

Силлимановские лекции

На средства, пожертвованные в память о Хепсе Эли Силлимане, президент и сотрудники Йельского университета ежегодно представляют цикл лекций, призванных обозначить присутствие Божие и Промысел Божий в том виде, в коем они проявляются в мире природы и нравственности. По искреннему убеждению завещателя, всякое упорядоченное изложение явлений природы или фактов истории способствует достижению этой цели более успешно, нежели догматическое или полемическое богословие, которое посему должно быть исключено из материалов лекций. Темы преимущественно выбираются из областей естественных наук и истории; особое внимание уделено астрономии, химии, геологии и анатомии.

Рэй Курцвейл

Предисловие к третьему изданию

Информационные технологии кардинально изменили жизнь человека, затронув каждую ее сторону — от бизнеса до политики и искусства. Сегодня, в силу стремительного роста мощности компьютеров, с одной стороны, и одновременного улучшения соотношения «производительность — цена» — с другой, информационная эра неумолимо продолжает расширять сферу своего влияния. Учитывая данное обстоятельство, можно утверждать, что важнейшим информационным процессом, требующим всестороннего изучения, является сам человеческий интеллект. Вероятно, эта книга представляет собой первое серьезное исследование взаимосвязи между мышлением человека и машиной — исследование с точки зрения математика, сформулировавшего фундаментальную архитектуру компьютерной эры.

В рамках грандиозного проекта по изучению механизмов работы мозга достигнуты значительные успехи: мы достаточно далеко продвинулись в обратном проектировании парадигм человеческого мышления и сегодня применяем их для создания более умных машин. Разумно ожидать, что искусственный интеллект (ИИ), построенный на базе данных принципов, в итоге воспарит над человеческим разумом. Однако, я полагаю, цель этого дерзкого предприятия состоит не в том, чтобы потеснить нас, а в том, чтобы расширить границы того, что уже сейчас можно смело назвать человеко-машинной цивилизацией. Именно это делает человеческий род поистине уникальным.

Итак, каковы же ключевые идеи, лежащие в основе информационной эры? По моему мнению, их пять. Джон фон Нейман выдвинул три из них, а также внес весомый вклад в четвертую. Клод Шеннон решил фундаментальную проблему надежности информации. Алан Тьюринг сформулировал идею универсальности машинных вычислений. Опираясь на Тьюринга и Шеннона, фон Нейман создал машину фон Неймана, которая была и остается фундаментальной архитектурой всех вы-

числительных машин, существующих на сегодняшний день.

В обманчиво скромном томике, который вы держите в руках, фон Нейман описывает предложенную им модель машинных вычислений и постулирует принципиальную эквивалентность живого мозга и искусственной машины. Он признает глубокие структурные различия, однако затем обращается к принципу эквивалентности всех вычислений, предложенному Тьюрингом, и предлагает рассматривать методы работы мозга как совокупность вычислений. Книга в высшей степени провидческая, особенно если учесть, что она была написана более полувека назад, когда нейронаука располагала лишь самыми примитивными инструментами. Фон Нейман предчувствовал быстрое развитие технологий и видел его неизбежные последствия в грядущей сингулярной трансформации человеческого существования.

Рассмотрим эти пять основных идей более подробно.

В 1940 году, услышав слово «computer», люди понимали, что речь идет об аналоговой вычислительной машине. В аналоговых машинах числа выражались разными уровнями

напряжения, а специализированные компоненты могли выполнять такие арифметические функции, как сложение и умножение. Главный недостаток подобных машин заключался в весьма низкой степени точности. Числа могли быть представлены с точностью до одной сотой, а так как уровни напряжения, представляющие эти числа, обрабатывались большим количеством арифметических операторов, ошибки имели тенденцию накапливаться. При выполнении достаточно длинных вычислений результаты оказывались в высшей степени неточными, а то и вообще бессмысленными.

Это явление знакомо всякому, кто хоть раз записывал музыку на магнитную ленту. Первая копия была немного шумнее оригинала (шум представляет собой случайные неточности). На второй копии (т. е. копии копии) шумов было еще больше, а десятая содержала один только шум.

Это наводит на мысль, что с той же проблемой столкнется и мир цифровых вычислительных машин. Рассмотрим передачу цифровой информации по некоему каналу. Поскольку идеальных каналов не существует, любому из них изначально будет присущ определенный

уровень ошибок. Допустим, мы выбрали канал с вероятностью правильной передачи каждого бита 0,9. Если я отправлю сообщение длиной в один бит, вероятность точной передачи его через этот канал будет равна 0,9. Допустим, я отправил 2 бита. Теперь точность составит $0,9^2 = 0,81$. А если я отправлю 1 байт (8 битов)? В таком случае вероятность, что я получу его правильно, окажется менее 0,5 (0,43, если быть точным). Вероятность точной отправки пяти байтов составит около 1 процента.

На первый взгляд наилучший способ обойти эту проблему — сделать канал более точным. Предположим, канал совершает всего одну ошибку на миллион битов. Если я отправлю файл размером в полмиллиона байт (примерно столько «весит» скромная программа или база данных), вероятность правильной его передачи составит менее 2 процентов, хотя изначально точность канала была весьма высока. Учитывая, что ошибка в один бит может испортить всю компьютерную программу, это не лучший выход. Вне зависимости от исходной точности канала вероятность ошибки стремительно возрастает с увеличением размера сообщения. Выходит, эта проблема неразрешима?

Аналоговые вычислительные машины тоже аккумулируют неточности, но если ограничиться небольшим набором вычислений, они оказываются весьма полезными. Цифровые машины, с другой стороны, требуют наличия постоянной связи не только между двумя разными машинами, но и между собственными составными частями. Так, память связана с центральным процессором, внутри которого происходит постоянный обмен данными между регистрами, а также между регистрами и арифметическим устройством. Внутри арифметического устройства информация передается от одного битового регистра к другому. Если считать, что частота ошибок стремительно возрастает с увеличением количества таких связей и что ошибка в один бит может нарушить весь процесс, то необходимо признать: цифровые машины обречены.

Таково было общее мнение, пока Шеннон не провозгласил первую ключевую идею информационной эры. Он продемонстрировал, что мы можем создавать произвольно точные сообщения, используя самые ненадежные каналы передачи информации. В своей знаковой статье «Математическая теория связи», опубликованной в журнале *Bell System Technical*

Journal в июле и октябре 1948 года, Шеннон предложил теорему кодирования для каналов с шумами, которая гласила: если у вас есть доступный канал с любым коэффициентом ошибок (за исключением 50 процентов на бит, поскольку это означает, что канал передает чистый шум), вы можете передать сообщение с любой желаемой степенью точности. Другими словами, частота появления ошибок может составлять один бит из n бит, где n может быть сколь угодно большим. Например, если у вас есть канал, который правильно передает биты информации только в 51 проценте случаев (т. е. передает правильный бит чуточку чаще, чем неправильный), вы тем не менее можете передавать сообщения таким образом, что неправильным окажется только один бит из миллиона, один бит из триллиона или один бит из триллиона триллионов.

Как такое возможно? Ответ: через избыточность.

Сейчас подобное решение может показаться элементарным, но в то время оно было далеко не очевидным. Рассмотрим простой пример. Если я передам каждый бит три раза и применю «принцип большинства», то я значительно повышу надежность результата. Если

этого будет недостаточно, я могу увеличить избыточность, пока не получу необходимую надежность. Повторение информации — самый простой способ достичь высокой точности в каналах низкой точности. Однако данный подход не самый эффективный. В своей статье Шеннон не только заложил основы теории информации, но и предложил оптимальные методы обнаружения и исправления ошибок. Эти методы позволяли добиться любой желаемой точности через любой неслучайный канал.

Читатели более старшего возраста наверняка помнят телефонные модемы, в которых постоянно что-то шипело и шелкало, так как они передавали информацию по аналоговым линиям с высоким уровнем помех. Однако те же модемы могли передавать цифровые данные с очень высокой точностью — благодаря теореме Шеннона для канала с шумами.

Аналогичная проблема и аналогичное решение существуют и для цифровой памяти. Вы когда-нибудь задумывались, почему CD, DVD и программные диски продолжают работать даже после того, как упали на пол или были поцарапаны? Этим мы тоже обязаны Шеннону. Процесс вычислений состоит из трех элементов: связи (которая, как я уже го-

ворил, имеет место как внутри, так и между машинами), памяти и логических вентилей (которые выполняют арифметические и логические функции). Точность логических вентилей можно сделать произвольно высокой с помощью особых кодов для обнаружения и исправления ошибок. Именно благодаря теореме Шеннона мы можем обрабатывать большие и сложные цифровые данные, используя для этого достаточно длинные алгоритмы.

Вторая важная идея, на которую опирается наш информационный век, — универсальность машинных вычислений. В 1936 году Алан Тьюринг описал «машину Тьюринга» — абстрактную вычислительную машину, которая состоит из бесконечно длинной ленты, разделенной на клетки с цифрами 1 или 0. Машина считывает одну клетку за другой и содержит набор правил в виде пронумерованных состояний, фактически представляющих собой хранимую в памяти программу. Каждое правило предписывает машине совершить одно действие, если в считываемой клетке стоит 0, и другое действие, если в считываемой клетке стоит 1. Возможные действия включают запись 0 или 1, перемещение ленты на одну клетку вправо или влево, оста-

новку ленты. Каждое состояние содержит номер следующего состояния, в которое должна перейти машина. Завершив алгоритм, машина останавливается; выход процесса остается на ленте. Хотя лента теоретически бесконечна, любая программа (которая не подразумевает бесконечный цикл) использует конечную часть ленты; следовательно, если мы ограничимся конечной памятью, машина по-прежнему сможет решать широкий круг задач.

В машине Тьюринга нет ничего сложного, верно? На самом деле именно этого и добивался ученый. Он хотел, чтобы его машина была максимально простой (но не проще, перефразируя Эйнштейна). Позже Тьюринг и его бывший учитель, Алонзо Черч сформулировали тезис Черча — Тьюринга, согласно которому задача, которая не может быть решена машиной Тьюринга, не может быть решена никакой *другой* машиной. Хотя собственно машина Тьюринга способна выполнять крайне ограниченное количество команд и одновременно обрабатывает всего один бит, она может вычислить все, что может вычислить любая вычислительная машина.

Строгие интерпретации тезиса Черча — Тьюринга предполагают принципиальную эк-

вивалентность того, что человек может думать или знать, и того, что может быть вычислено машиной. Основная идея заключается в том, что человеческий мозг подчиняется естественным законам; следовательно, его способность обрабатывать информацию не может превосходить таковую у машины (и соответственно у машины Тьюринга).

В своей статье Тьюринг заложил теоретические основы машинных вычислений. Хотя это целиком и полностью его заслуга, важно отметить, что большое влияние на него оказала лекция, прочитанная Джоном фон Нейманом в 1935 году в Кембридже (Англия). Лекция была посвящена идее программы, которую можно хранить в памяти — концепция, позднее воплощенная в машине Тьюринга. На фон Неймана в свою очередь произвела глубочайшее впечатление статья Тьюринга 1936 года, где были изложены принципы машинных вычислений и которую в конце 1930-х — начале 1940-х годов он включил в список обязательной литературы, составленный для своих коллег.

В той же работе Тьюринг сообщает о другом неожиданном открытии, а именно — о проблеме неразрешимых задач. Неразрешимые задачи — это хорошо описанные задачи с однознач-

ным ответом, который, однако, не может быть вычислен на машине Тьюринга (т. е. на *любой* машине). Это противоречит постулату XIX века, гласящему, что все задачи, которые могут быть описаны, в конечном счете будут решены. Тьюринг показал, что неразрешимых задач столько же, сколько и разрешимых. В своей «Теореме о неполноте» 1931 года Курт Гедель приходит к аналогичному выводу. Таким образом мы оказываемся в странной ситуации: с одной стороны, мы можем описать задачу и доказать, что однозначный ответ существует, а с другой — знаем, что ответ никогда не будет найден.

Гораздо больше можно сказать о философском значении работ Тьюринга, Черча и Геделя, однако в рамках данного предисловия достаточно отметить следующее. Тьюринг показал, что в основе всех машинных вычислений лежит очень простой механизм. Поскольку машина Тьюринга (и, следовательно, любая вычислительная машина) способна определять дальнейший образ действий, исходя из результатов предыдущих операций, она способна принимать решения и моделировать произвольно сложные иерархии данных.

К декабрю 1943 года Тьюринг спроектировал и построил «Колосс», который часто на-

зывают первым компьютером в истории электронных вычислительных машин. «Колосс» предназначался для выполнения одной-единственной задачи — декодирования радиосообщений, зашифрованных нацистской машиной Enigma, — и не мог быть перепрограммирован для выполнения другой задачи. Зато с дешифровкой он справлялся блестяще: считается, что именно благодаря ему союзники смогли взять верх над немецкими «люфтваффе» и выиграть решающую битву за Британию.

Все вышеизложенное послужило своеобразным фундаментом, на котором Джон фон Нейман создал архитектуру современного компьютера — машину фон Неймана, составляющую основу практически каждого автомата, изобретенного за последние шестьдесят шесть лет: от микроконтроллера стиральных машин до самых больших суперкомпьютеров. Это третья ключевая идея информационной эры. В статье «Первый проект отчета о EDVAC»*, датированной 30 июня 1945 года, фон Нейман изложил революционные прин-

* EDVAC (Electronic Discrete Variable Automatic Computer) — одна из первых электронных вычислительных машин на двоичной основе. — *Здесь и далее примеч. ред.*

ципы в области машинных вычислений. Модель фон Неймана состоит из центрального процессора, в котором выполняются арифметические и логические операции, блока памяти, в котором хранятся программа и данные, устройства массовой памяти, счетчика команд и каналов ввода-вывода. Данная концепция описана в первой части этой книги. Хотя статья фон Неймана, по сути, представляла собой внутренний проектный документ, она не только стала библией для конструкторов вычислительных машин 1940-х и 1950-х годов, но и значимо повлияла на архитектуру всех компьютеров, построенных с тех пор.

Машина Тьюринга не была рассчитана на практическое применение. Теоремы Тьюринга не имели никакого отношения к эффективности решения задач; они лишь позволяли определить круг тех задач, которые могли быть решены посредством машинных вычислений. Фон Нейман, наоборот, ставил своей целью формулирование практических принципов вычислительной машины. Так, в машине фон Неймана однобитовые вычисления Тьюринга заменены многобитовыми словами (как правило, количество битов кратно восьми). Если машина Тьюринга тратит огромное количе-

ство времени на перемещение ленты вперед и назад, чтобы сохранять и извлекать промежуточные результаты, то машина фон Неймана, напротив, снабжена памятью с произвольным доступом, поэтому любой элемент данных может быть извлечен немедленно.

Одним из важнейших принципов, предложенных фон Нейманом, является принцип хранимой программы, который он сформулировал десятью годами ранее: программа хранится в той же памяти, что и данные. Это позволяет перепрограммировать вычислительную машину для выполнения различного рода задач, а также использовать самомодифицирующийся код, допускающий применение некоторых форм рекурсии. До того времени практически все вычислительные машины, включая «Колосс» Тьюринга, могли решать только те задачи, для решения которых были предназначены изначально. Хранимая программа придала вычислительной машине поистине универсальный характер. Так, идея Тьюринга об универсальной машине получила реальное воплощение.

Другой ключевой принцип фон Неймана заключается в том, что каждая инструкция должна содержать код арифметической или логической операции, которую необходимо

выполнить, а также адрес операнда в памяти. Данная формула впервые была предложена фон Нейманом в рамках его публикации, посвященной совместно с Дж. Преспером Эккертом и Джоном Мокли проекту EDVAC. Сам EDVAC был запущен только в 1951 году; к тому времени уже существовали другие вычислительные машины с хранимыми программами, в частности Манчестерская малая экспериментальная машина, ENIAC*, EDVAC** и BINAC***. Все они были созданы под влияни-

* ENIAC (русс. ЭНИАК) — электронный числовой интегратор и вычислитель, сокр. от Electronic Numerical Integrator and Computer, первый электронный цифровой вычислитель общего назначения, который можно было перепрограммировать для решения широкого спектра задач.

** EDVAC (русс. ЭДВАК) — сокр. от Electronic Discrete Variable Automatic Computer, одна из первых электронных вычислительных машин. В отличие от своего предшественника ЭНИАКа, это был компьютер на двоичной, а не десятичной основе.

*** BINAC (русс. БИНАК) — сокр. от англ. Binary Automatic Computer, двоичный автоматический компьютер, электронный компьютер первого поколения, построенный в США компанией Eckert–Mauchly Computer Corporation и запущенный в апреле или августе 1949 года. Формально считается первым коммерческим электронным компьютером.

ем работ фон Неймана теми же конструкторами: Эккертом и Мокли. Фон Нейман принял непосредственное участие в проектировании ряда этих машин, включая более позднюю версию ENIAC, которая поддерживала хранимую программу.

Можно назвать несколько предшественников архитектуры фон Неймана, однако ни один из них не являлся истинной машиной фон Неймана, за одним неожиданным исключением. «Марк I» Говарда Эйкена, построенный в 1944 году, имел элемент программируемости, но не использовал хранимую программу. Эта вычислительная машина считывала инструкции с перфорированной бумажной ленты, а затем выполняла каждую команду отдельно. Поскольку ветвления в программе отсутствовали, ее нельзя рассматривать как пример архитектуры фон Неймана.

Еще до «Марка I», в 1941 году, Конрадом Цузе была создана вычислительная машина Z3. Она тоже считывала программу с ленты (точнее, с перфорированной киноплёнки) и не могла совершать условные переходы. Интересно, что Цузе получил поддержку от немецкого Научно-исследовательского ин-

ститута авиации, который использовал Z3 для своих расчетов, однако его просьба о государственном финансировании замены реле на электронные трубки была отклонена. Нацисты считали, что машинные вычисления «не помогут выиграть войну».

Единственный истинный предшественник концепции фон Неймана появился на целых 100 лет раньше. В аналитической машине Чарльза Бэббиджа, описанной им впервые в 1837 году, идея хранимой программы реализована с помощью перфорированных карт, заимствованных из ткацкого станка Жаккара. Память с произвольным доступом была рассчитана на одну тысячу слов по 50 десятичных цифр каждое (около 21 килобайта). Инструкции не только содержали код операции и номер операнда, как и все современные языки программирования, но и включали ветвление и циклы. Это значит, что описанное Бэббиджем представляло собой настоящую машину фон Неймана. Впрочем, он явно не мог вообразить себе всех технических и организационных сложностей, связанных с проектированием такого рода автоматов, а потому его аналитическая машина так и осталась недостроенной. Неясно, знали ли пионеры ин-

форматики XX века, в том числе фон Нейман, о наработках Бэббиджа.

Хотя машина Бэббиджа так и не была опробована на практике, она тем не менее привела к возникновению такой области, как разработка программного обеспечения. Ада Байрон, графиня Лавлейс и единственный законнорожденный ребенок поэта лорда Байрона, написала программы для аналитической машины Бэббиджа (которые, между прочим, ей приходилось отлаживать в уме), а также перевела статью итальянского математика Луиджи Менабреа, снабдив ее собственными пространственными комментариями. Она писала, что «аналитическая машина плетет алгебраические узоры, как жаккардовый ткацкий станок — цветы и листья». Вероятно, Ада Байрон первой предположила возможность создания искусственного интеллекта, но пришла к выводу, что аналитическая машина «не может породить нечто новое».

Концепция Бэббиджа поистине удивительна, учитывая эпоху, в которой он жил и трудился. Однако к середине XX века его работа затерялась в тумане десятилетий. Именно фон Нейман сформулировал ключевые принципы компьютера, каким мы знаем его сегодня.

ня, а его машина была и остается основной моделью машинных вычислений.

Впрочем, мы не должны забывать, что элементы машины фон Неймана постоянно обмениваются данными; если бы не теоремы Шеннона и методы, которые он разработал для передачи и хранения цифровой информации, ее бы не существовало.

Это подводит нас к четвертой важной идее — поискам способов наделить компьютеры интеллектом и опровергнуть заключение Ады Байрон о неспособности машины мыслить творчески. Алан Тьюринг первым сформулировал такую цель в своей статье 1950 года «Вычислительные машины и разум», которая включает ныне знаменитый тест Тьюринга. Последний сводится к следующему вопросу: достиг ли искусственный интеллект уровня человеческого интеллекта? В настоящей книге, после обсуждения предлагаемой им архитектуры, фон Нейман рассматривает человеческий мозг. В конце концов, этот орган — наилучший пример интеллектуальной системы, которым располагает человечество. Если мы сможем изучить его методы, мы сможем использовать биологические парадигмы для создания более умных машин. Эта книга

представляет собой самое первое серьезное исследование человеческого мозга с точки зрения математика и пионера информационной эры. До фон Неймана информатика и нейронаука были подобны двум островам без моста между ними.

По иронии судьбы, последней работой одного из самых блестящих математиков XX века и одного из пионеров компьютерной эры стало исследование человеческого интеллекта в рамках серии лекций для Йельского университета. К сожалению, из-за болезни фон Нейман так и не прочел эти лекции и даже не закончил рукопись. Тем не менее они остаются блестящим и пророческим предвестием самого, на мой взгляд, дерзкого и важного проекта человечества.

Фон Нейман начинает с описания сходств и различий между вычислительной машиной и человеческим мозгом. Учитывая, что писал он в 1955 и 1956 годах, рукопись удивительно точна, особенно в тех аспектах, которые предлагаются для сравнения. Фон Нейман отмечает, что выход нейронов носит цифровой характер: аксон либо порождает нервный импульс, либо нет. В то время это было далеко не очевидно: выходной сигнал вполне мог оказаться анало-

говым. Так, дендриты, например, работают по аналоговому принципу. Фон Нейман описывает вычисления, которые производит нейрон, как взвешенную сумму входных сигналов с неким порогом. Данная модель работы нервной клетки легла в основу такого направления, как коннекционизм — построение искусственных систем (как с точки зрения технического, так и программного обеспечения) по принципу организации и функционирования живого нейрона. Первая такая коннекционистская система была создана в 1957 году Фрэнком Розенблаттом и представляла собой программное обеспечение для IBM-704 Авиационной лаборатории Корнелла.

Сегодня мы гораздо больше знаем о том, как именно нейроны сочетают входные сигналы, хотя основополагающая идея об аналоговом принципе работы дендритов сохраняется. С нашей стороны было бы неразумно ожидать, что фон Нейману, писавшему в 1957 году, будут известны точные механизмы обработки информации в нервной системе, однако ключевые моменты, на которых он строит свои выводы, остаются в силе.

На основании представления об универсальности вычислений фон Нейман прихо-

дит к заключению, что, несмотря на разницу в архитектуре и строительных блоках мозга и вычислительной машины, его машина тем не менее может имитировать работу мозга. Обратное, впрочем, неверно, поскольку мозг не является машиной фон Неймана и не имеет хранимой программы как таковой. Его алгоритм заложен в самой его структуре.

Фон Нейман справедливо утверждает, что нейроны способны выявлять закономерности во входных данных, которые, как мы теперь знаем, закодированы в соответствующих концентрациях нейротрансмиттеров. Во времена фон Неймана другие способы — создание и разрушение связей между нервными клетками — еще не были известны.

Фон Нейман отмечает, что скорость обработки информации нейроном чрезвычайно низкая, порядка 100 вычислений в секунду, однако мозг компенсирует это массовым параллелизмом. Все его 10^{10} нейронов обрабатывают информацию одновременно (это число является достаточно точным; современные оценки колеблются между 10^{10} и 10^{11}). На самом деле правильнее говорить даже не об отдельных нервных клетках, а об их связях, ко-

личество которых в среднем составляет около 10^3 на 1 нейрон.

Описания механизмов функционирования нейронов, предложенные фон Нейманом, на удивление точны, учитывая «первобытное» состояние нейронауки в то время. Единственное, с чем я не могу согласиться, — это оценка емкости памяти мозга. Фон Нейман полагает, что мозг хранит поступающую в него информацию на протяжении всей жизни человека. Шестьдесят лет — это примерно 2×10^9 секунд. Если допустить, что каждый нейрон из 10^{10} принимает 14 единиц информации в секунду (на самом деле эта цифра в три раза меньше), то емкость памяти мозга будет составлять около 10^{20} бит. Однако реальность такова, что мы помним лишь очень небольшую часть наших мыслей и опыта. Кроме того, эти воспоминания сохраняются не в виде конфигураций битов низшего уровня (такого, как видеоизображение), а скорее как последовательности более высокоуровневых шаблонов. Кора головного мозга представляет собой иерархию распознавателей шаблонов. Некоторые из них распознают определенные топологические формы, например перекладину в заглавной букве «А». На следующем уровне распозна-

ются конкретные буквы, например буква «А». На еще более высоком уровне распознаются слова, например «арбуз». В другой части коры может происходить распознавание предметов (арбуз); в третьей части — услышанного слова («арбуз»). На гораздо более высоком концептуальном уровне распознаватель может заключить: «Это было забавно». Наши воспоминания о событиях и мыслях кодируются в рамках этих высокоуровневых распознаваний. Допустим, мы хотим вспомнить некий опыт. Происходящее в нашей голове не будет иметь ни малейшего сходства с воспроизведением видеосигналов. Скорее, мы вспоминаем последовательность высокоуровневых шаблонов. Фактически мы вынуждены заново, шаг за шагом, воссоздавать прошлый опыт, ибо наша память не сохраняет подробности в явной форме.

Вы можете с легкостью убедиться в этом сами. Например, попытайтесь припомнить свою последнюю прогулку. Что вы помните? Как выглядел пятый человек, которого вы встретили? Вы видели детскую коляску? А почтовый ящик? Что вы увидели, когда свернули за угол первого дома? Если вы проходили мимо нескольких магазинов, что было во второй

витрине? Возможно, вам удастся восстановить ответы на эти вопросы, руководствуясь теми единичными подсказками, которые сохранила ваша память, однако большинство из нас не могут в подробностях вспомнить прошлые события. Машины, напротив, вспоминают легко. В этом и заключается одно из преимуществ искусственного интеллекта.

В настоящей книге очень мало сведений, которые в значительной степени расходятся с тем, что известно сейчас. На сегодняшний день мы не в состоянии дать исчерпывающее полное описание мозга, а потому не можем ждать этого и от работы 1956 года по обратной инженерии мозга. С учетом данной оговорки формулировки фон Неймана поразительно точны, а сведения, на которых он основывает свои умозаключения, до сих пор сохраняют актуальность. Описывая механизмы работы мозга, фон Нейман показывает, каким образом современный компьютер, несмотря на очевидные различия, может выполнять те же самые операции. Так, аналоговые механизмы мозга можно реализовать через цифровые механизмы, поскольку цифровые вычисления способны эмулировать аналоговые величины с любой желаемой степенью точности (а точ-