

# Искусственный интеллект: КОГНИТИВНОЕ НАЧАЛО

## En Artificial Intelligence: Cognitive Beginning

**V. A. Artamonov,**  
PhD (Eng., Grand Doctor), Full Professor,  
the Full Member of IAIT  
artamonov@itzashita.ru

**E. V. Artamonova,**  
PhD (Eng.), the Member of IAIT  
admin@itzashita.ru

**A. E. Safonov,**  
M. Sc. IT  
mail@iwebi.group

The International Academy of  
Information Technologies (IAIT)

Three types of artificial intelligence are considered in this article. The authors discuss the problems and threats that artificial intelligence carries for society. In this paper, the most popular machine learning algorithms are analyzed, it highlights the pros and cons of these methods. The article focuses on the methods of machine learning, and provides a mathematical justification for these methods.

**Keywords:** artificial intelligence, symbolism, connectionism, machine learning, neural networks, algorithms, regression, classification

В работе рассмотрены три вида искусственного интеллекта, а также обозначены несомые им проблемы и угрозы для социума. Авторы детально разбирают наиболее популярные алгоритмы машинного обучения, особенно выделяя плюсы и минусы этих методов. Основной акцент в статье делается на методах машинного обучения искусственного интеллекта, дается математическое обоснование этих методов.

**Ключевые слова:** искусственный интеллект, символизм, коннекционизм, машинное обучение, нейронные сети, алгоритмы, регрессия, классификация

**Владимир Афанасьевич Артамонов,**  
доктор технических наук, профессор,  
академик МАИТ  
artamonov@itzashita.ru

**Елена Владимировна Артамонова,**  
кандидат технических наук, член МАИТ  
admin@itzashita.ru

**Александр Евгеньевич Сафонов,**  
магистр информационных технологий  
mail@iwebi.group

Международная академия  
информационных технологий (МАИТ)

## Введение

С момента введения в 1956 году Джоном МакКарти понятия «искусственный интеллект» (ИИ) для описания науки изучения разума путем воссоздания его ключевых признаков на компьютере, дальнейшее развитие получило сформировавшееся ранее среди философов новое научное направление – **когнитивистика**<sup>1</sup>. Когнитивистика объединяет, в частности, теорию познания, когнитивную психологию, нейрофизио-

логию, когнитивную лингвистику, невербальную коммуникацию и теорию искусственного интеллекта. По мере развития кибернетики и появления первых компьютеров идея аналогии человеческого разума и вычислительной машины начала выходить как прорывная в основе миропознания и во многом заложила основы теории когнитивизма. Процесс мышления сравнивался с работой компьютера, который получает стимулы из окружающего мира и генерирует информацию, доступную для наблюдения. Помимо символов как результатов контакта разума с внешним миром объектом исследований стали мыслительные образы (или представления). Таким образом, произошло разделение на «внешние» (предметы, объекты и пр.) и «внутренние» представления.

На вопрос, существует ли мир, когнитивная наука отвечает: «Неизвестно, но существуют наши представления об этом мире». Однако когнитивизм также вернул декартовский скептицизм и оставил без внимания субъективные переживания

<sup>1</sup> Когнитивистика (Cognitive Science) – наука о познании.

и эмоции. В когнитивной науке используются два стандартных вычислительных подхода к моделированию когнитивных систем: символизм (классический подход) и коннекционизм (более поздний подход).

*Символизм* основывается на предположении о том, что человеческое мышление подобно мышлению компьютера с центральным процессором, последовательно обрабатывающим единицы символической информации.

*Коннекционизм* основывается на предположении, что человеческое мышление не может быть уподоблено центральному цифровому процессору из-за несовместимости с данными нейробиологии, а может имитироваться при помощи искусственных нейронных сетей, которые состоят из «формальных» нейронов, выполняющих параллельную обработку данных.

Прогресс в когнитивистике, как полагают ученые, позволит «разгадать загадку разума», то есть описать и объяснить протекающие в мозгу человека процессы, ответственные за высшую нервную деятельность. Это позволит создать системы так называемого «сильного» и «суперсильного» искусственного интеллекта, который будет обладать способностями к самостоятельному обучению, творчеству, свободному общению с человеком.

Говоря формальным языком, искусственный интеллект – это область компьютерной науки, ориентированная на создание «интеллектуальных» компьютеров и машин, имитирующих человеческие действия и реакции через машинное обучение (МО)<sup>2</sup>, распознавание речи и решение задач.

На сегодняшний день различают три вида искусственного интеллекта.

1. *Слабый (Narrow AI)* – это ИИ в сегодняшнем понимании. Он за-

программирован на выполнение какой-либо конкретной задачи, будь то мониторинг погоды, управление роботами или анализ корпоративных данных. Такой ИИ может работать в режиме реального времени, но извлекает информацию лишь из ограниченного набора данных. В результате эта система способна справляться только с одной проблемой, решению которой она обучена.

2. *Сильный ИИ (General AI)* схож с человеческим интеллектом. Иными словами, он может успешно выполнять любые умственные задачи и обладать когнитивными способностями, в том числе сознанием, которые пока под силу только людям.

3. *Суперсильный ИИ (Super AI)*, по определению философа Ника Бострома (Университет Оксфорд), «это любой интеллект, который значительно превосходит когнитивные способности человека практически во всех областях». Такой ИИ превзойдет нас во всех аспектах: от творчества до жизненной мудрости и решения научных проблем. Машинные будут способны демонстрировать интеллект, который мы не видели даже у самых одаренных представителей человечества. Появление именно этого типа ИИ беспокоит многих представителей общественности, а по мнению Илона Маска, именно он приведет к вымиранию людей как вида.

Тем не менее, до настоящего времени не существует единой концепции (парадигмы) анализа и синтеза систем искусственного интеллекта, что породило массу мифов и догматических толкований этого научного направления [1].

*Следует заметить, что в английском языке словосочетание «artificial intelligence» (именно это понятие ввел МакКарти) не имеет антропоморфной окраски, которую оно приобрело в традиционном русском переводе: сло-*

*во «intelligence» в используемом контексте скорее означает «умение рассуждать разумно», а вовсе не «интеллект» (для которого есть английский аналог «intellect»).*

В философии не решен вопрос о природе и статусе человеческого интеллекта. Нет и точного критерия достижения компьютерами «разумности», хотя на заре искусственного интеллекта был предложен ряд гипотез, например, тест *Тьюринга*<sup>3</sup> или гипотеза Ньюэлла – *Саймона*<sup>4</sup>.

Создание с помощью рукотворного оборудования разумной системы вместо нашего собственного мозга в виде клеток и тканей должно было стать иллюстрацией полного понимания этой проблемы и повлечь за собой практическое воплощение идеи в виде создания умных устройств или даже роботов.

Выдающиеся советские математики Андрей Николаевич Колмогоров и Владимир Игоревич Арнольд доказали в 1957 году ряд теорем о том, что любая непрерывная функция нескольких переменных может быть представлена в виде комбинации конечного числа функций меньшего числа переменных, и именно это стало математическим обоснованием для построения нейронных сетей (НС) [2, 3]. Было доказано, что соответствие между зависимыми элементами различных множеств или функций может быть представлено нейросетью фиксированной размерности с прямыми связями с определенным количеством «нейронов» входного слоя, увеличенным числом «нейронов» каждого следующего скрытого слоя с определенными функциями активации и «нейронами» выходного слоя с неизвестными функциями активации. Причем, НС могут настраиваться или «обучаться».

*Для человека, мало знакомого с математическими теориями, все это*

<sup>2</sup> Машинное обучение (Machine Learning, ML) – свод методов и правил в области искусственного интеллекта, набор алгоритмов и программ, которые применяют, чтобы создать машину, которая учится на собственном опыте.

<sup>3</sup> Тест Тьюринга – эмпирический тест, идея которого была предложена Аланом Тьюрингом в 1950 году. Тьюринг задался целью определить, может ли машина мыслить. Стандартная интерпретация этого теста звучит следующим образом: «Человек взаимодействует с одним компьютером и одним человеком. На основании ответов на вопросы он должен определить, с кем он разговаривает: с человеком или компьютерной программой. Задача компьютерной программы – ввести человека в заблуждение, заставив сделать неверный выбор».

<sup>4</sup> Гипотеза была сформулирована Алленом Ньюэллом и Гербертом Саймоном в 1976 году. Основанием для гипотезы стало успешное применение созданной ими программы – универсального решателя задач – для моделирования рассуждений человека.

звучит несколько сложно, но имеет принципиальное значение для ответа на вопросы: возможно ли создать искусственный интеллект и подлежит ли он последующей настройке и машинному обучению?

## Машинное обучение как когнитивное начало искусственного интеллекта

Классическое машинное обучение (Classical Machine Learning) строится на классических статистических алгоритмах и решает вопросы, связанные с принятием решений на основе данных. В классическом машинном обучении с учителем специалист по ИИ, обучающий систему, размечает данные, задает машине определенные примеры и наблюдает за ее прогрессом. Задачами, которые решаются при помощи обучения с учителем, являются, например, классификация и регрессия. Машинное обучение без учителя включает в себя следующие типы: кластеризация, обобщение, поиск правил. Эти алгоритмы часто применяются в технологиях интеллектуального анализа данных (Data Mining) [4] и науке о данных (Data Science) [5].

Далее с позиций когнитивизма более подробно рассмотрим некоторые классы задач, решаемых с помощью МО.

1. **Классификация** – наиболее популярная задача машинного обучения. Она в чем-то схожа с тем, как ребенок учится определять форму и размер предметов, складывая их в отдельные кучки.

**Задача классификации:** предсказание категории объекта и разделение объектов согласно определенным и заданным заранее признакам. То есть машина сортирует данные по нужным категориям: одежду – по цветам, сезонам или ткани, книги – по жанрам, авторам, языкам написания, информационные ресурсы – по степени их защищенности, письма – по личной или рабочей

направленности, спам-составляющей и т. д.

2. **Регрессия** – это метод, когда по заданному набору признаков необходимо спрогнозировать некоторую целевую переменную.

**Задача регрессии:** предсказание места на числовой прямой. Например, загруженность дорог в зависимости от времени суток и время на путь из пункта А в пункт Б в зависимости от пробок. Или каким будет объем рынка определенных товаров через два года. И даже скорость развития определенной болезни при общих показателях здоровья человека. Поскольку регрессия запрограммирована на работу с числами, ее встраивают в различные вычислительные системы, даже в классический Excel.

**Примечание:** систему классификации можно «доучить» до способности решать задачи регрессии. На практике это выглядит как понимание не только класса объекта, но и его близости к тому или иному показателю. Например, яблоко свежее или испорченное, а если ближе к испорченному, то на сколько процентов.

3. **Кластерный анализ** – задача разбиения заданной выборки данных (объектов) так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров значительно отличались друг от друга.

**Задача кластеризации:** используя все имеющиеся данные, предсказать соответствие объектов выборки их классам, сформировав таким образом кластеры. Кластеризацию применяют для анализа и поиска признаков, по которым можно объединить объекты, сжатия данных и поиска новизны (что не входит ни в один кластер). В чем отличие классификации от кластеризации? При классификации у вас есть набор определенных классов, вы учите машину на наборе примеров и потом хотите знать, к какому классу принадлежит новый объект. При кластеризации вы используете алгоритм, который пытается сгруппировать

набор объектов и определить, существует ли какая-либо взаимосвязь между ними, то есть машина учится сама.

4. **Поиск ассоциативных правил** – метод, который активно используется в маркетинге для изучения поведения покупателя и составления типичного шаблона покупок.

**Задача поиска ассоциативных правил:** находить закономерности в потоке данных. Например, для анализа паттернов поведения<sup>5</sup> пользователей на web-сайтах или для правильной расстановки товаров на полках в магазине. Ведь не зря жевательные резинки, бритвенные приборы и батарейки расположены в прикассовой зоне, а сладости недалеко от чая/кофе.

## Алгоритмы машинного обучения

В теории МО существует такое понятие, как теорема No Free Lunch, которую именуют «**фольклорной**»<sup>6</sup>. Ее суть заключается в том, что не существует такого алгоритма, который оказался бы лучшим выбором для каждой задачи, что в особенности касается обучения с учителем [6]. Например, нельзя сказать, что нейронные сети всегда работают лучше, чем деревья решений, и наоборот.

На эффективность алгоритмов влияет множество факторов вроде размера и структуры набора данных. По этой причине приходится опробовать разные алгоритмы, проверяя эффективность каждого на тестовом наборе данных, и затем выбирать лучший вариант. Само собой разумеется, нужно выбирать среди алгоритмов, соответствующих решаемой задаче. Если проводить аналогию с бытовыми делами, то при уборке дома вы, скорее всего, будете использовать пылесос, метлу или швабру, но никак не лопату.

Алгоритмы машинного обучения можно описать как обучение целевой функции  $F$ , которая наилучшим

<sup>5</sup> Паттерн поведения – это привычка действовать определенным образом в определенной ситуации. По сути, все наше поведение можно свести к набору паттернов. Какие-то из них могут быть продуктивными, какие-то – деструктивными. Первые проявляются в адекватных отношениях между людьми, вторые провоцируют конфликты и мешают человеку воплощать себя и реализоваться в мире.

<sup>6</sup> No Free Lunch – в переводе с англ. означает «нет бесплатного обеда». Соответствует русскому выражению «бесплатный сыр бывает только в мышеловке».

образом соотносит входные переменные  $X$  и выходную переменную  $Y$ :  $Y = F(X)$ . Мы не знаем, что представляет собой функция  $F$ . Ведь если бы знали, то использовали бы ее напрямую, а не пытались обучить с помощью различных алгоритмов. Наиболее распространенной задачей в машинном обучении является предсказание значений  $Y$  для новых значений  $X$ . Это называется прогнозированием, и наша цель – сделать как можно более точное предсказание.

Далее представляем вашему вниманию краткий обзор наиболее популярных алгоритмов, используемых в машинном обучении.

### Наивный Байесовский классификатор

Наивный байесовский классификатор<sup>7</sup> – простой, но удивительно эффективный алгоритм. Модель состоит из двух типов вероятностей, которые рассчитываются с помощью тренировочных данных: вероятности каждого класса и условной вероятности для каждого класса при каждом новом значении  $x$ . После расчета вероятностной модели ее можно использовать для предсказания с новыми данными при помощи формулы Байеса<sup>8</sup> (см. врезку).

Искомая апостериорная вероятность выражается через априорную вероятность (*prior probability*) класса  $P(y = k)$  (смысл очевиден: если дан случайный объект, то с какой вероятностью он принадлежит классу  $k$ ) и вероятностью или плотностью (*density*) распределения объектов класса  $P(x|y = k)$ . Смысл  $P(x|y = k)$  тоже понятен: если мы будем рассматривать только объекты класса  $k$ , то они тоже распределены, а именно, дискретное распределение описывается вероятностями, а непрерывное – плотностями. В приведенной формуле заключена одна из таких функций, в которую мы подставили  $x$  (признаковое описание). В знамена-

**Врезка**

$$p_k(x) = P(y = k|x) = \frac{P(x|y = k) \cdot P(y = k)}{P(x)}$$

теле формулы стоит  $P(x)$  – это уже вероятность/плотность распределения всех объектов, но, по сути, она не очень нужна: выбор максимальной вероятности принадлежности к классам не зависит от этого значения. При наличии вещественных данных и, предполагая нормальное распределение, рассчитать эти вероятности не составляет особой сложности. Вкратце можно сказать, что результат теоремы Байеса – это возможность определения вероятности из знания других вероятностей, что и лежит в основе решения задач классификации.

### Метод опорных векторов

Метод опорных векторов (*Support Vector Machines, SVM*) [7] – это набор контролируемых методов обучения, используемых для классификации, регрессии и обнаружения выбросов. Основная идея метода – перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Две парал-

лельных гиперплоскости строятся по обеим сторонам гиперплоскости, разделяющей классы. Разделяющей будет гиперплоскость, максимизирующая расстояние до двух параллельных гиперплоскостей. Алгоритм работает в предположении, что чем больше разница или расстояние между этими параллельными гиперплоскостями, тем меньше будет средняя ошибка классификатора.

Гиперплоскости – это границы решений, которые классифицируют данные. В методе опорных векторов гиперплоскость выбирается так, чтобы наилучшим образом разделять точки в плоскости входных переменных по их классу: 0 или 1. В двумерной плоскости это можно представить как линию, которая полностью разделяет точки всех классов (рис. 1). Во время обучения алгоритм ищет коэффициенты, которые помогают лучше разделять классы гиперплоскостью.

Расстояние между гиперплоскостью и ближайшими точками данных называется разницей. Лучшая или оптимальная гиперплоскость,

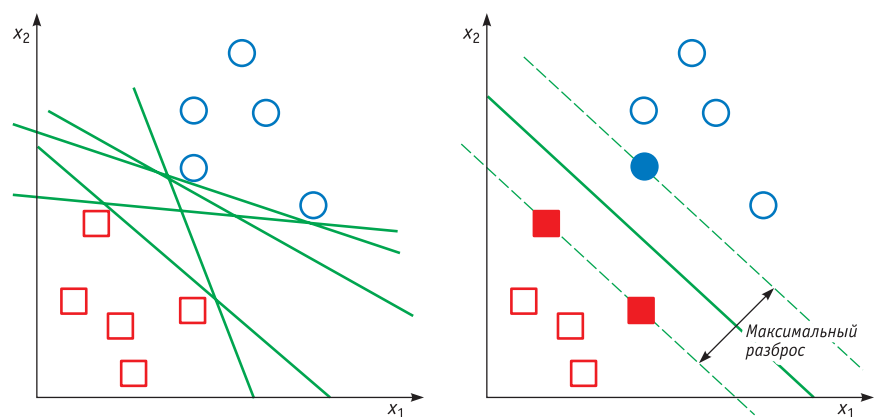


Рис. 1. Оптимальная разделяющая гиперплоскость для метода опорных векторов

<sup>7</sup> В различных библиотеках алгоритмов МО, скорее всего, идет речь о реализации наивного байесовского классификатора (*naive Bayes*). Связано это со сложностью оценки плотностей вероятностей. Вероятность класса можно оценить пропорцией, в которой представители этого класса есть в обучающей выборке. А вот плотность распределения в классе в многомерном пространстве сложно оценить надежно из-за неопределенности размерности. Поэтому делают сильное (наивное) предположение, что все признаки независимы и тогда плотность представляется в виде произведения плотностей по отдельным признакам.

<sup>8</sup> Правильно фамилия ученого (родился в 1701 году) произносится Бейз, то есть надо бы говорить «формула Бейза». В том виде, в котором мы знаем формулу Байеса, она принадлежит его современнику Лапласу.



разделяющая два класса, – это линия с наибольшей разницей (максимальным разбросом). Только эти точки, называемые опорными векторами, имеют значение при определении гиперплоскости и при построении классификатора. Для определения значений коэффициентов, максимизирующих разницу, используются специальные алгоритмы оптимизации. Метод опорных векторов, наверное, один из самых эффективных классических классификаторов, на который определенно стоит обратить внимание; он также входит в число наиболее популярных и обсуждаемых алгоритмов машинного обучения. Перечислим достоинства этого метода:

- эффективен в пространствах больших размеров;
- эффективен в случаях, когда количество измерений превышает количество образцов;
- использует подмножество обучающих точек в функции принятия решений (называемых опорными векторами), поэтому также эффективен с точки зрения памяти;
- универсален: для функции принятия решения могут быть указаны различные функции ядра (предоставляются общие ядра, но также можно указать собственные).

К недостаткам метода можно отнести следующие моменты:

- если количество функций намного превышает количество выборок, следует избегать чрезмерной подгонки результатов при выборе функций ядра, и тогда процедура регуляризации имеет решающее значение;
- SVM не предоставляют напрямую оценки вероятностей: они рассчитываются с использованием слож-

ной пятикратной перекрестной проверки.

### Линейная регрессия

Линейная регрессия (*Linear Regression*) [8] – пожалуй, один из наиболее известных и понятных алгоритмов в статистике и машинном обучении, используемый для прогнозного моделирования событий и объектов. Соответственно, такая хорошо подходит в качестве первого алгоритма для изучения основ МО.

Прогнозное моделирование, в первую очередь, касается минимизации ошибки модели или, другими словами, как можно более точного прогнозирования. Оно заимствует алгоритмы из разных областей, включая статистику, и использует их в этих целях.

Линейную регрессию можно представить в виде уравнения, которое описывает прямую, наиболее точно показывающую взаимосвязь между входными переменными  $X$  и выходными переменными  $Y$ . Для составления этого уравнения нужно найти определенные коэффициенты  $B$  для входных переменных, например,  $Y = B_0 + B_1 \cdot X$  (рис. 2).

Зная  $X$ , мы должны найти  $Y$ , и цель линейной регрессии заключается в поиске значений коэффициентов  $B_0$  и  $B_1$ .

Для оценки регрессионной модели используются различные методы линейной алгебры или метода наименьших квадратов. Линейная регрессия определяет корреляцию между переменными.

### Логистическая регрессия

Логистическая регрессия (*logistic или logit regression*) [8] – еще один алгоритм, пришедший в машинное

обучение прямо из статистики. Ее хорошо использовать для задач бинарной классификации (это задачи, в которых на выходе мы получаем один из двух классов).

Логистическая регрессия похожа на линейную тем, что в ней тоже требуется найти значения коэффициентов для входных переменных. Разница заключается в том, что выходное значение преобразуется с помощью нелинейной или логистической функции. Логистическая функция выглядит как прописная буква  $S$  и преобразовывает любое значение в число в пределах от 0 до 1. Это весьма полезно, так как мы можем применить правило к выходу логистической функции для привязки к 0 и 1 (например, если результат функции меньше 0,5, то на выходе получаем 1) и предсказания класса (рис. 3). Логистическая регрессия выдает ответ в виде числа в промежутке от 0 до 1. Если число ниже определенного порога значения – то объект относится к первому классу, а если выше – то ко второму. Калибровка порогового значения для разделения объектов на классы подбирается в ходе калибровки алгоритма (см. рис. 3).

Благодаря тому, как обучается модель, предсказания логистической регрессии можно использовать для отображения вероятности принадлежности образца к классу 0 или 1. Это полезно в тех случаях, когда нужно иметь больше обоснований для прогнозирования. Как и в случае с линейной регрессией, логистическая регрессия выполняет свою задачу лучше, если убрать лишние и похожие переменные. Модель логистической регрессии быстро обучается и хорошо подходит для задач бинарной классификации.

### Линейный дискриминантный анализ

Логистическая регрессия используется, когда нужно отнести образец к одному из двух классов. Если классов больше, чем два, то лучше использовать алгоритм линейного дискриминантного анализа (*Linear Discriminant Analysis, LDA*) [9] (рис. 4). Метод используется в статистике, распознавании образов и МО для

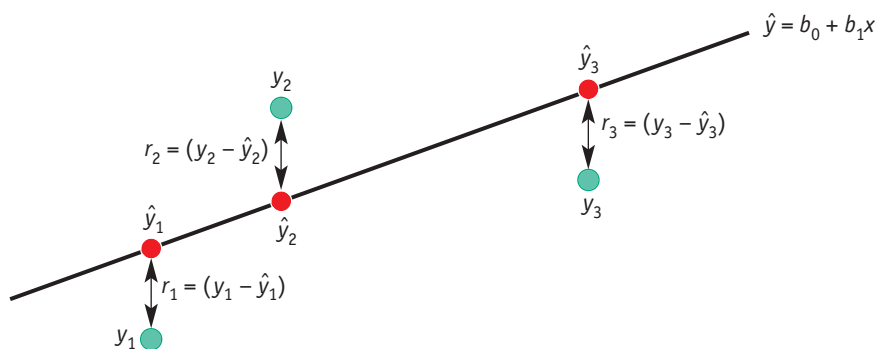


Рис. 2. Графическая модель линейной регрессии

поиска линейной комбинации признаков, которая описывает или разделяет два или более классов или событий. Получившаяся комбинация может быть использована как линейный классификатор, или, что более часто, для снижения размерности перед классификацией. Представление LDA довольно простое. Оно состоит из статистических свойств данных, рассчитанных для каждого класса. Для каждой входной переменной это включает:

- среднее значение для каждого класса;
- дисперсию, рассчитанную по всем классам.

Предсказания производятся путем вычисления дискриминантного значения для каждого класса и выбора класса с наибольшим значением. Предполагается, что данные имеют нормальное распределение, поэтому перед началом работы рекомендуется удалить из них аномальные значения. Это простой и эффективный алгоритм для задач классификации.

**Деревья принятия решений**

Дерево решений (*Decision Tree*) [8] можно представить в виде двоичного дерева, знакомого многим по алгоритмам и структурам данных. Каждый узел представляет собой входную переменную и точку разделения для этой переменной (при условии, что переменная – число).

Листовые узлы – это выходная переменная, которая используется для предсказания. Предсказания производятся путем прохода по дереву к листовому узлу и вывода значения класса на этом узле (рис. 5).

Деревья быстро обучаются и дают предсказания. Кроме того, они точны для широкого круга задач и не требуют особой подготовки данных. Ограничения метода: далеко не все модели можно описывать деревьями.

**Алгоритм случайного леса**

Алгоритм случайного леса (*Random Forest*) [8] является продолжателем метода деревьев решений. Из простых деревьев строится группа (лес), каждое дерево немного отличается от своих собратьев. Деревья

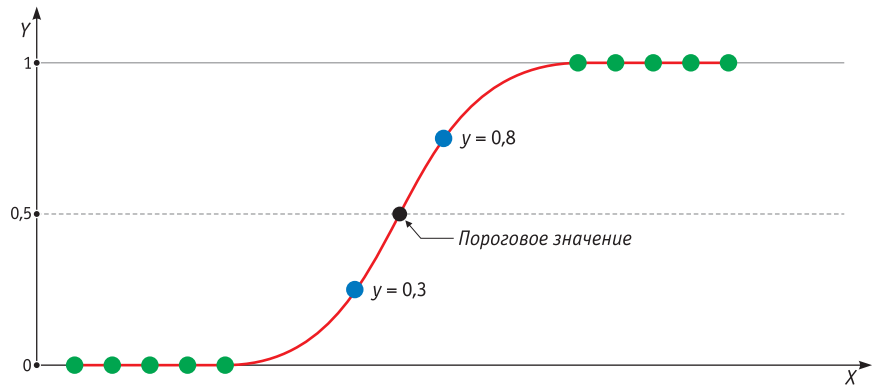


Рис. 3. Логистическая функция

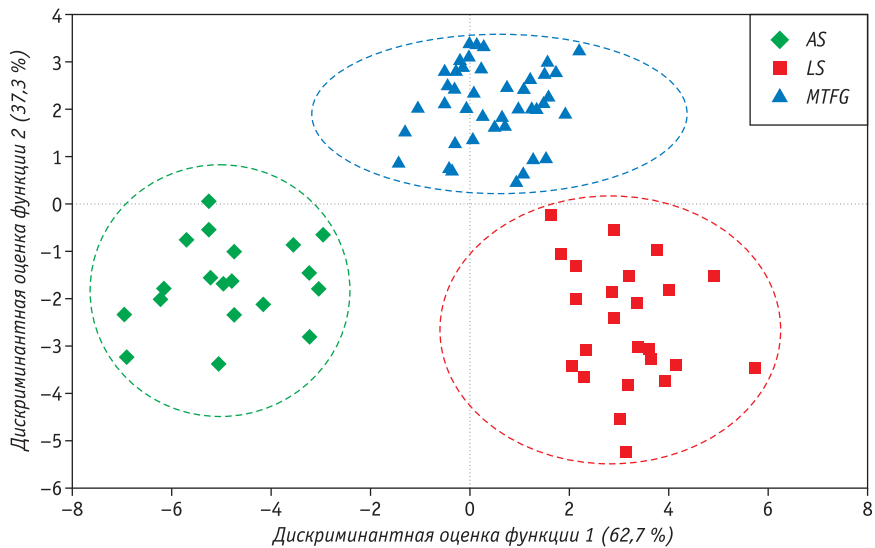


Рис. 4. Дискриминантный анализ для трех классов событий

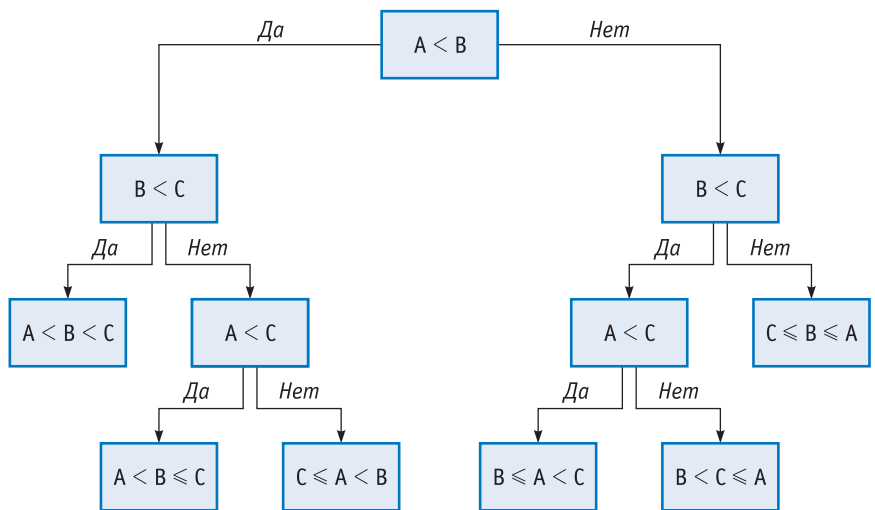


Рис. 5. Представление алгоритма в виде карты возможных результатов из ряда взаимосвязанных выборов

в лесу голосуют за те или иные варианты решений, и самое часто встречающееся решение становится ответом системы (рис. 6).

Области применения алгоритма такие же, где применимы простые

решающие деревья: для предсказаний на основе параметров и для классификации.

Объединение простых алгоритмов в группы часто дает впечатляющие результаты. Казалось бы, при-

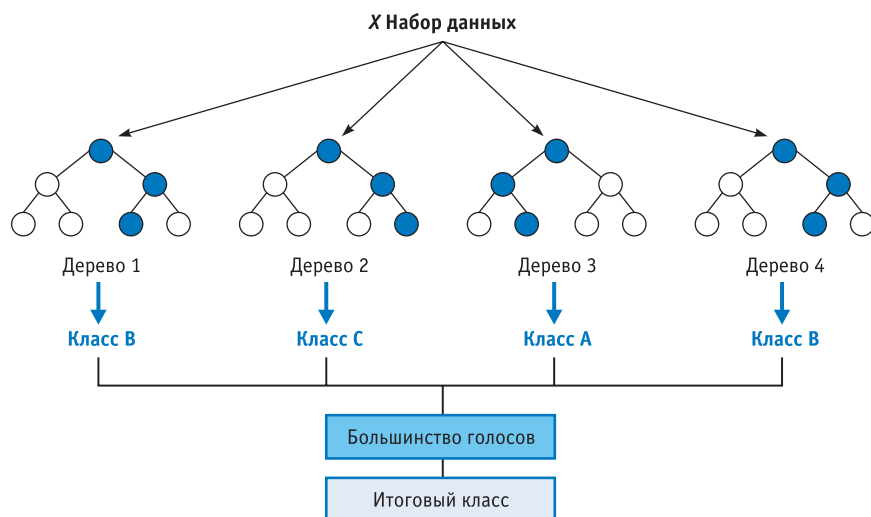


Рис. 6. Алгоритм случайного леса выдает самое популярное решение деревьев

митивные по своей сути решающие деревья вместе способны давать ответы на весьма сложные вопросы. Однако, как и у традиционных решающих деревьев, сфера применения случайного леса весьма ограничена.

### Нейронные сети

Нейронная сеть (*Neural Network*) [10] – это последовательность нейронов, соединенных между собой синапсами. Структура нейронной сети пришла в мир ИИ напрямую из биологии. Благодаря такой структуре, машина обретает способность анализировать и даже запоминать различную информацию. Нейронные сети также могут 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. Нейронные сети используются для решения сложных задач, которые требуют аналитических вычислений, подобных тем, что совершает человеческий мозг.

Самыми распространенными сферами применения нейронных сетей являются классификация, предсказание и распознавание.

**Классификация** – то есть распределение данных по параметрам. Например, на вход дается набор людей и нужно решить, кому из них давать кредит, а кому нет. Эту работу может сделать нейронная сеть, анализируя

такую информацию как возраст, платежеспособность, кредитная история и т. д.

**Предсказание** – возможность предсказывать следующий шаг. Например, рост или падение акций, основываясь на ситуации на фондовом рынке.

**Распознавание** – в настоящее время самая широкая сфера применения нейронных сетей. Используется, например, в поисковых сервисах при поиске изображений или в камерах телефонов при определении положения и выделении лица владельца, а также во многих других случаях.

Чтобы понять принцип работы нейронной сети, необходимо взглянуть на ее составляющие и их параметры.

**Нейрон** – это вычислительная единица, которая получает информацию, производит над ней простые вычисления и передает ее дальше. Нейроны делятся на три основных типа (рис. 7а): входной (крайний левый), скрытый (посредине) и выходной (крайний правый). В том случае, когда нейросеть состоит из большого количества нейронов, вводят термин слой. Соответственно, есть входной слой, который получает информацию,  $n$  скрытых слоев (обычно их не больше трех), которые ее обрабатывают, и выходной слой, выводящий результат. У каждого из нейронов есть два основных параметра: входные данные (*input data*) и выходные данные (*output data*). В случае входного нейрона:  $input = output$ . В осталь-

ных – в поле *input* попадает суммарная информация всех нейронов с предыдущего слоя, после чего она нормализуется с помощью функции активации (пока что просто представим ее как  $f(x)$ ) и попадает в поле *output*.

Важно помнить, что нейроны оперируют числами в диапазоне  $[0, 1]$  или  $[-1, 1]$ . А как же тогда обрабатывать числа, которые выходят за пределы указанного диапазона? Самый простой ответ – разделить 1 на это число. Такой процесс называется нормализацией, и он очень часто используется в нейронных сетях.

**Синапс** – это связь между двумя нейронами. У синапсов есть один параметр – вес  $w$ . Благодаря ему, входная информация изменяется, когда передается от одного нейрона к другому. Допустим, есть 3 нейрона, которые передают информацию следующему. Тогда у нас есть 3 веса, соответствующие каждому из этих нейронов. Доминирующей в следующем нейроне будет информация от того нейрона, у которого вес больше. Совокупность весов нейронной сети или матрица весов – это своеобразный мозг всей системы. Именно благодаря этим весам входная информация обрабатывается и превращается в результат. Важно помнить, что во время инициализации нейронной сети веса расставляются в случайном порядке. Как это все работает, рассмотрим на примере, изображенном на рис. 7б.

На нем представлена часть нейронной сети, где входные нейроны обозначены как  $I$ , скрытый нейрон как  $H$ , а буквой  $w$  – их веса. Из формулы видно, что входная информация – это сумма всех входных данных, умноженных на соответствующие им веса. Тогда дадим на вход 1 и 0. Пусть  $w_1 = 0,4$  и  $w_2 = 0,7$ . Входные данные нейрона  $H_1$  будут следующими:  $1 \cdot 0,4 + 0 \cdot 0,7 = 0,4$ .

Теперь, когда у нас есть входные данные, мы можем получить выходные данные, подставив входное значение в функцию активации. Функция активации – это способ нормализации входных данных. То есть, если на входе у нас будет большое число, пропустив его через функцию активации, получаем выходные дан-

ные в нужном диапазоне и передаем их дальше. Указанное действие мы повторяем для всех слоев, пока не дойдем до выходного нейрона.

Запустив такую сеть в первый раз, мы увидим, что ответ далек от правильного, потому что сеть не натренирована. Чтобы улучшить результаты мы будем ее тренировать (обучать). Следует заметить, что процессом обучения нейронных сетей, как правило, занимаются специалисты, обладающие компетенциями в областях Data Mining и Data Science. МО нейронных сетей часто требует высокой вычислительной мощности, настолько высокой, что на обычном ноутбуке этот процесс может занять годы. Иногда требуется применение специализированного вычислительного оборудования, например видеокарт, хотя всегда есть возможность взять видеокарту в аренду в облаке. Для отладки и настройки алгоритма требуется большое число тестовых данных. Настройка сложной модели – процесс трудоемкий, требующий соответствующей компетенции и опыта. Однако, несмотря на минусы, альтернатив этому алгоритму мало, а облачные технологии позволяют использовать для его работы почти неограниченные мощности.

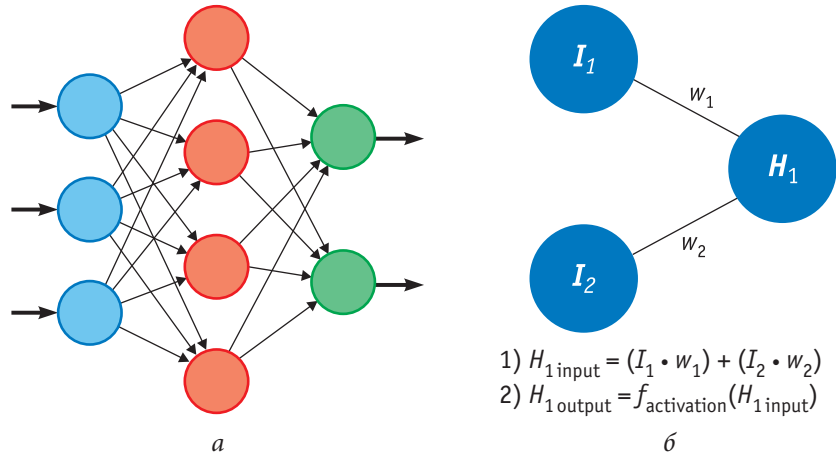


Рис. 7. Обобщенная структура нейросети

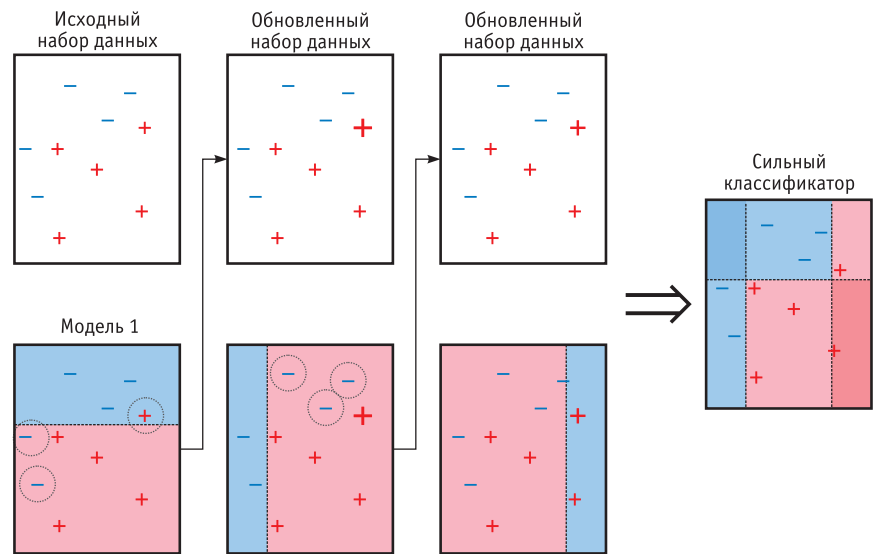


Рис. 8. Обобщенная модель бустинга

**Бустинг (Boosting)**

Бустинг (*boosting*), в переводе с английского «усиление» – это композиционный метаалгоритм машинного обучения, применяемый, главным образом, для уменьшения смещения (погрешности оценки), а также дисперсии в обучении с учителем. Также он определяется как семейство алгоритмов машинного обучения, преобразующих слабые обучающие алгоритмы в сильные. Суть метода [8] заключается в том, что сильный классификатор создается на основе слабых: каждая новая модель учится на ошибках предыдущей (рис. 8). То есть каждый раз добавляются все новые и новые модели, которые пытаются исправить ошибки своих предшественников. Так продолжается до тех пор, пока прогнозы не станут безошибочными, либо не исчерпается лимит на количество моделей. Этот подход часто дает не-

ожиданно точные и ценные результаты, и поэтому его стоит применять для самых разных задач.

Минус метода: модели могут быть весьма большими, поэтому в бустинг необходимо включать наборы из других моделей, что усложняет построение итоговой системы.

**Метод K-ближайших соседей**

Метод K-ближайших соседей (*K-nearest Neighbors, KNN*) – простой и очень эффективный алгоритм [8]. Модель KNN представлена всем набором тренировочных данных. Предсказание для новой точки делается путем поиска K ближайших соседей в наборе данных и суммирования выходной переменной для этих K-экземпляров (рис. 9). Вопрос лишь в том, как определить сходство между экземплярами данных. Если все признаки имеют один и тот же мас-

штаб (например, сантиметры), то самый простой способ заключается в использовании евклидова расстояния – числа, которое можно рассчитать на основе различий с каждой входной переменной.

KNN зачастую требует много памяти для хранения всех данных, но

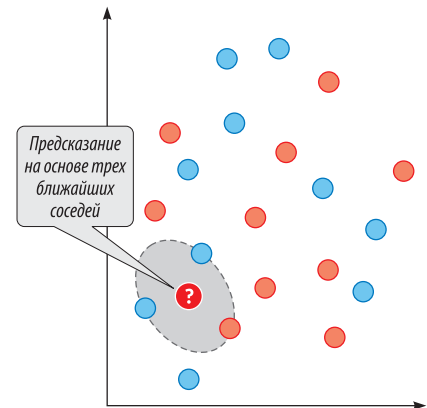


Рис. 9. Идея метода KNN



зато быстро делает предсказание. Обучающие данные подлежат обновлению, чтобы предсказания оставались точными с течением времени. Идея ближайших соседей может плохо работать с многомерными данными (множеством входных переменных), что негативно скажется на эффективности алгоритма при решении задачи. Это называется проклятием размерности. Таким образом, стоит использовать лишь наиболее важные для предсказания переменные.

Недостаток KNN заключается в необходимости хранить весь тренировочный набор данных.

### Сети векторного квантования (LVQ)

Если KNN показал себя хорошо, но имеются ограничения по памяти, то есть смысл попробовать алгоритм сетей векторного квантования (*Learning Vector Quantization, LVQ*), который лишен этого недостатка (рис. 10)

LVQ представляет собой набор кодовых векторов. Вначале они выбираются случайным образом и в течение определенного количества итераций адаптируются так, чтобы наилучшим образом обобщить весь набор данных. После обучения эти векторы могут использоваться для предсказания так же, как это делается в KNN. Алгоритм ищет ближайшего соседа (наиболее подходящий кодовый вектор) путем вычисления расстояния между каждым кодовым вектором и новым экземпляром дан-

ных. Затем для наиболее подходящего вектора в качестве предсказания возвращается класс (или число в случае регрессии). Лучшего результата можно достичь, если все данные будут находиться в одном диапазоне, например, от 0 до 1.

*Еще раз подчеркнем, что нами были рассмотрены далеко не все алгоритмы машинного обучения. Здесь собраны те из них, которые наиболее часто используются для решения основных задач классификации и прогнозирования.*

### Заключение

В данной работе мы рассмотрели основные когнитивные аспекты искусственного интеллекта, сфокусировав свое внимание на задачах и алгоритмах машинного обучения. Следует заметить, что все рассмотренное всецело применимо в основном к классу ИИ, именуемому «слабым» – Narrow AI – и ориентированному на решение одной задачи, пусть и очень большой. Также констатируем и тот факт, что научным сообществом накоплен определенный опыт реализации этого AI в различных сферах промышленности, транспорта, медицины, образования, сельского хозяйства, военном деле и т. д.

Значительным положительным аспектом в плане эффективности разработки приложений для ИИ и МО является использование мультипарадигмального языка программирования Python, который являет-

ся одним из наиболее значительных фреймворков для указанных областей. Этот язык обладает двумя главными особенностями. Во-первых, использует императивное программирование в противоположность символическому. Императивная программа выполняет вычисления по мере того, как вводятся данные. Благодаря этому свойству программа становится более гибкой. Во-вторых, задействуется динамический вычислительный граф в противоположность статическому. Это означает, что во время выполнения система порождает структуру графа, более подходящую для реализации динамических нейронных сетей и некоторых других алгоритмов МО. Библиотека подпрограмм Python включает большой набор полезных переносимых функций, начиная с возможностей для работы с текстом и заканчивая средствами для написания приложений МО. Дополнительные возможности, такие как математическое моделирование, работа с оборудованием, написание web-приложений или разработка игр, могут реализовываться посредством обширного количества сторонних библиотек, а также интеграцией библиотек, написанных на C или C++. При этом и сам интерпретатор Python может интегрироваться в проекты, написанные на этих языках. Существует и специализированный репозиторий программного обеспечения, написанного на Python, – PyPI [11], предоставляющий средства для простой установки пакетов в операционную систему и ставший стандартом де-факто для Python.

Тем не менее, человечество стоит перед глобальной проблемой создания новых поколений ИИ, классов – *сильного ИИ (General AI)*, схожего с человеческим интеллектом, и *суперсильного ИИ (Super AI)*, превосходящего когнитивные способности человека практически во всех областях. Иными словами, речь идет о техническом воспроизводстве функционирования человеческого мозга путем моделирования работы нейронов головного мозга.

Нейрон состоит из ядра, тела нейрона и специальных отростков: нескольких дендритов, которые вос-

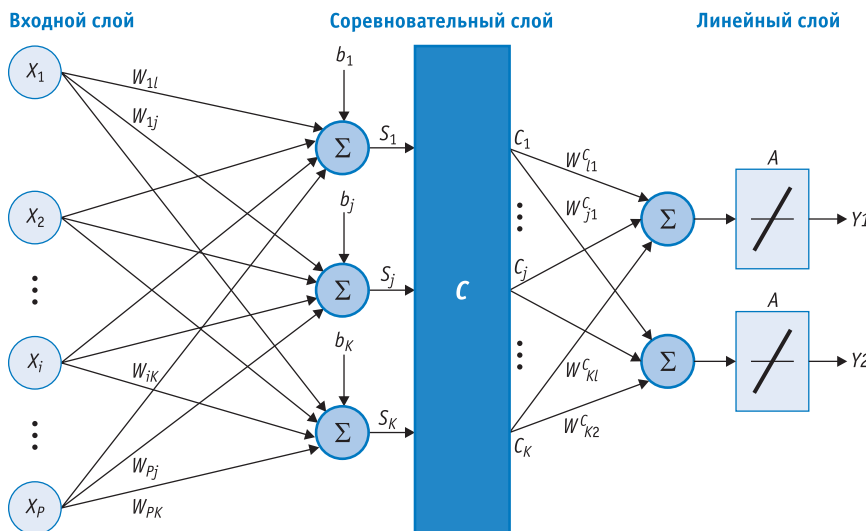


Рис. 10. Обобщенная схема алгоритма LVQ

принимают информацию, а также аксона, передающего сигналы другим нейронам или клеткам мышц. Каждый нейрон взаимодействует с другим нейроном посредством специальных электрических и химических сигналов, то есть является электрически возбудимой клеткой. Место контакта между двумя нейронами называется синапсом. Синапс представляет собой элемент информации, которая передается от нейрона к нейрону, образуя связи. Таких нейронов в нашем мозге больше 100 млрд, и они, соединяясь, образуют нейронную сеть. Сигналы в синапсах передаются с помощью химических нейромедиаторов (аминокислот и различных органических соединений) или посредством электрических сигналов, когда ионы кальция проходят по белковым каналам из одной клетки в другую.

Хотя механизмы передачи сигналов хорошо изучены, проблема, стоящая перед учеными и инженерами, связана с тем, что каждый из этих ста миллиардов нейронов образует от 1 до 20 тысяч соединений с другими нейронами, накапливая колоссальный объем информации (от 2 до 5 петабайт), а в процессе помимо электрических разрядов и ионов участвуют также сложные молекулы, способные как усиливать, так и ослаблять сигналы. По результатам визуальной реконструкции данных ученые обнаружили, что каждый синапс содержит около 1000 молекулярных «переключателей» наподобие аналоговых транзисторов. То есть отдельный синапс можно сравнить с микропроцессором. Получается, что количество «транзисторов» в человеческом мозге теперь нужно увеличить на три порядка. Их больше, чем реальных транзисторов во всех компьютерах на планете, включая маршрутизаторы, коммутаторы, промышленные контроллеры и другое ИКТ-оборудование вместе взятое [12].

Воспроизвести сеть из миллиардов клеток и тысяч постоянно возникающих и исчезающих между ними соединений не представляется реальным. Необходимо изобретать иную архитектуру со своими элементами. Проблема создания нового

поколения искусственного интеллекта усугубляется еще и тем, что человеческий мозг нельзя сравнивать с компьютером ни в коей мере. Мозг – не компьютер. Информация, воспоминания, опыт, образы, звуки и пр. не хранится в мозге в виде букв, цифр, нот, картинок или двоичных кодов. В мозге нет накопителей, с которыми работает центральный процессор, нет физических воспоминаний, которые извлекаются и обрабатываются алгоритмами. Информация восстанавливается нейронными связями тогда, когда нам это нужно. В мозге нет программного обеспечения. Человеческий мозг – это система, которая может обучаться, но при этом нам проще узнавать, чем запоминать.

Таким образом, перед человечеством стоит сложнейшая задача, сравнимая по экстремальности с открытиями XX века в ядерной физике и созданием ядерного оружия. Искусственный интеллект в его полном представлении пока не создан ни в одной стране мира. Есть лишь математические и компьютерные модели, имитирующие работу биологических нейронов. Нынешнее состояние информационных технологий и возможностей компьютерной техники не позволяет полностью реализовать подобные системы искусственного интеллекта.

Скорее всего, создателям нового поколения искусственного интеллекта придется обратиться к квантовой физике. Частицы, которые содержат информацию и участвуют в передаче сигнала – это молекулы и атомы, но при всей их микроскопичности взаимодействие получается настолько значительным, что людям, возможно, придется разрабатывать квантовую теорию мозга и использовать для создания искусственного интеллекта квантовые компьютеры, что само по себе является масштабной и очень сложной задачей.

Пока мы находимся довольно далеко от создания искусственного интеллекта нового поколения. Необходимы математические и физические модели сознания. Впереди у человечества много работы, и чем качественнее она будет сделана, тем выше

шансы людей дожить до следующего тысячелетия. ■

## ЛИТЕРАТУРА

1. Артамонов В. А., Артамонова Е. В. Искусственный интеллект: угроза или благо для человечества // Проблемы создания информационных технологий. Сб. научных трудов; [под ред. В. А. Сычика]. – Мн.: Государственное предприятие «Информационно-вычислительный центр Белстата». – 2022. – Вып. 32. – С. 5–17.
2. Колмогоров А. Н. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных // Известия АН СССР. – 1956. – Т. 108, № 2. – С. 179–182; английский перевод: Amer. Math. Soc. Transl., 17 (1961), p. 369–373.
3. Арнольд В. И. О функции трех переменных // Известия АН СССР. – 1957. – Т. 114, № 4. – С. 679–681; английский перевод: Amer. Math. Soc. Transl., 28 (1963), p. 51–54.
4. Кречетов Н. Продукты для интеллектуального анализа данных // Рынок программных средств. – 1997. – № 14–15. – С. 32–39.
5. Журавлева Е. Ю. Эпистемический статус цифровых данных в современных научных исследованиях // Вопросы философии. – 2012. – № 2. – С. 113–123.
6. Wolpert D. H., Macready W. G. No free lunch theorems for optimization // IEEE Transactions on Evolutionary Computation, 1, 67–82. DOI: org/10.1109/4235.585893/.
7. Вьюгин В. Математические основы теории машинного обучения и прогнозирования. – МЦМНО. – 2013. – 390 с.
8. 10 популярных алгоритмов машинного обучения [Электронный ресурс]. – URL: <https://mcs.mail.ru/blog/samye-populyarnye-algoritmy-mashinnogo-obucheniya> (дата обращения: 25.04.2022).
9. Hardle W., Simar L. Applied Multivariate Statistical Analysis. – Berlin Heidelberg: Springer, 2007.
10. Голубев Ю. Ф. Нейросетевые методы в мехатронике. – М.: Изд-во Моск. ун-та. – 2007. – 157 с.
11. Галерея лучших модулей Python; [перевод статьи Samuel Martins: Most useful modules every python developer should know] [Электронный ресурс]. – URL: <https://nuancesprog.ru/p/10672/> (дата обращения: 25.04.2022).
12. В человеческом мозге столько же «транзисторов», сколько их в мировой ИТ-инфраструктуре. [Электронный ресурс]. – URL: [https://www.cell.com/neuron/fulltext/S0896-6273\(10\)00766-X/](https://www.cell.com/neuron/fulltext/S0896-6273(10)00766-X/) (дата обращения: 25.04.2022).