

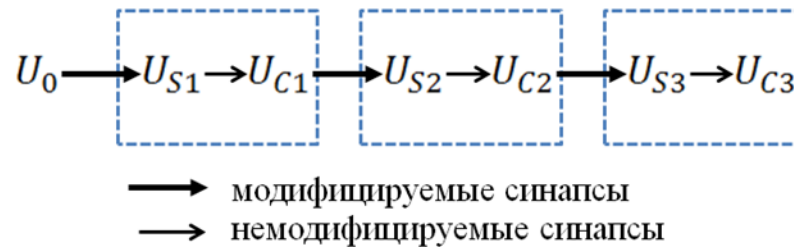
Сверточные нейронные сети

В середине прошлого столетия ученые Торстен Визель и Дэвид Хьюбел исследовали зрительную кору головного мозга кошки и обнаружили, что существует два различных типа клеток. Так называемые *простые клетки* реагируют на прямые линии (полосы света, края стимулов) под определенным углом. *Сложные клетки* реагируют на движение линий (полос света, краев стимулов) в определенных направлениях. Спустя некоторое время, это открытие нашло применение в прикладной области информатики: Кунихика Фукусима воплотил идею простых и сложных клеток в неокогнитроне.

Неокогнитрон состоит из ретины (сетчатки) и последовательности групп нейронов, каждая из которых, в свою очередь, содержит два типа нейронных слоев. Простые нейроны (из слоев) принимают сигналы от сложных нейронов предыдущего слоя и обнаруживают одинаковые признаки, но в разных местах сигнала.

Сверточные нейронные сети

Сложные нейроны (из слоев) объединяют отклики простых нейронов из своей группы и стремятся редуцировать зависимость реакции на признаки от их положения. Таким образом, неоконгитрон приобретает способность детектировать определенные комбинации признаков, независимо от их положения на сетине



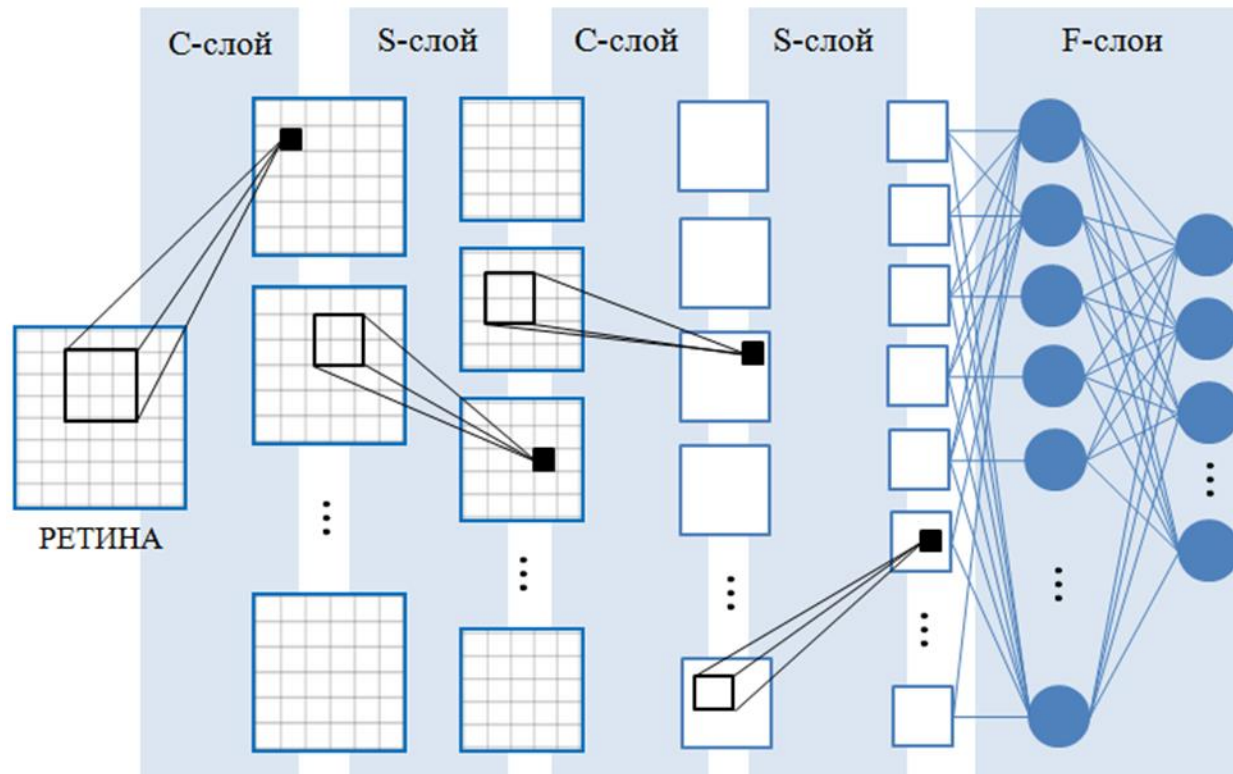
Упрощенная структура неоконгитрона

Отличие неоконгитрона от многослойного персептрона заключается в трех архитектурных принципах:

- локальные рецептивные поля,
- разделяемые веса,
- пространственная субдискретизация.

Сверточные нейронные сети

Развивая идею неокогнитрона, Ян ЛеКун вместе с соавторами предложили структуру классической сверточной нейронной сети (СНС) и алгоритм для ее обучения с учителем, СНС, предложенные ЛеКуном, это иерархичные многослойные нейронные сети, в которых сходятся три архитектурные идеи неокогнитрона, обеспечивающие некоторую степень инвариантности представления данных.

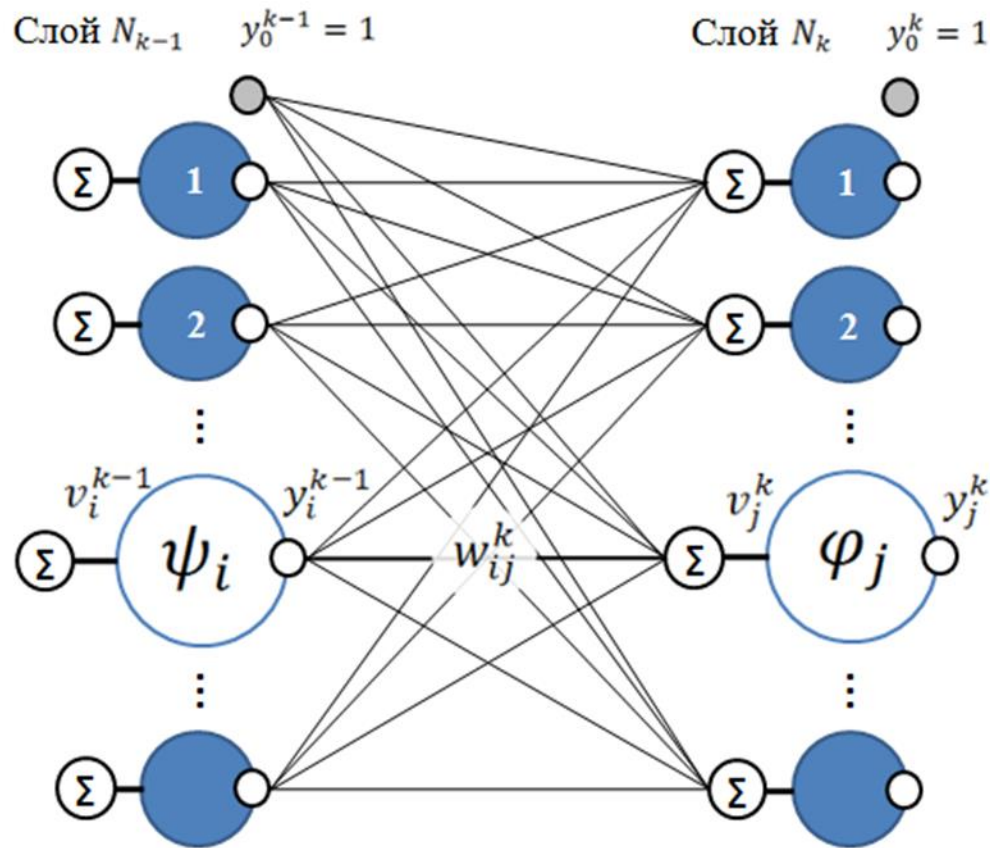


Сверточные нейронные сети

Устройство СНС заключается в чередовании *сверточных слоев* (convolutional, С-слои), реализующих концепцию простых клеток зрительной коры с гибкими (обучаемыми) весами, и *субдискретизирующих слоев* (sub-sampling, S-слои), построенных по примеру сложных клеток кортекса с жесткими весами. Завершается подобная каскадная модель, как правило, полносвязными нейронными слоями (fully-connected, F-слои).

Пусть — $\{F_1, F_2, \dots, F_{k_1}, \dots, C_1, C_2, \dots, C_{k_2}, \dots, S_1, S_2, \dots, S_{k_2}, \dots\}$ множество полносвязных (F_k), сверточных (C_k) и субдискретизирующих (S_k) слоев, образующих СНС. Обозначим связи между нейронами смежных слоев посредством матриц $W = \{w_{ij}^k\}$, где индексы i и j обозначают номера нейронов, связываемых весами, из соответствующих слоев $k-1$ и k .

Сверточные нейронные сети



Полносвязный нейронный слой.

Индукированные локальные поля v_j^k и активации y_j^k нейронов N_k слоя определяются согласно алгоритму прямого распространения сигнала (feedforward) значениями функций активации предыдущего слоя N_{k-1} и рассчитываются следующим образом:

$$v_j^k = \sum_i w_{ij}^k y_i^{k-1}, \quad y_j^k = \varphi_j(v_j^k).$$

Сверточные нейронные сети

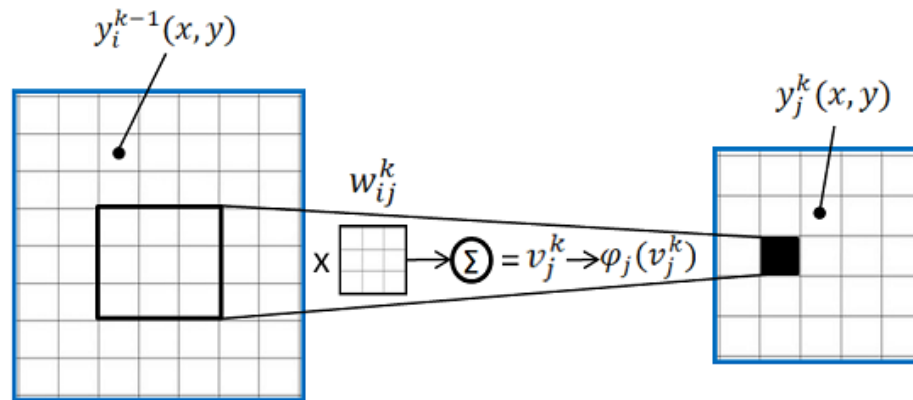
Существует большое множество различных функций активации нейронов. Наиболее распространенные из них

Название функции	Выражение
Линейная (linear)	$\varphi(x) = ax + b$
Логистическая (logit)	$\varphi(x) = \log(1 + e^x)$
Сигмоида (sigmoid)	$\varphi(x) = \frac{1}{1 + e^x}$
Гиперболический тангенс (th)	$\varphi(x) = a \operatorname{th} bx$
Пороговая (Хевисайда, hs)	$\varphi(x) = \begin{cases} 1, & x \geq T \\ 0, & x < T \end{cases}$
Линейная спрямленная (rectified linear, relu)	$\varphi(x) = \max(0, x)$

Сверточный слой.

Каждый сверточный слой C_k проецирует с помощью ядра и функции активации φ_j карты признаков Y_i^{k-1} на карту признаков Y_j^k . При построении образа ядро свертки w_{ij}^k размером $s \times s$ ячеек скользит по карте Y_i^{k-1} . Результаты каждой операции свертки отображаются в отдельную ячейку новой карты признаков Y_j^k посредством функции активации φ_j .

Сверточные нейронные сети



$$y_j^k(x, y) = \varphi_j \left(\sum_{i \in I} \sum_{(u_x, u_y) \in U} w_{ij}^k(u_x, u_y) y_i^{k-1}(x + u_x, y + u_y) \right),$$

Сверточный слой реализует сразу две архитектурные парадигмы. Каждой ячейке y_j^k карты сверточного слоя можно поставить в соответствие нейрон с синапсами w_{ij}^k , связанными с некоторой локальной областью $\check{y}_i^{k-1}$:

$$\check{y}_i^{k-1} = \{y_i^{k-1}(x + u_x, y + u_y) \mid \forall (u_x, u_y) \in U\},$$

с заданными функциями активации φ_j . Тогда у всех нейронов j -й карты k -го слоя будут общие веса (т.н. принцип разделяемых весов), но различные входы, определяемые координатами x и y (которые соответствуют локальным рецептивным полям).

Сверточные нейронные сети

Свертка в сверточных нейросетях — это цифровой фильтр, который сам обучается. При этом этот фильтр можно трактовать как как нейрон, выполняющий операцию взвешенного суммирования входных сигналов. Веса нейронов — это коэффициенты ядра свертки.

При применении его ко всему изображению вся эта процедура представляет собой прохождение частей изображения через слой одинаковых нейронов. Результат записывается в новое изображение, называющееся картой признаков.

В сверточном слое используют нескольких сверток с различными ядрами. При этом формируется несколько карт признаков. Каждая карта признаков, полученная при применении одной свертки отражает одинаковые локальные признаки во всех частях изображения

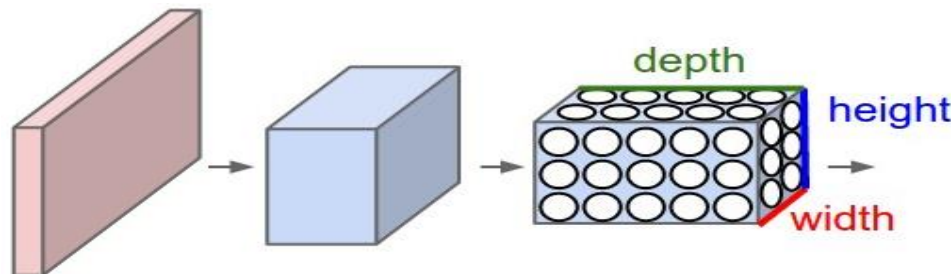
Карта признаков представляет собой набор нейронов, каждый из которых имеет тем большее значение, чем больше свяанный с ним фрагмент изображения похож на ядро.

Сверточные нейронные сети

Одна свертка – одна карта признаков



Каждый сверточный уровень преобразует предыдущую карту признаков в несколько карт последующих признаков, которые можно представить в виде 3D массива



Сверточные нейронные сети

Субдискретизирующий слой – слой пулинга.

Линейный субдискретизирующий слой S_k , заключающий в себе принцип пространственной субдискретизации, понижает размерность карт признаков, используя технику усреднения значений соседних пикселей. Для этого отображаемая карта признаков разбивается на равные непересекающиеся области. Формула отображения карты признаков Y_i^{k-1} на карту признаков Y_j^k выглядит следующим образом:

$$y_j^k(x, y) = \varphi_j \left(w_{ij}^k \times \sum_{(u_x, u_y) \in U} y_i^{k-1}(sx + u_x, sy + u_y) \right),$$

где k – коэффициент децимации, $U = (1..s) \times (1..s)$. При моделировании субдискретизирующего слоя часто используются следующие параметры:

$$s = 2, w_{ij}^k = \frac{1}{s^2} \equiv \text{const.}$$

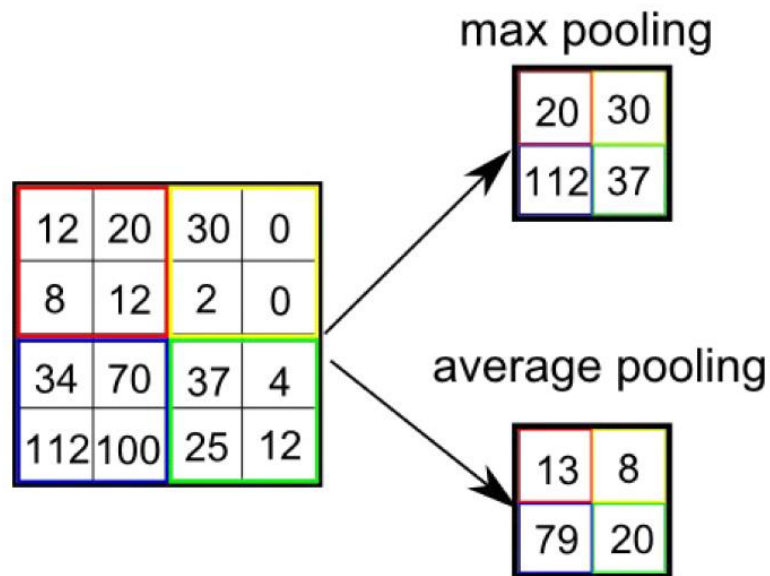
Улучшить работу нейронной сети помогает использование нелинейной децимации. Наиболее распространенным и эффективным нелинейным способом субдискретизации является слой типа *max-pooling*, использующийся для обучения больших сверточных сетей. В этом случае формула отображения карт признаков выглядит следующим образом:

$$y_j^k(x, y) = \max_{(u_x, u_y) \in U} \left(y_i^{k-1}(sx + u_x, sy + u_y) \right)$$

Сверточные нейронные сети

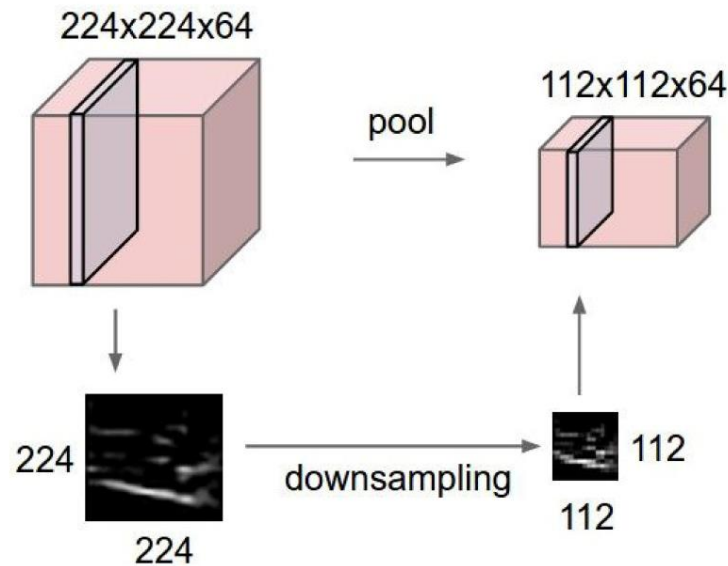
Пулинг основан на том, что изображения обладают свойством локальной скоррелированности пикселей — соседние пиксели, как правило, не сильно отличаются друг от друга. Таким образом, если из нескольких соседних получить какой-либо агрегат, то потери информации будут незначительными.

Операция pooling (max pool, avg pool)



Сверточные нейронные сети

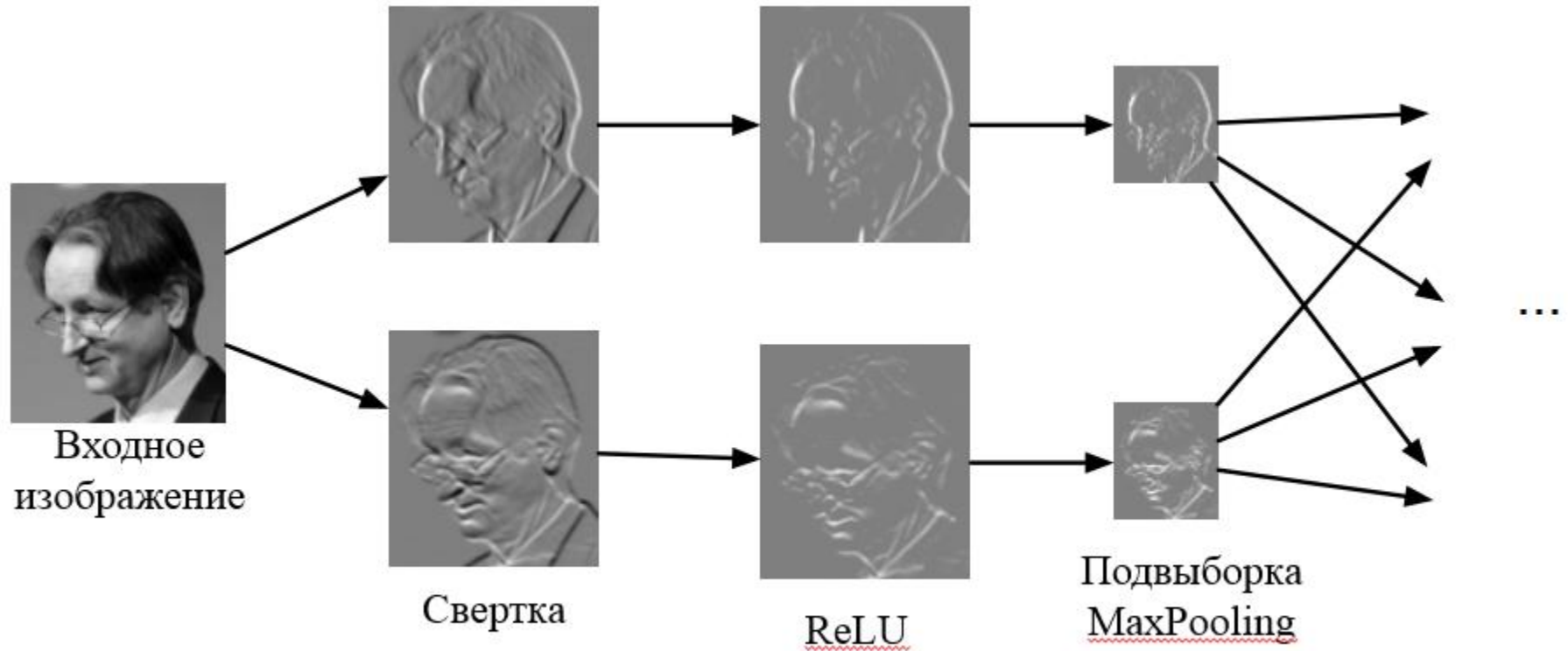
CNN: Pooling слой (downsampling)



Pooling-слой. Есть кубик какого-то объема — 3 канала, 10, или 100 каналов, которые насчитали свертками. Он просто уменьшает его по ширине и по высоте, остальные размерности не трогает. Уменьшение по высоте и ширине как правило в два раза. Поэтому окно пулинга берут 2x2 со сдвигом 2.

Сверточные нейронные сети

Пример свертки и подвыборки



Применена выпрямленная линейная функции активации (rectified linear unit, ReLU), которая выражается формулой: $f(s) = \max(0, s)$

Сверточные нейронные сети

Выходные слои.

В зависимости от типа решаемой задачи, может потребоваться интерпретировать выходы сети как вероятности распознаваемых классов. Тогда для нейронов выходного слоя используется функция активации софтмакс (softmax), являющаяся обобщением логистической функции:

$$\varphi_j(y^k) = P(\lambda = j|y^k) = \frac{e^{(y^k)^T w_{*j}^k}}{\sum_l e^{(y^k)^T w_{*l}^k}},$$

где λ — переменная, содержащая индекс класса. Предсказание сети $\hat{\lambda}$, индекс наиболее вероятного класса, определяется по формуле:

$$\hat{\lambda} = \operatorname{argmax}_j \varphi_j(y^k)$$

Сверточные нейронные сети

У CNN меньше параметров, чем у FNN

CNN

- вход картинка 100x100
- два свёрточных слоя по 100 плоскостей каждый (conv 5x5 и subsampling 2)
- полносвязный слой на 100 нейронов
- выход: 10 классов
- число параметров примерно **70К** ($25*100 + 25*100 + 25*25*100 + 100*10$)

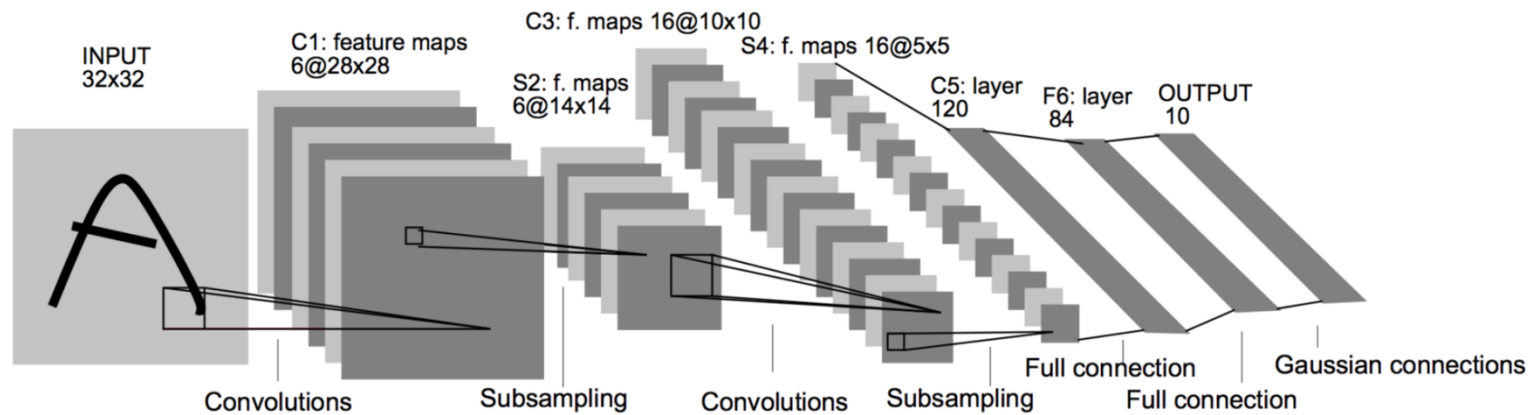
FNN

- вход: картинка 100x100
- три скрытых слоя по 100 нейронов каждый
- выход: 10 классов
- число параметров примерно **1М** ($10000*100 + 100*100 + 100*100 + 100*10$)

У СНС гораздо меньшее количество используемых весов по сравнению с любой полносвязной сетью, что упрощает процесс обучения

Сверточные нейронные сети

Первая сверточная сеть Ле Куна – в ней представлены основные принципы сверточных сетей. Использовалась для распознавания рукописных почтовых индексов. Эта архитектура была стандартным шаблоном для построения сверточных сетей вплоть до недавнего времени: свертка чередуется с пулингом несколько раз, затем несколько полносвязных слоев.



Основные строительные блоки — свертки 5×5 со сдвигом 1 и пулинг 2×2 со сдвигом 2. Свертки играют роль детекторов признаков, а пулинг (или сабсемплинг) используют для уменьшения размерности,

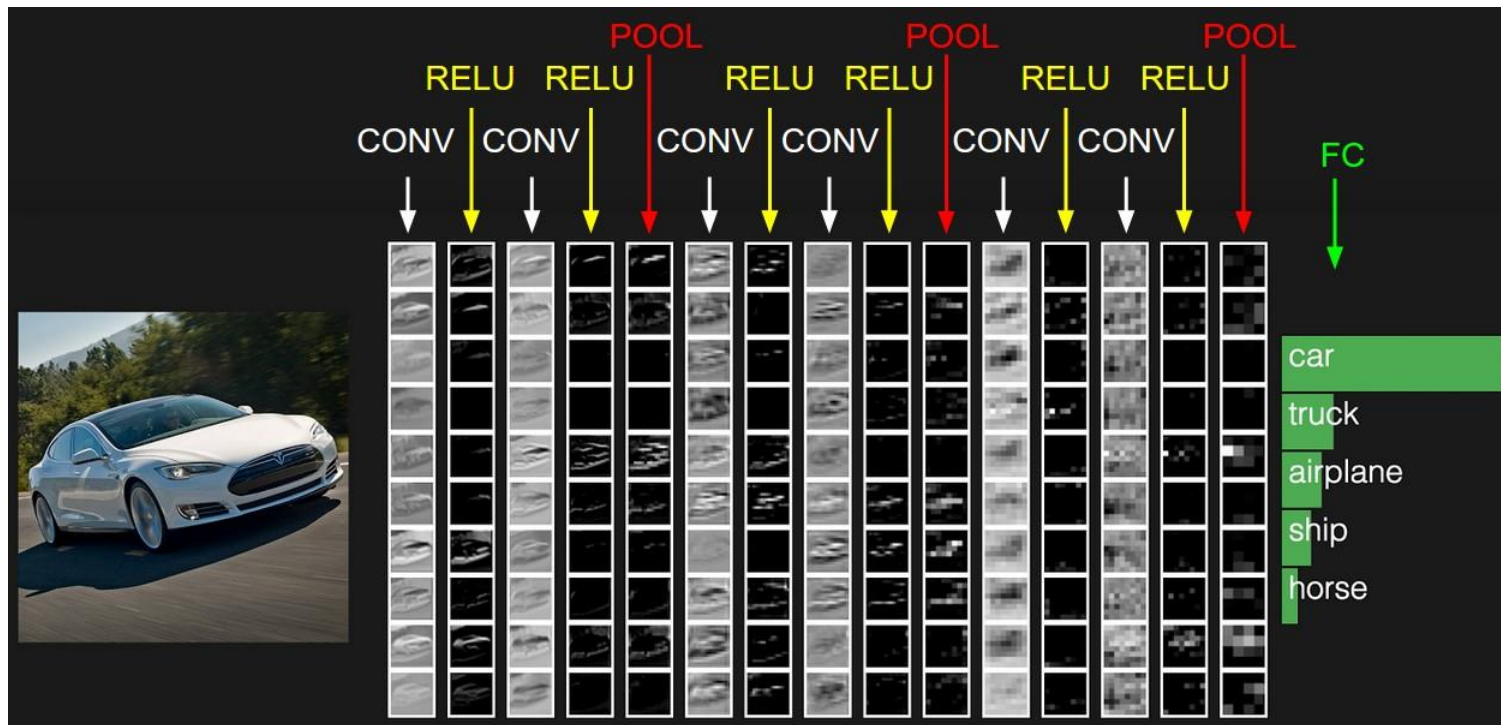
Сверточные нейронные сети

СНС состоит из нескольких пар чередующихся сверточных и субдискретизирующих слоев, после прохождения которых исходное изображение вырождается в вектор либо скаляр и поступает на вход многослойному персептрону либо любой другой выбранной пользователем полносвязной сети.

При последовательном прохождении пар чередующихся сверточных и субдискретизирующих слоев был получен бипирамидальный эффект, что означает увеличение количества карт признаков на каждой паре С-S слоев при одновременном уменьшении пространственного разрешения карт по сравнению с предыдущей С-S парой.

Сверточные нейронные сети

Функция активации нейронов как в сверточной, так и в выходной полносвязной сети может быть любой, удовлетворяющей условию гладкости, так как обучение структуры производится с помощью метода обратного распространения ошибки.



СНС построенная по архитектуре Ле Куна

Сверточные нейронные сети

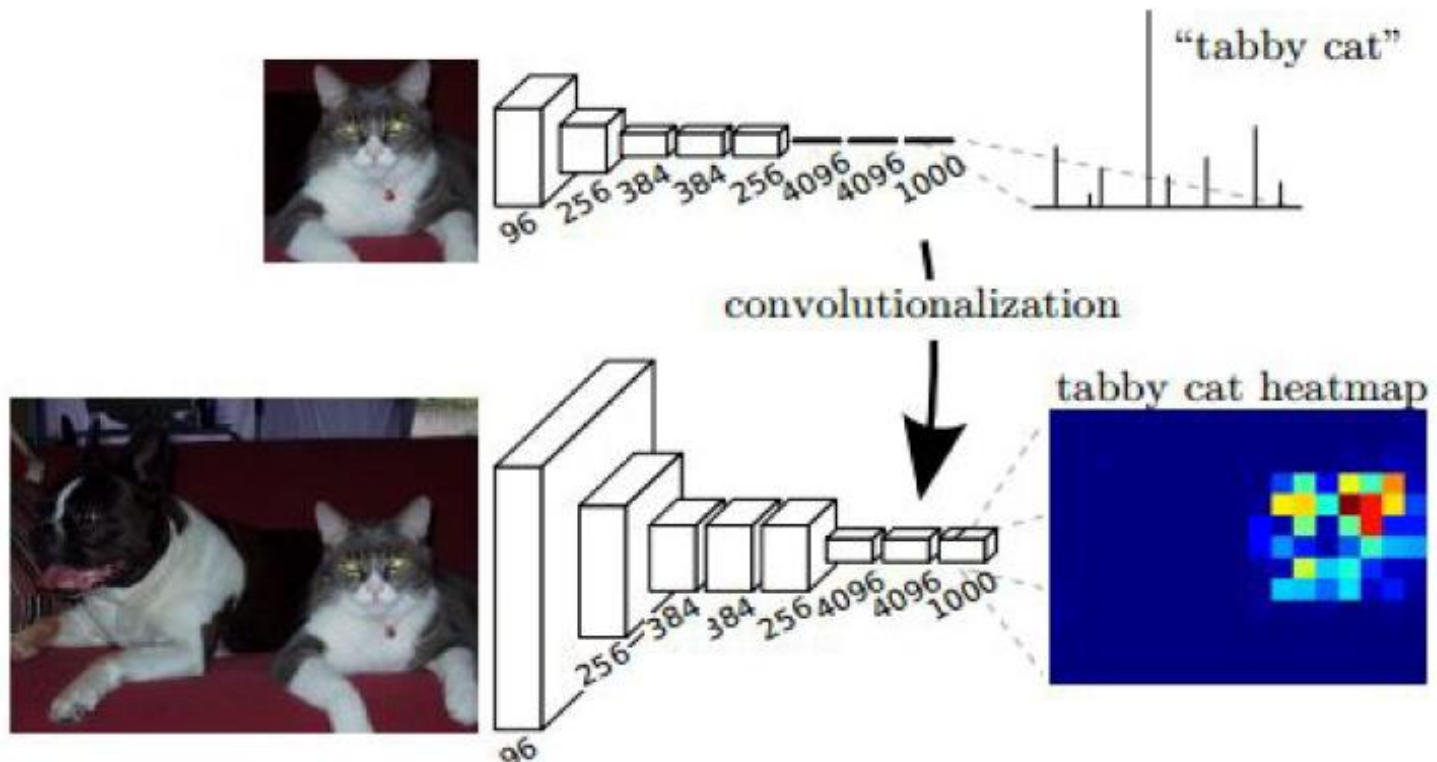
Сети глубокого обучения

Полностью сверточная сеть (FCN)

В ней полносвязный слой преобразуется в свертку. Для этого надо пересчитать веса полносвязного слоя в ядро свертки. При этом для эквивалентного сверточного слоя необходимо установить размерность фильтра равной размерности карты признаков, а количество фильтров должно равняться количеству нейронов данного слоя.

Например, пусть на вход полносвязного слоя поступают 512 карт признаков размером 7×7 (блок $7 \times 7 \times 512$), а его выход содержит 4096 нейронов. Тогда размер фильтра будет 7×7 (размер карты признаков), количество фильтров – 4096. Следовательно, выход будет просто $1 \times 1 \times 4096$, поскольку только один столбец глубины «подходит» по входному тому, давая идентичный результат как начальный уровень FC.

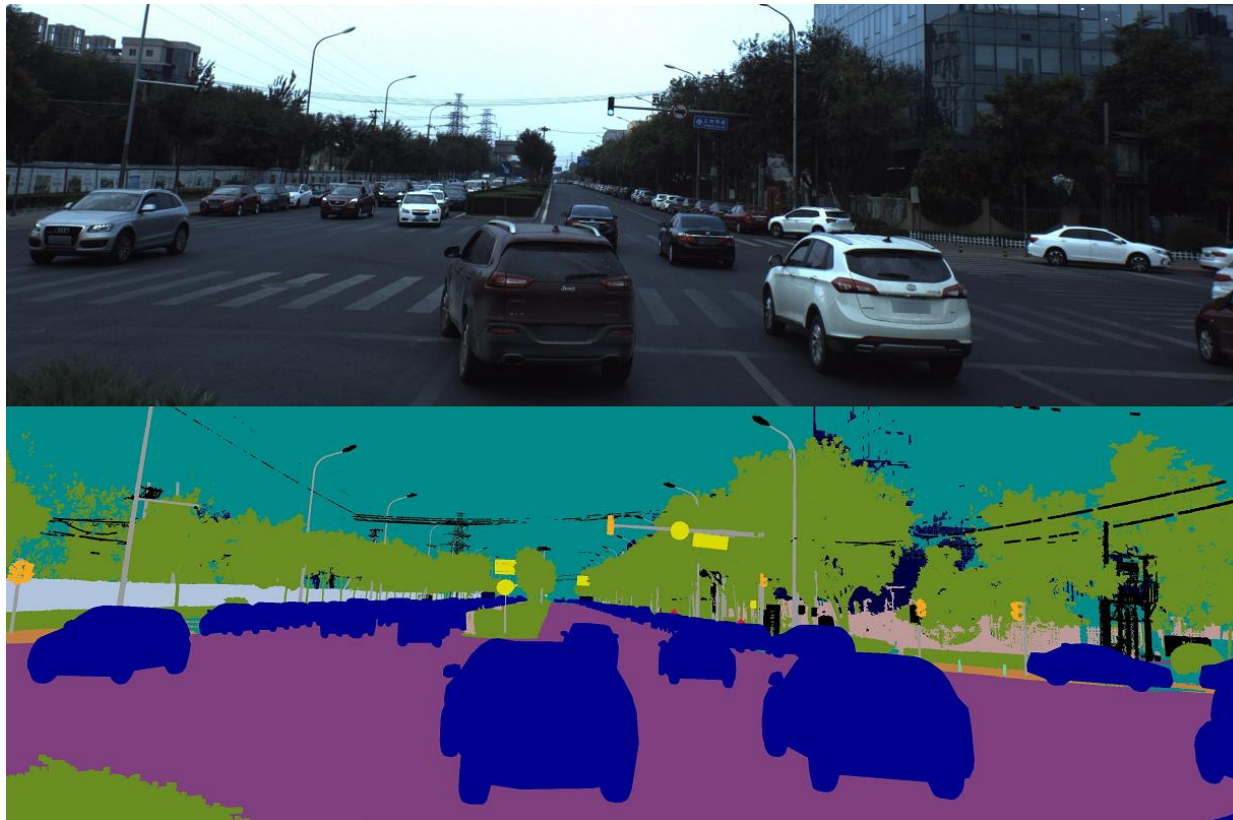
Сверточные нейронные сети



Пусть СНС научили определять 1000 классов. Потом полносвязный слой трансформировали в сверточный слой. Тогда эта нейросеть как бы работает тем же самым окном, на которое она была обучена (например 100x100), но теперь она может пробежаться этим окном по всему изображению и построить как бы тепловую карту на выходе — где в этом конкретном изображении находится конкретный класс.

Сверточные нейронные сети

Можно построить, например, 1000 этих Heatmap для всех своих классов и потом использовать это для определения места нахождения объекта на картинке. Такое использование сети называется семантической сегментацией.



Пример точной семантической сегментации – то что хочется получить
Однако использовать напрямую тепловую плохо, т.к. она дает неточную сегментацию, особенно на границах объектов

Сверточные нейронные сети

Основные проблемы семантической сегментации при использовании глубоких сверточных нейронных сетей:

- 1.уменьшение пространственного разрешения;**
- 2.существование объекта в различных масштабах;**
- 3.уменьшение локальной точности.**

Первая проблема возникает при использовании последовательных слоев уменьшения разрешения(max-pooling). Как следствие, на выходе сети карта признаков имеет существенно меньшее разрешение, нежели вход. Очевидно, что для задачи семантической сегментации это крайне нежелательно. Один из возможных способов преодоления этой проблемы - это замена всех слоев уменьшения разрешения на расширенные сверточные слои (*atrous convolutions*) с увеличенным коэффициентом расширения.

Сверточные нейронные сети

Вторая проблема обусловлена различными расстояниями от камеры до объекта в процессе съемки. Существуют три основных категории решения данной проблемы:

1. пирамида изображений (image pyramid) - через сеть пропускаются несколько вариантов одного и того же изображения в разных масштабах и далее совмещаются выходы;
2. кодер-декодер архитектура (encoder-decoder) - различные масштабы изображения преобразуются в кодере, а разрешение восстанавливается до исходного с помощью декодера;

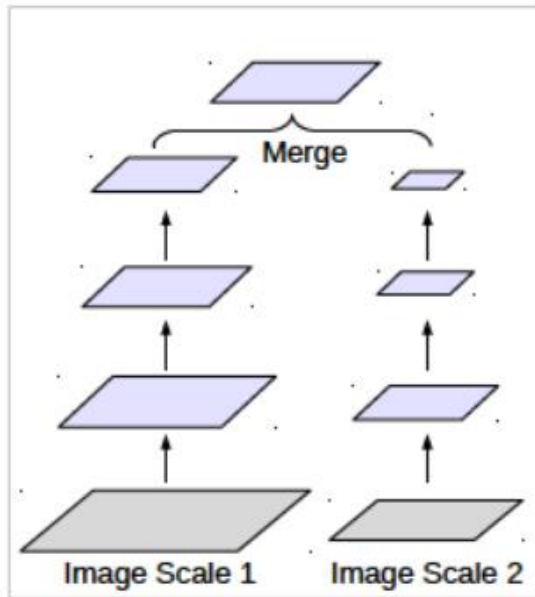
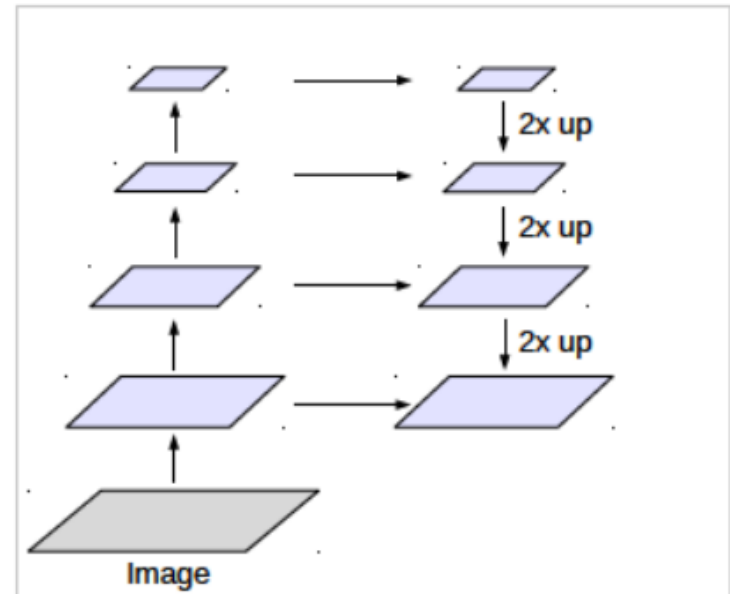


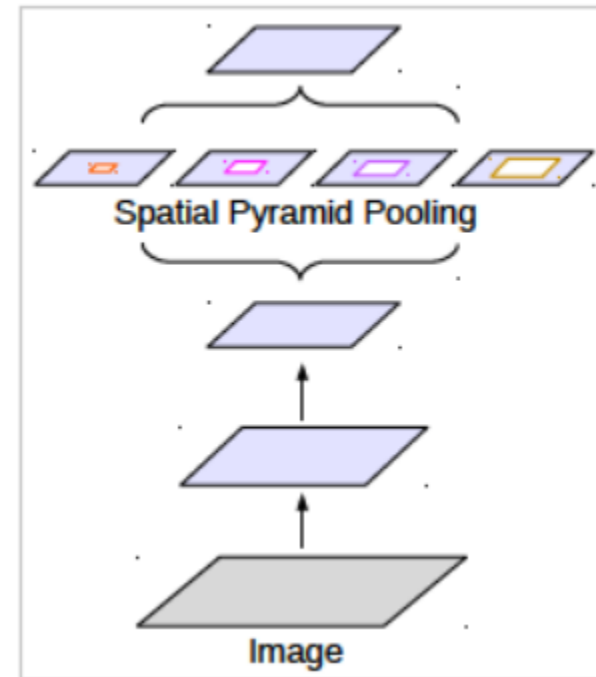
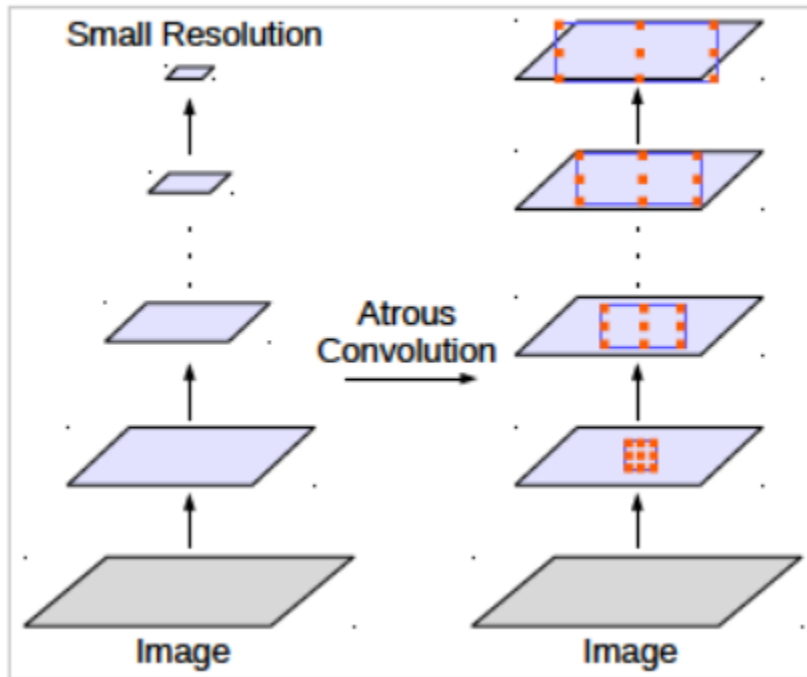
image pyramid



encoder-decoder

Сверточные нейронные сети

3. пространственное объединение в пирамиду (spatial pyramid pooling) - входящий сигнал преобразуется фильтрами с разными степенями расширения и тем самым разными полями зрения.

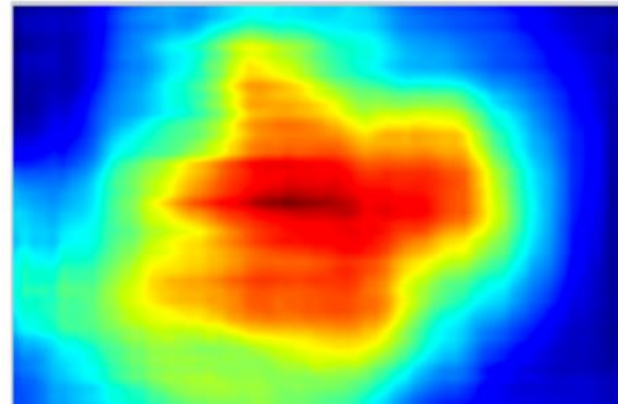


Обычные и расширенные свертки.

Пространственное объединение в пирамиду

Сверточные нейронные сети

Третья проблема обусловлена пространственной инвариантностью СНС. Особенно хорошо это наблюдается на границах сегментируемых объектов

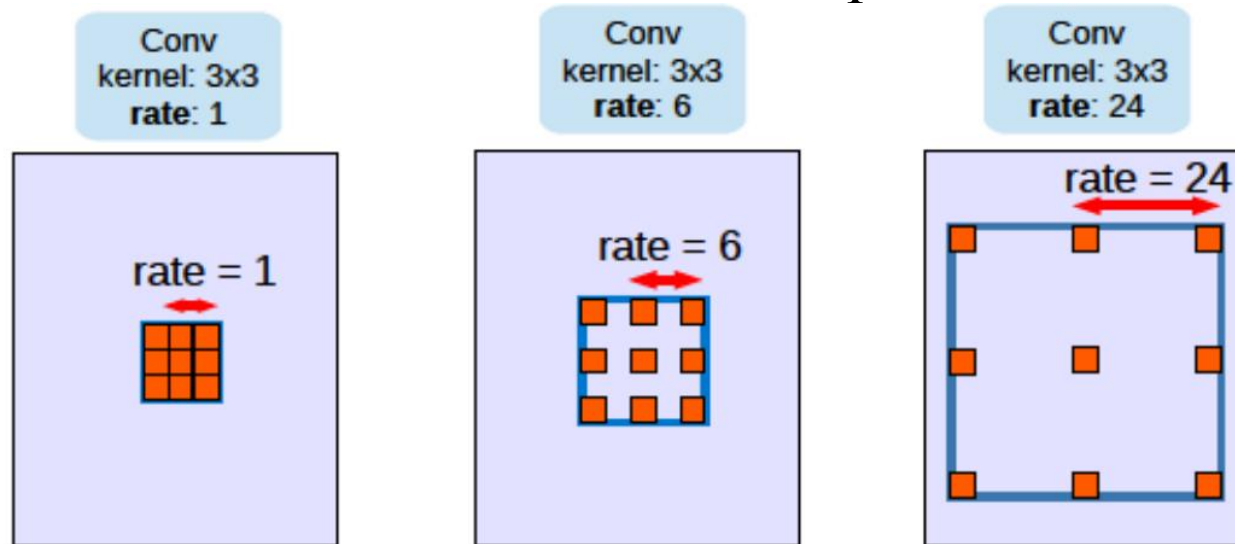


Для ее решения архитектура сети строится по принципу кодер-декодер, где роль кодера - обработка и кодирование контекстной информации на изображении путем уменьшения разрешения, а декодера - восстановление изображения до исходного разрешения и восстановление четких границ объектов.

Сверточные нейронные сети

Расширенные свертки (atrous convolution)

Расширенные сверточные слои (atrous convolutions) - сверточные слои с расширенными фильтрами, в которых в области между ненулевыми значениями фильтра заполняются нулями. Это является обобщением обычного сверточного слоя.



Двумерные расширенные свертки с различной степенью расширения rate.

По сути rate - параметр регулировки между локальной точностью (маленькое поле зрения) и ассимиляцией контекста (большое поле зрения).

Сверточные нейронные сети

Данный тип свертки позволяет увеличивать поле зрения фильтра для захвата информации в разных масштабах и контролировать пространственное разрешение выхода слоя. В сравнении со стандартной сверткой с большими ядрами, расширенные сверточные слои позволяют эффективно увеличить поле зрения без увеличения количества параметров.

Как правило данный тип свертки применяется в каскадном (последовательном) или пирамидном (параллельном) режиме с различными степенями расширения. Это позволяет решить проблему с нахождением объекта в разных масштабах.

Пространственное объединение в пирамиду (Atrous Spatial Pyramid Pooling - ASPP)

Несмотря на то, что сверточные сети и показывают относительную инвариантность к масштабу изображения (достигается тренировкой на наборе данных, содержащем различные масштабы одного и того же объекта), явная обработка различных масштабов может.

Сверточные нейронные сети

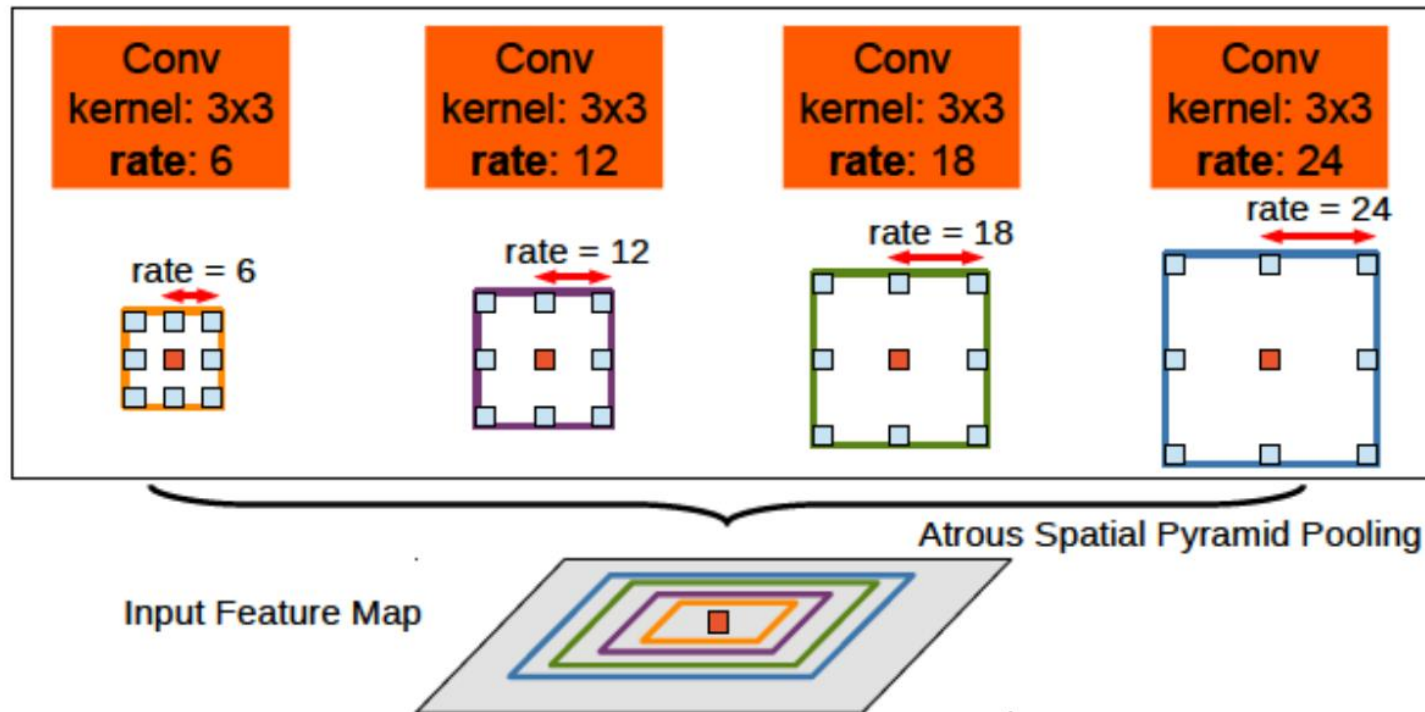
улучшить способность сети обрабатывать большие и маленькие объекты.

Один из возможных методов решения проблемы объекта в разных масштабах — создать несколько масштабов одного и того же изображения, пропустить через сеть и объединить выходы. Практически доказано, что это улучшает качество сегментации. Однако, основным недостатком данного подхода является большое количество вычислений, так как нужно обработать несколько вариантов одного и того же изображения.

Для устойчивой сегментации объектов в разных масштабах применяют расширенное пространственное объединение в пирамиду (Atrous Spatial Pyramid Pooling – ASPP). Оно преобразует входной сигнал фильтрами с разными значениями повышения дискретизации (и тем самым разными областями зрения), таким образом усваивая объекты и контекст изображения различного масштаба

Сверточные нейронные сети

Как правило используются четыре параллельных расширенных сверточных слоя с различными степенями расширения. После этого все выходы объединяются в один.

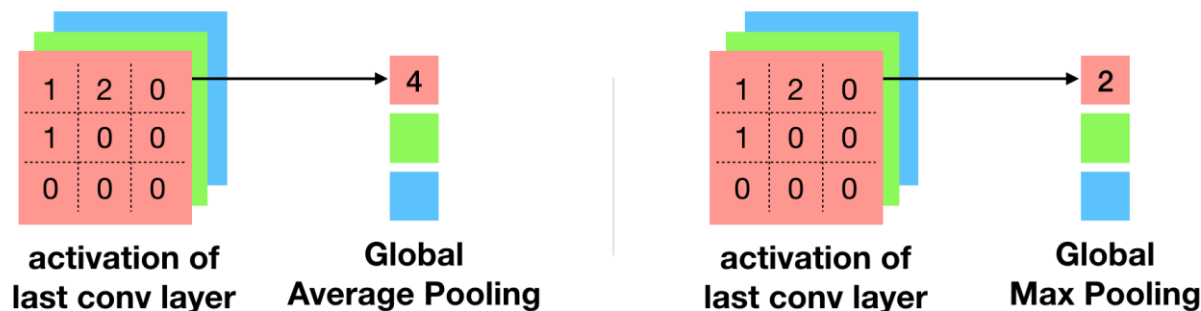


С увеличением степени расширения, количество валидных весов фильтра (т.е. тех, которые перемножаются непосредственно на значения пикселей изображения, а не на достроенные участки) уменьшается. Так, если значение степени расширения станет

Сверточные нейронные сети

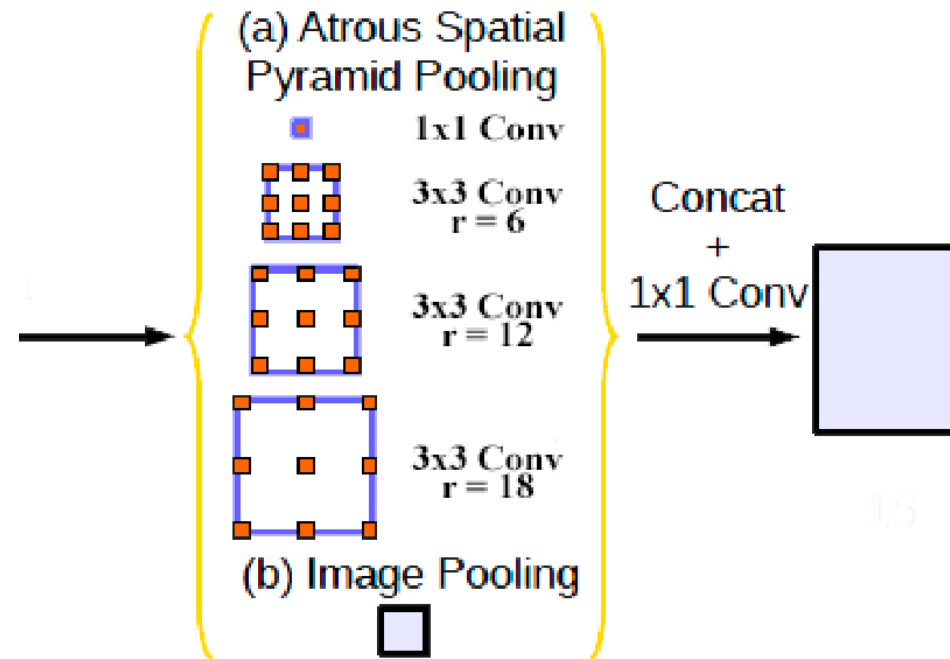
близкой к разрешению карты признаков, то фильтр вырождается в простой фильтр 1×1 , т.к. только центральное значение фильтра будет попадать на исходные пиксели.

Для решения данной проблемы и усвоения глобального контекста изображения выделяются дополнительные признаки на уровне изображения (*image level features*). Процесс извлечения дополнительных признаков состоит из следующих этапов: сначала применяется глобальный усредненный пулинг (*global average pooling*) к последнему сверточному слою. При этом в каждой карте признаков просто ищется либо сумма всех признаков либо их среднее значение.



Сверточные нейронные сети

Затем производится свертка 1×1 свертка с 256 фильтрами и билинейная интерполяция до требуемого пространственного разрешения.



Таким образом модуль расширенного пространственного объединения в пирамиду состоит из следующих частей:

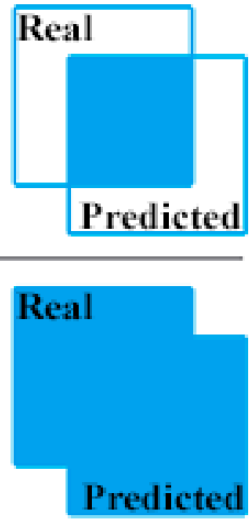
1. расширенная свертка 1×1 , количество фильтров 256, $r = 1$;
2. расширенная свертка 3×3 , количество фильтров 256, $r = 6$;

Сверточные нейронные сети

- 3.расширенная свертка 3x3, количество фильтров 256, r = 12;
- 4.расширенная свертка 3x3, количество фильтров 256, r = 18;
- 5.дополнительные признаки на уровне изображения.

Выходные признаки всех пяти веток объединяются и проходят через свертку 1x1 с 256 фильтрами.

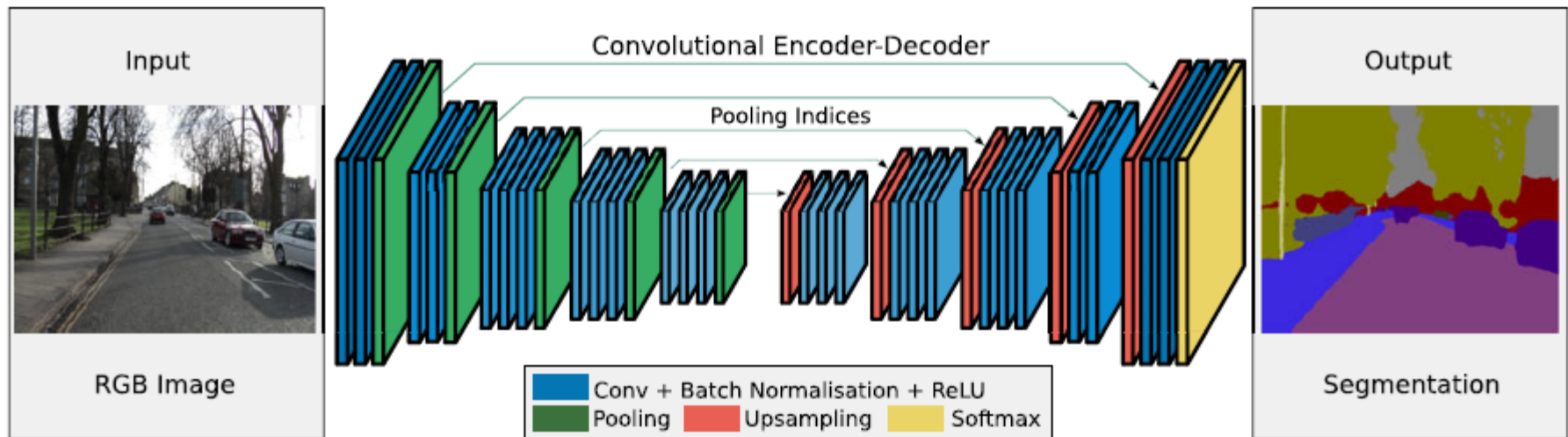
Качество семантической сегментации оценивается по критерию IoU (Intersection over Union – IoU). Он вычисляется как отношение площади пересечения реального объекта и предсказанного к их объединению

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Сверточные нейронные сети

Encoder-decoder архитектура

Архитектура SegNet для сегментации изображений



- Нейронная сеть стоит из двух частей — encoder (“сверточная”) и decoder (“разверточная”). Encoder преобразует входное изображение в многомерное представление признаков. Он выполняет функцию извлечения признаков. Decoder создает сегментированное изображение на основе признаков, полученных от сверточной сети. Последний слой сети - классификатора soft-max, который оценивает вероятности принадлежности одному из классов объектов каждого пикселя.

Сверточные нейронные сети

Первая часть сети состоит из пяти сверточных слоев с наборами фильтров (например, 256, 128, 64, 64, 64). Осуществляется нормализация данных. В качестве активационных функций используется «выпрямитель» (ReLU, rectified linear unit). Далее следуют слои субдискретизации (pooling) (обычно, с окном 2×2). Проходя через этот слой, изображение уменьшается в два раза.

Вторая часть сети является зеркальным отражением первой. Размер изображения необходимо восстановить до исходного и сформировать карту вероятностей, основываясь на признаках входного изображения. С этой целью применяются слои повышающей дискретизации (upsampling-слои) в комбинации со сверточными слоями. Например, представленная нейронная сеть имеет 13 слоев.

Сверточные нейронные сети

Пространственное объединение в пирамиду (ASPP), так же, как и кодер-декодер архитектура, используется в глубоких нейронных сетях для задач семантической сегментации. Первые способны кодировать контекстную информацию в разных масштабах путем преобразования входных признаков через фильтры или слои уменьшения разрешения с разными параметрами. Вторые способны обрабатывать четкие границы объектов путем последовательного восстановления пространственной информации.

Реализация сверток

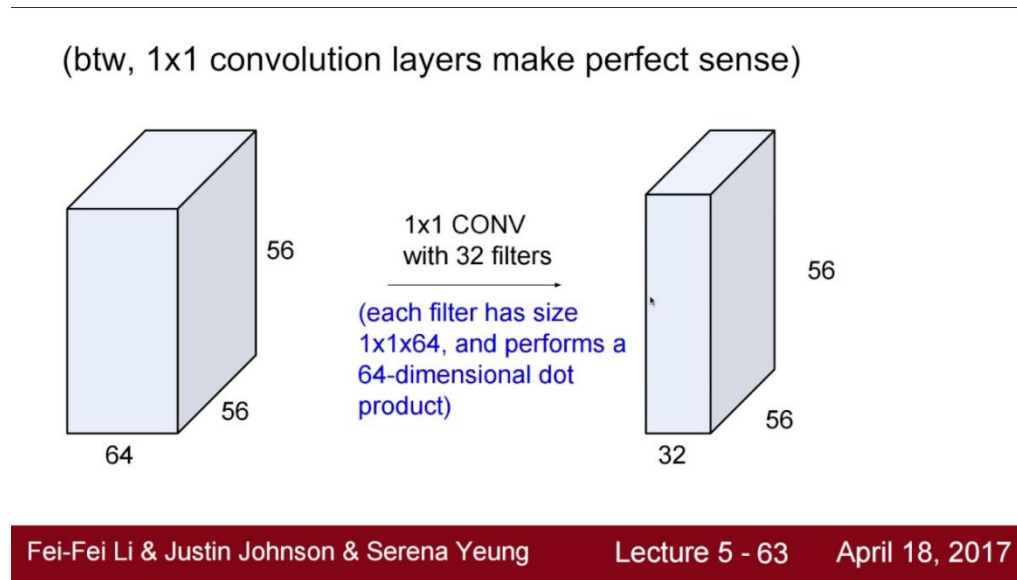
Что такое свертка 1×1 ?

Свертка 1×1 просто отображает входной пиксель со всеми его каналами на выходной пиксель, не глядя ни на что вокруг себя. Она часто используется для уменьшения количества каналов глубины. Фактически это скалярное произведение тензора на вектор размерность которого равна глубине тензора.

Сверточные нейронные сети

Например если на входе имеем $[32 \times 32 \times 6]$, то свёртка 1×1 будут эффективно выполнять 6-мерное скалярное произведение (поскольку входная глубина состоит из 6 каналов). В результате получим одну карту признаков $[32 \times 32 \times 1]$.

Если хотим уменьшить глубину до N каналов то надо сделать N сверток с различными векторами (фильтрами)



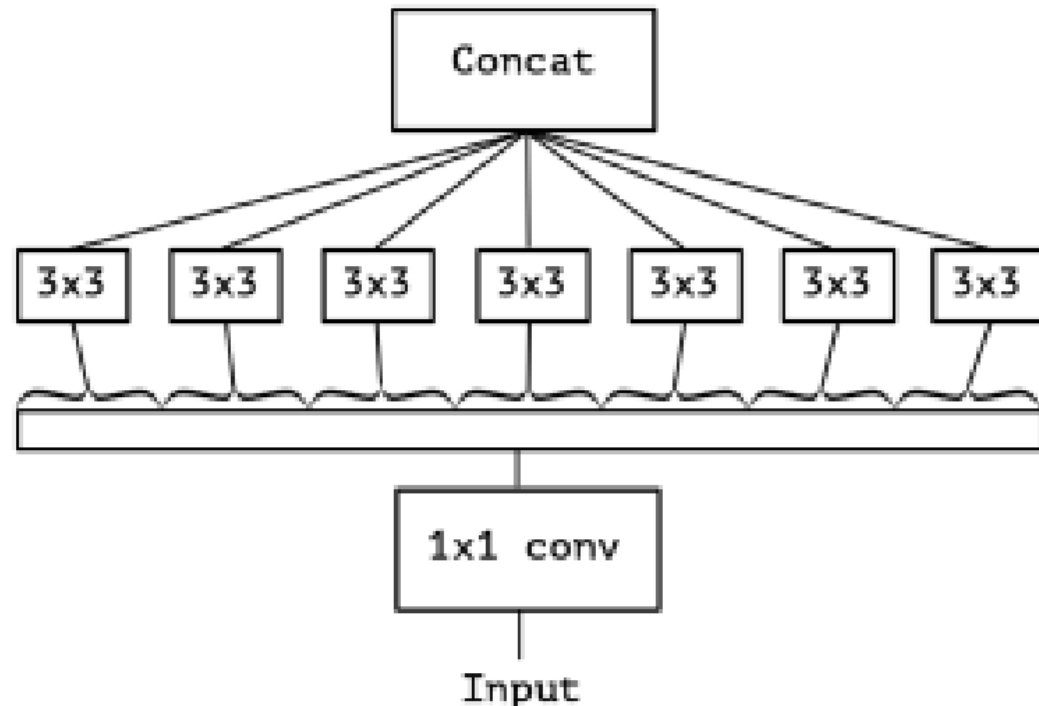
Сверточные нейронные сети

В последнее время наблюдается тенденция увеличения количества слоев для улучшения качества работы сети. Однако, ограничение в производительности вычислительных машин попросту не дают обучать и использовать такие архитектуры. Одним из возможных способов решения данной проблемы является поканальная раздельная свертка (*depthwise separable convolution*).

Классический сверточный слой обрабатывает как межканальную информацию (свёртка применяется ко всем каналам сразу), так и пространственную информацию (взаимосвязь соседних точек внутри одного канала). Предположение о том, что эти два вида информации можно обрабатывать последовательно без потери качества работы сети и легло в основу поканальной раздельной свертки

Сверточные нейронные сети

Таким образом, стандартная свертка раскладывается на поканальную свертку (*pointwise convolution*), которая обрабатывает только межканальную корреляцию, и пространственную свертку (*spatial convolution*), которая обрабатывает только пространственную корреляцию в рамках одного канала.



Сверточные нейронные сети

Два этапа поканальной раздельной свертки:

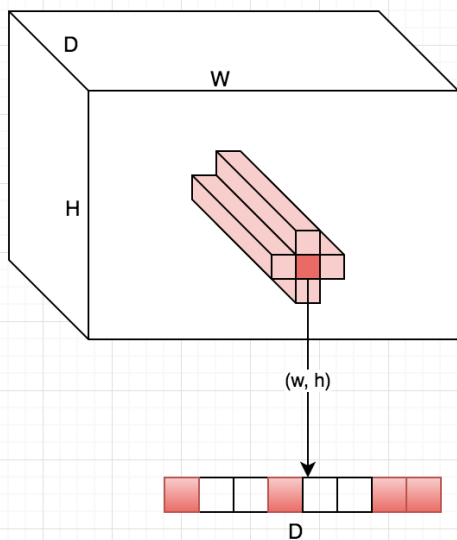
1. входящий тензор сворачивается ядром $1 \times 1 \times n_c$
 n_c — количество каналов входящего тензора;
2. в получившемся результате каждый канал сворачивается по отдельности сверткой $3 \times 3 \times 1$

На самом деле сначала делается свертка 3×3 по каждому каналу, а уже затем 1×1 . Практически доказано, что порядок этих операций не влияет на конечный результат.

При этом исследования показали, что можно построить оптимальную нейронную сеть для данного датасета, анализируя корреляции нейронов предыдущего слоя и объединяя коррелированные нейроны в группы, которые будут нейронами следующего слоя. Таким образом, нейроны поздних слоев «смотрят» на сильно пересекающиеся области исходного изображения.

Сверточные нейронные сети

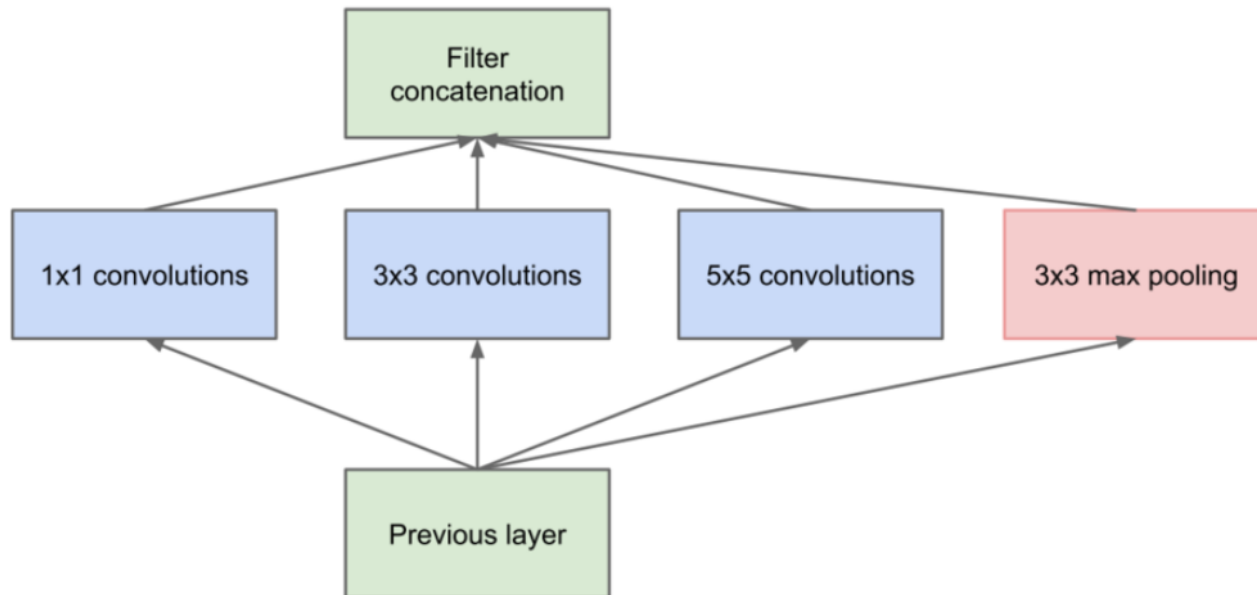
На ранних слоях (ближе к входу) коррелированные нейроны будут концентрироваться в локальных областях. Это значит, что если несколько нейронов в одной координате (w, h) во всех картах признаков могут выучить примерно одно и то же, то в тензоре после первого слоя их активации будут располагаться на небольшом регионе в районе некоторой точки (w, h) , а распределены будут вдоль размерности банка фильтров — D .



Как поймать эти корреляции и превратить их в один признак? На помощь приходит идея сверток 1×1 . Продолжая эту идею, можно предположить, что чуть меньшее количество коррелированных кластеров будет чуть большего размера, например 3×3 . То же самое справедливо для 5×5 и т. д.

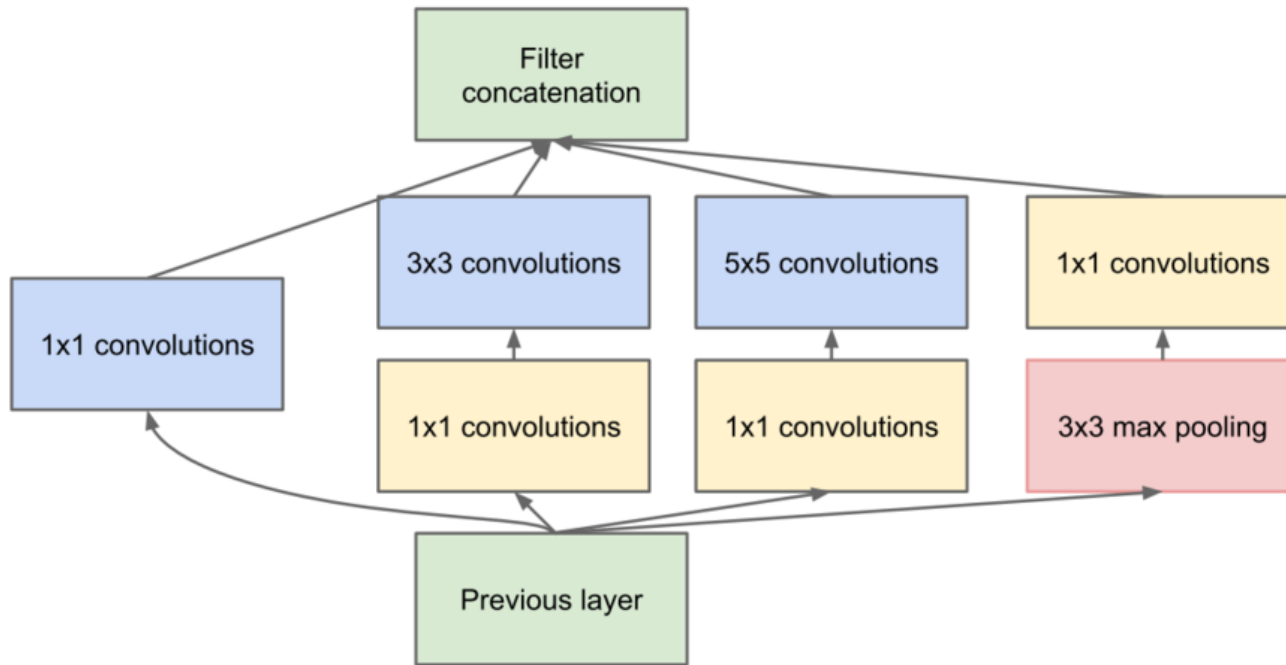
Сверточные нейронные сети

После вычисления карт признаков для каждой свертки их необходимо каким-либо способом агрегировать, например, конкатенировать фильтры, чтобы не терять оригинальный образ из предыдущего слоя, помимо свертки добавляются еще пулинг-признаки.



Блок сети

Сверточные нейронные сети



Блок сети с поканальной раздельной сверткой

Сверточные нейронные сети

Пусть у нас 256 свойств на входе и 256 на выходе, и пусть выполняется только свёртки 3×3 . Получаем $256 \times 256 \times 3 \times 3$ свёрток (589 000 операций умножения с накоплением). Уменьшим количество свойств для свёртывания до 64 ($256/4$). В этом случае сначала выполним свёртку 1×1 $256 \rightarrow 64$, затем ещё 64 свёрток во всех ветках, а потом снова применим свёртку 1×1 из 64 $\rightarrow 256$ свойств. Количество операций:

$$256 \times 64 \times 1 \times 1 = 16\,000$$

$$64 \times 64 \times 3 \times 3 = 36\,000$$

$$64 \times 256 \times 1 \times 1 = 16\,000$$

Всего около 70 000, снизили количество операций почти в 10 раз!

Но при этом мы не потеряли обобщённости в этом слое.

Причина успеха в том, что входные свойства коррелированы, а значит можно избавиться от избыточности, правильно комбинируя свойства со свёртками 1×1 . А после свёртывания с меньшим количеством свойств можно на следующем слое опять развернуть их в значимую комбинацию.

Сверточные нейронные сети

Современные архитектуры

Рассмотренные идеи сверток легли в основу архитектур Inception-v1. На ее основе были построены Inception-v3, Inception-v4 и др.

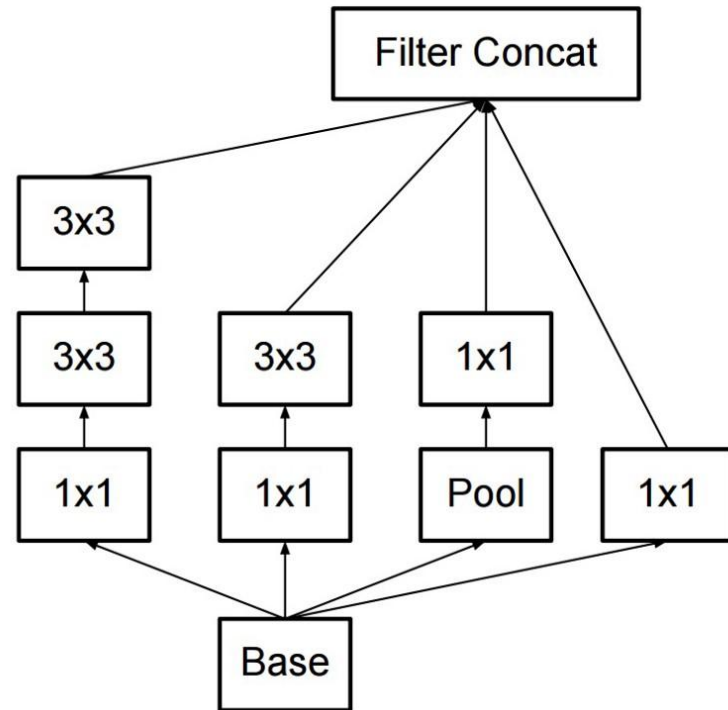
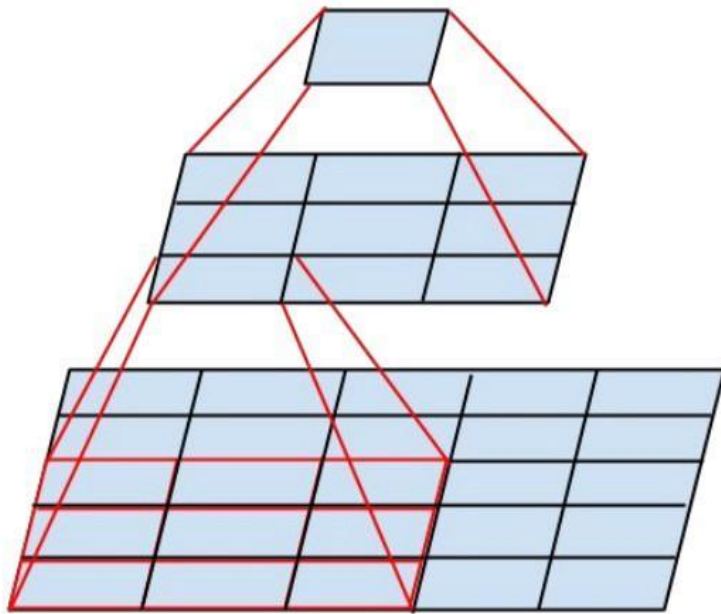
Inception-v3 (2015)

В феврале 2015-го в качестве второй версии Inception была представлена архитектура Batch-normalized Inception. Пакетная нормализация (batch-normalization) вычисляет среднее и среднеквадратичное отклонение всех карт распределения свойств в выходном слое, и нормализует их отклики с этими значениями. Это соответствует «отбеливанию» данных, то есть отклики всех нейронных карт лежат в одном диапазоне и с нулевым средним. Такой подход облегчает обучение, потому что последующий слой не обязан запоминать смещения (offsets) входных данных и может заниматься только поиском лучших комбинаций свойств.

Сверточные нейронные сети

Сеть Inception v3

1. По мере возможности используются только свёртки 3×3 . Учитывая, что фильтры 5×5 и 7×7 можно декомпозировать с помощью нескольких 3×3 . Тогда новый Inception-модуль выглядит так:

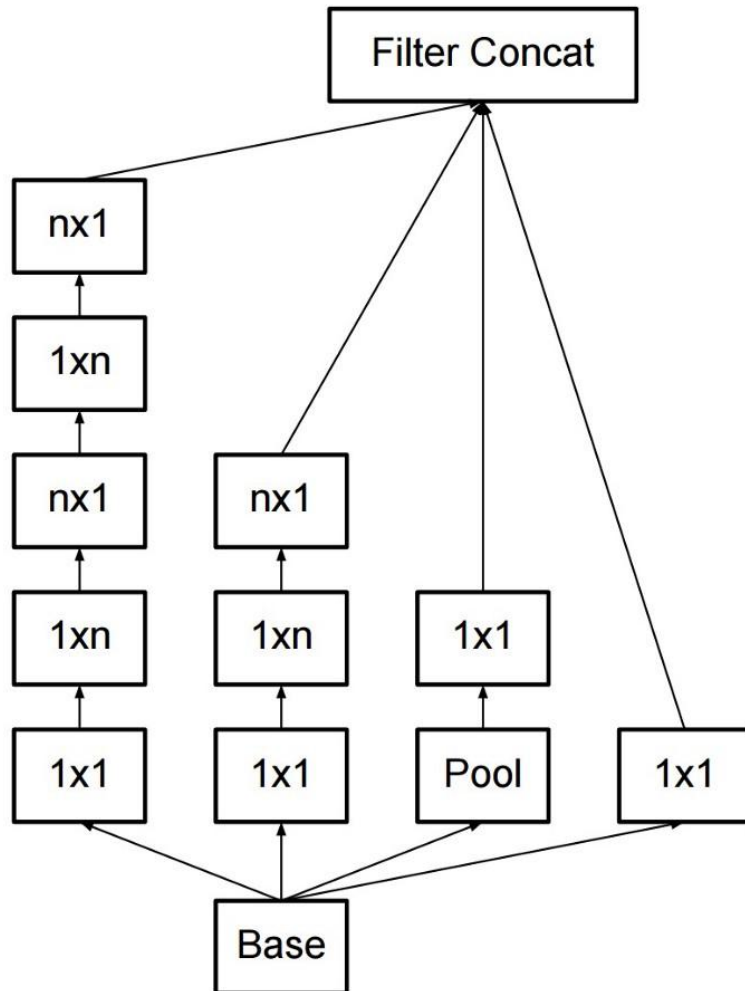


Переход от сверток 5×5 к двум сверткам 3×3 позволили увеличить скорость вычислений. Стоимость свертки 5×5 в 2.78 раза выше, чем свертки 3×3 . Т.е. использование сверток 3×3 увеличивает производительность.

Сверточные нейронные сети

Сеть Inception v3

2. Фильтры также можно декомпозировать с помощью сглаженных свёрток в более сложные модули:

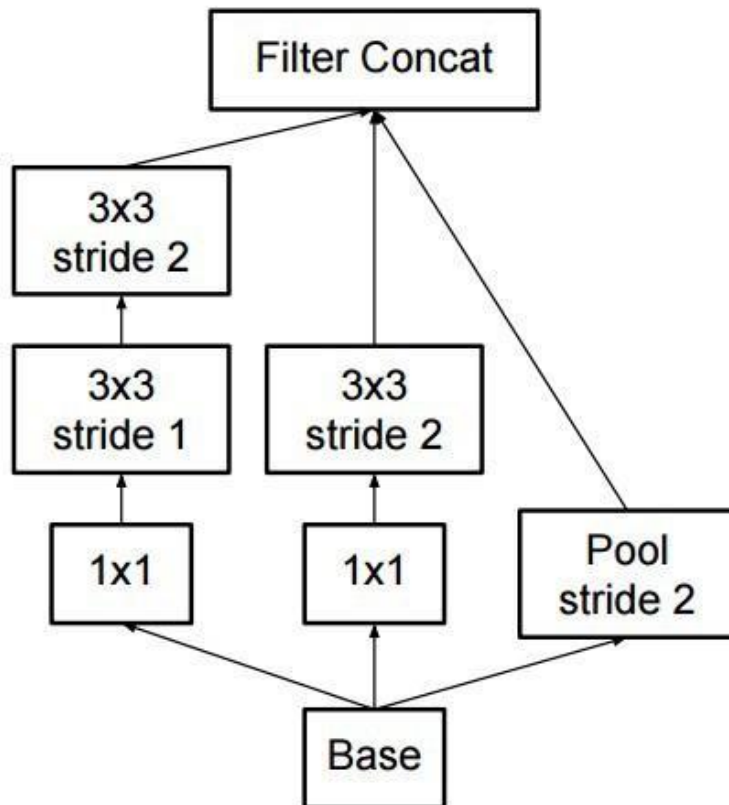


Свертки размера $n \times n$ представляются в виде комбинации сверток $1 \times n$ и $n \times 1$. На пример, результат свертки 3×3 эквивалентен последовательности сверток 1×3 и 3×1 . Выполнить такую последовательность сверток на **33%** дешевле, чем одну свертку 3×3 convolution.

Сверточные нейронные сети

Сеть Inception v3

3. Inception-модули могут с помощью pooling-а уменьшать размер данных. Это аналогично выполнению свёртки со страйдами параллельно с простым pooling-слоем:

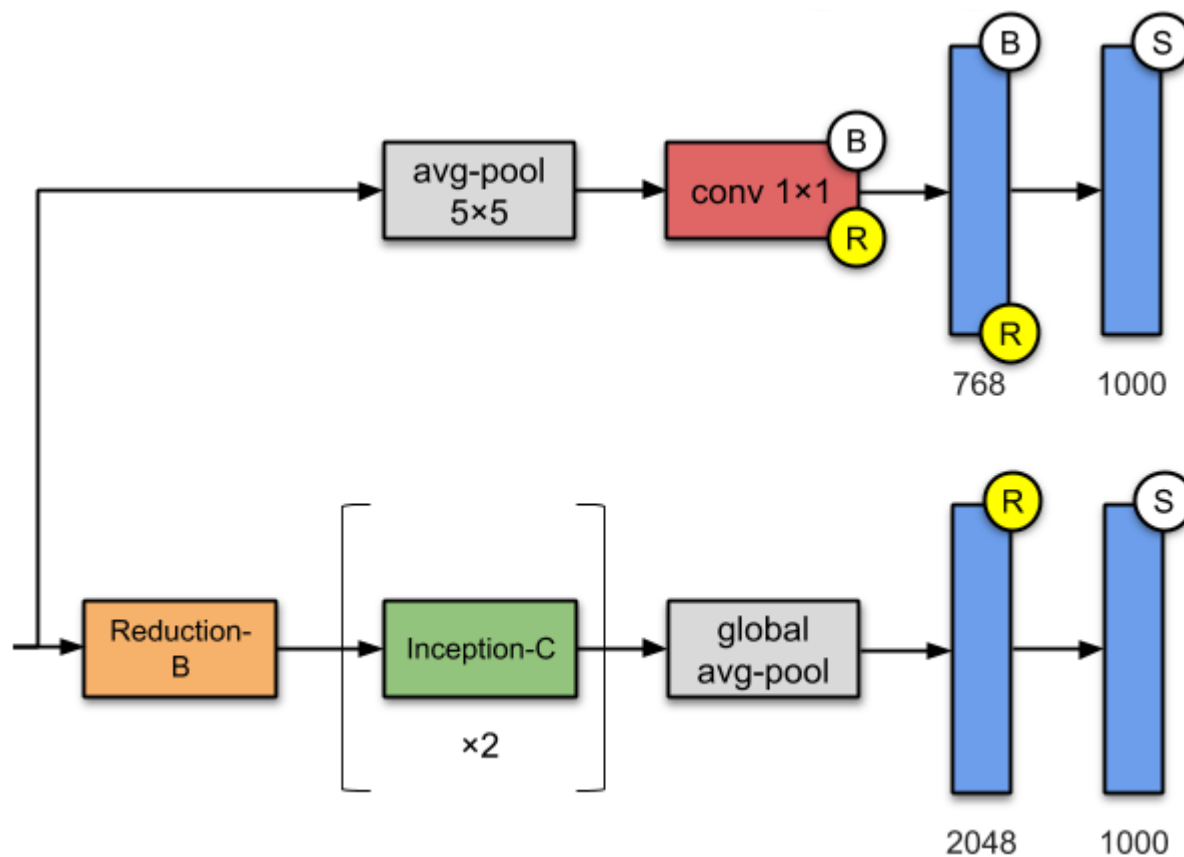
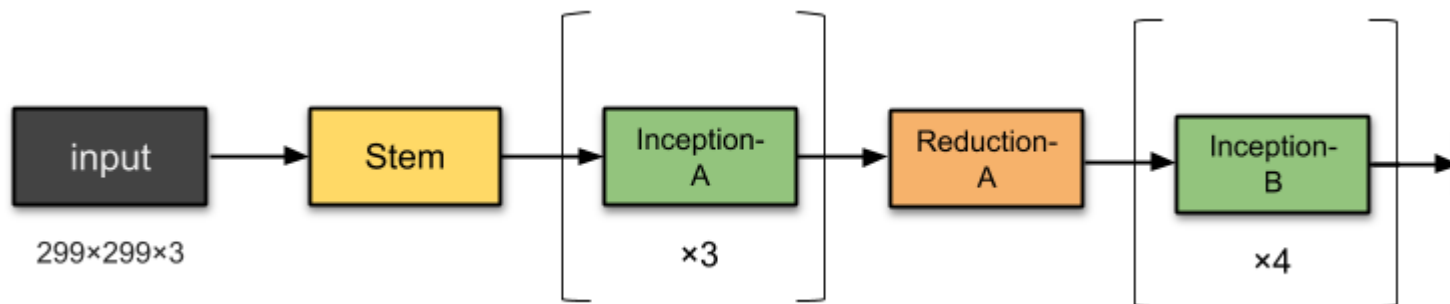


Банки фильтров были расширены (стали шире, но не глубже), что позволило избежать representational bottleneck. Если была бы увеличена глубина, возникло бы избыточное снижение размерности, что привело бы к потере информации.

В качестве финального классификатора Inception использует pooling-слой с softmax.

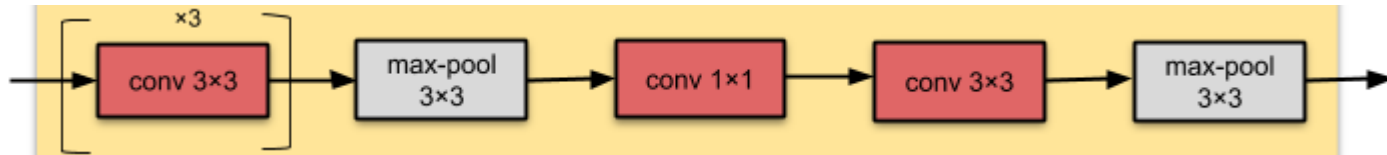
Сверточные нейронные сети

Архитектура Inception v3

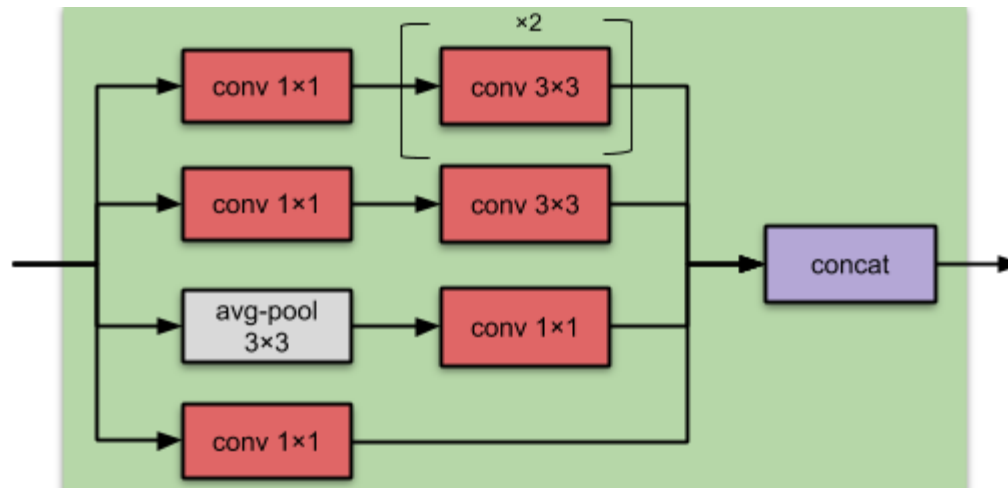


Сверточные нейронные сети

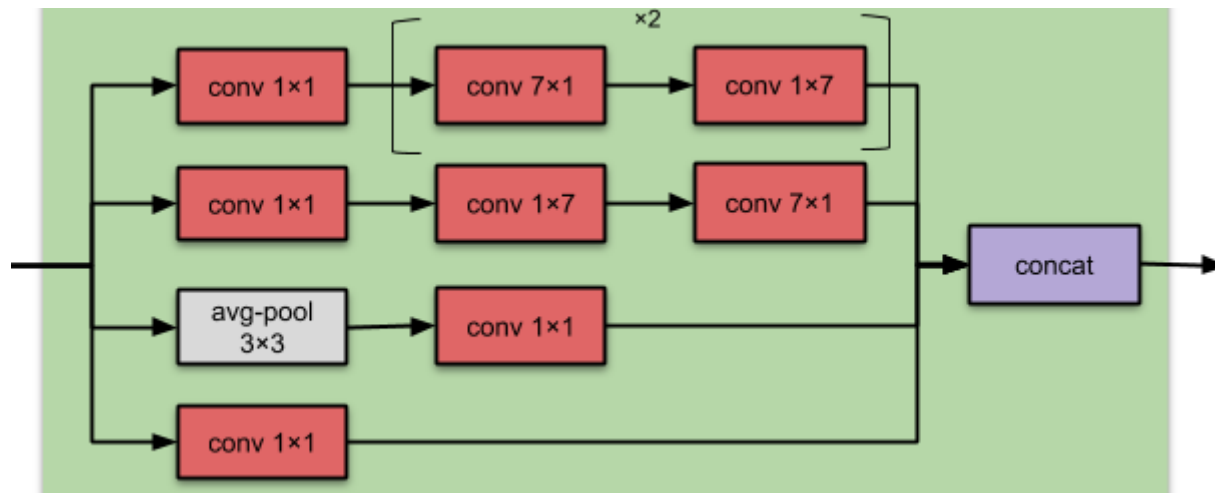
Stem



Inception-A

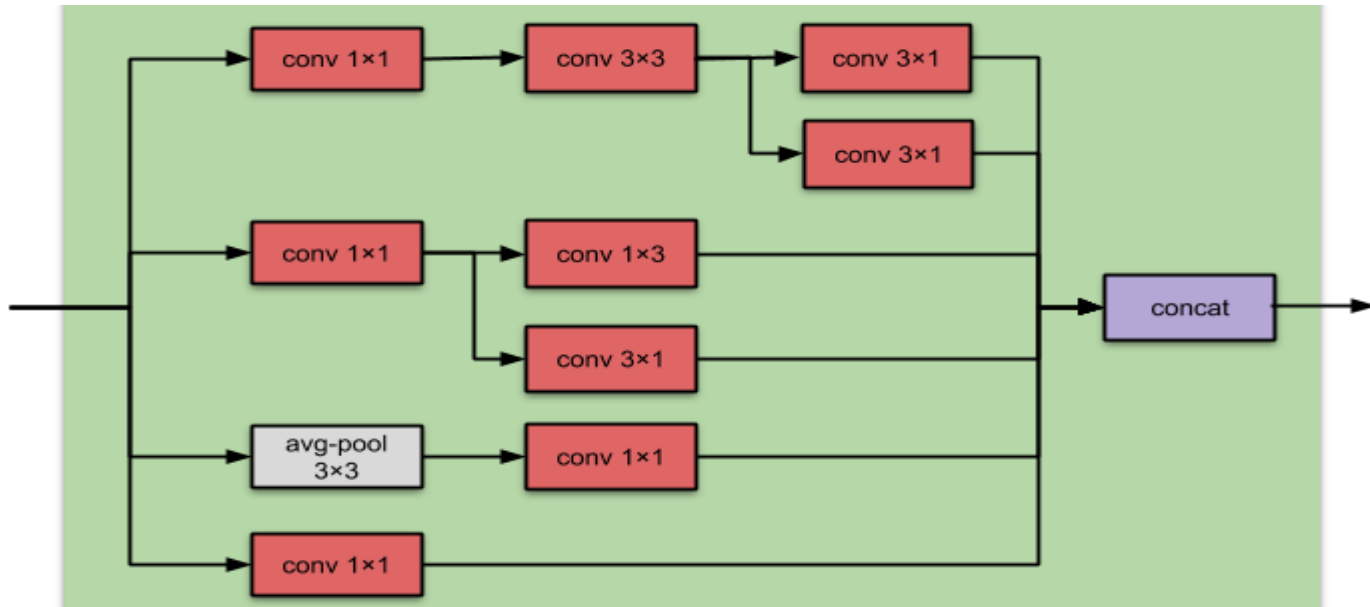


Inception-B

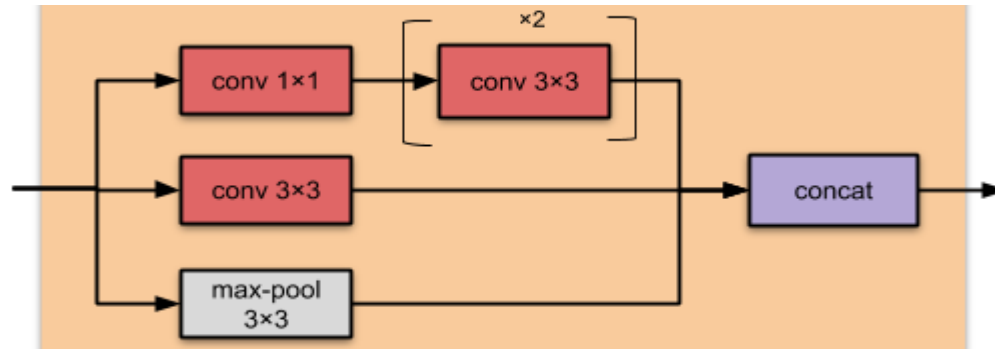


Сверточные нейронные сети

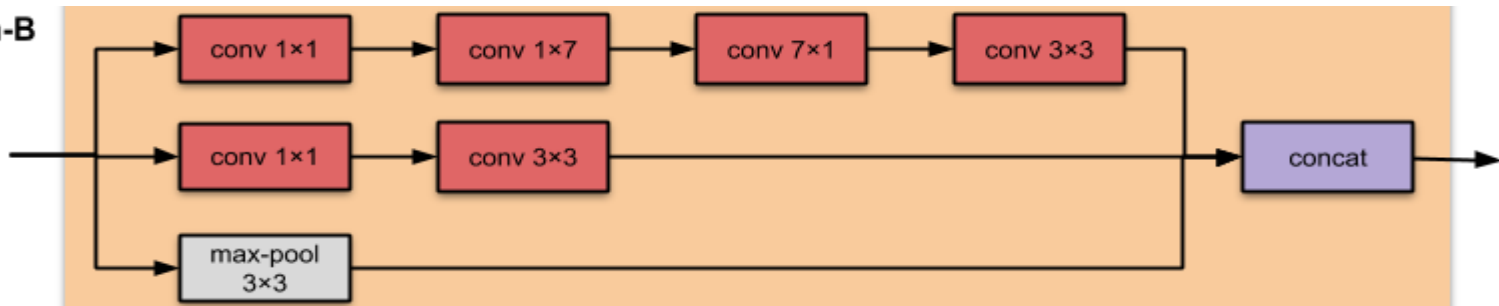
Inception-C



Reduction-A



Reduction-B



Сверточные нейронные сети

Сеть Resnet

В декабре 2015-го, примерно в то же время, как была представлена архитектура Inception v3, произошла революция — опубликовали ResNet. В ней заложены простые идеи: подаём выходные данные двух успешных свёрточных слоёв и обходим входные данные для следующего слоя.

Ранее было замечено, что если просто добавлять больше слоев, то качество такой модели растёт до некоторого предела, а затем начинает падать. Эту проблему называют *degradation problem*. Однако нашли такую топологию, при которой качество модели растёт при добавлении новых слоев.

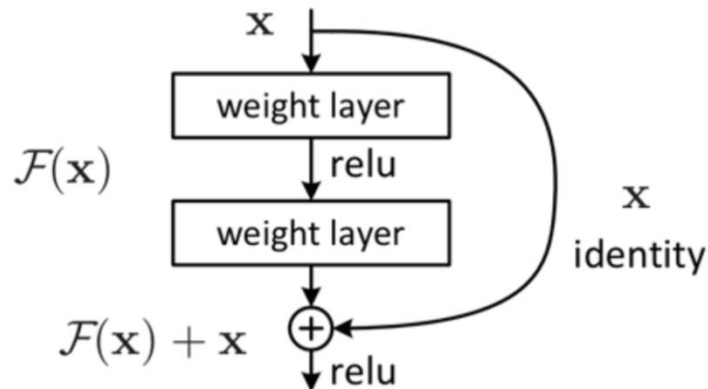
Сверточные нейронные сети

Известно, что нейросеть может аппроксимировать почти любую функцию, например некоторую сложную функцию $H(x)$. Тогда справедливо, что такая сеть легко выучит residual-функцию (остаточная функция): $F(x) = H(x) - x$. Очевидно, что наша первоначальная целевая функция будет равна $H(x) = F(x) + x$. Если мы возьмем некоторую сеть, и добавим к ней еще слоев двадцать, то нам хотелось бы, чтобы глубокая сеть вела себя как минимум не хуже своего неглубокого аналога.

Проблема деградации подразумевает, что сложная нелинейная функция, полученная добавлением нескольких слоев, должна выучить тождественное преобразование, в случае если на предыдущих слоях был достигнут предел качества. Но этого не происходит по каким-то причинам, возможно, оптимизатор просто не справляется с тем, чтобы настроить веса так, чтобы сложная нелинейная иерархическая модель делала тождественное преобразование.

Сверточные нейронные сети

Возможно добавить shortcut-соединение. Тогда оптимизатору будет легче сделать все веса близкими к нулю, нежели создавать тождественное преобразование.



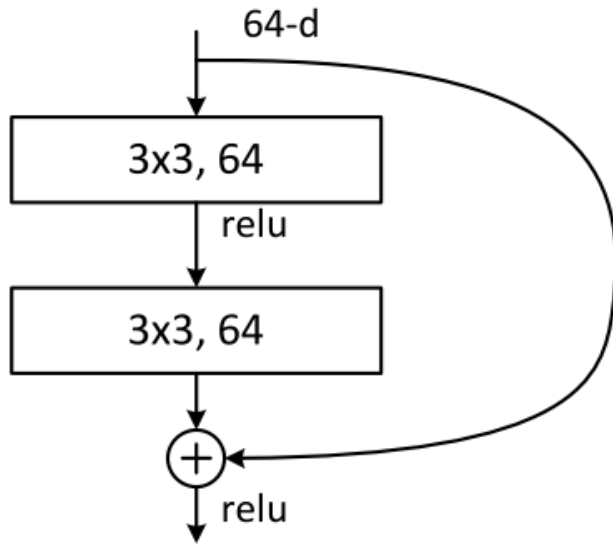
$\mathcal{H}(x)$ is the true function we want to learn

Let's pretend we want to learn

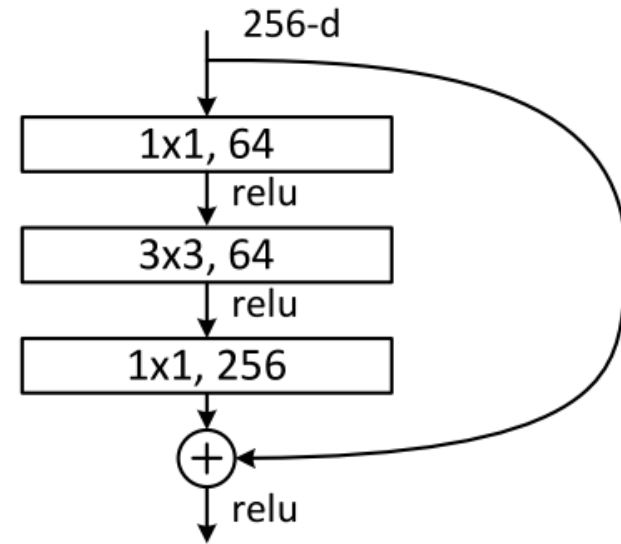
$$\mathcal{F}(x) := \mathcal{H}(x) - x$$

Другое объяснение этого приема – при проведении сверток на большом числе слоев многие значения в картах признаков становятся близкими к нулю и следовательно бесполезно присоединять дальше новые сверточные слои. Короткий проход не позволяет сделаться нулями большинство признаков.

Сверточные нейронные сети



Блок сети ResNet-34



Блок сети ResNet-50/101/152

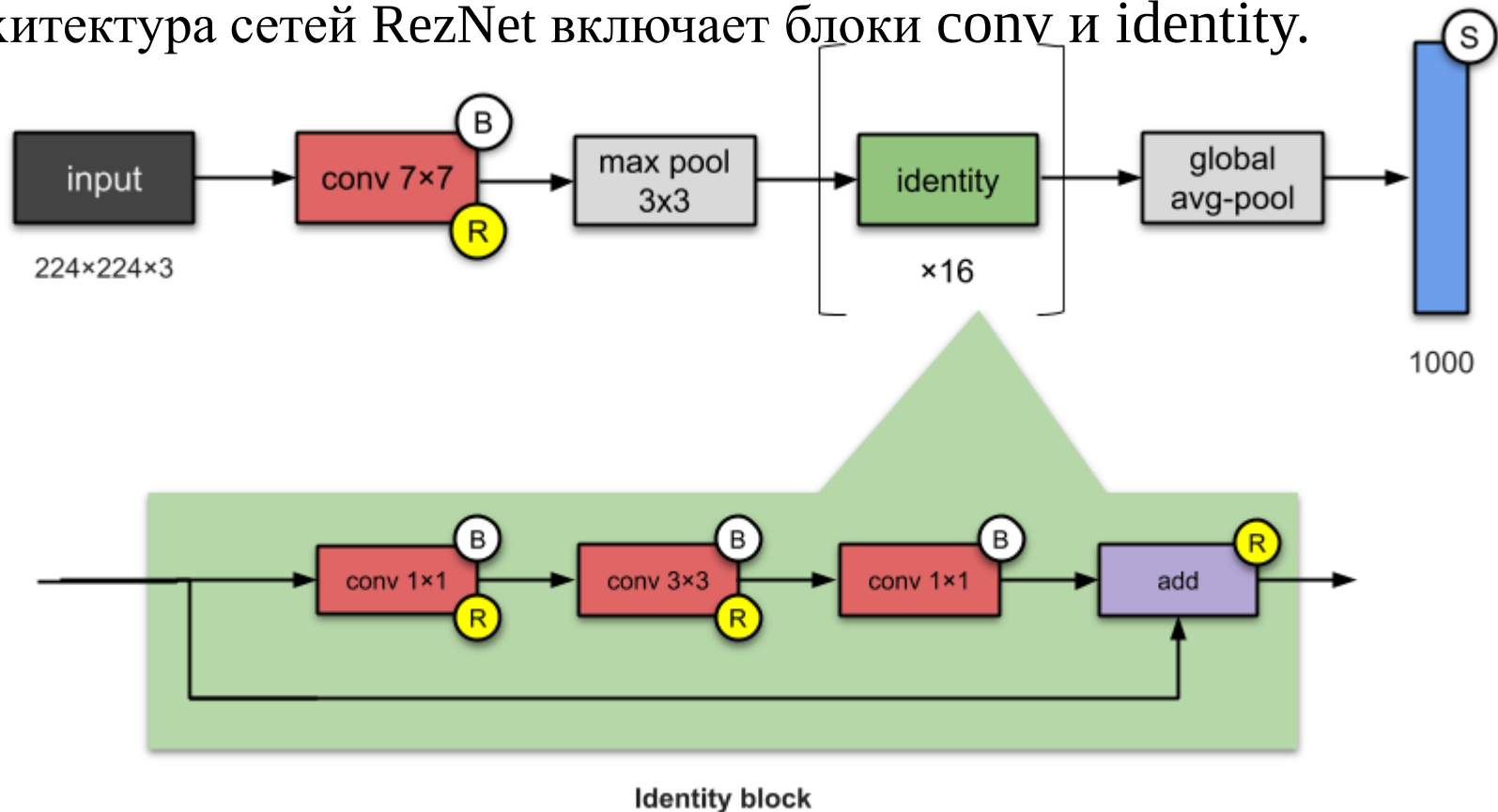
В левом блоке используется понижение глубины до 64, затем свертка для всех 64 карт и восстановление до первоначальной глубины.

В качестве финального классификатора в ResNet используется pooling-слой с softmax.

Сверточные нейронные сети

ResNet – одна из первых архитектур, где используется batch normalisation (пакетная нормализация — метод, который позволяет повысить производительность и стабилизировать работу искусственных нейронных сетей за счет того, что некоторым слоям нейронной сети на вход подаются данные, предварительно обработанные и имеющие нулевое математическое ожидание и единичную дисперсию).

Архитектура сетей ResNet включает блоки conv и identity.



Сверточные нейронные сети

Архитектура ResNet:

- ResNet можно рассматривать как систему одновременно параллельных и последовательных модулей: во многих модулях in-out-сигнал приходит параллельно, а выходные сигналы каждого модуля соединяются последовательно.
- ResNet можно рассматривать как несколько ансамблей параллельных или последовательных модулей.
- Выяснилось, что ResNet обычно оперирует блоками относительно небольшой глубины в 20-30 слоёв, работающих параллельно, а не прогоняя последовательно по всей длине сети.
- Поскольку выходной сигнал возвращается назад и подаётся как входной, как это делается в RNN, ResNet можно считать улучшенной правдоподобной моделью коры головного мозга.

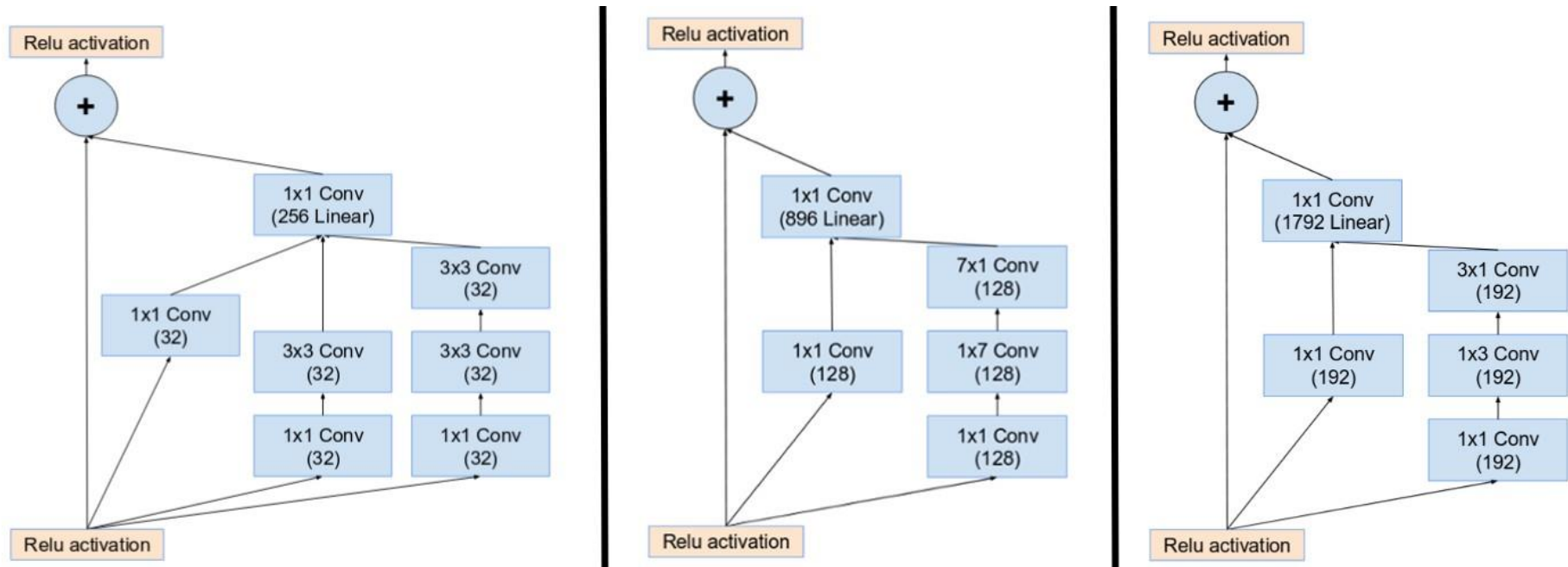
Сверточные нейронные сети

Inception-ResNet v1 and v2

Идея – рассмотреть residual (остаточные) connections, которые добавляют выход результатов свертки модуля Inception “ко входу”. Residual connection — это способ проброса градиентов в глубину, напрямую, когда мы «добавляем ко входу», то есть вход остается неизменным, а наш смысловой блок просто делает какую-то добавку.

Для использования residual связей, вход и выход после свертки должны иметь одинаковые размерности. Для этого используется свертка 1x1 после основной свертки, это позволяет согласовать размерности глубин (Глубина увеличивается после свертки).

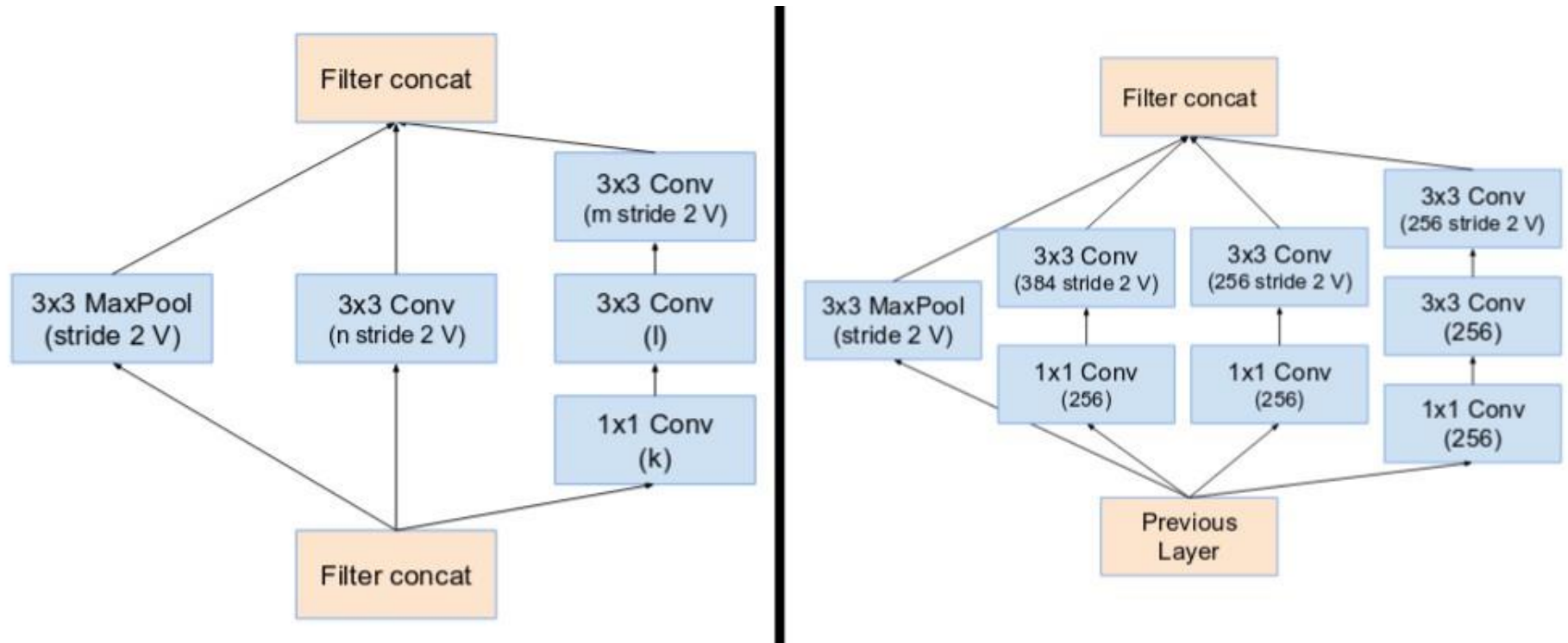
Сверточные нейронные сети



(Слева направо) Inception модули A,B,C в Inception ResNet. Слой пулинга заменен residual связью, также добавлена свертка 1×1 перед сложением.

Сверточные нейронные сети

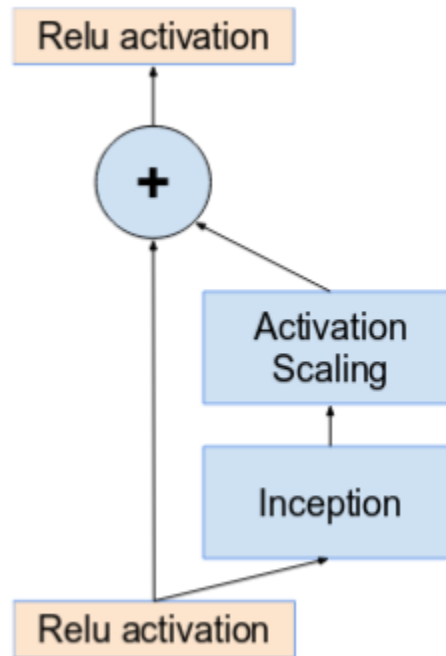
Операции пулинга реализуются в Reduction модулях.



(Слева направо) Reduction модуль А (сокращение размерности с 35x35 до 17x17) и Reduction модуль В (сокращение размерности с 17x17 до 8x8)

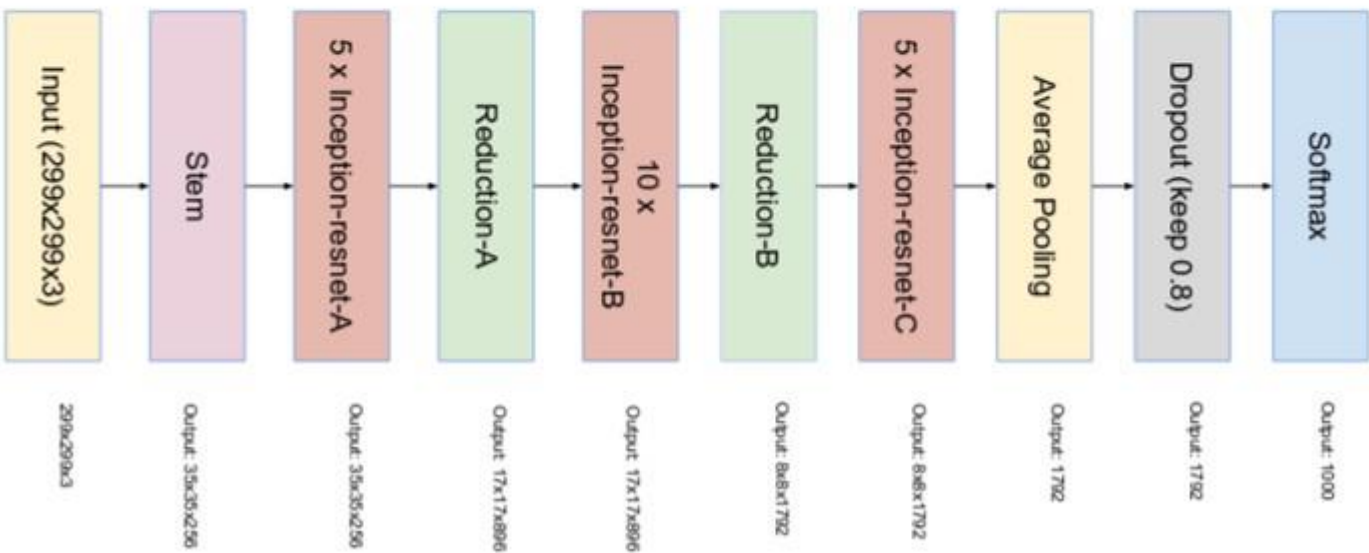
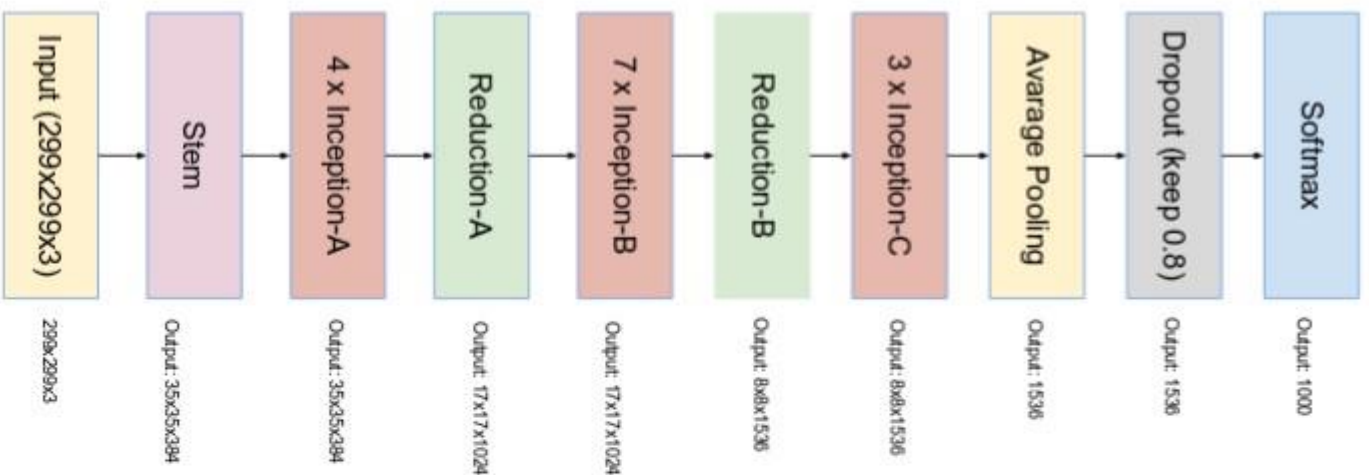
Сверточные нейронные сети

Residual модули с более глубокими архитектурами приводят к “умиранию” сети, если число фильтров превышает 1000. Таким образом, для увеличения стабильности осуществляется масштабирование с коэффициентом от 0.1 до 0.3.



Сверточные нейронные сети

Архитектура Inception-ResNet имеет вид:



Сверточные нейронные сети

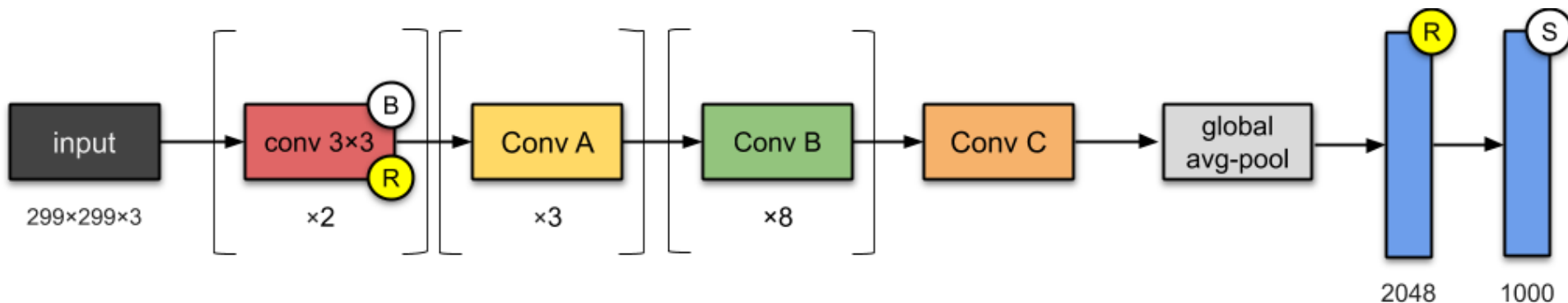
Xception (2016)

Xception построена на основе Inception. Inception модули заменены depthwise separable convolutions.

Xception “экстремально” развивает идеи Inception (этим определяется название):

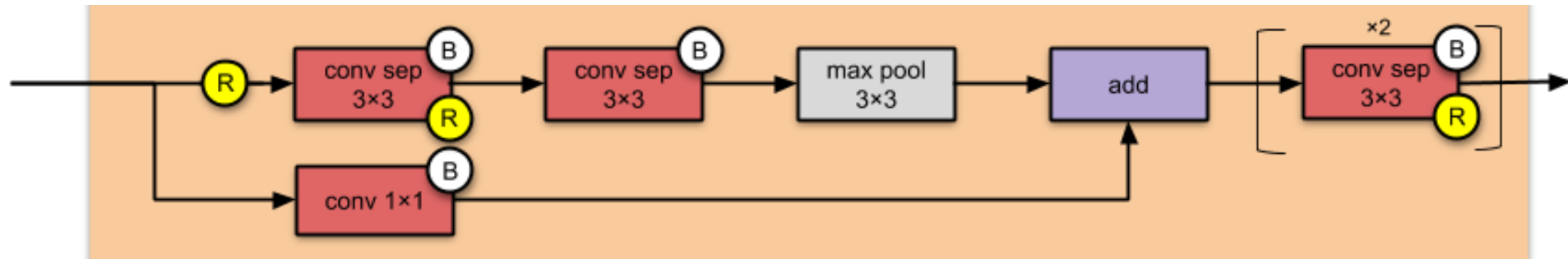
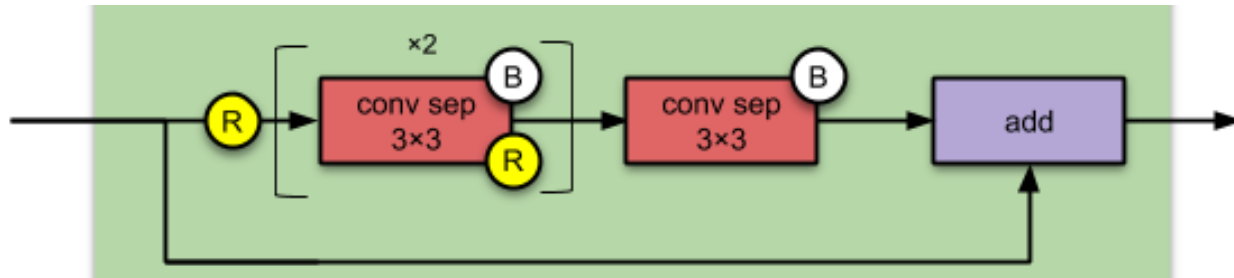
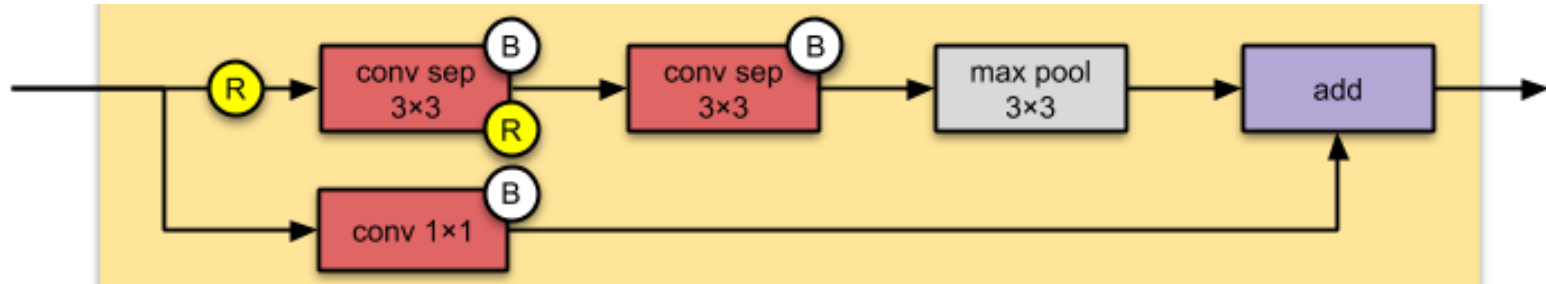
- свертка 1×1 выполняется по каждому каналу;
- для каждого выхода выполняется свертка 3×3 .

Архитектура сети имеет вид:



Сверточные нейронные сети

Xception

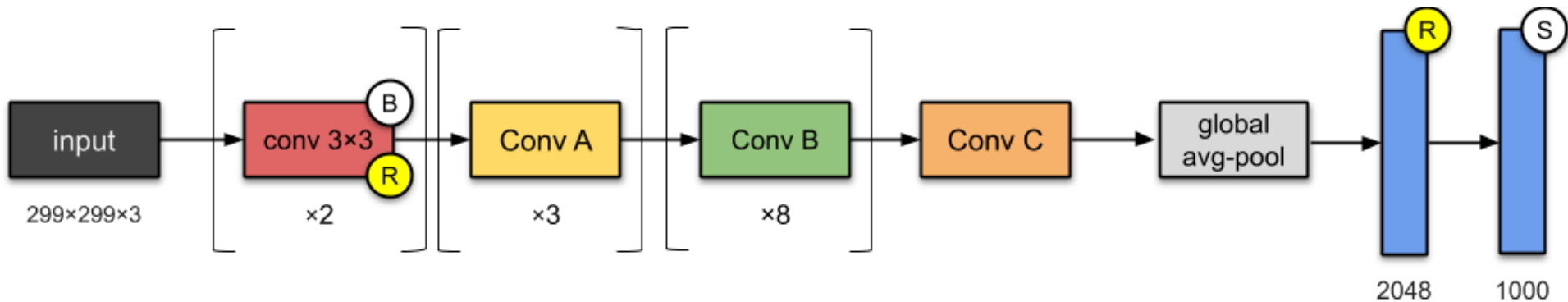


Сверточные нейронные сети

Xception (2016)

Xception построена на основе Inception. Inception модули заменены depthwise separable convolutions. It has also roughly the same number of parameters as Inception-v1 (23M).

Xception takes the Inception hypothesis to an eXtreme



Сверточные нейронные сети

В целом, если воспринимать нейрон как простую логистическую регрессию, простой классификатор, тогда нейросеть — это просто иерархический классификатор. Сначала выделяются простые признаки, из них комбинируются сложные признаки, из них еще более сложные, еще более сложные и в конце концов можно скомбинировать какой-то очень сложный признак — конкретный человек, конкретная машина, слон, что угодно еще.

Современные архитектуры сверточных нейросетей сильно усложнились. Те нейросети, которые побеждали на последних соревнованиях — это уже не просто какие-то сверточные слои, Pooling-слои. Это прямо законченные блоки.

По сути, внутри те же самые базовые компоненты: те же самые свертки и пулинги. Просто их больше, они достаточно хитро объединены. Используются прямые связи, которые позволяют вообще не трансформировать исходный сигнал, а просто передать его на выход. Засчет этого градиенты не затухают. Это дополнительный способ прохода градиента с конца нейросети к началу. Это тоже помогает обучать такие сети.

Сверточные нейронные сети

Область практического применения

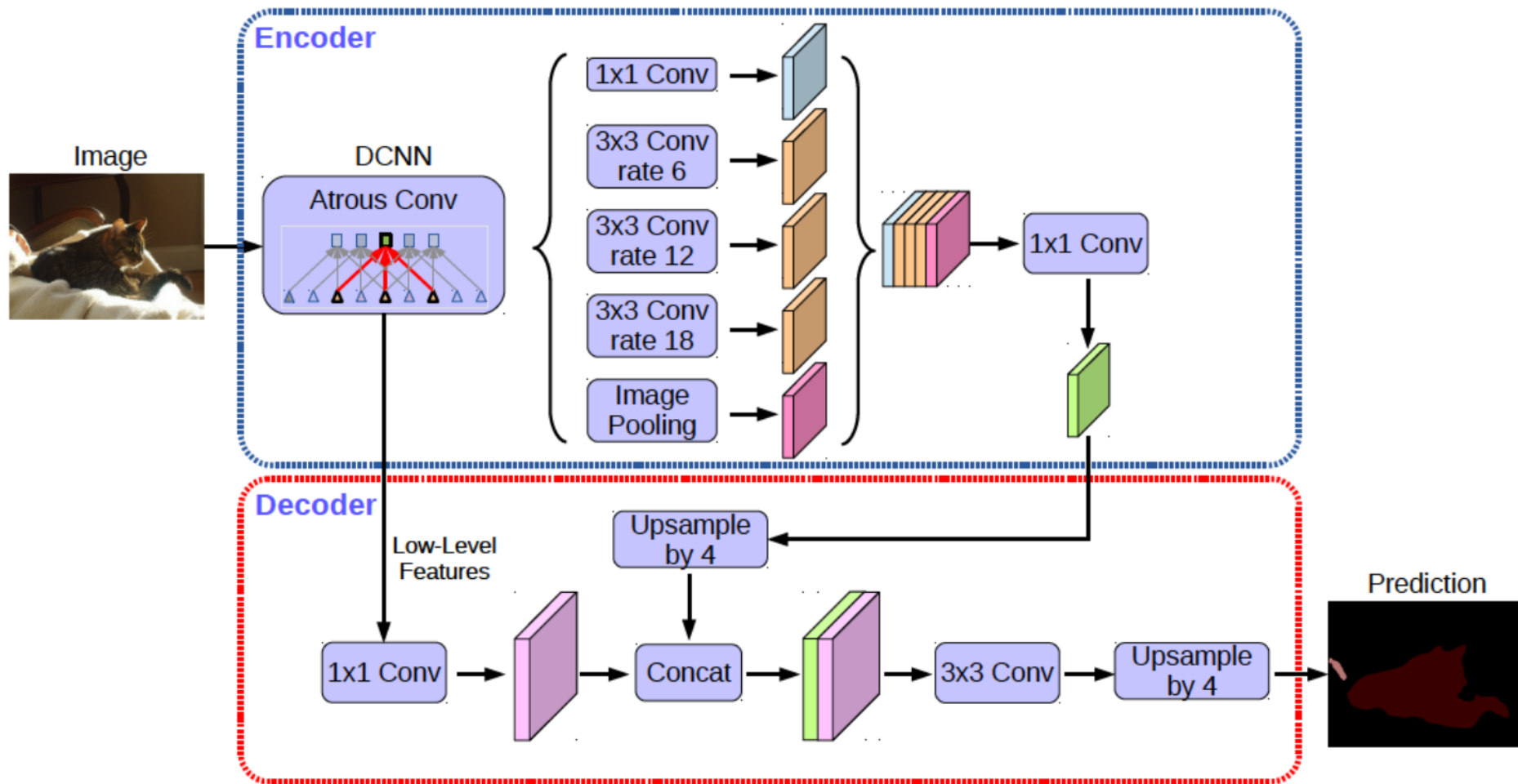
Сверточные нейросети решают 3 основные задачи:

- Классификация. Вы подаете картинку, и нейросеть просто говорит — у вас картинка про собаку, про лошадь, еще про что-то, и выдает класс.
- Детекция — это более продвинутая задачка, когда нейросеть не просто говорит, что на картинке есть собака или лошадь, но находит еще Bounding box — где это находится на картинке.
- Сегментация. На мой взгляд, это самая крутая задача. По сути, это попиксельная классификация. Здесь мы говорим про каждый пиксель изображения: этот пиксель относится к собаке, этот — к лошади, а этот еще к чему-то. На самом деле, если вы умеете решать задачу сегментации, то остальные 2 задачи уже автоматически даны.

Сверточные нейронные сети

Пример сверточной нейронной сети

На основе архитектуры Coder – Decoder и ASPP (пространственное объединение в пирамиду) разработана сеть DeepLabV3+ - победитель ImageNet 2017



Сверточные нейронные сети

Сеть разрабатывалась для решения задачи semantic segmentation - задача попиксельной классификации изображений, то есть каждому пикселю нужно присвоить класс, например, "дорога", "автомобиль", "дерево". Решается она в области беспилотного транспорта, контроля качества изделий, анализа медицинских изображений, обработки изображений со спутников.

Кодер состоит из следующих частей:

- адаптированная сеть Xception без полносвязных слоев и пространственного объединения
- ASPP модуль

На вход сети поступает изображение $513 \times 513 \times 3$. Выходное изображение меньше исходного в 16 раз. Данный выход может быть билинейно увеличен до исходного разрешения, что может считаться базовым декодером. Однако, такое исполнение декодера может неточно восстанавливать границы объектов. Поэтому был предложен простой и достаточно эффективный декодер.

Сверточные нейронные сети

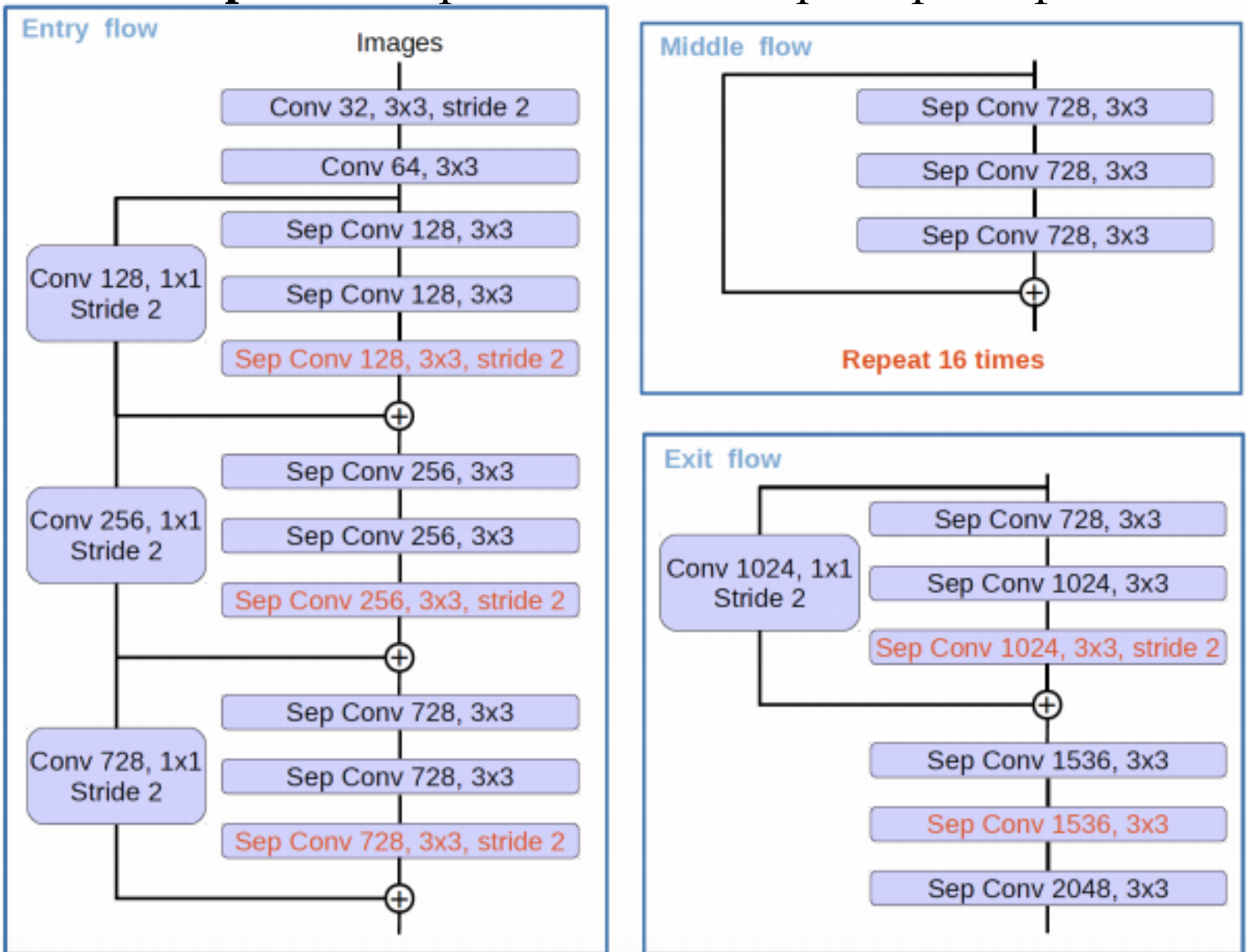
Выход кодера сначала увеличивается билинейно в 4 раза, а затем соединяется с соответствующими низкоуровневыми признаками изображения, извлекаемыми кодером, которые имеют такое же пространственное разрешение. Предварительно к этим низкоуровневым признакам применяется свертка 1×1 для уменьшения количества каналов.

Соответствующие признаки обычно имеют большое количество каналов (256 или 512), что может перевешивать значимость высокоуровневых признаков и усложнит процесс обучения.

После соединения признаков применяется свертка 3×3 для «очистки» признаков, а затем билинейная интерполяция с множителем $4.f$

Сверточные нейронные сети

Пример - сеть **Xception** для решения задачи развертки фазы

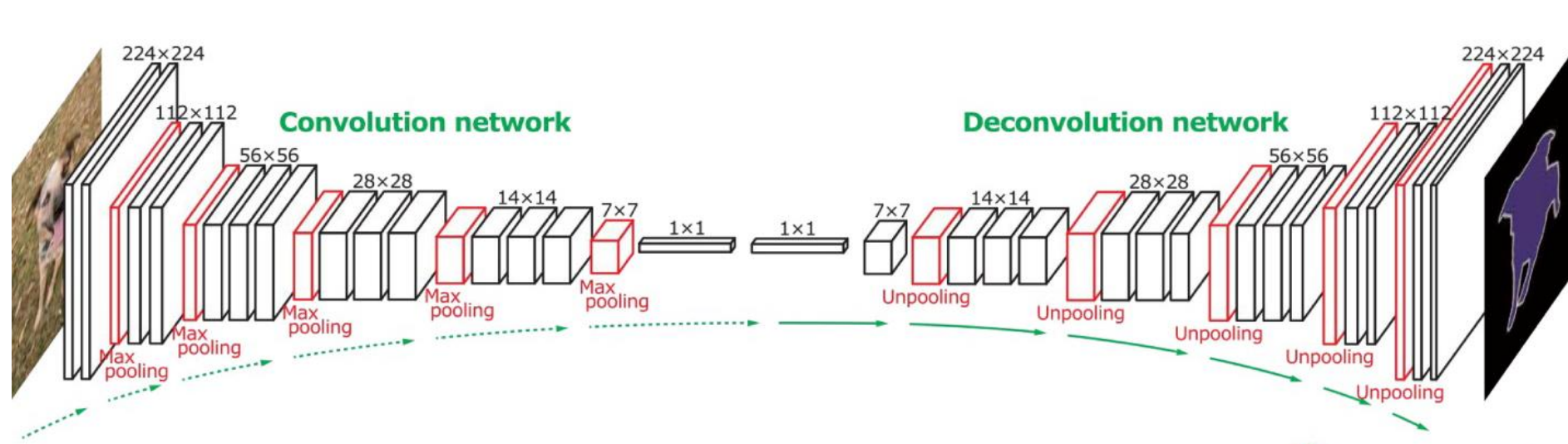


(T. Zang Rapid and robust two-dimensional phase unwrapping via deep learning, 2019)

Сверточные нейронные сети

Deconvolution (Transposed convolution) network

Это обучаемый Upsampling. То есть вы уменьшили в какой-то момент ваше изображение до какого-то небольшого размера, Потом можно взять этот вектор и раскрыть.



Сверточные нейронные сети

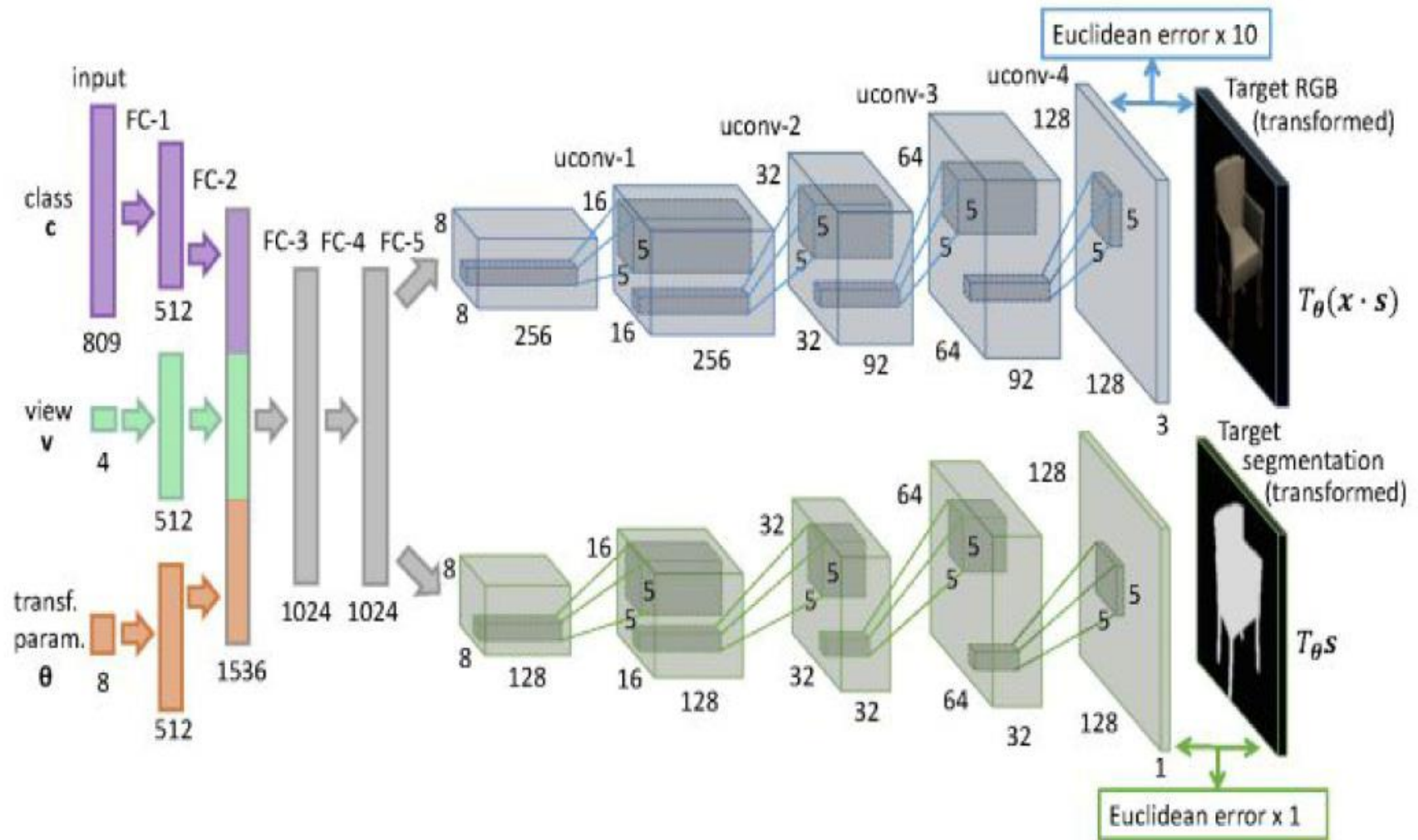
Эта архитектура позволяет обучать нейросети, которые из входной картинке получают какую-то выходную картинку. То есть вы можете подавать образцы входа/выхода, а то, что посередине, обучится само.

Такую сеть можно использовать для генерации картинок.

Например вместо первой части используем какую-то опять полносвязную сеть, которую обучим — то, что ей на вход подается, например, номер класса: сгенерируй стул под таким-то углом и в таком-то виде. Эти слои генерируют какое-то внутреннее представление этого стула, а потом оно разворачивается в картинку.

Сверточные нейронные сети

Пример сверточной сети для генерации изображений



Использованные материалы

В презентации использованы материалы, предоставленные:

Бутырским Евгением Юрьевичем, доктором физико-математических наук, профессором кафедры теории управления СПбГУ;

Гришкиным Валерием Михайловичем, кандидат технических наук, доцентом кафедры компьютерного моделирования и многопроцессорных систем.

А также следующие источники:

https://github.com/vdumoulin/conv_arithmetic

<https://habr.com/ru/company/oleg-bunin/blog/340184/>

http://bioinformaticsinstitute.ru/sites/default/files/vvedenie_v_deep_learning.pdf