

Начальные сведения

Нейронные сети

Многие идеализируют нейронные сети, считая это чудом искусственного интеллекта, но этой статьей хотелось бы развеять мифы о том, что лежит в их основе и как они работают. Как говорит нам Википедия и многие определения в научно-популярной литературе *“Нейро́нная сеть (также искусственная нейронная сеть, ИНС) — это математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма”*. *Искусственный нейрон*, в свою очередь - это «часть» искусственной нейронной сети, представляющая собой часто простую функцию одного аргумента.

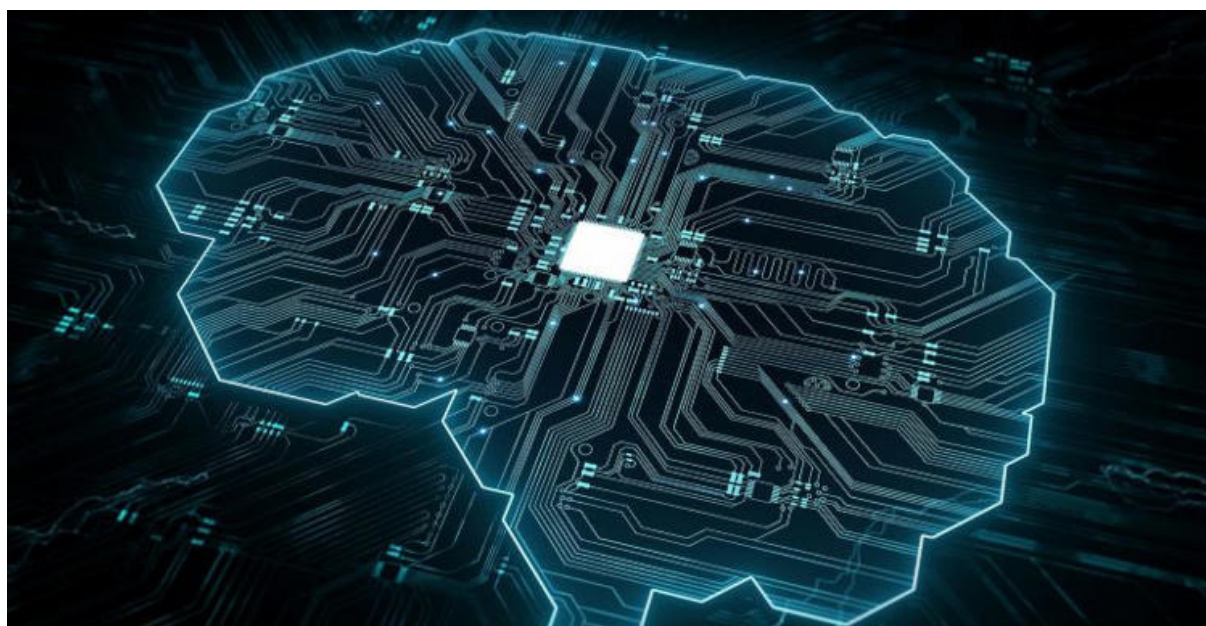
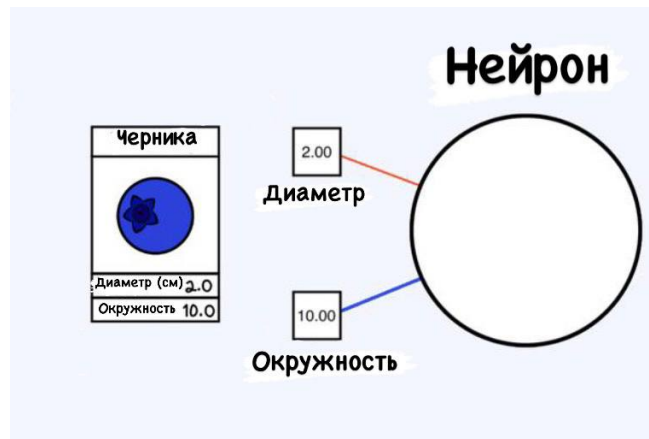
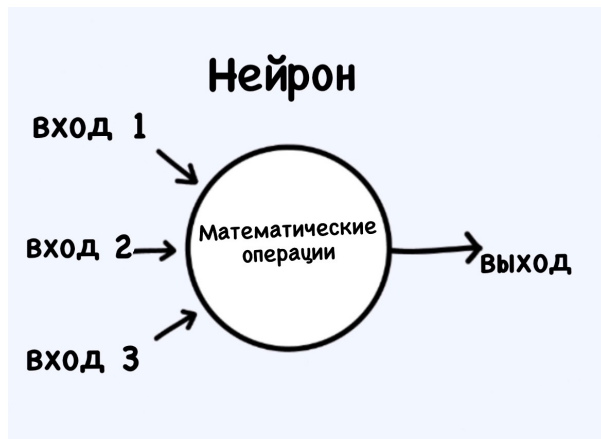


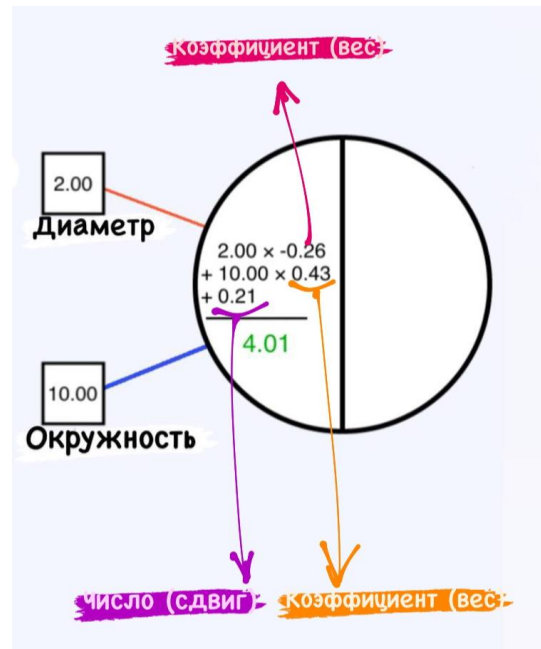
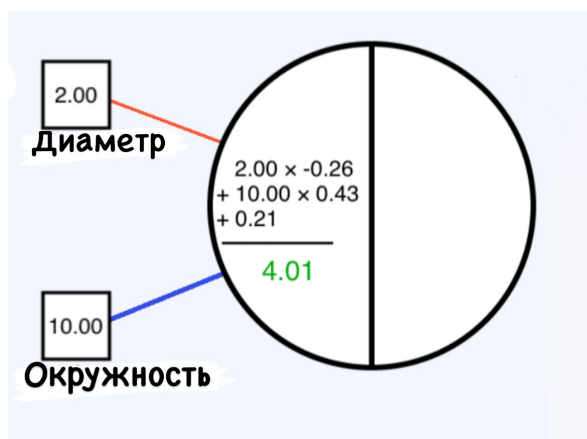
Фото: <https://3dnews.ru/1047285/postroennaya-po-obraztsu-chelovecheskogo-mozga-neyroset-uspeshno-vipolnila-kognitivnie-zadaniya/amp>

Итак, нейрон являет собой ядро вычислений математических операций с несколькими входными данными и выходной переменной.



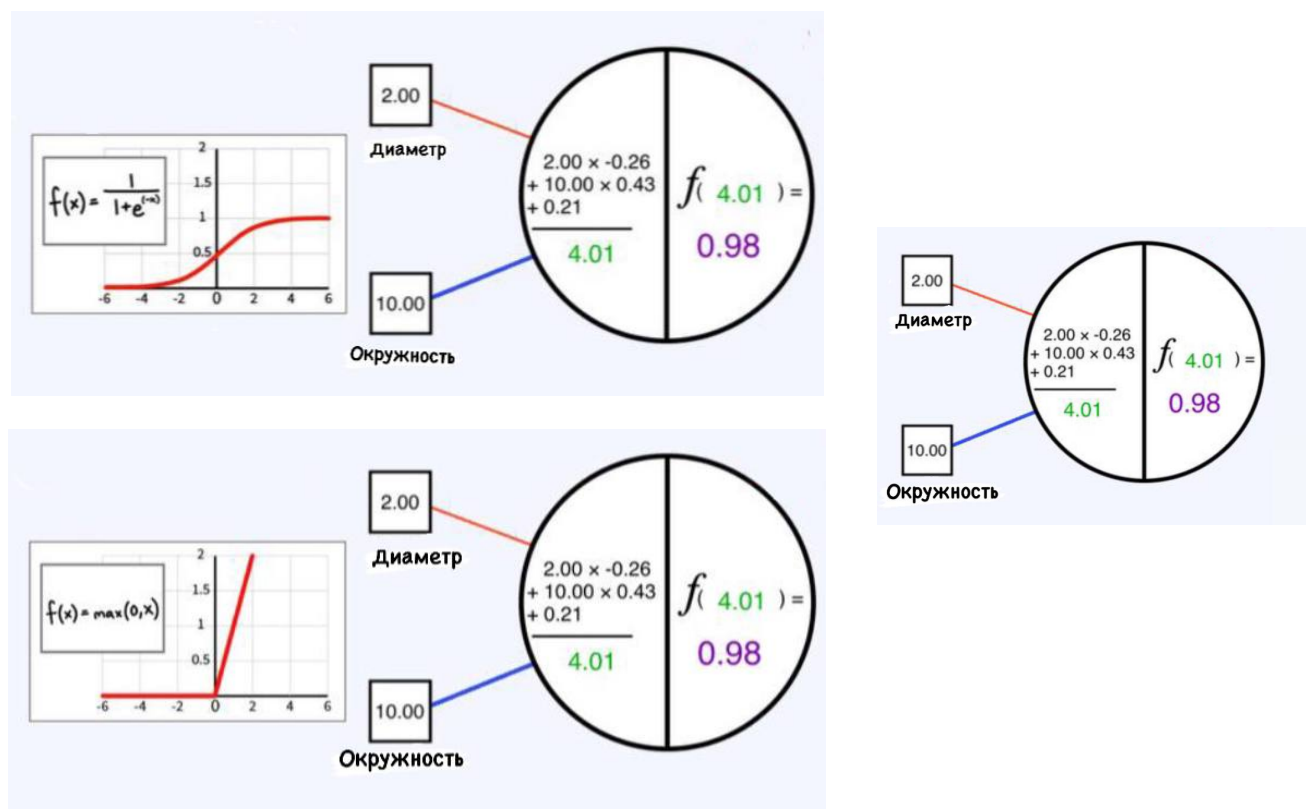
Пусть на входе нейрона у нас есть две переменные — диаметр ягоды и окружность. В более сложной системе входами нейрона являются выходные сигналы других нейронов.

Каждый вход в результате перемножается с определенным числом, называемым *весом* или *весовым коэффициентом*, складывается в линейную зависимость с другими входами, умноженными на коэффициент и суммирующийся с числом, называемым *сдвигом*. Веса и сдвиг задаются оператором, реализующим нейронную сеть.



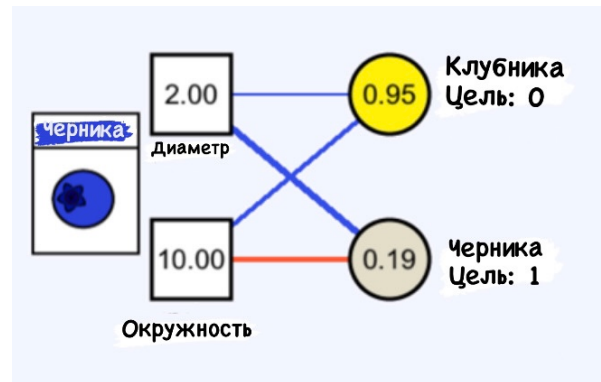
Полученный результат линейной функции используется как аргумент в одномерной функции активации. Функции активации могут быть

различными. Самая широко используемая — логистическая функция, имеющая пределы функции от 0 до 1. Ещё одна из распространённых функций активации — *Rectified Linear Unit (ReLU function)* - усеченное линейное преобразование. $\text{ReLU}(x) = \max(0, x)$.

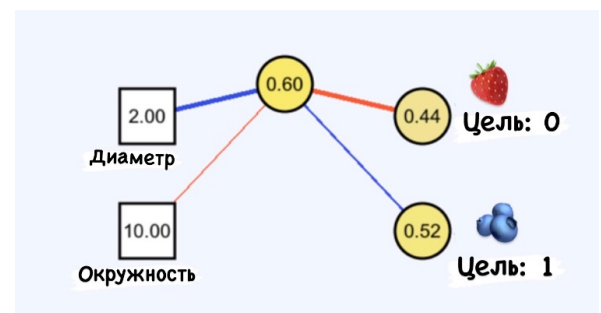


В результате каждый нейрон оперирует входными сигналами, который он получает, и сигналами, которые он посылает другим нейронам. Отдельные нейроны при совокупности образуют нейронную сеть.

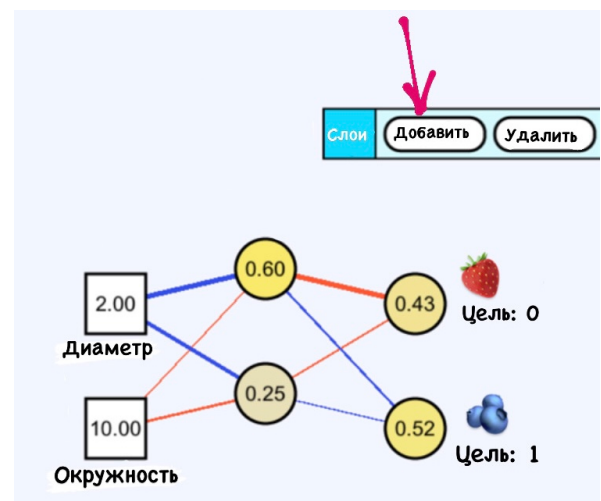
Классическая задача, которую с успехом решают нейронные сети — классификация. Рассмотрим её на примере ягод — черники и земляники. Предположим, что задача нейронной сети — отличать ягоды. Таким образом на выходе итогового нейрона будет или 0, или 1 — это целевые значения.



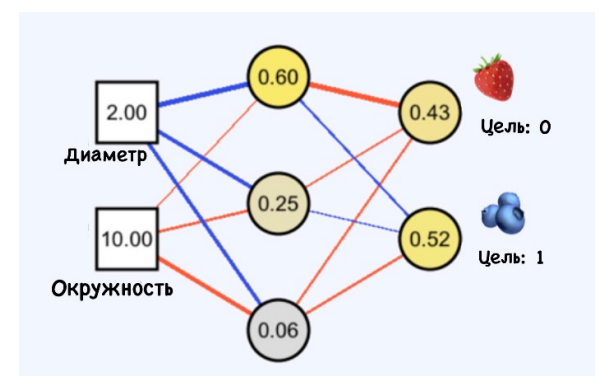
Для наиболее точного определения класса определяемой переменной используют скрытые слои, где выход каждого слоя — вход для нового, так называемый процесс *forward propagation* (алгоритм прямого распространения).



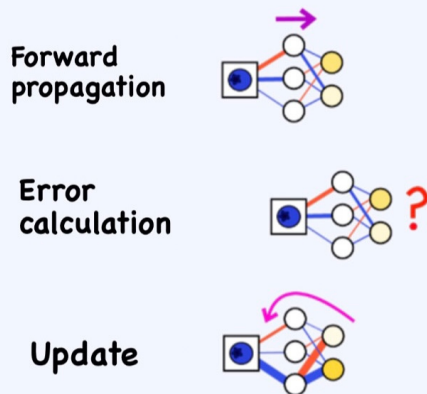
В скрытый слой возможно добавить дополнительный нейрон.



В нейронной сети дополнительные нейроны в скрытых слоях ограничены лишь вычислительными способностями аппаратной части. При наличии дополнительных нейронов возможно проектирование нейронной сети с алгоритмом *back propagation* (алгоритм обратного распространения)



Back propagation

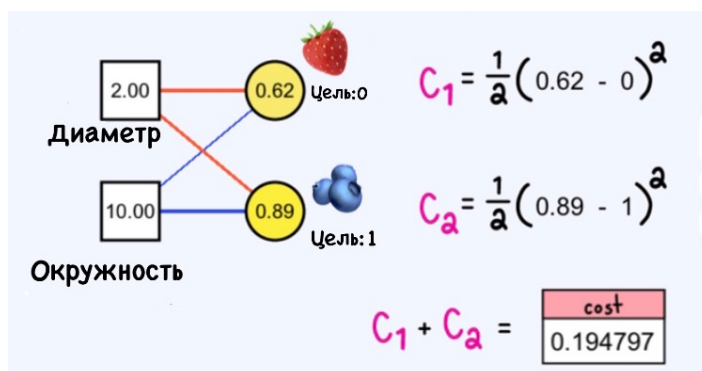


Алгоритм обратного распространения состоит из трёх шагов:

- *Forward propagation* (алгоритм прямого распространения) — задаём входные значения, вычисляем выходные;
- *Error calculation* (вычисление ошибки) — показывает какова точность произведённых расчетов;
- *Update* (обновление весов) — изменение весов в функциях для снижения ошибки расчета.

$$C(w, b) = \frac{1}{2n} \sum_x (\text{target}_x - \text{activation}_x)^2$$

Ошибка рассчитывается с помощью *cost function* (функция потерь), которая для каждой системы подбирается индивидуально. В приведённом примере функция потерь $C(w, b)$ оперирует следующими переменными — n - количество единиц данных, x - количество нейронов и является функцией разности целевого значения и полученного (функции активации). Выходное значение зависит от весов и от свободного коэффициента (сдвига).



Расчеты для нашего примера приведены функция потерь составляет 0.19.

В следующей части мы обсудим какими способами можно найти минимум функции потерь, а также архитектуру нейронных сетей.