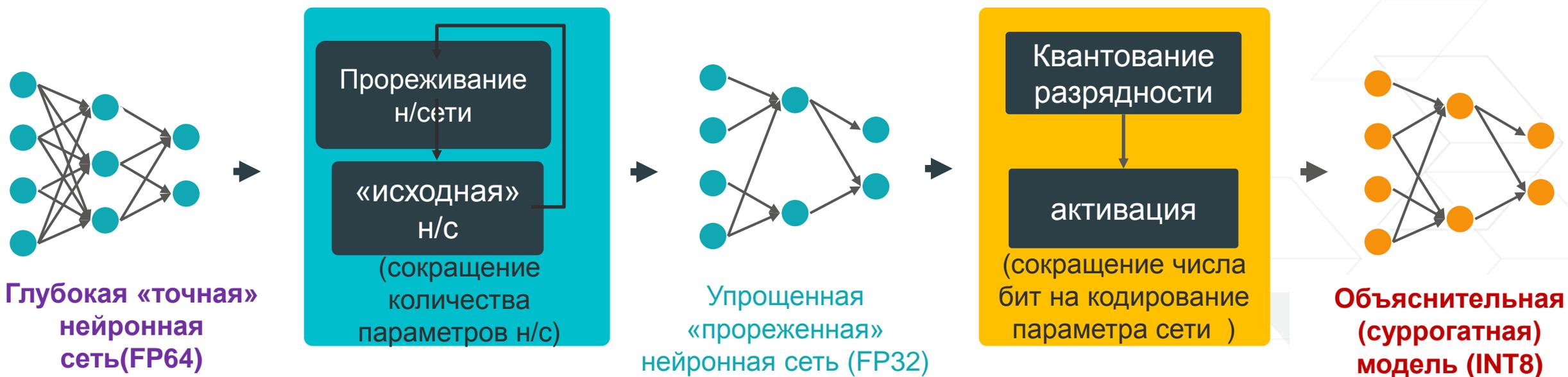
## AUC

- не зависит от масштаба . Оценивает **насколько хорошо ранжируются прогнозы**, а не их абсолютные значения.
- не зависит от порога классификации . Измеряет **качество прогнозов модели независимо от выбранного порога классификации**.



ПОЛИТЕХ

# ТОЧНОСТЬ ПРОГНОЗА VS ИНТЕРПРЕТИРУЕМОСТЬ РЕЗУЛЬТАТА



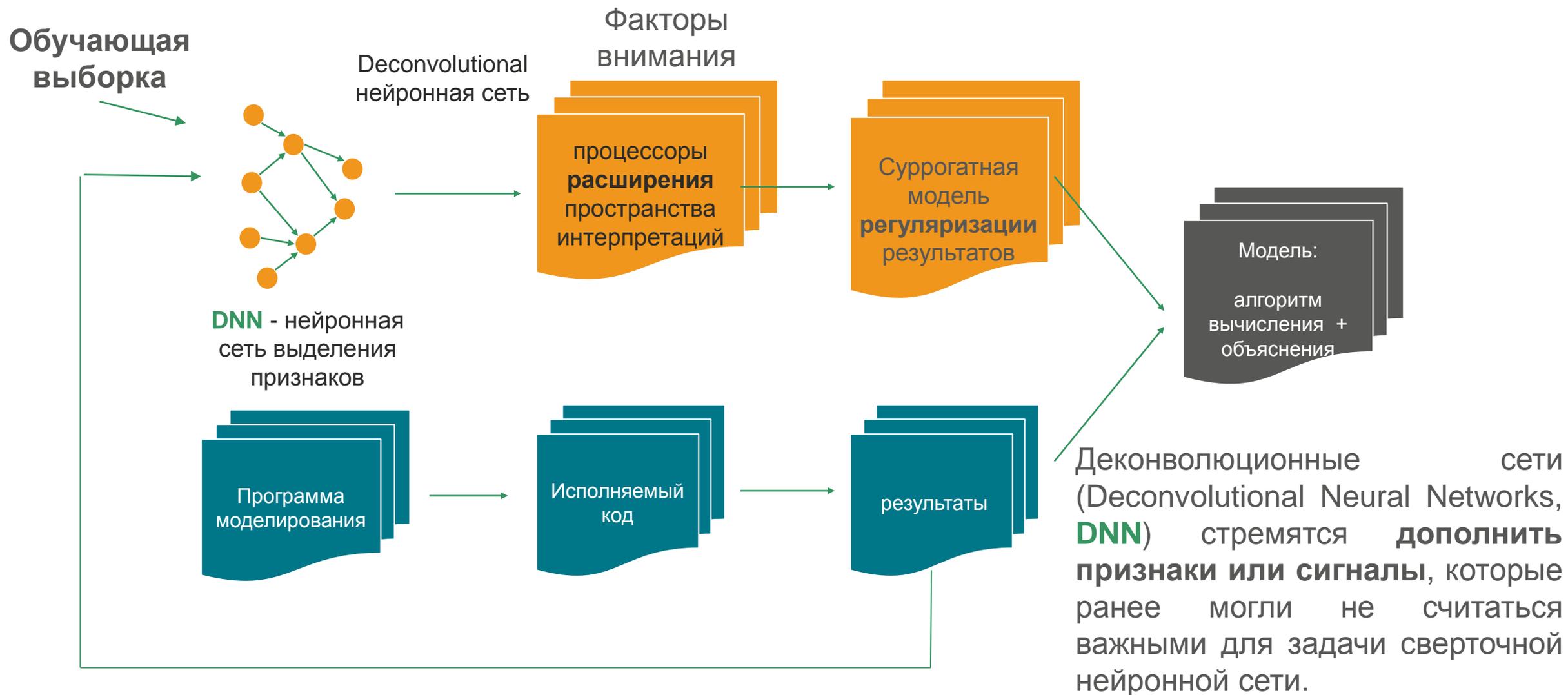
Процесс включает два инструментальных этапа

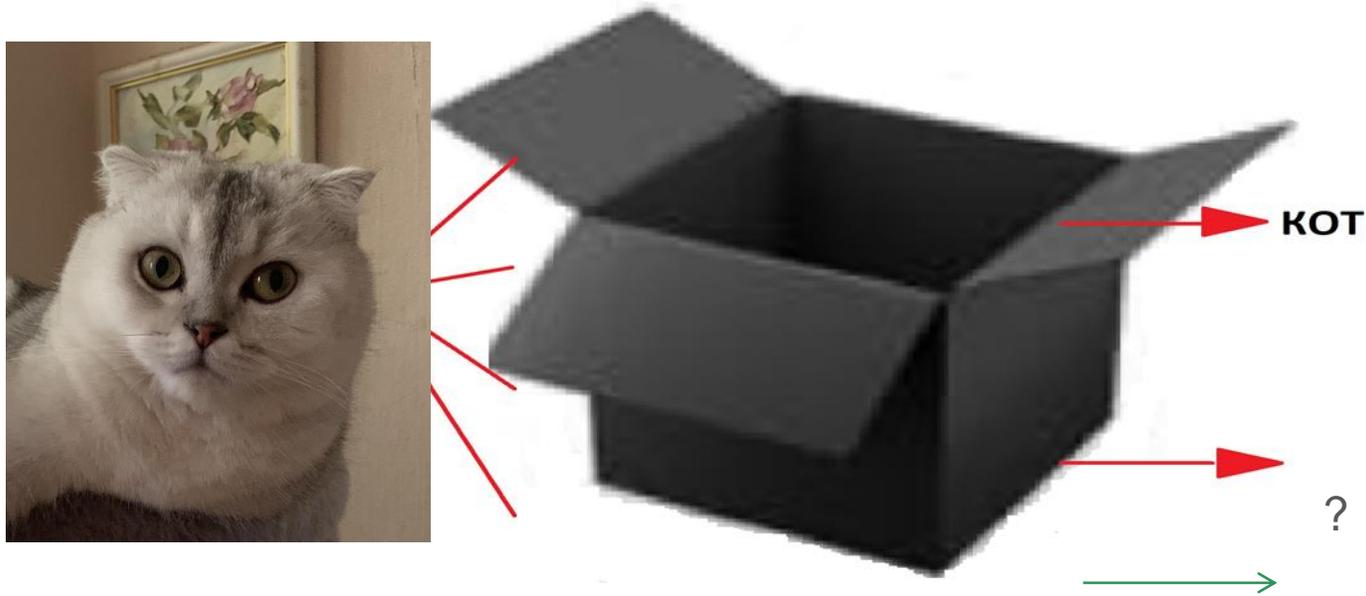
- Сокращение числа параметров н/с – **«р» раз** ?
- Сокращение разрядности данных - **«q» раз** ?

Эффекты

- Сжатие размера модели 5x ~ 100x
- Сокращение времени вычислений 1.5x - 10x

*Объяснительную модель можно: 1) быстро посчитать; 2) результаты легко интерпретируются и 3) использовать как параметр регуляризации*



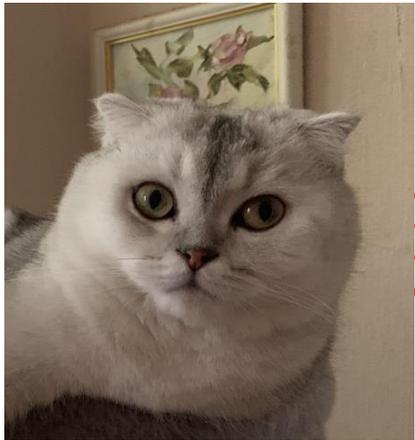


«черный ящик» модели **предсказывает**, что на фотографии изображен **КОТ** с вероятностью **0.98**



ПОЛИТЕХ

# МОДЕЛЬ С ТОЧКИ ЗРЕНИЯ ПОЛЬЗОВАТЕЛЯ С ОБЪЯСНЕНИЕМ



2 М пикселей



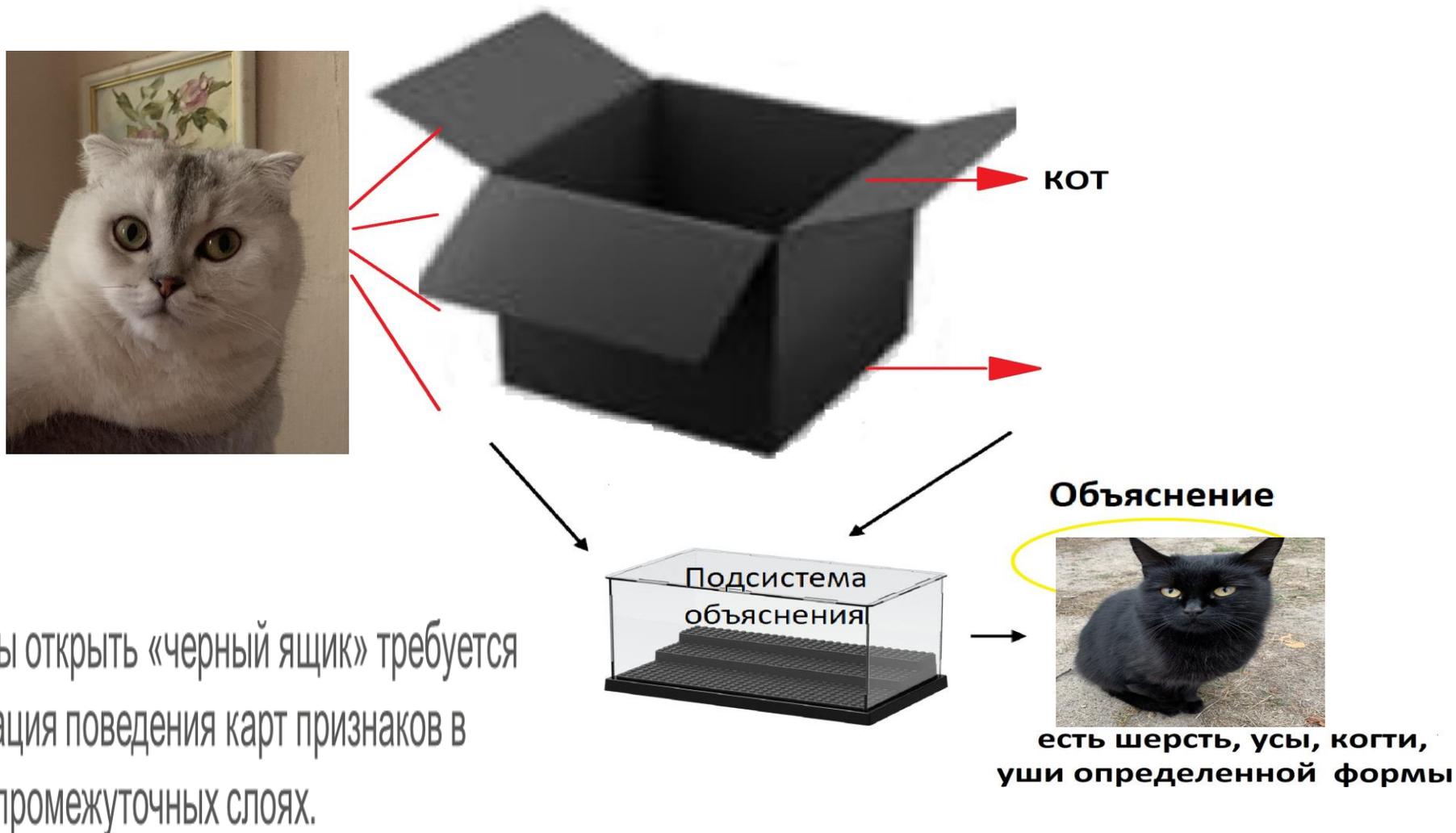
КОТ

есть шерсть, усы, когти,  
уши определенной формы

N понятий

«черный ящик» предсказывает, что на фотографии - **КОТ** с вероятностью 0.98, так как у него есть шерсть, усы, уши определенной формы»

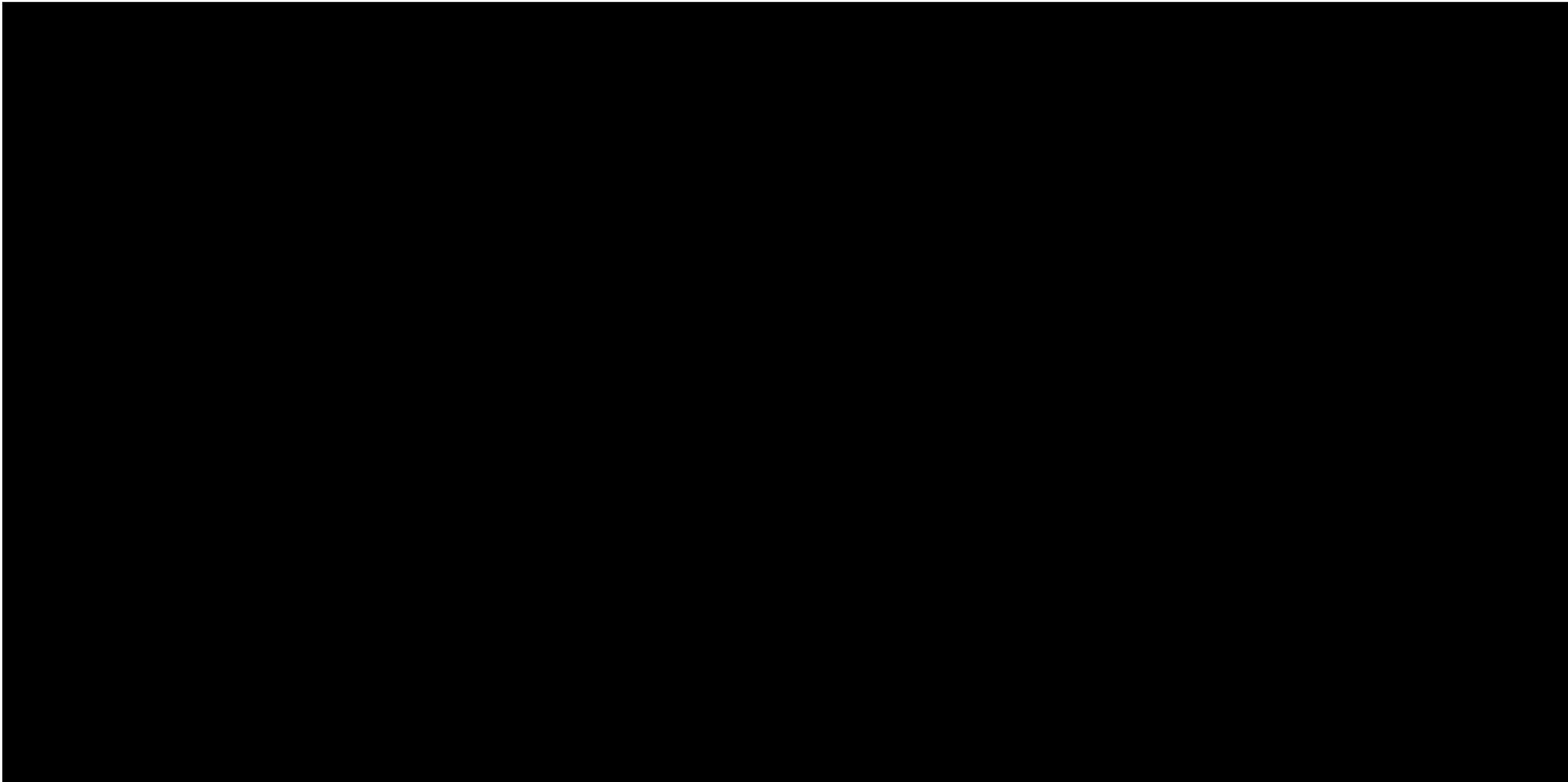
Итак, должна быть не только система расчетов, но и подсистема дедуктивных объяснений в терминах понятий, объем которых N многократно меньше 2M



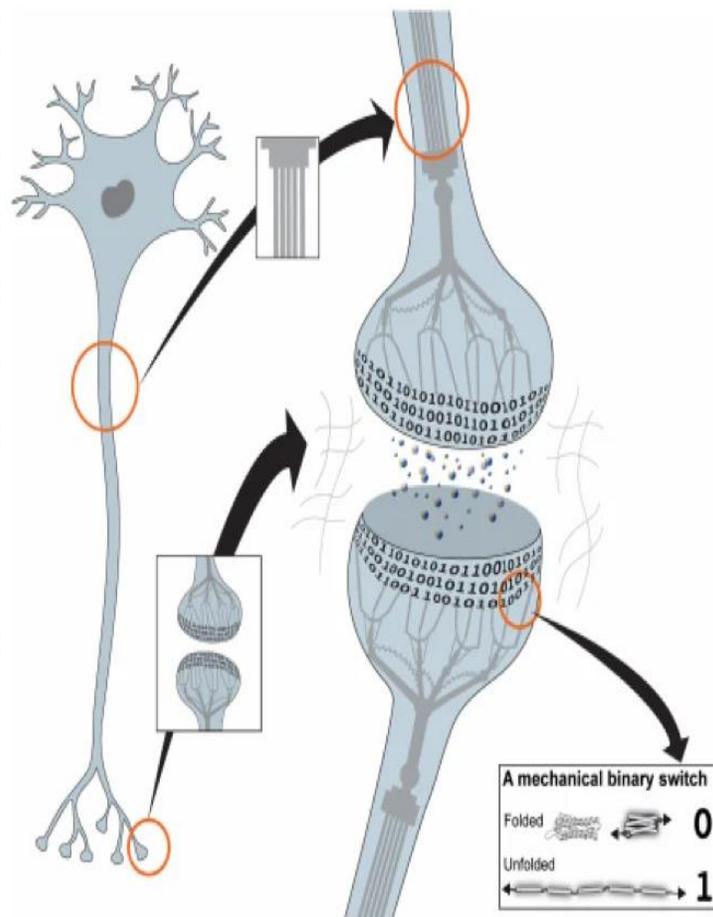
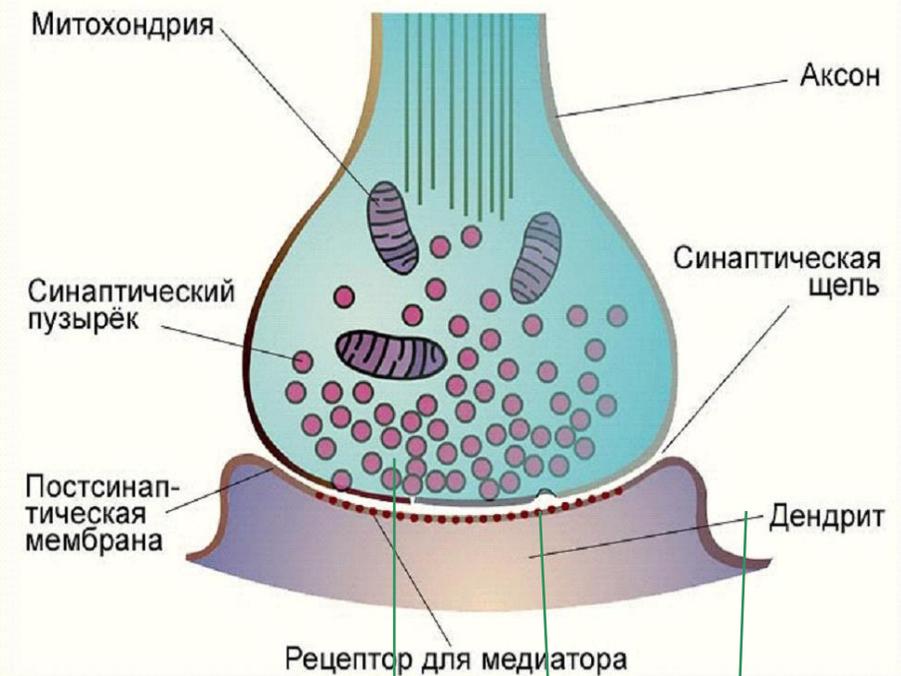
- **программирования** МТ– следование принципу хранимой программы /данных и **последовательного** выполнения **статического** множества операций ..... булевой алгебры.
- машинное обучение искусственных нейронных сетей – «запутывание» программы /данных и **параллельное** выполнения операций отображения **динамического** множеств входных данных на множество выходных данных (аналог процессов с обострением)

:

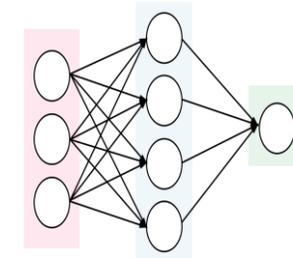
**Актуальность** этих двух подходов высокая, а повысить «остроту» компьютерных инструментов можно за счет «обучения» компьютерных систем реализовывать различные классы алгоритмов, реконфигурируя свою «вычислительную структуру».



# ПРИМЕР РЕШЕНИЯ : БИОИНСПИРИРОВАННАЯ СРЕДА ВИРТУАЛЬНОГО МОДЕЛИРОВАНИЯ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ, ПРОТЕКАЮЩИХ В АССОЦИАТИВНОЙ ПАМЯТИ

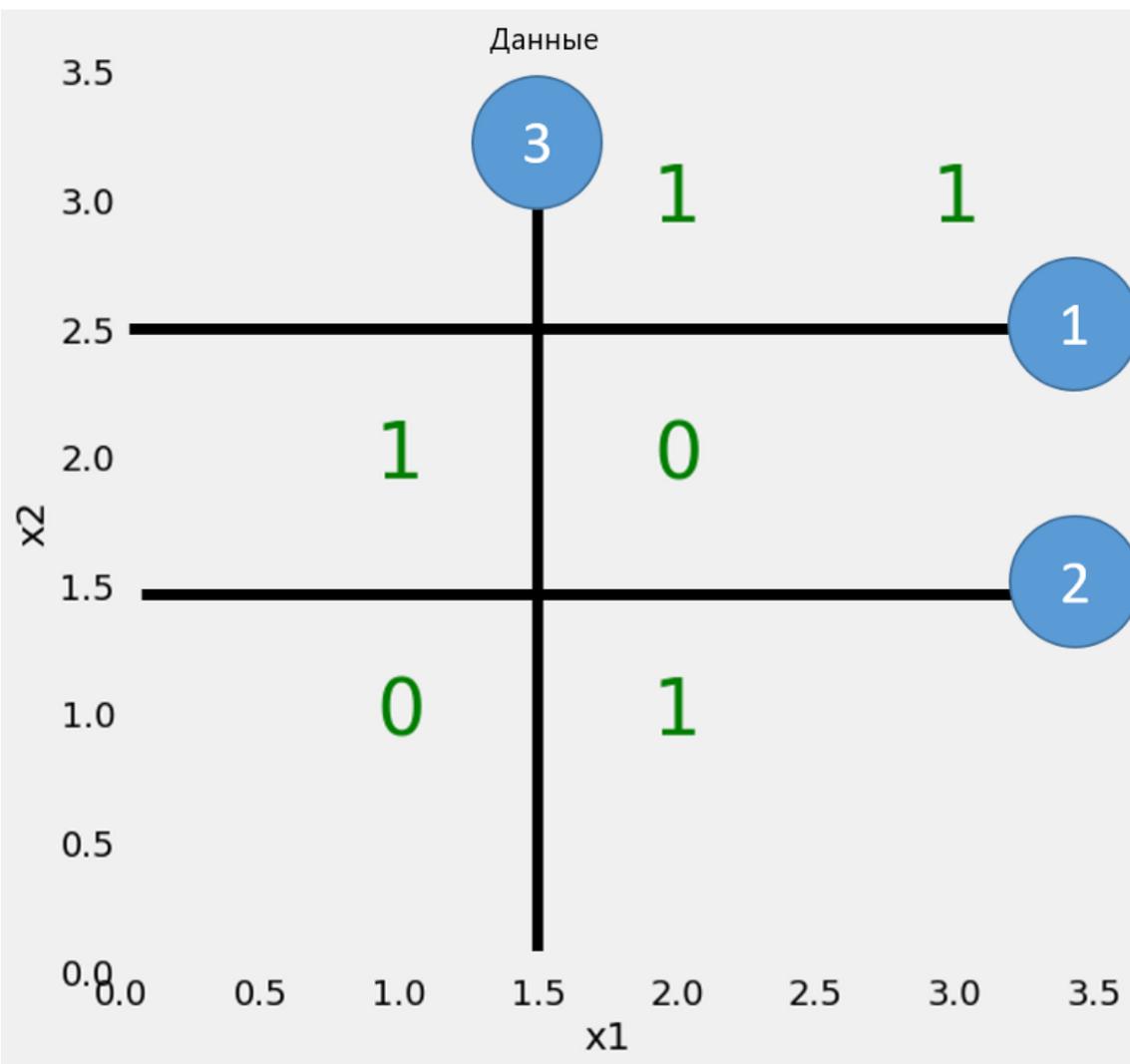


Физические каналы передачи информации – цепочка аксонов, инициирующих нейромедиаторные **виртуальные каналы**, являющиеся частью пространства состояний для хранения «паттернов» - аналога **аддитивного ситуационного базиса**



Нейромедиаторы — целостный носитель (сборка) **данных и программ**, представляемый через множество морфизмов .

Системная конфигурация «виртуальные» информационные каналы передачи нейромедиаторов (категорная конструкция)



Загрязнение Джини — вероятность неверной маркировки в узле случайно выбранного образца.

в верхнем (корневом) узле вероятность неверной классификации образца равна 44.4 %..

Загрязнение Джини узла  $n$  равно 1 минус сумма отношений класса к общему количеству образцов  $p_i$ , возведённых в квадрат, для каждого из множества классов  $J$  (в нашем случае это всего 2 класса)

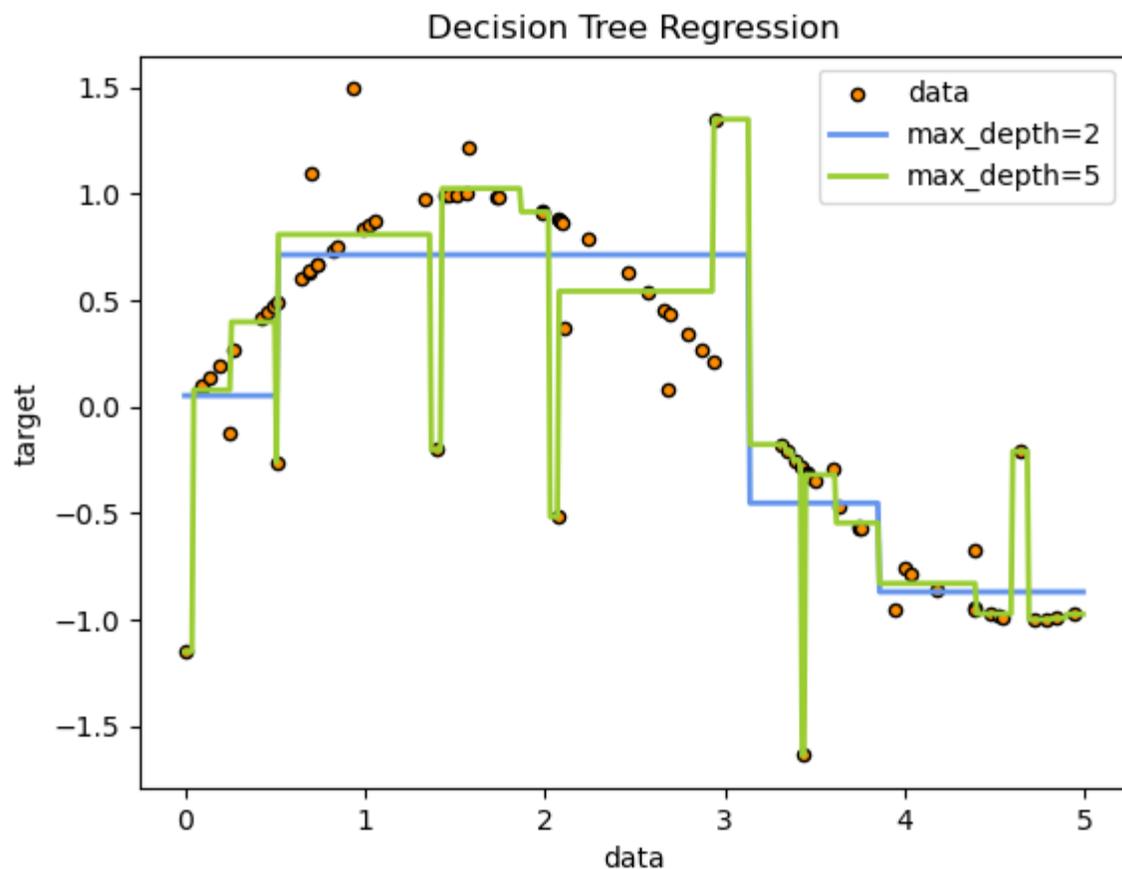
$$I_{root} = 1 - \left( \left( \frac{2}{6} \right)^2 + \left( \frac{4}{6} \right)^2 \right) = 1 - \frac{5}{9} = 0.444$$

В каждом узле дерево решений ищет такое значение определённого параметра, которое приведёт к максимальному уменьшению загрязнения Джини.

$$I_G(n) = 1 - \sum_{i=1}^J (p_i)^2$$

Деревья решений (DT) представляют собой непараметрический контролируемый метод обучения, используемый для классификации и регрессии. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной путем изучения простых правил принятия решений, выведенных из данных. Функции. Дерево можно рассматривать как кусочно-постоянное приближение.

# ПРИМЕР: ИСПОЛЬЗОВАНИЕ ДЕРЕВО РЕШЕНИЙ ДЛЯ



1D-регрессия с деревом решений.

Деревья решений используется для соответствия синусоидальной кривой с добавлением шумного наблюдения. В результате это изучает локальные линейные регрессии, аппроксимирующие синусоидальную кривую.

Если максимальная глубина дерева (контролируемая параметром) установлена слишком высоко, то деревья решений **учатся слишком точно** и реагируют на случайный шум в потоке данных

процесс деления повторяется с использованием «жадной», рекурсивной процедуры, пока дерево не достигнет максимальной глубины или в каждом узле не останутся только образцы одного класса. Общезвешенное загрязнение Джини должно уменьшаться с каждым уровнем. В нашем случае на втором уровне оно составит 0.333:

$$I_{\text{second layer}} = \frac{n_{\text{left}}}{n_{\text{parent}}} * I_{\text{left node}} + \frac{n_{\text{right}}}{n_{\text{parent}}} * I_{\text{right node}} = \frac{4}{6} * 0.5 + \frac{2}{6} * 0.0 = 0.333$$



"LOOK DEEP INTO  
NATURE, AND THEN YOU  
WILL UNDERSTAND  
BETTER."

*ALBERT EINSTEIN*

Добро пожаловать в «суперкомпьютерное море» объяснительного интеллекта