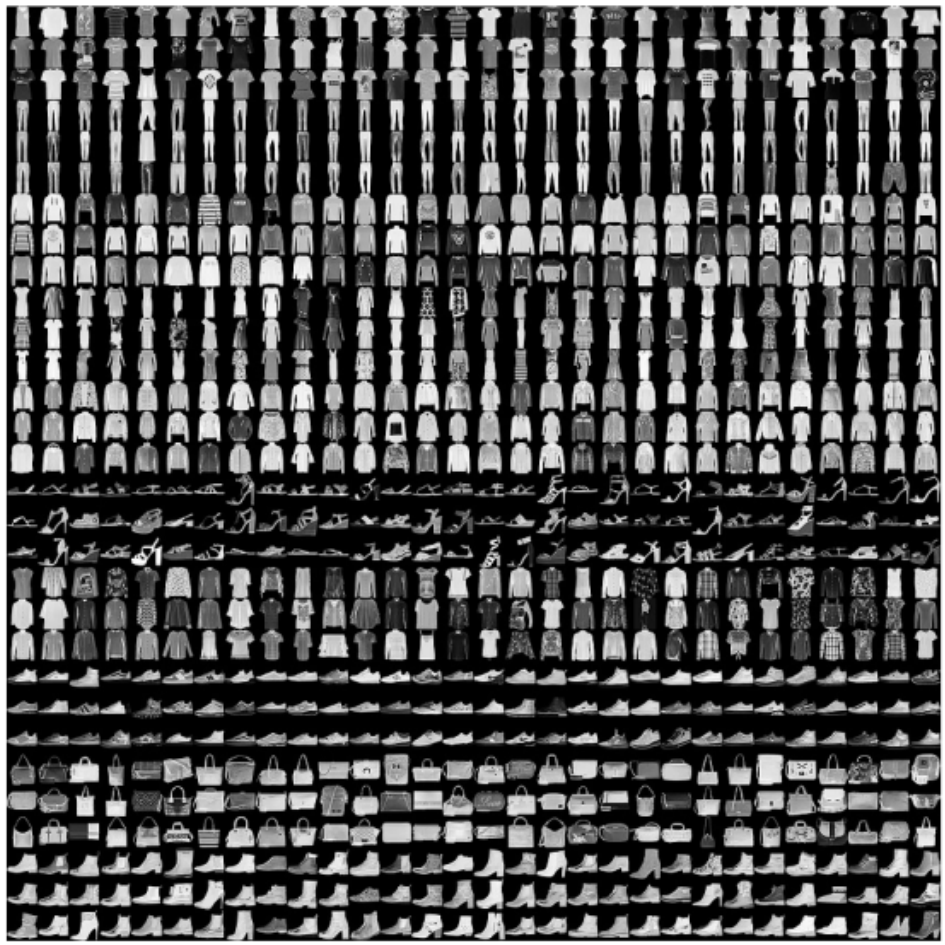
Глубокое обучение для компьютерного зрения – Сверточные нейронные сети и трансферное обучение

13 ноября 2026 г.

Fashion MNIST

Ранее мы видели, что нейронная сеть с одним скрытым слоем может достичь точности 85% и более на этом наборе данных.

Как мы можем добиться лучших результатов?



Sample of the Fashion MNIST images by Yuzamei.
Source: Wikimedia Commons. License: CC BY-SA.

Обработка изображений «естественным» способом

Когда мы преобразуем матрицу изображения в длинный вектор и передаем его в полносвязный слой, происходит несколько «нежелательных» вещей:

Обработка изображений «естественным» способом

Когда мы преобразуем матрицу изображения в длинный вектор и передаем его в полносвязный слой, происходит несколько «нежелательных» вещей:

- Нам нужно изучить «слишком много» параметров.

Обработка изображений «естественным» способом

Когда мы преобразуем матрицу изображения в длинный вектор и передаем его в полносвязный слой, происходит несколько «нежелательных» вещей:

- Нам нужно изучить «слишком много» параметров.
  - Сведение цветного изображения размером 3024×3024 пикселей (с вашего телефона) к одному плотному слою из 100 нейронов генерирует приблизительно ? параметров.

Обработка изображений «естественным» способом

Когда мы преобразуем матрицу изображения в длинный вектор и передаем его в полносвязный слой, происходит несколько «нежелательных» вещей:

- Нам нужно изучить «слишком много» параметров.
  - Сведение цветного изображения размером 3024×3024 пикселей (с вашего телефона) к одному плотному слою из 100 нейронов генерирует приблизительно 2.7 миллиарда параметров.

Обработка изображений «естественным» способом

Когда мы преобразуем матрицу изображения в длинный вектор и передаем его в полносвязный слой, происходит несколько «нежелательных» вещей:

- Нам нужно изучить «слишком много» параметров.
  - Сведение цветного изображения размером 3024×3024 пикселей (с вашего телефона) к одному плотному слою из 100 нейронов генерирует приблизительно 2.7 миллиарда параметров.
  - Это ресурсоёмкий с точки зрения вычислений, требующий большого объёма данных и повышающий риск переобучения.

Обработка изображений «естественным» способом

Когда мы преобразуем матрицу изображения в длинный вектор и передаем его в полносвязный слой, происходит несколько «нежелательных» вещей:

- Нам нужно изучить «слишком много» параметров.

Обработка изображений «естественным» способом

Когда мы преобразуем матрицу изображения в длинный вектор и передаем его в полносвязный слой, происходит несколько «нежелательных» вещей:

- Нам нужно изучить «слишком много» параметров.
- Мы теряем локальные пространственные связи между пикселями, определяющими особенности изображения.

Обработка изображений «естественным» способом

Когда мы преобразуем матрицу изображения в длинный вектор и передаем его в полносвязный слой, происходит несколько «нежелательных» вещей:

- Нам нужно изучить «слишком много» параметров.
- Мы теряем локальные пространственные связи между пикселями, определяющими особенности изображения.
- Мы не учимся один раз и не используем это снова и снова.

Обработка изображений «естественным» способом

Когда мы преобразуем матрицу изображения в длинный вектор и передаем его в полносвязный слой, происходит несколько «нежелательных» вещей:

- Нам нужно изучить «слишком много» параметров.
- Мы теряем локальные пространственные связи между пикселями, определяющими особенности изображения.
- Мы не учимся один раз и не используем это снова и снова.
  - Если какой-либо элемент изображения (например, вертикальная линия или круг) появляется в разных местах изображения, нейронная сеть должна “изучать его один раз и использовать снова и снова”, а не изучать его отдельно каждый раз.

Обработка изображений «естественным» способом

- Когда мы преобразуем матрицу изображения в длинный вектор и передаем его в полносвязный слой, происходит несколько «нежелательных» вещей:
  - Нам нужно изучить «слишком много» параметров.
  - Мы теряем локальные пространственные связи между пикселями, определяющими особенности изображения.
  - Мы не учимся один раз и не используем это снова и снова.

Обработка изображений «естественным» способом

- Когда мы преобразуем матрицу изображения в длинный вектор и передаем его в полносвязный слой, происходит несколько «нежелательных» вещей:
  - Нам нужно изучить «слишком много» параметров.
  - Мы теряем локальные пространственные связи между пикселями, определяющими особенности изображения.
  - Мы не учимся один раз и не используем это снова и снова.
- Для устранения этих недостатков были разработаны сверточные слои.

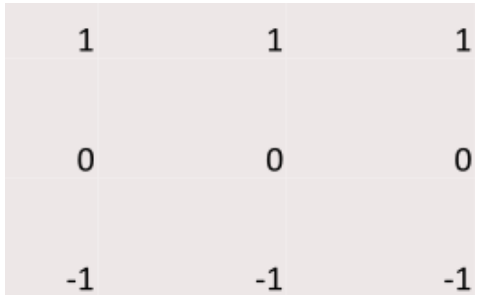
Сверточные слои и фильтры

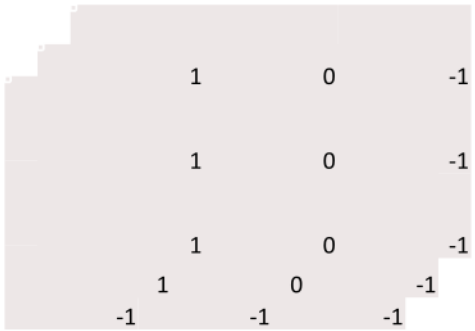
- Сверточный фильтр — это небольшая квадратная матрица чисел.

1	1	1
0	0	0
-1	-1	-1

Сверточные слои и фильтры

- Сверточный фильтр — это небольшая квадратная матрица чисел.
- Сверточный слой состоит из одного или нескольких сверточных фильтров.





Сверточные слои и фильтры

Тщательно выбирая значения фильтра и «применяя» его к изображению, можно обнаружить различные особенности изображения (как мы вскоре продемонстрируем).

Сверточные слои и фильтры

Тщательно выбирая значения фильтра и «применяя» его к изображению, можно обнаружить различные особенности изображения (как мы вскоре продемонстрируем).

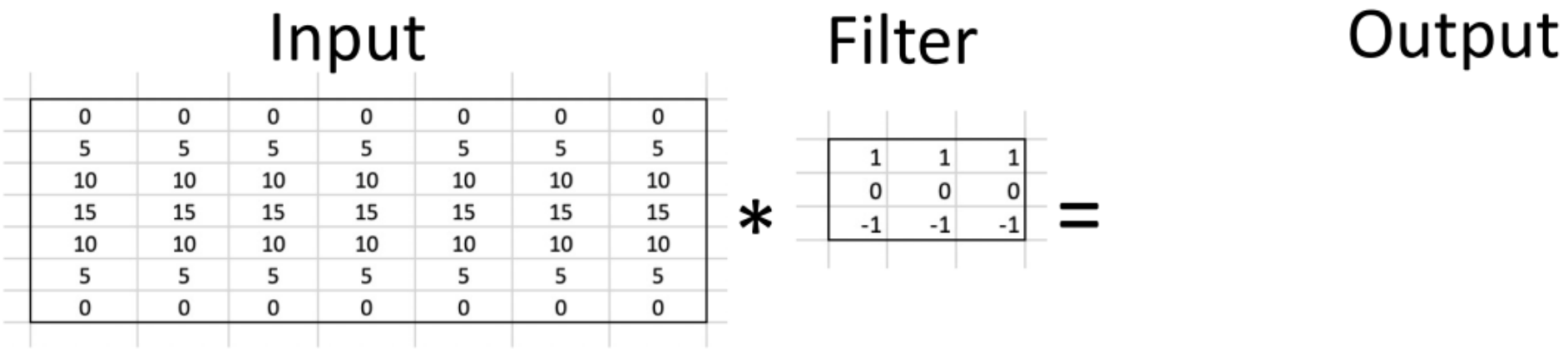
Этот фильтр способен обнаруживать горизонтальные линии!

1	1	1
0	0	0
-1	-1	-1

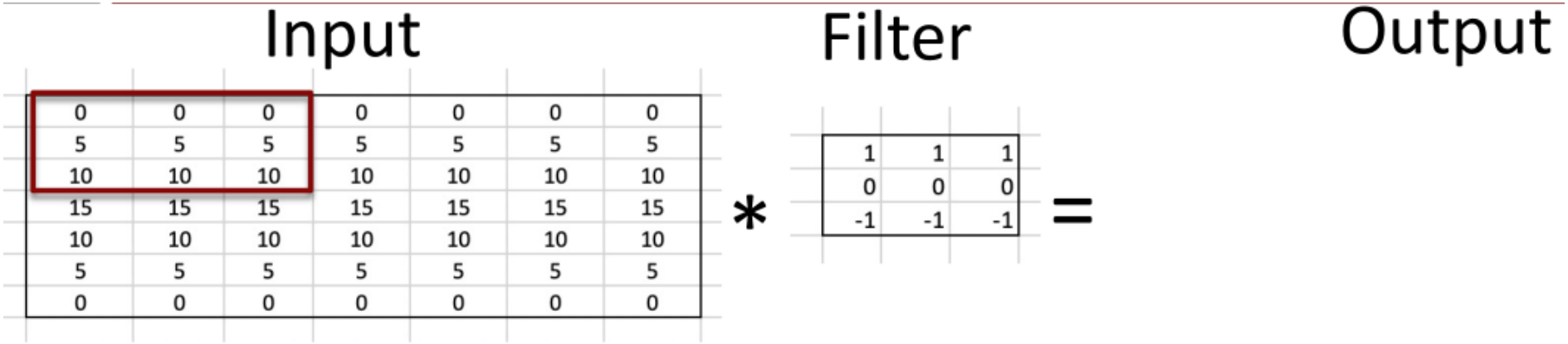
Этот фильтр способен обнаруживать вертикальные линии!

1	0	-1
1	0	-1
1	0	-1

Применение сверточного фильтра к изображению называется «операцией свертки».



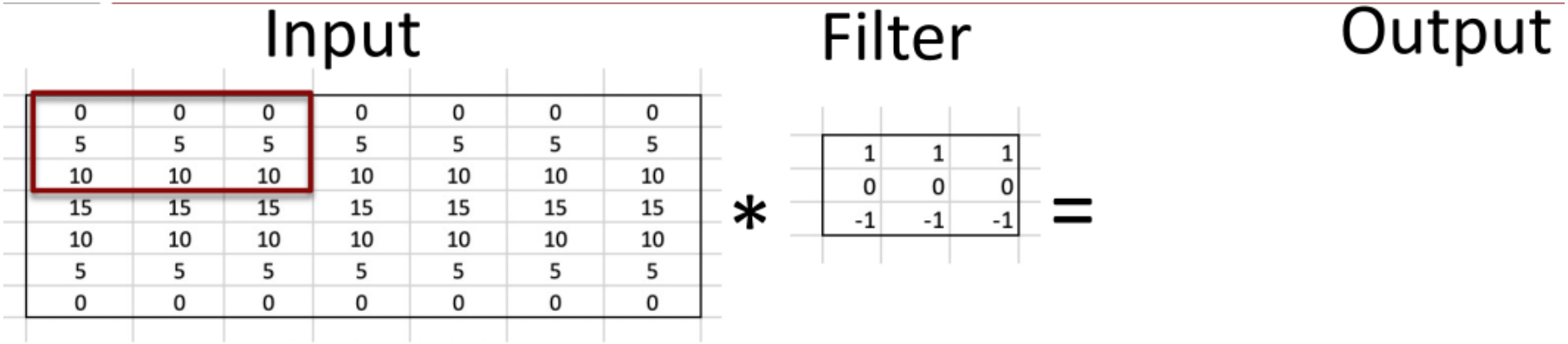
Применение сверточного фильтра к изображению называется «операцией свертки».



Операция свертки

- Наложение фильтра в верхний левый угол изображения.

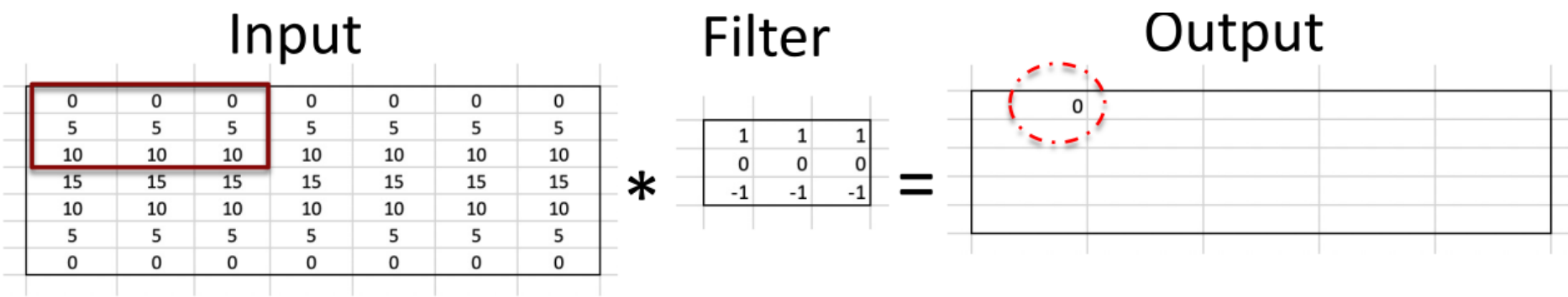
Применение сверточного фильтра к изображению называется «операцией свертки».



Операция свертки

- Наложение фильтра в верхний левый угол изображения.
- Умножьте совпадающие элементы и сложите их:
  - $0*1+0*1+0*1+5*0+5*0+5*0+10*-1+10*-1+10*-1 = -30$

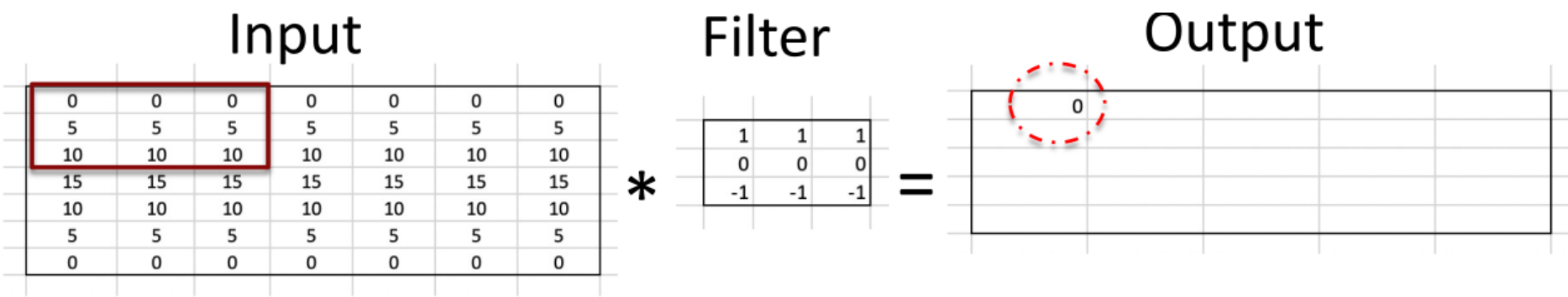
Применение сверточного фильтра к изображению называется «операцией свертки».



Операция свертки

- Наложение фильтра в верхний левый угол изображения.
- Умножьте совпадающие элементы и сложите их:
  - $0*1+0*1+0*1+5*0+5*0+5*0+10*-1+10*-1+10*-1=-30$

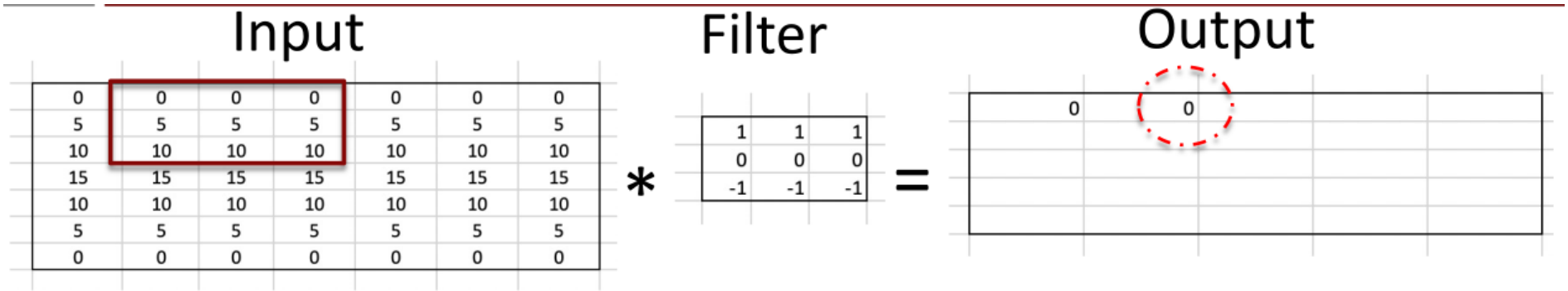
Применение сверточного фильтра к изображению называется «операцией свертки».



Операция свертки

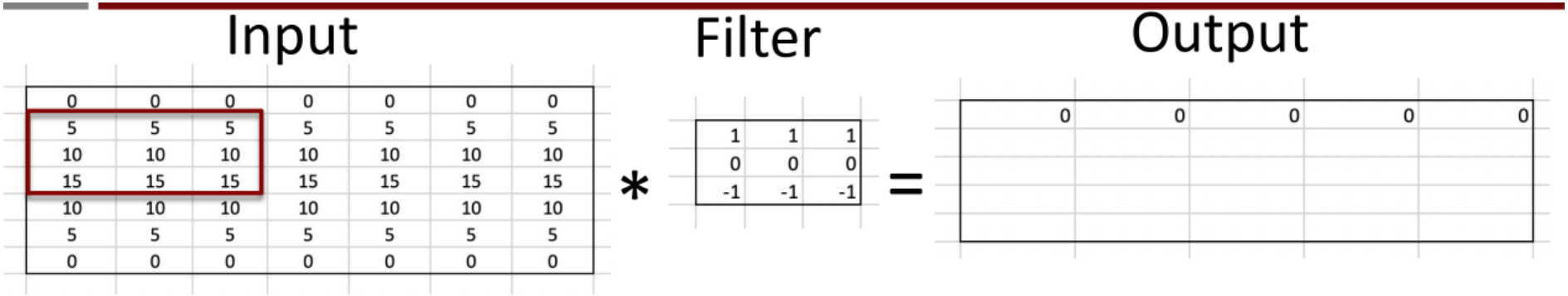
- Наложение фильтра в верхний левый угол изображения.
- Умножьте совпадающие элементы и сложите их:
  - $0*1+0*1+0*1+5*0+5*0+5*0+10*-1+10*-1+10*-1 = -30$
- Применть ReLU*: $max(0, -30) = 0$. Это число находится в верхнем левом углу выходных данных.

Операция свертки



Сдвиньте окно на один шаг вправо и повторите этот процесс, чтобы получить второе число на выходе.

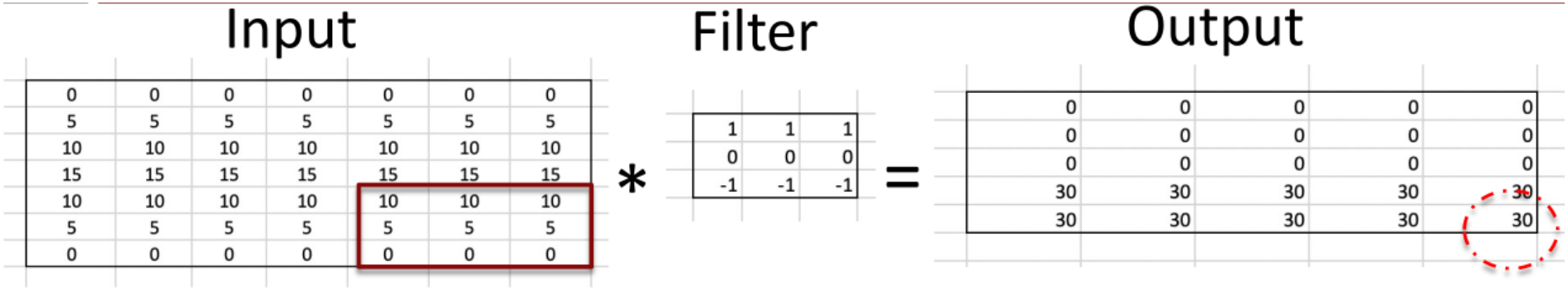
Операция свертки



Сдвиньте окно на один шаг вправо и повторите этот процесс, чтобы получить второе число на выходе.

Закончив первый ряд, перейдите к началу второго ряда и продолжайте, как и раньше...

Операция свертки



Сдвиньте окно на один шаг вправо и повторите этот процесс, чтобы получить второе число на выходе.

Закончив первый ряд, перейдите к началу второго ряда и продолжайте, как и раньше...

... до тех пор, пока не дойдете до нижнего правого угла

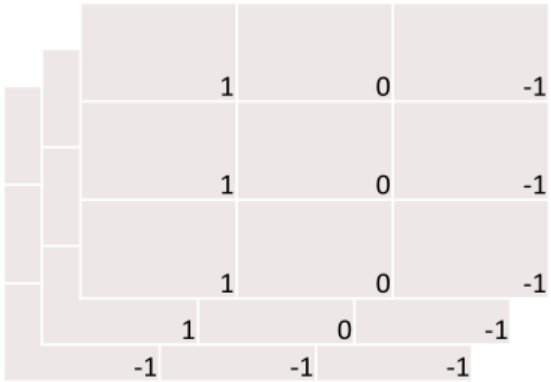
Тщательно подобрав значения в фильтре и применив операцию свертки, можно обнаружить различные особенности изображения.

Посмотрите HODL-Lec-3-Convolution-Example.xlsx

Опционально:
Посмотрите https://setosa.io/ev/image-kernels/, чтобы попрактиковаться с различными фильтрами.

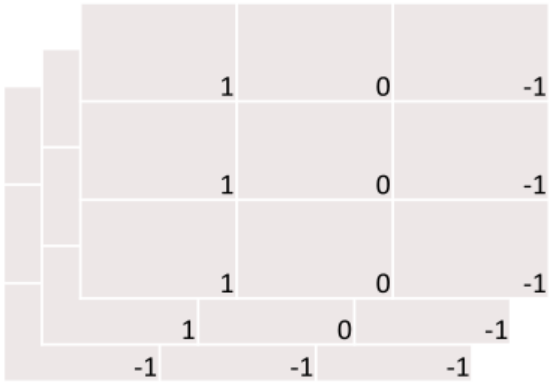
Сверточные слои

- Сверточный слой состоит из одного или нескольких сверточных фильтров.



Сверточные слои

- Сверточный слой состоит из одного или нескольких сверточных фильтров.



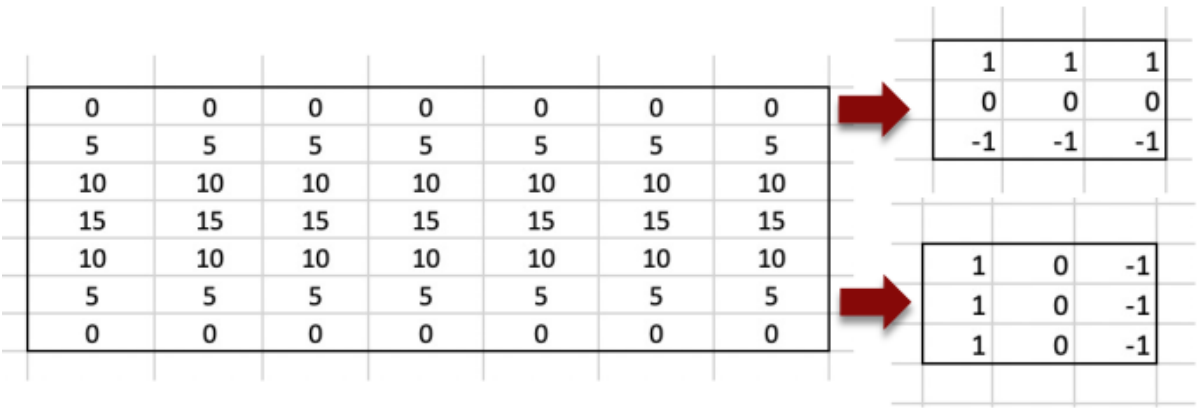
- Каждый фильтр можно рассматривать как специализированный инструмент для обнаружения определенного признака (например, горизонтальной линии, дуги, вертикальной линии).

Применение сверточного слоя к изображению

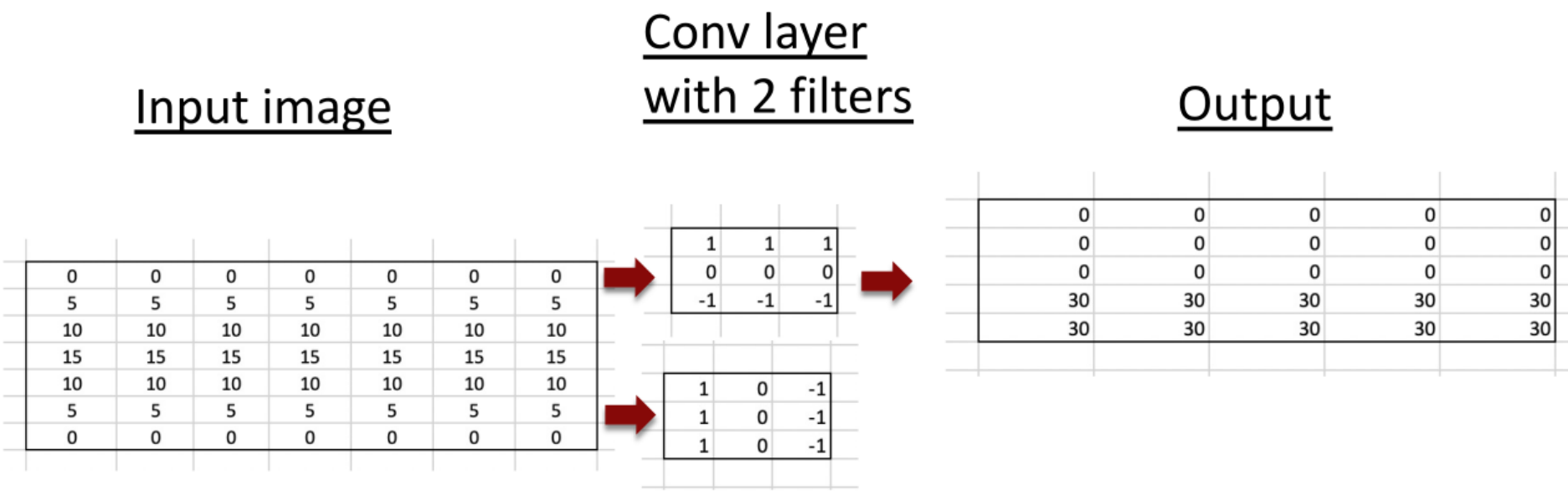
Input image

Conv layer with 2 filters

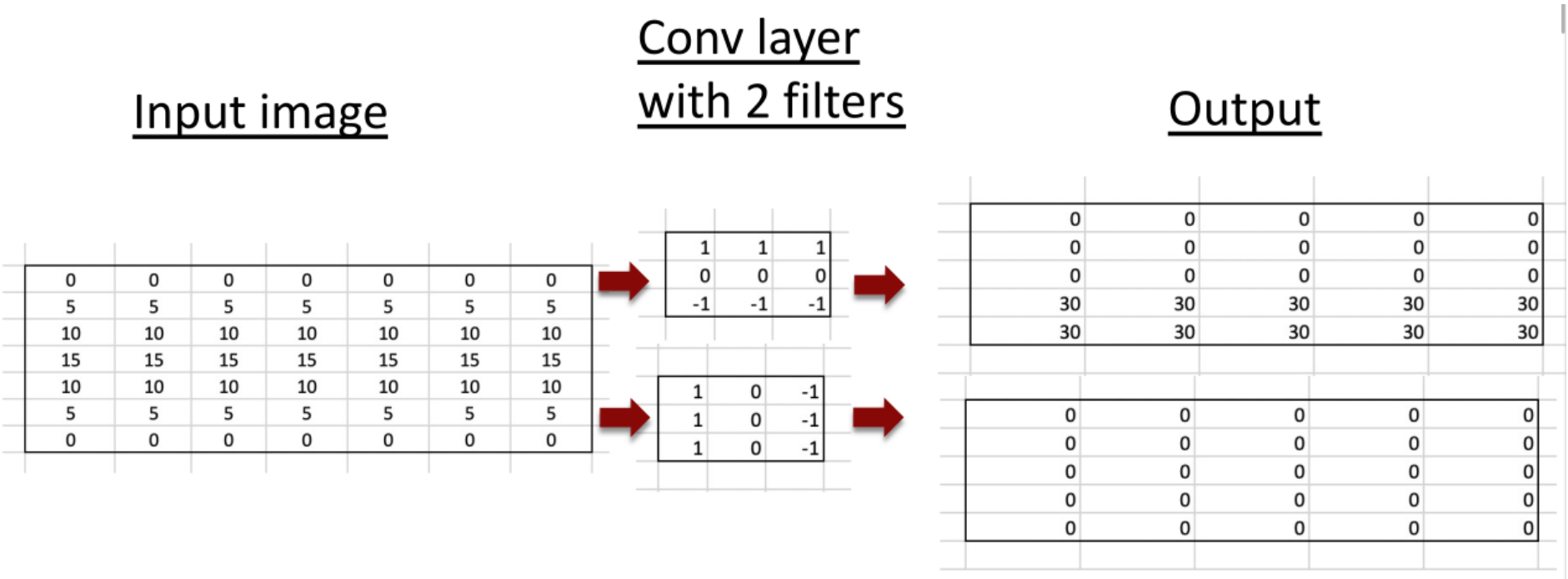
Output



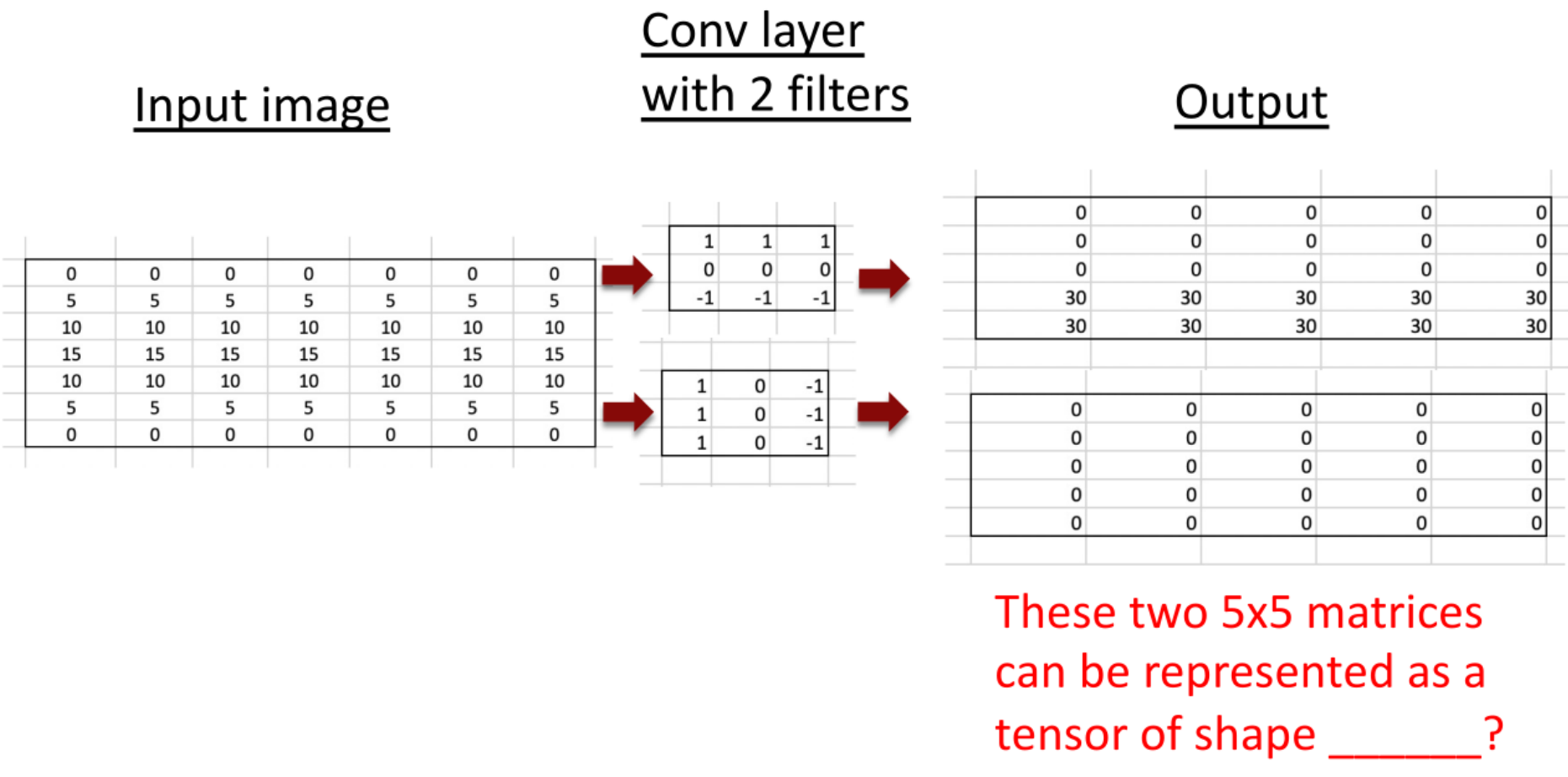
Применение сверточного слоя к изображению



Применение сверточного слоя к изображению



Применение сверточного слоя к изображению



Применение сверточного слоя к изображению

Input image

0	0	0	0	0	0	0
5	5	5	5	5	5	5
10	10	10	10	10	10	10
15	15	15	15	15	15	15
10	10	10	10	10	10	10
5	5	5	5	5	5	5
0	0	0	0	0	0	0

Conv layer with 2 filters

1	1	1
0	0	0
-1	-1	-1

1	0	-1
1	0	-1
1	0	-1

Output

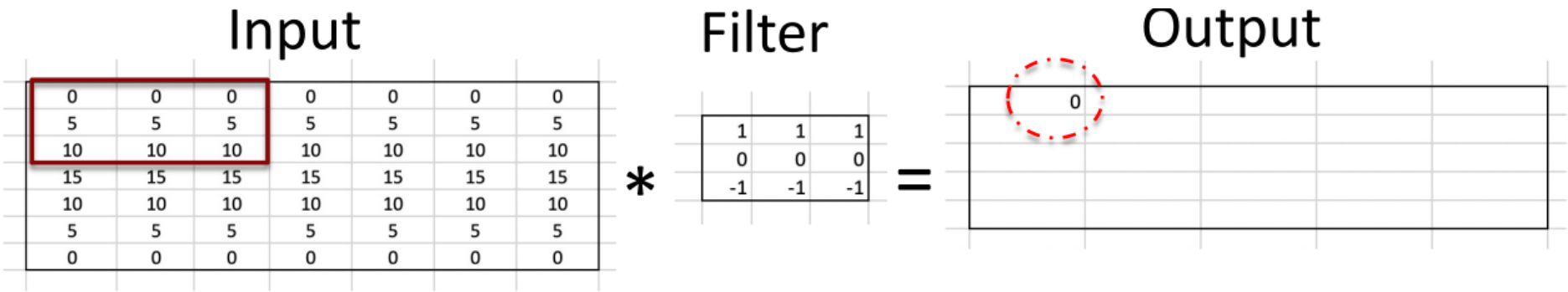
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
30	30	30	30	30
30	30	30	30	30

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

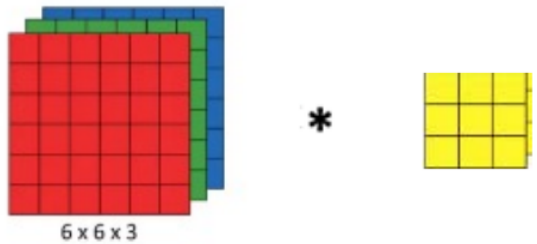
These two 5x5 matrices can be represented as a tensor of shape 5 x 5 x 2 or 2 x 5 x 5

Применение сверточного фильтра к цветному изображению

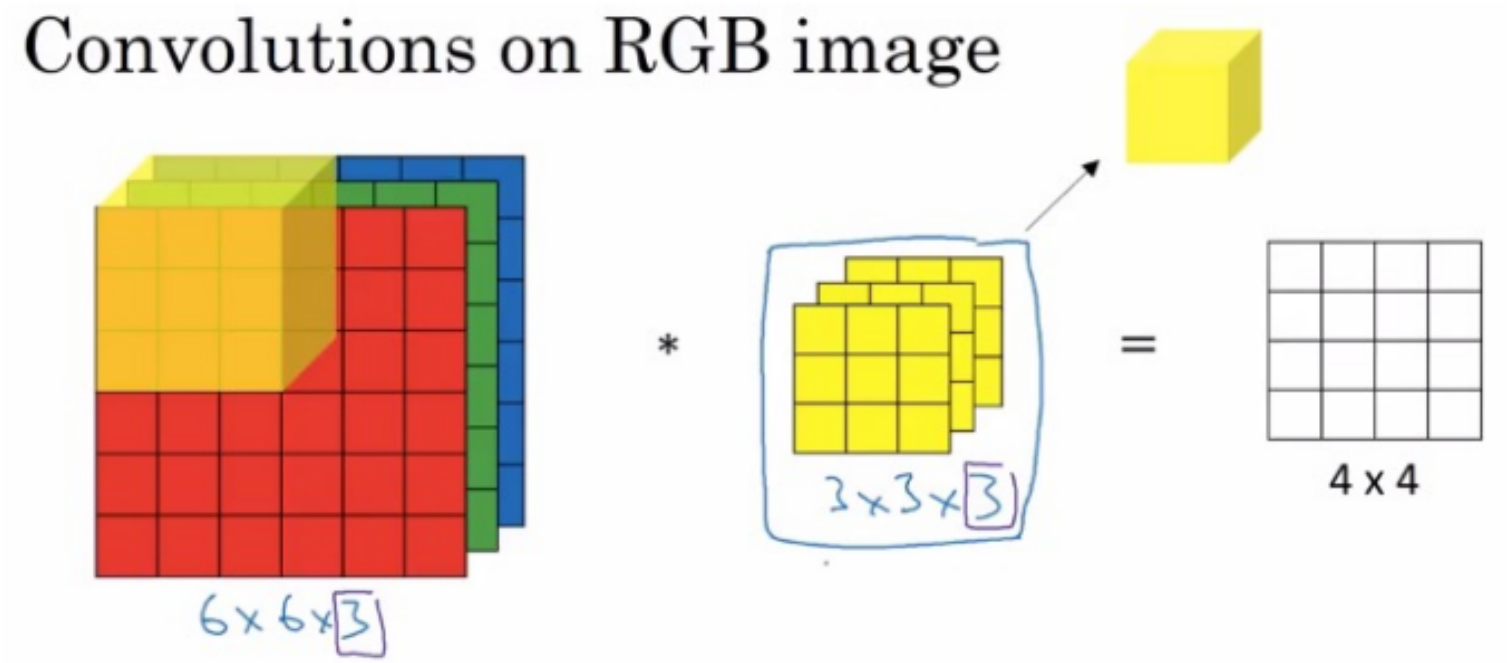
Мы знаем, как применить сверточный фильтр к двумерному тензору (например, к изображению в оттенках серого).



Как следует применять сверточный фильтр к тензору 3-го ранга (например, к цветному изображению)?

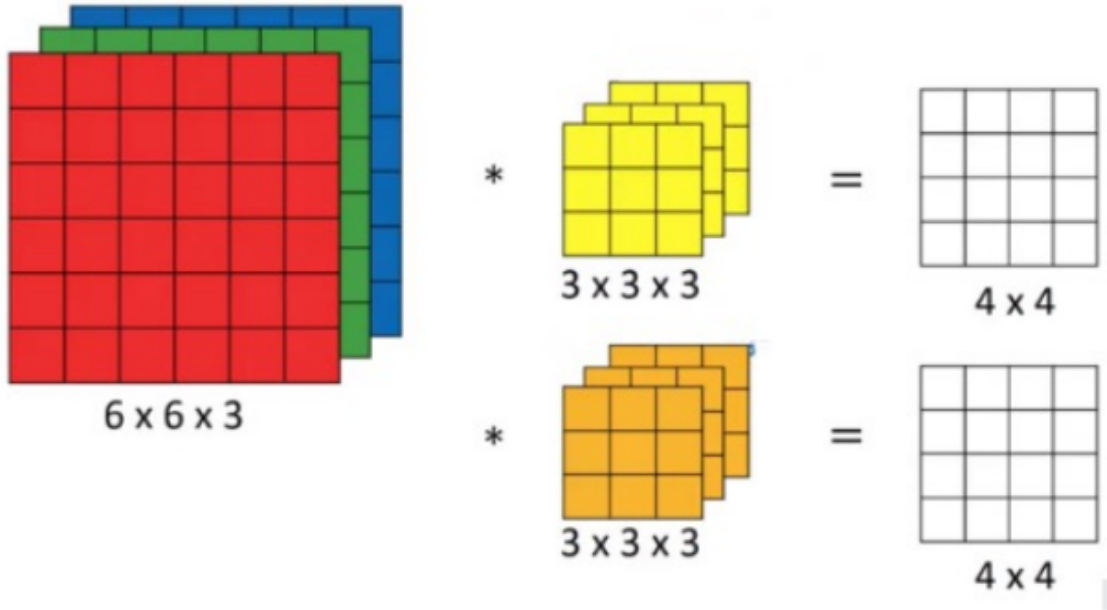


Применение сверточного фильтра к цветному изображению



- Мы делаем фильтр рангом 3, задаем ему ту же глубину, что и входному сигналу.
- Остальные аспекты операции свертки остаются неизменными.

Применение сверточного фильтра к цветному изображению



Если бы применили 2 фильтра, на выходе получился бы тензор с формой 4 x 4 x 2.

Если бы применили f фильтров, на выходе получился бы тензор с формой 4 x 4 x f

Более подробную информацию о работе сверточных фильтров и слоев см. в главе 8.1 учебника.

Главная идея

- Эти фильтры кажутся превосходными, но как нам получить числа в каждом фильтре?

Главная идея

- Эти фильтры кажутся превосходными, но как нам получить числа в каждом фильтре?
- На самом деле, сверточные фильтры раньше разрабатывались вручную. Исследователи в области компьютерного зрения вложили много усилий в разработку фильтров, способных обнаруживать различные типы признаков изображения.

Главная идея

- Эти фильтры кажутся превосходными, но как нам получить числа в каждом фильтре?
- На самом деле, сверточные фильтры раньше разрабатывались вручную. Исследователи в области компьютерного зрения вложили много усилий в разработку фильтров, способных обнаруживать различные типы признаков изображения.
- Когда мы разобрались, как обучать глубокие нейронные сети с большим количеством весов, у нас возникла важная идея: рассматривать числа в фильтре как веса и просто обучать их на основе данных, точно так же, как мы обучаем все остальные веса.

Главная идея

- Эти фильтры кажутся превосходными, но как нам получить числа в каждом фильтре?
- На самом деле, сверточные фильтры раньше разрабатывались вручную. Исследователи в области компьютерного зрения вложили много усилий в разработку фильтров, способных обнаруживать различные типы признаков изображения.
- Когда мы разобрались, как обучать глубокие нейронные сети с большим количеством весов, у нас возникла важная идея: рассматривать числа в фильтре как веса и просто обучать их на основе данных, точно так же, как мы обучаем все остальные веса.
  - Это возможно, потому что сверточный фильтр — это просто нейрон, подобный тому, что мы видели на предыдущих лекциях, хотя и особый (см. приложение).

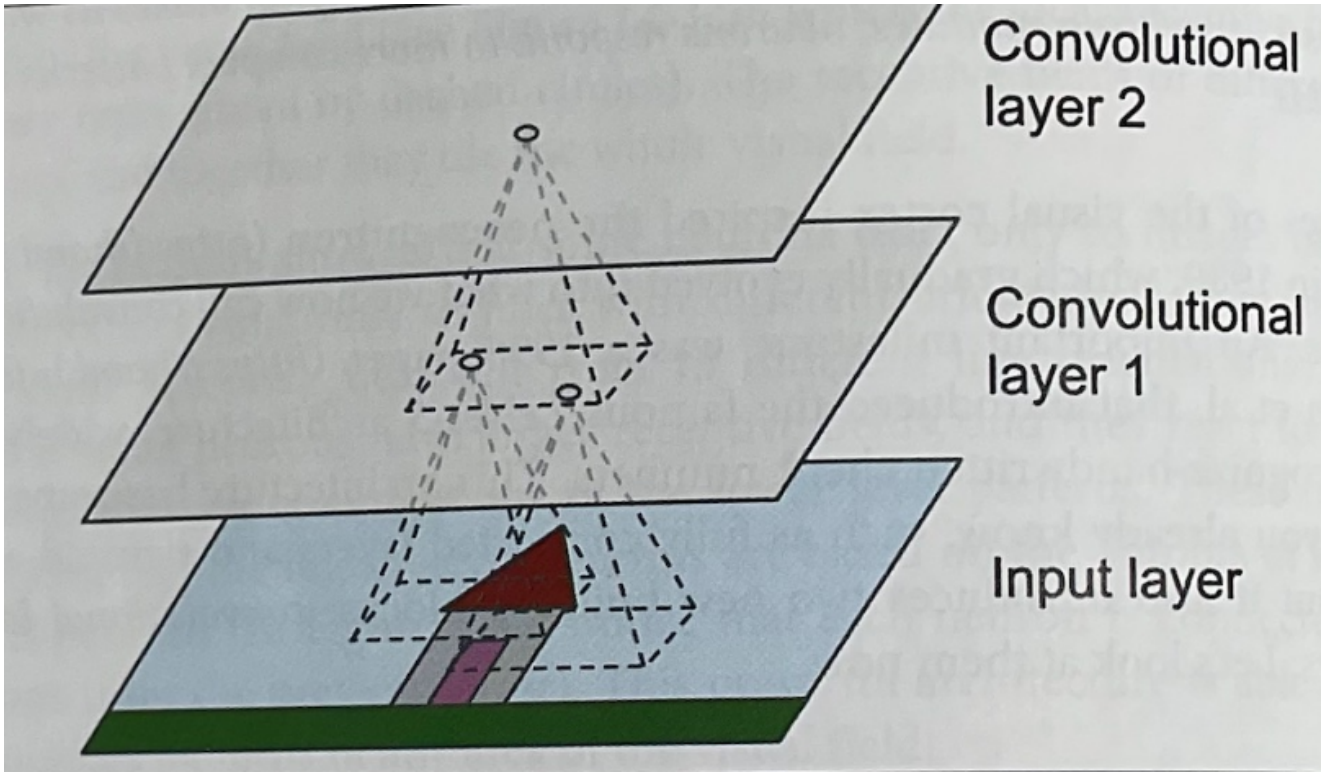
Главная идея

- Эти фильтры кажутся превосходными, но как нам получить числа в каждом фильтре?
- На самом деле, сверточные фильтры раньше разрабатывались вручную. Исследователи в области компьютерного зрения вложили много усилий в разработку фильтров, способных обнаруживать различные типы признаков изображения.
- Когда мы разобрались, как обучать глубокие нейронные сети с большим количеством весов, у нас возникла важная идея: рассматривать числа в фильтре как веса и просто обучать их на основе данных, точно так же, как мы обучаем все остальные веса.
  - Это возможно, потому что сверточный фильтр — это просто нейрон, подобный тому, что мы видели на предыдущих лекциях, хотя и особый (см. приложение).
  - Таким образом, весь наш механизм — нейроны, слои, функции потерь, градиентный спуск — идеально подходит для этой цели.

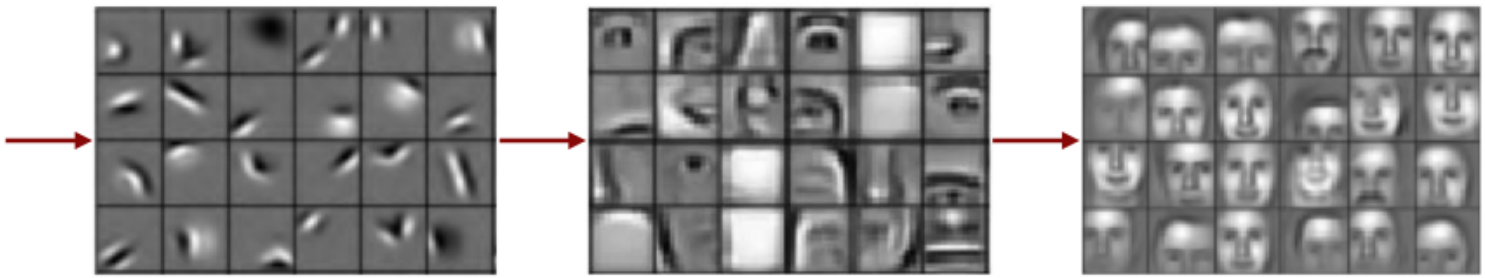
Главная идея

- Эти фильтры кажутся превосходными, но как нам получить числа в каждом фильтре?
- На самом деле, сверточные фильтры раньше разрабатывались вручную. Исследователи в области компьютерного зрения вложили много усилий в разработку фильтров, способных обнаруживать различные типы признаков изображения.
- Когда мы разобрались, как обучать глубокие нейронные сети с большим количеством весов, у нас возникла важная идея: рассматривать числа в фильтре как веса и просто обучать их на основе данных, точно так же, как мы обучаем все остальные веса.
- Это стало поворотным моментом в области компьютерного зрения и привело к значительному улучшению алгоритмических возможностей.

Верхнии сверточные слои «видят» больше информации из исходных входных данных, чем нижнии.



В результате нейронная сеть с большим количеством сверточных слоев может обучаться более сложным свойствам.



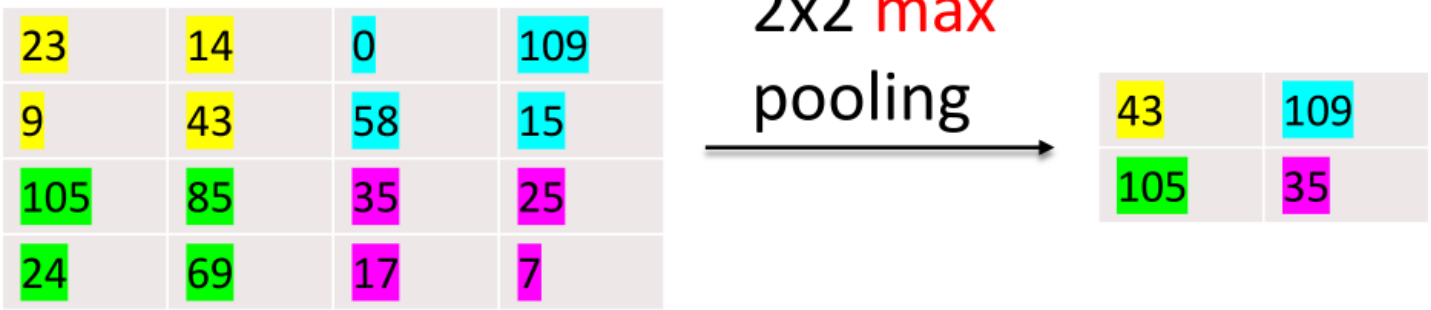
lines => edges, circles => faces!

Слои объединения(polling)

Слои пулинга (также называемые слоями понижения или субдискретизации) уменьшают размер тензора, получаемого на выходе из сверточного слоя.

Слои объединения(polling)

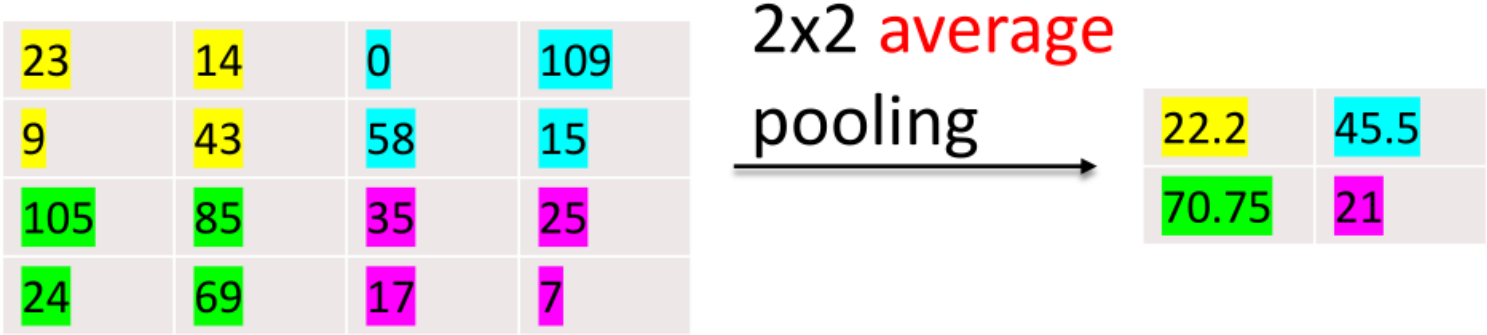
Слои пулинга (также называемые слоями понижения или субдискретизации) уменьшают размер тензора, получаемого на выходе из сверточного слоя.



- В максимальном пулинге мы берем максимальное значение из каждого блока 2x2.

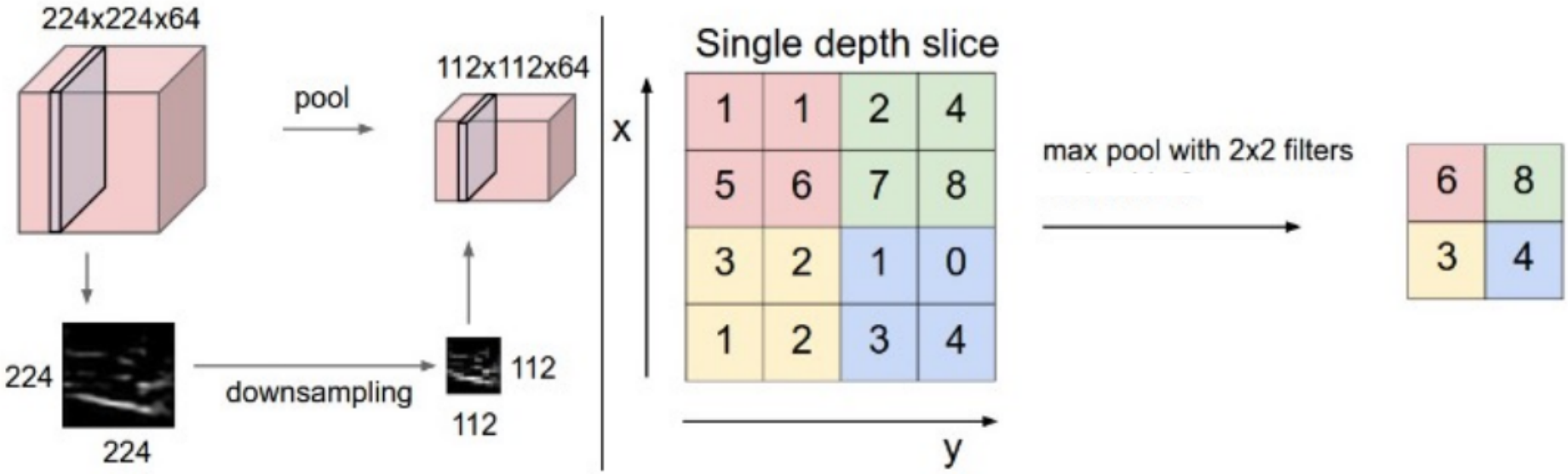
Слои объединения(polling)

Слои пулинга (также называемые слоями понижения или субдискретизации) уменьшают размер тензора, получаемого на выходе из сверточного слоя.



- В максимальном пулинге мы берем максимальное значение из каждого блока 2x2.
- В усредняющем пуллинге мы берем среднее значение каждого блока 2x2.

Слои объединения(polling)



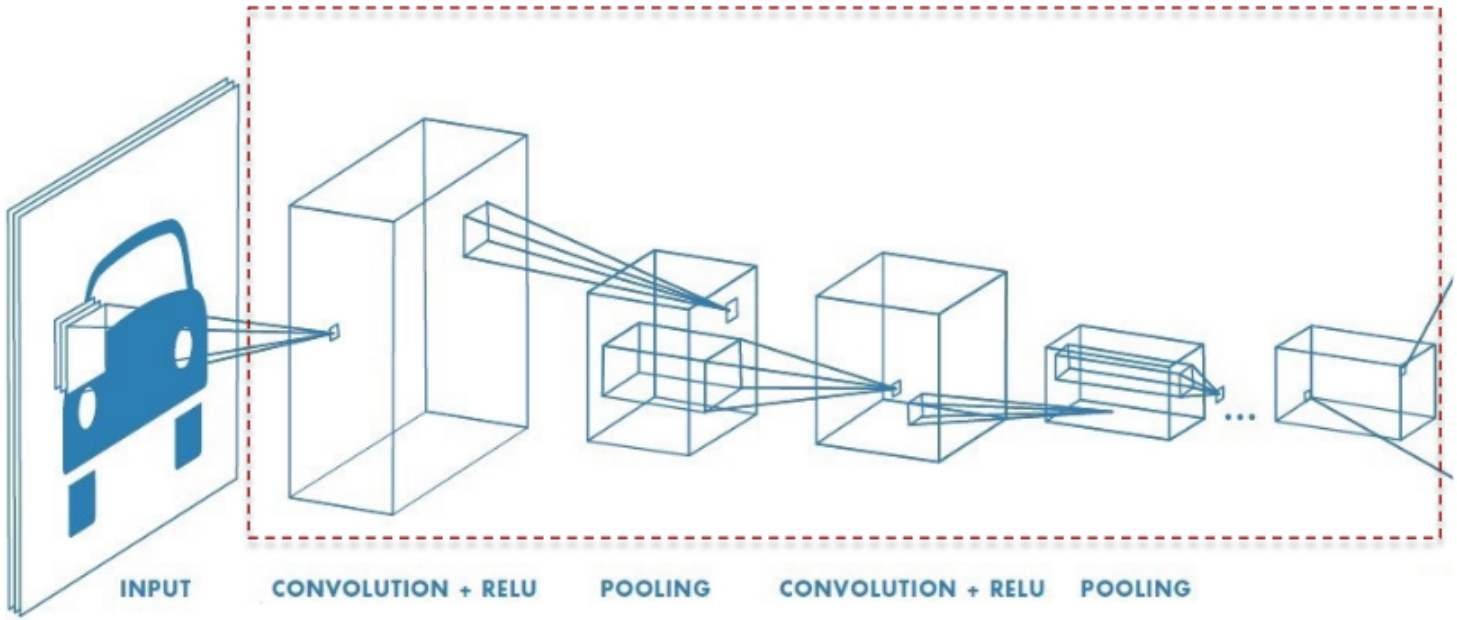
Интуитивное понимание макс пулинга уровня

- Максимальное объединение (max-pooling) действует как условие «ИЛИ»: если признак присутствует где-либо во входных данных, максимальное объединение его обнаружит, то есть максимальное объединение действует как детектор признаков.

Интуитивное понимание макс пулинга уровня

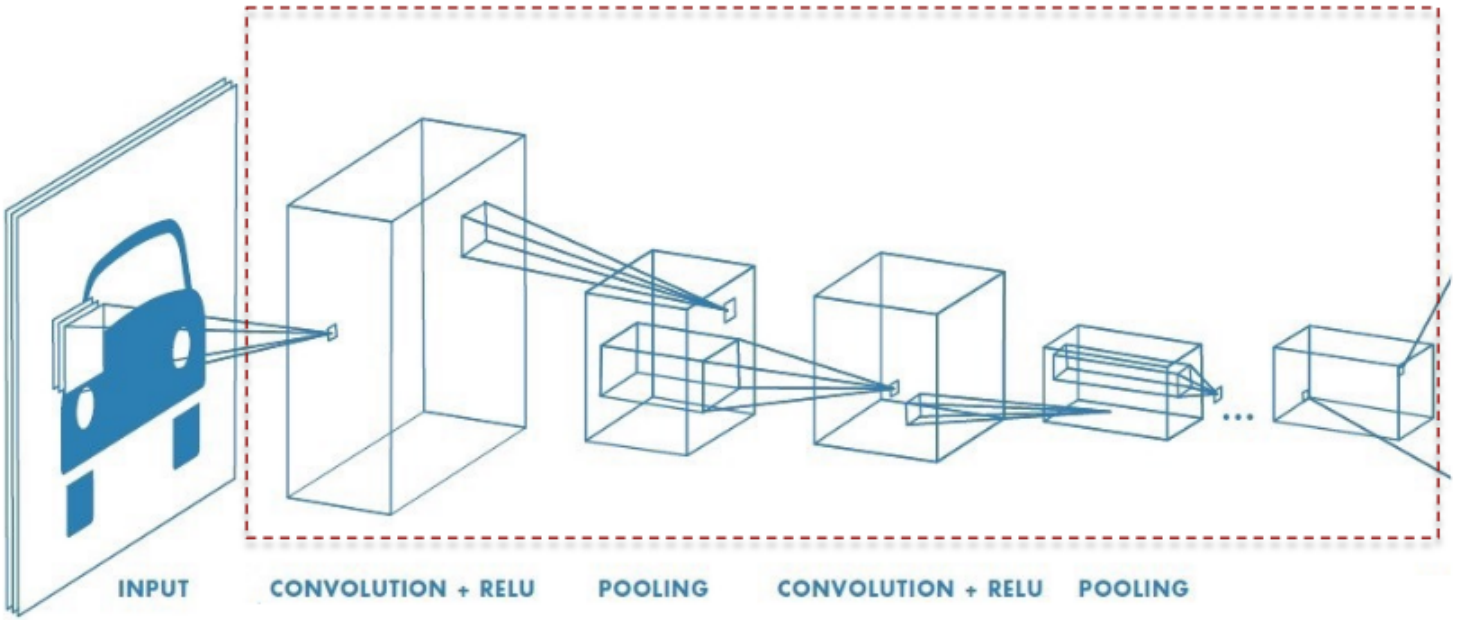
- Максимальное объединение (max-pooling) действует как условие «ИЛИ»: если признак присутствует где-либо во входных данных, максимальное объединение его обнаружит, то есть максимальное объединение действует как детектор признаков.
- Поскольку последовательные сверточные слои могут «видеть» все большую часть исходного входного изображения, следующие за ними слои максимального пулинга также могут обнаруживать наличие признака во все большей части исходного входного изображения.

Архитектура базовой сверточной нейронной сети



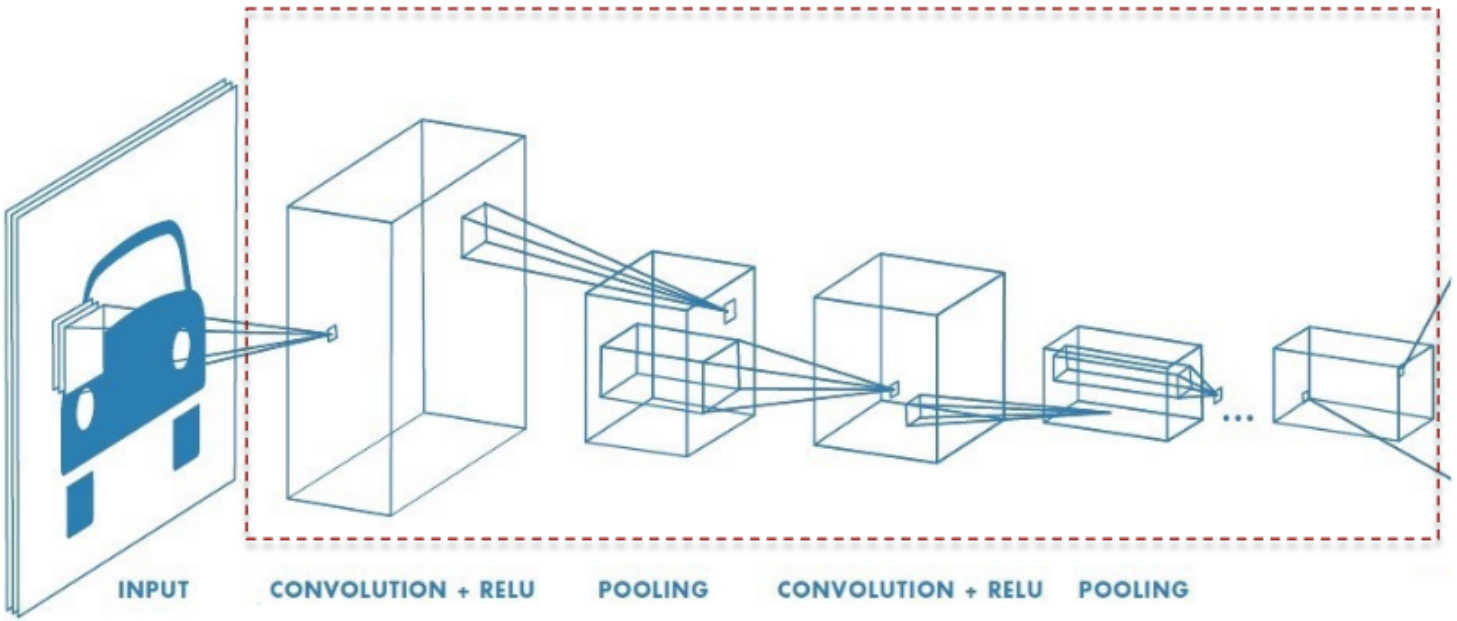
Последовательность сверточных блоков

Архитектура базовой сверточной нейронной сети



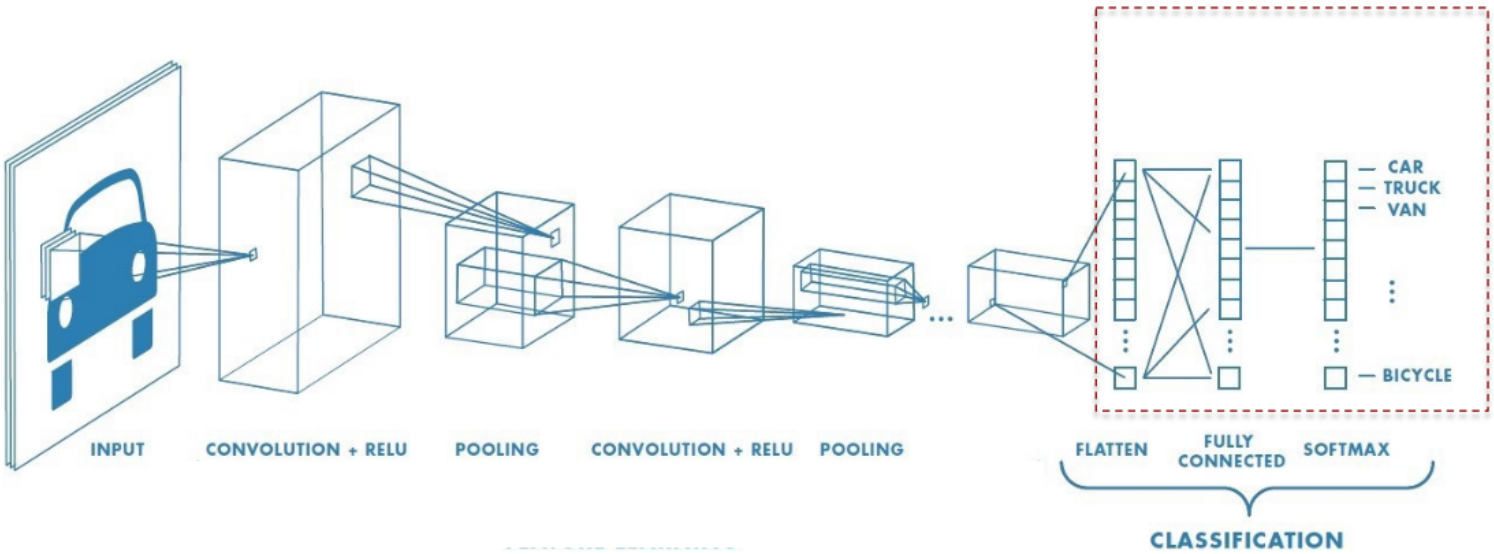
Каждый сверточный блок обычно состоит из 1-2 сверточных слоев, за которыми следует слой пулинга.

Архитектура базовой сверточной нейронной сети



Каждый последующий блок, как правило, будет иметь большую глубину, чем предыдущий, но меньшую высоту/ширину. Сравните это с этим.

В конце мы преобразуем тензор в одномерный поток, пропускаем его через полносвязные слои, а затем через выходной слой.



Итоговый тензор преобразуется в длинный вектор и передается через 0 или более скрытых слоев на выходной слой.

Colab: Давайте решим задачу Fashion MNIST с помощью сверточных слоев!

Ссылка на colab

Далее: Мы будем работать с цветными изображениями.

Увлекательное приложение: Классификатор сумок и обуви, основанный на менее чем 100 изображениях!

Собранный из интернета набор данных, содержащий менее 100 цветных изображений сумок и обуви.

Используя этот небольшой набор данных, мы создадим нейронную сеть глубокого обучения, которая сможет с высокой точностью классифицировать вашу обувь или сумку прямо на уроке!



Colab: Давайте создадим классификатор обуви и сумок с использованием сверточных слоев!

Ссылка на colab

Можем ли мы сделать лучше? У нас всего 100 примеров каждого класса.

Перенос обучения с использованием предварительно обученных нейронных сетей.

Перенос знаний в обучение использует преимущества двух исследовательских направлений.

Тенденция 1: Исследователи разработали архитектуры нейронных сетей, хорошо подходящие для различных типов данных. Например:

тип данных	Архитектура
Все	Остаточные сети
Изображения	Сверточные уровни
Последовательности (например, естественный язык, аудио, видео, последовательности генов)	Трансформеры

Перенос знаний в обучение использует преимущества двух исследовательских направлений.

Тенденция 1: Исследователи разработали архитектуры нейронных сетей, хорошо подходящие для различных типов данных. Например:

тип данных	Архитектура
Все	Остаточные сети
Изображения	Сверточные уровни
Последовательности (например, естественный язык, аудио, видео, последовательности генов)	Трансформеры

Тенденция 2: Используя эти архитектурные инновации, исследователи обучили высокопроизводительные глубокие нейронные сети на множестве больших реальных наборов данных. Доступно множество предварительно обученных моделей!

Перенос обучения предполагает адаптацию такой предварительно обученной сети к конкретной задаче, а не проектирование и обучение сети с нуля.

Можно ли применить трансферное обучение для создания более совершенного классификатора сумок/обуви?[1]

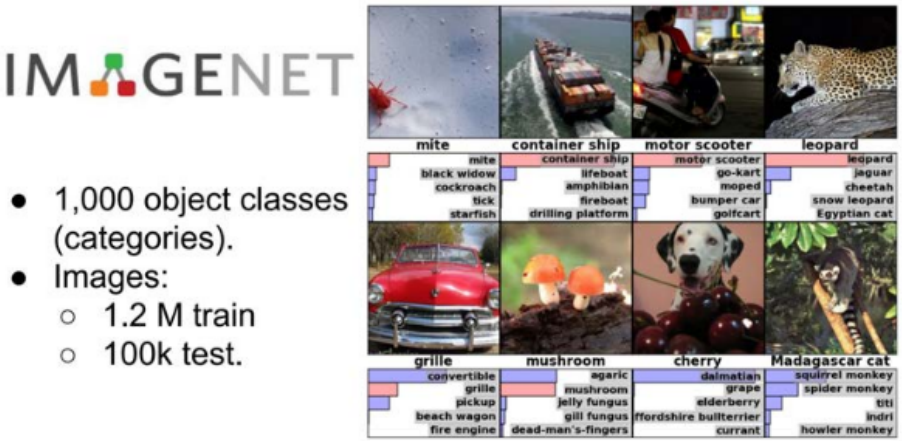
Сумки и обувь — это «повседневные предметы», и вы можете оглядеться вокруг и посмотреть, есть ли какие-либо нейронные сети, обученные на наборе данных изображений «повседневных предметов».

[1] Первым делом нужно посмотреть, не создал ли кто-нибудь уже подобный классификатор и не выложил ли его на GitHub.

Можно ли применить трансферное обучение для создания более совершенного классификатора сумок/обуви?

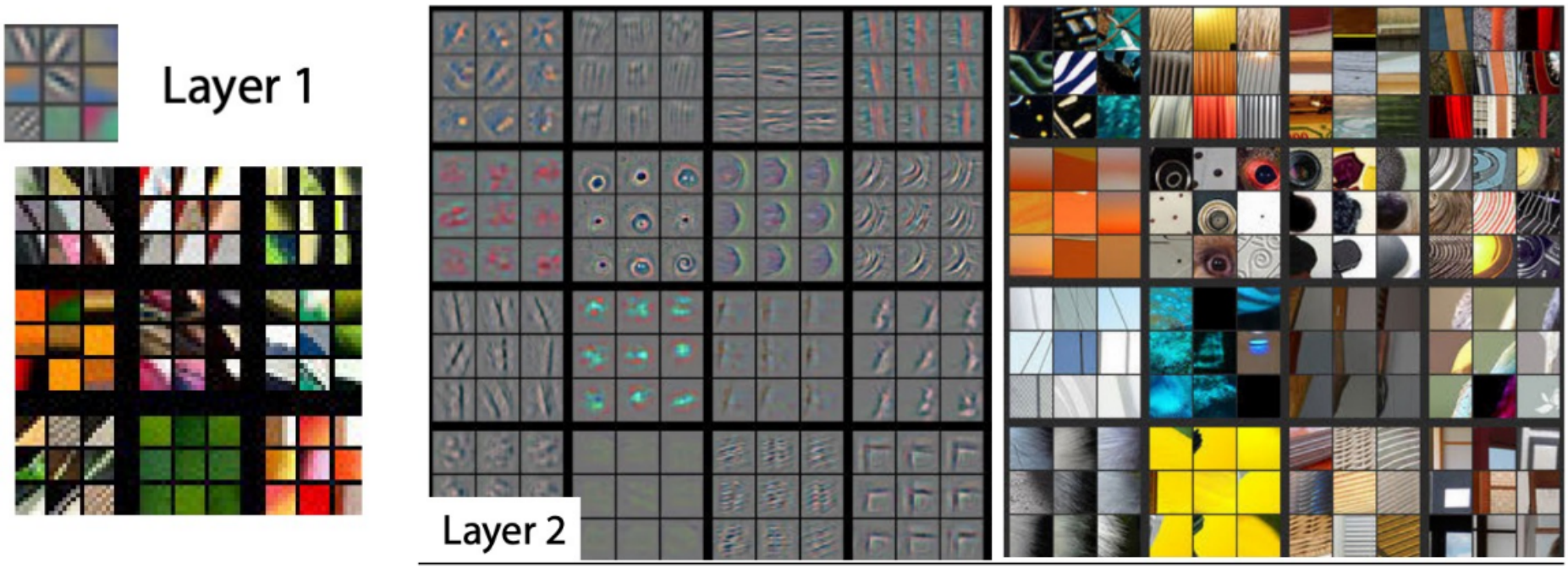
Сумки и обувь — это «повседневные предметы», и вы можете оглядеться вокруг и посмотреть, есть ли какие-либо нейронные сети, обученные на наборе данных изображений «повседневных предметов».

Оказывается, набор данных ImageNet содержит миллионы изображений 1000 категорий повседневных предметов. Мы можем поискать нейронные сети, обученные на ImageNet.

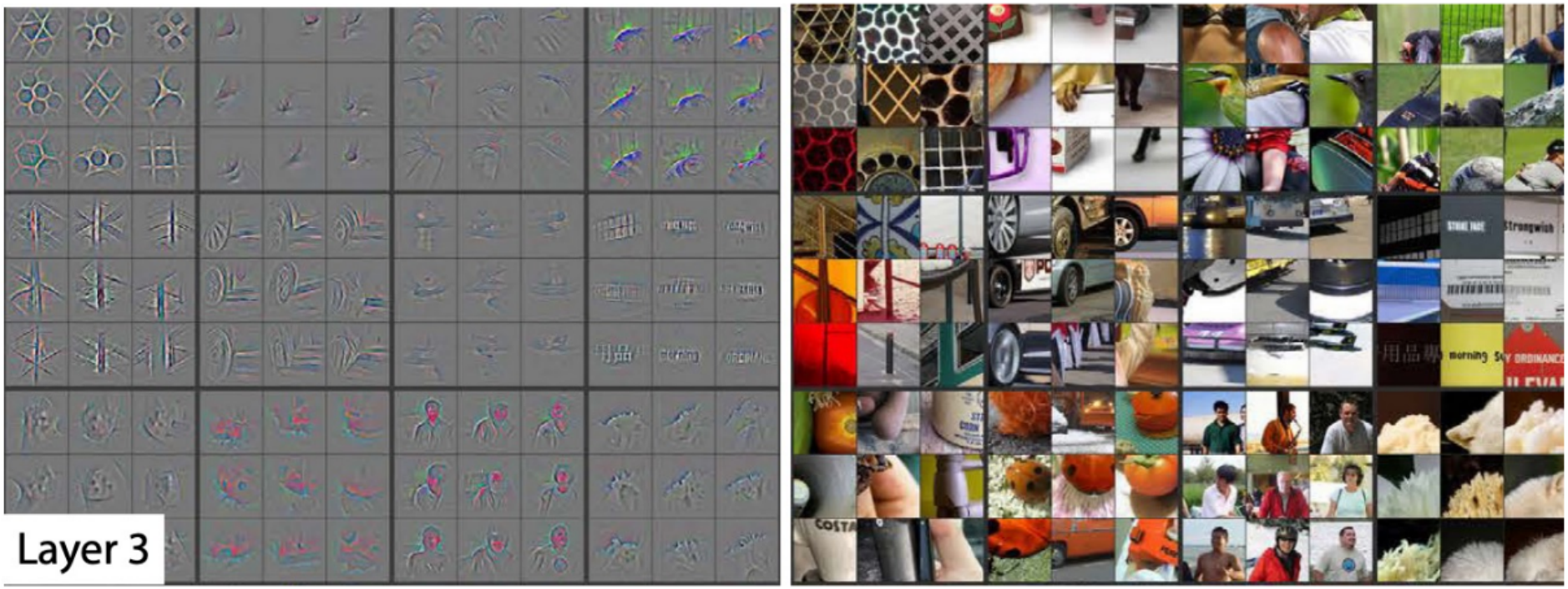


https://www.slideshare.net/xavigiro/image-classification-on-imagenet-d1l4-2017-upc-deep-learning-for-computer-vision/

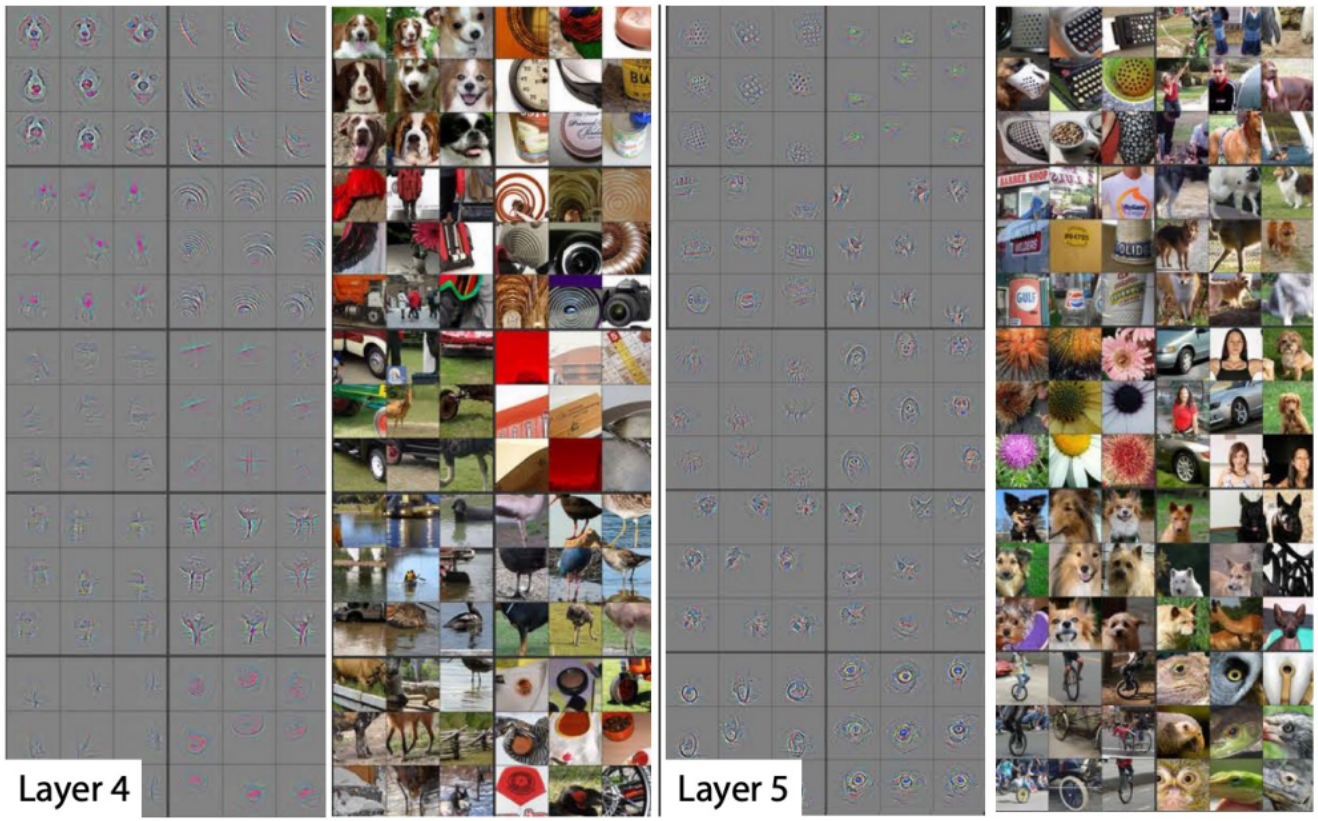
Нейронная сеть, обученная на ImageNet и демонстрирующая хорошие результаты на этой платформе, по сути, разработала интеллектуальное иерархическое представление этих объектов на всех своих слоях.



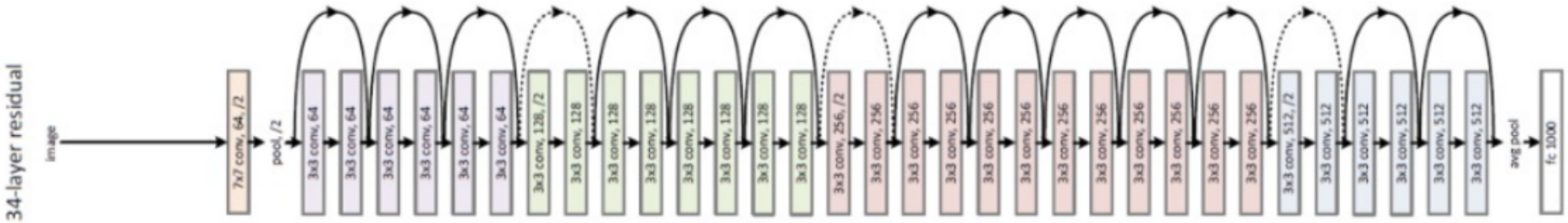
Нейронная сеть, обученная на ImageNet и демонстрирующая хорошие результаты на этой платформе, по сути, разработала интеллектуальное иерархическое представление этих объектов на всех своих слоях.



Нейронная сеть, обученная на ImageNet и демонстрирующая хорошие результаты на этой платформе, по сути, разработала интеллектуальное иерархическое представление этих объектов на всех своих слоях.

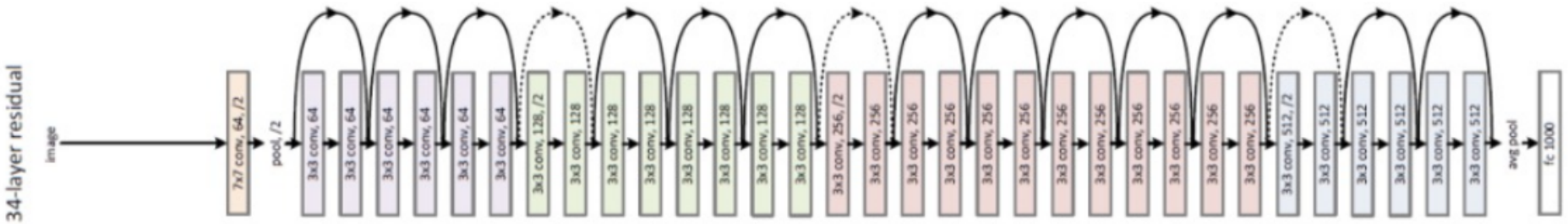


Семейство нейронных сетей ResNet было обучено на ImageNet и показало очень хорошие результаты в соответствующем конкурсе ImageNet.



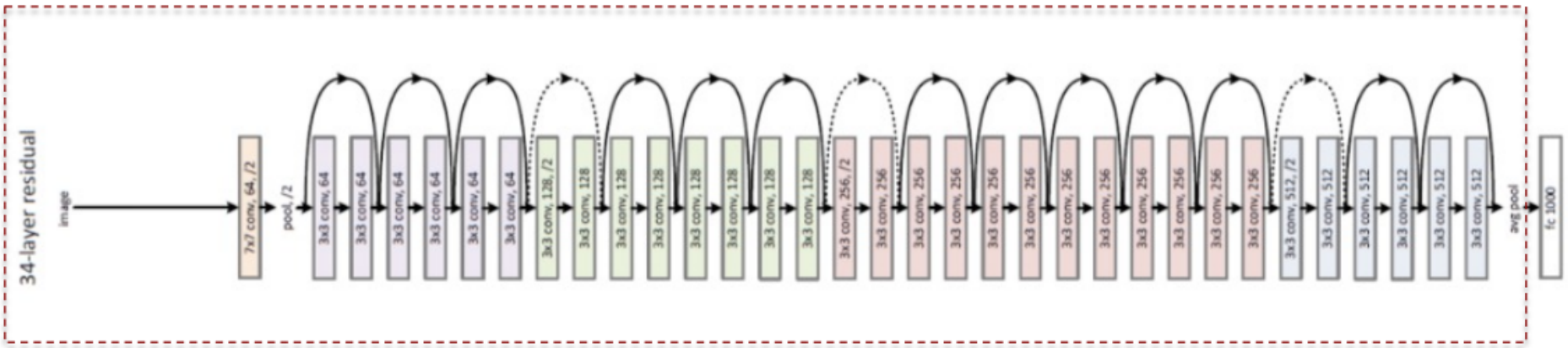
Можно ожидать, что (полученные) веса и смещения слоев ResNet будут отражать «знание» (как показано на предыдущих слайдах) о характеристиках миллионов повседневных изображений, на которых обучалась сеть.

Но мы не можем использовать ResNet в его нынешнем виде.

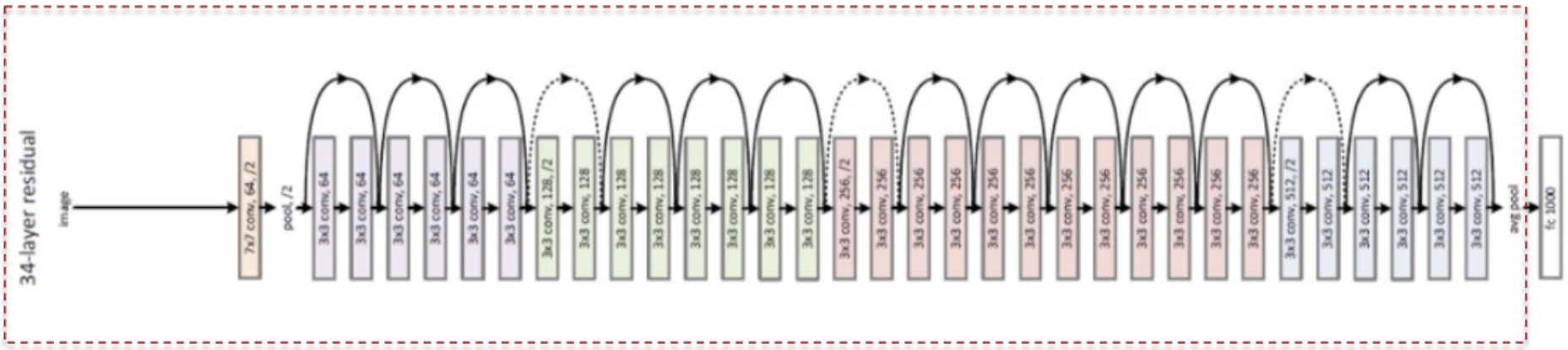


Помните, что ResNet был разработан для классификации изображений по 1000 категориям. Наша задача другая – нас интересуют только две категории: сумки и обувь.

Итак, мы берём ResNet и останавливаемся непосредственно перед последним слоем.



Мы можем обрабатывать наши изображения с помощью этой «безголовой» сети ResNet.

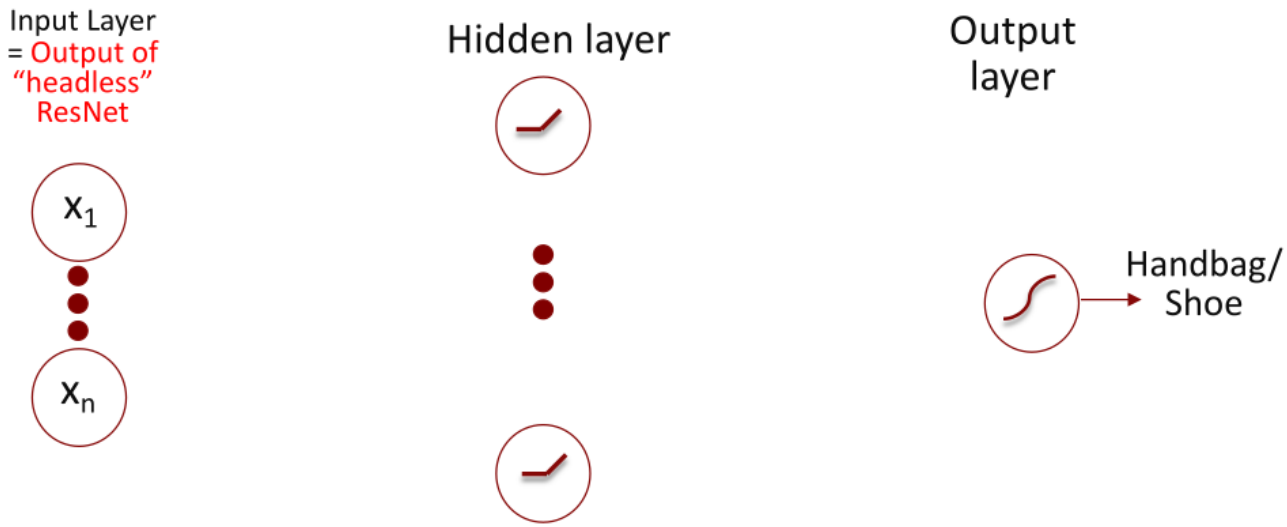


В результате получится «умное представление» повседневного изображения, которое можно использовать для различных видов категоризации.

Представьте, что изображение «помечено» множеством общих тегов, и мы можем использовать эти теги так, как нам угодно.

Мы можем рассматривать выходные данные «безголовой ResNet» как интеллектуальное представление исходного изображения и использовать его для обучения простой нейронной сети с одним скрытым слоем.

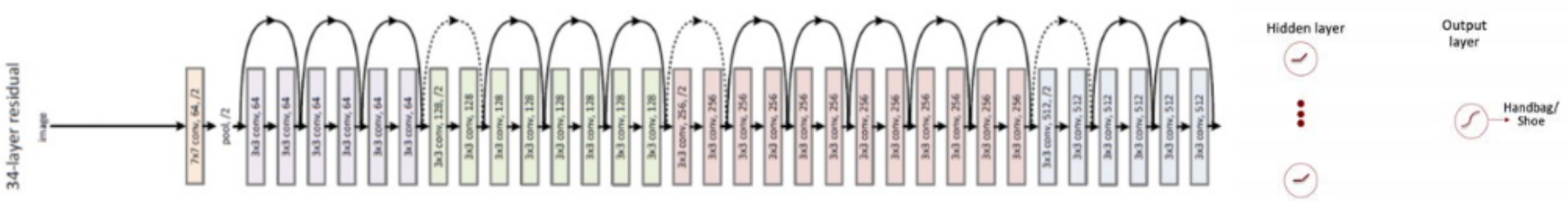
Мы можем рассматривать выходные данные «безголовой ResNet» как интеллектуальное представление исходного изображения и использовать его для обучения простой нейронной сети с одним скрытым слоем.



Поскольку входные данные для скрытого слоя находятся на более высоком уровне абстракции, он может научиться классифицировать сумки и обувь, имея очень мало примеров.

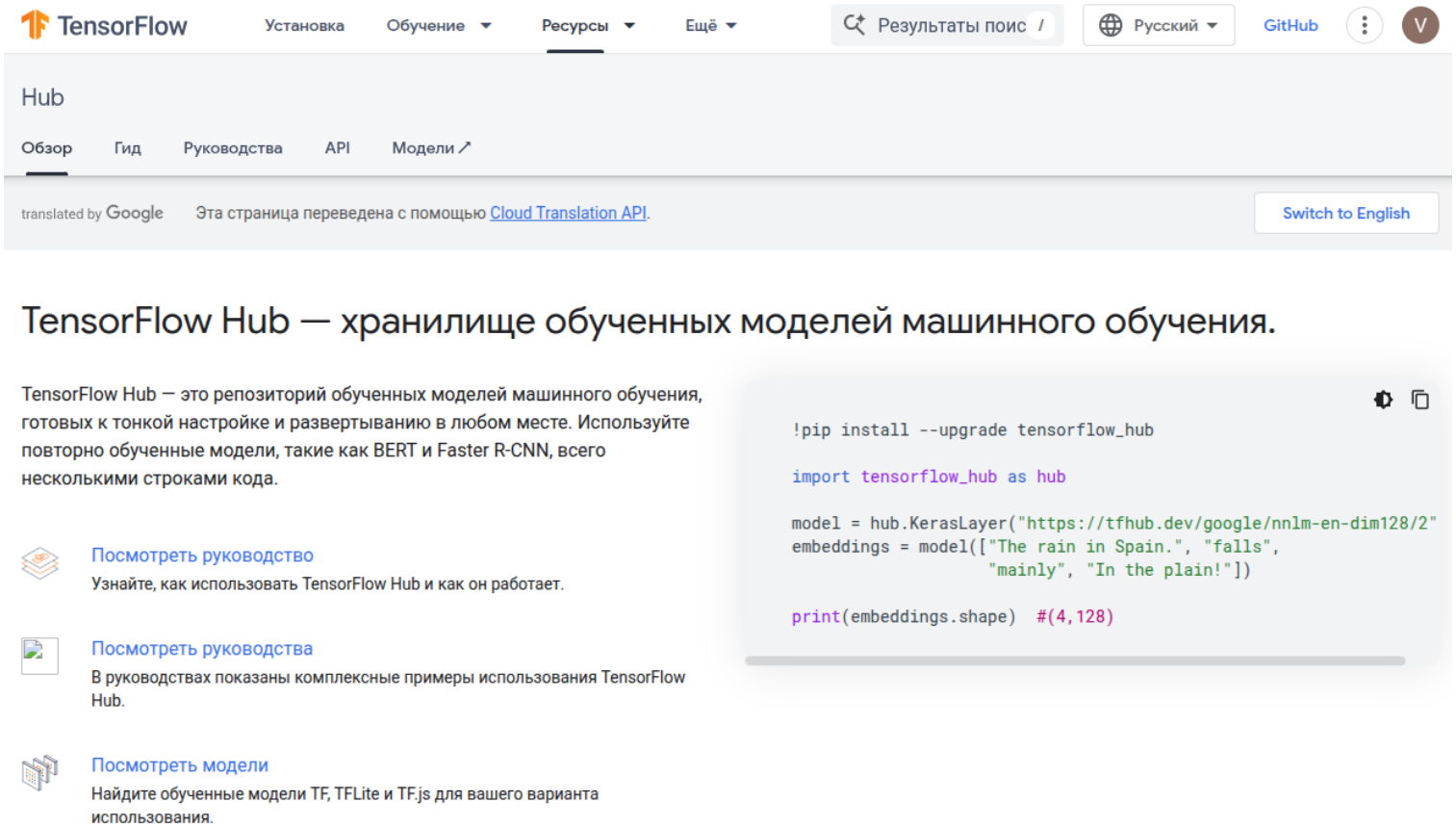
В этом и заключается основная идея трансферного обучения...

Это основная идея трансферного обучения, но можно использовать и более сложные методы.



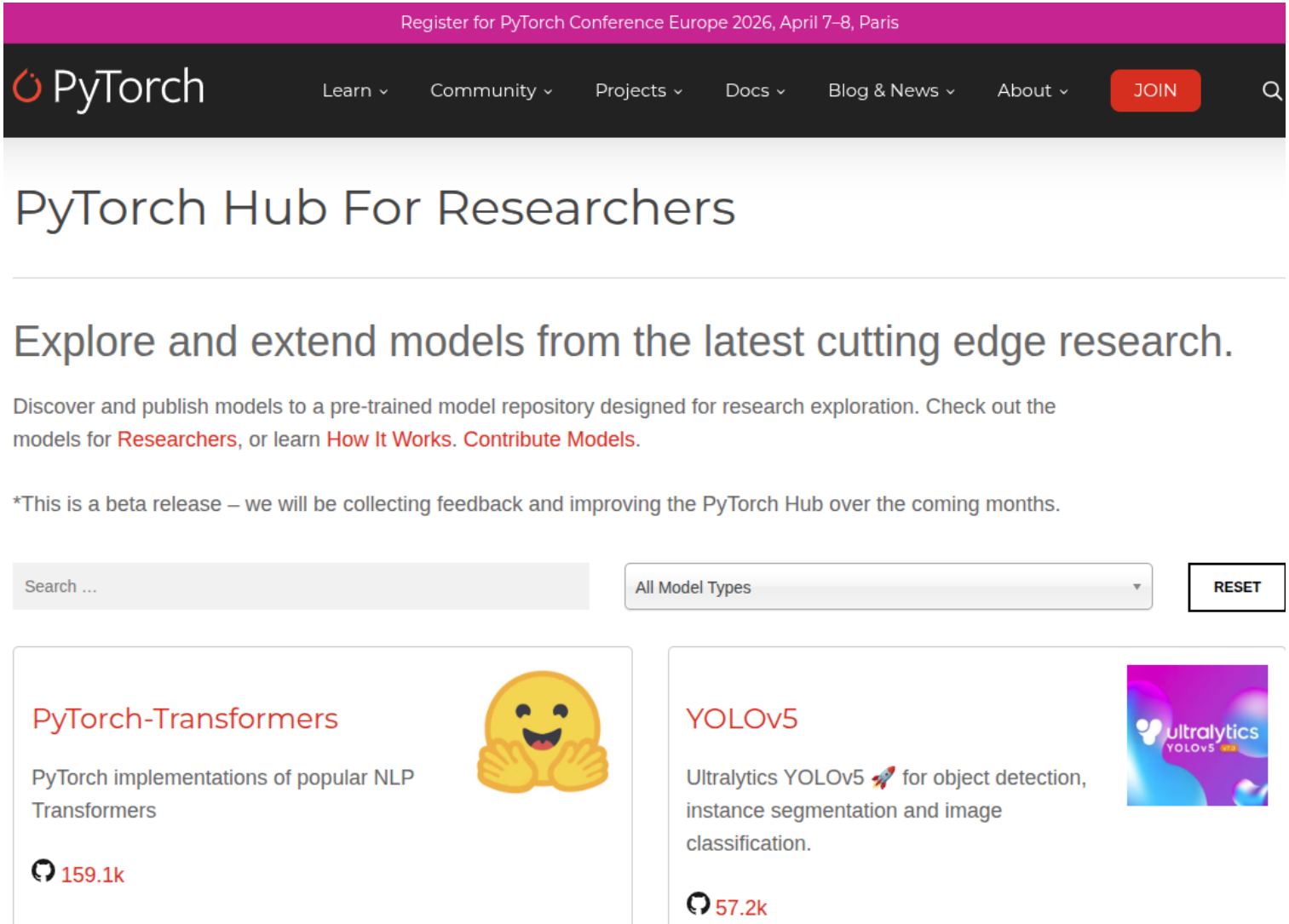
- Вы можете соединить «безголовую ResNet» с нашей небольшой нейронной сетью и обучить всю сеть от начала до конца. Это называется тонкой настройкой.
- Однако начинать обучение ОБЯЗАТЕЛЬНО следует с весами и смещенями, которые поставляются с ResNet, а не с нуля.
- Вы изучите это в домашнем задании.

Предварительно обученные модели можно найти в TensorFlow Hub.



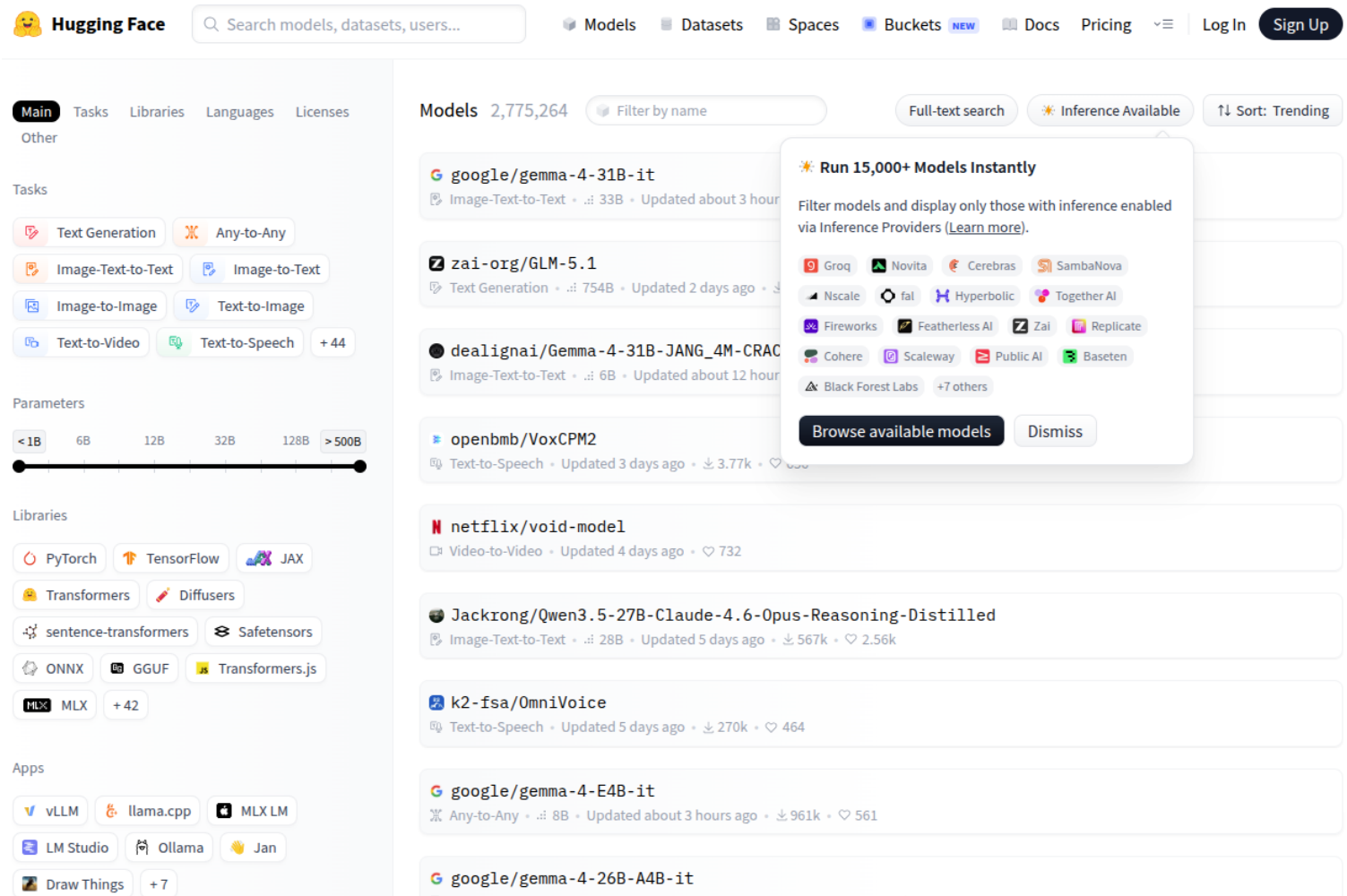
https://www.tensorflow.org/hub

pyTorch hub



https://pytorch.org/hub/

... и Hugging Face Hub



https://huggingface.co/models

Приложение

Сверточный фильтр — это всего лишь слегка модифицированный нейрон.

- Традиционный нейрон (в первом скрытом слое) подключен ко всем пикселям входного изображения. В отличие от него, нейрон сверточного фильтра подключен только к пикселям в небольшой области входного изображения.
- Скольжение фильтра по изображению можно рассматривать как использование разных фильтров для каждого окна, но с одинаковыми весами.
- Преимущества
  - Сохраняет местную близость
  - Гораздо меньше параметров
  - Инвариантность к сдвигу – позволяет обнаружить один и тот же шаблон независимо от того, где он появляется на изображении.