**Определение**

**Свёрточная нейронная сеть** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *convolutional neural network*, *CNN*) — специальная архитектура [искусственных нейронных сетей](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), предложенная [Яном Лекуном](https://ru.wikipedia.org/wiki/%D0%9B%D0%B5%D0%BA%D1%83%D0%BD,_%D0%AF%D0%BD) в 1988 году[[1]](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C#cite_note-1) и нацеленная на эффективное [распознавание образов](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D1%80%D0%B0%D1%81%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2)[[2]](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C#cite_note-deeplearning-2), входит в состав технологий [глубокого обучения](https://ru.wikipedia.org/wiki/%D0%93%D0%BB%D1%83%D0%B1%D0%BE%D0%BA%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *deep learning*). Использует некоторые особенности [зрительной коры](https://ru.wikipedia.org/wiki/%D0%97%D1%80%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BA%D0%BE%D1%80%D0%B0)[[3]](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C#cite_note-robust_face_detection-3), в которой были открыты так называемые простые клетки, реагирующие на прямые линии под разными углами, и сложные клетки, реакция которых связана с активацией определённого набора простых клеток. Таким образом, идея свёрточных нейронных сетей заключается в чередовании свёрточных слоёв ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *convolution layers*) и субдискретизирующих слоёв ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *subsampling layers* или [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *pooling layers*, слоёв подвыборки). Структура сети — однонаправленная (без обратных связей), принципиально многослойная. Для обучения используются стандартные методы, чаще всего [метод обратного распространения ошибки](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BE%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D1%80%D0%B0%D1%81%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BE%D1%88%D0%B8%D0%B1%D0%BA%D0%B8). [Функция активации нейронов](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%B0%D1%86%D0%B8%D0%B8) (передаточная функция) — любая, по выбору исследователя.

Название архитектура сети получила из-за наличия операции [свёртки](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BA%D0%B0_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7)), суть которой в том, что каждый фрагмент изображения умножается на матрицу (ядро) свёртки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения.

**Структура сверточных нейронных сетей**

Входной слой логически представляется как матрица пикселей

• Каждый внутренний слой логически представляется как совокупность карт признаков (feature map) [карт активации (activation map)], имеющих одинаковые размеры

• Выходной слой имеет обычную линейную структуру и связывается с последним внутренним слоем по принципу «все-со-всеми»

• Нейроны входного слоя и всех внутренних слоев связываются избирательно (не все со всеми) • Связи строятся двумя чередующимися способами: 1) свертка, 2) пулинг

• Свертка увеличивает количество карт признаков, сохраняя их размер (при использовании набивки padding), либо незначительно уменьшая их размер

• Пулинг уменьшает размеры карт признаков с сохранением их количества

**Слой свертки**

Слой свёртки — это основной блок свёрточной нейронной сети. Слой свёртки включает в себя для каждого канала свой фильтр, ядро свёртки которого обрабатывает предыдущий слой по фрагментам (суммируя результаты поэлементного произведения для каждого фрагмента). Весовые коэффициенты ядра свёртки (небольшой матрицы) неизвестны и устанавливаются в процессе обучения.

Чтобы нейросеть «узнала» предмет, нужно проделать с изображением несколько типовых операций на каждом слое. Ключевая из этих операций — свёртка. Во время свёртки нейросеть удаляет лишнее и оставляет полезное — то, что поможет проанализировать изображение. Например, линии, края или ровные области. Свёртку можно создавать для каждого признака. Нейросеть будет сама подбирать их во время распознавания и классификации на каждом свёрточном слое.

**Принцип работы операции свертки**

Операцию свёртки можно представить следующим алгоритмом:

1. Скользящее окно, называемое фильтром, с размером (n,n) двигается по входному признаку. Количество движений определяется заданным количеством фильтров.
2. Каждый полученный шаблон имеет форму (n,n,d), где d — глубина входного признака.
3. Каждый шаблон умножается на своё ядро свёртки, в результате, формируется **выходная карта признаков**. Полученная выходная карта признаков имеет форму (h,w,N), где h и w — длина и ширина, полученные в результате отсечения, а N — количество фильтров.

Количество фильтров — гиперпараметр, поэтому выбирается самостоятельно. Обычно его подбирают как степень двойки с увеличением количества фильтров по мере увеличения глубины архитектуры. А ядра свёртки являются обучаемыми параметрами.

**Карты признаков**

* Одна карта признаков со своим ядром распознает один признак, например «уши»
* Для распознавания другого признака, например «клюва», добавляется еще одна карта признаков, имеющая свое собственное ядро
* На практике внутренний слой может включать в себя несколько десятков и даже сотен карт признаков

**Слой активации**

Скалярный результат каждой свёртки попадает на [функцию активации](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%B0%D1%86%D0%B8%D0%B8), которая представляет собой некую нелинейную функцию. Слой активации обычно логически объединяют со слоем свёртки (считают, что функция активации встроена в слой свёртки). Функция нелинейности может быть любой по выбору исследователя, традиционно для этого использовали функции типа [гиперболического тангенса](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D0%B1%D0%BE%D0%BB%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8) (𝑓(𝑥)=tanh⁡(𝑥), 𝑓(𝑥)=|tanh⁡(𝑥)|) или [сигмоиды](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%B3%D0%BC%D0%BE%D0%B8%D0%B4%D0%B0" \o "Сигмоида) (𝑓(𝑥)=(1+𝑒−𝑥)−1). Однако в 2000х годах была предложена[[4]](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C#cite_note-Hahnloser2000-4) и исследована[[5]](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C#cite_note-glorot2011-5) новая функция активации — ReLU (сокращение от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *rectified linear unit*), которая позволила существенно ускорить процесс обучения и одновременно упростить вычисления (за счёт простоты самой функции)[[6]](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C#cite_note-nair2010-6), что означает блок линейной ректификации, вычисляющий функцию 𝑓(𝑥)=max(0,𝑥) . То есть по сути это операция отсечения отрицательной части скалярной величины. По состоянию на 2017 год эта функция и её модификации (Noisy ReLU, Leaky ReLU и другие) являются наиболее часто используемыми функциями активации в глубоких нейросетях, в частности, в свёрточных. Существует методика определения оптимального числа блоков линейной ректификации[[7]](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C#cite_note-Romanuke4-7).

**Слой пулинга**

Из признаков, которые выделил свёрточный слой, выбирает самые важные, а несущественные удаляет. К результату, который получился во время пулинга, можно снова применить свёрточный слой и сделать несколько циклов. Это нужно, чтобы выстроить иерархию признаков: от самых примитивных, например кусочков контура, до более сложных, например глаз кошки или формы её ушей.

Первые слои нейросети анализируют мелкие детали изображения — ворсинки, трещины. Размер изображения может быть 2×2 или 3×3 пикселя — по такому маленькому контексту сложно выделить глаза, уши кошки или деревья, на фоне которых она сидит, но можно найти цвета или перепады света, то есть границы разных объектов. На следующих слоях появляются объекты посложнее — круги и другие формы.

Суть этого слоя заключается в уменьшении размерности карты признаков.

Pooling имеет две разновидности: max-pooling и average-pooling. В большинстве случаев применяется max-pooling. Операция Pooling схожа с операцией свертки:

1. Скользящее окно, обычно это окно (2,2), двигается по карте признаков.
2. Из выбранного шаблона выбирается максимальное (max-pooling) или среднее (average-pooling) значение.
3. Формируется уменьшенная в размере карта признаков.

На рисунке ниже показано, как из матрицы (4,4) получается выходная карта (2,2) после операции max-pooling и average-pooling.

**Полносвязный слой**

После нескольких прохождений свёртки изображения и уплотнения с помощью пулинга система перестраивается от конкретной сетки пикселей с высоким разрешением к более абстрактным картам признаков, как правило, на каждом следующем слое увеличивается число каналов и уменьшается размерность изображения в каждом канале. В конце концов, остаётся большой набор каналов, хранящих небольшое число данных (даже один параметр), которые интерпретируются как самые абстрактные понятия, выявленные из исходного изображения.

Эти данные объединяются и передаются на обычную полносвязную нейронную сеть, которая тоже может состоять из нескольких слоёв. При этом полносвязные слои уже утрачивают пространственную структуру пикселей и обладают сравнительно небольшой размерностью (по отношению к количеству пикселей исходного изображения).

**Пример сети**

Нейроны полносвязного слоя нейросети объединены со всеми нейронами входного слоя. Обычно это последний слой. Он должен выдать результаты нелинейной функции, с которой работала нейросеть. Все предыдущие слои нужны для обработки изображения, тогда как полносвязный слой должен выделить ключевые характеристики и выполнить классификацию — определить, что на картинке, или найти конкретный предмет.

В полносвязном слое много нейронов с разными весами, поэтому для вывода результатов применяют вектор нейронов. Например, нейросеть создана для определения типов транспорта. Ее задача — установить, к какому из 20 классов принадлежит объект. Алгоритм выдает результат вектора, на основании которого сеть делает прогноз.

**Обучение сети**

Нейросеть учится подбирать оптимальные веса нейронов и делать прогнозы с помощью **метода обратного распространения ошибки**.

Сначала тренировочное изображение проходит через всю сеть. На этом этапе веса установлены произвольно, поэтому на выходе сеть определит немного признаков и не сможет точно ответить, что изображено на картинке.

Здесь можно оценить функцию потерь — насколько отличается фактический выход алгоритма от прогнозированного.

Далее веса нейронов корректируют, начиная с ближайших к выходу. Это позволяет уменьшить функцию потерь и называется методом обратного распространения ошибки.

**Сфера применения**

В 2010-е годы искусственные сверточные нейронные сети имели широкое использование в медицине и государственных учреждениях при:

* распознавании рукописного текста;
* классификации документов, таких как СНИЛС или паспорт;
* распознавании на снимках новообразований и иных патологий.

В настоящее время сверточные нейронные сети применяются в системах видеонаблюдения, видеоаналитики и беспилотных автомобилях. Например, камеры в комплексе с такими сетями помогают роботам компании «Яндекс» и беспилотным такси понимать, что происходит на дороге.

Другой популярной областью применения принципа работы сверточных нейронных сетей является генеративное искусство. Когда человек вводит запрос, нейронная сеть генерирует на его основе изображение. Преобразование текста является задачей различных алгоритмов, а генерация изображения – сверточных нейронных сетей. Наиболее распространенными моделями для генерации изображений являются такие сверточные нейронные сети, как Midjourney, Stable Diffusion, Dream, DALL-E 2 и ruDALL-E.

Преимущества[[править](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C&veaction=edit&section=8) | [править код](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C&action=edit&section=8)]

* Один из лучших алгоритмов по распознаванию и классификации изображений.
* По сравнению с полносвязной нейронной сетью (типа перцептрона) — гораздо меньшее количество настраиваемых весов, так как одно ядро весов используется целиком для всего изображения, вместо того, чтобы делать для каждого пикселя входного изображения свои персональные весовые коэффициенты. Это подталкивает нейросеть при обучении к обобщению демонстрируемой информации, а не попиксельному запоминанию каждой показанной картинки в мириадах весовых коэффициентов, как это делает перцептрон.
* Удобное распараллеливание вычислений, а следовательно, возможность реализации алгоритмов работы и обучения сети на [графических процессорах](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80).
* Относительная устойчивость к повороту и сдвигу распознаваемого изображения.
* Обучение при помощи классического [метода обратного распространения ошибки](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BE%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D1%80%D0%B0%D1%81%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BE%D1%88%D0%B8%D0%B1%D0%BA%D0%B8).

Недостатки[[править](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C&veaction=edit&section=9) | [править код](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C&action=edit&section=9)]

* Слишком много варьируемых параметров сети; непонятно, для какой задачи и вычислительной мощности какие нужны настройки. Так, к варьируемым параметрам можно отнести: количество слоёв, размерность ядра свёртки для каждого из слоёв, количество ядер для каждого из слоёв, шаг сдвига ядра при обработке слоя, необходимость слоёв субдискретизации, степень уменьшения ими размерности, функция по уменьшению размерности (выбор максимума, среднего и т. п.), передаточная функция нейронов, наличие и параметры выходной полносвязной нейросети на выходе свёрточной. Все эти параметры существенно влияют на результат, но выбираются исследователями эмпирически. Существует несколько выверенных и прекрасно работающих конфигураций сетей, но не хватает рекомендаций, по которым нужно строить сеть для новой задачи.