Journal of Applied Linguistics and Lexicography

РУБРИКА «ПРОСТО О СЛОЖНОМ»

Нейронные сети и компьютерная обработка языка

О. В. Митренина✉[1]

[1] Санкт-Петербургский государственный университет, 199034, Россия, Санкт-Петербург, Университетская наб., д. 7/9

Artificial Neural Networks and Natural Language Processing

O. V. Mitrenina✉[1]

[1] Saint Petersburg State University, 7–9 Universitetskaya Emb., Saint Petersburg 199034, Russia

Почему дети учат язык лучше, чем взрослые

Однажды несколько монахов, живших на краю Скитской пустыни, обнаружили у себя корзину. В корзине плакал чернокожий младенец, который, несомненно, был подкинут эфиопским караваном, проходившим тут накануне. Растроганные таким непредвиденным подарком небес, братья стали кормить младенца и усердно заботиться о нем.

Шло время. И вот как-то один из монахов, весьма обеспокоенный, сказал:

— Нужно, чтобы кто-нибудь из нас выучил эфиопский язык.

— Но почему? — воскликнули изумленные братья.

— Потому что скоро младенцу исполнится год, и он начнет говорить, а никто из нас не знает его языка.

В этой истории из книги «Отцы-пустынники смеются» монах предполагал, что ребенок хранит в голове язык своих родителей. На самом деле дети усваивают тот язык, который слышат вокруг себя. Взрослый человек теряет детскую способность усваивать и вынужден зубрить. Он может бесконечно повторять грамматику и лексику, но так и не научится говорить на новом языке как на родном.

Почему же маленькие дети учат язык таким способом, который уже недоступен взрослым?

Может быть, дело в том, что ребенок усваивает свой первый язык на «чистую голову», а голова взрослого уже заполнена родным языком, и новый язык помещается туда очень плохо? Нет, это не главная причина. Ребенок может расти в двуязычной среде, и тогда он станет билингвом — легко усвоит два языка как родные, а может усвоить и три. Без всякой

зубрежки. Достаточно, чтобы с ним и вокруг него много говорили на этих языках — они не перепутаются и спокойно «поместятся в голове».

Но с годами способность усваивать язык исчезает даже тогда, когда ни один язык не выучен. Известны истории, когда дети росли вне языковой среды. Девочку под кодовым именем Изабелла обнаружили в шестилетнем возрасте. До шести лет она не слышала человеческой речи. Через год она уже полностью овладела языком и пошла в обычную школу. Другую девочку, Джини, нашли, когда ей было тринадцать. Несмотря на титанические усилия специалистов, она так и не научилась строить предложения более чем из двух или трех слов. Языковые навыки у нее так и не выработались.

Разгадка связана с разницей в устройстве мозга взрослого человека и ребенка. Мозг состоит из нейронов. Это нервные клетки, которые способны передавать информацию друг другу. Для этого между нейронами должны установиться связи. У новорожденного младенца много нейронов, но мало связей между ними. Эти связи с огромной скоростью начинают возникать по мере того, как ребенок знакомится с внешним миром и осваивает новые навыки. В мозгу у него с огромной скоростью выстраиваются нервные цепочки — нейронные пути, нейронные сети. Затем эти нейроны и созданные ими сети начинают покрываться особым веществом, которое называется миелин. Так закрепляется нейронная сеть. Она становится более прочной и удобной, но менее гибкой. Знания фиксируются. Мы начинаем хуже учиться, зато более эффективно использовать накопленный жизненный опыт.

К семи годам выработка миелина уменьшается, сети закрепляются хуже, но к этому времени наш организм успевает выработать основные жизненные навыки. Не все, конечно. Но «глубинные» — почти все. Далее можно переучиваться и доучиваться, но на это будут требоваться дополнительные усилия.

Однако не все потеряно: новые связи продолжают образовываться. Особенно если мы их приучаем образовываться — заставляем себя пробовать что-то новое и учиться. Даже поход домой менее привычной дорогой оживляет процесс создания новых нейронных цепочек. Опасно ограничивать себя только привычными ситуациями и избегать тех, для которых требуются внутренние усилия.

Первые попытки научить компьютер человеческому языку

Если в научно-фантастическом фильме появляется робот, он обязательно говорит на человеческом языке. И это не удивительно — если в фильме он думает и ходит, почему бы ему еще и не разговаривать. Ведь это наиболее простой и естественный механизм коммуникации с человеком.

Но в жизни всё оказалось сложнее. Роботы пока не научились свободно общаться с людьми. Глобальная задача «научить компьютер человеческому языку» заменилась на более скромные пожелания из серии «пусть поможет хотя бы в этой области». Вот только некоторые из них:

- переводить тексты с одного языка на другой, чтобы каждый раз не нанимать переводчика;

- просмотреть тысячу-другую блогов и составить отчет о том, что говорят о вашем новом продукте, чтобы не нанимать трех застревающих в сети менеджеров;
- то же самое, но изучая не содержание сообщений, а настроение пишущих — в конце концов, содержание забывается, а настроения сохраняются надолго;
- отвечать на вопросы клиентов — вопросы у них повторяются, и один чат-бот может заменить два десятка менеджеров по работе с клиентами.

Вначале машины пытались научить теми же методами, какими обучают взрослых людей. В их память загружали слова и правила соединения этих слов. Если глагол *ПРОСНУТЬСЯ* соединить с существительным *КОШКА* и поставить в прошедшее время, то получится *КОШКА ПРОСНУ-ЛАСЬ*. А если кошек несколько? Тут форма слова зависит от числа. *ТРИ КОШК-И, ПЯТЬ КОШ-ЕК, ДВАДЦАТЬ ОДНА КОШ-КА*. Побольше правил, и компьютер научится выдавать человеческие предложения. Будет говорить, как взрослый человек, который учил новый язык по книжкам и не жил в той среде, где на этом языке общаются.

Так работает **подход, основанный на правилах**. Компьютеру дают набор слов и правила их обработки. Это его «знания». Накопив их, он получает предложения, которые должен обрабатывать. Он находит в своем словаре нужные слова или фрагменты слов и применяет к ним правила из своего списка правил.

В итоге компьютер осваивает язык для какой-то конкретной задачи, но совершает много ошибок. Прикладной лингвист, словно опытный учитель, анализирует ошибки своего ученика, дает ему дополнительные правила и новые слова.

При таком способе обучения не используется главное преимущество компьютера — умение хранить и быстро обрабатывать огромные массивы информации (чисел). Человеку трудно сложить в уме восемь десятизначных чисел, а мой простенький ноутбук за одну секунду успеет сделать это сто миллионов раз.

Кто учит лазать по деревьям птицу, которая умеет летать? Так и компьютеру в конце концов предложили другое обучение, не человеческое, а машинное. Так появился **статистический подход**. Это были еще не нейросети, но это был уже «компьютерный» способ анализа языка, а не «человеческий».

Язык не так случаен, как кажется

Но если не использовать правила грамматики, то что может узнать компьютер о языке?

Вспомним Шерлока Холмса. В рассказе «Пляшущие человечки» он разгадал надписи, которые все принимали за детские рисунки. Вот первый текст, который попал ему в руки:

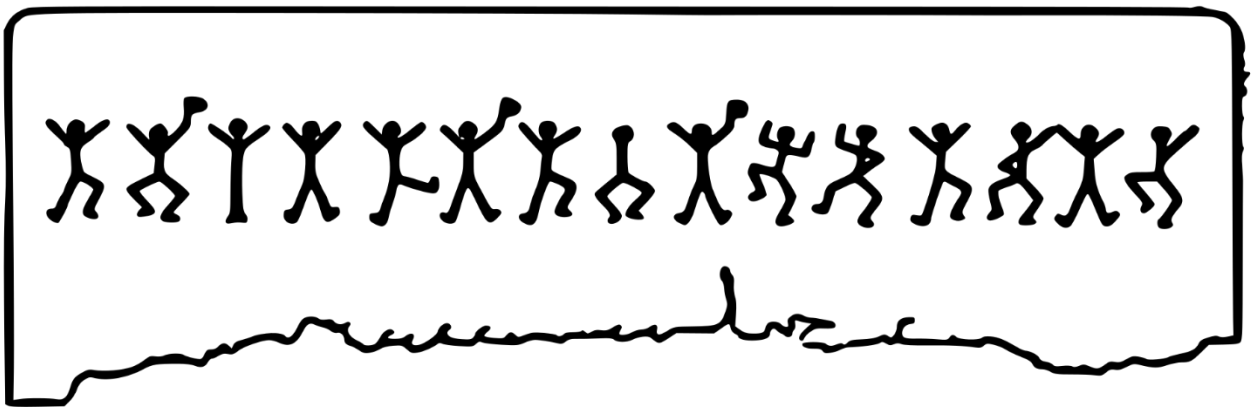

Рис. 1. Первое зашифрованное письмо, оказавшееся у Шерлока Холмса.

Холмс догадался, что флажки разбивают цепочку на слова. Он вычислил, какой человечек встречается чаще всего, и предположил, что это буква E. Она в английском языке самая частая:

Рис. 2. Самый частый человечек из первого письма.

После этого Холмс попытался понять, в каком окружении встречается E, и это подход компьютерного лингвиста. Но имевшийся у него набор текстов был слишком мал — всего одна надпись. Лишь получив еще две надписи, Холмс нашел слово из пяти букв, которое начиналось и заканчивалось буквой E. Он вспомнил имя хозяйки — *ELSIE* — и вычислил еще три буквы, а потом разгадал и всё остальное.

Тут важно, что Холмс не использовал ни правила, ни грамматику языка, а оперировал только статистическими данными о языке.

Компьютер, как и Холмс, может находить закономерности в текстах. Для этого он должен усвоить некоторые знания о языке. Ведь Холмс заранее знал, что самая частая буква — E, и что слово *ELSIE* вполне может встретиться в записке.

Что и как может узнать компьютер?

Откуда компьютер может узнать о языке, если не давать ему правил из школьной грамматики? Точно так же, как и ребенок, компьютер может узнать о свойствах языка прямо из текстов. Для этого ему требуется много текстов, признаки для анализа и алгоритм обучения, по которому он может что-то узнать. Рассмотрим каждый из этих трех пунктов внимательнее.

(1) Много текстов, на которых будет учиться компьютер

Набор таких текстов называется **обучающим корпусом**. Корпус может быть размеченным или нет. В размеченном корпусе заранее указаны «готовые ответы», а иногда и признаки текстов. Например, обучающим корпусом может стать большая коллекция электронных писем, каждое из которых имеет пометку «не спам» или «спам». Эти пометки и являются «готовыми ответами», изучив их, компьютер научится вылавливать новые письма со спамом.

Обучающим корпусом с «готовыми ответами» может стать большой набор диалогов. Его можно рассматривать как пары «реплика — готовый ответ на реплику». Обработав их, компьютер научится поддерживать беседу. Даже и человек, оказавшись в новом для него обществе — среди разбойников, например, или среди врачей — может послушать их язык, а затем научиться «говорить как они».

Если мы хотим заставить компьютер переводить с одного языка на другой, то надо дать ему большой обучающий корпус переведенных текстов: много предложений на исходном языке и их переводы. И никакой грамматики, заметьте! В качестве «готовых ответов» здесь выступают готовые переводы. Иногда в такой корпус добавляли признаки: устанавливали соответствия между словами или словосочетаниями. И напрасно. Как оказалось, без них машина переводит лучше.

Иногда для обучения используется неразмеченный корпус — просто тексты без всяких отметок. Тогда машина должна сама находить в них признаки и закономерности. Именно с нейросетями она научилась делать это весьма хорошо.

(2) Признаки для анализа

Но если компьютер должен обработать множество текстов, то что именно он должен анализировать, какие параметры?

Холмс, расшифровывая пляшущих человечков, использовал признак частоты букв, а также некоторый хранившийся у него в голове словарь. Компьютеру тоже нужны **признаки** — особенности, характерные черты, — с помощью которых он будет анализировать тексты. Например, для оценки тональности текста пригодится словарь: если в сообщении есть слово *ГАДОСТЬ,* то, вероятно, это сообщение с сильной отрицательной тональностью. А если слово *НЕПЛОХОЙ*, то тональность, скорее всего, умеренно-положительная. Но признаки могут быть и менее очевидными. Например, при распознавании лиц компьютер будет анализировать признаки изгиба линий — такие, которые описать словами было бы невозможно. В текстах, как и в лицах, есть свои неявные признаки, которые компьютер сможет выявить.

(3) Алгоритм обучения

Как компьютер должен обрабатывать язык на основе обучающего корпуса и признаков? Так же, как и мозг ребенка: он должен находить не «правильный ответ» на основании четких правил, а «наиболее вероятный» ответ на основании уже полученных примеров правильных

конструкций. В большинстве случаев это бывает достаточно. Компьютерные алгоритмы постоянно совершенствуются, и всё более повышается вероятность правильного ответа.

Рассмотрим на примере, как компьютер использует вероятностные алгоритмы, чтобы научиться обращаться с языком.

В известном анекдоте человек после вечеринки садится в такси и на вопрос таксиста «Куда вам?» отвечает:

— К удавам не хочу.

— КУДА ВАМ НАДО?! — четко выговаривая слова, спрашивает таксист.

— Ну, надо так надо… Поехали к удавам, — обреченно соглашается пассажир.

Наш жизненный опыт подсказывает, что фраза «Куда вам надо» гораздо более вероятна, чем «К удавам надо». Особенно в контексте диалога с таксистом. Компьютер тоже может выбирать из нескольких возможных предложений наиболее вероятное. Это полезно, например, при составлении субтитров к видеозаписи или при автоматическом переводе. Можно не объяснять машине, что в предложении *КОШКА ПРОСНУЛСЯ* сказуемое неправильно согласуется с подлежащим по роду. Проще использовать вероятность. Сочетание *КОШКА ПРОСНУЛАСЬ* более вероятно, чем *КОШКА ПРОСНУЛСЯ*.

Как оценить вероятность предложения на основе корпуса? Самый простой способ — просто посмотреть, какие предложения встречались чаще. Но это плохой способ. Даже если корпус очень большой, он не охватывает всех возможных предложений.

Другой способ — использовать вероятности входящих в предложение слов. Какие-то слова встречаются чаще, какие-то — реже. С помощью корпуса можно легко посчитать вероятности слов. Например, если в корпусе 5 000 000 вхождений слов, а слово *СОБАКА* среди них встретилось 50 раз, то вероятность слова СОБАКА — одна стотысячная. Запишем это подобием формулы, где *p* означает вероятность:

$$p\,(\text{собака}) = \frac{\text{число вхождений слова СОБАКА}}{\text{число всех вхождений слов}} = \frac{50}{5\,000\,000} = \frac{1}{100\,000}$$

На основе корпуса можно заранее посчитать и запомнить вероятности всех слов. Тогда вероятность любого предложения (даже не встречавшегося в корпусе) можно посчитать, просто перемножив вероятности входящих в него слов, как это принято в теории вероятностей:

$$p\,(\text{птица сидит на крыше}) = p\,(\text{птица}) \cdot p\,(\text{сидит}) \cdot p\,(\text{на}) \cdot p\,(\text{крыше})$$

Этот способ тоже не очень хорош. Если слова переставить, вероятность не изменится. Но интуиция нам подсказывает, что вероятность предложения *НА ПТИЦА КРЫШЕ СИДИТ* должна быть ниже, чем *ПТИЦА СИДИТ НА КРЫШЕ*.

Предыдущие слова подсказывают нам, какими могут быть следующие слова, и это можно использовать при вычислении вероятностей. При этом, как показал 100 лет назад математик Андрей Марков, можно учитывать не весь предшествующий текст, а только несколько предыдущих слов. Если взять слишком большой предшествующий контекст, качество вычислений улучшится не слишком сильно, а сложность обработки возрастет. Поэтому в популярных лингвистических моделях стали брать контекст из двух предыдущих слов. Так появилась триграммная модель языка. От слова *триграмма* — три идущих подряд слова.

Триграммная модель языка

Триграммная модель языка оценивает вероятности предложений. Для ее создания машине нужен большой корпус. Первым делом она соберет список входящих в него слов. Точнее, словоформ. Ведь *СОБАКА и СОБАКОЙ* — это разные последовательности символов, поэтому машина запомнит их как две разные единицы, не вникая, что за зверь за ними стоит, какая это часть речи и в какой грамматической форме. Важно только, что обе словоформы встретились в обучающем корпусе.

После этого для каждой тройки словоформ компьютер посчитает условную вероятность. Какова вероятность того, что после слов *МНЕ ПОДАРИЛИ* встретится слово *СОБАКУ*? Какова вероятность того, что после слов *МНЕ ПОДАРИЛИ* встретится слово *НОСКИ*? И все остальные варианты сочетаний. Условная вероятность записывается так:

p (собаку | мне подарили)
p (носки | мне подарили)
p (носков | мне подарили)
p (носками | мне подарили)

Справа от вертикальной черты указывается контекст *МНЕ ПОДАРИЛИ*. В предложении он предшествует слову *СОБАКА*, которое записывается слева от черты. Полная запись означает вероятность появления слова *СОБАКА* при условии, что у нас уже есть слова *МНЕ ПОДАРИЛИ.*

Если рассматривать все возможные тройки, то большинство сочетаний будут казаться бессмысленными. Какова вероятность того, что после слов *СТАКАНЕ О* встретится слово *ГОРЕ*?

p (горе | стакане о)

Однако, это фрагмент из стихотворения Введенского:

и тишина была в стакане
о горе птичка говорит одна
не вижу солнечного я пятна

а мир без солнечных высоких пятен
и скуп и пуст и непонятен

На всякий случай надо посчитать вероятности для всех теоретически возможных троек слов. Отдельно считаются вероятности того, что слово встретится в самом начале предложения и перед ним ничего не будет, а также вероятность концов предложений во всех контекстах.

Все эти вероятности вычисляются на исходном корпусе и запоминаются. Они называются параметрами модели. После этого вероятность любой последовательности словоформ считается перемножением параметров:

$$p(\text{птица сидит на крыше}) = p(\text{птица} \mid **) \cdot p(\text{сидит} \mid *\text{птица}) \cdot p(\text{на} \mid \text{птица сидит}) \cdot p(\text{крыше} \mid \text{сидит на}) \cdot p(\text{КОНЕЦ} \mid \text{на крыше})$$

Последний параметр соответствует вероятности конца предложения после слов *НА КРЫШЕ*. Звездочки в двух первых параметрах означают начало предложения.

Осталось только понять, как на основе корпуса посчитать параметры модели — вероятности того, что слово встретится в том или в ином контексте. Машина может сделать это очень легко. Для *p (собаку | мне подарили)* она посчитает, сколько раз в корпусе встретился контекст *МНЕ ПОДАРИЛИ*. Например, 100 раз. А затем — сколько раз после этого контекста встретилось нужное слово *СОБАКУ*. Всего один раз. Значит, параметр — 1/100 = 0,01. Конечно, чаще встретятся *НОСКИ* — раз 70. Параметр для них будет 70/100 = 0,7.

$$p(\textit{собаку} \mid \textit{мне подарили}) = \frac{\text{число сочетаний МНЕ ПОДАРИЛИ СОБАКУ}}{\text{число сочетаний МНЕ ПОДАРИЛИ}} = \frac{1}{100} = 0{,}01$$

$$p(\textit{носки} \mid \textit{мне подарили}) = \frac{\text{число сочетаний МНЕ ПОДАРИЛИ НОСКИ}}{\text{число сочетаний МНЕ ПОДАРИЛИ}} = \frac{70}{100} = 0{,}7$$

Модель можно немного скорректировать («сгладить») и добавить математическую обработку случаев, которые не встретились в обучающем корпусе.

Машинное обучение и нейросети

В триграммной модели языка машина использует всего один признак: частоту появления слова после двух других слов.

Можно задавать машине другие признаки и решать другие задачи, используя математические методы. Например, можно взять корпус электронных писем, часть из них пометить как спам, а остальные — как не спам. Компьютер вычислит, с какой частотой и в

каких контекстах встречаются слова типа *скидка* и *распродажа*, а затем научится находить спам среди новых сообщений. Похожим образом работает определение языка текста: компьютер сможет определять, что один абзац написан на английском, а другой — на испанском.

Классические алгоритмы машинного обучения хорошо справляются с подобными задачами классификации. Важно, чтобы у текстов были простые признаки. Например, характерные слова или знаки, типичные последовательности слов или повторяющаяся частотность. Нужно дать машине эти признаки, и она начнет анализировать данные.

Можно не размечать тексты заранее, но дать машине набор признаков. На их основании машина научится выделять похожие между собой элементы и объединять их в группы. Это называется кластеризация, и она используется, например, в контекстной рекламе или для рекомендации статей.

Но человек не всегда в состоянии найти подходящие признаки. Он хорошо замечает их только в самых очевидных случаях, но в более общих задачах, связанных с большим количеством признаков, начинает ошибаться. Как, например, отличить по стилю одного автора от другого? Какие признаки подобрать для машинного перевода?

Наконец, как устроить сам алгоритм обработки больших массивов информации?

Для работы со сложными данными всё чаще стали использовать нейронные сети. Именно они дают лучшие результаты в тех случаях, когда признаков так много, что непонятно, какие из них влияют на результат.

Нейронные сети возникли как попытка смоделировать на компьютере работу человеческого мозга. Мозг состоит из нейронов, каждый из которых принимает сигналы от других нейронов. Если уровень сигнала достаточно высок, нейрон передает его дальше.

Первый компьютер, моделирующий нейронную сеть, был создан в 1958 году американским ученым Фрэнком Розенблаттом. В его основе лежала простая модель, для которой Розенблатт придумал название **перцептрон.**

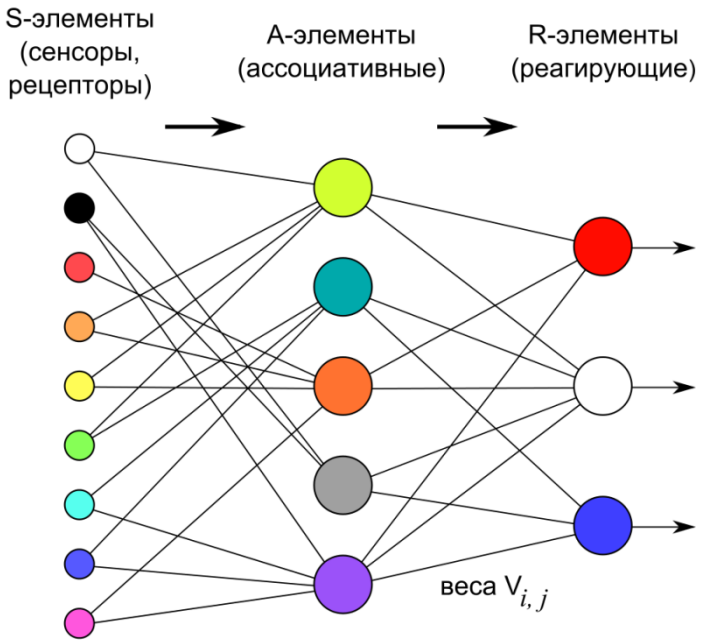

Рис. 3. Схема перцептрона Розенблатта.

Подробнее о том, как устроен перцептрон, будет рассказано в одном из следующих номеров нашего журнала, пока приведем лишь очень краткое описание. Нейроны в перцептроне расположены слоями. Каждый нейрон передает сигнал только нейронам следующего слоя. У каждой передающей связи есть свой вес, сигнал уменьшается или увеличивается пропорционально этому весу. Если вес 2, сигнал становится в два раза сильнее. А если он 0,1, то в десять раз слабее. Даже в жизни мы придаем большое значение сообщениям от некоторых людей, а слова других почти полностью пропускаем мимо ушей. Такая разная оценка сообщений приходит с жизненным опытом. Нейрону тоже надо подобрать правильный вес для каждой связи, и тогда на выходе получится нужный ответ.

Алгоритмы искусственных нейронных сетей совершенствовались, но принципиальный рывок произошел в 1974 году, когда А. И. Галушкин в МИФИ и Пол Вербос в Гарварде одновременно и независимо друг от друга предложили **метод обратного распространения ошибки**. Этот метод позволял машине настраивать веса, чтобы результат был всё более точным. Вначале веса выбираются случайно. Затем по сети передается сигнал и оценивается ошибка на выходе. После этого, двигаясь по сети в обратном направлении, от последнего слоя к первому, нужно «подкручивать» веса так, чтобы ошибка становилась чуть меньше. Затем пропускается новый обучающий сигнал, сравниваем результат с ответом и опять поправляем веса, двигаясь от последнего слоя к первому. Для этого алгоритма был разработан хороший математический аппарат. Так обучается сеть.

Количество слоев и типы связей — это архитектура нейросети. Архитектуру можно поменять. Например, разрешить нейрону передавать информацию самому себе, тогда сеть станет **рекуррентной**. Можно пропустить сеть через «бутылочное горлышко» — добавить слой с малым числом нейронов, чтобы собрать только самые важные признаки. Такой слой называется **свёрточным**. Можно связать все нейроны со всеми, без всякого расслоения.

Получится **цепь Маркова, машина Больцмана** или **сеть Хопфилда** — в зависимости от того, как нейрон обрабатывает входящие значения.

Разработчики и сами не всегда понимают, почему в каких-то задачах одна архитектура эффективнее другой. Они просто экспериментируют, а машина начинает гладко переводить, определять авторов текста или поддерживать беседу.

«Человек, не мешай!»

Итак, эволюция машинной обработки языка идет по пути «человек, не мешай!». Сначала компьютер избавляется от человеческих правил и переходит на машинное обучение, которое опирается на признаки, предложенные человеком. Затем он начинает находить нужные признаки самостоятельно, используя нейросети. Человек придумывает только архитектуру этих сетей. На следующем этапе, вероятно, оптимальную архитектуру тоже будет выбирать машина, но этому ее должен научить человек.

Далее возникает вопрос: сможет ли машина выйти на следующий уровень и без человека придумать архитектуру той сети, которая будет выбирать архитектуру другим сетям? А пойти еще дальше? Научится ли она обучаться всему вообще без человека? Ведь и у нас в голове как-то сам собой подбирается путь для установления нейронных связей. Станет ли такая система эффективнее, чем человеческий мозг?

Сейчас это часть большой философской проблемы, и современные философы разделяются тут на два примерно равных лагеря. Одни считают, что именно так и произойдет, потому что всё должно моделироваться. Другие считают, что в человеческом восприятии всегда будет оставаться что-то невербализуемое, так называемые *qualia*, которые недоступны машине в принципе. Согласно представлениям этих философов, машина сможет смоделировать запах ландышей с химической стороны дела, но не сможет с главной — со стороны нашего ощущения от того, как это бывает, когда ты ощутил запах ландышей.

Но нейронные сети уже водят автомобиль лучше, чем человек. Они умеют генерировать на экране телевизора изображение диктора, который произносит заготовленный текст так искусно, что его не отличить от диктора-человека. Более того, если взять видеозапись вашего выступления, нейросеть сможет объединить ее с самым хулиганским в мире текстом и породит новую запись, на которой вы будете произносить этот текст своим голосом и в своей манере.

Все эти технологии появились буквально в самое последнее время. Эта научная область развивается так быстро, что учебники за ней не успевают. Но в Сети появляются курсы по созданию нейронных сетей. Лучшее, на мой взгляд, место для обучения нейросетям дистанционно — это Университет искусственного интеллекта https://neural-university.ru/.

И бесспорно то, что будущее технологий в ближайшие десятилетия — за нейронными сетями. Открытие этих технологий изменит мир так же сильно, как изменило его появление компьютеров.