



# Введение в искусственный интеллект

Университетская суббота 23 октября 2021 года

А.В. Якушин, к.п.н., доцент, ВМК МГУ

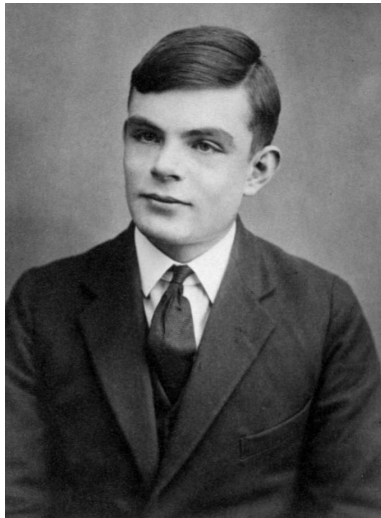
Факультет ВМК МГУ имени М.В. Ломоносова

## Что такое ИИ?



## Три закона робототехники (1942 год):

1. «Робот не может причинить вред человеку или своим бездействием допустить, чтобы человеку был причинён вред».
2. «Робот должен повиноваться всем приказам, которые даёт человек, кроме тех случаев, когда эти приказы противоречат Первому Закону».
3. «Робот должен заботиться о своей безопасности в той мере, в которой это не противоречит Первому или Второму Законам».



## Тест Тьюринга (1950 год):

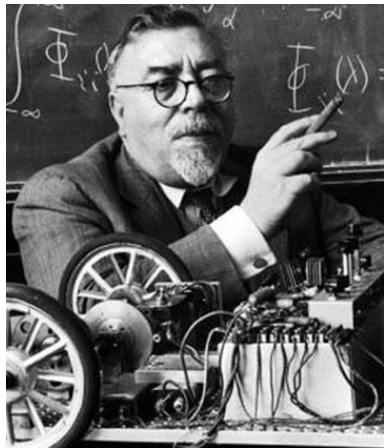
1. «Человек взаимодействует с одним компьютером и одним человеком.
2. На основании ответов на вопросы он должен определить, с кем он разговаривает: с человеком или компьютерной программой.
3. Задача компьютерной программы — ввести человека в заблуждение, заставив сделать неверный выбор».



Дартмутский семинар (1956 год, Дартмутский колледж, США):

«Мы предлагаем исследование искусственного интеллекта сроком в 2 месяца с участием 10 человек летом 1956 года в Дартмутском колледже, Гановер, Нью-Гемпшир. Исследование основано на предположении, что всякий аспект обучения или любое другое свойство интеллекта может в принципе быть столь точно описано, что машина сможет его симулировать. Мы попытаемся понять, как обучить машины использовать естественные языки, формировать абстракции и концепции, решать задачи, сейчас подвластные только людям, и улучшать самих себя. Мы считаем, что существенное продвижение в одной или более из этих проблем вполне возможно, если специально подобранная группа учёных будет работать над этим в течение лета».

- Джон фон Нейман
- Уоррен Мак-Каллок
- Уолтер Питтс
- Фрэнк Розенблатт
- Марвин Ли Минский  
(который на самом деле Массачусетский)
- Джуда Перл
- Лесли Гэбриел Вэлиант
- Йошуа Бенжио
- Джеффри Хинтон
- Ян Лекун
- Джон Хопфилд
- Теуво Кохонен
- Александр Фролов
- Виталий Дунин-Барковский



Его детище, кибернетика — наука об управлении и связях в машинах и живых организмах, родилось из сплава прежде не пересекавшихся математики, биологии, социологии. В своем фундаментальном труде «Кибернетика» (1948 год) сформулировал основные ее положения.

Кибернетика включает изучение обратной связи, чёрных ящиков и производных концептов, таких как управление и коммуникация в живых организмах, машинах и организациях, включая самоорганизации. Она фокусирует внимание на том, как что-либо (цифровое, механическое или биологическое) обрабатывает информацию, реагирует на неё и изменяется или может быть изменено, для того чтобы лучше выполнять первые две задачи.



«Требовалось найти слово, связанное с управлением, регулированием. Пришло на ум греческое, похожее на «рулевой», что по-английски звучит как «кибернетика». Так Винер его и оставил.»

Две основные проблемы кибернетики:

- Проблема N 1 кибернетики:  
Каким местом человек думает?
- Проблема N 2 кибернетики:  
Как он это этим местом делает?



**Искусственный интеллект** — это способность компьютерных систем выполнять такие задачи, для которых ранее предполагалось, что необходим человеческий разум. (artificial intelligence, AI)

Искусственный интеллект это НЕ искусственное сознание.

Более правильно говорить «искусственный разум».

В английском языке словосочетание artificial intelligence не имеет антропоморфной окраски, которую оно приобрело в традиционном русском переводе: слово intelligence в используемом контексте скорее означает «умение рассуждать разумно», а вовсе не «интеллект» (для которого есть английский аналог intellect).

Первый подход: моделируется поведение живых организмов. (робот собака, шахматная программа)

Второй подход: моделируется устройство биологических систем. (бионический протез)

Теория сильного ИИ: ИИ может получить способности мыслить и осознать себя как личность.

Теория слабого ИИ: ИИ не может мыслить и осознавать себя.

## Области применения искусственного интеллекта

- самоуправляемые автомобили,
- поиск потерявшихся людей (Lacmus),
- биометрия,
- распознавание рукописного ввода,
- генерация контента.

- распознавание и генерация речи:
- автоматические переводчики,
- голосовые интерфейсы (Siri, Cortana, Алиса),
- чат-боты («Порфирьевич»).

- обнаружение спама,
- банковский скоринг,
- юридические консультации (Legal Tech),
- дейтинг,
- медицинская диагностика,
- разработка лекарств (Insilico Medicine).

- Проект «NEON» от Samsung,
- Проект «Kórsafn» от Microsoft и Бьорк,
- «Project Debater» от IBM.

## Инструменты ИИ



- обучение с учителем,
- обучение без учителя,
- обучение с подкреплением,
- самообучение.

- сети прямого распространения,
- рекуррентные сети,
- сети с долгой краткосрочной памятью,
- стохастические сети,
- глубинные сети и т. д.

- генетический алгоритм,
- нейроэволюция,
- роевой интеллект.

## Формализация

- производить вычисления с числами
- выполнять численные расчеты
- осуществлять арифметические действия
- считать
- вычислять по формуле

Формализация это описание задачи на формальном языке, например на языке математики или на языке программирования.

Хорошо формализованная задача: «вычислить в десятичной системе счисления  $2+3$ ».

Плохо формализованная задача: «купи что-нибудь вкусное».

Формализация информации о некотором объекте — это ее отражение в определенной форме. Можно еще сказать так: формализация — это сведение содержания к форме. Формулы, описывающие физические процессы, — это формализация этих процессов. Радиосхема электронного устройства — это формализация функционирования этого устройства. Ноты, записанные на нотном листе, — это формализация музыки и т.п.

Формализованная информационная модель — это определенные совокупности знаков (символов), которые существуют отдельно от объекта моделирования, могут подвергаться передаче и обработке. Реализация информационной модели на компьютере сводится к ее формализации в форматы данных, с которыми "умеет" работать компьютер.

Но можно говорить и о другой стороне формализации применительно к компьютеру. Программа на определенном языке программирования есть формализованное представление процесса обработки данных.

Формальные описания существенно различны, даже могут быть получены существенно различными способами на основании некоторого «исходного описания» задачи в терминах проблемной области.

Не всё можно заранее сформулировать (из-за наличия различного типа неопределенностей (НЕ-факторов - неполноты знаний, нечёткости критериев и т.д.))

Где нужно открыть магазин? Кто изображен на фото? Зачем этот человек идет в магазин? Почему у ящиков есть столы? Сколько ангелов поместится на кончике иглы? Как повысить уровень прибыли? Почему ты не делаешь так как я хочу, но я не хочу говорить тебе что я хочу потому что тогда ты будешь делать так как я хочу потому что я тебе сказала что я так хочу, а я хочу чтоб ты сам догадался чего я хочу.



Для решения задачи на компьютере она должна быть формализована.

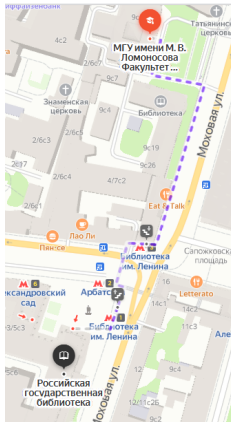
Искусственный интеллект хорошо решает хорошо формализованные задачи.

Искусственный интеллект плохо решает плохо формализованные задачи.

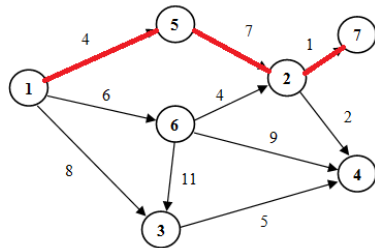
Например, чтобы решить задачу о построении кратчайшего маршрута доставки пиццы клиенту необходимо построить карту местности, нанести на нее все объекты и уметь рассчитывать расстояние между ними.

Основная **проблема** — разработать инструментарий (формальный язык), который позволит проводить формализацию плохо формализуемых задач.

## Что видит человек



## Как это представлено в компьютере



# Бредогенератор

Репозиторий исходного кода

Фирмой «Хониуэлл Иннкорпорейтед» разработан генератор псевдотекста, использующий SIMP-таблицы (Simplified Integrated Modular Prose – упрощённая интегрированная модульная проза).

Данный генератор позволяет генерировать общеупотребительные псевдонаучные фразы.

Его работа основана на генерации случайного четырёхзначного числа и выборке из четырёх SIMP-таблиц соответствующих частей предложения.

Таблица А	Таблица В	Таблица С	Таблица D
В частности	гиперповерхность в пространстве состояний	находит широкое применение и требует	более тонкой аппаратурной реализации.
С другой стороны,	постоянный поток эффективной информации	сводит до минимума затраты при условии	оборудования четвёртого поколения.
Однако	отличительная особенность выбранных критериев	указывает на пределы применимости	тестирования четвёртого поколения

Введите количество фраз: 5

Как показывают приведённые выше соображения независимый принцип функционирования свидетельствует о необходимости более тщательного анализа более строгой стандартизации основных модулей.

Аналогично гиперповерхность в пространстве состояний признаёт значимость других систем и необходимость разветвления сети сопровождения и поддержки.

Например независимый принцип функционирования подразумевает более основательное использование теории тестирования четвёртого поколения.

Нетрудно видеть, что комплексная программа испытаний требует применения оборудования четвёртого поколения.

Как показывают приведённые выше соображения комплексная программа испытаний требует применения более строгой стандартизации основных модулей.

## Нечеткие множества

*Множества состоят из элементов.*

Запись  $x \in M$  означает, что  $x$  является элементом множества  $M$ .

Есть некоторый объект. Когда он попадает в множество, то становится его **элементом**.

Пример

$$M = \{a, b, c\}$$

Пример

$$a \in M; 5 \notin M$$



Нечеткая логика предназначена для формализации неточных или приближенных рассуждений, позволяющих более адекватно описывать ситуации с неопределенностью. Понятие нечетких множеств (англ.: fuzzy sets) как обобщение обычных (четких) множеств было введено американским ученым Л. Заде в 1965 г.

Традиционный метод представления элемента множества  $A$  состоит в применении характеристической функции  $\mu_A(x)$ , которая равна 1, если этот элемент  $x$  принадлежит к множеству  $A$ , или равна 0 в противном случае.

Для обычных множеств элемент либо принадлежит множеству ( $\mu_A(x) = 1$ ), либо нет ( $\mu_A(x) = 0$ ). Третьего не дано (пресловутый принцип исключения третьего). Следствием теории четких множеств является булева логика, все то множество схем рассуждений и выводов, которые опираются на понятие характеристической функции.

В теории нечетких множеств вместо характеристической функции используется функция принадлежности  $m_A : X \in [0,1]$ .  $m_A$  – это субъективная оценка степени принадлежности элемента  $x$  к множеству  $A$ .

Понятие «маленького числа» (на множестве  $A = \{0,1,2,3,4,5,6,7,8,9,10\}$ ) можно определить в виде нечеткого множества:

число 0 однозначно является маленьким ( $m_A = 1$ ),

число 1 – тоже ( $m_A = 1$ )

число 2 – уже не очень маленькое ( $m_A = 0.8$ ).

Это тем более касается чисел 3 ( $m_A = 0.5$ ) и 4 ( $m_A = 0.1$ , т.е. 4 – это почти наверняка немаленькое число). числа от 5 до 10 – однозначно не маленькие ( $m_A = 0$ ).

Число	0	1	2	3	4	5	6	7	8	9	10
$m_A$	1	1	0.8	0.5	0.1	0	0	0	0	0	0

Не обязательно использовать числовые оценки. Зачастую, с точки зрения взаимодействия с пользователем, целесообразнее использовать т.н. «лингвистические переменные» – термины типа «много», «мало», «высокий», «низкий» и т.п.

Рассмотрим понятие «высокий–низкий»

Рост	2.20	2.10	2.00	1.90	1.80	1.70	1.60
$m_A$	1	1	0.8	0.6	0.4	0.2	0.0

Таким образом мы можем формализовать понятие «высокий рост», например для системы распознавания образов, чтобы компьютер мог определить: «на фото человек высокого роста».

Также можно задавать команды роботу: «остановись недалеко от ямы».

**Упражнение:** придумайте нечеткую функцию принадлежности для понятия «в комнате холодно».

# Принятие решения о выборе кредита на основе нечеткой логики

Репозиторий исходного кода

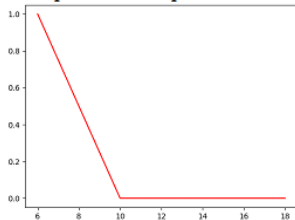
Принятие решения о выборе кредита на основе нечеткой логики.

Оценка уровня выгоды кредита. На основании выбранной процентной ставки и срока кредитования, программа будет сообщать о том, насколько выгодный для получателя кредит по заданным параметрам.

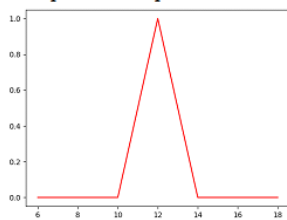
## Пример 2. Лингвистическая переменная «Хорошая ставка»



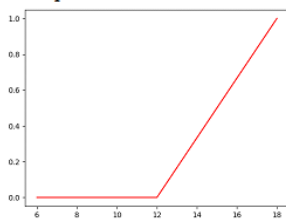
Переменная «хорошая ставка»



Переменная «средняя ставка»



Переменная «плохая ставка»



ЕСЛИ ставка хорошая и срок хороший, то берем  
ЕСЛИ ставка средняя, то берем по необходимости  
ЕСЛИ ставка плохая или срок плохой, то не берем

Введите % ставку

10

Введите срок (месяцев)

24

Процентная ставка: хор:0.34782608695652173 сред:0 плохо:0

Продолжительность кредита: хор:0.0 сред:0.7982456140350878 плохо:0

Хорошие условия



## Что такое машинное обучение?

Машинное обучение играет важную роль в современном бизнесе и исследованиях. Оно использует различные алгоритмы, включая искусственные нейронные сети, для помощи компьютерным системам в постепенном улучшении их эффективности. Алгоритмы машинного обучения автоматически строят математические модели, используя имеющиеся данные, чтобы принимать решения, не будучи явно запрограммированными для принятия этих решений.

Впервые термин машинного обучения был введен в обиход пионером искусственного интеллекта, изобретателем самообучающейся программы для игры в шашки Артуром Сэмюэлом в 1959 году:

**Машинное обучение** это область исследований, которая даёт компьютеру возможность учиться, не будучи явно запрограммированным.

В одной из первых книг по машинному обучению «Machine Learning», написанной Томом Митчеллом в 1997, даётся следующее определение:

Говорят, что компьютерная программа учится из опыта  $\mathcal{E}$  по отношению к какому-то классу задач  $\mathcal{T}$  и критерию качества  $\mathcal{P}$ , если её качество на задачах из  $\mathcal{T}$ , измеренное с помощью  $\mathcal{P}$ , улучшается с использованием опыта  $\mathcal{E}$ .

## Классификация алгоритмов

В зависимости от постановки задачи используются различные алгоритмы:

- Обучение с учителем.
- Обучение без учителя.
- Частичное обучение.
- Обучение с подкреплением.
- Динамическое обучение.
- Активное обучение.
- Метаобучение.

Рассматривается решение следующей задачи. Имеется множество объектов (ситуаций) и множество возможных ответов (откликов, реакций). Существует некоторая зависимость между ответами и объектами, но она неизвестна. Известна только конечная совокупность прецедентов — пар (объект, ответ), называемая обучающей выборкой. На основе этих данных требуется восстановить зависимость, то есть построить алгоритм, способный для любого объекта выдать достаточно точный ответ. Для измерения точности ответов определённым образом вводится функционал качества.

Под учителем понимается либо сама обучающая выборка, либо тот, кто указал на заданных объектах правильные ответы.

## Типы входных данных

- Признаковое описание или матрица объекты-признаки — наиболее распространённый случай. Каждый объект описывается набором своих характеристик, называемых признаками. Признаки могут быть числовыми или нечисловыми.
- Матрица расстояний между объектами. Каждый объект описывается расстояниями до всех остальных объектов обучающей выборки. С этим типом входных данных работают немногие методы, в частности, метод ближайших соседей, метод парзеновского окна, метод потенциальных функций.
- Временной ряд или сигнал представляет собой последовательность измерений во времени. Каждое измерение может представляться числом, вектором, а в общем случае — признаковым описанием исследуемого объекта в данный момент времени.
- Изображение или видеоряд.

## Типы откликов

- Задачи классификации — множество возможных ответов конечно. Их называют идентификаторами (именами, метками) классов.
- Задачи регрессии — ответы являются действительными числами или векторами.

Пусть  $X$  — множество описаний объектов,  $Y$  — множество допустимых ответов. Существует неизвестная целевая зависимость — отображение  $y^* : X \rightarrow Y$ , значения которой известны только на объектах конечной обучающей выборки  $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$ . Требуется построить алгоритм  $a : X \rightarrow Y$ , который приближал бы неизвестную целевую зависимость как на элементах выборки, так и на всём множестве  $X$ .

Говорят также, что алгоритм должен обладать способностью к обобщению эмпирических фактов, или выводить общее знание (закономерность, зависимость) из частных фактов (наблюдений, прецедентов).

Данная постановка является обобщением классических задач аппроксимации функций. В классической аппроксимации объектами являются действительные числа или векторы. В реальных прикладных задачах входные данные об объектах могут быть неполными, неточными, неоднородными, нечисловыми. Эти особенности приводят к большому разнообразию методов обучения с учителем.



Рассматривается решение следующей задачи. Известны только описания множества объектов (обучающей выборки), и требуется обнаружить внутренние взаимосвязи, зависимости, закономерности, существующие между объектами.

## Типы входных данных

- Признаковое описание объектов. Каждый объект описывается набором своих характеристик, называемых признаками. Признаки могут быть числовыми или нечисловыми.
- Матрица расстояний между объектами. Каждый объект описывается расстояниями до всех остальных объектов обучающей выборки.

- **Кластеризация.** Выборка объектов разбивается на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались. Исходная информация представляется в виде матрицы расстояний.
- **Поиск правил ассоциации.** Исходная информация представляется в виде признаковых описаний. Задача состоит в том, чтобы найти такие наборы признаков, и такие значения этих признаков, которые особенно часто (неслучайно часто) встречаются в признаковых описаниях объектов.
- **Заполнение пропущенных значений.** Исходная информация представляется в виде признаковых описаний. Значения некоторых признаков для некоторых объектов могут отсутствовать. Необходимо восстановить пропущенные значения признаков.
- **Сокращение размерности.** Исходная информация представляется в виде признаковых описаний, причём число признаков может быть достаточно большим. Задача состоит в том, чтобы представить эти данные в пространстве меньшей размерности, по возможности, минимизировав потери информации.
- **Визуализация данных.** Некоторые методы кластеризации и снижения размерности строят представления выборки в пространстве размерности два. Это позволяет отображать многомерные данные в виде плоских графиков и анализировать их визуально, что способствует лучшему пониманию данных и самой сути решаемой задачи.

Один из методов машинного обучения, использующий при обучении как размеченные, так и неразмеченные данные. Обычно используется небольшое количество размеченных и значительный объём неразмеченных данных. Частичное обучение является компромиссом между обучением без учителя (без каких-либо размеченных обучающих данных) и обучением с учителем (с полностью размеченным набором обучения). Было замечено, что неразмеченные данные, будучи использованными совместно с небольшим количеством размеченных данных, могут обеспечить значительный прирост качества обучения.

Под качеством обучения подразумевается некий функционал качества, например, среднеквадратичная ошибка. Сбор размеченных данных для задачи обучения зачастую требует, чтобы квалифицированный эксперт вручную классифицировал объекты обучения. Затраты, связанные с процессом разметки, могут сделать построение полностью размеченного набора прецедентов невозможным, в то время как сбор неразмеченных данных сравнительно недорог. В подобных ситуациях ценность частичного обучения сложно переоценить.

Примером частичного обучения может послужить сообучение: два или более обучаемых алгоритма используют один и тот же набор данных, но каждый при обучении использует различные — в идеале некоррелированные — наборы признаков объектов.

Альтернативный подход заключается в моделировании совместного распределения признаков и меток. В таком случае для неразмеченных данных метки могут трактоваться как пропущенные данные. Для построения оценки максимального правдоподобия обычно используется EM-алгоритм.

Обучение с подкреплением, идея которого была почерпнута в смежной области психологии, является подразделом машинного обучения, изучающим, как агент должен действовать в окружении, чтобы максимизировать некоторый долговременный выигрыш. Алгоритмы с частичным обучением пытаются найти стратегию, приписывающую состояниям окружающей среды действия, которые должен предпринять агент в этих состояниях. В экономике и теории игр обучение с подкреплением рассматривается в качестве интерпретации того, как может установиться равновесие.

Окружение обычно формулируется как марковский процесс принятия решений (МППР) с конечным множеством состояний, и в этом смысле алгоритмы обучения с подкреплением тесно связаны с динамическим программированием. Вероятности выигрышей и перехода состояний в МППР обычно являются величинами случайными, но стационарными в рамках задачи.

При обучении с подкреплением, в отличие от обучения с учителем, не предоставляются верные пары «входные данные-ответ», а принятие субоптимальных решений (дающих локальный экстремум) не ограничивается явно. Обучение с подкреплением пытается найти компромисс между исследованием неизученных областей и применением имеющихся знаний. Баланс изучения-применения при обучении с подкреплением исследовался в задаче многоорукого бандита.

Формально простейшая модель обучения с подкреплением состоит из:

1. множества состояний окружения  $S$ ;
2. множества действий  $A$ ;
3. множества вещественнозначных скалярных «выигрышей».

Обучение с подкреплением особенно хорошо подходит для решения задач, связанных с выбором между долгосрочной и краткосрочной выгодой. Оно успешно применялось в различных областях, таких как робототехника, управление лифтами, телекоммуникации, шашки и нарды.

При обучении с подкреплением, в отличие от обучения с учителем, не предоставляются верные пары «входные данные-ответ», а принятие субоптимальных решений (дающих локальный экстремум) не ограничивается явно. Обучение с подкреплением пытается найти компромисс между исследованием неизученных областей и применением имеющихся знаний. Баланс изучения-применения при обучении с подкреплением исследовался в задаче многоорукого бандита.

Формально простейшая модель обучения с подкреплением состоит из:

1. множества состояний окружения  $S$ ;
2. множества действий  $A$ ;
3. множества вещественнозначных скалярных «выигрышей».

Обучение с подкреплением особенно хорошо подходит для решения задач, связанных с выбором между долгосрочной и краткосрочной выгодой. Оно успешно применялось в различных областях, таких как робототехника, управление лифтами, телекоммуникации, шашки и нарды.

## Модель машинного обучения



**Размеченные данные** — это группа данных с присвоенными справочными тегами или выходной информацией. Например, массив фотографий котов, в котором указано, что это именно фотографии котов.

Данные о параметрах цветков ириса для определения вида ириса:

sepal_length	sepal_width	petal_length	petal_width	class(метка)
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
7.0	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
6.3	3.3	6.0	2.5	Iris-virginica
5.8	2.7	5.1	1.9	Iris-virginica

**Неразмеченные данные** — это группа данных, у которых отсутствует «метка».

Данные о клиентах магазина для проведения сегментации:

sex	age	marital	housing	balance
M	35	married	yes	1000
M	30	single	yes	2000
F	32	married	yes	500
M	32	married	no	300
F	33	single	no	30000
F	27	married	yes	4000

**Выборка** (sample, set) — конечный набор прецедентов (объектов, случаев, событий, испытуемых, образцов, и т.п.), некоторым способом выбранных из множества всех возможных прецедентов, называемого **генеральной совокупностью**.

**Обучающая выборка** (training sample) — выборка, по которой производится настройка (оптимизация параметров) модели зависимости.

**Тестовая** (или контрольная) выборка (test sample) — выборка, по которой оценивается качество построенной модели. Если обучающая и тестовая выборки независимы, то оценка, сделанная по тестовой выборке, является несмещённой.

**Проверочная** выборка (validation sample) — выборка, по которой осуществляется выбор наилучшей модели из множества моделей, построенных по обучающей выборке.

В каждой задаче машинного обучения ставится вопрос оценки результатов моделей.

Без введенных критериев, невозможно будет ни оценить «успешность» модели, ни сравнить между собой два различных алгоритма. Именно поэтому важно учесть правильный выбор метрик для поставленной задачи, хотя множество существующих метрик может запутать и, в конечном счете, привести к неоптимальному решению.

В машинном обучении различают оценки качества для задачи классификации и регрессии. Причем оценка задачи классификации часто значительно сложнее, чем оценка регрессии.

В данном случае, данные поделены всего на два класса. Рассматриваемые нами метрики основаны на использовании следующих исходов: истинно положительные (TP), истинно отрицательные (TN), ложно положительные (FP) и ложно отрицательные (FN). Для наглядности, можно преобразовать в матрицу ошибок.

	Метка(1)	Метка(0)
Распознано (1)	TP	FP
Распознано (0)	FN	TN

Пусть банк использует систему классификации заёмщиков на кредитоспособных и некредитоспособных. При этом первым кредит выдаётся, а вторые получают отказ. Таким образом, обнаружение некредитоспособного заёмщика ( $y=1$ ) можно рассматривать как «сигнал тревоги», сообщающий о возможных рисках.

Любой реальный классификатор совершает ошибки:

- Кредитоспособный заёмщик распознается моделью как некредитоспособный и ему отказывается в кредите. Данный случай можно трактовать как «ложную тревогу».
- Некредитоспособный заёмщик распознаётся как кредитоспособный и ему ошибочно выдаётся кредит. Данный случай можно рассматривать как «пропуск цели».

Поскольку с точки зрения логики задачи нам важнее правильно распознать некредитоспособного заёмщика с меткой  $y=1$ , чем ошибиться в распознавании кредитоспособного, будем называть соответствующий исход классификации положительным (заёмщик некредитоспособен), а противоположный - отрицательным (заемщик кредитоспособен  $y=0$ ).

Аделаида любит ходить на быстрые свидания. Однажды она задумалась о том насколько хорошо работает ее внутренний распознаватель качества потенциальных кавалеров. Она составила следующую таблицу (результаты за последний месяц):

	Хороший(1)	Плохой(0)
Предсказано (1)	96	8
Предсказано (0)	4	19

**accuracy** — доля правильных ответов алгоритма:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{96+19}{96+19+4+8} = 0,91$$

Точностью (**precision**) называется доля правильных ответов модели

$$precision = \frac{TP}{TP+FP} = \frac{96}{96+8} = 0,92$$

Полнота (**recall**) — это доля истинно положительных классификаций.

$$recall = \frac{TP}{TP+FN} = \frac{96}{96+4} = 0,96$$

**Средняя квадратичная ошибка** (англ. Mean Squared Error, MSE). MSE применяется в ситуациях, когда нам надо подчеркнуть большие ошибки и выбрать модель, которая дает меньше больших ошибок прогноза. Грубые ошибки становятся заметнее за счет того, что ошибку прогноза мы возводим в квадрат. И модель, которая дает нам меньшее значение среднеквадратической ошибки, можно сказать, что у этой модели меньше грубых ошибок.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2$$

**Средняя абсолютная ошибка** (англ. Mean Absolute Error, MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \tilde{y}_i|$$



## Пример машинного обучения

Пусть мы хотим научить компьютер распознавать четные и нечетные числа. Для этого у нас есть размеченные данные следующего вида (data01.csv):

```
number;target
```

```
100 ;0
```

```
101 ;1
```

Здесь target это признак: 0 — четное, 1 — нечетное.

Пусть мы хотим научить компьютер распознавать четные и нечетные числа. Для этого у нас есть размеченные данные следующего вида (data01.csv):

```
number;target  
100 ;0  
101 ;1
```

Здесь target это признак: 0 — четное, 1 — нечетное.

Как это все работает?

У нас есть обучающая выборка, которая содержит и число и метку (четное или нечетное).

Далее мы берем тестовую выборку и число из нее. Смотрим на какое из чисел обучающей выборки оно больше всего похоже и говорим, что у него будет такая же метка и так далее для всех чисел тестовой выборки.

Задача построение модели машинного обучения состоит в том, что мы хотим подобрать коэффициенты неизвестной функции (функция распознавания), которая наиболее хорошо находит «похожие» числа в обучающей выборке. Далее с помощью этой функции можно распознавать произвольные числа.

Размер данных 100  
(контрольная выборка 20):

	Нечетное(1)	Четное(0)
Предсказано (1)	5	8
Предсказано (0)	4	3

Размер данных 10000  
(контрольная выборка 2000):

	Нечетное(1)	Четное(0)
Предсказано (1)	1029	6
Предсказано (0)	4	961

Размер данных 1000  
(контрольная выборка 200):

	Нечетное(1)	Четное(0)
Предсказано (1)	44	61
Предсказано (0)	47	48

Размер данных 100000  
(контрольная выборка 20000):

	Нечетное(1)	Четное(0)
Предсказано (1)	9966	0
Предсказано (0)	0	10034

## Что такое нейронные сети?

- 1943г. У. МакКаллок, У. Питтс. Статья о вычислениях в сетях формальных нейронов.
- 1951г. М. Минский. Первый экспериментальный нейрокомпьютер Snark
- 1961г. Ф. Розенблат. Создание персептрона, идея обучения на примерах.
- 1969г. М. Минский, С. Пейперт. Книга «Перцептроны».
- 1974г. П.Дж.Вербос, А.И.Галушкин. Алгоритм обратного распространения ошибки.
- 1986г. Д.И.Румельхарт, С.И.Барцев. Развитие метода обратного распространения ошибки
- 1998г. Я. ЛеКун. Сверточные сети.
- 2012г. Глубокое обучение.

**Нейронная сеть** — это последовательность нейронов, соединенных между собой синапсами. Структура нейронной сети пришла в мир программирования напрямую из биологии. Благодаря такой структуре, машина обретает способность анализировать и даже запоминать различную информацию. Нейронные сети также способны не только анализировать входящую информацию, но и воспроизводить ее из своей памяти.

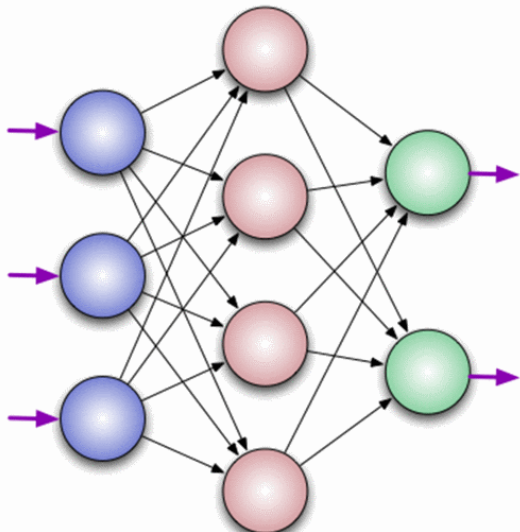
Другими словами, нейросеть это машинная интерпретация мозга человека, в котором находятся миллионы нейронов передающих информацию в виде электрических импульсов.



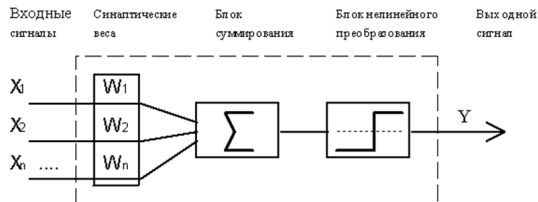
**Нейронная сеть** — попытка с помощью математических моделей воспроизвести работу человеческого мозга для создания машин, обладающих искусственным интеллектом.

**Искусственная нейронная сеть** обычно обучается с учителем. Это означает наличие обучающего набора (датасета), который содержит примеры с истинными значениями: тегами, классами, показателями.

Неразмеченные наборы также используют для обучения нейронных сетей, но мы не будем здесь это рассматривать.



Нейрон — это вычислительная единица, которая получает информацию, производит над ней простые вычисления и передает ее дальше. Они делятся на три основных типа: **входной** (синий), **скрытый** (красный) и **выходной** (зеленый).



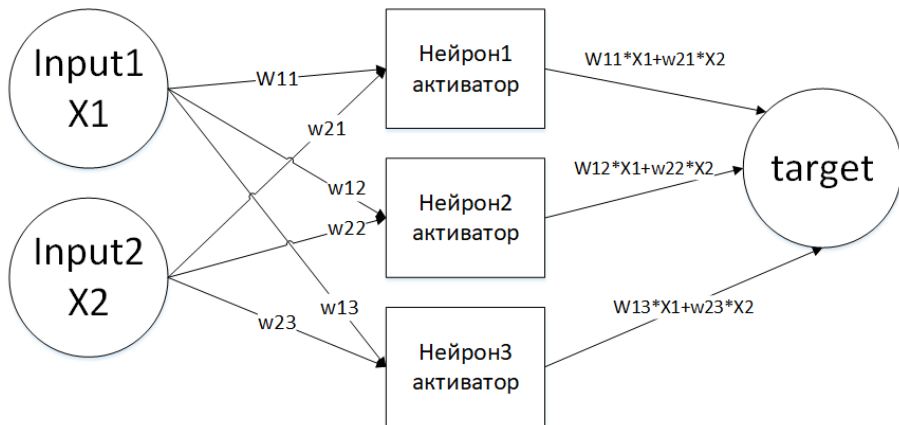
Искусственный нейрон (artificial neuron), представляет собой упрощенную модель биологического нейрона. Все, что делает искусственный нейрон — это принимает сигналы со многих входов, обрабатывает их единым образом и передает результат на многие другие искусственные нейроны, т.е. делает то же самое, что и нейрон биологический.

Связи между искусственными нейронами называются синаптическими, или просто синапсами. У синапса имеется один параметр — весовой коэффициент, в зависимости от его значения происходит то или иное изменение информации, когда она передается от одного нейрона к другому

## Обучение нейронной сети

1. Нам нужны размеченные данные:  $X = (x_1, x_2, \dots, x_n)_k \Rightarrow y_k$ . Т.е. есть входной вектор  $X_k$  и значение  $y_k$ , которое ему соответствует.
2. Рассмотрим задачу обучить нейронную сеть складывать числа. Размеченные данные:

```
input1;input2; target  
4414; 9628; 14042  
6639; 3626; 10265  
436; 5954; 6390  
952; 1718; 2670  
6704; 7714; 14418
```



Пусть Нейрон1(0.4;0.5); Нейрон2(0.8;0.2); Нейрон3(0.8;0.8).

Вход:  $X1 = 4414$ ,  $X2 = 9628$ . Правильный ответ: 14042.

Нейрон1 формирует выход:  $4414 * 0,4 + 9628 * 0,5 = 6579,6$ .

Нейрон2 формирует выход:  $4414 * 0,8 + 9628 * 0,2 = 5456,8$ .

Нейрон3 формирует выход:  $4414 * 0,8 + 9628 * 0,8 = 11233,6$ .

Нейрон1 ошибка:  $14042 - 6579,6 = 7462,4$ .

Нейрон2 ошибка:  $14042 - 5456,8 = 8585,2$ .

Нейрон3 ошибка:  $14042 - 11233,6 = 2808,4$ .

Самая маленькая ошибка у нейрона3. Будем его использовать для вычислений. Его активатор будет работать. Активаторы других нейронов не работают.

Задача обучения нейронной сети состоит в том, чтобы **подобрать такие коэффициенты нейронов**, чтобы выходной результат нейронной сети минимально отличался от эталонного результата.

```
model = keras.Sequential([  
    keras.layers.Flatten(input_shape=(2,)),  
    keras.layers.Dense(20, activation=tf.nn.relu),  
    keras.layers.Dense(20, activation=tf.nn.relu),  
    keras.layers.Dense(1)])
```

Результат работы (2 цикла обучения):

$9551+2833=12386.035$   
 $7708+8923=16633.42$   
 $2435+150=2586.16$   
 $7394+517=7913.83$   
 $5509+4049=9559.64$

Результат работы (10 циклов обучения):

$9551+2833=12383.76$   
 $7708+8923=16630.53$   
 $2435+150=2584.69$   
 $7394+517=7910.34$   
 $5509+4049=9557.68$