

Глубокое обучение

Лекция 5: Transfer learning

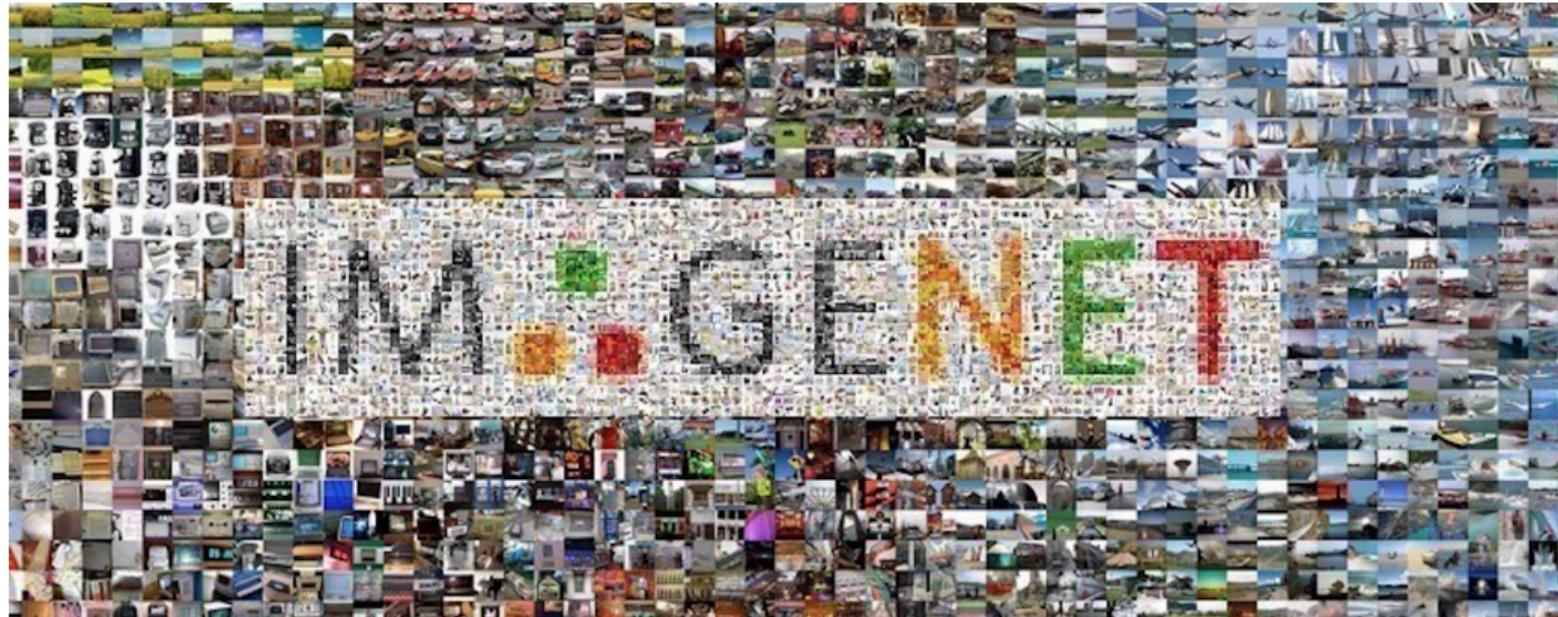
Agenda

- Соревнование ImageNet
- Различные архитектуры сверточных сетей
- История про Метрику
- Transfer learning

Соревнование ImageNet

ImageNet

- около 10 миллионов размеченных изображений из интернета



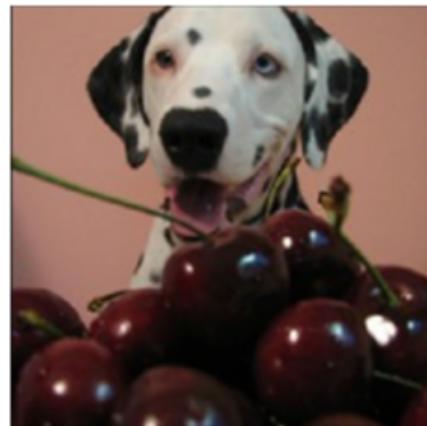
<http://www.image-net.org/>

ImageNet

- выборка очень большая и неоднородная, постоянно пополняется, соревнования на ней проводятся каждый год
- обычно изображение требуется отнести к одному из 1000 классов, можно давать несколько ответов
- если один из пяти вариантов оказался верным, то классификация считается верной
- до 2012 года лучшие алгоритмы дают ошибку в 25%
- в 2012 году на арену выходят глубокие нейронные сети

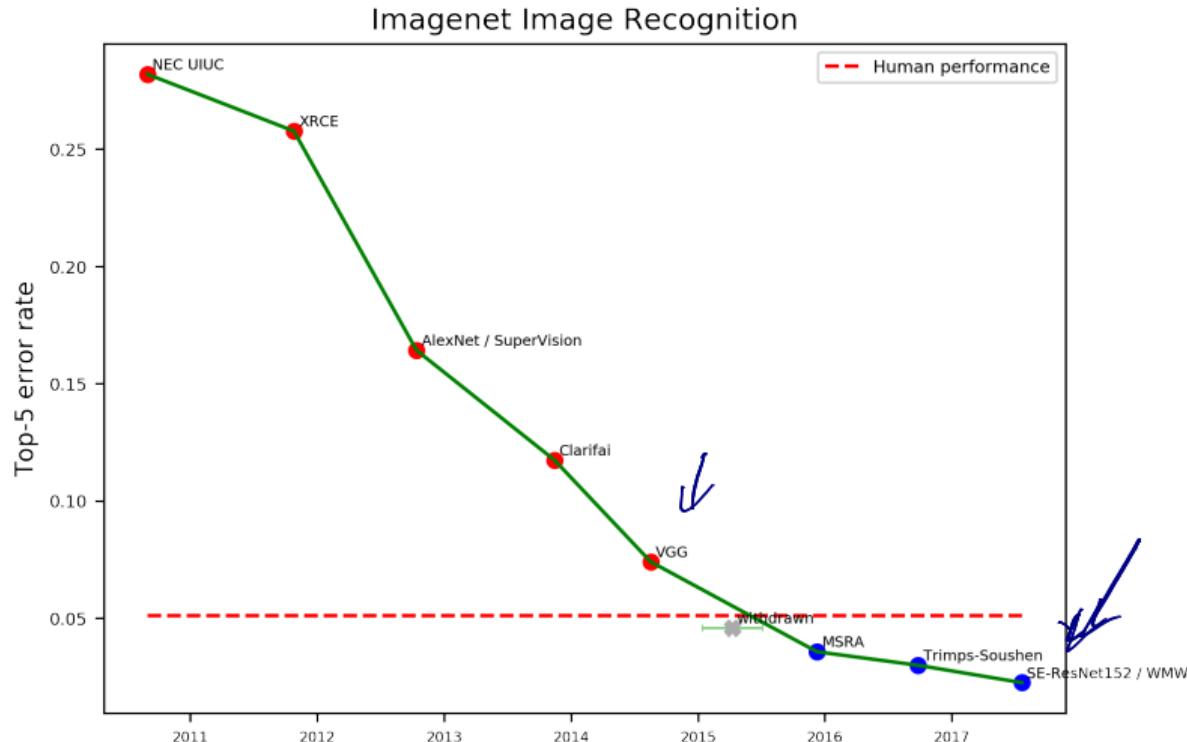
ImageNet

- бывают спорные изображения: тут вишня, если распознать как далматинец, будет неправильно



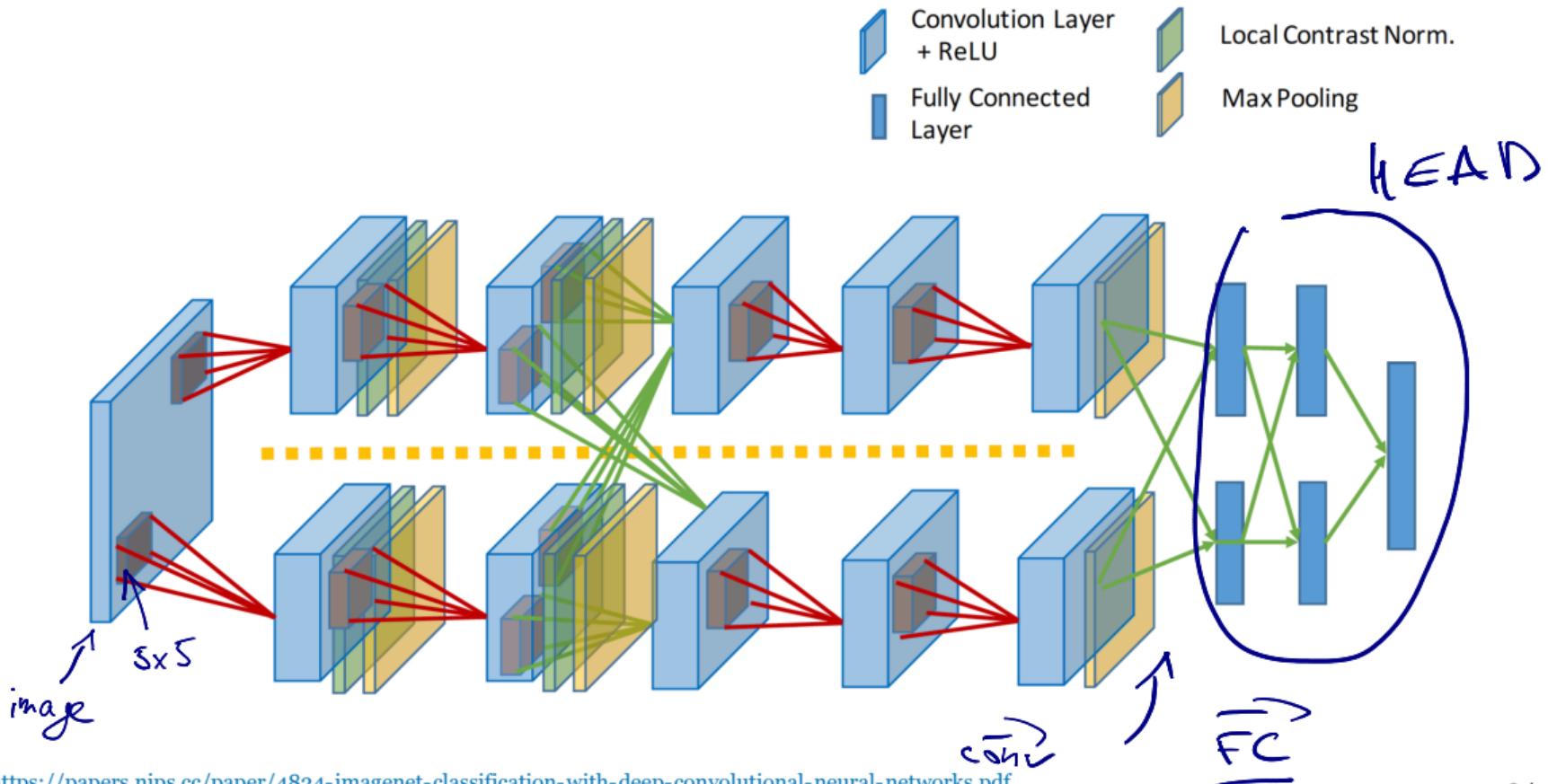
- Можно попробовать сразиться с компьютером:
<https://cs.stanford.edu/people/karpathy/ilsvrc/>

Точность сетей на ImageNet



<https://www.eff.org/ai/metrics>

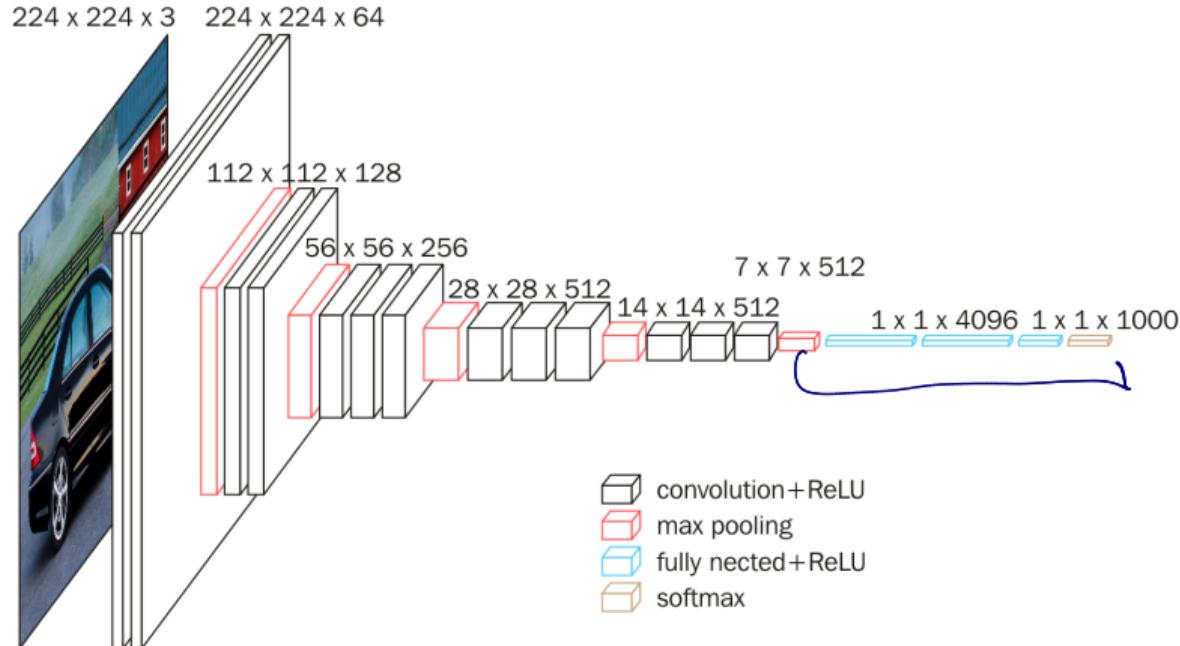
AlexNet (2012)



AlexNet (2012)

- Тот же LeNet из 1998г. но увеличен в 1000 раз и дополненный трюками (Dropout, ReLU, Data augmentation)
- Свёртки 11×11 , 5×5 , 3×3
- Уронила ошибку с 25% до 15.4%
- 60 миллионов параметров
- Училась 6 дней на 2 GPU
- Попробовали сделать ансамблю из таких сеток, ошибка упала до 11.7%

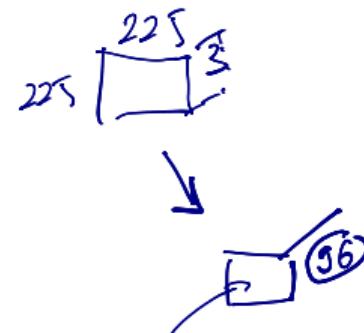
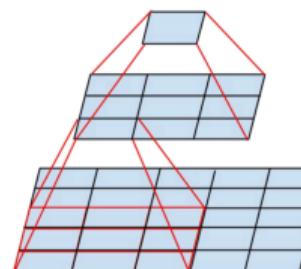
VGG (2014)



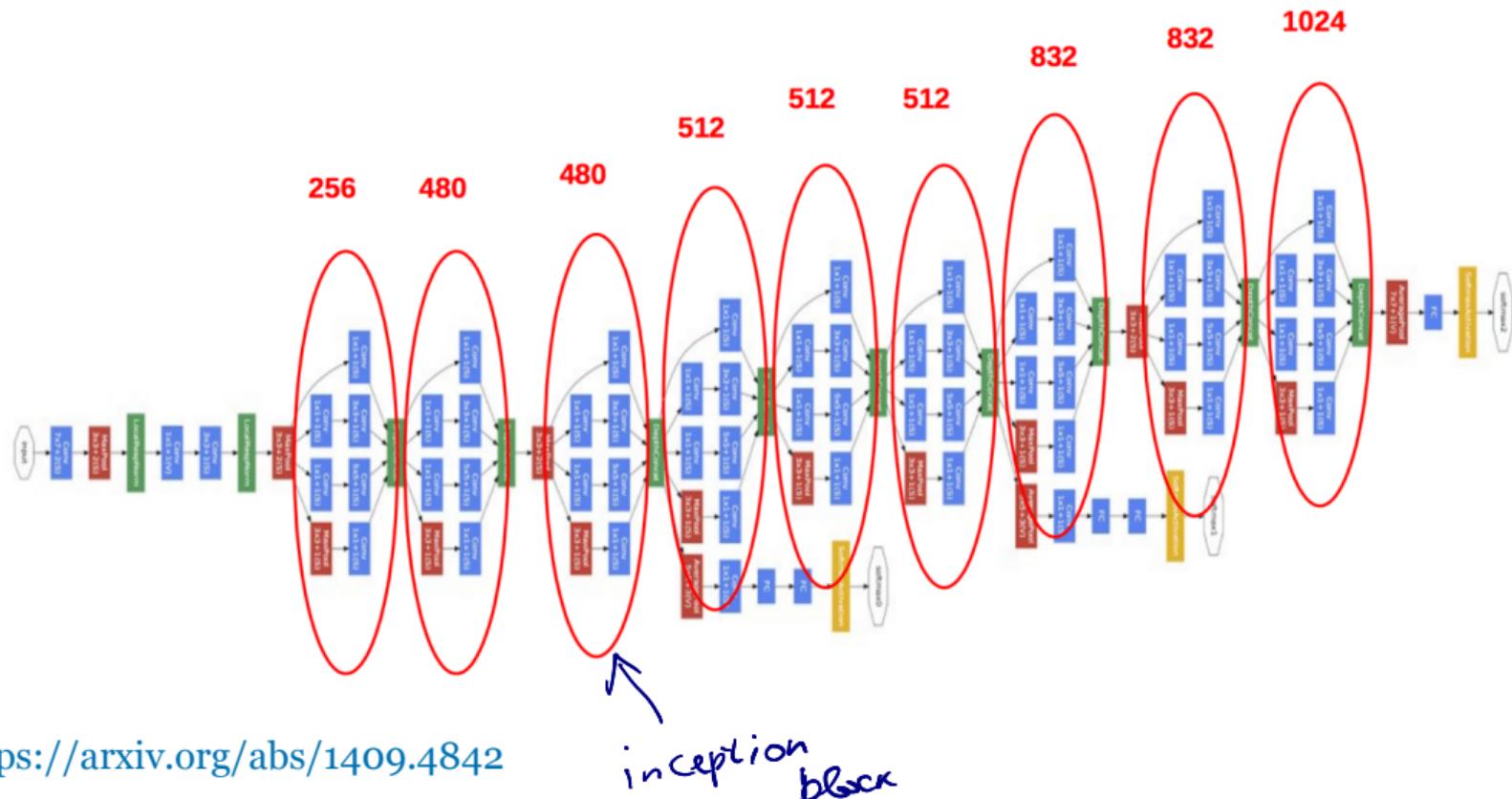
https://www.datalearner.com/paper_note/content/300035

VGG (2014)

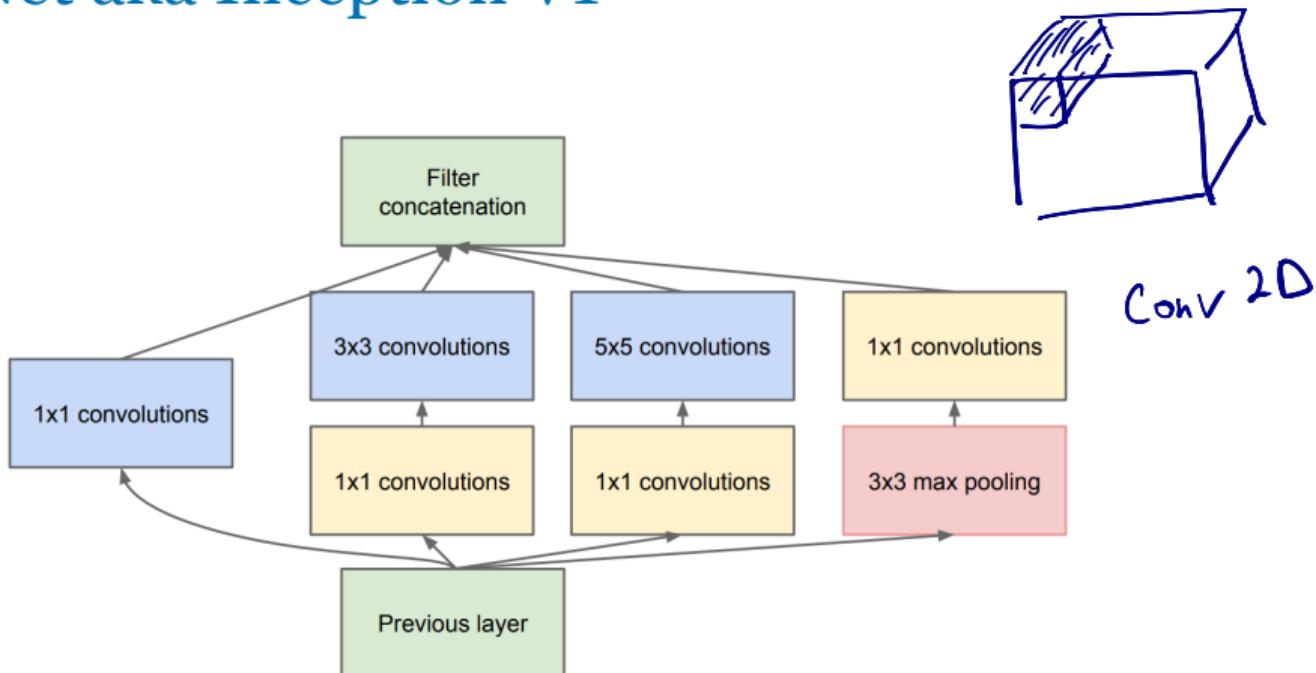
- 138 миллионов параметров
- Училась 2 – 3 недели на 4 GPU
- Ошибка упала до 6.8%
- Свёртки только 3×3 , но намного больше фильтров для экономии параметров (две свёртки 3×3 покрывают поле 5×5 и требуют 18 параметров, а свёртка 5×5 требует 25 параметров)



GoogleLeNet aka Inception V1 (2014)



GoogleLeNet aka Inception V1



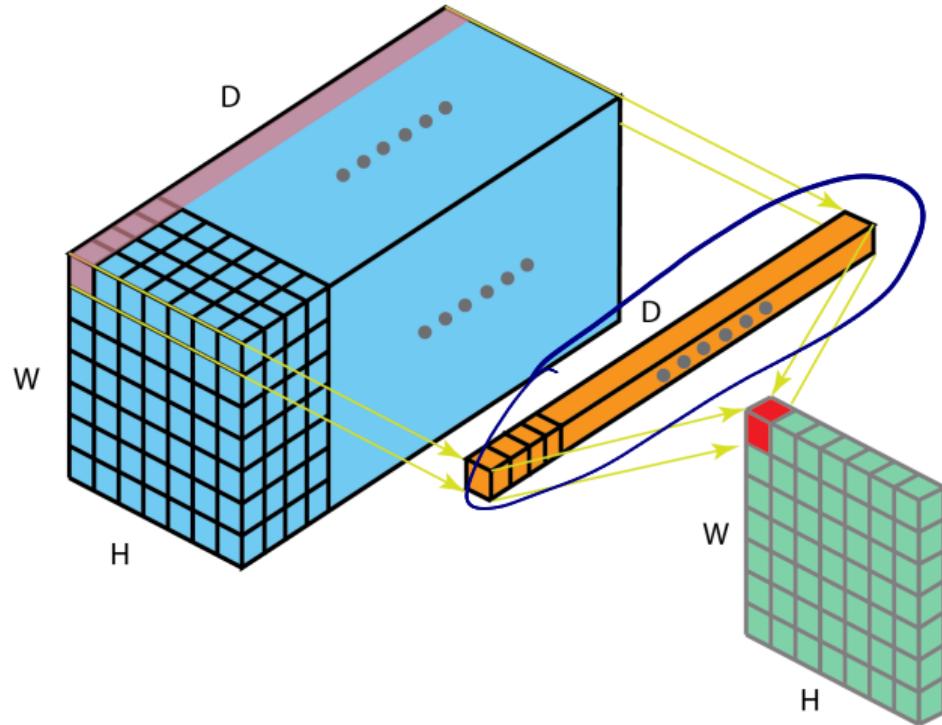
(b) Inception module with dimension reductions

GoogleLeNet aka Inception V1 (2014)

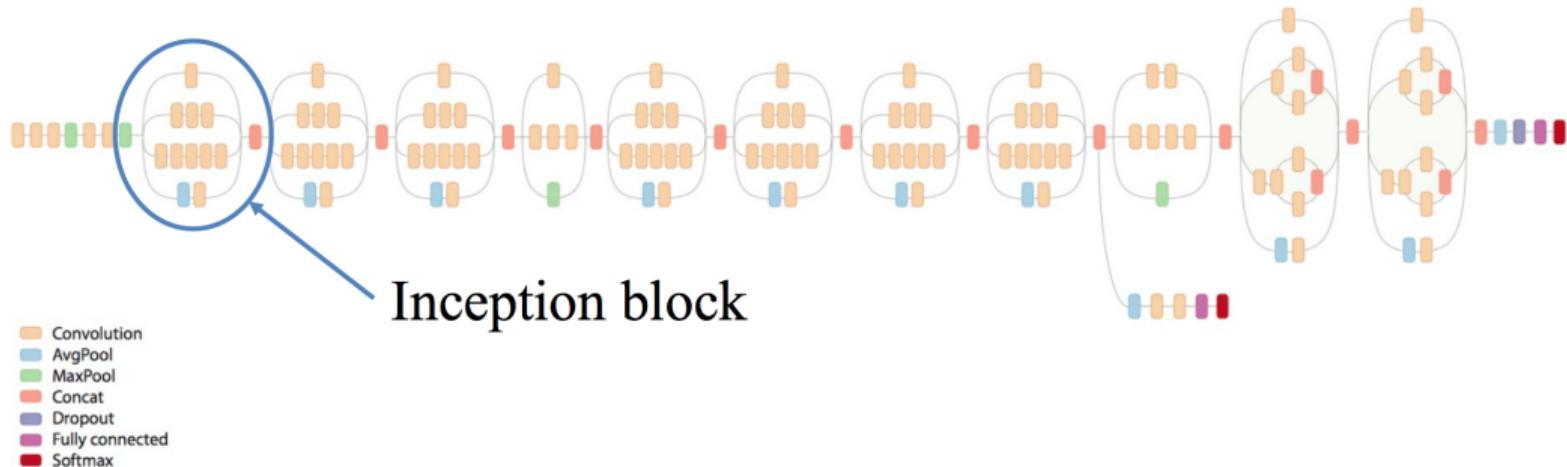
- Парни из Google решили усовершенствовать AlexNet.
- Уменьшаем свёртки, саму **сетку собираем из компонент** (всего 9 блоков).
- На каждом слое используется ни одна свёртка, а несколько разных, что помогает реагировать на сигналы разного масштаба и улучшает работу, используются **свёртки 1×1** .
- **Несколько дополнительных классификаторов на разных уровнях.** Идея в том, что такие классификаторы позволяют «протолкнуть» градиенты к ранним слоям и тем самым уменьшить эффект затухания градиента.
- Параметров в 10 раз меньше, чем в AlexNet, работает лучше, 6.7% ошибок

Свёртка 1×1

- Позволяет сократить размерность по числу каналов
- Представляет из себя полносвязный слой по фильтрам
- В Inception мы делаем много разных свёрток, а потом свёрткой 1×1 выбираем из них лучшее, агрессивно понижая размерность

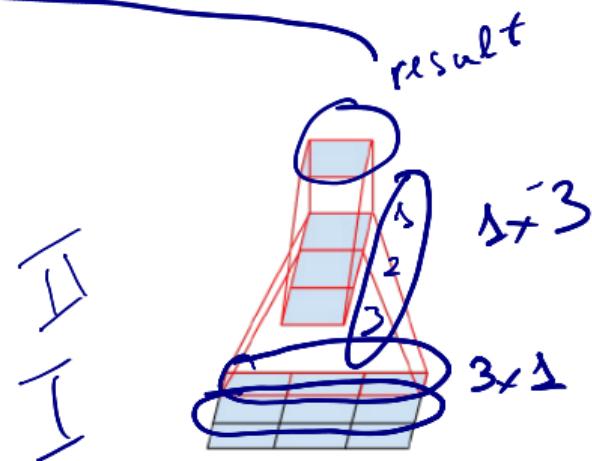
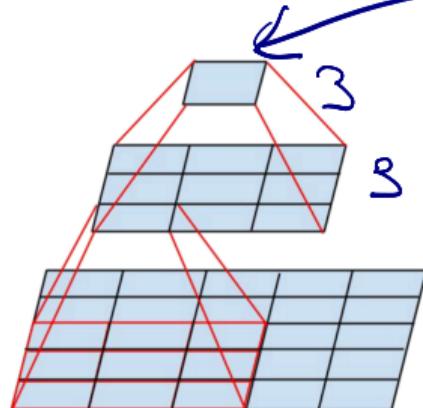


Inception V3 (2015)



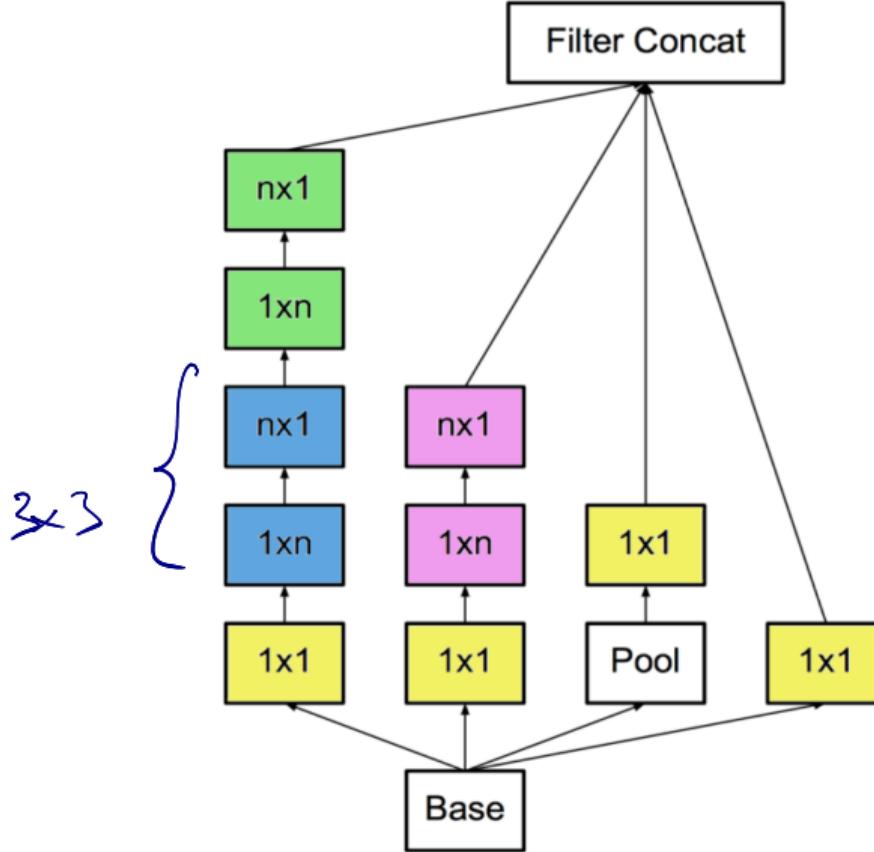
<https://arxiv.org/abs/1512.00567>

Свёртка $n \times 1$ и $1 \times n$



- В VGG заменяли большие свёртки на последовательные 3×3 для экономии
- Пойдём дальше и заменим их на последовательные 3×1 и 1×3
- Экономим ещё больше параметров, для каждой свёртки 0 вместо 9

Inception V3 (2015)

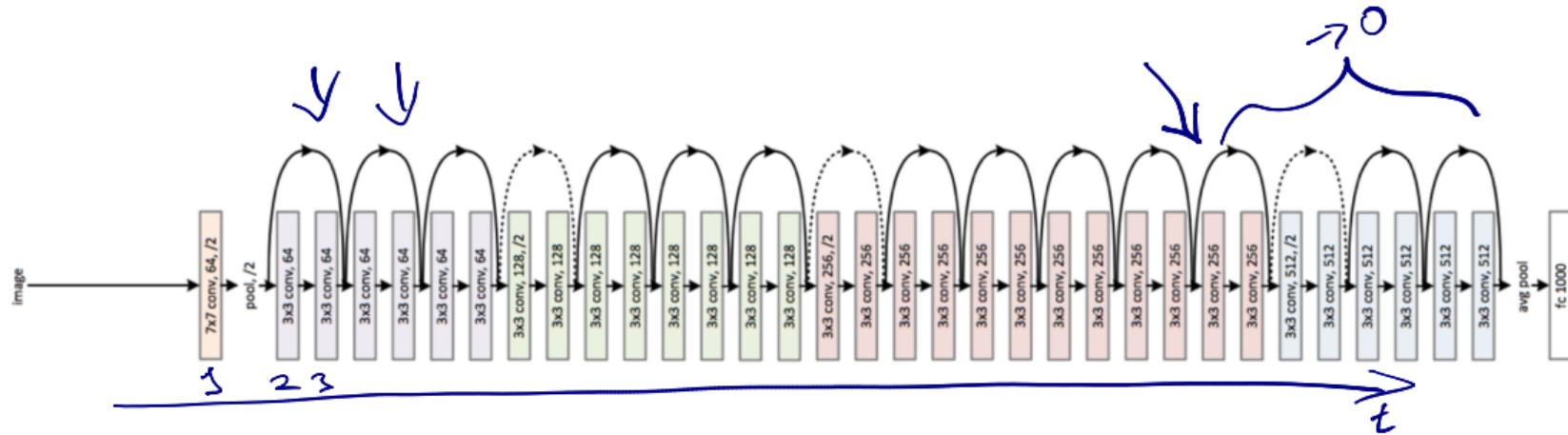


Inception V3 (2015)

- Сетка собирается из 11 inception layers, добавили BatchNorm (то же самое без него Inception V2)
- В результате исследований сформулировали принципы обучения глубоких свёрточных сетей:
 1. Избегайте representation bottlenecks, нельзя резко снижать размерность слоя, это надо делать плавно: от начала к концу
 2. Свёртку нужно разбивать на более мелкие части для экономии ресурсов и увеличения размера сетки
 3. Нельзя резко увеличивать глубину, забивая на ширину, надо растить сбалансированно
- Итоговое качество 4.2% ошибок, ансамбль из 4-х моделей дал 3.8%.

ResNet (Microsoft) (2015)

Residual Network



- 152 слоя, ошибка составила 3.75%, ансабль из сеток дал 3.57%

<https://arxiv.org/abs/1512.03385>

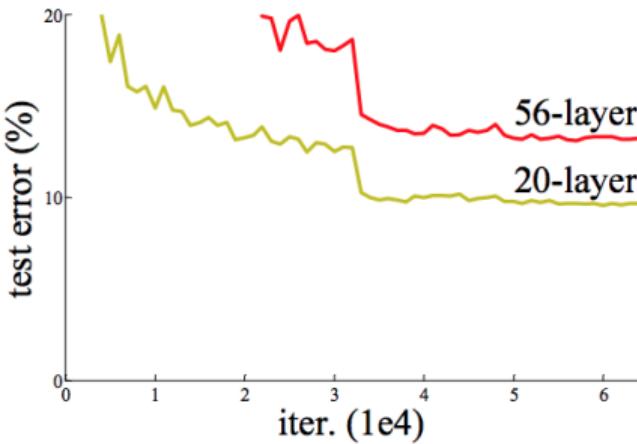
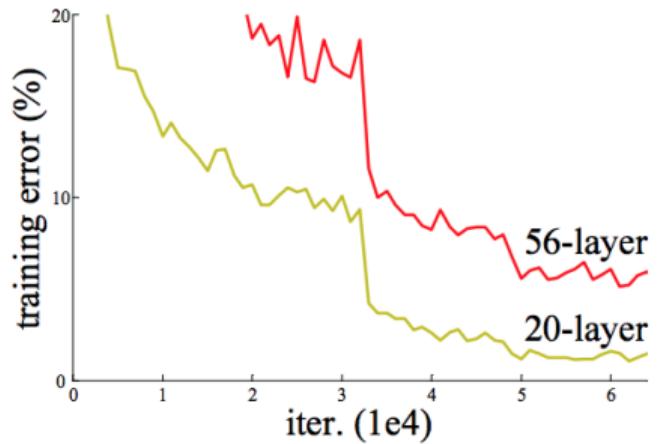
$$x_0 \rightarrow f(x_0) + x_0 \rightarrow f_2(x_1) + x_1$$

$\underbrace{\hspace{10em}}$
 x_2

$$(x_0 + f_1(x_0)) + f_2(x_0 + f_1(x_0)) + f_3(x_0 + f_1(x_0) + f_2(x_0 + f_1(x_0))) + \dots$$

ResNet

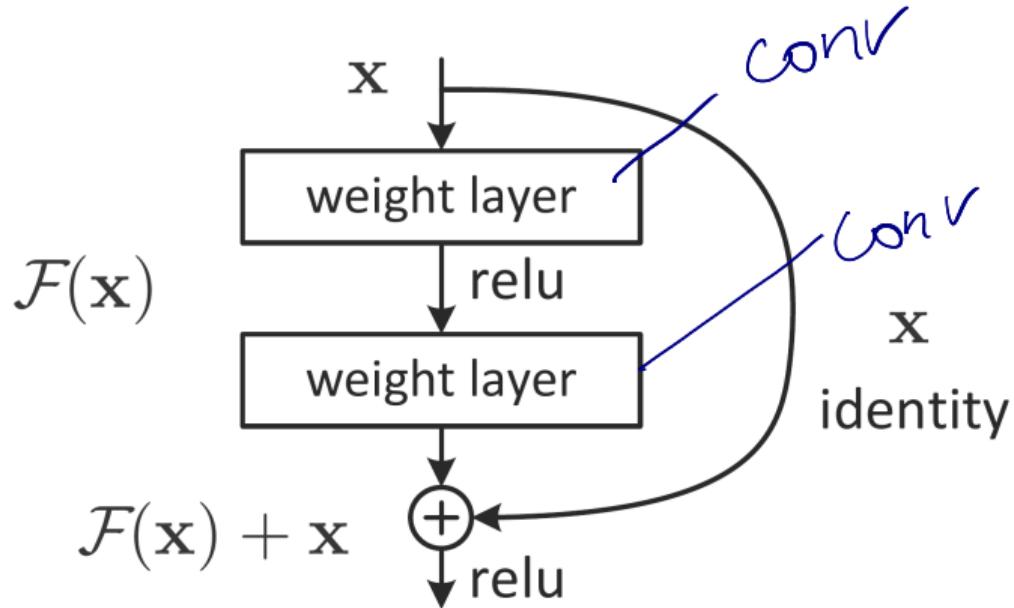
$$\cancel{x_{n+1}} = x_n + \ell(x_n)$$



ResNet

- VGG из 20 слоёв и VGG из 56 слоёв, большая сеть обучается хуже
- **Проблема:** слои инициализированы шумом, если какой-то один слой не натренирован, он убивает работу сети, через него не проходит полезный сигнал
- Чем больше слоёв, тем более ярко выражен этот эффект
- **Решение:** Будем посыпать вход на выход и давать слою возможность немного его подправить (residual слой)
- Идея чем-то похожа на бустинг, сеть сама решает когда заканчивать подправлять выходы (грубо говоря, сама выбирает глубину)

ResNet (Microsoft) (2015)



<https://arxiv.org/abs/1512.03385>

ResNet (Microsoft) (2015)

- **Идея:** более глубокие уровни должны улавливать разницу между новым и тем, что было раньше
- Ключевым элементом архитектуры является связь, которая пропускает несколько слоёв, передавая результат предыдущего слоя
- Такое изменение позволило полностью отказаться от таких техник регуляризации, как DropOut
- Градиенты не взрываются, свойства ResNet активно пытаются сейчас изучать

Что было дальше?

- В 2016 году Google попробовал Inception-Resnet и поставил ансамблем новый рекорд, 3.08%

<https://arxiv.org/abs/1602.07261>

- После Google задумались о переносе своих огромных архитектур в мобильные телефоны и стали работать над их компактностью

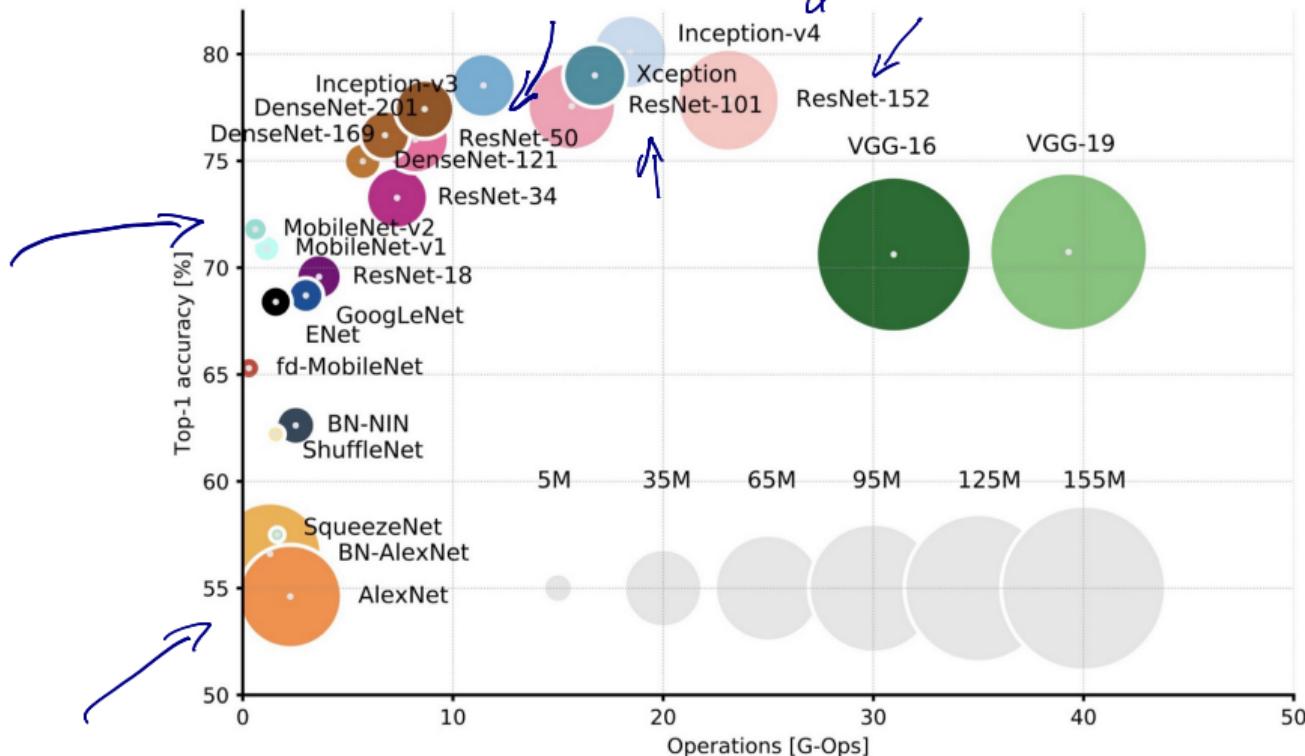
<https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html>

- В целом область развивается очень динамично и всплывает куча идей, на хабре иногда есть рубрика "Читаем статьи за вас", там можно посмотреть сколько идей возникло только за осень 2017

<https://habr.com/ru/company/ods/blog/343822/>

Слайд из первой презентации

2. Сложность сетей растёт



<https://towardsdatascience.com/neural-network-architectures-156e5bad51ba>

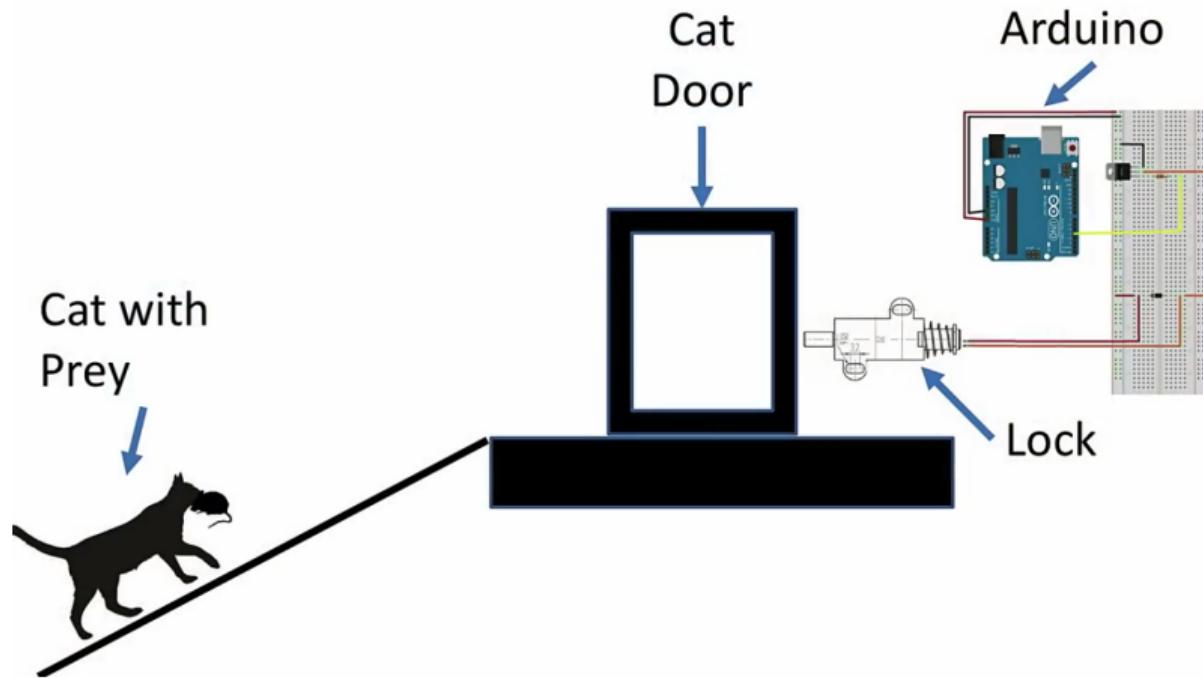
История про Метрику

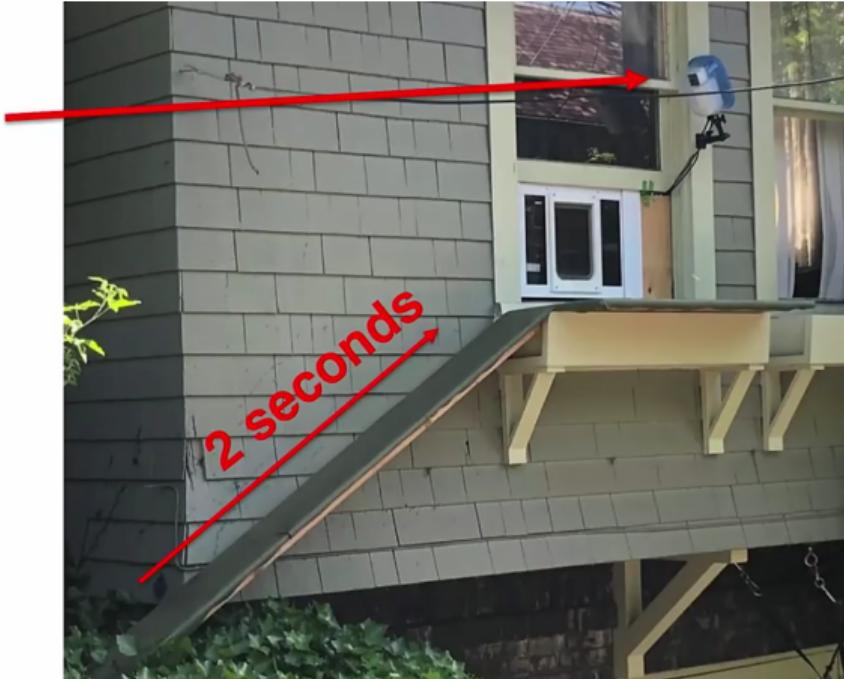
Метрика - это кот



<https://www.youtube.com/watch?v=1A-Nf3QIJjM>





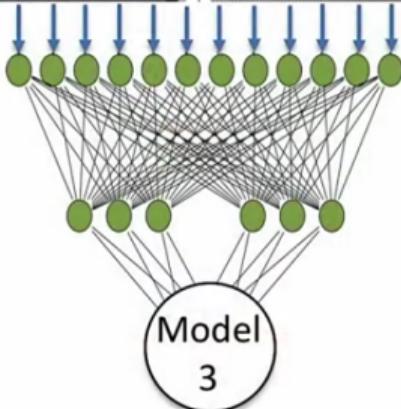
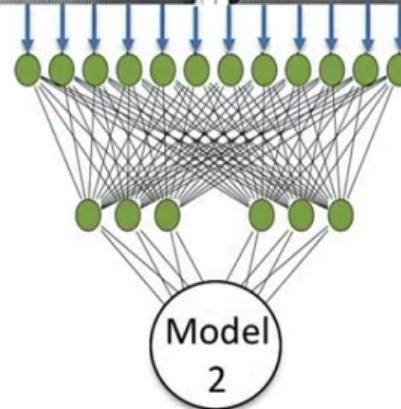
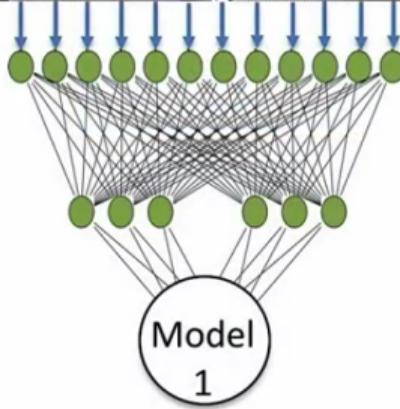
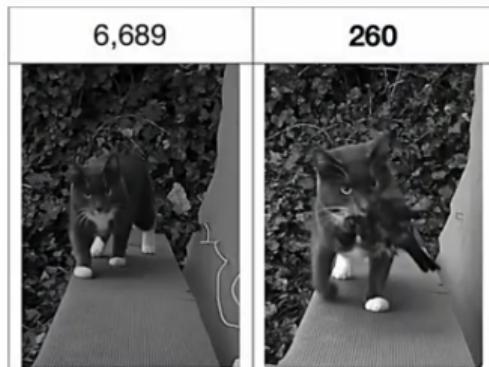
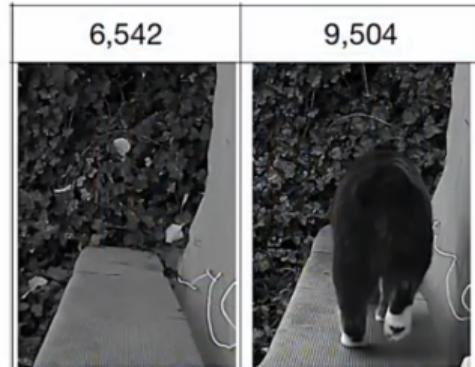




VS.



Image Type	No Cat	Cat not on approach	Cat on approach	Cat with prey
Count of Images	6,542	9,504	6,689	260
Example				



CRITTER DETECTED!

Step 1:
Lock the door!



Step 2:
Text me pics

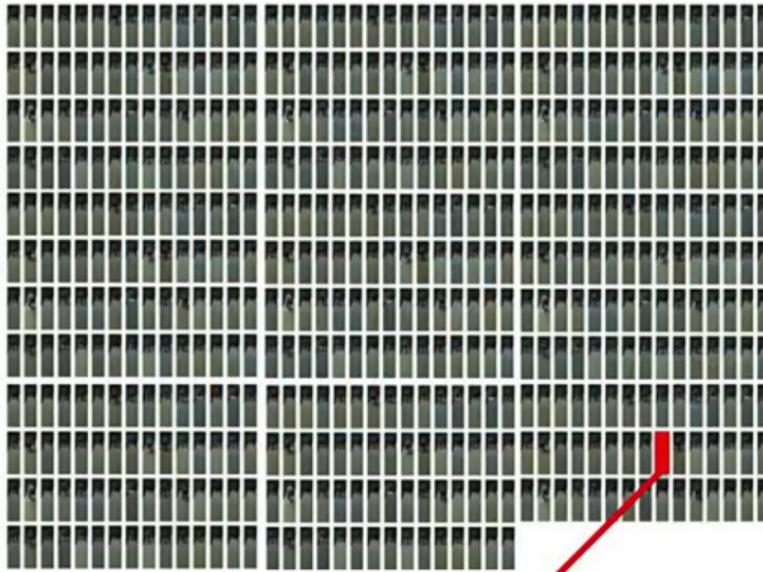


Step 3:
Donate blood
money



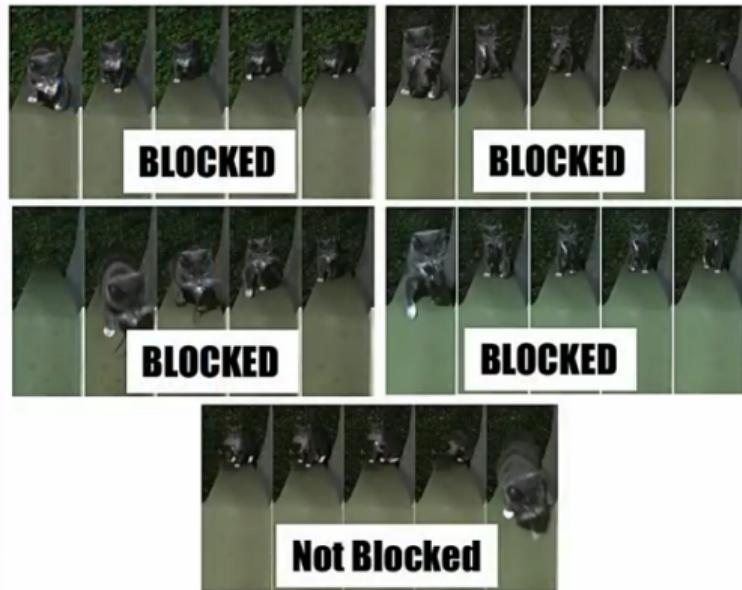
Audubon

Innocent Entries



1 Unfair Lockout

Entries with Prey



4/5 Critters Blocked

Как он это сделал?

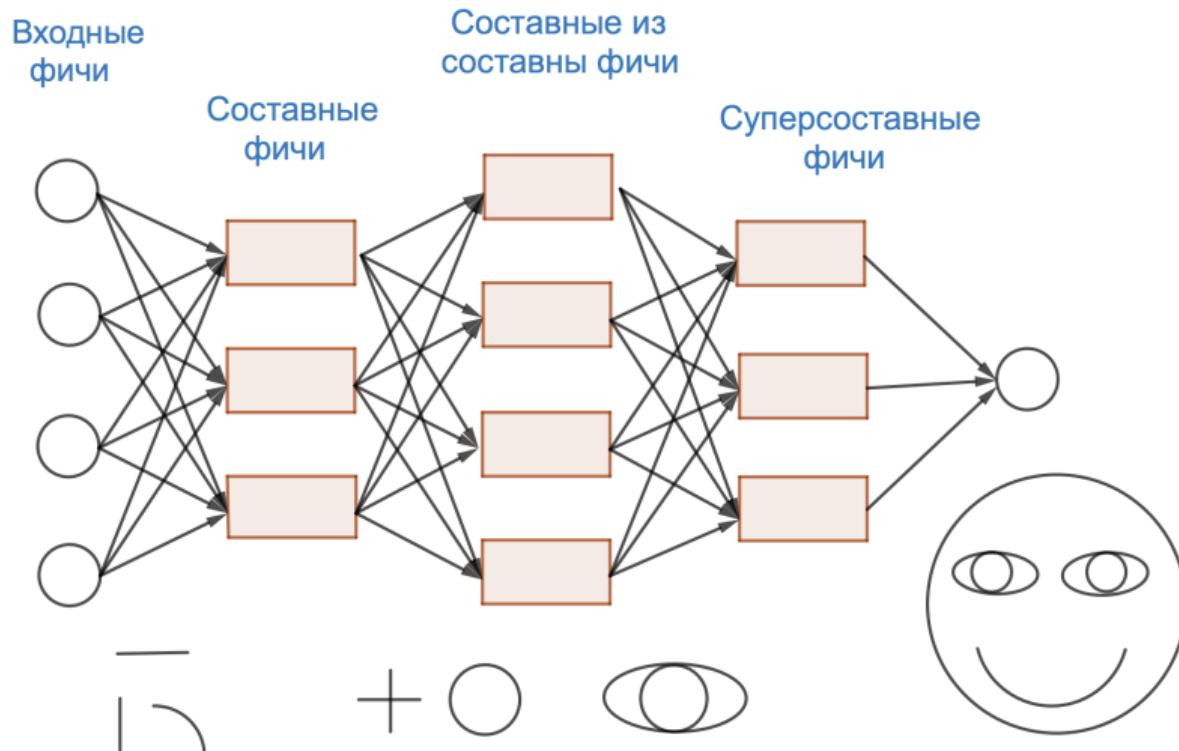
- В выборке было всего лишь 260 фотографий кота с добычей, неужели этого хватило для обучения сетки?
- На самом деле сетку с нуля никто не учил, делался transfer learning

Transfer learning

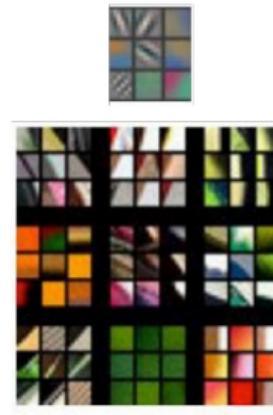
Как он это сделал?

- На практике свёрточные сети с нуля обучаются только огромные компании
- Это происходит из-за ограниченности ресурсов
- Уже обученные архитектуры пытаются адаптировать под новые задачи, это называется **transfer learning**

Что выучивают нейросети



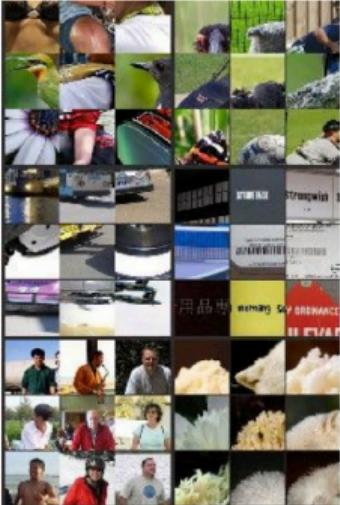
Что выучивают нейросети



Layer 1



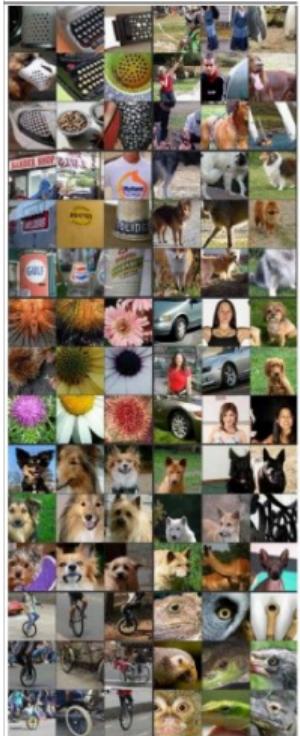
Layer 2



Layer 3



Layer 4

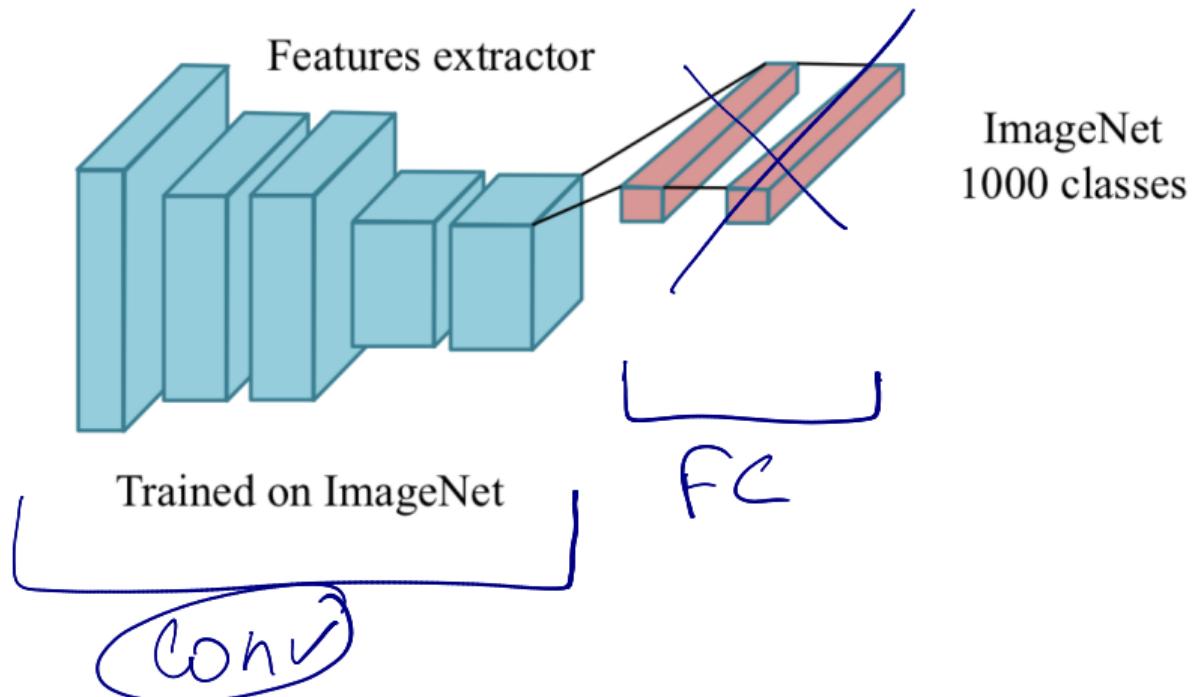


Layer 5

<https://arxiv.org/pdf/1311.2901.pdf>

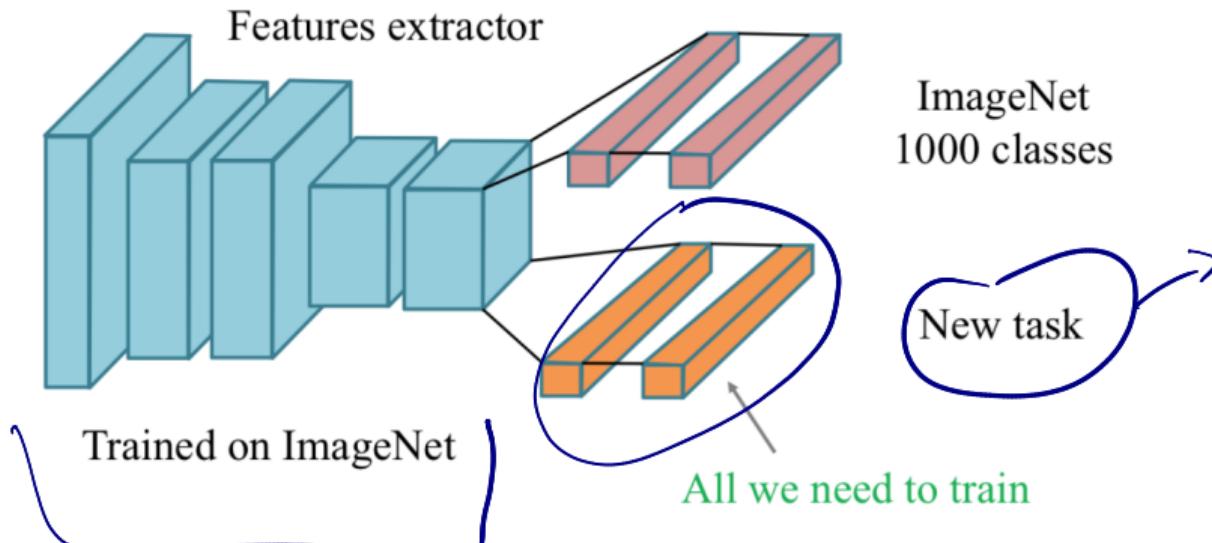
Transfer learning

- Глубокие сети извлекают из изображений сложные фичи, но для их обучения нужно много данных...



Transfer learning

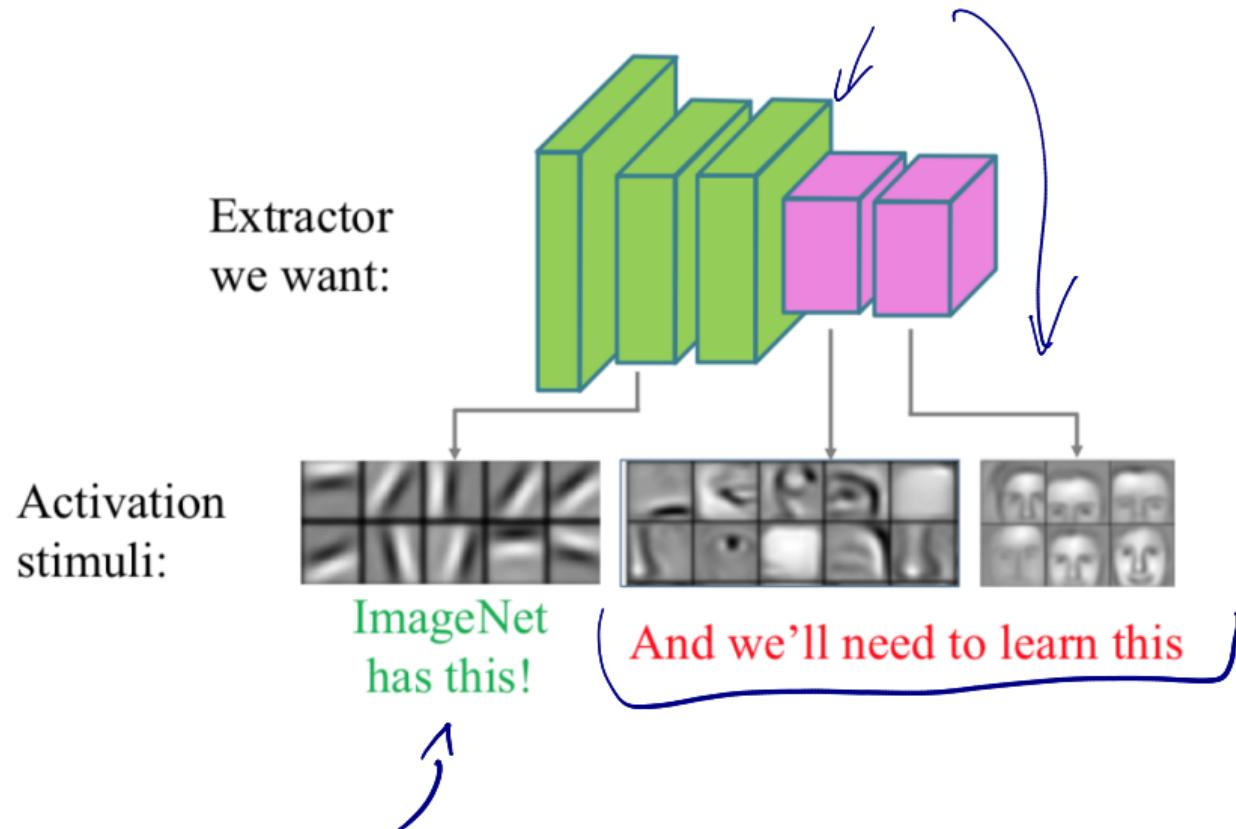
- Глубокие сети извлекают из изображений сложные фичи, но для их обучения нужно много данных...
- Давайте повторно использовать уже предобученную сеть!



Transfer learning

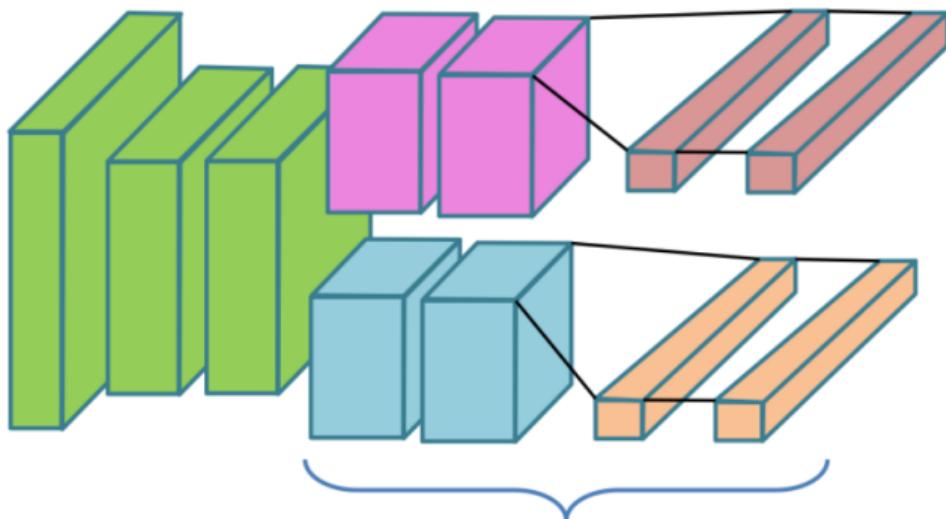
- Нужно меньше данных для обучения, так как нас интересуют лишь последние слои
- Это работает если наша задача похожа на ту, для которой обучалась используемая сетка
- Например, если мы хотим распознавать эмоции, в датасете для нашей сетки должны были быть человеческие лица

Transfer learning



Transfer learning

ImageNet features extractor



ImageNet
1000 classes

New task

All we need to train

Крокодил learning

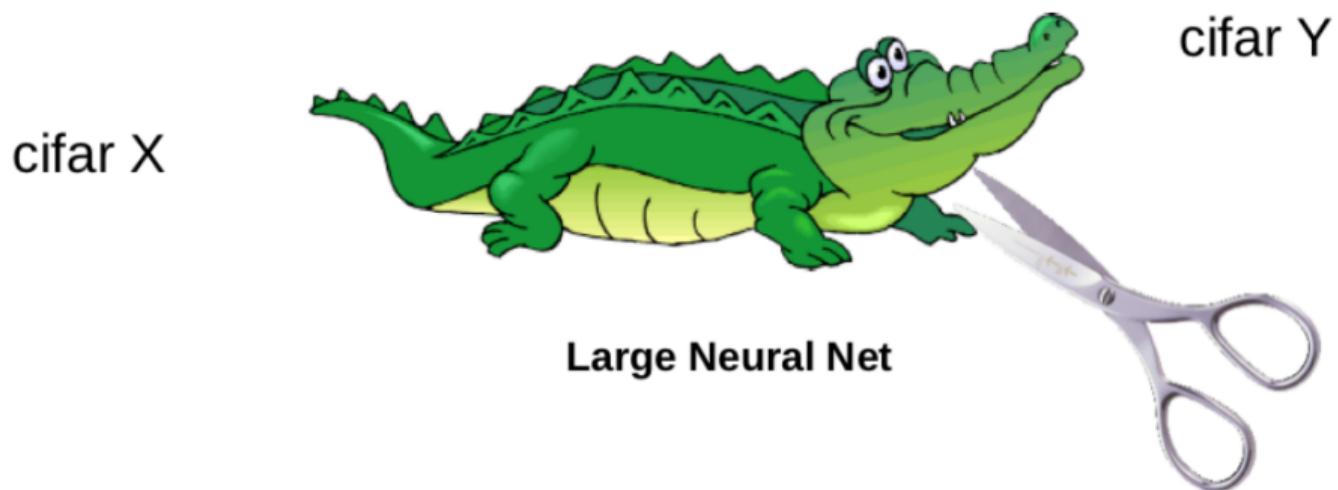
cifar X



cifar Y

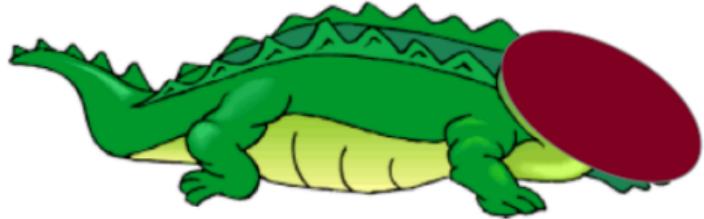
Large Neural Net

Крокодил learning



Крокодил learning

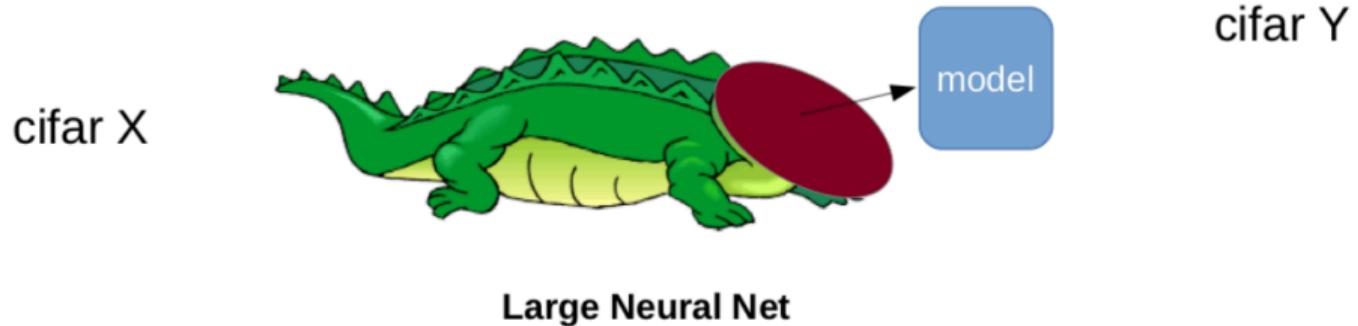
cifar X



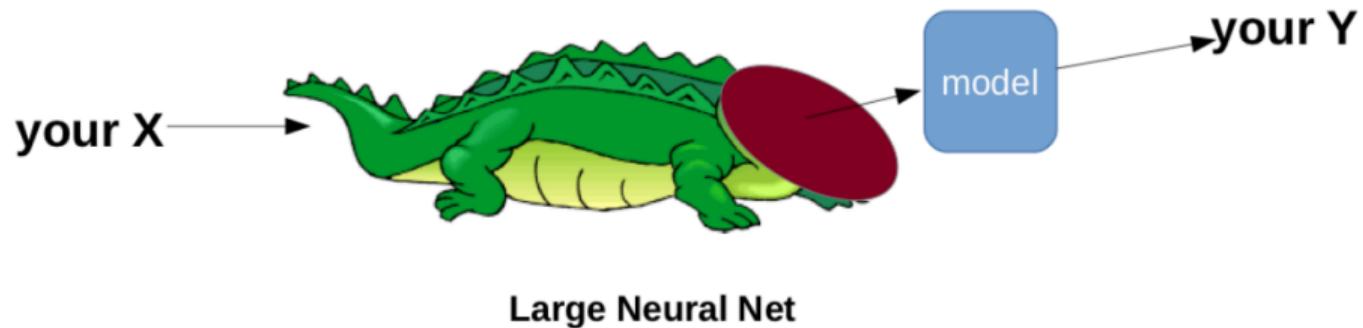
cifar Y

Large Neural Net

Крокодил learning



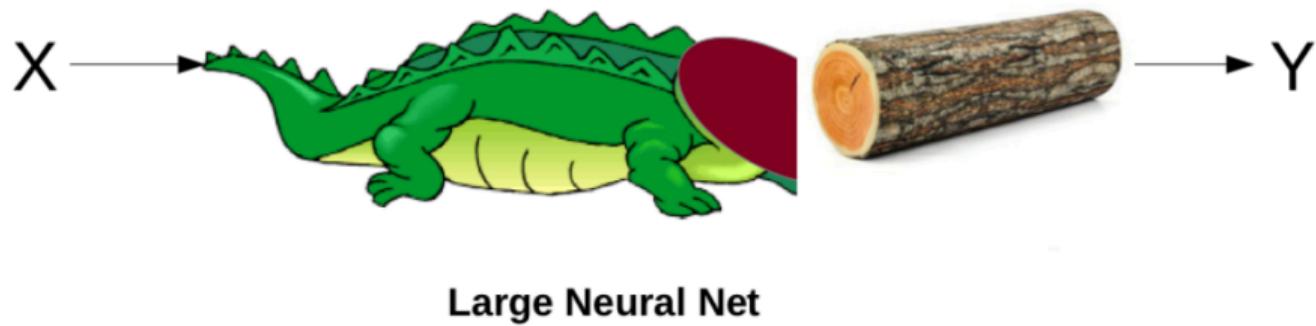
Крокодил learning



Крокодил learning

- Отрезали у крокодила голову
- Используем тело крокодила как экстрактор фичей
- Вместо головы крокодила можно прикрепить что угодно
- Даже случайный лес и бустинг
- В экстракторе фичей веса модели обычно не дообучают, дообучение касается только новой головы крокодила

Крокодил learning



Зоопарки моделей

- Перед тем как решать задачу с нуля, убедитесь, что готового решения
ещё нет
- Зоопарк предобученных PyTorch моделей
<https://pytorch.org/vision/stable/models.html>
- Другой большой зоопарк: <https://modelzoo.co/>

Собираем своего крокодила!