Свёрточные сети (convolutional neural networks, CNNs)

Антон Осокин

ФКН ВШЭ

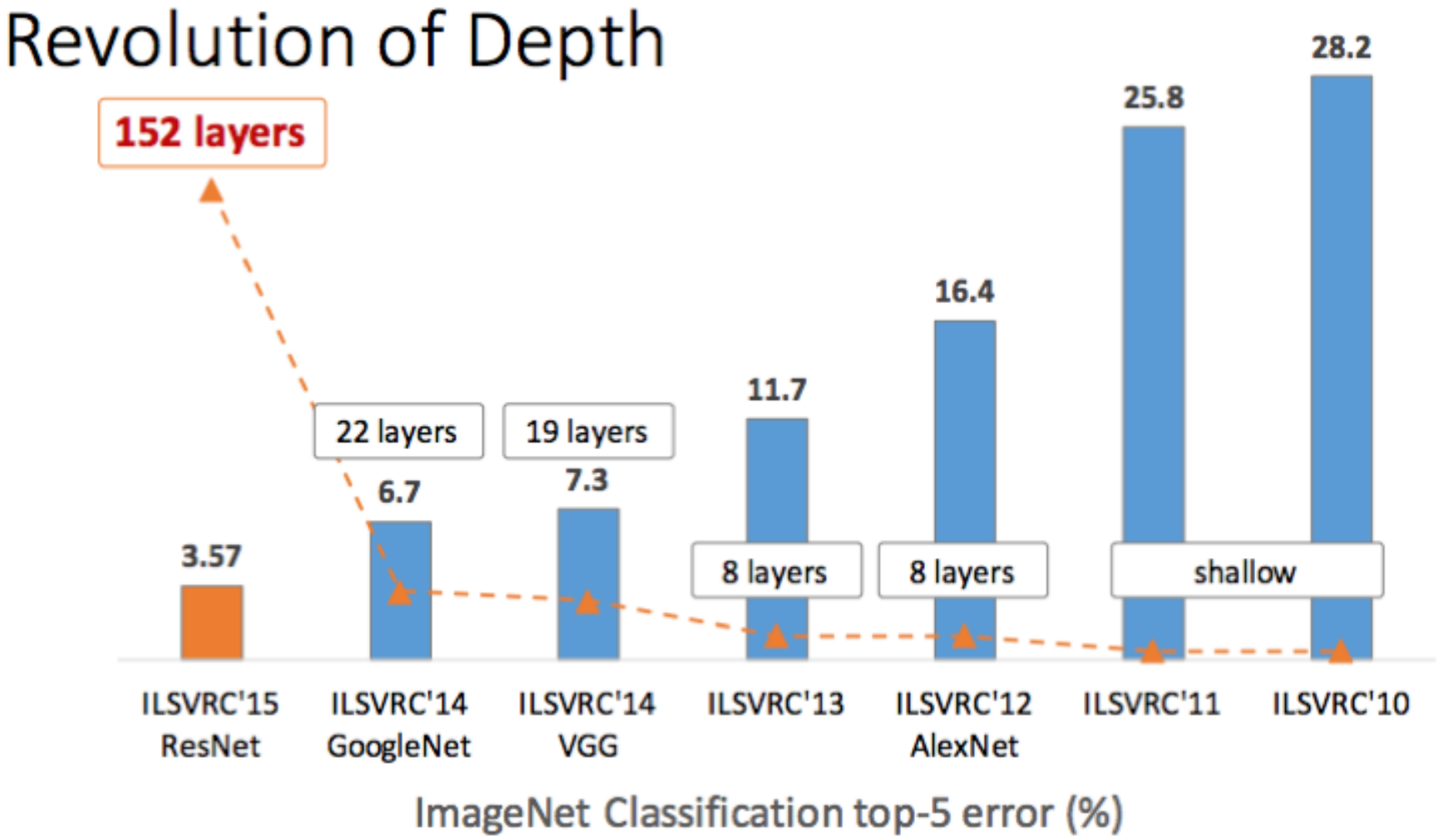
22.09.2017

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

ImageNet challenge

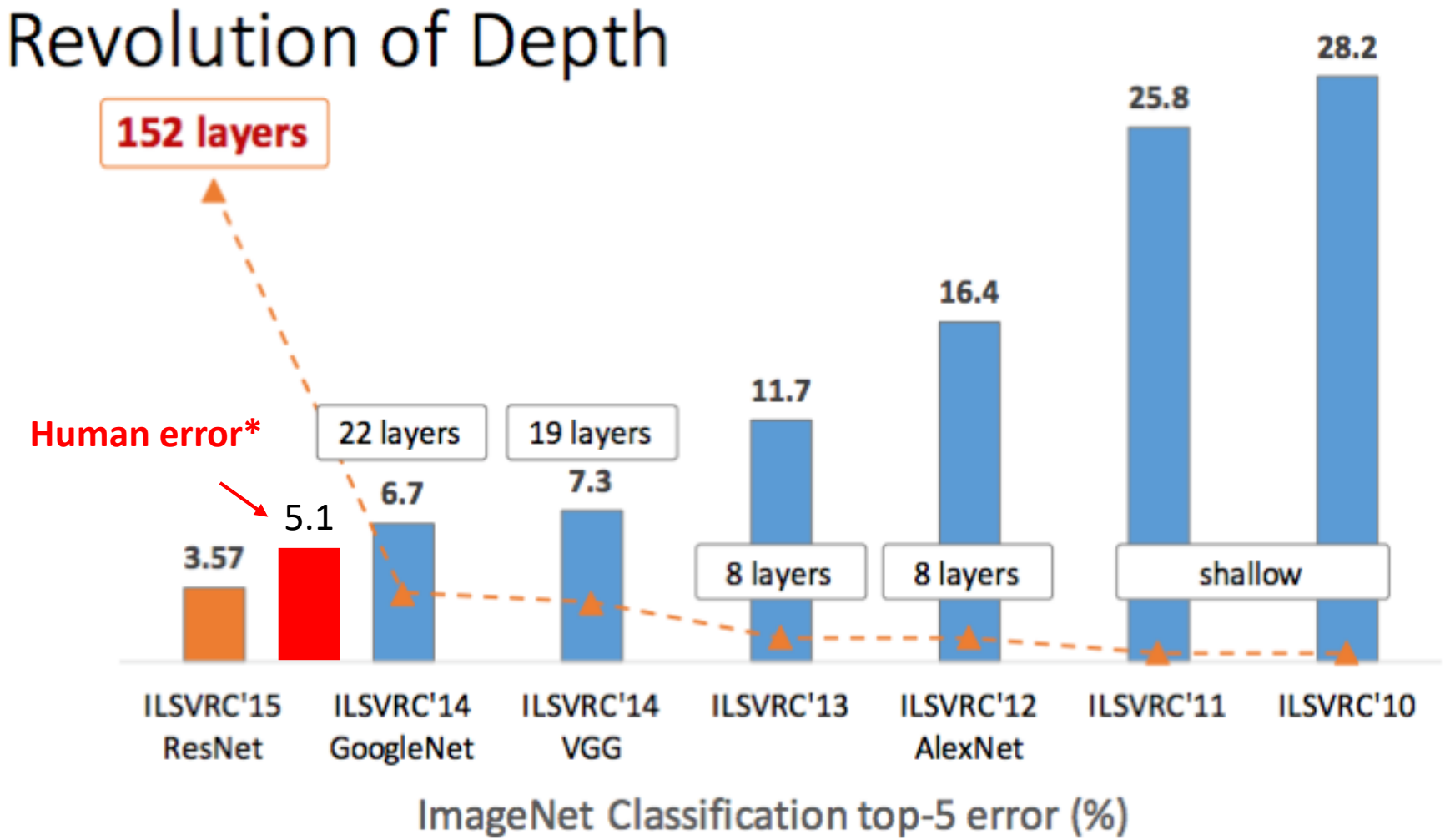
- Классификация изображений
- 1.2M изображений
- 1000 классов
- Данные из интернета
- Аннотация при помощи Amazon MTurk

ImageNet challenge



plot credit: Kaiming He

ImageNet challenge



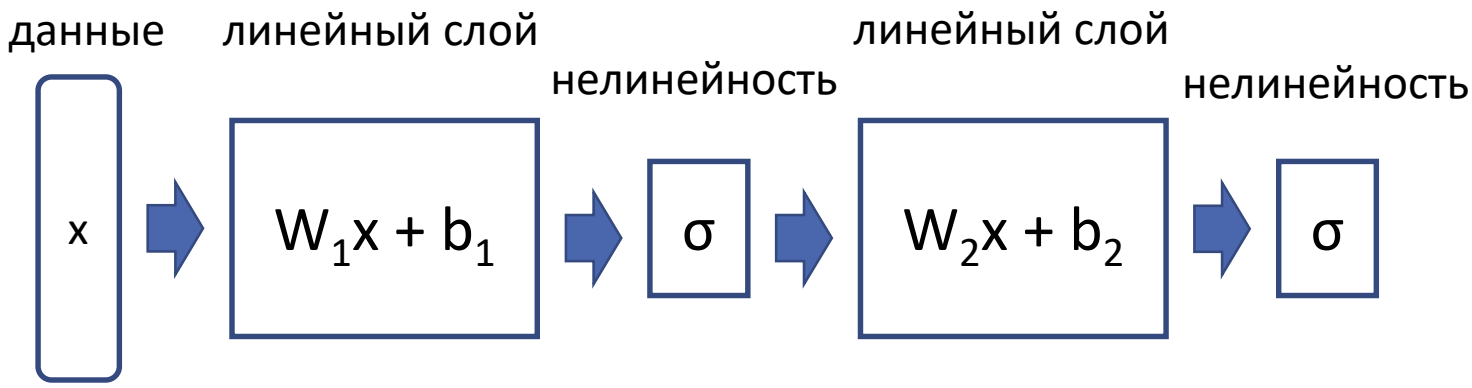
* - by Andrej Karpathy (no big claims)

plot credit: Kaiming He

План лекции

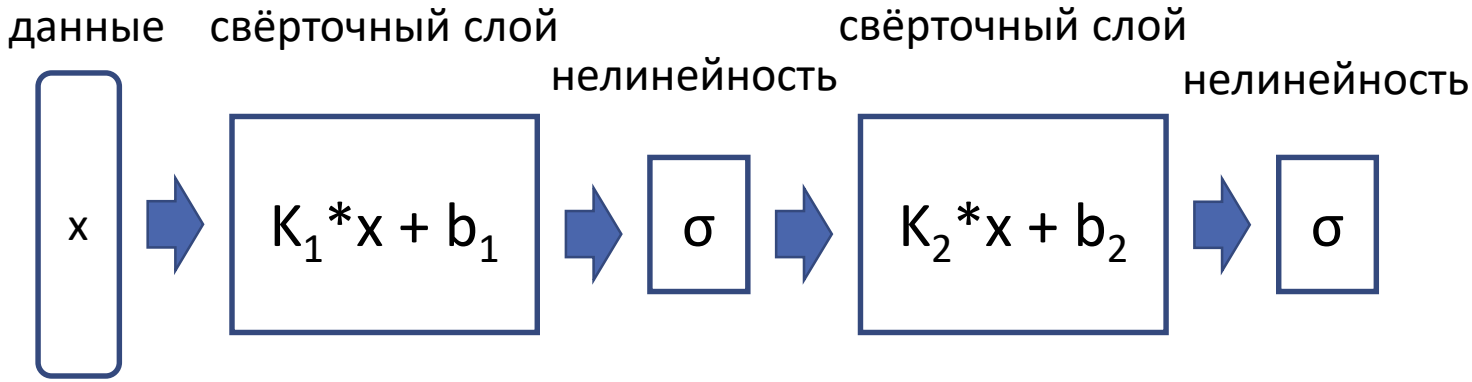
- Нейросети для классификации изображений
- Свёрточные сети
  - LeNet
  - AlexNet
- Операции свёртки и pooling
- Визуализация сетей
- Современные архитектуры
  - VGGnet
  - GoogleNet (Inception)
  - ResNet

Обычные нейросети



- Слишком много параметров
- Для картинки 100 x 100 x 3 – первая размерность 30к
- Очень быстро переобучаются
- Нужно использовать более компактные параметризации

Свёрточные сети



- Свёртка (кросс-корреляция)

$$(K * x)(i,j,k) = \sum_{u=i-\delta}^{i+\delta} \sum_{v=j-\delta}^{j+\delta} \sum_{c} x(u,v,c) K(u-i+\delta, v-j+\delta, c, k)$$

- Гораздо меньше параметров (kernel, filter)
- Фильтры не зависят от размеров картинки

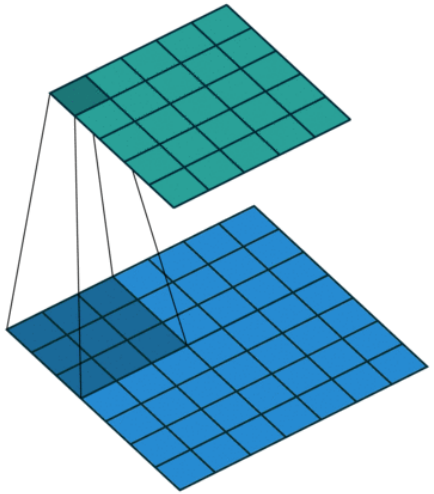
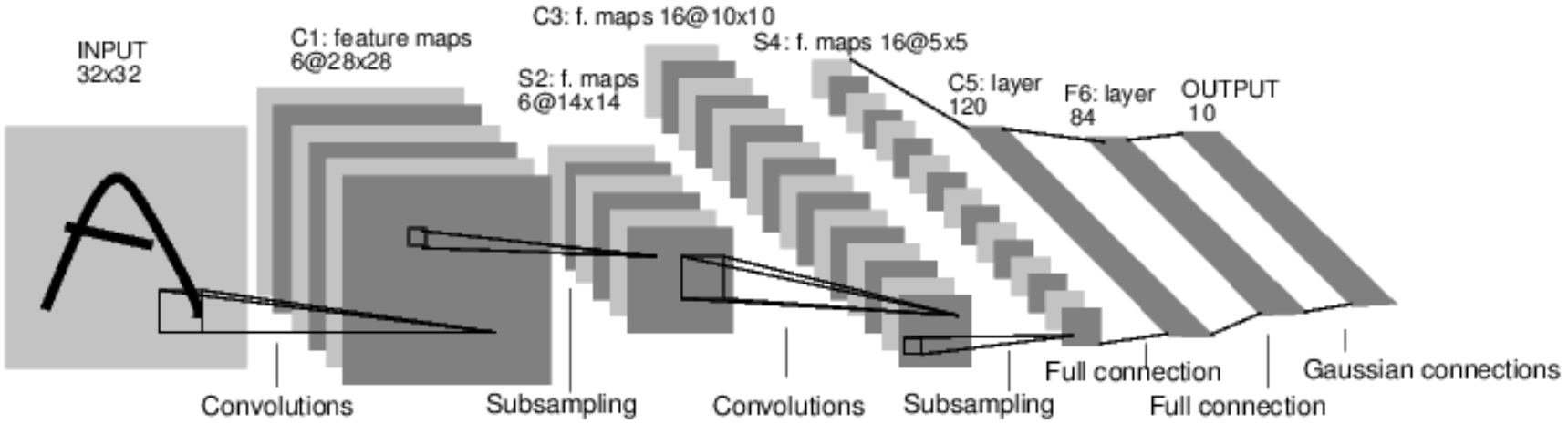
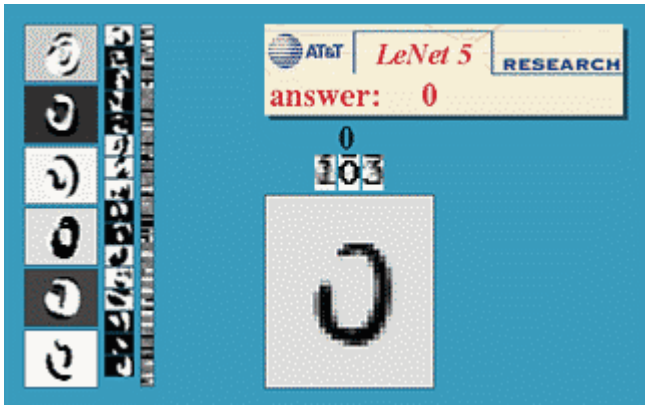


diagram: theano tutorials

Свёрточные сети: LeNet (1998)

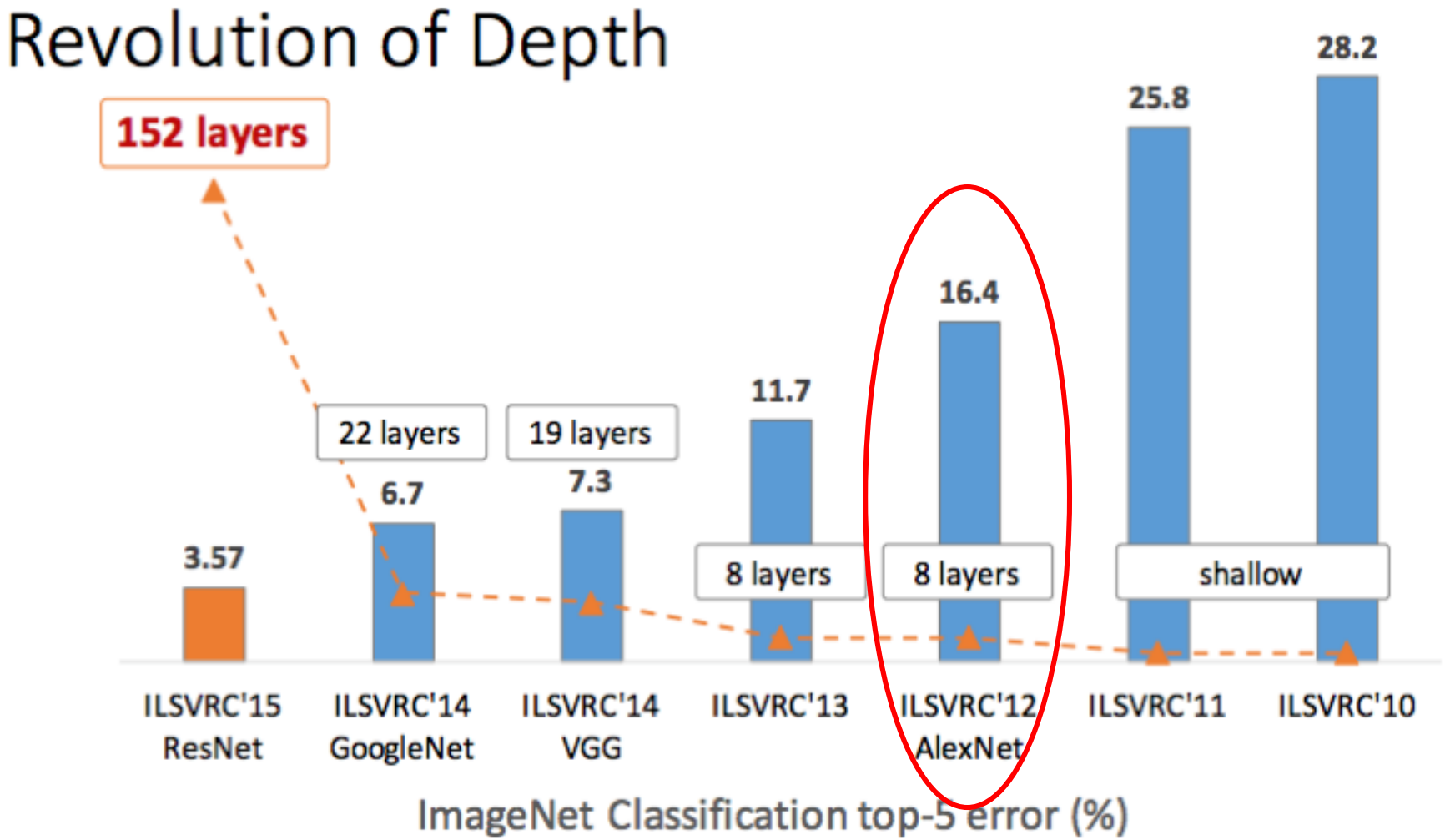


- Average pooling
- Sigmoid nonlinearity
- Fully connected layers
- Успешно классифицировала MNIST



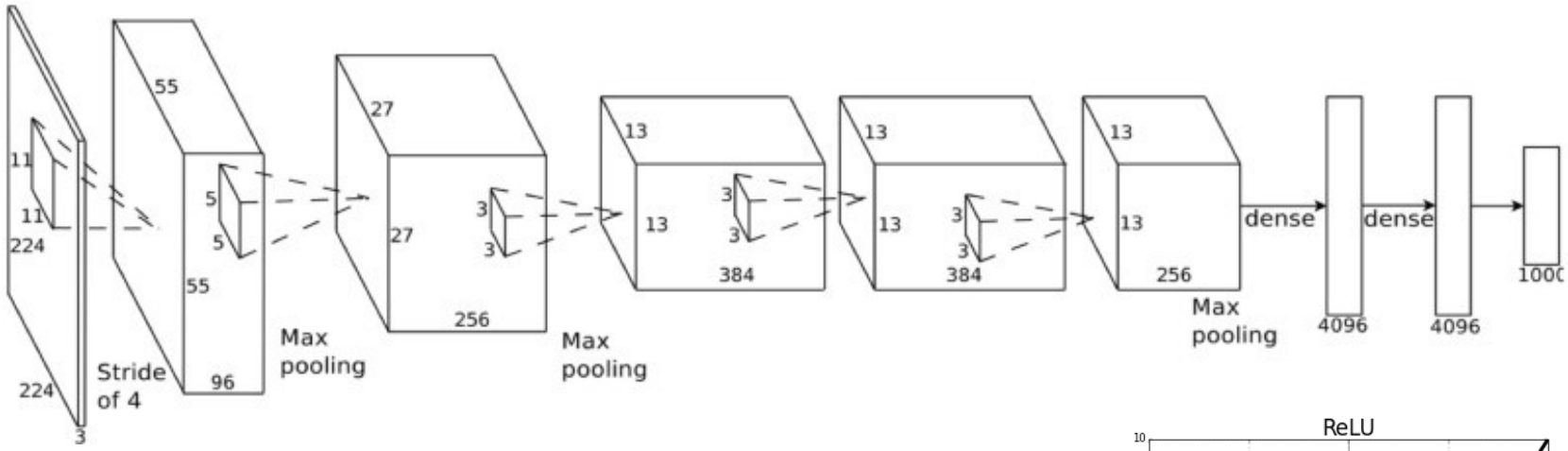
[LeCun et al., 1998]

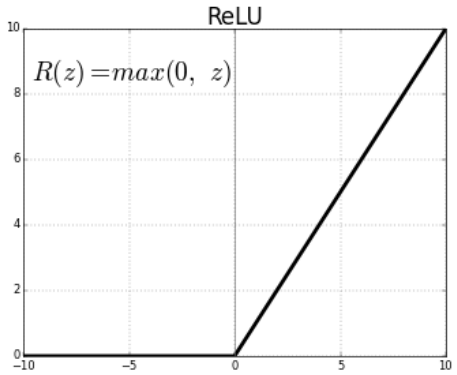
ImageNet challenge



plot credit: Kaiming He

Свёрточные сети: AlexNet (2012)





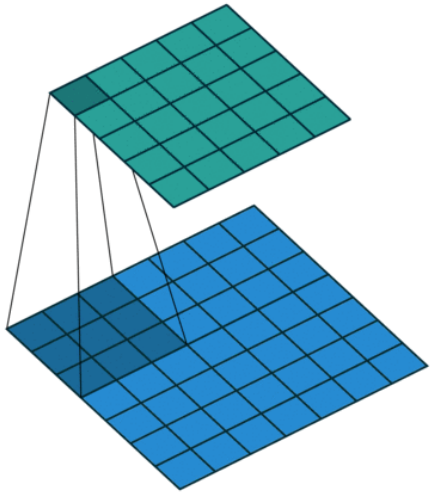
- Max pooling
- ReLu nonlinearity
- Fully connected layers
- More data and bigger model (60M params)
- + Data augmentation (flips and random samples)
- + Dropout regularization
- + GPUs (50x speed up)
- + 1 week of training on 2 GPUs

[Krizhevsky et al., 2012]

Свёртка (convolution)

Свёртка (кросс-корреляция)

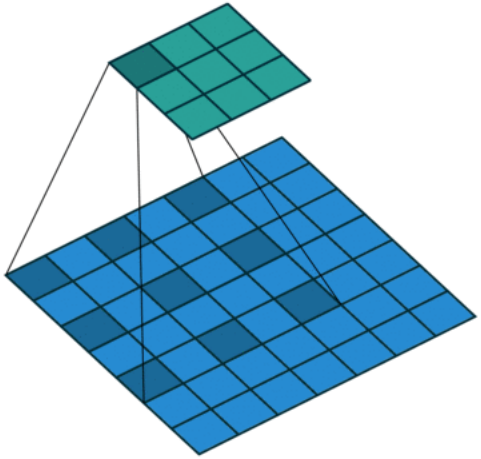
$$(K * x)(i,j,k) = \sum_{u=i-\delta}^{i+\delta} \sum_{v=j-\delta}^{j+\delta} \sum_{c} x(u,v,c) K(u-i+\delta, v-j+\delta, c, k)$$



Параметры (pytorch):

- Количество каналов на входе и выходе
- Размеры ядра
- Смещение (stride) – большие значения понижают разрешение
- Padding (нет, нули, зеркальный)
- Dilation – увеличить область зависимости
- Размер выхода сети:

$$L_{out} = floor((L_{in} + 2pad - dil(kernel - 1) - 1)/stride + 1)$$



Реализация свёртки

Свёртка (кросс-корреляция)

$$(K * x)(i,j,k) = \sum_{u=i-\delta}^{i+\delta} \sum_{v=j-\delta}^{j+\delta} \sum_{c} x(u,v,c) K(u-i+\delta, v-j+\delta, c, k)$$

Свёртка – линейная операция.

Основная идея – использовать матричное умножение:

$$\begin{pmatrix} k_1 & k_2 \\ k_3 & k_4 \end{pmatrix} * \begin{pmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{pmatrix} \Rightarrow \begin{pmatrix} k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 & 0 \\ 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 \\ 0 & 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{pmatrix} = \begin{pmatrix} k_1x_1 + k_2x_2 + k_3x_4 + k_4x_5 \\ k_1x_2 + k_2x_3 + k_3x_5 + k_4x_6 \\ k_1x_4 + k_2x_5 + k_3x_7 + k_4x_8 \\ k_1x_5 + k_2x_6 + k_3x_8 + k_4x_9 \end{pmatrix}$$

Очень эффективные реализации: NVIDIA cuDNN, Nervana kernels

Специализация: метод Винограда для сверток 3x3, преобразования Фурье для больших свёрток

Дифференцирование свёртки

Свёртка (кросс-корреляция)

$$(K * x)(i,j,k) = \sum_{u=i-\delta}^{i+\delta} \sum_{v=j-\delta}^{j+\delta} \sum_{c} x(u,v,c) K(u-i+\delta, v-j+\delta, c, k)$$

Свёртка – линейная операция.

Производная – тоже матричное умножение

$$y = \begin{pmatrix} k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 & 0 \\ 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 \\ 0 & 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 \end{pmatrix} x$$

$$\frac{d\mathcal{L}}{dx} = \begin{pmatrix} k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 & 0 \\ 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 \\ 0 & 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 \end{pmatrix}^T \frac{d\mathcal{L}}{dy}$$

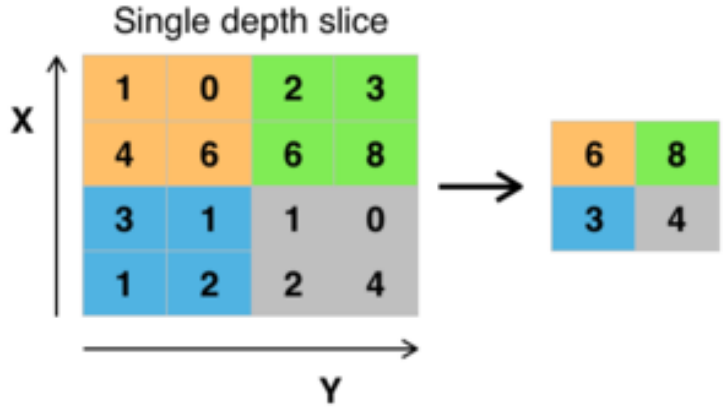
Операция upconvolution (conv-transpose) позволяет увеличить пространственное разрешение

Pooling

Pooling – агрегация признаков (max, sum)

Pooling слой агрегирует соседние активации

(пространственно или из разных фильтров)

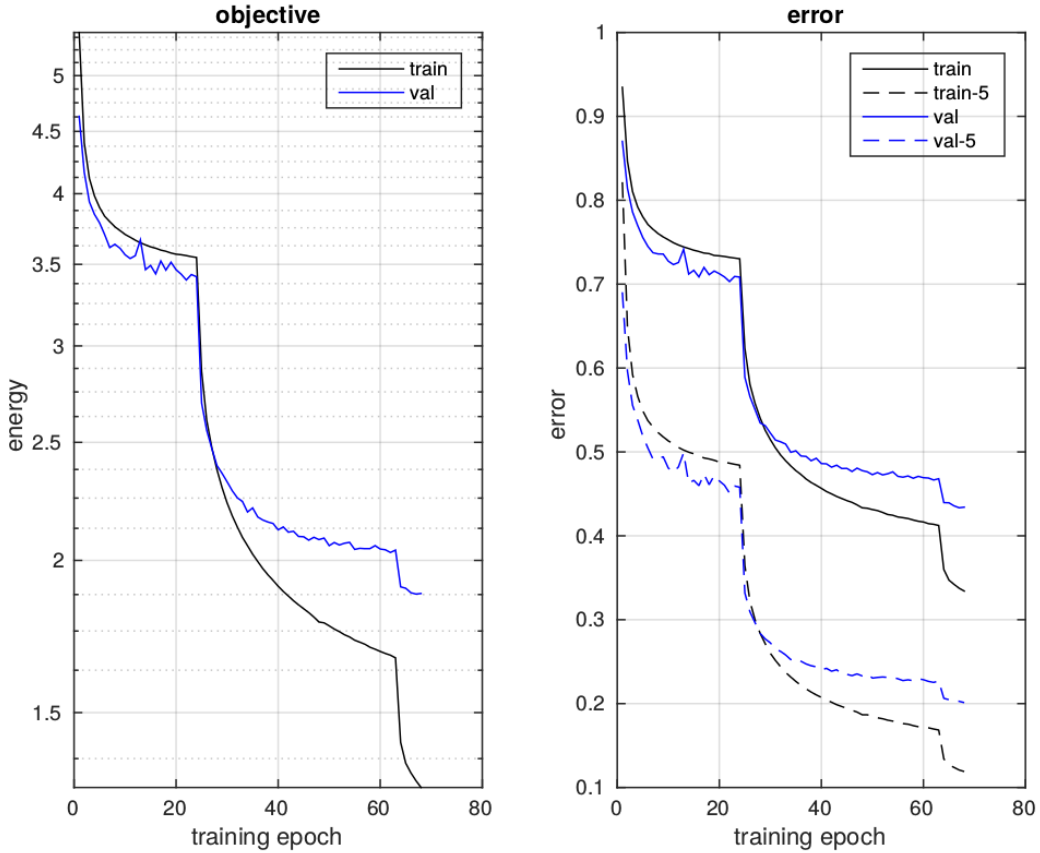


Max-pooling обеспечивает инвариантность к небольшим сдвигам (иногда полезно, иногда нет)

Дифференцирование – вернуть градиент в позиции максимумов

Как понять, что сеть обучается?

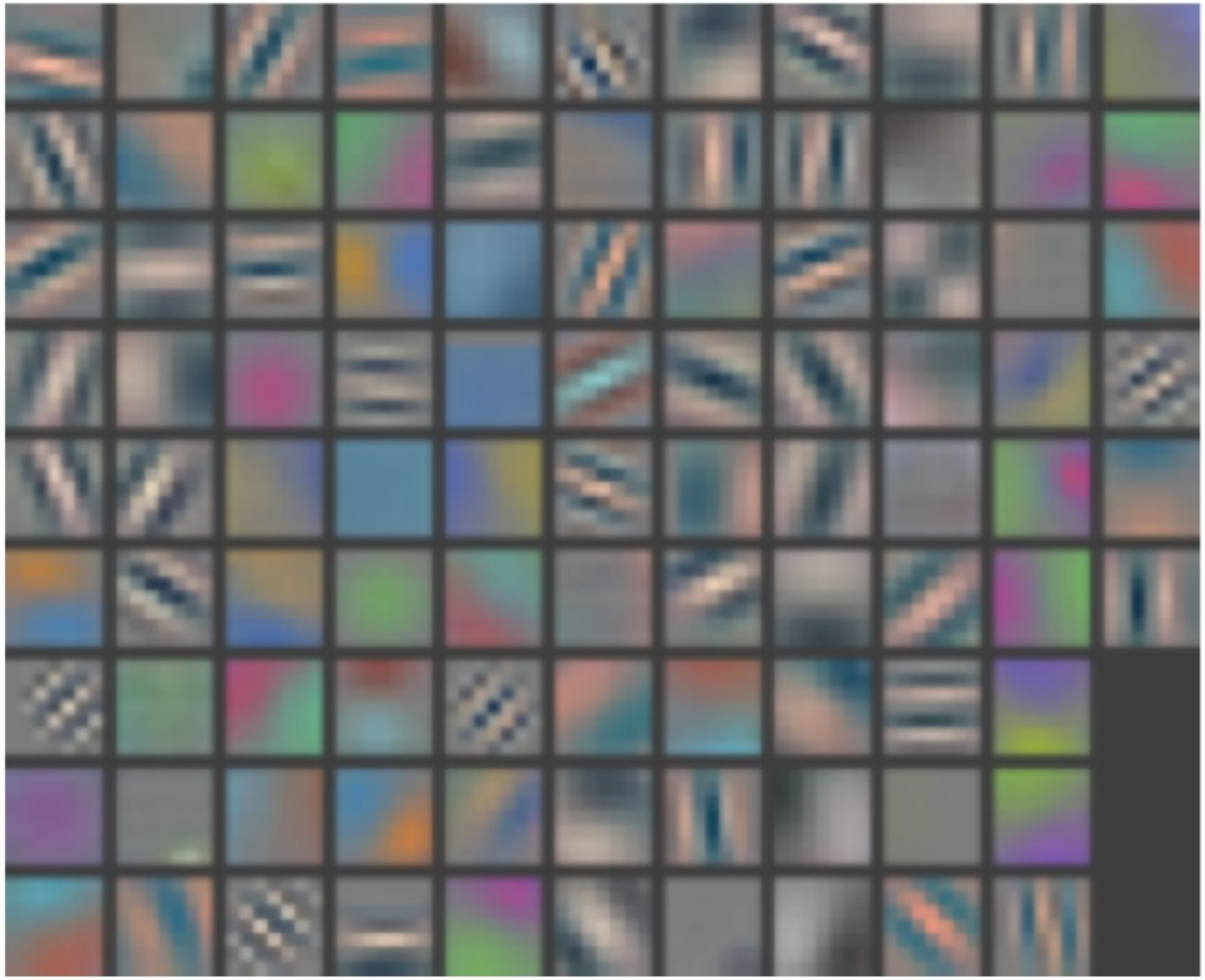
Смотреть на ошибки на обучении и валидации



SGD with momentum and learning rate decrease!

Визуализация свёрточных сетей

Фильтры первого слоя:

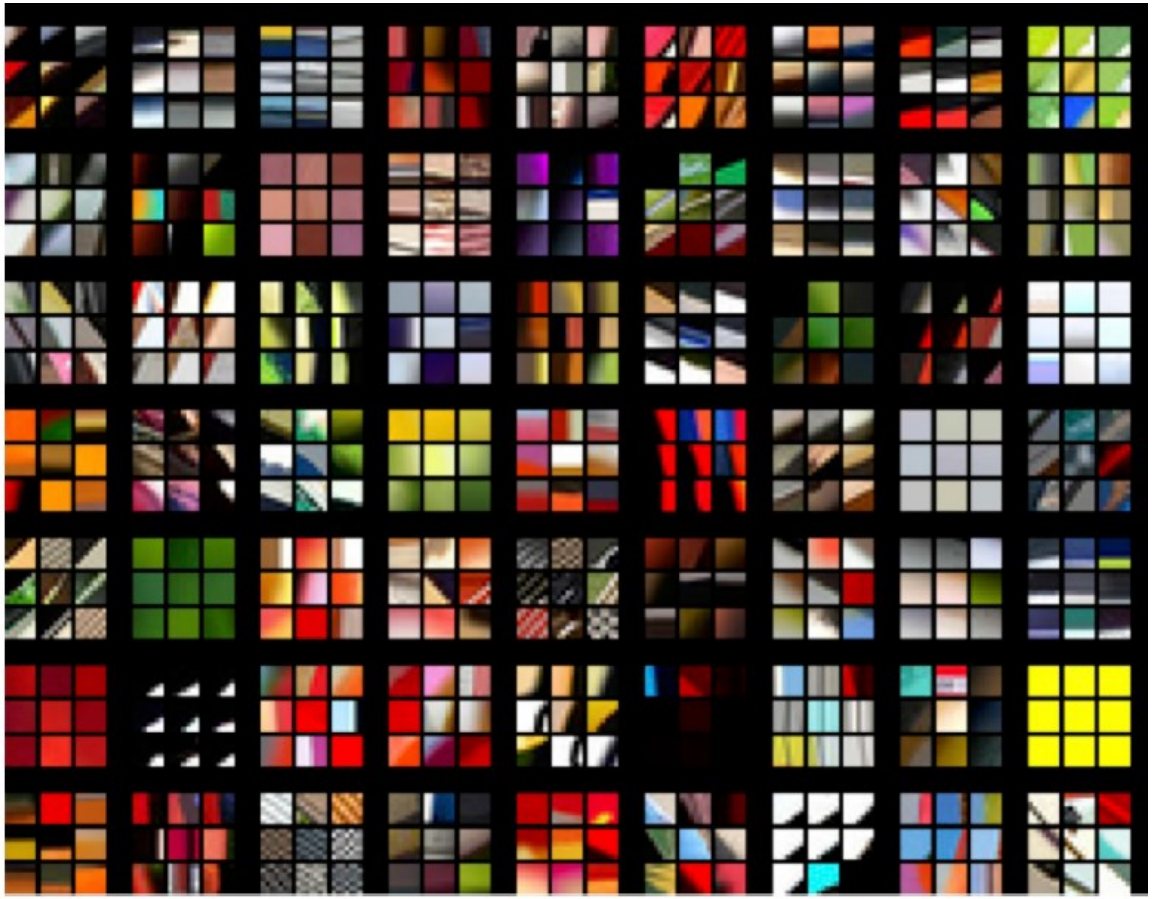


Визуализация свёрточных сетей

Оставшиеся слои – надо «спроецировать» на пиксели

Один из способов – искать патчи, дающие наибольший отклик

Слой 1:



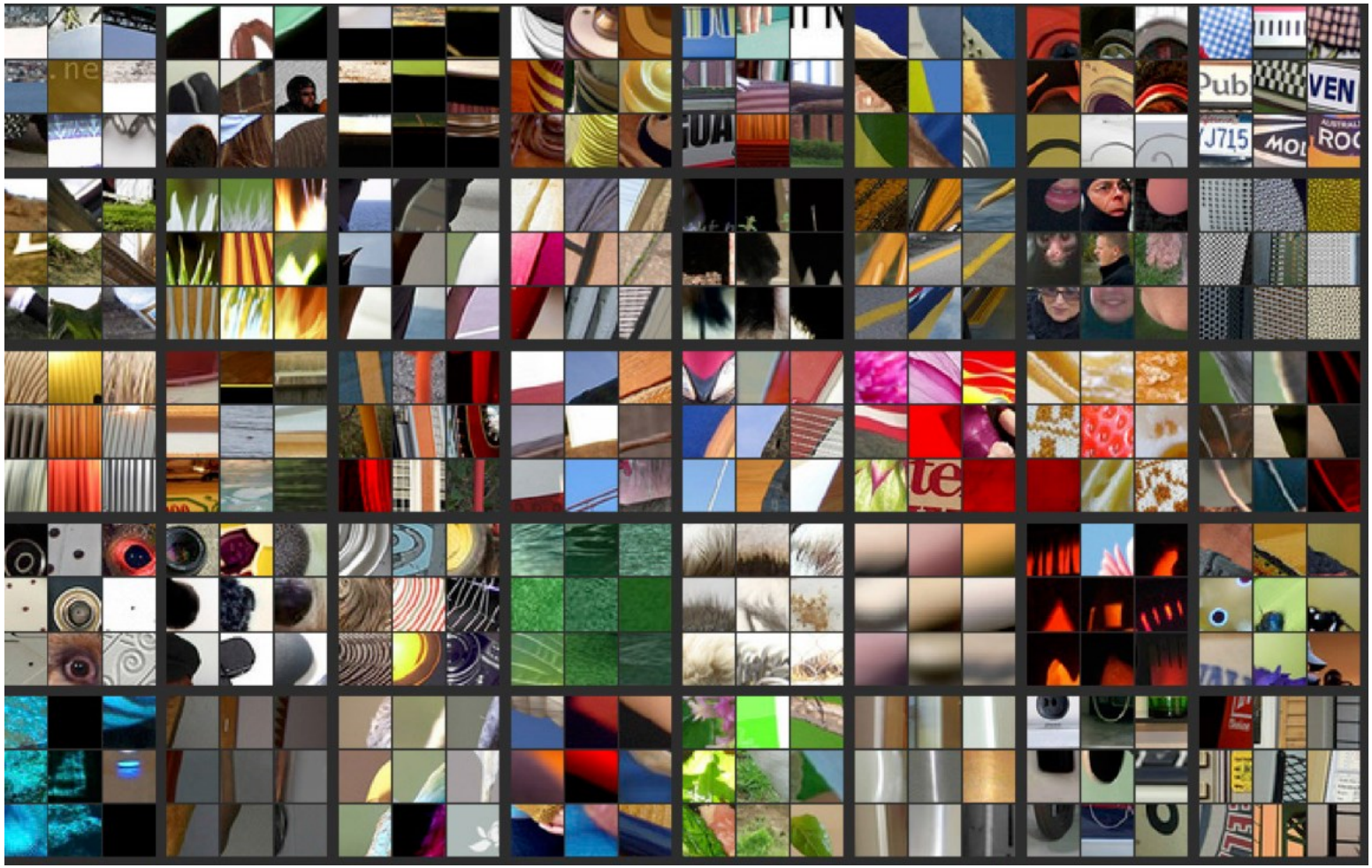
[Zeiler, Fergus 2014]

Визуализация свёрточных сетей

Оставшиеся слои – надо «спроецировать» на пиксели

Один из способов – искать патчи, дающие наибольший отклик

Слой 2:



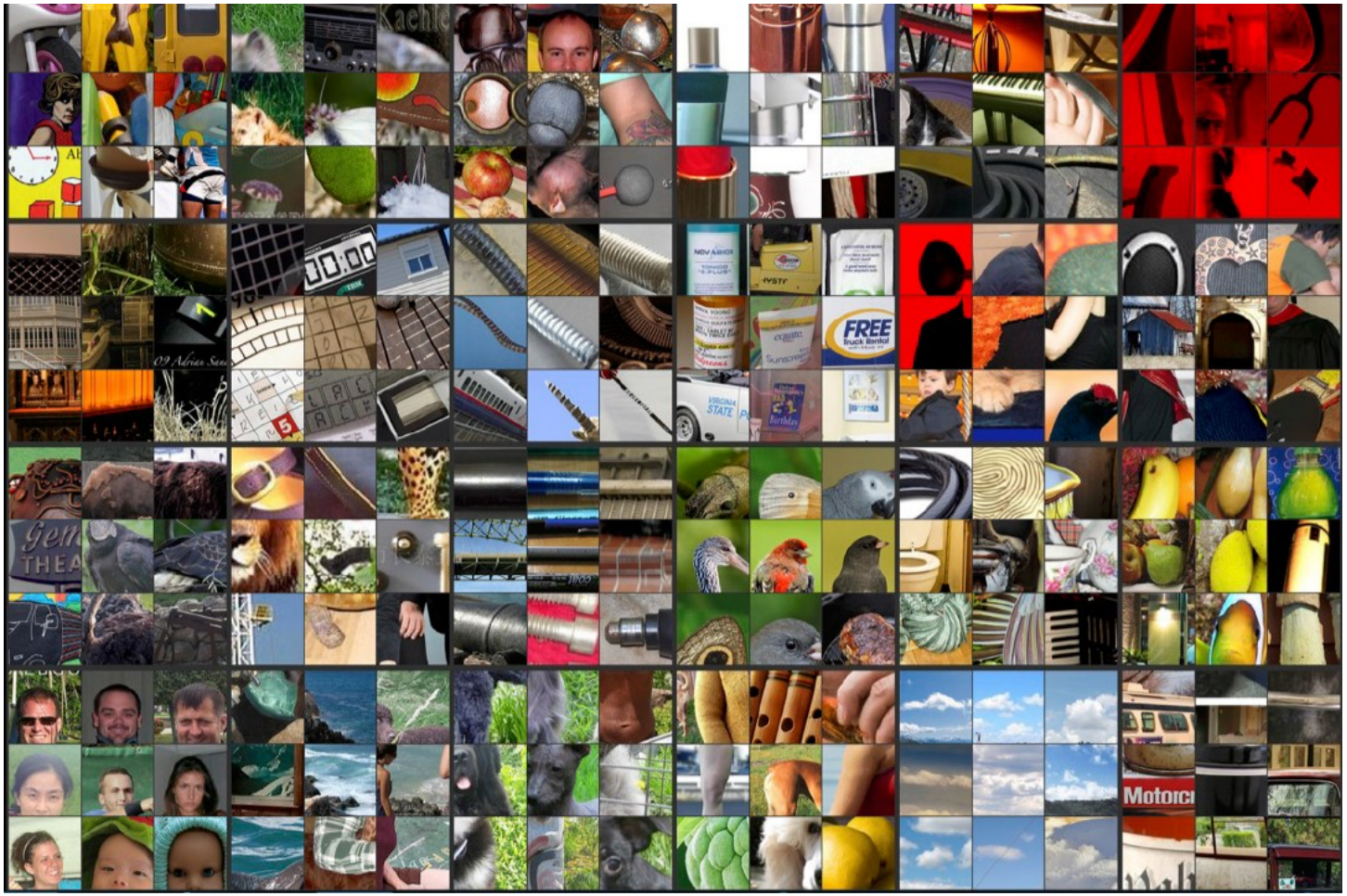
[Zeiler, Fergus 2014]

Визуализация свёрточных сетей

Оставшиеся слои – надо «спроецировать» на пиксели

Один из способов – искать патчи, дающие наибольший отклик

Слой 3:



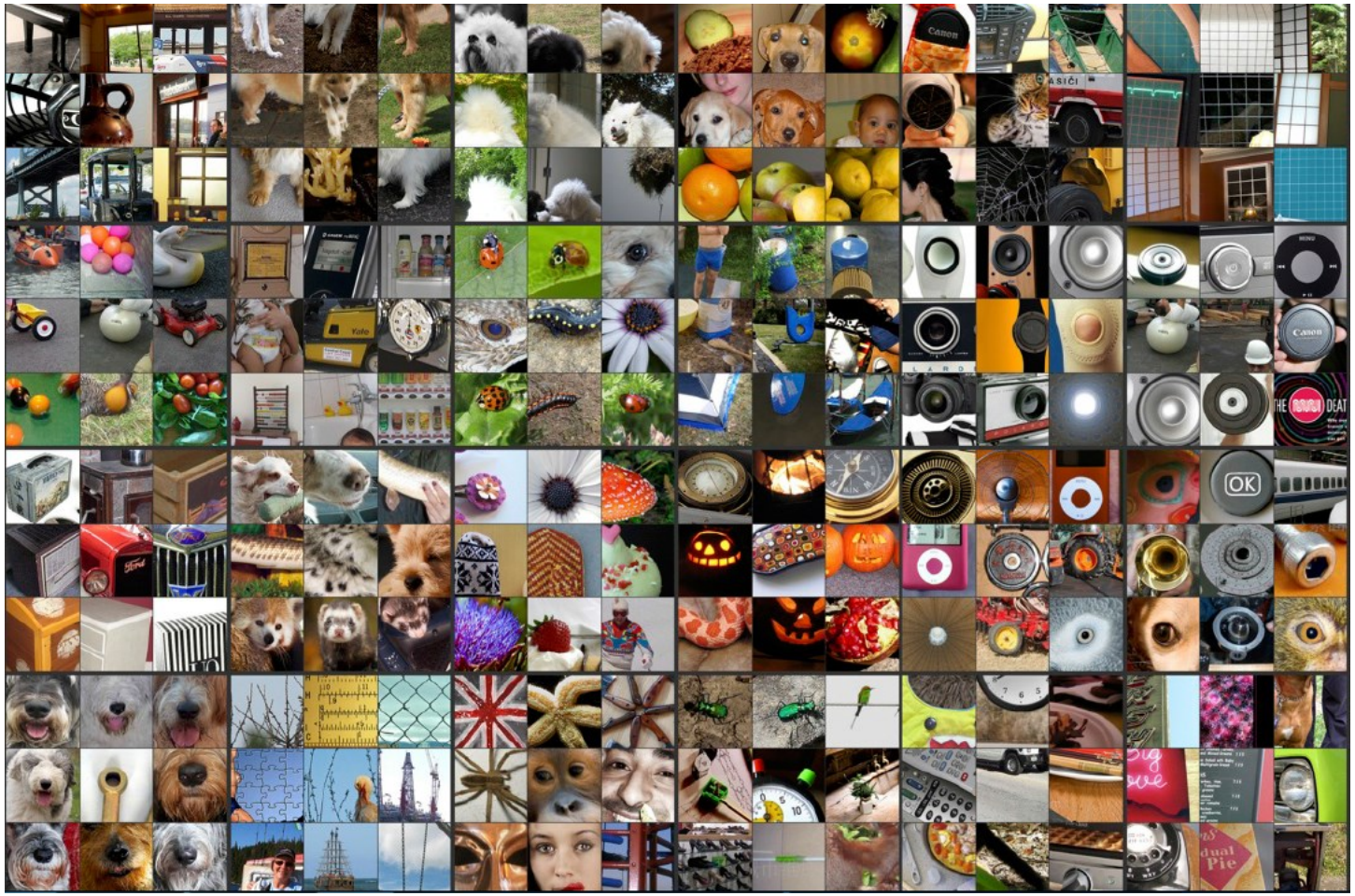
[Zeiler, Fergus 2014]

Визуализация свёрточных сетей

Оставшиеся слои – надо «спроецировать» на пиксели

Один из способов – искать патчи, дающие наибольший отклик

Слой 4:



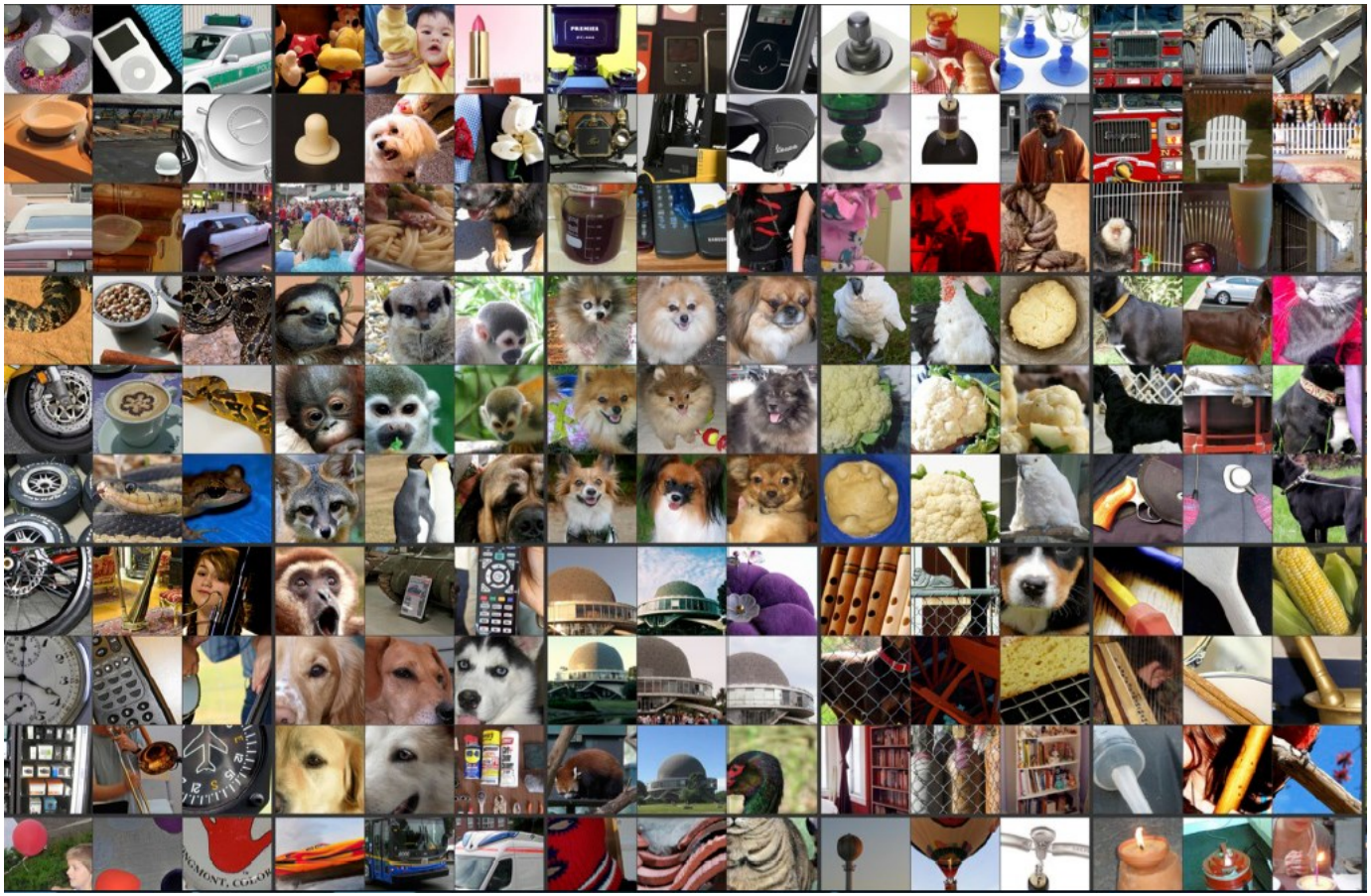
[Zeiler, Fergus 2014]

Визуализация свёрточных сетей

Оставшиеся слои – надо «спроецировать» на пиксели

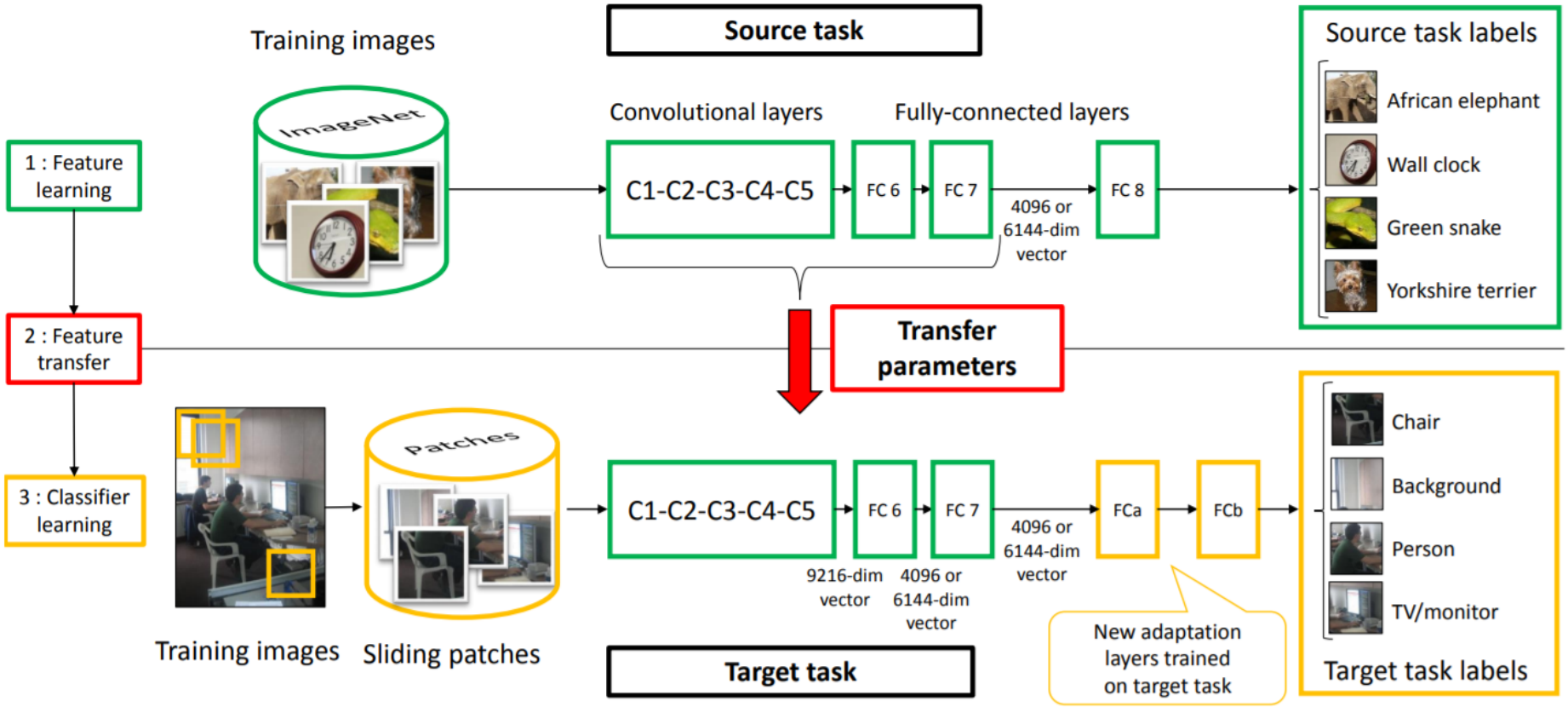
Один из способов – искать патчи, дающие наибольший отклик

Слой 5:



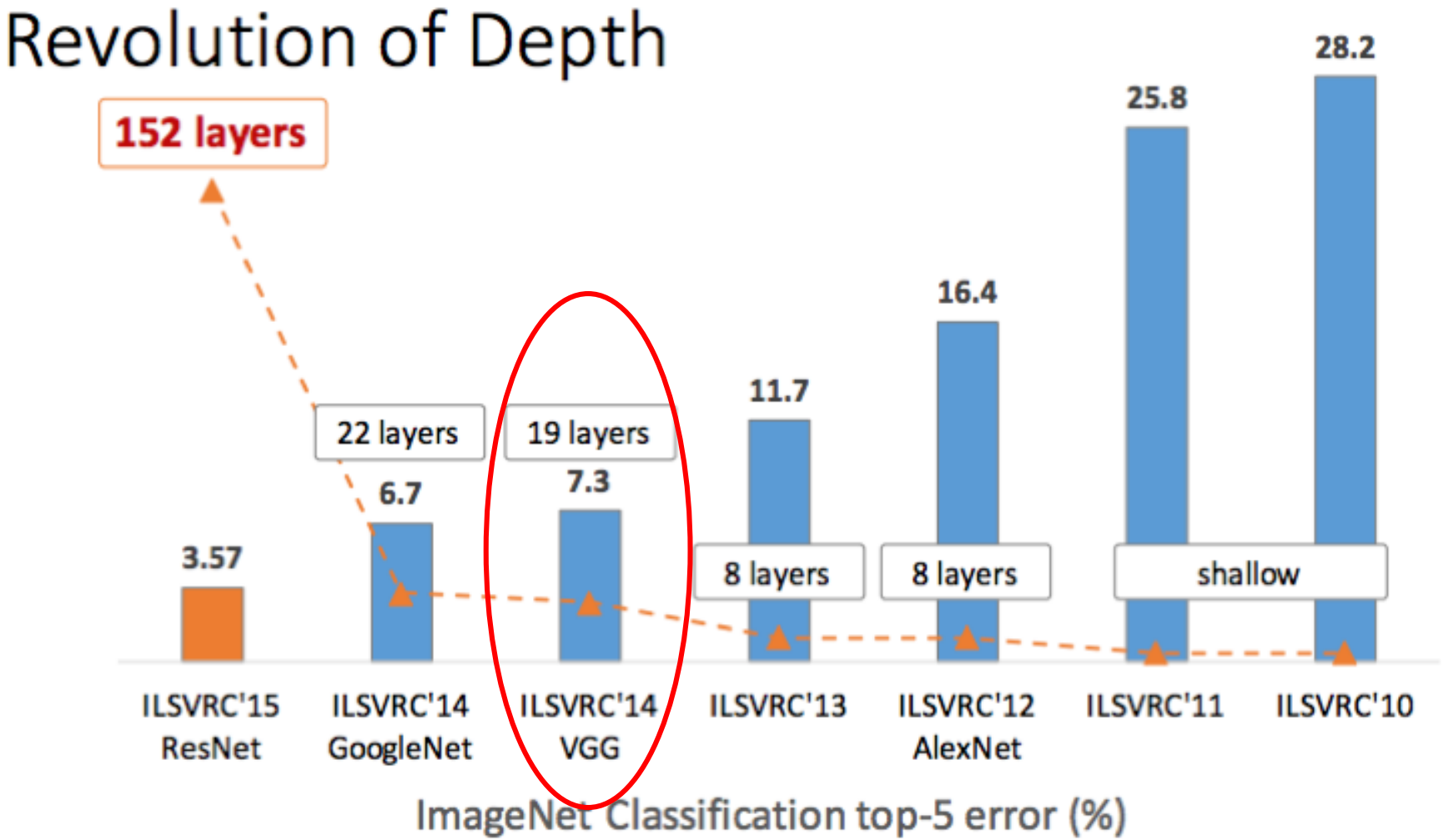
[Zeiler, Fergus 2014]

Перенос признаков CNN



[Oquab et al., 2014]

ImageNet challenge



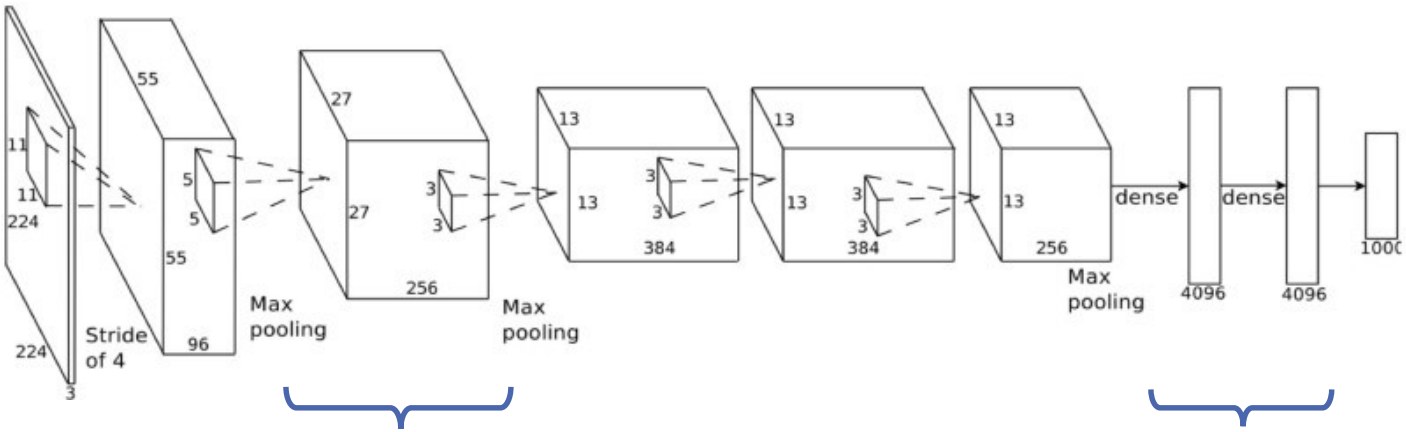
plot credit: Kaiming He

Архитектуры: VGG (2014)

- Каскад свёрток 3x3 вместо больших свёрток

(меньше параметров и быстрее, чем 7x7)

- 140M параметров (у AlexNet 60M)
- Более сбалансированные вычисления



Большинство вычислений

Большинство параметров

16 weight layers
224 × 224 RGB
conv3-64
conv3-64
maxpool
conv3-128
conv3-128
maxpool
conv3-256
conv3-256
conv3-256
maxpool
conv3-512
conv3-512
conv3-512
maxpool
conv3-512
conv3-512
conv3-512
maxpool
FC-4096
FC-4096
FC-1000
soft-max

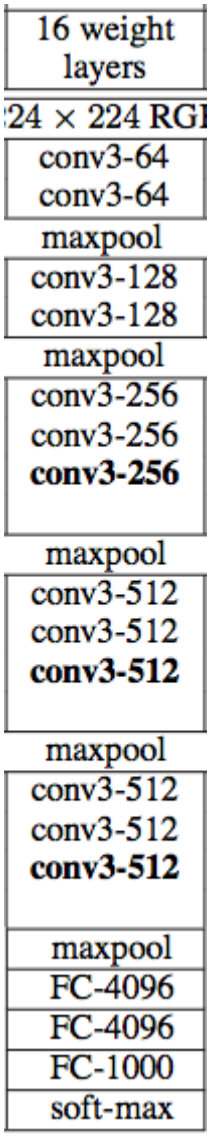
[Simonyan & Zisserman, 2014]

Архитектуры: VGG (2014)

- Каскад свёрток 3x3 вместо больших свёрток

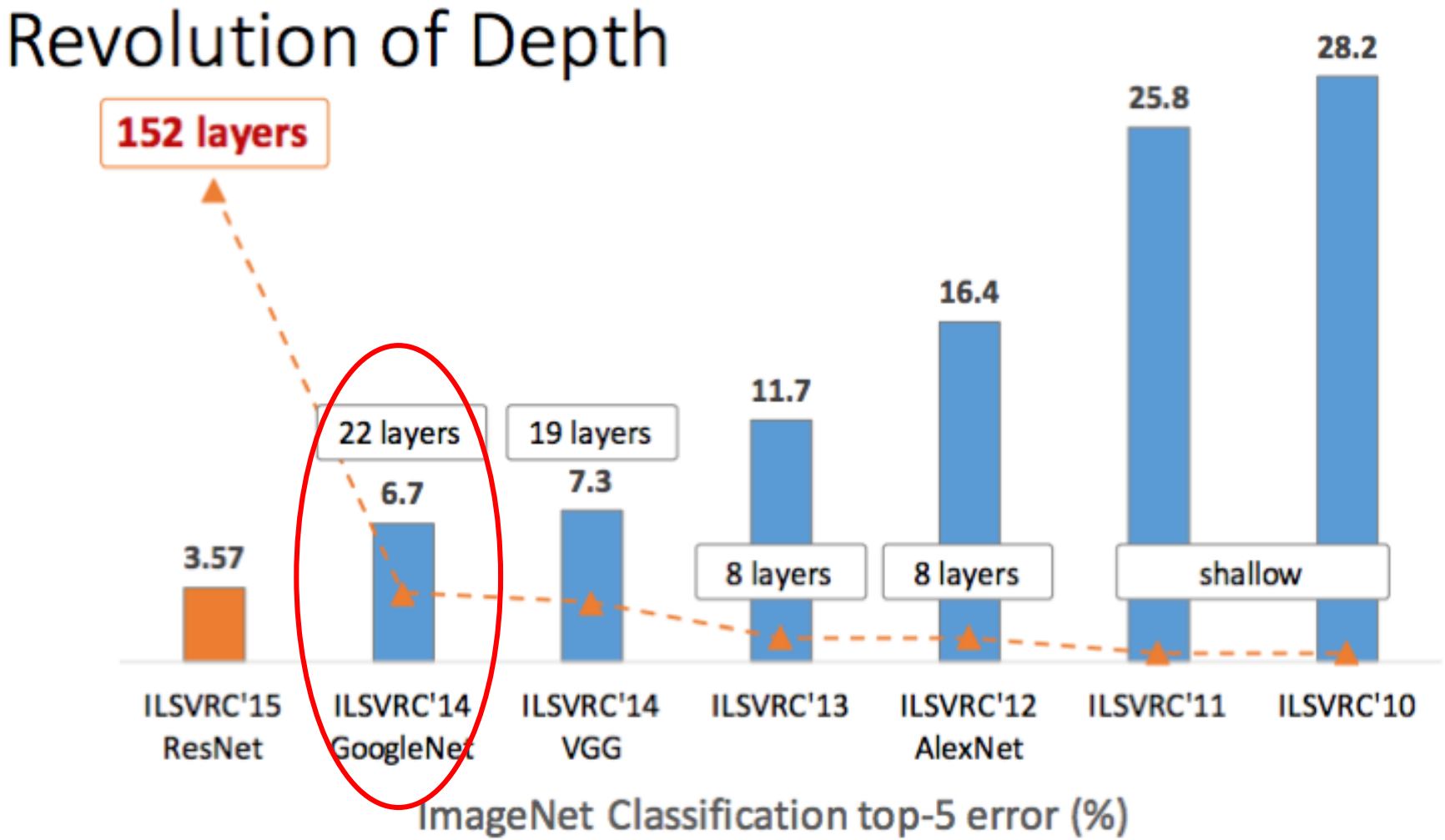
(меньше параметров и быстрее, чем 7x7)

- 140M параметров (у AlexNet 60M)
- Более сбалансированные вычисления
- Не обучается целиком (затухает градиент)
- Несколько стадий обучения разной глубины
- Обучение 4 Titan Black GPUs 2-3 недели



[Simonyan & Zisserman, 2014]

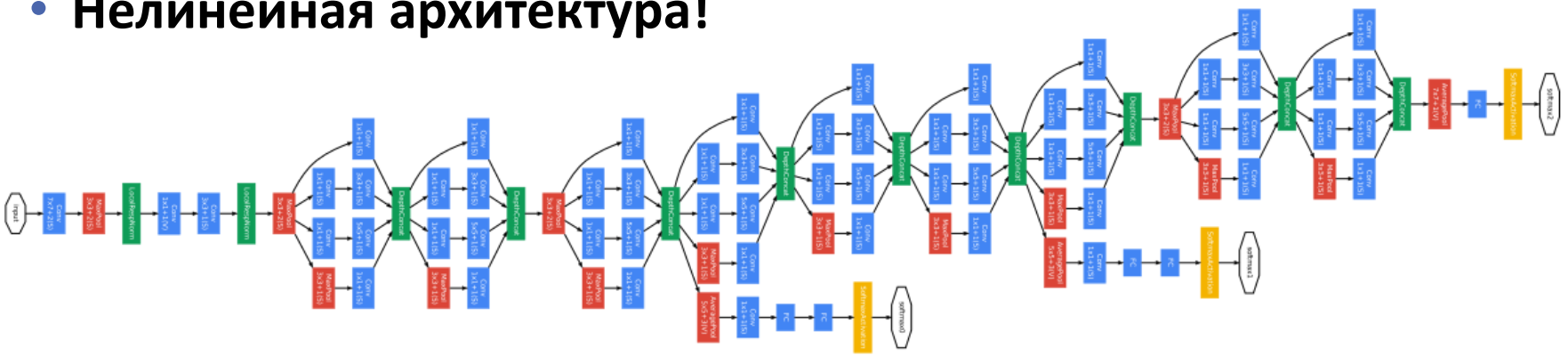
ImageNet challenge



plot credit: Kaiming He

Архитектуры: GoogleNet (2014)

- **Нелинейная архитектура!**

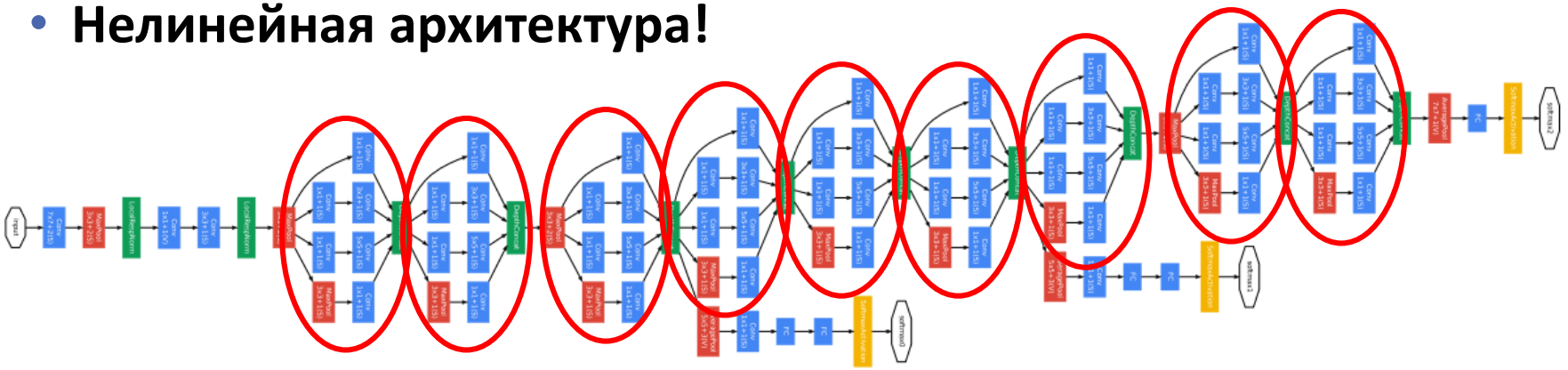


- Макс. глубина: 22 слоя с параметрами
- Нет полносвязных слоев
- 12x меньше параметров (чем в AlexNet)

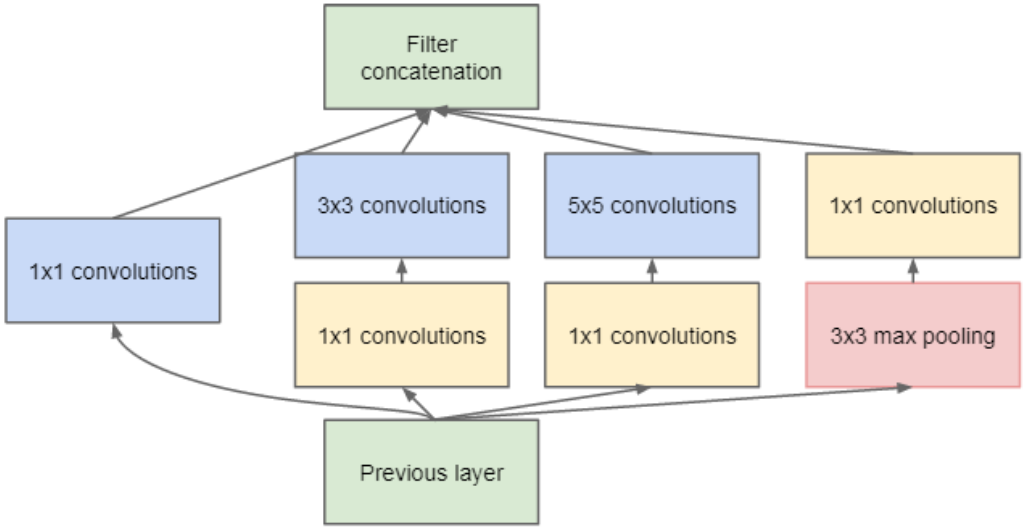
[Szegedy et al., 2015]

Архитектуры: GoogleNet (2014)

- **Нелинейная архитектура!**



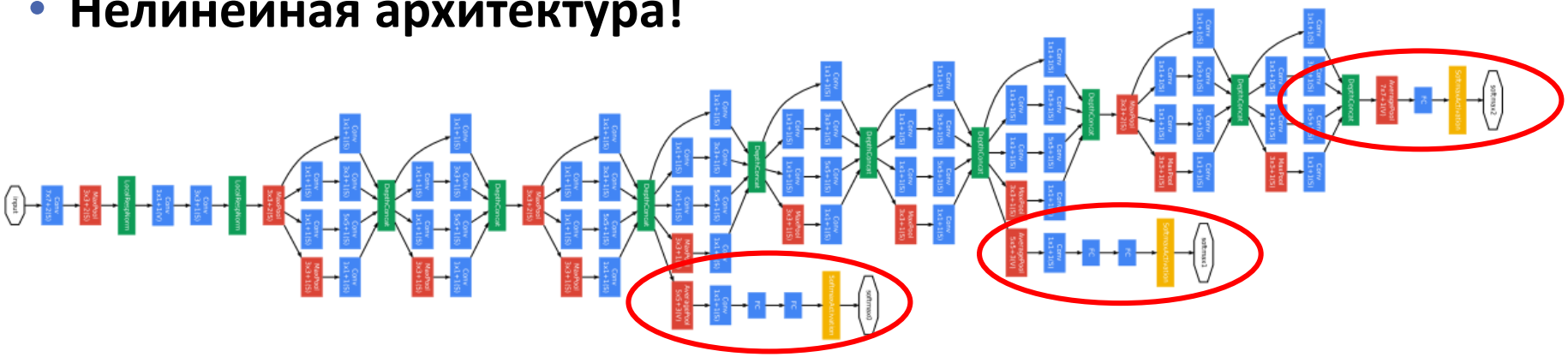
- Основной блок – Inception module (9 штук)
- 1x1 свёртки – уменьшение размерности



[Szegedy et al., 2015]

Архитектуры: GoogleNet (2014)

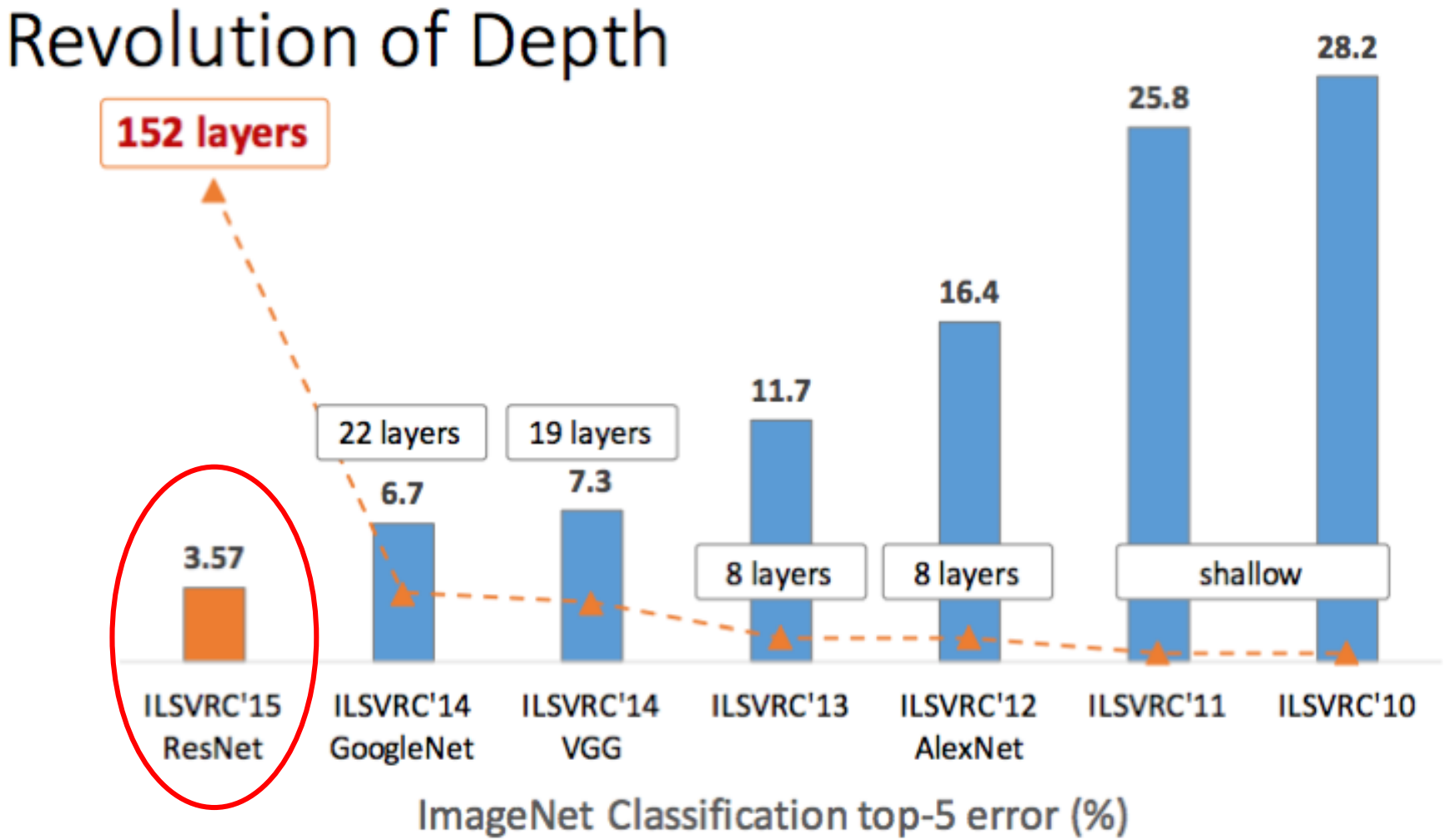
- **Нелинейная архитектура!**



- Очень глубокая сеть, целиком не обучается
- Один и тот же блок с функций потерь в нескольких местах
- «Проталкивает» градиент внутрь сети
- Обучалось на облаке CPU
- Добавился BatchNorm, Residual blocks и т.д. (Inception-v4)

[Szegedy et al., 2015]

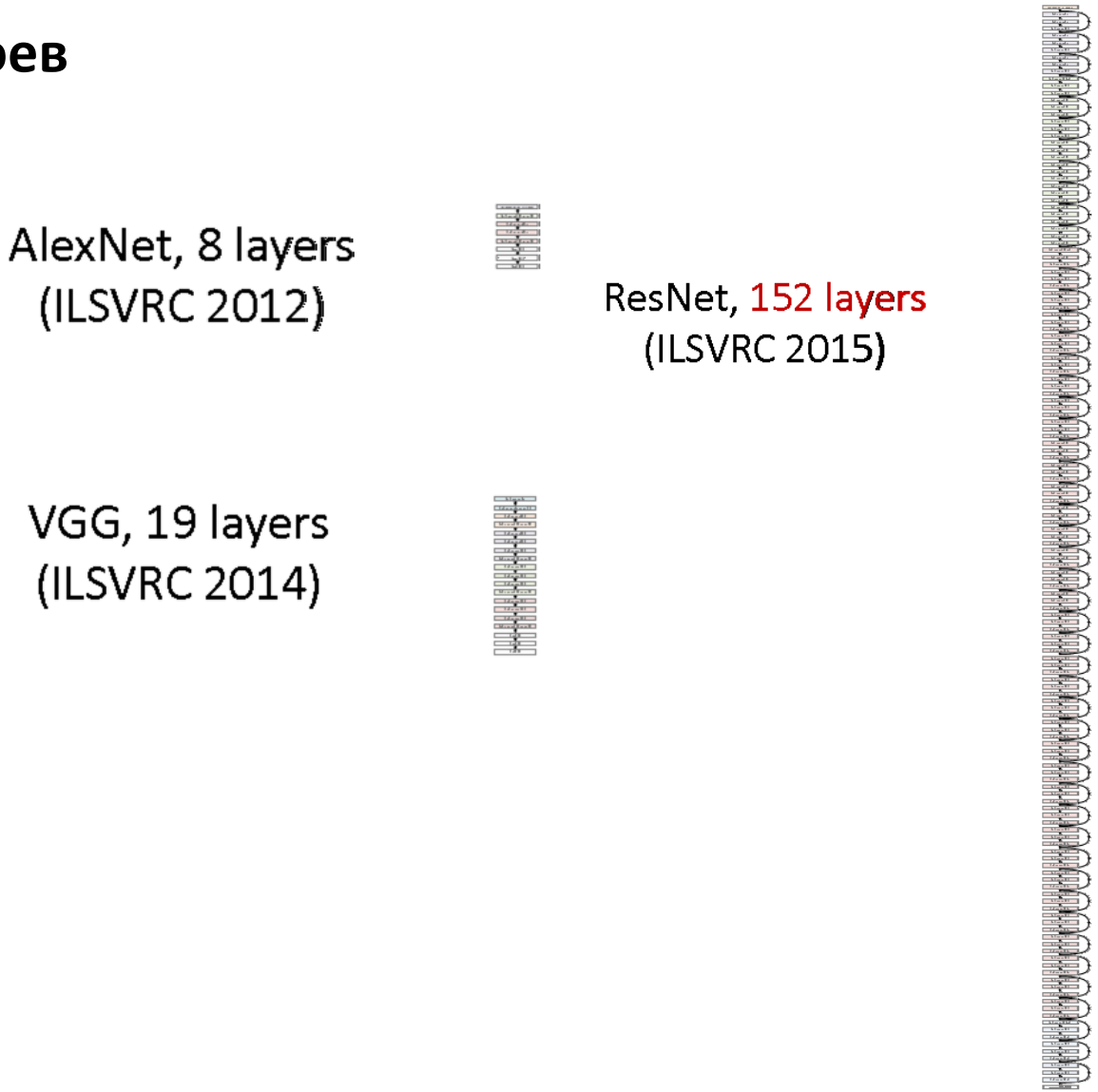
ImageNet challenge



plot credit: Kaiming He

Архитектуры: ResNet (2015)

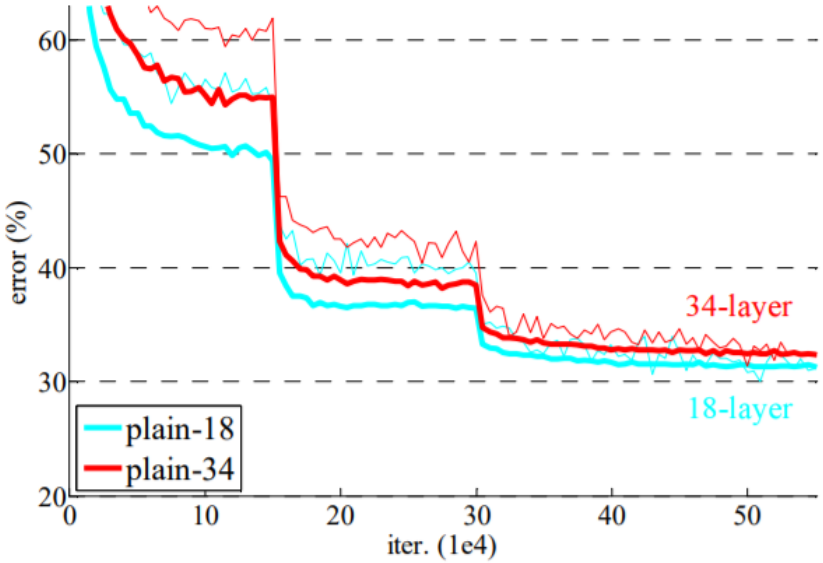
- **Ultra deep! 100+ слоев**



[He et al., 2015]

Архитектуры: ResNet (2015)

- **Ultra deep! 100+ слоев**
- Просто добавление слоев не работает



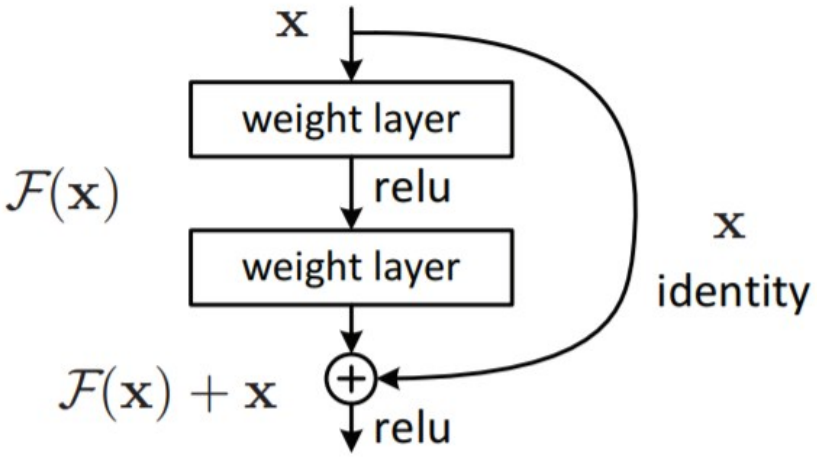
[He et al., 2015]

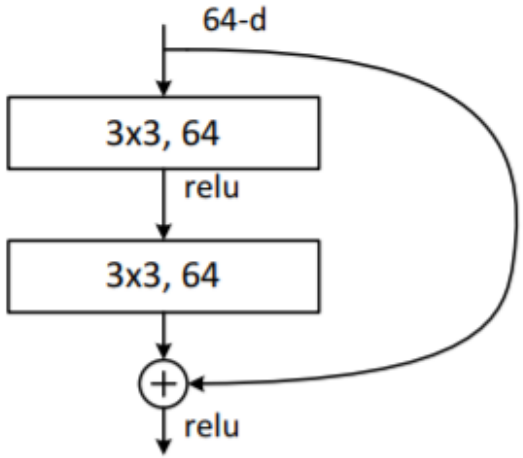
Архитектуры: ResNet (2015)

- **Ultra deep! 100+ слоев**
- Просто добавление слоев не работает
- Основная идея – остаточный блок

добавление тождественной связи

- Пропускает градиент вглубь





[He et al., 2015]

Заключение

- Свёрточные сети перевернули компьютерное зрение
- Основной инструмент – обучение с учителем
- Признаки переносятся на другие задачи!
- На следующей неделе – применения сетей для других задач зрения
- Для обучения свёрток нужны GPU (вывод можно и на CPU)
- Есть стандартные библиотеки и зоопарки моделей
- Обучать сети с нуля долго и сложно
- Дообучать (fine-tuning) гораздо проще!
- Полезный трюк – заморозить почти все слои