

# Особенности работы сверточных нейронных сетей

В.О. Скрипачев, М.В. Гуйда, Н.В. Гуйда, А.О. Жуков

**Аннотация.** В статье детально рассмотрено применение свертывающихся нейронных сетей (CNNs) для применения в решении задач по обнаружению объектов на аэрокосмических снимках. Раскрыта структура свертывающихся нейронных сетей, учитывая специфику применения. Имея в виду, творческую сущность проблемы распознавания объектов на аэрокосмических снимках проиллюстрирована высокая вариативность решения поставленной задачи. Рассмотрены подходы к построению свертывающихся нейронных сетей с учетом ключевых факторов эффективности решения задачи обнаружения объектов на аэрокосмических снимках.

Особое внимание уделено внутренней структуре построения свертывающихся нейронных сетей. Описано выполнение в сетях операций кросс-корреляции (свертки). Наглядно показан механизм сдвига (шага) свертки. Описано применение двумерных фильтров для одноканального и трехканального изображений. Разобрано применение режимов дополнения изображения при свертке: без дополнения (режим «valid»), с дополнением (режимы «same» и «full»).

Рассмотрены наиболее часто применяемые фильтры в CNNs для определения характерных линий (горизонтальных, вертикальных, наклонных), фильтры улучшения границ (ребер, краев), фильтры сглаживающие экстремальные значения данных.

Сделан общий вывод о применении сверточных нейронных сетей и стратегии их применения с указанием наиболее приемлемой конструкции CNNs для обнаружения объектов на аэрокосмических снимках.

**Ключевые слова:** свертывающиеся нейронные сети (CNNs), многослойные перцептроны (MLP), свертка трехканального изображения с применением трехмерного фильтра, свертка слоя изображения двумерным фильтром, свертка с дополнением в режиме “full”, основные параметры сверточного слоя, выполнение операции свертки (кросс-корреляции), свертка с дополнением в режиме “same”, фильтры градиента для обнаружения ребер, фильтры Собеля для обнаружения ребер, высокочастотный фильтр (фильтр улучшения границ)

Статья получена 3 июня 2022.

В. О. Скрипачев, к.т.н., начальник отдела, Российский технологический университет (РТУ МИРЭА), Москва, Россия (e-mail: skripachev@mirea.ru).

М. В. Гуйда, к.т.н., старший научный сотрудник, Московский государственный университет имени М.В.Ломоносова (МГУ им. М.В. Ломоносова), Москва, Россия (e-mail: guida.mv21@physics.msu.ru).

Н. В. Гуйда, инженер 1 категории, Особое конструкторское бюро Московского энергетического университета (ОКБ МЭИ), Москва, Россия (e-mail: gujda.nv@okbmei.ru).

А. О. Жуков, д.т.н., профессор, ведущий научный сотрудник, Институт астрономии РАН, заместитель директора по научной работе ФГБНУ «Аналитический центр», Россия (e-mail: aozhukov@mail.ru).

(ребер, краев), низкочастотный фильтр, фильтры градиента для обнаружения ребер с шагом в 45 градусов, фильтры Лапласа для выявления ребер.

## I. ВВЕДЕНИЕ

Традиционная парадигма выражения правил на языке программирования не всегда может работать для решения проблемы. Такие сценарии, как, например, компьютерное зрение, очень сложно реализуются с помощью программирования на основе правил. Вместо этого, если выдать компьютеру достаточное количество данных, которые описаны (или маркированы) как то, что подлежит распознаванию. Учитывая, что компьютеры действительно хороши в обработке данных и поиске совпадающих шаблонов, то потенциально возможно «обучить» систему решать эту задачу.

Традиционное программирование может быть представлено в виде верхней диаграммы.

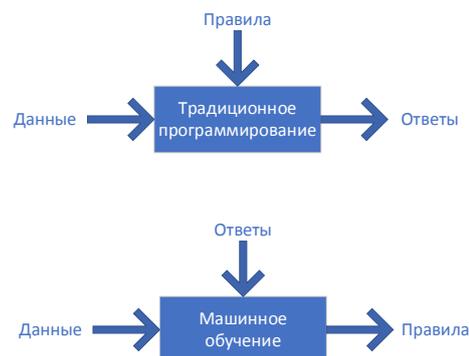


Рисунок 1 - Диаграммы традиционного программирования и машинного обучения.

Правила и данные поступают на вход, а ответы получаем на выходе. Правила выражаются на языке программирования, а данные могут приходиться из различных источников, начиная с локальных переменных и заканчивая базами данных.

Машинное обучение перестраивает эту диаграмму так, что ответы и данные подаются на вход, а правила получают на выходе. Кратко, смысл машинного обучения можно описать так: на вход компьютера подаются примеры того, что необходимо видеть, - и компьютер на основе этих данных формирует правила. Это особенно ценно для задач, которые невозможно решить выведением правил человеком.

Новая парадигма заключается в том, что сначала подготавливается множество примеров, которые затем соответствующим образом размечаются, чтобы дать компьютеру соответствие между изображением и классом объекта, который на нем изображен (это может

быть все что угодно: животное, ходьба, самолет, дождь и т.д.). Эти данные становятся входными и выходными данными для правил, которые выводит машина. Алгоритм машинного обучения определяет особые шаблоны в каждом наборе данных, определяющие отличия каждого набора. Применяя их затем на неразмеченных данных, машина может установить, что на изображении показаны ходьба, бег, или что-то иное. В этом заключается смысл данной парадигмы программирования. Это больше чем просто новый способ делать уже известную вещь. Это открывает новые возможности, которые было невозможно воплотить раньше. Сегодня, основной моделью машинного обучения является нейронная сеть, хотя имеется и множество других алгоритмов.

Самой простой реализацией нейронной сети является полносвязная нейронная сеть (FCNN). Полносвязные нейронные сети отлично работают, когда обрабатываются табличные данные. Под таблицей предполагается, что данные состоят из строк, соответствующих примерам, и столбцов, соответствующих характеристикам. С табличными данными возможно предположить, что шаблоны, которые подлежат вычислению, могут включать взаимодействия между признаками, но априорно не предполагают никакой структуры относительно того, как взаимодействуют эти признаки.

Иногда действительно не хватает знаний, чтобы строить более сложные сетевые архитектуры. В этих случаях многослойный перцептрон (MLP) может быть лучшим решением из того, что возможно применить для решения задачи. Однако для многомерных перцептивных данных такие сети без структуры могут стать слишком сложными для обучения.

Для примера рассмотрим нейронную сеть классификации картинок по двум классам. Предположим, что проводится тщательная работа по сбору информации, аннотированного набора данных в виде фотографий размером 1 мегапиксель. Это означает, что каждый вход в сеть имеет один миллион измерений, что является существенным значением для входной последовательности. Даже радикальное сокращение измерений до одной тысячи и использование специальных методов [3], потребует полносвязного слоя, характеризующегося параметрами  $10^6 * 10^3 = 10^9$ . Если нет значительного количества аппаратных ресурсов, таланта к распределенной оптимизации и терпения, изучение параметров этой сети может оказаться невозможным. Также, необходимо учитывать скрытый слой размером 1000 нейронов, которые необходимы для обеспечения хорошего представления изображений. Таким образом, работоспособная система по-прежнему будет требовать миллиарды параметров. Кроме того, для обучения классификатора по такому количеству параметров, может потребоваться сбор огромного набора данных.

Есть некоторые методы, которые позволяют снизить вычислительную сложность для полносвязной сети. Например, замена вещественных матричных умножений при расчете обучаемых параметров в полностью связанных слоях на гамильтонские произведения

кватернионов обеспечивает снижение вычислительной сложности до 1/4 от традиционной [3]. Этот подход может быть эффективно использован при работе с большими полносвязными сетями при ограниченных вычислительных ресурсах, например в небольших исследовательских коллективах, однако не обеспечивает универсального решения проблемы.

Несмотря на все вышесказанное, сегодня и люди, и компьютеры вполне могут отличать зрительные образы, например кошек от собак. И это некоторым образом противоречит приведенным выше оценкам. Причина заключается в том, что для распознавания образов на изображениях используются специальные нейронные сети.

## II. ИСПОЛЬЗОВАНИЕ СВЕРТЫВАЮЩИХСЯ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РАСПОЗНАВАНИЯ ОБЪЕКТОВ

Свертывающиеся нейронные сети (CNNs) - это один из творческих способов, который разработан для использования некоторых известных математических методов для распознавания объектов на изображениях по значениям их пикселей [4].

Представьте, что необходимо обнаружить объект на изображении. Представляется разумным, что какой бы метод ни использовался для распознавания объектов, он не должен быть чрезмерно связан с точным расположением объекта на изображении. В идеале, система должна уметь использовать знания о возможном расположении объекта относительно других объектов. В верхнем секторе картинки располагается водный массив, и алгоритм должен понять, что это не небо, несмотря на его расположение. Как пример, рассмотрим аэрофотоснимок плотной городской застройки (рисунок 2). На аэрофотоснимке изображено множество различных зданий. Цель - найти заданное здание, например, с количеством этажей более 4. Несмотря на известные признаки найти его может оказаться не так просто из-за множества отвлекающих факторов.

Сверточная нейронная сеть систематизирует эту идею пространственной инвариантности, используя ее для получения полезных представлений с меньшим количеством параметров.



Рисунок 2 – Плотная городская застройка.

На начальных этапах обработки изображения CNN должна работать одинаково, независимо от места размещения объекта на изображении. То есть должен соблюдаться принцип инвариантности перевода. Первые слои сети должны фокусироваться на локальных регионах без учета изображений в других регионах. В более глубоких слоях сети локальные представления агрегируются для предсказания объекта на уровне всего изображения.

### III. ПРИНЦИП РАБОТЫ СВЕРТЫВАЮЩИХСЯ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РАСПОЗНАВАНИЯ ОБЪЕКТОВ

Рассмотрим работу многослойных перцептронов (MLP) с двумерным изображением  $X$  в качестве входа и его промежуточные скрытые представления  $H$ , подобные матрицам в математике, где  $X$  и  $H$  имеют одинаковую форму. Пусть  $[[X]_{i,j}]$  и  $[H]_{i,j}$  описывают расположение пикселя с координатами  $(i,j)$  во входном и скрытом представлениях. Следовательно, чтобы каждая скрытая единица получала на вход каждый входной пиксель, необходимо использовать матрицу весов, представляющую параметры как тензор четвертого порядка  $W$ . Предположим, что  $U$  содержит сдвиги (bias), тогда скрытый полносвязный слой может быть выражен следующей формулой:

$$[H]_{i,j} = [U]_{i,j} + \sum_k \sum_l [W]_{i,j,k,l} [X]_{k,l} = [U]_{i,j} + \sum_a \sum_b [V]_{i,j,a,b} [X]_{i+a,j+b}, \quad (1)$$

где переход от  $W$  к  $V$  является полностью искусственным, поскольку существует однозначное соответствие между коэффициентами в обоих тензорах четвертого порядка. Необходимо переиндексировать индексы  $(k, l)$  так, что  $k = i + a$  и  $l = j + b$ . Другими словами,  $[V]_{i,j,a,b} = [W]_{i,j,i+a,j+b}$ . Индексы  $a$  и  $b$  охватывают как положительные, так и отрицательные смещения, охватывающие всё изображение. Для любого данного  $(i, j)$  в скрытом представлении  $[H]_{i,j}$ , необходимо вычислить это значение, суммируя пиксели в  $x$ , центрированные вокруг  $(i, j)$  и взвешенные по  $[V]_{i,j,a,b}$ . Так как  $V$  и  $U$  не зависят от  $(i, j)$ , что следует из свойства инвариантности, то получается  $[V]_{i,j,a,b} = [V]_{a,b}$  и  $U$  является константой, например  $u$ , тогда:

$$[H]_{i,j} = u + \sum_a \sum_b [V]_{a,b} [X]_{i+a,j+b}, \quad (2)$$

Эта операция представляет собой свертку. Фактически происходит взвешивание пикселей на  $(i + a, j + b)$  в непосредственной близости от точки  $(i, j)$  с коэффициентами  $[V]_{a,b}$ , что требует гораздо меньше коэффициентов, чем  $[V]_{i,j,a,b}$ , поскольку это не зависит от расположения на изображении.

Обработка информации осуществляется локально, в некоторой области вокруг  $(i, j)$ . Вне некоторого

диапазона  $|a| > \Delta$  или  $|b| > \Delta$ ,  $[V](a, b) = 0$ . Таким образом,  $[H](i, j)$  можно записать следующим образом:

$$[H]_{i,j} = u + \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} [V]_{a,b} [X]_{i+a,j+b} \quad (3)$$

свёрточные нейронные сети (CNN) являются специальным семейством нейронных сетей, содержащих свёрточные слои. В научном сообществе глубокого обучения  $V$  называется ядром свёртки, фильтром или просто весами слоя, которые часто являются обучаемыми параметрами. Когда область свёртки невелика, разница по сравнению с полносвязной сетью может быть огромной. В то время как раньше требовались миллиарды параметров, чтобы представить только один слой в сети обработки изображений, сейчас, как правило, нужно всего несколько сотен, без изменения размерности входных данных или скрытых представлений. Цена, уплаченная за это резкое сокращение параметров, заключается в том, что функции теперь являются инвариантными при обработке и что слой может включать только локальную информацию, когда определяет значение каждой скрытой активации. Все обучение зависит от применяемого индуктивного смещения [5]. Когда эта ошибка согласуется с реальностью, получаются эффективные модели выборки, которые хорошо обобщают невидимые данные. Однако, если эти сдвиги не согласуются с реальностью, например, если изображения окажутся не инвариантными при преобразовании, наши модели будут плохо соответствовать нашим тренировочным данным.

В случае, когда изображение содержит более одного канала (3 для обычных и более 3 для аэрокосмических изображений), свёрточный фильтр становится трехмерным или более. В случае, если канала 3, то тогда на входе имеется тензор третьего порядка, и становится достаточно удобным формировать скрытые представления также, как тензоры третьего порядка. То есть вместо того, чтобы иметь одно скрытое представление, соответствующее каждому пространственному местоположению, получается вектор скрытых представлений. Такое представление можно рассматривать как множество двумерных сеток, наложенных друг на друга. Эти сетки также называются картами признаков, так как каждая из них представляет собой пространственный набор усвоенных признаков для последующего слоя. Интуитивно можно представить, что на более низких слоях, которые ближе к входу, некоторые каналы могут стать специализированными для распознавания рёбер, в то время как другие могут распознавать текстуры. Для многоканального изображения скрытое представление вычисляется следующим образом [6]:

$$[H]_{i,j,d} = \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} \sum_c [V]_{a,b,c,d} [X]_{i+a,j+b,c} \quad (4)$$

где  $d$  индексирует выходные каналы в скрытом представлении  $H$ , а  $(a, b)$  - представляет собой координаты элементов фильтра. Последующий свёрточный слой будет также принимать тензор третьего порядка  $H$  в качестве входа. Таким образом

формируется формула сверточного слоя для произвольного количества каналов, где  $V$  является фильтром данного сверточного слоя.

Строго говоря, термин “свёртка” является не совсем корректным. Данная операция является по сути кросс-корреляцией, так как входной тензор и тензор фильтра объединяются для получения выходного тензора через операцию кросс-корреляции.

Рассмотрим случай, когда на входе двумерный тензор размером  $3 \times 3$ , и к нему применяется фильтр в виде двумерного тензора размером  $2 \times 2$ .

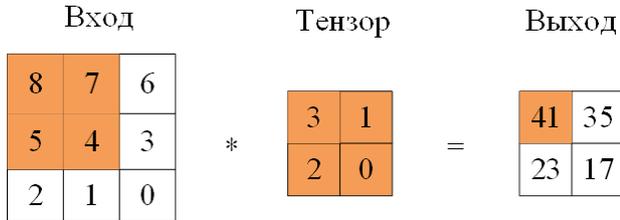


Рисунок 3 - Пример выполнения операции свертки (кросс-корреляции).

В двумерной операции перекрестной корреляции необходимо начинать с окна свёртки, расположенного в верхнем левом углу входного тензора, и перемещать его через входной тензор слева направо и сверху вниз. Когда окно свёртки сдвигается в определённое положение, входной субтензор, содержащийся в этом окне, и тензор фильтра умножаются поэлементно, а результирующий тензор суммируется. Этот результат даёт значение выходного тензора в соответствующем месте. В рассматриваемом случае выходной тензор имеет высоту 2 и ширину 2, а четыре элемента получают путем двумерной операции перекрестной корреляции:

$$\begin{aligned}
 8 \times 3 + 7 \times 1 + 5 \times 2 + 0 \times 4 &= 41, \\
 7 \times 3 + 6 \times 1 + 4 \times 2 + 3 \times 0 &= 35, \\
 5 \times 3 + 4 \times 1 + 2 \times 2 + 1 \times 0 &= 23, \\
 4 \times 3 + 3 \times 1 + 2 \times 2 + 0 \times 0 &= 19.
 \end{aligned}$$

Следует обратить внимание, что по каждой оси выходной размер немного меньше входного. Поскольку у фильтра ширина и высота больше единицы, возможно правильно вычислить перекрестную корреляцию только для тех мест, где фильтр полностью вписывается в изображение. Выходной размер определяется как входной размер  $n_h \times n_w$  минус размер свёрточного фильтра  $k_h \times k_w$ :

$$(n_h - k_h + 1) \times (n_w - k_w + 1) \quad (5)$$

Как было сказано выше, свёрточный слой содержит фильтры, которые выполняют операции свертки на входном изображении относительно его размеров. Количество фильтров может быть произвольным, но как правило его выбирают равным удвоенному количеству слоев входного изображения. Если на входе сети изображение имеет 3 канала (RGB) то к изображению применяется 6 фильтров.

Гиперпараметры фильтра включают размер, шаг свертки и дополнение изображения (padding).

Размер фильтра. Могут использоваться двумерные и

трехмерные фильтры. Рассмотрим сначала применение двумерных фильтров. В одном сверточном слое может применяться несколько фильтров. Тогда для каждого фильтра на выходе получится своё изображение. Например, если входное изображение имело размерность  $I \times I$ , и к нему было применено  $n$  фильтров  $F_x \times F_y$ , то выход будет иметь размерность:

$$n \times (I - F_x + 1) \times (I - F_y + 1).$$

Если количество каналов в изображении больше одного, то указанная выше размерность выхода будет получена для каждого слоя.

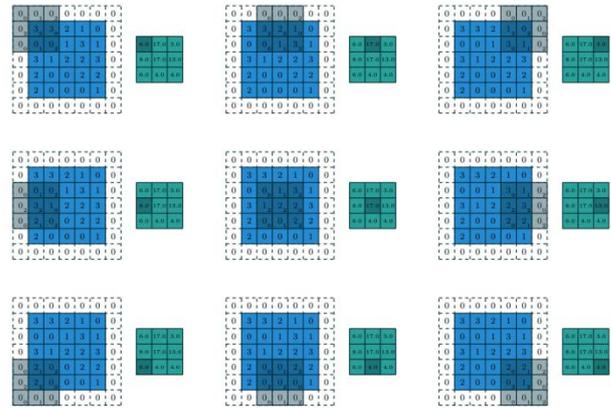


Рисунок 4 - Свёртка слоя изображения двумерным фильтром

Свёртка трехканального изображения с применением трехмерного фильтра происходит аналогично, но в этом случае скалярное произведение считается еще и по всем каналам изображения. Трехмерный фильтр может быть использован для усреднения информации о цветах исходного изображения в первом сверточном слое. Если размер фильтра будет равен размеру изображения, то на выходе слоя будет одно изображение вместо трех. В примере, показанном на рисунке 5, к входным данным размера  $6 \times 6 \times 3$  применяется фильтр размером  $3 \times 3 \times 3$ , что при шаге свертки равном 1 дает на выходе матрицу размером  $4 \times 4$  (рисунок 5).

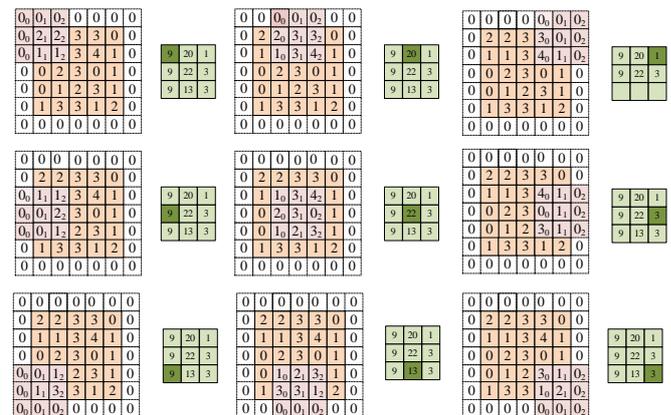


Рисунок 5 - Свёртка слоя изображения двумерным фильтром

Свёртка трехканального изображения с применением трехмерного фильтра происходит аналогично, но в этом случае скалярное произведение считается еще и по всем каналам изображения. Трехмерный фильтр может быть использован для усреднения информации о цветах

исходного изображения в первом сверточном слое. Если размер фильтра будет равен размеру изображения, то на выходе слоя будет одно изображение вместо трех. В примере, показанном на рисунке 5, к входным данным размера  $6 \times 6 \times 3$  применяется фильтр размером  $3 \times 3 \times 3$ , что при шаге свертки равном 1 дает на выходе матрицу размером  $4 \times 4$  (рисунок 6).

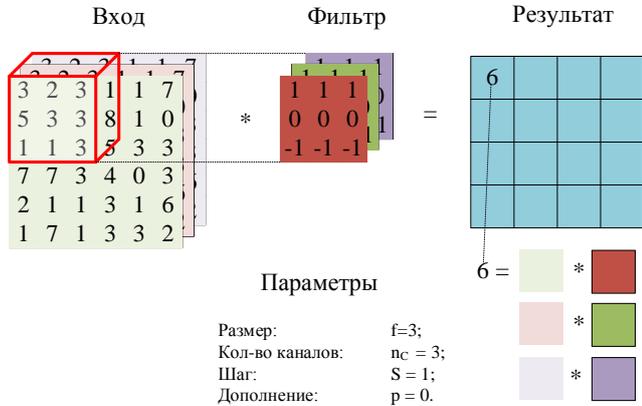


Рисунок 6 - Свертка трехканального изображения трехмерным фильтром

Шаг свертки. Это число пикселей, на которые перемещается фильтр после каждой операции свертки. На рисунке 7 показана работа фильтра с шагом свертки 3.

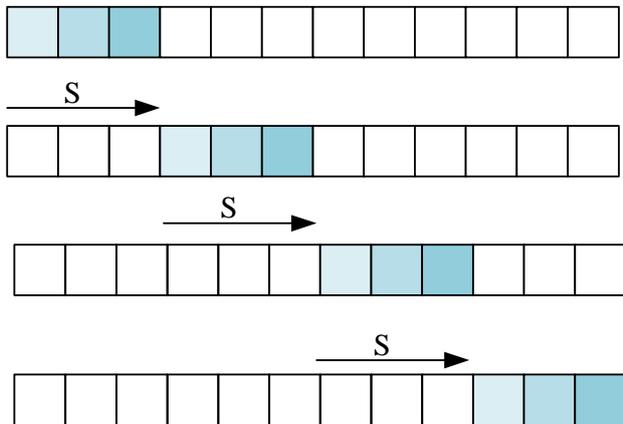


Рисунок 7 - Шаг свертки

Дополнение изображения. Применение операции свертки уменьшает изображение. Поэтому пиксели, находящиеся на границе изображения участвуют в меньшем количестве сверток, чем внутренние. Для снижения влияния этого фактора в сверточных слоях используется дополнение изображения. Это процесс добавления пикселей по границе входного изображения. Дополнение может устанавливаться вручную или автоматически в одном из трех режимов: valid, same и full. В режиме "valid" пиксели по границе изображения не добавляются (рисунок 8). В этом режиме последняя свертка отбрасывается, если измерение не соответствует измерению изображения.

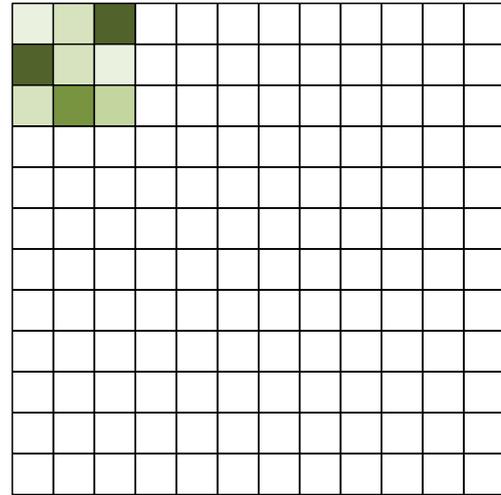


Рисунок 8 - Свертка без дополнения (режим дополнения "valid")

В режиме "same" выходные изображения предыдущего слоя дополняются пикселями так, чтобы после свертки сохранился размер изображения. Необходимое количество пикселей рассчитывается по формулам:

$$P_{start} = \left\lfloor \frac{S \lfloor \frac{I}{S} \rfloor - I + F - S}{2} \right\rfloor \quad (6)$$

$$P_{end} = \left\lceil \frac{S \lfloor \frac{I}{S} \rfloor - I + F - S}{2} \right\rceil \quad (7)$$

В этом случае дополнение обеспечивает размер карты признаков  $\lfloor \frac{I}{S} \rfloor$ .

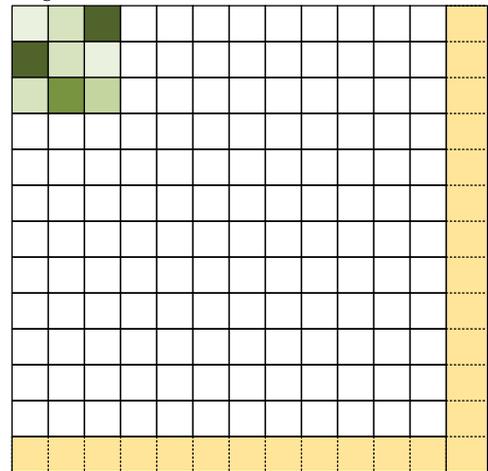


Рисунок 9 - Свертка с дополнением в режиме "same"

Для обеспечения полного сохранения изображения используется дополнение в режиме "full". В этом режиме изображение дополняется со всех сторон, как показано на рисунке 10. Для расчета используются следующие формулы:

$$P_{start} \in [0, F - 1] \quad (8)$$

$$P_{end} = F - 1 \quad (9)$$

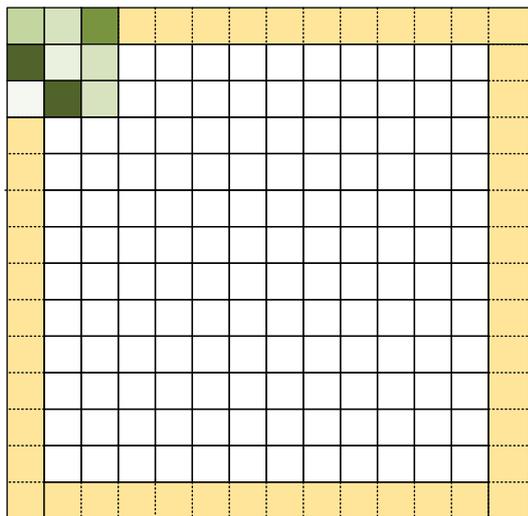


Рисунок 10 - Свертка с дополнением в режиме “full”

Заполнение пикселей может осуществляться как нулями 00[ABC]00 (zero shift), так и значащими цифрами: AA[ABC]CC (border extension); BA[ABC]CB (mirror shift); BC[ABC]AB (cyclic shift).

Размер изображения  $O$ , получаемого после применения фильтра размером  $F$ , при дополнении  $P$ , шаге  $S$ , и входном размере изображения  $I$ , определяется по формуле:

$$O = \frac{I - F + P_{start} + P_{end}}{S} + 1 \quad (10)$$

На рисунке 11 графически показаны описанные выше параметры:

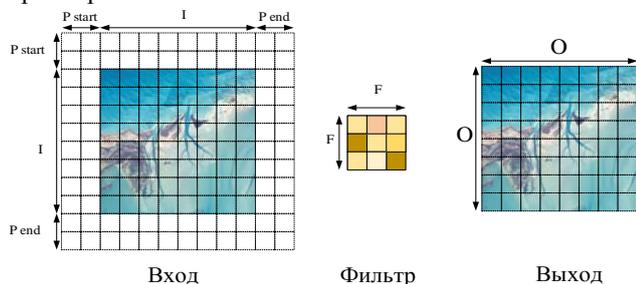


Рисунок 11 - Основные параметры сверточного слоя

В одном сверточном слое как правило применяется не более одного параметра сдвига (bias) на фильтр. Шаг свертки в большинстве случаев меньше размера фильтра. Общее число настраиваемых параметров для слоя свертки определяется по формуле:

$$N_{params} = (F \times F \times C + 1) \cdot K, \quad (11)$$

где  $K$  - количество фильтров применяемых сверточном слое,  $C$  - количество слоев в изображении.

Ниже рассмотрены наиболее известные фильтры, применение которых к изображению дает определенный результат.

Фильтры градиента (Таблица 1) могут использоваться для обнаружения ребер с шагом в 45 градусов:

Таблица 1 - Фильтры градиента для обнаружения ребер с шагом в 45 градусов

Градиент Восток (фильтр 3 x 3)	1 0 -1 2 0 -2 1 0 -1
Градиент Север (фильтр 3 x 3)	-1 -2 -1 0 0 0 1 2 1
Градиент Северо-восток (фильтр 3 x 3)	0 -1 -2 1 0 -1 2 1 0
Градиент Северо-запад (фильтр 3 x 3)	-2 -1 0 -1 0 1 0 1 2
Градиент Юг (фильтр 3 x 3)	1 2 1 0 0 0 -1 -2 -1
Градиент Запад (фильтр 3 x 3)	-1 0 1 -2 0 2 -1 0 1

Фильтры Лапласа (Таблица 2) часто используются для выявления ребер. Он часто применяется к изображению, которое сначала было сглажено, для уменьшения его чувствительности к шуму:

Таблица 2 - Фильтры Лапласа для выявления ребер

Лапласиан 3x3	0 -1 0 -1 4 -1 0 -1 0
Лапласиан 5x5	0 0 -1 0 0 0 -1 -2 -1 0 -1 -2 17 -2 -1 0 -1 -2 -1 0 0 0 -1 0 0

Фильтры градиента, могут использоваться для обнаружения ребер. Для получения лучших результатов, необходимо применить алгоритм сглаживания перед применением алгоритма обнаружения ребер (Таблица 3):

Таблица 3 - Фильтры градиента для обнаружения ребер

Обнаружение линий По горизонтали	-1 -1 -1 2 2 2 -1 -1 -1
Обнаружение линий По левой диагонали	2 -1 -1 -1 2 -1 -1 -1 2
Обнаружение линий По правой диагонали	-1 -1 2 -1 2 -1

	2 -1 -1
Обнаружение линий По вертикали	-1 0 -1 -1 2 -1 -1 2 -1

Также, для обнаружения ребер хорошо подходит фильтр Собеля (Таблица 4):

Таблица 4 - Фильтры Собеля для обнаружения ребер

Sobel По горизонтали	-1 -2 -1 0 0 0 1 2 1
Sobel По вертикали	-1 0 1 -2 0 2 -1 0 1

Фильтр увеличения резкости (высокочастотный) подчеркивает сравнительное различие в значениях с соседями. Высокочастотный фильтр вычисляет фокальную суммарную статистику для каждой ячейки входных данных с помощью взвешенной окрестности фильтра. Это позволяет выявить границы между пространственными объектами (например, границу между водным массивом и лесом), таким образом, подчеркивая ребра между объектами. Высокочастотный фильтр называется фильтром улучшения границ (ребер, краев). Фильтр определяет, какие ячейки нужно использовать в окрестности, и то, насколько их взвешивать (перемножая их) (Таблица 5):

Таблица 5 - Высокочастотный фильтр (фильтр улучшения границ (ребер, краев))

Острый	0 -0.25 0 -0.25 2 -0.25 0 -0.25 0
Острый II	-0.25 -0.25 -0.25 -0.25 3 -0.25 -0.25 -0.25 -0.25
Слияние 3x3	-1 -1 -1 -1 9 -1 -1 -1 -1
Слияние 5x5	-1 -3 -4 -3 -1 -3 0 6 0 -3 -4 6 21 6 -4 -3 0 6 0 -3 -1 -3 -4 -3 -1

Фильтры сглаживания (низкочастотные) сглаживают данные, уменьшая локальное изменение и удаляя шум. Низкочастотный фильтр вычисляет среднее значение для каждой окрестности. Эффект в том, что большие и маленькие значения в каждой окрестности будут усредняться, что уменьшит экстремальные значения данных (Таблица 6):

Таблица 6 - Низкочастотный фильтр

Сгладить Среднее арифметическое	0.111 0.111 0.111 0.111 0.111 0.111 0.111 0.111 0.111
Сглаживание 3x3	1 2 1 2 4 2 1 2 1
Сглаживание 5x5	1 1 1 1 1 1 4 4 4 1 1 4 12 4 1 1 4 4 4 1 1 1 1 1 1

#### IV. ЗАКЛЮЧЕНИЕ

Таким образом, можно сделать вывод, что сверточные нейронные сети являются вариацией многослойных перцептронов. Нейроны в этих сетях имеют общий вес, в отличие от перцептронов, где каждый нейрон имеет отдельный вектор весов. Наличие общих весов приводит к сокращению общего числа обучаемых весов, в результате чего возникает разреженность [7, 8].

Используя стратегию общих весов, нейроны могут выполнять свертки на данных с фильтром свертки, содержащим эти веса. За этим следует операция субдискретизации, которая в форме нелинейной дискретизации постепенно уменьшает пространственный размер представления, тем самым уменьшая количество вычислений и параметров в сети.

Существующий между сверточным слоем и слоем субдискретизации слой функции активации изменяет значение элемента в зависимости от его текущего значения. После нескольких сверточных и субдискретизационных слоев размер изображения (размер карты признаков) уменьшается, и из него извлекаются более сложные признаки. В результате, когда карта признаков становится достаточно маленькой, ее содержимое преобразуется в одномерный вектор, который подается на вход полносвязной нейронной сети. Последний слой этой сети рассматривается как выходной для всей CNN.

Работа выполнена при финансовой поддержке Фонда содействия инновациям (ФСИ) в рамках Договора 94C2/МОЛ/73887.

#### БИБЛИОГРАФИЯ

- [1] SATYAJIT MAITRA . ML Strategy for Machine Learning Projects. <https://medium.com/@ssatyajitmaitra/ml-strategy-for-machine-learning-projects-7b674e3bbb9>, Jul 8, 2019.
- [2] Naveen Mathew Nathan S. CNN vs fully-connected network for image processing, Towards Data Science <https://towardsdatascience.com/cnn-vs-fully-connected-network-for-image-processing-8c5b95d4e42f> .Sep 17, 2019.
- [3] Zhang, A., Tay, Y., Zhang, S., Chan, A., Luu, A. T., Hui, S. C., & Fu, J. (2021). Beyond fully-connected

layers with quaternions: parameterization of hypercomplex multiplications with  $1/n$  parameters. International Conference on Learning Representations.

- [4] В. О. Скрипачев, М. В. Гуйда, Н. В. Гуйда, and А. О. Жуков. Исследование сверточных нейронных сетей для обнаружения объектов на аэрокосмических снимках. International Journal of Open Information Technologies, 10(7):54–64, 2022.
- [5] Индуктивное смещение - Inductive bias Википедия site: wiki5.ru. [https://livepcwiki.ru/wiki/Inductive\\_bias](https://livepcwiki.ru/wiki/Inductive_bias).
- [6] Convolutional Neural Networks cheatsheet <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks>.
- [7] Aston Zhang, Zack C. Lipton, Mu Li, Alex J. Smola, Brent Werness, Rachel Hu, Shuai Zhang, Yi Tay, Anirudh Dagar, Yuan Tang. Dive into Deep Learning. <https://d2l.ai/index.html>.
- [8] Backpropagation In Convolutional Neural Networks Jefkine, 5 September 2016. <https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/>

# Features of convolutional neural networks

Vladimir Skripachev, Mikhail Guida, Nikolay Guida, Alexander Zhukov

**Abstract.** The article discusses in detail the use of collapsible neural networks (CNNs) for use in solving problems of detecting objects in aerospace images. The structure of collapsing neural networks is revealed, taking into account the specifics of the application. Bearing in mind the creative essence of the problem of object recognition in aerospace images, the high variability of the solution of the task is illustrated. Approaches to the construction of collapsing neural networks are considered, taking into account the key factors of the effectiveness of solving the problem of detecting objects in aerospace images.

Special attention is paid to the internal structure of the construction of collapsing neural networks. The execution of cross-correlation (convolution) operations in networks is described. The mechanism of the shift (step) of the convolution is clearly shown. The application of two-dimensional filters for single-channel and three-channel images is described. The application of image augmentation modes during convolution is analyzed: without addition ("valid" mode), with addition ("same" and "full" modes).

The most commonly used filters in CNNs for determining characteristic lines (horizontal, vertical, inclined), filters for improving boundaries (edges, edges), filters smoothing extreme data values are considered.

A general conclusion is made about the use of convolutional neural networks and their application strategy, indicating the most acceptable CNNs design for detecting objects in aerospace images.

**Keywords:** convolving neural networks (CNNs), multilayer perceptrons (MLP), convolution of a three-channel image using a three-dimensional filter, convolution of an image layer with a two-dimensional filter, convolution with an addition in the "full" mode, the main parameters of the convolution layer, performing a convolution operation (cross-correlation), convolution with an addition in the "same" mode, filters gradient for edge detection, Sobel filters for edge detection, High-frequency filter (filter for improving boundaries (edges, edges), low-frequency filter, gradient filters for edge detection in 45 degree increments, Laplace filters for edge detection.

## REFERENCES

- [1] SATYAJIT MAITRA . ML Strategy for Machine Learning Projects. <https://medium.com/@ssatyajitmaitra/ml-strategy-for-machine-learning-projects-7b674e3bbb9>, Jul 8, 2019.
- [2] Naveen Mathew Nathan S. CNN vs fully-connected network for image processing, Towards Data Science <https://towardsdatascience.com/cnn-vs-fully-connected-network-for-image-processing-8c5b95d4e42f> .Sep 17, 2019.
- [3] Zhang, A., Tay, Y., Zhang, S., Chan, A., Luu, A. T., Hui, S. C., & Fu, J. (2021). Beyond fully-connected layers with quaternions: parameterization of hypercomplex multiplications with 1/n parameters. International Conference on Learning Representations.
- [4] V. O. Skripachev, M. V. Guyda, N. V. Guyda, and A. O. Zhukov. Issledovanie svertochnykh neyronnykh setey dlya obnaruzheniya ob"ektov na aerokosmicheskikh snimkakh. International Journal of Open Information Technologies, 10(7):54–64, 2022.
- [5] Convolutional Neural Networks cheatsheet <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks>.
- [6] Inductive bias [site:wiki5.ru. https://livepcwiki.ru/wiki/Inductive\\_bias](https://livepcwiki.ru/wiki/Inductive_bias).
- [7] Aston Zhang, Zack C. Lipton, Mu Li, Alex J. Smola, Brent Werness, Rachel Hu, Shuai Zhang, Yi Tay, Anirudh Dagar, Yuan Tang. Dive into Deep Learning. <https://d21.ai/index.html>.
- [8] Backpropagation In Convolutional Neural Networks Jefkine, 5 September 2016. <https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/>