Урок 9. Можно ли научить компьютер мыслить как человек?

7 класс · Урок 9 из 34 · 45 минут

Можно ли научить компьютер мыслить как человек? Сравним программу, которая отвечает по правилам, с моделью, которая учится на примерах. Затем разберём, что по удачному ответу можно узнать о возможностях каждой.

Содержание урока

- 01. Цель урока
- 02. Ход урока
- 03. Этап 1. Вход в тему: похоже на разговор · 6 минут
- 04. Этап 2. Правило может создавать впечатление · 10 минут
- 05. Этап 3. Обучение вместо готовой подстановки · 16 минут
- 06. Этап 4. Что осталось непроверенным · 7 минут
- 07. Этап 5. Итоги урока · 6 минут

01. Цель урока

Понять, как правила и обучение помогают компьютеру решать задачи и почему выбранные признаки ограничивают результат.

К концу урока ученики смогут:

- различать впечатление разговора и способ получения ответа;
- объяснять нейросеть как систему соединённых вычислительных элементов;
- описывать персептрон через признаки, веса и порог;
- объяснять, что меняется при обучении по ошибкам;
- создавать в группе наглядную модель работы с признаками и обсуждать, какие сведения она теряет.

02. Ход урока

Этап	Мин	Учитель	Ученики
1. Вход в тему: похоже на разговор	6	Уточняет вопрос о мышлении.	Формулируют свой проверочный вопрос.
2. Правило может создавать впечатление	10	Объясняет шаблоны на примере ELIZA.	Создают диалог с характером.
3. Обучение вместо готовой подстановки	16	Объясняет сеть и показывает изменение весов.	Рисуют значки и макет сети, выбирают признаки.
4. Что осталось непроверенным	7	Помогает обсуждать изменение условия.	Придумывают разговор с новым условием.
5. Итоги урока	6	Проверяет понимание модели.	Объясняют обучение и потерю сведений.

03. Этап 1. Вход в тему: похоже на разговор 6 минут

Говорим ученикам

Сегодня обсудим, можно ли научить компьютер мыслить как человек. Сначала уточним, что именно хотим проверить. Связно отвечать, решать новую задачу и объяснять причину ответа — не одно и то же. По одной удачной фразе нельзя узнать весь способ работы системы.

Выберите один признак хорошего объяснения: например, способность применить мысль к новому условию. Запишите, какой вопрос вы задали бы, чтобы проверить этот признак. В конце урока вернёмся к вопросу и уточним его.

04. Этап 2. Правило может создавать впечатление 10 минут

Говорим ученикам

Чтобы программа отвечала на сообщения, можно задать правила преобразования текста. Если программа находит определённое слово, она может выбрать шаблон и подставить часть исходной фразы. Полученная реплика может подходить к сообщению, если заданный шаблон учитывает его содержание.

Другой способ — обучить модель на примерах. Сравним, как в одном случае человек задаёт готовые правила ответа, а в другом программа меняет настройки модели при обучении.

Реальный случай. Ранний собеседник ELIZA

В январе 1966 года Джозеф Вейценбаум из Массачусетского технологического института описал ELIZA — программу для текстового разговора. Она анализировала ключевые слова и строила ответы по правилам преобразования фраз. Уже тогда человек мог вводить обычные

письменные фразы и получать ответ компьютера. ELIZA показала один способ построения такого диалога: человек заранее задаёт правила, по которым программа составляет реплики. Дальше сравним этот способ с обучением на примерах.

Задание 1. Диалог с характером

Придумайте для текстового персонажа характер и тему разговора. Напишите две реплики человека и ответы персонажа: один ответ строится простой подстановкой слов, другой учитывает смысл всей реплики. Покажите диалог другой паре и попросите объяснить, где одного шаблона оказалось недостаточно. Затем улучшите неудачный ответ.

▼ **Для учителя: ответы и обсуждение**

Важны две разные работы: подстановка и содержательный ответ. Ученик должен показать, что теряется при механическом преобразовании, например отрицание, просьба или причина. Характер персонажа может быть любым. Это авторская модель разговора; не утверждать, что именно так устроена настоящая современная модель.

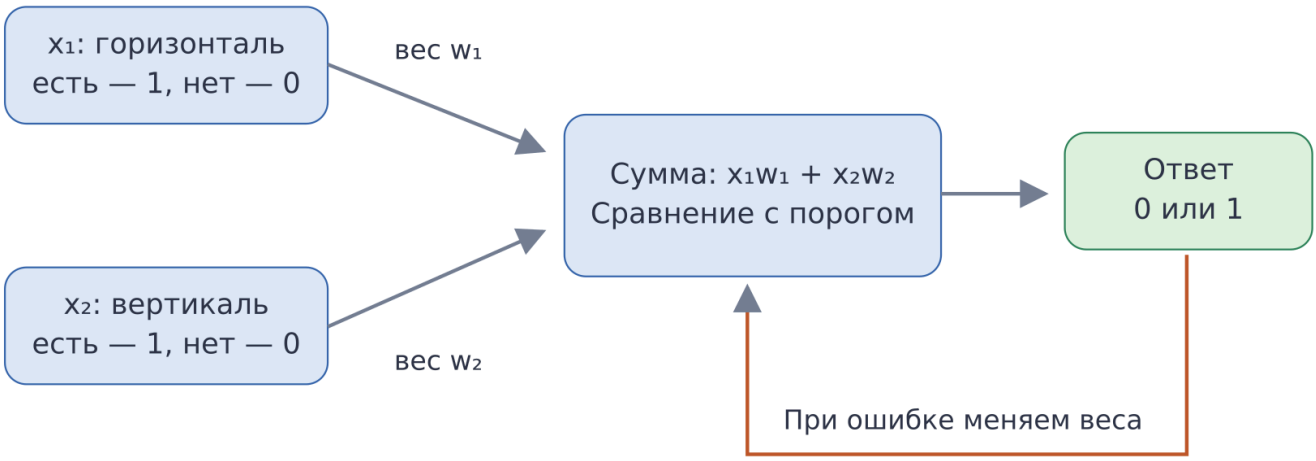
05. Этап 3. Обучение вместо готовой подстановки 16 минут

Говорим ученикам

Другой способ — обучать модель по примерам. В искусственной нейросети вычислительные элементы связаны между собой. Каждый получает числа, преобразует их и передаёт результат дальше. Обычно различают вход, промежуточные преобразования и выход. У простой модели промежуточных слоёв может не быть.

Персептрон — ранний простой вид обучаемой модели. Он получает признаки, умножает каждый на свой вес, складывает значения и сравнивает сумму с порогом. Вес определяет вклад признака в ответ. Для двух признаков при сумме не ниже порога ответ будет 1, иначе — 0.

Во время обучения правильный ответ уже известен. Если модель ошиблась, её веса меняют. В этом примере параметры — веса, то есть числовые настройки вычисления ответа. Если модель получает только два признака рисунка, её ответ будет зависеть от них; остальные детали до неё не дойдут.



Пороговое правило превращает числовые признаки в ответ. Стрелка обратного хода обозначает обучение по известному верному ответу, а не физическое соединение с живым мозгом.

Говорим ученикам

Посмотрим на короткий пример: модель должна отвечать «да», когда в знаке есть оба штриха — горизонтальный и вертикальный. Вначале она ошибается. После разбора двух обучающих примеров веса меняются, и на этих примерах ответы становятся верными. Вычисления покажет учитель; проследите, почему при ошибке пришлось изменить веса.

<details>
<summary>Для учителя: короткая демонстрация обучения</summary>

Модель отвечает 1, когда есть и горизонтальный, и вертикальный штрих, иначе 0. Порог 2; начальные веса (1,0). Покажите два вычисления сами: на входе (1,1) сумма 1 даёт ошибочный 0, поэтому веса становятся (2,1). На входе (1,0) сумма 2 даёт ошибочный 1, поэтому веса становятся (1,1). Теперь оба примера распознаются верно. Ученикам достаточно объяснить, что менялись веса, а не выполнять серию вычислений.

При желании нарисуйте рядом итоговую таблицу: (1,1) → 1; (1,0), (0,1), (0,0) → 0. Во время проверки веса не меняются. Это маленькая модель; не переносите её простое правило обучения на все виды нейросетей.
</details>

Задание 2. Чего модель не увидела

В группе придумайте два разных значка для своего увлечения или школьного дела. Дайте им разные значения и объясните значения друг другу. Нарисуйте макет сети: рисунок, два входных признака, вычисление с весами, ответ. На вход модель получает только сведения «есть горизонтальный штрих» и «есть вертикальный штрих».

Исследуйте, какие детали ваших значков исчезнут после такого описания. Доработайте рисунки так, чтобы два разных по смыслу значка получили одинаковые ответы на оба вопроса о штрихах. Обсудите, сможет ли эта модель различить их, даже после обучения. Предложите ещё один признак, по которому модель могла бы различать рисунки.

▼ Для учителя: ответы и обсуждение

Оценивать рисунки, путь сведений через макет и объяснение потери информации. Например, два знака могут содержать оба штриха, но отличаться изгибом, замкнутостью или расположением деталей. Одинаковые входные значения у одной и той же модели дают одинаковый ответ; новые веса не восстановят не переданный признак. Дополнительный признак должен действительно различать выбранные рисунки. Правило «оба штриха → 1» решает только свою задачу и не определяет значение значка.

Говорим ученикам

В показанном обучении ответы изменились, потому что изменились веса связей. В больших нейросетях вычислительных элементов и связей намного больше; обучение также меняет их числовые настройки по данным. Подробное устройство многослойных сетей изучим в следующих модулях.

06. Этап 4. Что осталось непроверенным 7 минут

Говорим ученикам

У рассмотренной модели только два признака. Она не получает цвет, размер или смысл знака. Теперь понятно, зачем выбирать признаки под задачу: чтобы различить значения ваших значков, модели нужны сведения, в которых значки действительно различаются. Ответ в разговоре тоже полезно проверить после изменения условия: так мы увидим, учитывает ли программа новую информацию.

В начале урока вы придумали вопрос к программе. Уточним его так, чтобы можно было проверить действие: например, изменится ли объяснение, когда изменятся важные условия задачи.

Задание 3. Вопрос, который проверяет смысл

Вернитесь к вопросу из начала урока. Превратите его в небольшой разговор: после первого ответа измените одно существенное условие и задайте уточняющий вопрос. Объясните партнёру, что в ответе покажет понимание связи между условиями. Можно использовать свой диалог или рисунки из урока.

▼ Для учителя: ответы и обсуждение

Принимать разные разговоры, если новое условие требует изменить содержание ответа. Нужно указать, что именно должно измениться и почему. Не требовать доказывать наличие сознания по одной беседе.

07. Этап 5. Итоги урока 6 минут

Говорим ученикам

Мы рассмотрели два способа получать ответы: заранее заданные правила и обучение модели на примерах. Персептрон помог увидеть, как меняются веса и почему важен выбор признаков. Теперь можно исследовать более сложные модели, задавая вопросы о том, какие сведения они получают и как учатся решать задачу.

Главное за урок

- Диалоговая программа может строить ответ по заранее заданному шаблону или использовать модель, обученную на примерах.
- В персептроне признаки описывают рисунок числами, веса задают вклад каждого признака, а сравнение общей суммы с порогом определяет ответ.
- При обучении персептрона программа меняет веса после ошибок, а при проверке сохраняет веса и оценивает ответы на новых примерах.
- Если два разных рисунка описаны одинаковыми признаками, одна и та же модель получит одинаковые входные значения и не сможет различить эти рисунки.

Итоговые вопросы

1. Как признаки рисунка, веса и порог участвуют в получении ответа персептрона?
2. Что меняется в персептроне после ошибки при обучении и что сохраняется при проверке на новых примерах?
3. Почему персептрон не различит два рисунка с разным смыслом, если получил о каждом только одинаковые сведения о горизонтальном и вертикальном штрихах?
4. Почему ответ, полученный простой подстановкой слов в шаблон, может пропустить отрицание или изменившееся условие реплики?

▼ Для учителя: ответы и обсуждение

1. Признаки дают числа на входе, каждый умножается на вес, сумма сравнивается с порогом. 2. При обучении меняются веса, при проверке веса сохраняют. 3. Различия не дошли до модели; новые веса не восстановят отсутствующие сведения. 4. Простой шаблон преобразует выделенные фрагменты текста и может не учитывать смысл связи между частями фразы.

▼ Для учителя: источники

Joseph Weizenbaum. ELIZA, Communications of the ACM. Январь 1966 года. Проверено 03.11.2026. Материал старше 2024 года.

Основание: Первичная статья описывает ключевые слова и правила преобразования входных предложений в ответы ELIZA.

Подтверждающая фраза: «Responses are generated by reassembly rules».

Cornell University. Professor’s perceptron paved the way for AI. Сентябрь 2019 года. Проверено 03.11.2026. Материал старше 2024 года.

Основание: Персептрон Розенблатта — ранняя обучаемая модель классификации; предсказание по входным признакам и изменение параметров при ошибках.

Подтверждающая фраза: «an algorithm that classifies input into two possible categories».