Искусственный нейрон: входы, веса и выход

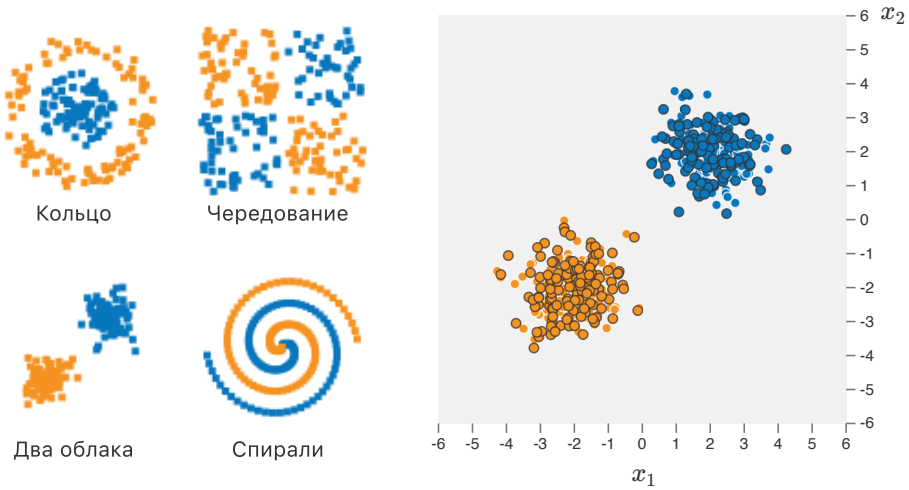
Робот-разведчик знает несколько безопасных и опасных мест на карте. Как ему оценить место, где он ещё не был?

Представим игру: роботу-разведчику нужно исследовать карту, на которой есть безопасная земля и опасные участки. Он уже побывал в нескольких местах: удачные посещения отмечены синим, неудачные — оранжевым. Теперь ему предстоит отправиться туда, где отметок ещё нет. Как понять, что его там ждёт?

Положение места задают два числа: горизонтальная координата x_1 и вертикальная x_2. Нам нужно правило, которое по этим двум входам предскажет один из двух ответов: «безопасно» или «опасно». Запомнить журнал посещений недостаточно: для нового места в нём ещё нет ответа. И заранее обещать точный прогноз для любой точки мы не можем — его качество предстоит проверить.

Посмотрите на четыре возможные карты. Каждая точка — известный пример, цвет — его класс. Какую границу вы провели бы между классами на каждой карте?

Четыре карты для робота



Два облака. Синий класс — безопасно, оранжевый — опасно.

Выберите картинку: увидите карту крупнее. Точки — известные примеры; синий класс условно безопасный, оранжевый — опасный.

Интерактивный эксперимент — в электронной версии урока.

Попробуем решить её с помощью **математической модели нейрона**. Сначала узнаем, что он вычисляет, затем сами настроим его на двух облаках. Только после

Заметка

Это **классификация на два класса**. Карты синтетические, из TensorFlow Playground: они помогают исследовать модель, но не описывают реальную местность. Построение маршрута — отдельная задача.

этого поручим подбор настроек алгоритму.

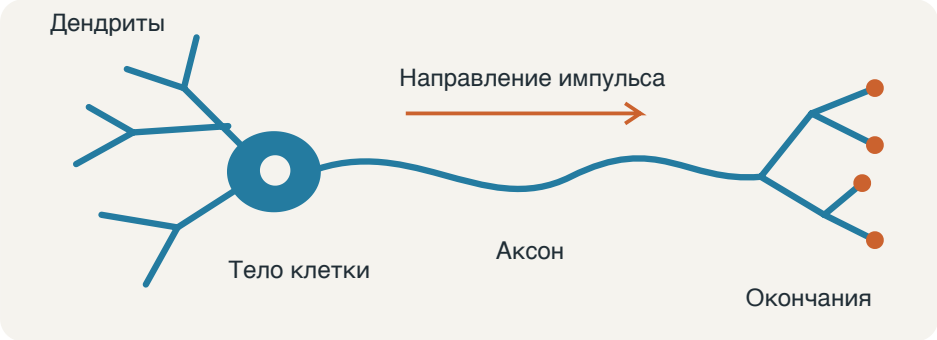
1. От живого нейрона к математической модели

Название пришло из биологии. В мозге взрослого человека около $8{,}6 \times 10^{10}$ нейронов — примерно 86 миллиардов. Между нейронами есть контакты, **синапсы**: через них клетки влияют друг на друга. Масштаб связей ещё больше: для неокортекса приводят оценку порядка $1{,}5 \times 10^{14}$ синапсов, то есть около 150 триллионов.

> **Заметка**
>
> Это оценки, а не точные константы для каждого человека. Azevedo и соавторы измеряли число нейронов в мозге взрослых мужчин; работа Pakkenberg и соавторов о синапсах относится к неокортексу. Числа характеризуют разные масштабы устройства мозга.

Сигналы распространяются по отросткам нейронов. Это визуальная иллюстрация, не запись измерений активности мозга.

Живой нейрон принимает сигналы через контакты с другими клетками. Их совместное действие влияет на появление импульса, который распространяется по аксону к следующим клеткам. Входящие влияния бывают возбуждающими и тормозными; устройство клетки существенно сложнее нашего будущего расчёта.



Дендриты и тело клетки принимают сигналы; аксон проводит импульс к окончаниям. Синапсы связывают клетки. Схема показывает общую организацию, без масштаба и деталей биофизики.

Возьмём из этой аналогии только идею: **несколько входов влияют на один выход**. У искусственного нейрона каждый вход умножается на свой **вес**, произведения складываются, затем результат преобразуется по выбранному правилу. Нейрон возвращает **одно число**. В нашем примере входы — координаты, а не сигналы живых клеток.

Координаты		Вклады входов		Сумма		Один выход
x_1, x_2	$\rightarrow$	w_1x_1, w_2x_2	$\rightarrow$	$s = w_1x_1 + w_2x_2$	$\rightarrow$	$a = \tanh(s)$

Два входа, два веса, один нейрон и одно выходное число. Правило Tanh сохраняет знак суммы и сжимает выход в интервал от –1 до 1; его график разберём чуть ниже. На карте нейрон всегда использует Tanh; ReLU и другие варианты сравним отдельно в шаге 3.

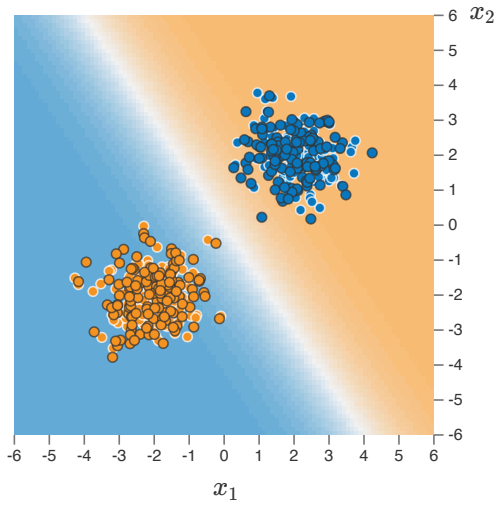
Договоримся о **решающем правиле**: если $a > 0$, прогнозируем «безопасно»; если $a \leq 0$ — «опасно». Это наша договорённость о смысле числа. Сам нейрон выдаёт число, а превращает его в название класса отдельное правило. Насыщенность фона показывает величину выхода; её пока нельзя читать как вероятность безопасности.

2. Входы и веса

Начнём с двух облаков. На карте синие точки обозначают безопасные места, оранжевые точки — опасные. Фон показывает **прогноз** нейрона. Оранжевая точка на синем фоне — ошибка: известный класс и прогноз разошлись.

Настройте сами. Двигайте два веса, пока все точки не окажутся на фоне своего цвета. Затем нажмите на картинку «Другая диагональ» и повторите. Обучение тоже ещё не запускается — каждую настройку выбираете вы.

Разделите облака двумя весами



Два облака. Синий класс — безопасно, оранжевый — опасно.

Состояние опыта: 0 проходов. Скрытых слоёв нет.

Веса: -0,6; -0,4.

Для каждого пикселя считаем сумму: $s = w_1 x_1 + w_2 x_2$

Выберите карту-картинку и меняйте два веса. Схема связей и большая карта показывают работу того же нейрона.

Интерактивный эксперимент — в электронной версии урока.

Положительный вес означает, что рост соответствующего входа увеличивает сумму; отрицательный — уменьшает. При нулевом весе этот вход не влияет на ответ. Поэтому вес горизонтальной координаты меняет влияние движения вправо, а вес вертикальной — движения вверх. Веса общие для всей карты: для каждой точки мы подставляем новые координаты в одно и то же правило.

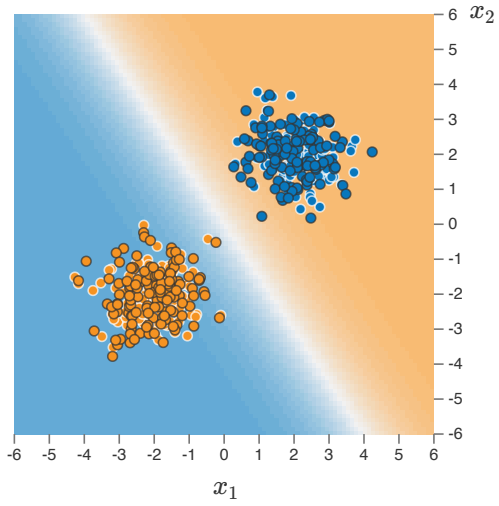
Например, при $x_1 = 2$, $x_2 = 1$, $w_1 = 3$ и $w_2 = -2$ сумма равна $3 \cdot 2 + (-2) \cdot 1 = 4$. Tanh сохраняет положительный знак, поэтому решающее правило даст «безопасно». Для другой точки изменятся координаты, но веса останутся теми же.

Как передвинуть границу

В первом опыте изменение весов поворачивает границу, но она всегда проходит через начало координат. А если нужная граница проходит в стороне? Добавим к сумме ещё одно настраиваемое число — **смещение** b: $s = w_1 x_1 + w_2 x_2 + b$.

Попробуйте. В следующем опыте установите оба веса равными 1. Теперь меняйте только смещение от −3 до 3: прямая движется, сохраняя направление. Затем оставьте смещение и измените один вес. Чем эти движения отличаются?

Смещение передвигает границу



Два облака. Синий класс — безопасно, оранжевый — опасно.

Состояние опыта: 0 проходов. Скрытых слоёв нет.

Веса: -0,6; -0,4. Смещение: 0.

Для каждого пикселя считаем сумму: $s = w_1 x_1 + w_2 x_2 + b$

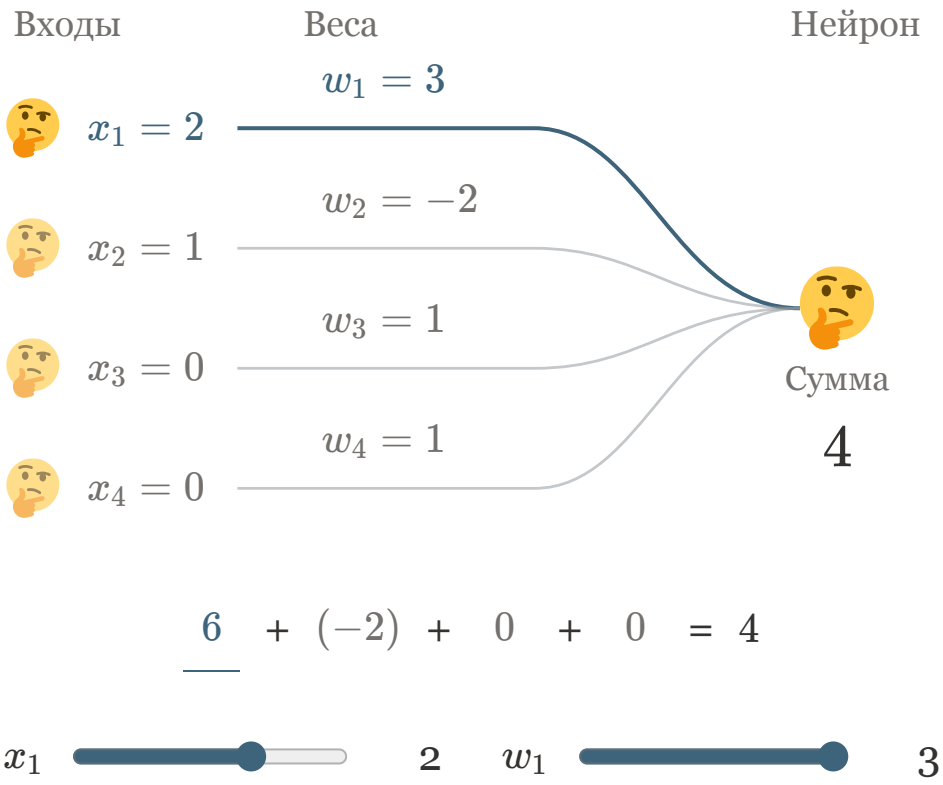
Два веса задают направление прямой; смещение позволяет сдвинуть её. Все три числа — общие настройки для всей карты.

Интерактивный эксперимент — в электронной версии урока.

Например, в точке $(2, 1)$ при обоих весах 1 сумма вкладов равна 3. Смещение -4 меняет её на -1, и ответ переходит на опасную сторону. **Для простоты дальнейших схем будем иногда опускать смещение**, но у нейрона это тоже настраиваемый параметр: алгоритм может подбирать его вместе с весами. В опытах с автоматическим обучением оно участвует в вычислении.

Вы только что подобрали веса вручную. **А можно ли сделать так, чтобы программа подбирала их за нас?** Можно: дать ей примеры с известными классами, измерять ошибку и понемногу менять веса. Это и будет обучение. Прежде чем нажать кнопку, разберём оставшиеся части нейрона — тогда будет понятно, что именно меняет алгоритм.

У нейрона может быть больше двух входов. На следующей схеме их четыре: например, к координатам робота можно добавить два показания датчиков. Каждый вход получает свой вес, а выход по-прежнему один. Вычисление такое же, как на карте; при нулевых третьем и четвёртом входах здесь получается наш пример $6 - 2 = 4$.



$2 \cdot 3 = 6$

Положительный вес: увеличение входа увеличивает сумму.

Исходный пример

Выберите связь и меняйте её вес. Предскажите, как изменится сумма, если вес отрицательный или нулевой.

Интерактивный эксперимент — в электронной версии урока.

> У нейрона входы 3 и 1, соответствующие веса 2 и −1. Какова сумма двух вкладов?

Заметка

Вход описывает конкретный пример. **Вес** — общая настройка модели. Переход к соседней точке меняет входы; обучение меняет веса.

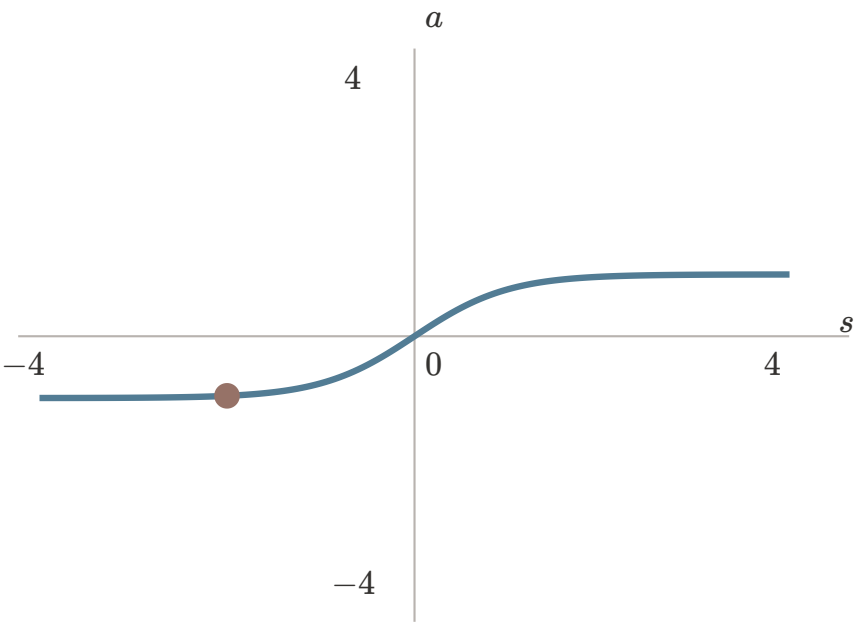
3. Функция активации

Функция активации преобразует сумму s в выход a. До сих пор мы использовали Tanh. Сравним четыре варианта: линейную функцию, ReLU, сигмоиду и Tanh. В каждом случае нейрон получает ту же сумму, но выдаёт другое число.

Попробуйте. Установите сумму -2 и переключите все четыре функции. Затем сделайте сумму положительной. Где выход стал нулём, где сохранил знак, а где остался положительным?

Одна сумма — четыре функции активации

Активация Tanh



Сумма до активации

$s = -2{,}00 \quad \longrightarrow \quad a = -0{,}96$

$$a = \tanh(s)$$

Выход между −1 и 1. Знак суммы сохраняется; эту активацию используем на карте.

Точка на графике связывает сумму по горизонтали с выходом по вертикали. Меняйте сумму и сравнивайте правила.

Интерактивный эксперимент — в электронной версии урока.

ReLU превращает отрицательную сумму в ноль, а положительную пропускает без изменения: при $s = -3$ выход 0, при $s = 4$ выход 4. **Сигмоида** даёт выход между 0 и 1; при нулевой сумме получается 0,5. **Tanh** даёт выход между −1 и 1 и сохраняет знак суммы. **Линейная активация** просто возвращает саму сумму.

Порог решения должен соответствовать выходу. Для Tanh на нашей карте это 0; для сигмоиды той же границе соответствовал бы порог 0,5. Мы не будем переключать активацию карты: оставим Tanh и уже знакомые цвета.

Вспомните движение прямой в опыте со смещением. Она отделяет положительную сумму от отрицательной: на самой границе $s = 0$. Tanh меняет насыщенность цвета по её сторонам, но не изгибает эту границу. Роль активации и роль смещения различаются: первая преобразует сумму, второе прибавляется к ней ещё до преобразования.

> Сумма с учётом смещения равна −3. Что выдаст нейрон с ReLU?
>
> ○ 3
>
> ○ Любое число от 0 до 1
>
> ○ −3
>
> ○ 0

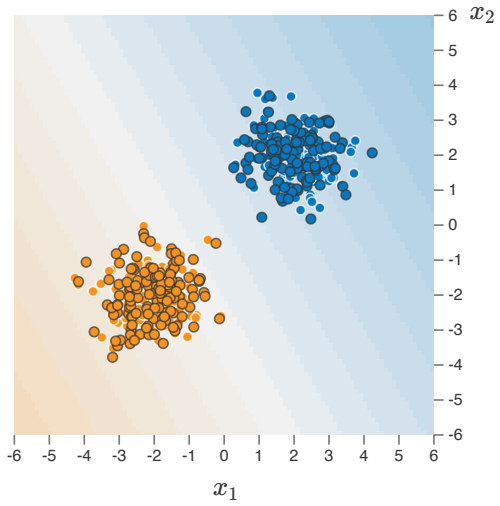
4. Обучение и применение

Вернёмся к нашему вопросу: как поручить подбор весов программе? Для каждой учебной точки известен желаемый ответ: +1 для безопасного места и −1 для опасного. Сравним выход нейрона с этим ответом. Чем сильнее расхождение, тем больше **ошибка**, которую измеряет выбранная **функция потерь**.

Алгоритм делает небольшое изменение двух весов и смещения, чтобы уменьшить ошибку на обучающих точках, затем повторяет расчёт. Точки и их метки остаются прежними. Ниже всё уже настроено: один нейрон, два входа, Tanh. Нам нужны только запуск, пауза и отдельный шаг.

Попробуйте. Нажмите кнопку одного прохода ▸| несколько раз и проследите за весами. Затем запустите обучение: как движется граница и как меняется ошибка? Остановите его и рассмотрите фон вокруг точек: это ответы уже обученного нейрона для разных координат.

Теперь веса подбирает алгоритм



Два облака. Синий класс — безопасно, оранжевый — опасно.

Состояние опыта: 0 проходов. Скрытых слоёв нет.

Веса: 0,056; 0,032. Смещение: 0,1.

Ошибка на обучении: 0,343; на проверке: 0,352.

Теперь доступны запуск, пауза и один проход. Входы, активация и число нейронов закреплены; сравните ошибку на обучении и проверке.

Интерактивный эксперимент — в электронной версии урока.

Когда мы нажимаем ▶, программа **подбирает параметры**. Когда после паузы рассматриваем ответ для нового места, она **применяет модель**: получает координаты и вычисляет один выход с неизменными весами. Для этого правильный ответ заранее не нужен. Хороший результат на известных местах ещё не гарантирует такого же результата на всей карте — поэтому мы оставили отдельные точки для проверки.

> Обучение остановлено. Робот оценивает соседнее место на той же карте. Что меняется в вычислении?
>
> ○ Число слоёв сети
>
> ○ Все веса обязательно подбираются заново
>
> ○ Правильные метки всех точек
>
> ○ Входные координаты; веса и смещение остаются прежними

5. Ограничения одного нейрона

Теперь выберите в этом же тренажёре **«Кольцо»**. В центре безопасная земля, вокруг неё — опасная область. У нейрона по-прежнему два входа, два веса и смещение. Запустите

Заметка

Половина точек участвует в обучении, половина отложена для проверки. Алгоритм меняет параметры только по первой половине. На графике — половина среднего квадрата расхождения выхода и метки, **не процент неверных ответов**. В ручном опыте мы считали верные классы; здесь следим за величиной расхождения числового выхода и метки.

обучение и посмотрите, сможет ли граница окружить центр.

Человеку нужна замкнутая граница, а модель умеет только проводить прямую. Как её ни поверни и ни передвинь, часть внешних точек останется на той же стороне, что и центральные. Продолжение обучения не меняет это ограничение: алгоритм ищет подходящие параметры внутри выбранной модели.

Представим несколько нейронов: один помогает ограничить участок слева, другой справа, ещё два — сверху и снизу. Их ответы можно передать следующему нейрону, который объединит эти числа. Так из ограничения одного блока возникает идея **скрытого слоя**.

Следующий сюжет

В «полносвязных слоях и многослойных сетях» продолжим задачу с кольцом: добавим нейроны и проследим, как их ответы складываются в общий прогноз. Затем свяжем эту же схему с классификацией изображений.

Вход описывает конкретный пример; веса и смещение задают общее правило.

Нейрон собирает взвешенную сумму, применяет активацию и возвращает одно число.

Обучение подбирает параметры по примерам. Одному нейрону с исходными координатами не хватает возможностей для кольца.

Домашняя работа

1. В тренажёре «Теперь веса подбирает алгоритм» сравните «Два облака» и «Кольцо». Для каждой карты выполните 100 проходов по данным, остановите обучение и запишите обе ошибки. Объясните разницу по форме границы.

2. Вход равен 5, вес равен 2, смещение равно −3. Чему равна сумма до активации?

Источники

- Azevedo et al., 2009: оценка числа нейронов (https://pubmed.ncbi.nlm.nih.gov/19226510/)
- Williams, Herrup, 1988: число нейронов и его регуляция (https://pubmed.ncbi.nlm.nih.gov/3284447/)
- Pakkenberg et al., 2003: нейроны и синапсы неокортекса (https://pubmed.ncbi.nlm.nih.gov/12543266/)
- Полносвязные нейросети: нейрон и активация (https://education.yandex.ru/handbook/ml/article/pervoe-znakomstvo-s-polnosvyaznymi-nejrosetyami)

- Оригинальный тренажёр взят из TensorFlow Playground (https://playground.tensorflow.org/).
  Авторы: Daniel Smilkov и Shan Carter.

Повторить эту тему