Deep Learning. Сверточные нейронные сети. Обзор архитектур

Егор Конягин

22 августа 2024 г.

Содержание

1. CNN. Повторение

2. VGG-16

3. GoogLeNet

4. ResNet (2015)

5. EfficientNet (2019)

CNN. Повторение

CNN. Повторение

Мы обсудили, что полносвязные нейронные сети в задаче анализа изображений будут иметь два существенных недостатка

- огромное кол-во параметров (порядка 40-100 млн);
- неспособность к локальному анализу изображения.

CNN. Принцип действия

В основе CNN лежит обучаемая свёртка!

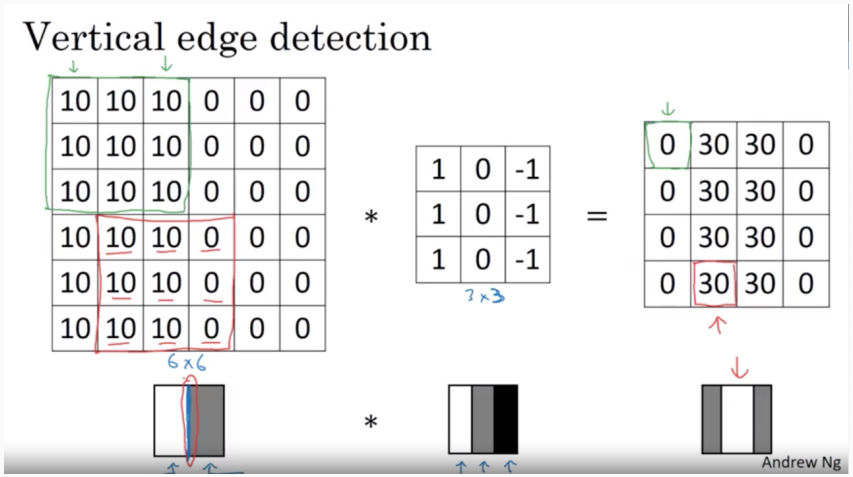

Рис. 1: Вычисление свертки изображения. Источник: Andrew Ng's classes

Если написать уравнения для backward propagation, то есть для вычисления $\frac{\partial J}{\partial w}$, то мы переходим к понятию обучаемой свёртки:

$$W = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix}. \tag{1}$$

CNN. Принцип действия - II

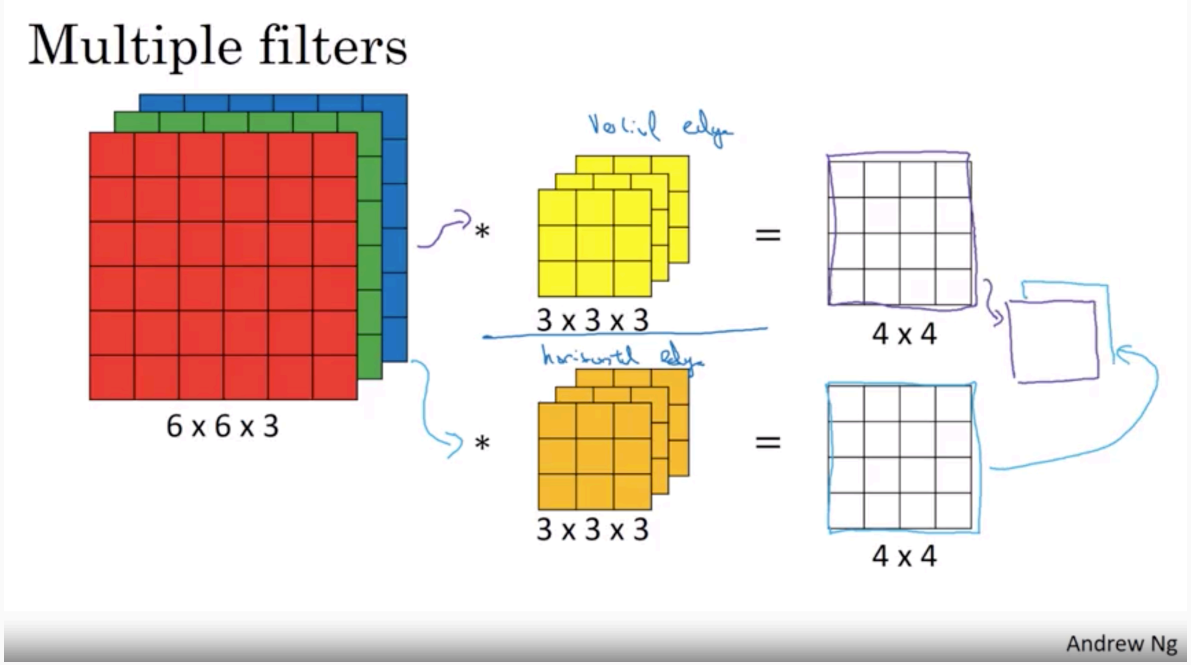

Рис. 2: Свертка над многоканальным изображением. Источник: Andrew Ng's classes

VGG-16

VGG-16 (2014)

Данная работа была разработана в 2014 году группой компьютерного зрения Visual geometry group, а именно Кареном Симоньяном и Эндрю Зиссерманом (Very Deep Convolutional Networks for Large-Scale Image Recognition).

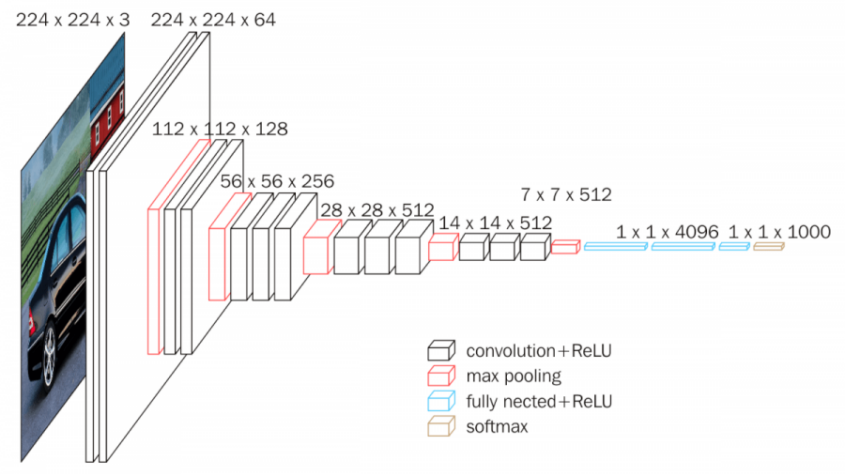

Рис. 3: Архитектура VGG-16

В отличие от AlexNet, все фильтры во всех слоях имеют размер 3x3.

VGG-16 - II. Недостатки

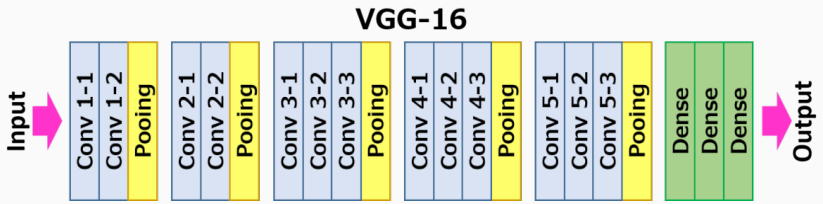

Рис. 4: Архитектура VGG-16. Слои

К сожалению, архитектура VGG-16 обладает двумя недостатками:

1. данная нейросеть обучается слишком медленно;
2. нейросеть имеет много параметров, которые занимают порядка 500 МБ при разворачивании (138 млн параметров).

В современных задачах нейросеть в качестве основной модели не используется!

GoogLeNet

GoogLeNet (2014)

Данная нейронная сеть существенно отличается от всех предыдущих нейросетей. Нам придется познакомиться со следующими концепциями перед тем, как непосредственно расмотреть эту архитектуру:

- свёртка с фильтром 1x1;
- модуль inception;
- global average pooling;
- adaptive average pooling.

GoogLeNet. 1x1 convolution

Рассмотрим две модели сверточной нейросети и посчитаем кол-во совершаемых операций:

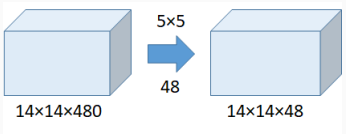

Рис. 5: Сверточный слой

Кол-во операций = (14×14×48)×(5×5×480) = 112.9M

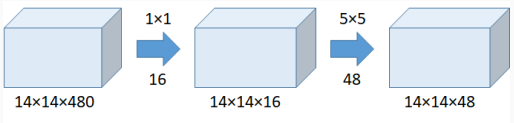

Рис. 6: Сверточный слой с использованием 1x1 convolution

Кол-во операций для свертки 1×1 = (14×14×16)×(1×1×480) = 1.5M
Кол-во операций для свертки 5×5 = (14×14×48)×(5×5×16) = 3.8M
Всего операций = 1.5M + 3.8M = 5.3M $\ll$ 112.M.

GoogLeNet. Inception module

Рассмотрим блок операций, которая называется inception module:

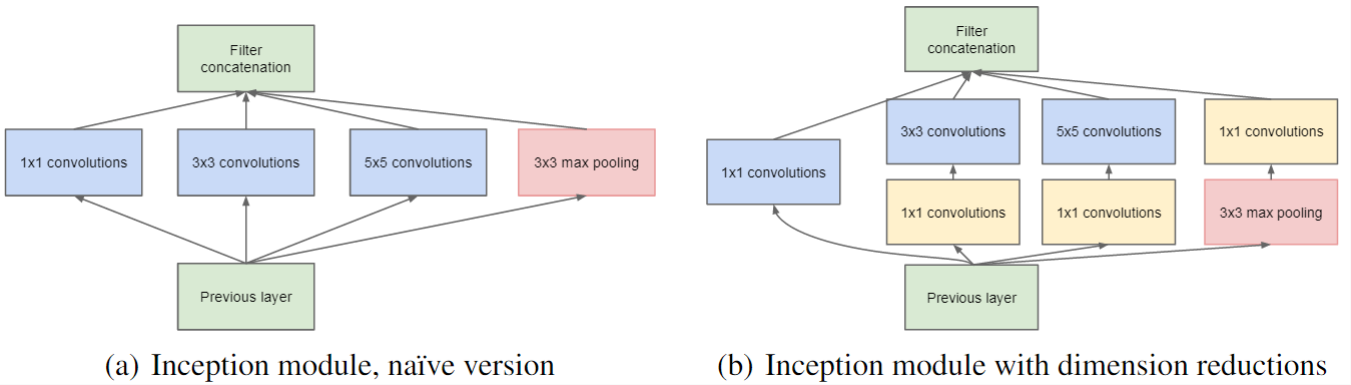

Рис. 7: Архитектура блока inception module. Свертка 1x1 нужна для снижения кол-ва операций

Такой блок - попытка решить проблему того, что детектируемый признак может быть разного масштаба. Большие объекты детектируются фильтрами большого размера, малые объекты - малыми фильтрами.

GoogLeNet. Global average pooling

Данный метод применяется для снижения кол-ва весов при переходе от сверточных к полносвязным слоям.

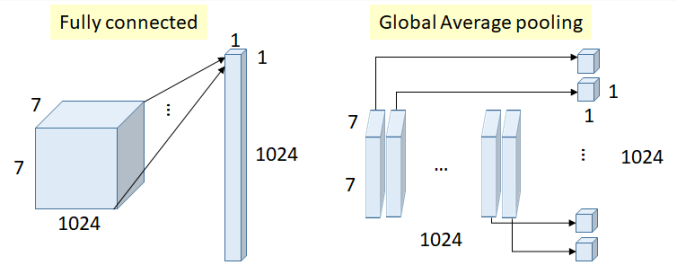

Рис. 8: Суть global avg pooling

Как было ранее замечено, большинство параметров в свёрточных нейросетях расположены в последних полносвязных слоях.

Кол-во весов слева: 7×7×1024×1024 = 51.3M.

Кол-во весов справа: 0. В данном случае считается среднее по каждой из матриц 7x7, это число записывается в 1024-мерный вектор.

Adaptive average pooling

Эта концепция похожа на global average pooling. Идея такой операции в том, чтобы всегда выдавать массив фиксированного размера (на вход приходит тензор размера (N_{batch}, C, H, W), на выход $(N_{batch}, C, D_{out}, D_{out})$, где D_{out} - заданный размер, в то время как H, W - произвольные размеры).

В рамках проведения этой операции автоматически подбирается окно пулинга и параметр stride, чтобы в результате операции получился тензор вышеописанных размерностей.

GoogLeNet. Архитектура

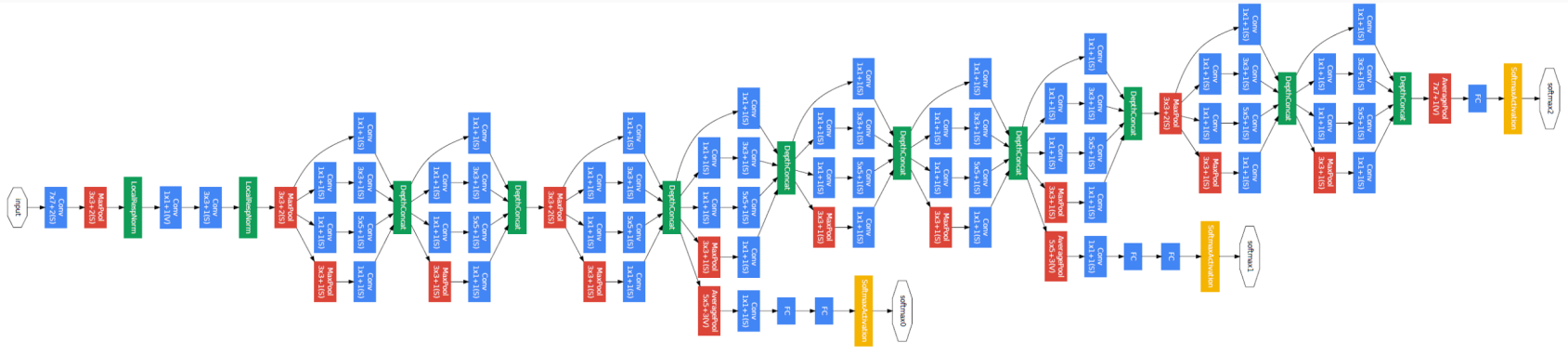

Рис. 9: Архитектура GoogLeNet

Несмотря на кажущуюся сложность, данная нейросеть имеет меньше 7 миллионов параметров (сравните с VGG-16). Будучи в 15 раз ”легче чем VGG-16, она не сильно уступила ей по качеству в классификации изображений.

ResNet (2015)

ResNet. Мотивация

Дальнейшее развитие нейросетей вглубь сопровождалось серьезными проблемами. Они были вызваны затухающими градиентами. Таким образом, градиент уменьшался при прохождении от последних слоев к предыдущим, и самые первые слои нейросети учились все хуже и хуже.

Тем не менее, увеличение количества слоев должно было улучшить механизм извлечения визуальных признаков из изображения, поэтому были опубликованы различные способы борьбы с затухающими градиентами. Наиболее популярные из них:

- skip connection,
- gradient clipping,
- batch normalization,
- блок LSTM в рекуррентных нейросетях*.

ResNet. Skip connection

Основу нейросети ResNet составляет т.н. блок skip (shortcut) connection:

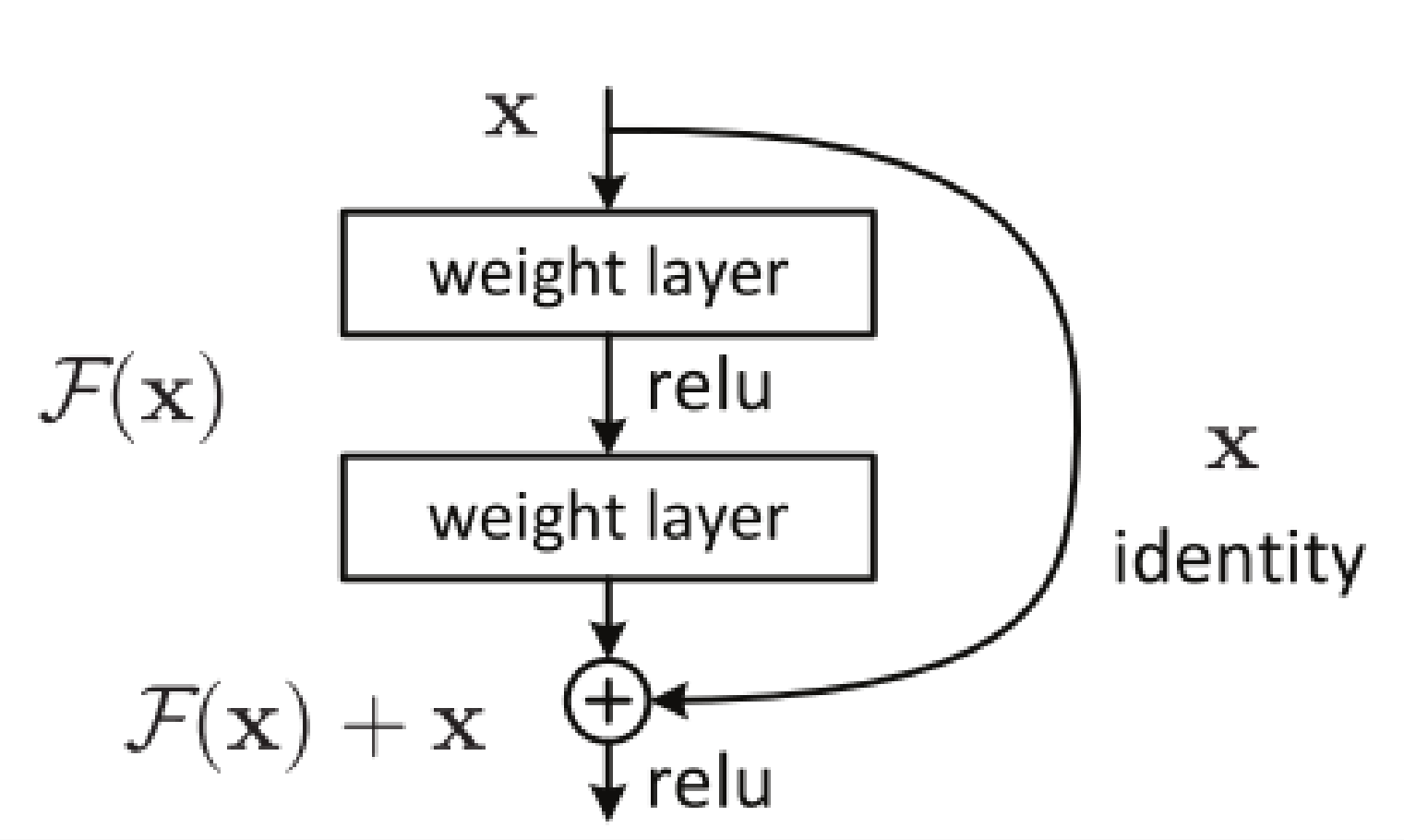

Рис. 10: Архитектура skip connection

ResNet-18

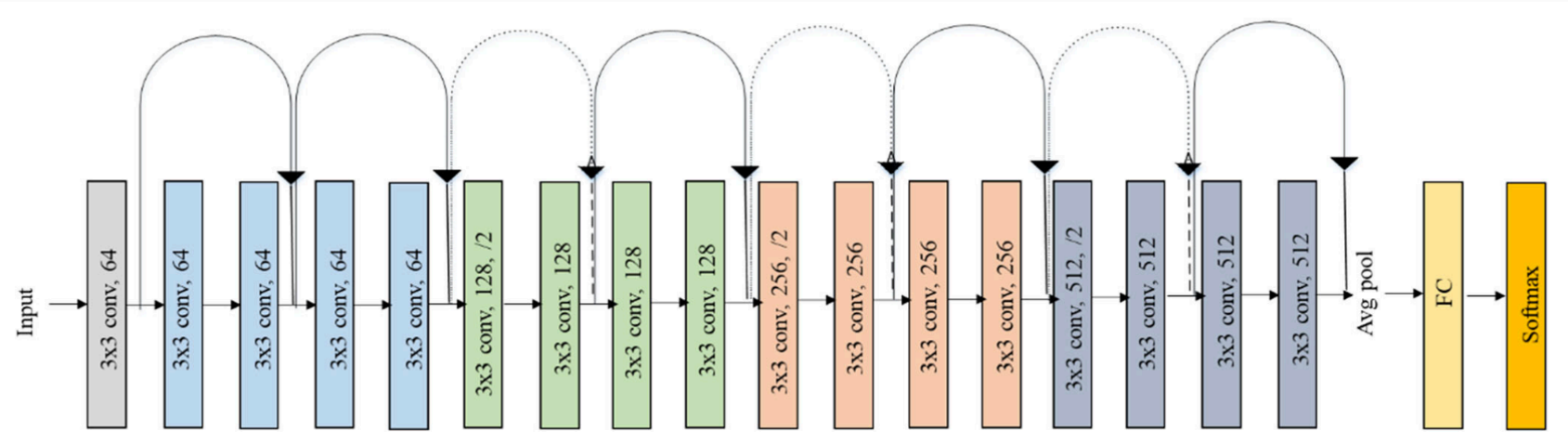

Рис. 11: Архитектура ResNet-18

ResNet состоит из блоков skip connection, которые завершаются одним fc-слоем (т. е. полносвязным). Кол-во слоев в сети варьируется в различных модификациях от 18 до 1002. Кол-во параметров: $\sim$ 20 000 000.

EfficientNet (2019)

Масштабирование нейросетей

Сверточные нейросети принципиально можно улучшать тремя способами:

1. увеличивая количество слоев (вглубь),
2. увеличивая количество каналов в каждом слое (вширь),
3. используя входные картинки бóльшего разрешения

Масштабирование нейросетей

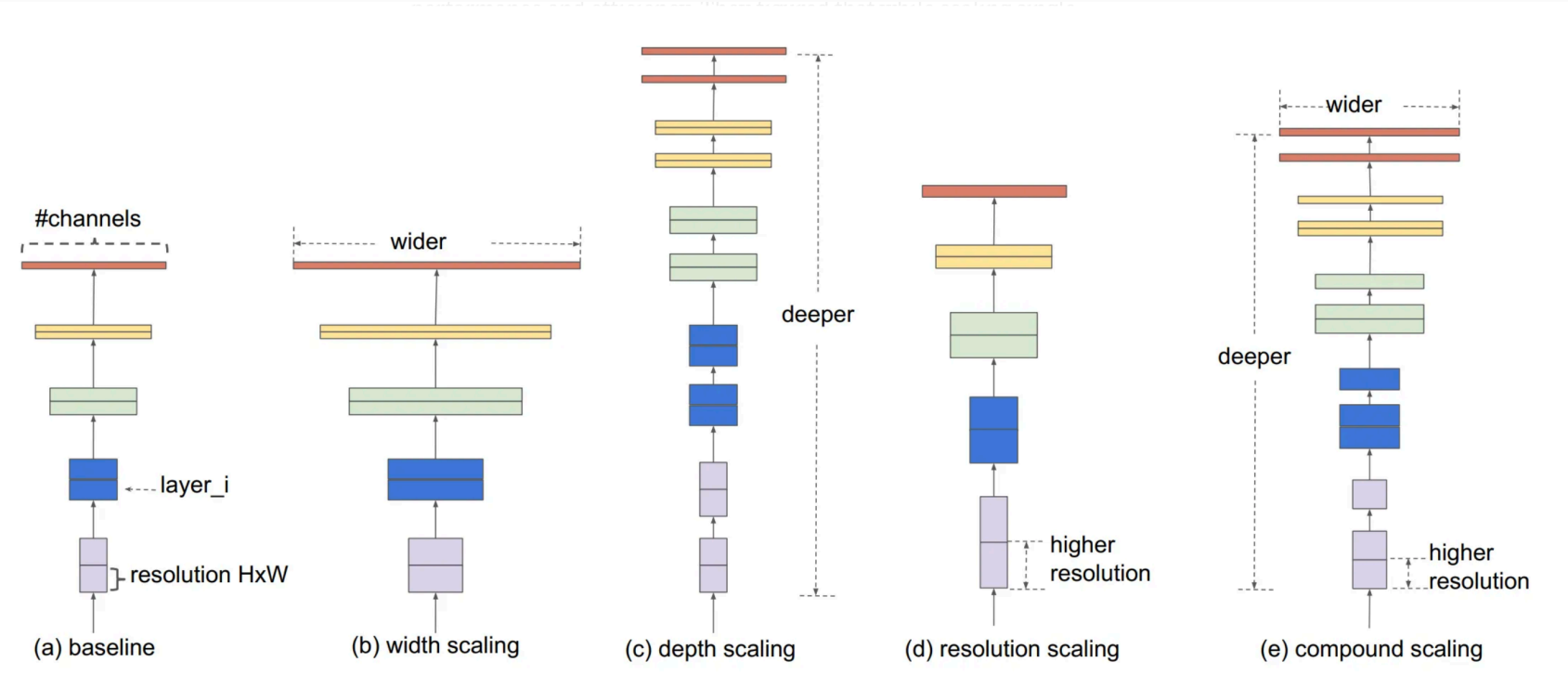

Рис. 12: Различные методы масштабирования нейросетей: вглубь и вширь

EfficientNet. Мотивация

Чем выше разрешение, тем больше каналов нужно для хорошей обучаемости сети и тем больше нужно слоев.

Однако кол-во слоев и кол-во каналов здесь влияют друг на друга. Пусть d, w, r - глубина нейросети, ширина нейросети (кол-во каналов в первом слое) и разрешение картинки, соответственно. Тогда

$$d = \alpha^\phi, \quad w = \beta^\phi, \quad r = \gamma^\phi, \tag{2}$$

$$\text{s.t} \quad \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \tag{3}$$

Параметр ϕ в данном случае (при фиксированных остальных параметрах) регулирует размер нейросети.

ResNet. Bottleneck

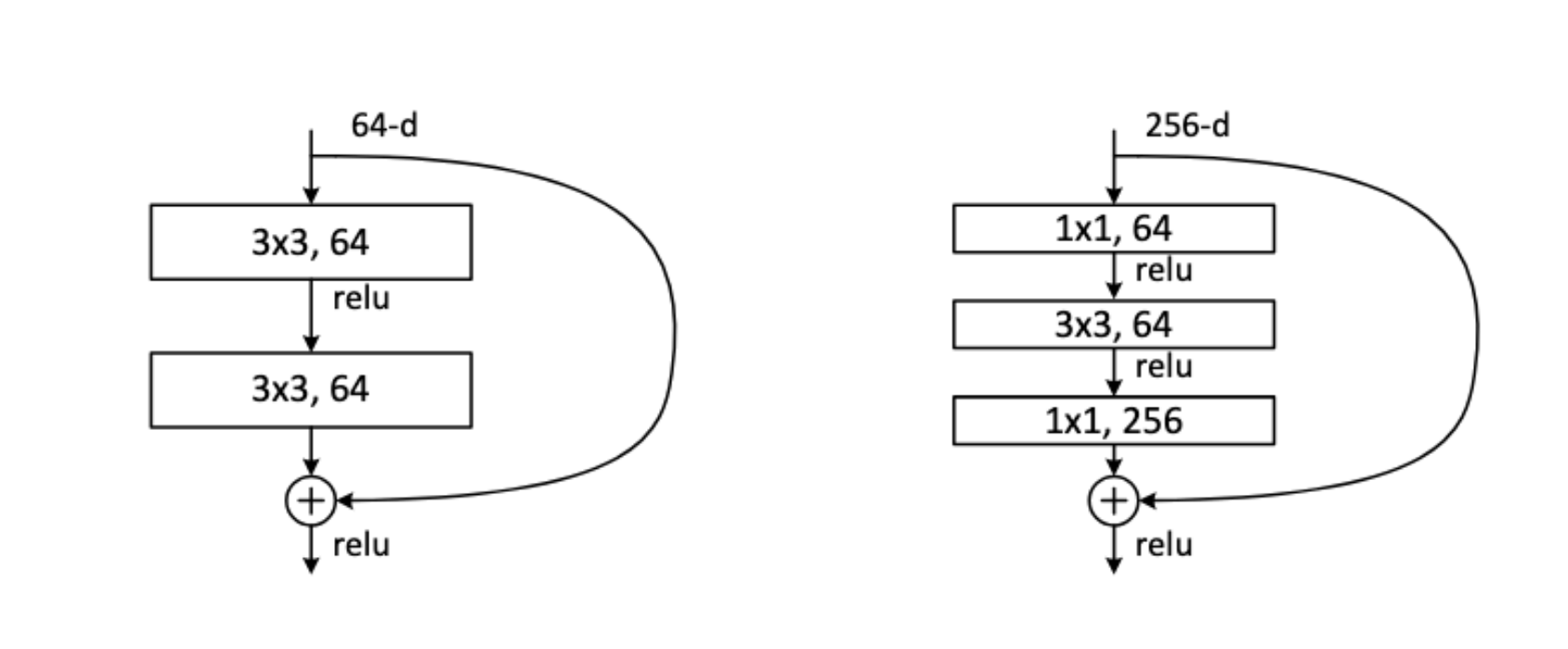

Рис. 13: Блоки bottleneck в сети ResNet

В нейросетях ResNet большой глубины используются блоки ‘bottleneck‘: входное количество каналов сокращается, признаки извлекаются, затем количество каналов снова увеличивается.

EfficientNet. MBConv

В нейросетях EfficientNet используются блоки, похожие на bottlenecks, но вывернутые наизнанку (inverse residual block): входное кол-во каналов растет, признаки извлекаются, затем кол-во каналов снова уменьшается.

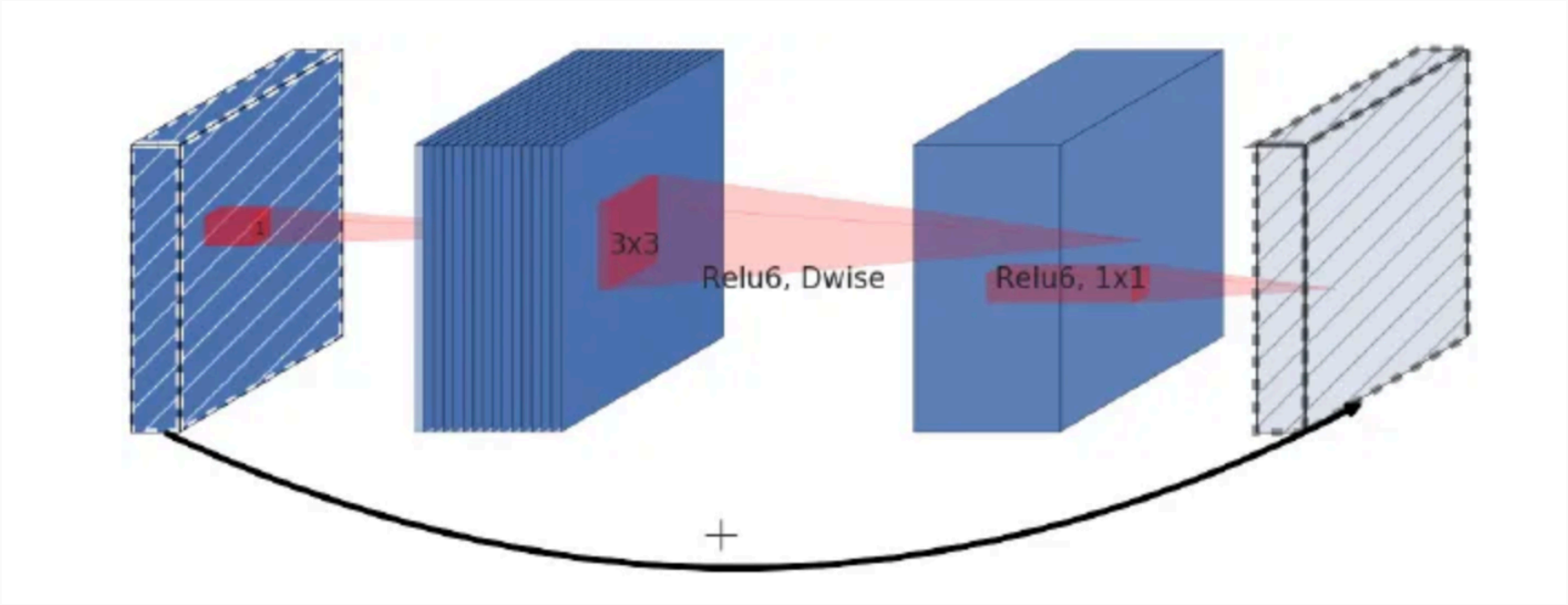

Рис. 14: Блоки inverse bottleneck в сети EfficientNet

Другие сверточные модели

- ResNeXt (2016)
- Xception (2017)
- MobileNet (2017)
- EfficientNet V2 (2021)
- ConvNeXt (2022)

Summary

Мы рассмотрели следующие архитектуры нейронных сетей:

1. VGG-16;
2. GoogLeNet;
3. ResNet;
4. EfficientNet.