

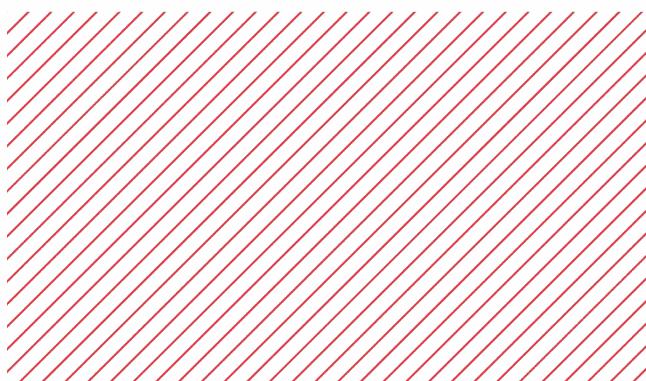
академия
больших
данных



ML пайплайн Ускорение моделей

Иван Карпухин

Ведущий программист-исследователь в
команде машинного зрения



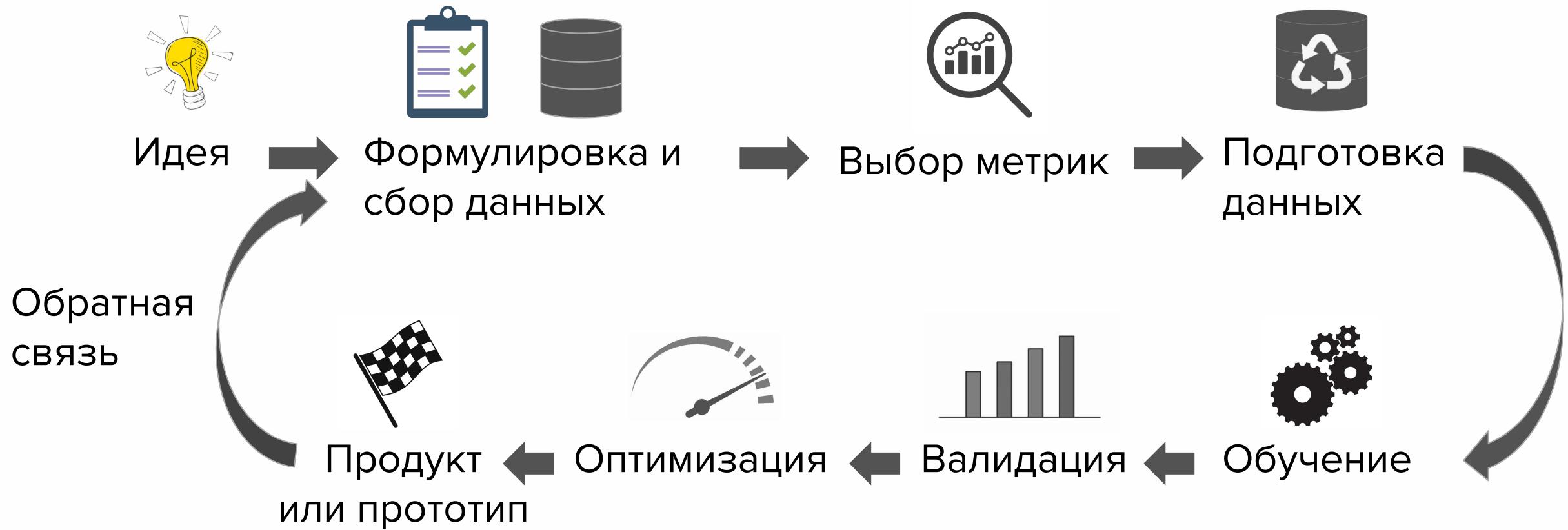
ML сложнее, чем кажется

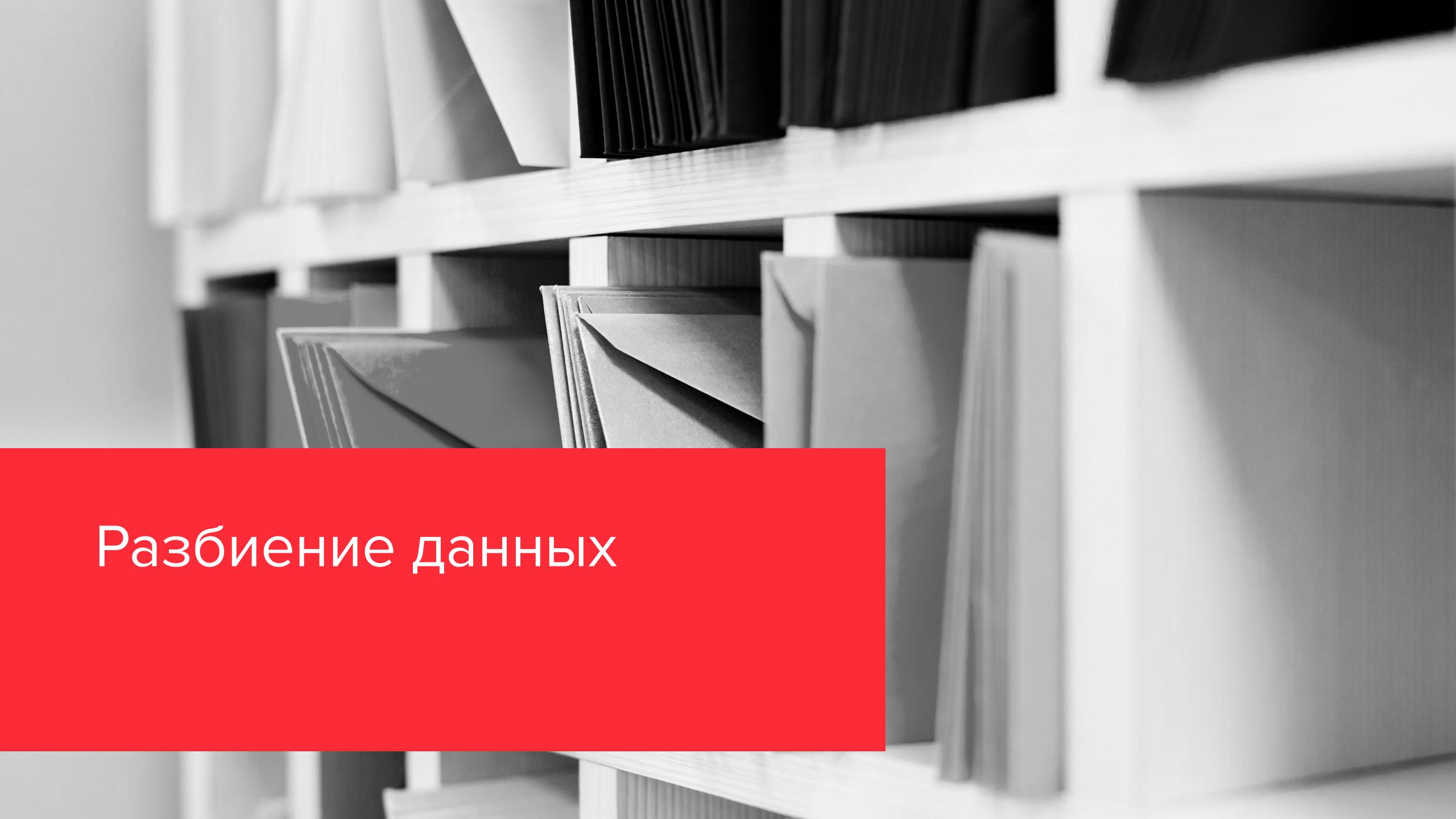
В теории:



ML сложнее, чем кажется

На практике:





Разбиение данных

Bias/Variance recap

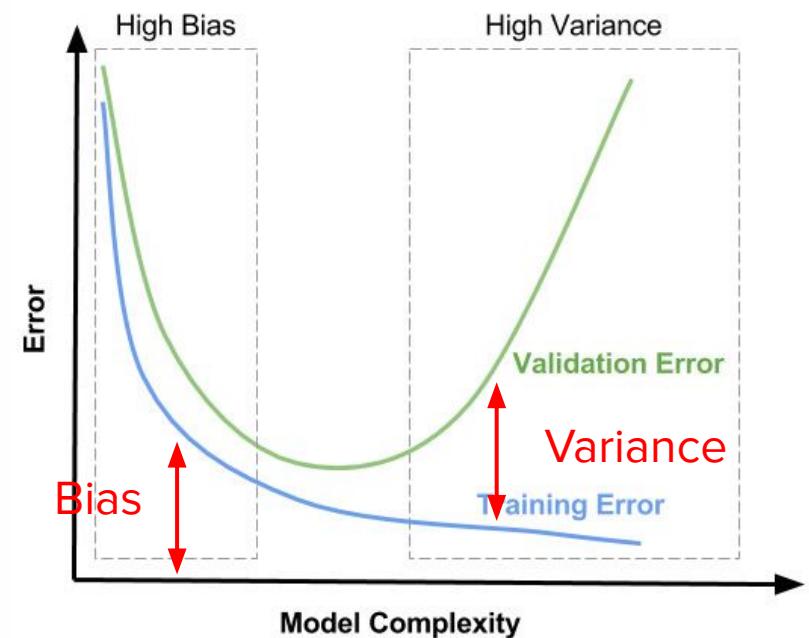
Train set и Validation set из одного распределения

Bias - величина ошибки на Train

Variance - разница ошибок Validation и Train

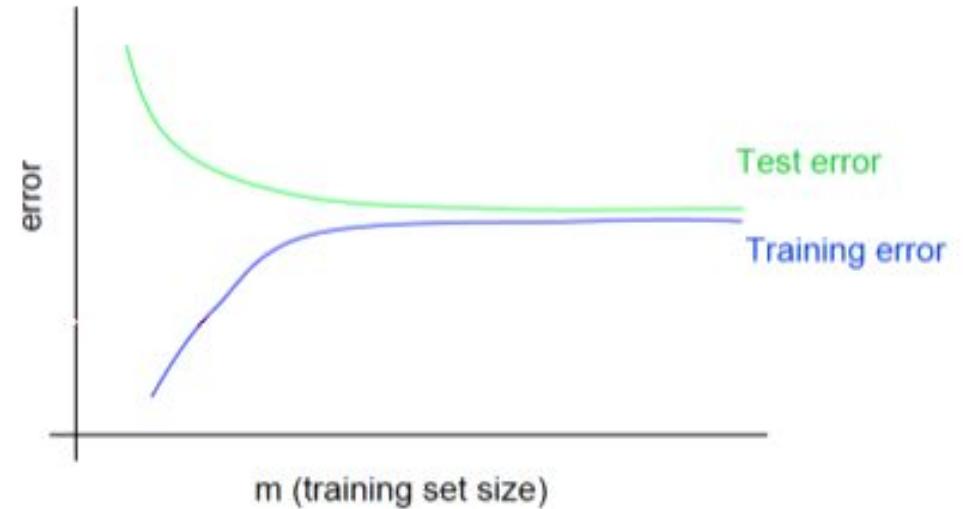
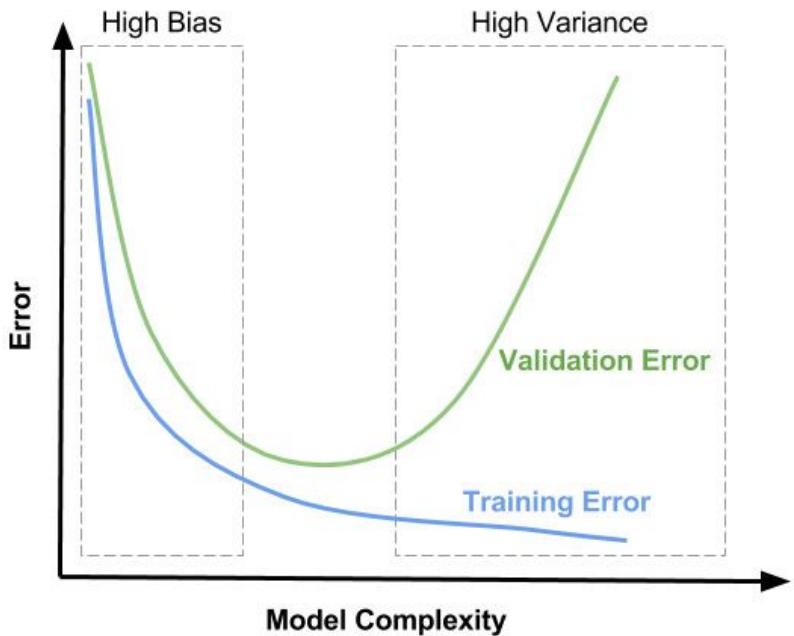
Описывают соответствие модели и данных

Терминология из анализа MSE*



* https://en.wikipedia.org/wiki/Bias-variance_tradeoff#Bias-variance_decomposition_of_mean_squared_error

Bias/Variance





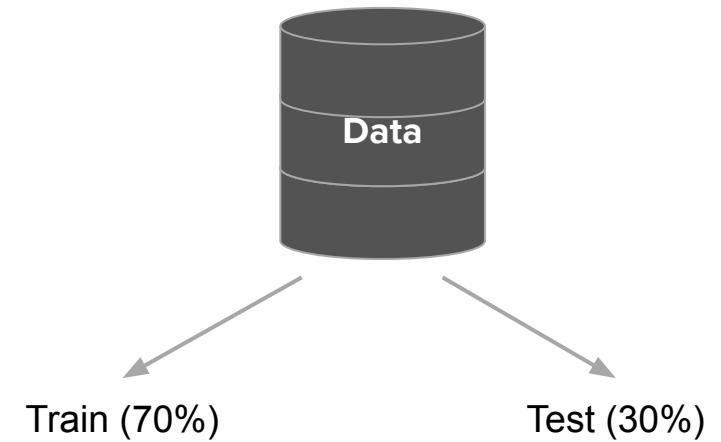
Данные

- Какие корпусы нужны?
- Какого размера?
- Из какого распределения?

Train/Test

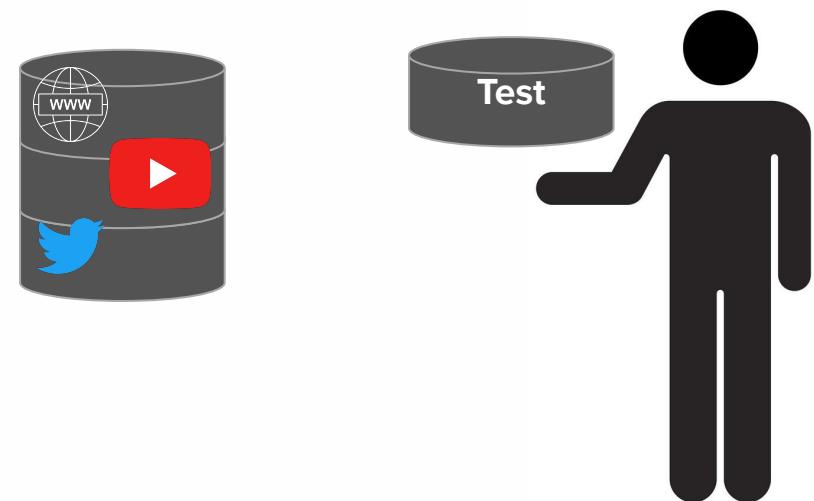
Причина 1

- Алгоритм переобучается под Train
- Нужен независимый Test для оценки



Причина 2

- Train большой, но из другого домена
- Test от заказчика

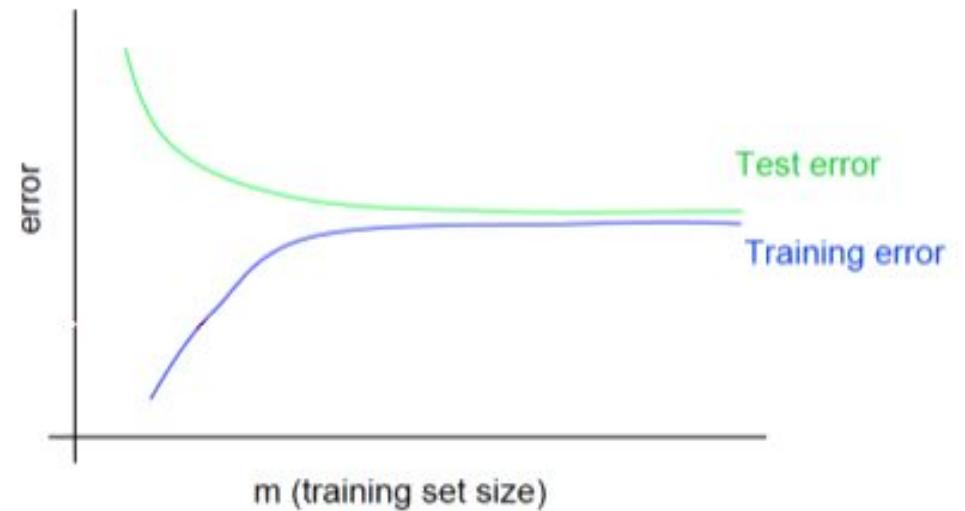
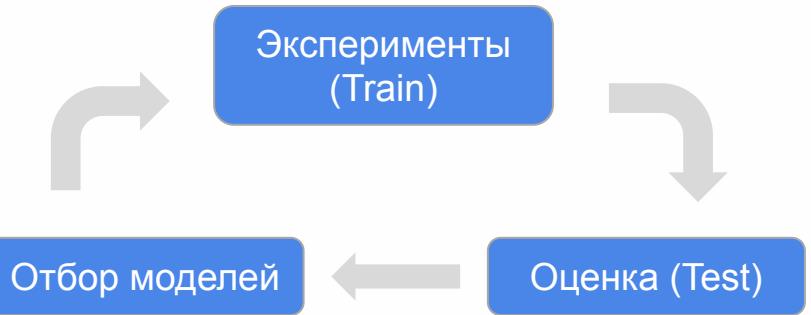


Train/Dev/Test

Отбираем модели по Test метрикам

=> переобучаемся под Test

Увеличить Test?



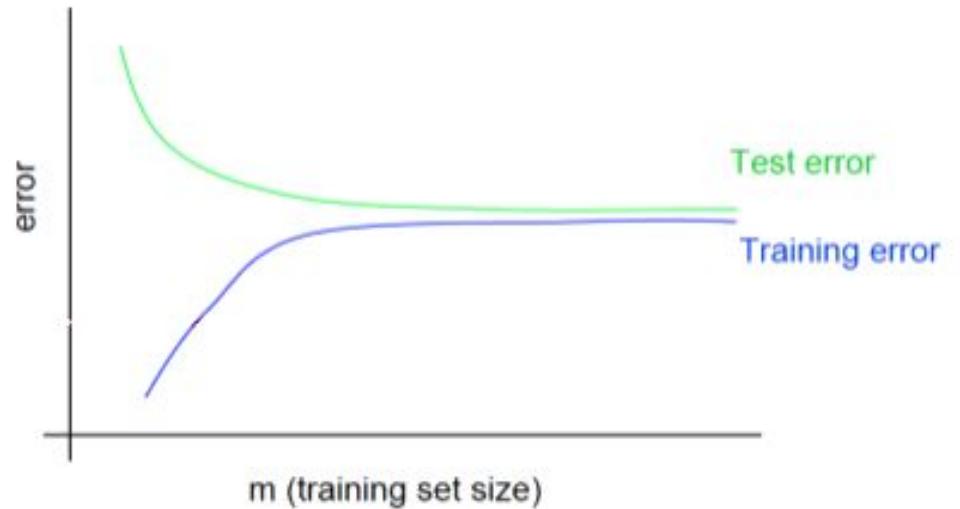
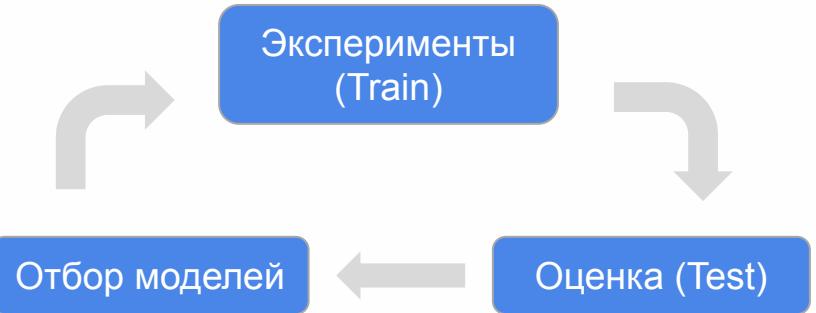
Train/Dev/Test

Отбираем модели по Test метрикам

=> переобучаемся под Test

Увеличить Test?

Сперва оценить степень переобучения



Train/Dev/Test

Отбираем модели по Test метрикам

=> переобучаемся под Test

Решение: Dev корпус



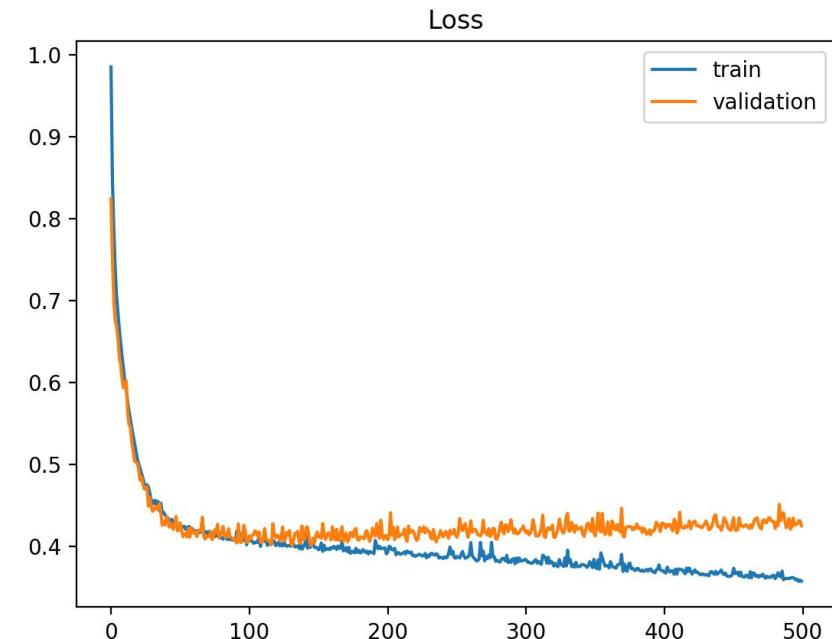
Корpus	Размер	Распределение	Назначение
Train	10.000-1.000.000	М.б. смещеннное	Обучение
Dev	1000 - 10.000	Несмещеннное	Отбор модели
Test	1000 - 10.000	Несмещеннное	Оценка модели

Проблема

$$Error_{dev} - Error_{train} = 0.1$$

На сколько переобучилась модель?

Как улучшить качество на Dev?



Проблема

$$Error_{dev} - Error_{train} = 0.1$$

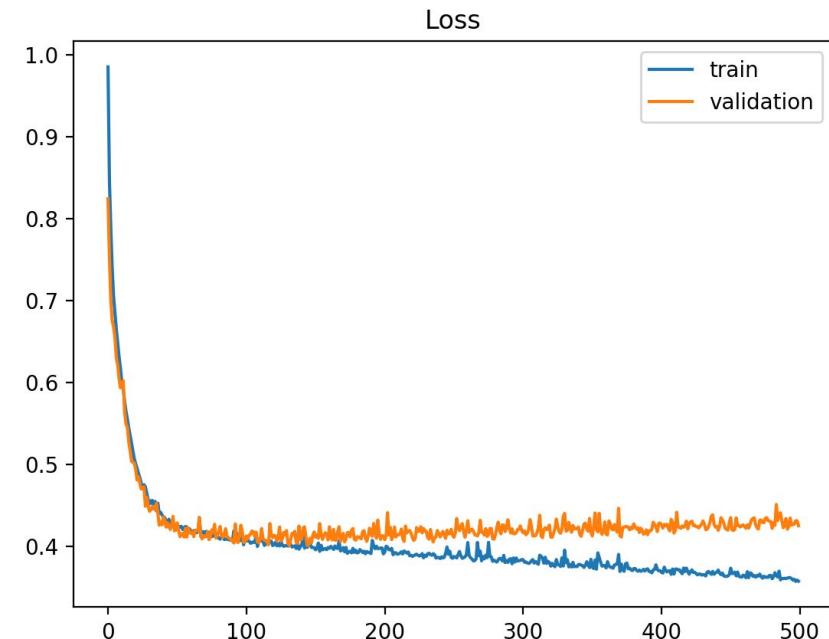
Train - смещённый

Dev - несмещённый

Уменьшать число параметров?

Увеличивать Train?

Искать несмещённые данные для Train?



Train-dev set



Размер Test set



Как оценить размер Test



Y^* - правильный ответ

$$Error = \begin{cases} 0, & Y = Y^* \\ 1, & Y \neq Y^* \end{cases}$$

Error - случайная величина Bernoulli(p)



Как оценить размер Test

N_0 - число правильных классификаций

N_1 - число ошибок

Размер Test: $N = N_1 + N_0$

Error - случайная величина Bernoulli(p)

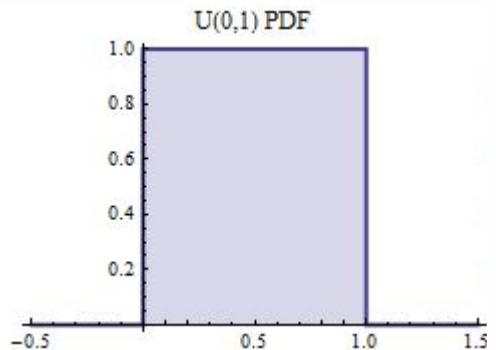
$P(p = x | N_0, N_1) ?$

Как оценить размер Test

$$P(p = x | N_0, N_1) = \frac{P(N_0, N_1 | p = x) P(p = x)}{\int_y P(N_0, N_1 | p = y) P(p = y) dy}$$

$P(p = x)$: Uniform(0, 1)

$$P(N_0, N_1 | p = x) = x^{N_1} (1 - x)^{N_0}, x \in [0, 1]$$

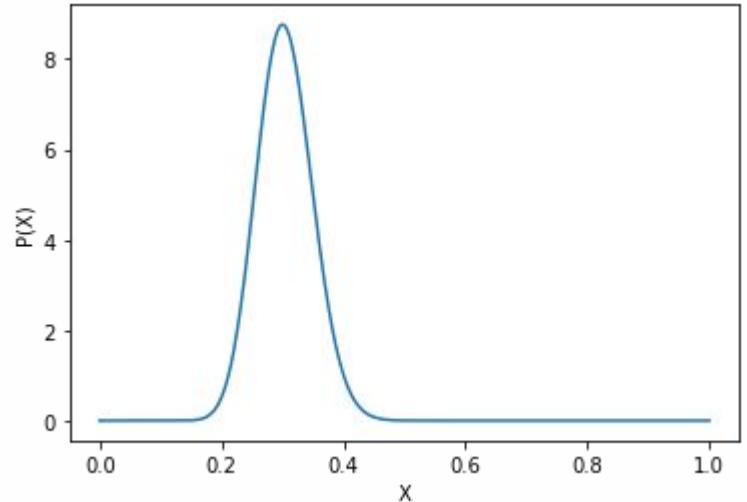


Как оценить размер Test

$$P(p = x | N_0, N_1) = \frac{P(N_0, N_1 | p = x) P(p = x)}{\int_y P(N_0, N_1 | p = y) P(p = y) dy}$$

$P(p = x)$: Uniform(0, 1)

$$P(N_0, N_1 | p = x) = x^{N_1} (1 - x)^{N_0}, x \in [0, 1]$$



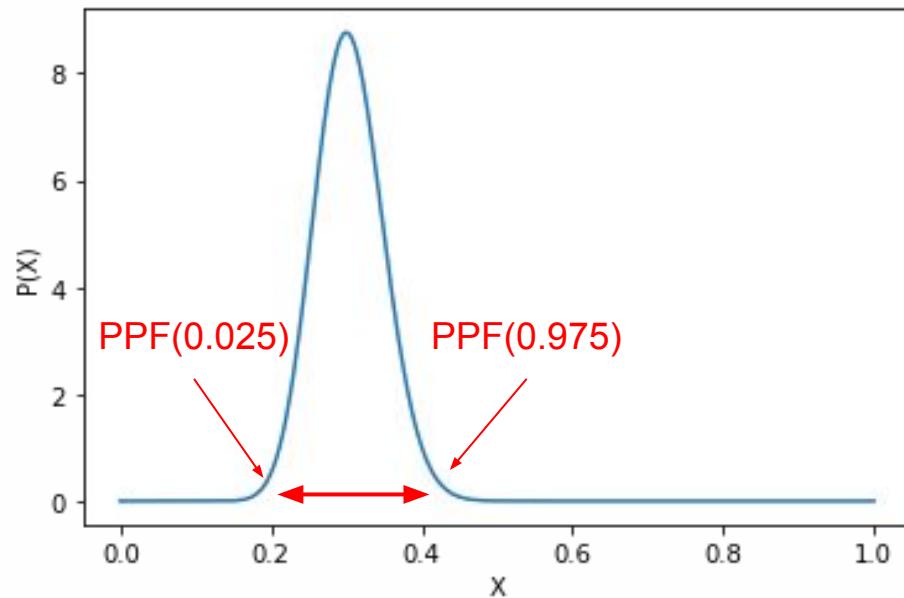
$$P(p = x | N_0, N_1) = Beta(N_1 + 1, N_0 + 1) = \frac{x^{N_1} (1 - x)^{N_0}}{B(N_1 + 1, N_0 + 1)}$$

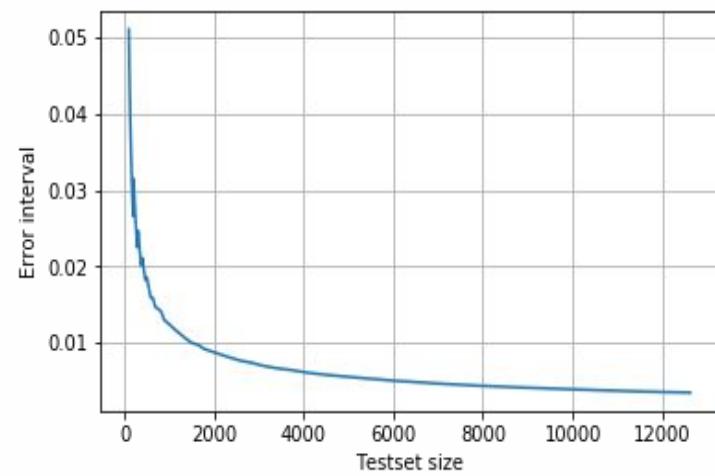
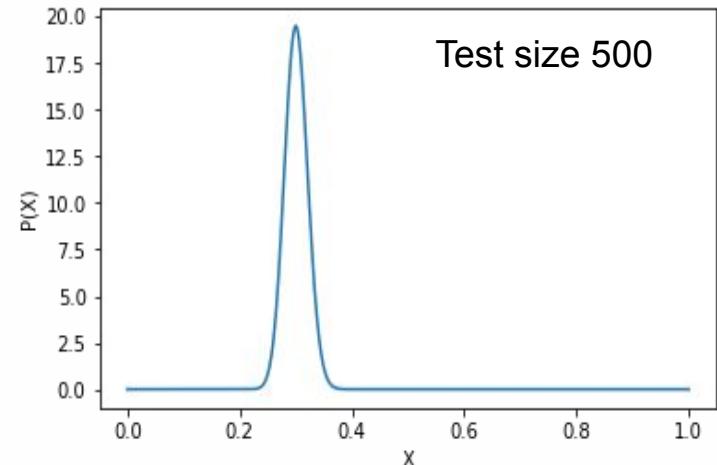
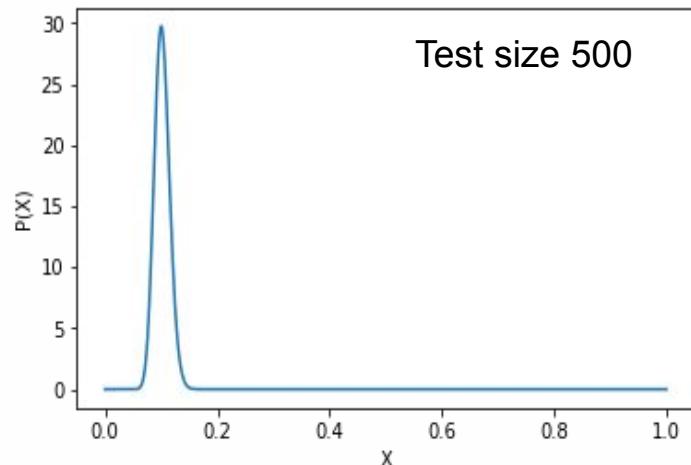
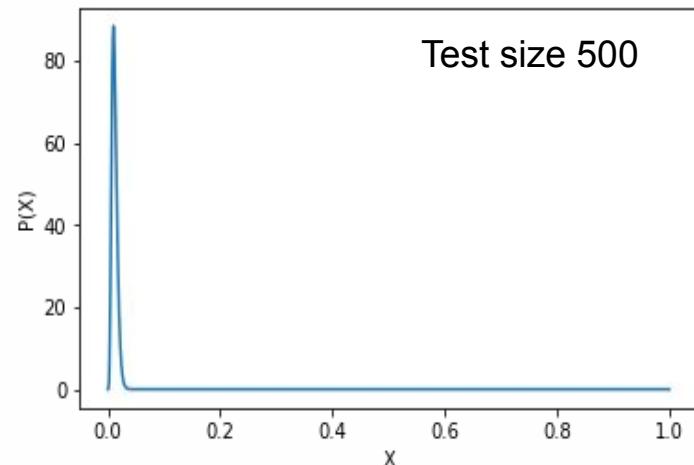
Как оценить размер Test

Доверительный интервал с уровнем доверия $\alpha = 0.95$?

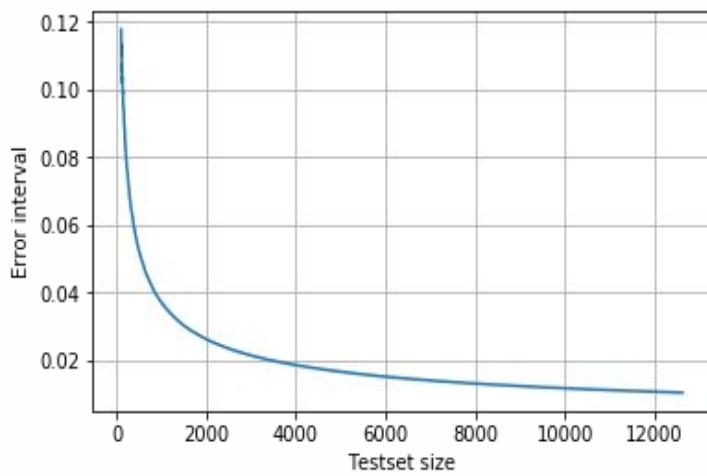
$$PPF_p(x) = \alpha : P(p \leq x) = \alpha$$

$$\Delta = PPF\left(\alpha + \frac{1 - \alpha}{2}\right) - PPF\left(\frac{1 - \alpha}{2}\right)$$

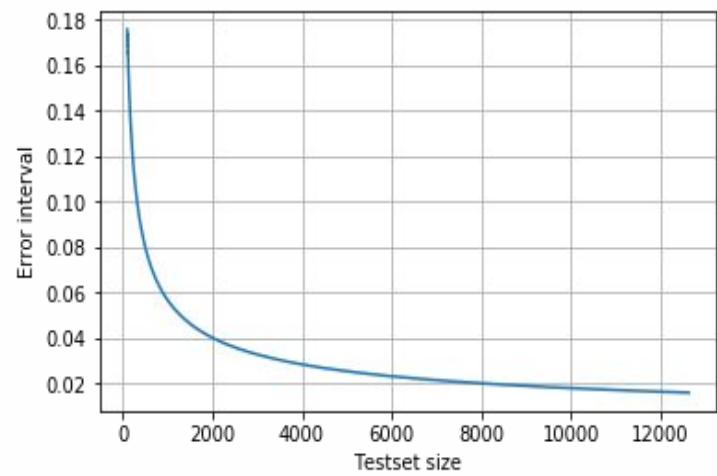




Mean Error = 0.01



Mean Error = 0.1



Mean Error = 0.3

Метрики

Виды метрик

Технические

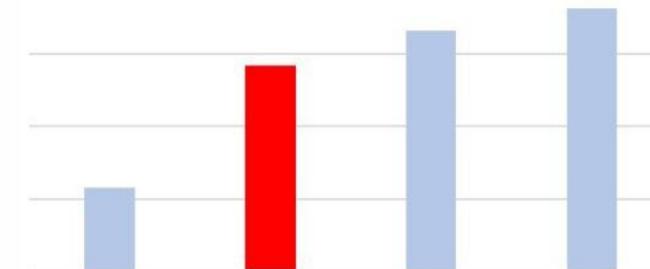
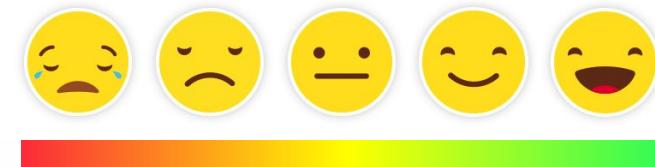
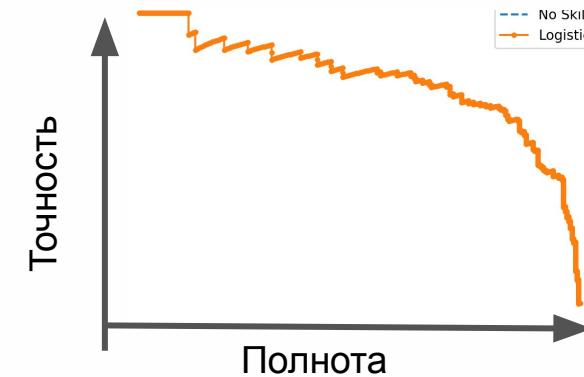
- оценивают подсистемы
- выявляют возможности для улучшений

Продуктовые

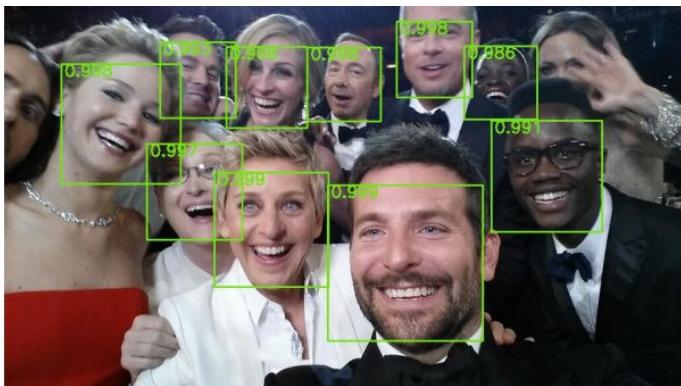
- связаны с бизнесом
- оценивают систему целиком
- одно число

Метрики из статей и стандартов

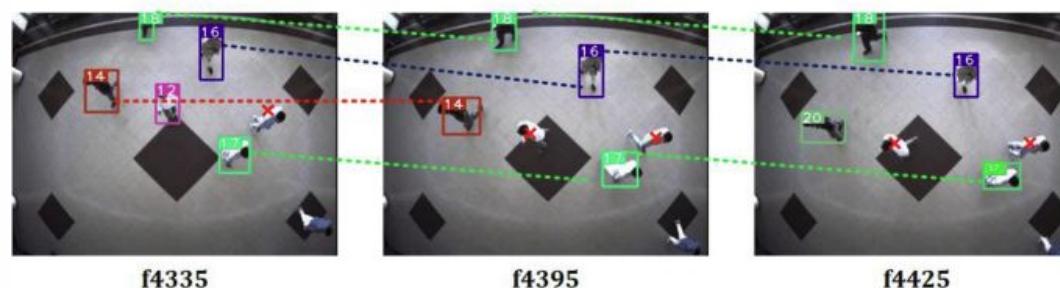
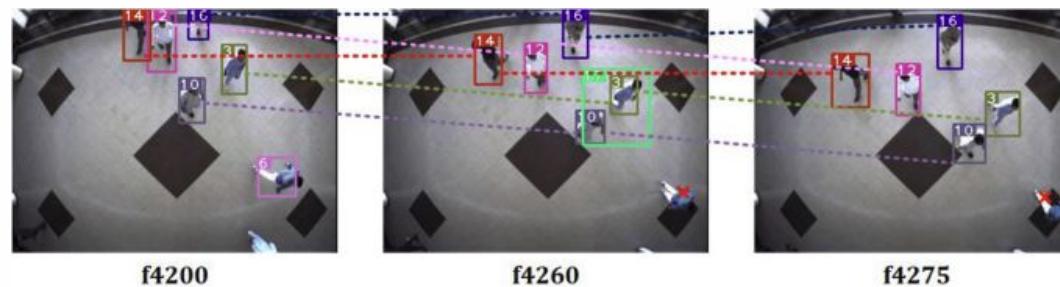
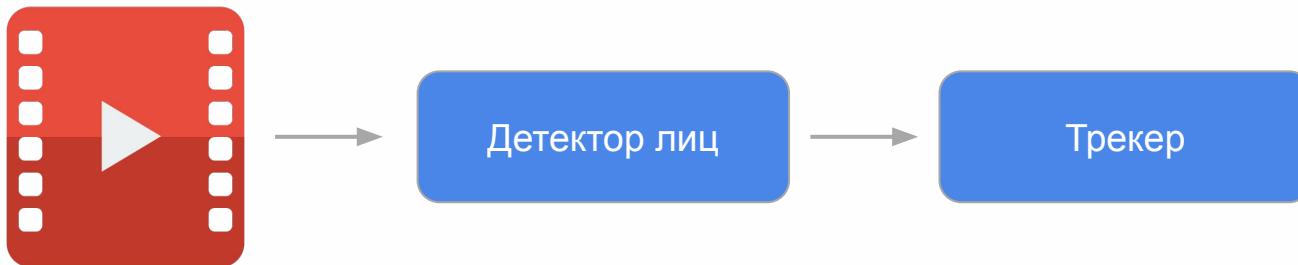
- сравнение с конкурентами



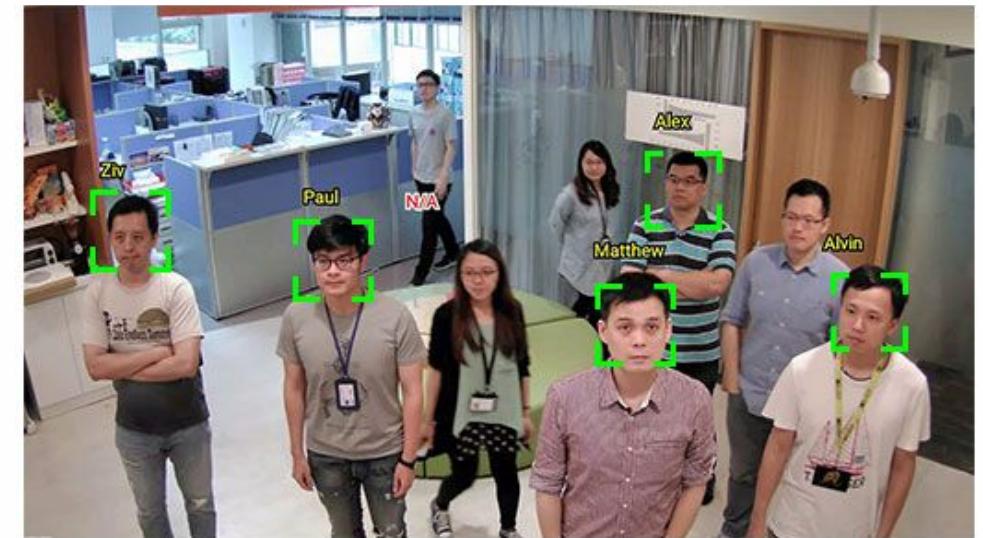
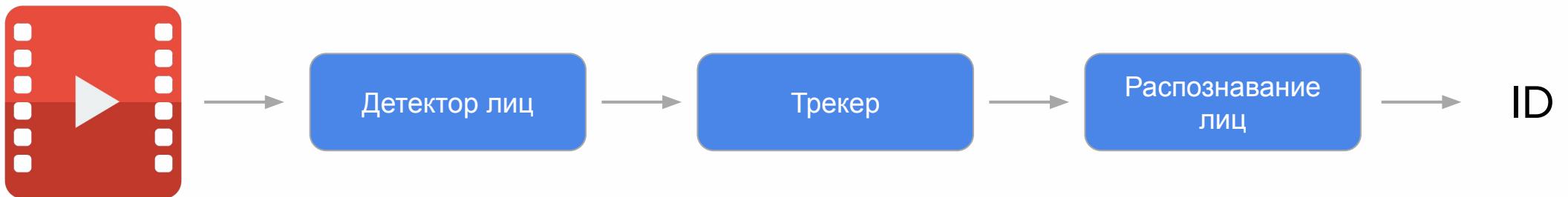
Детектирование и распознавание людей



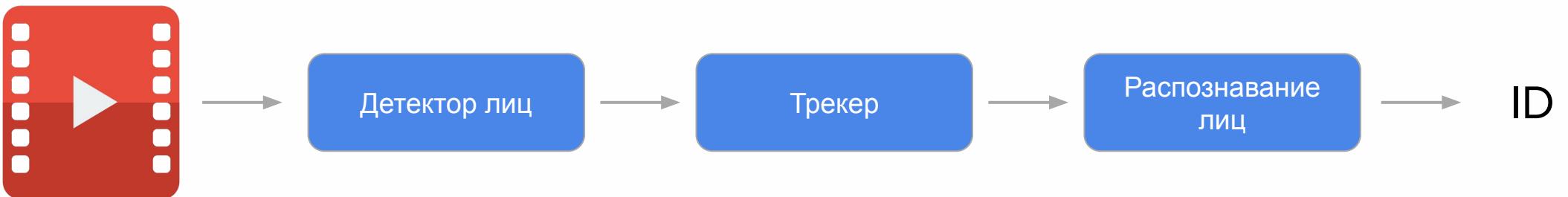
Детектирование и распознавание людей



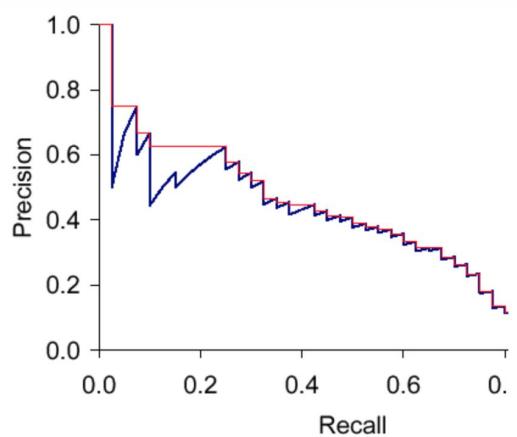
Детектирование и распознавание людей



Технические метрики



Average precision



Mostly tracked
Mostly lost
Identity switches
Fragmentation

Accuracy 1 / 1000



Продуктовые метрики

Задача:

- у заказчика есть база сотрудников с фото
- нужно найти посторонних людей на видео с камеры

Какое число выбрать в качестве продуктовой метрики?



Продуктовые метрики

Какие параметры важны:

- скорость работы пайплайна (ms / frame)
- частота ложных срабатываний (1 / hour)
- вероятность правильной классификации постороннего



Усреднение метрик

1. Можно связать частоту ложных срабатываний с вероятностью правильной классификации сотрудника
2. Вероятности правильной классификации сотрудника и постороннего можно усреднить (mean, harmonic mean)

Как быть с быстродействием?



Ограничение метрик

Какие параметры важны:

- скорость работы пайплайна (ms / frame)
- частота ложных срабатываний (1 / hour)
- вероятность правильной классификации постороннего

1. Ложные срабатывания допустимы не чаще 1 / час (в среднем)
2. Нужно обрабатывать кадр быстрее 200ms на CPU

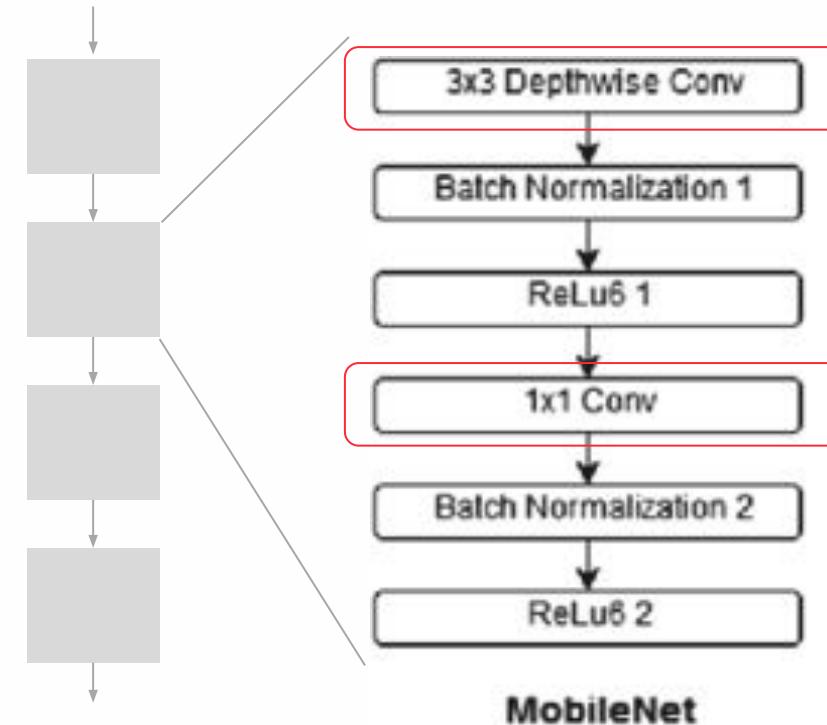
=> остается один свободный параметр - вероятность обнаружения постороннего

Ускорение моделей

Архитектура

Пример: MobileNet

Conv 3x3xn \rightarrow Depthwise 3x3 + Conv 1x1xn

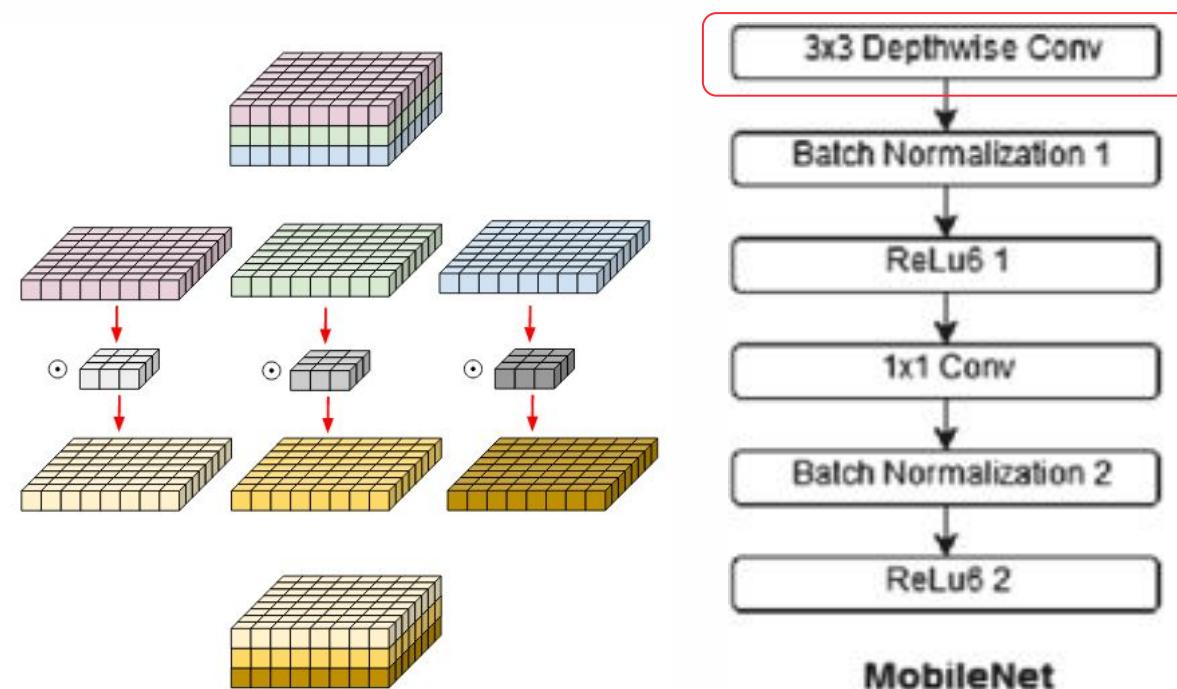


Архитектура

Пример: MobileNet

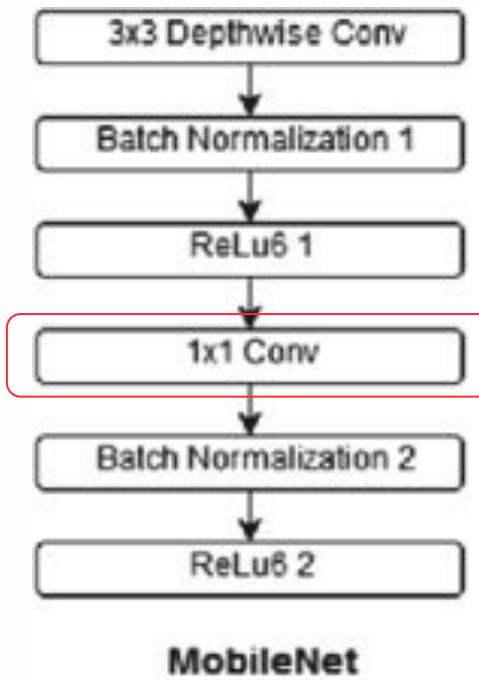
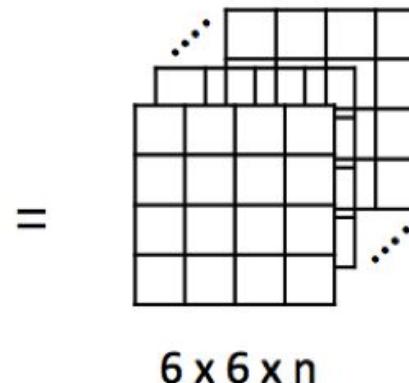
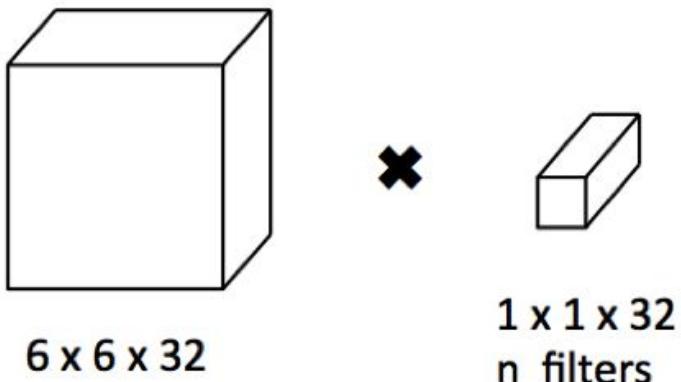
Ядро обычного Conv: $3 \times 3 \times n$

Depthwise Conv: 3×3



Архитектура

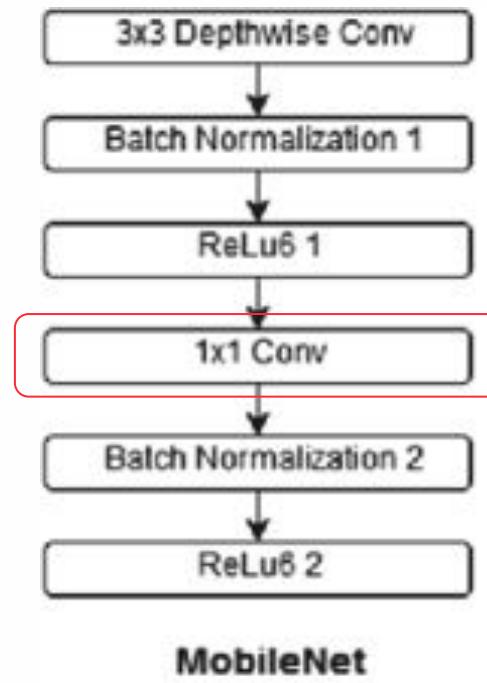
Свёртка $1 \times 1 \times n$



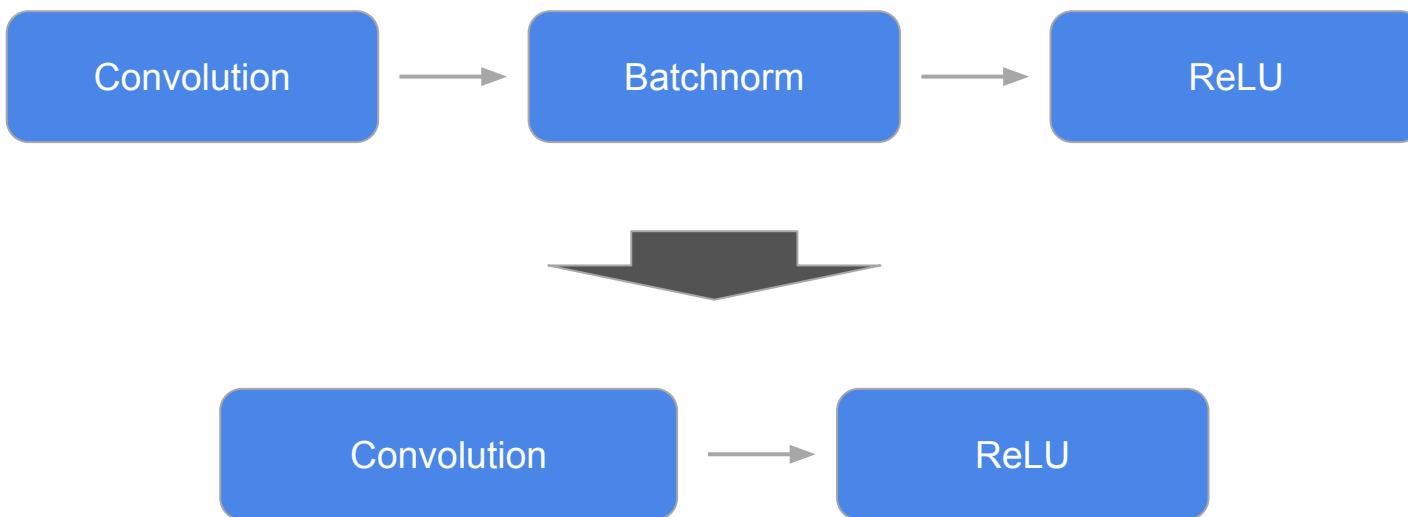
Архитектура

Пример: MobileNet

Слой	# умножений на один выход
Conv $3 \times 3 \times n$	$9n$
Depthwise conv 3×3	9
Conv $1 \times 1 \times n$	n
Depthwise + Conv $1 \times 1 \times n$	$9 + n$



Архитектура. Layer fusion





Оптимизация вычислений

- Оптимизация GPU ядер
- Использование специальных инструкций CPU (# AVX)

Стандартные реализации PyTorch уже хорошо оптимизированы
(cuDNN под капотом)

Инференс моделей в TensorRT работает быстрее для Float 16

Квантование сетей



Квантование

Приближённые быстрые вычисления с пониженной точностью:

- Float 16
- Int 8

Два подхода:

- Квантование обученной модели
- Quantization Aware Training (QAT)

Умножение вещественных чисел

$$N = Mn^p$$

M - мантисса

р - порядок

$$N_1 N_2 = M_1 M_2 n^{p_1 + p_2}$$

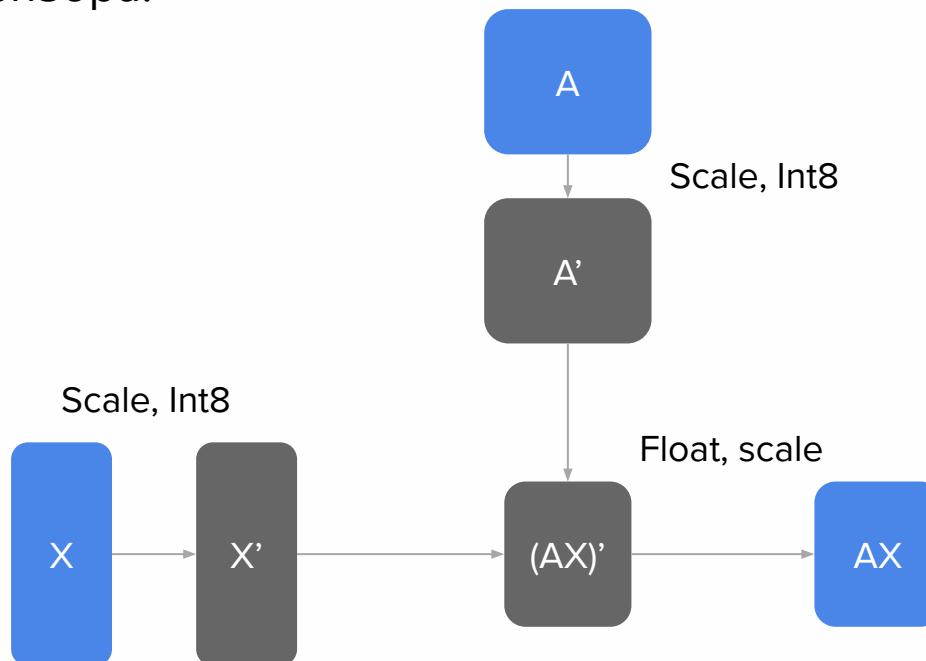
Квантование Int8

Идея:

- один порядок для всех элементов тензора:

$$N_i = M_i n^p$$

- свой порядок у каждого тензора



Умножение в Int8

Int8 принимает 256 значений: [-128, 127]

$$N = M \cdot n^p$$

Умножим вещественные числа:

M - мантисса
p - порядок

$$15.4 \cdot 0.7 = 10.78$$

Можно вынести масштаб, чтобы мантисса стала целочисленной:

$$154 \cdot 10^{-1} \cdot 7 \cdot 10^{-1} = 1078 \cdot 10^{-2} = 10.78 \quad \# \text{Переполнение, т.к. } 1078 > 127$$

Если мантиссы не хватает, округляем:

$$15.4 \approx 15$$

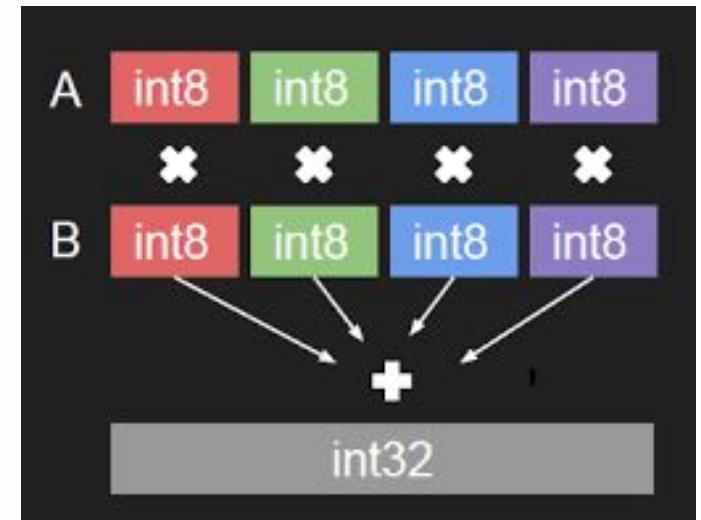
$$15 \cdot 7 \cdot 10^{-1} = 105 \cdot 10^{-1} = 10.5$$

105 не приводит к переполнению

Скалярное произведение в Int8

$$(A, B) = a_1 * b_1 + a_2 * b_2 + \dots + a_N * b_N$$

- Результат умножения и сложение в int32
- В конце int32 квантуется в int8



Пример

a = 121.5 b = 65.8 c = -12 d = 118.4

r = a * b + c * d

Входной масштаб: $254 = 127 \times 2$

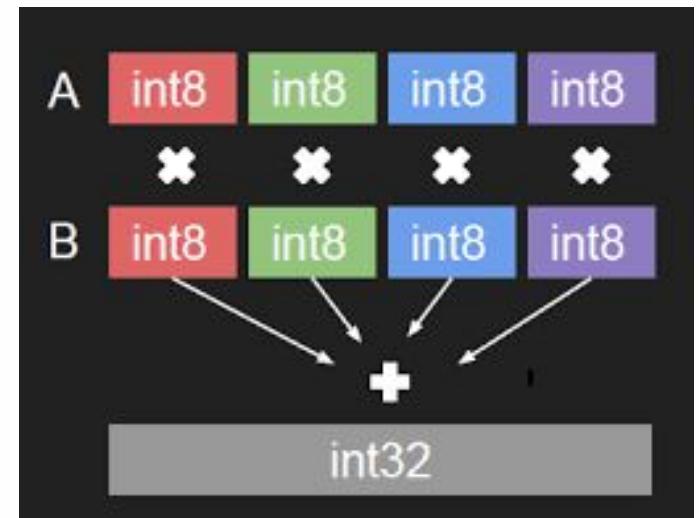
Выходной масштаб: 2000

1. $a' = 61 \quad b' = 33 \quad c' = -6 \quad d' = 59$ (int8)

2. $a' * b' = 2013 \quad c' * d' = -354$ (int32)

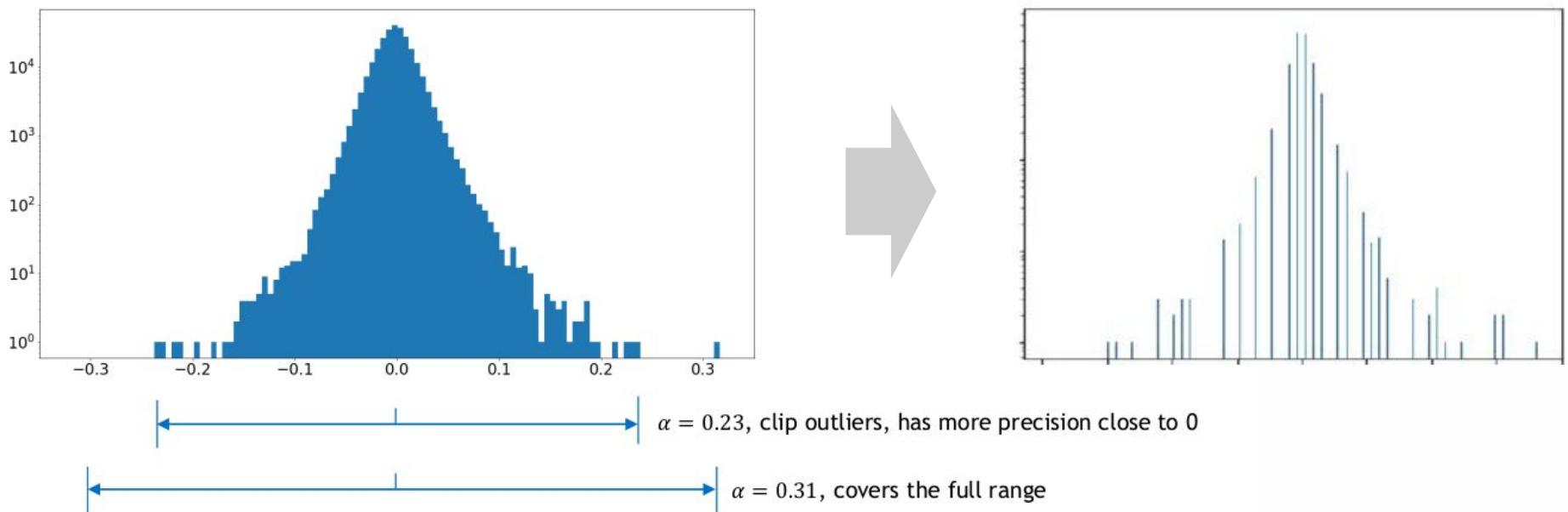
3. $a' * b' + c' * d' = 1659$ (int32)

4. $r = \text{round}(\text{clip}(1659 * 127 / 2000), -128, 127) = 105$ (int8)



Как выбрать масштаб для тензора

1. Прогнать данные через модель
2. Построить распределение выходов тензора
3. Выбрать параметры квантования, сохраняющие максимум информации о распределении

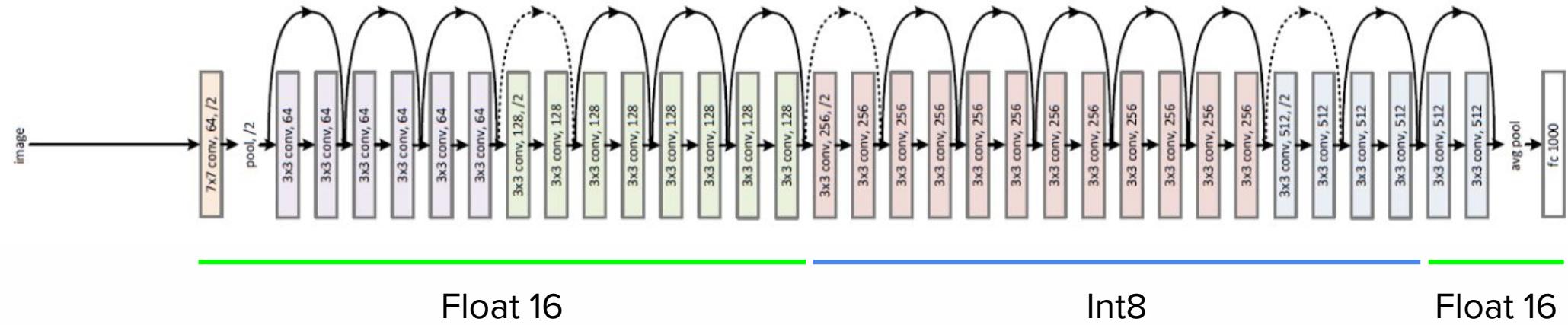




Вопросы

Mixed precision

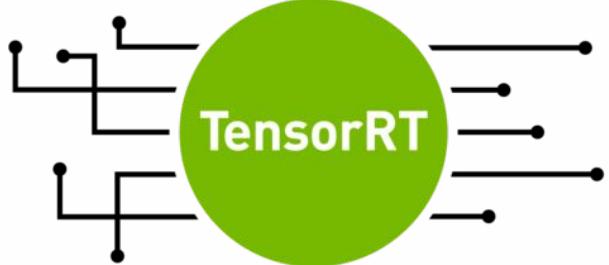
Чтобы сохранить точность, можно квантовать только часть сети



Фреймворки

Фреймворки для квантования обученных моделей:

- TensorRT для Nvidia GPU
- OpenVINO для Intel CPU





Примеры результатов*

Ускорение:

	Batch size 1			Batch size 8			Batch size 128		
	FP32	FP16	Int8	FP32	FP16	Int8	FP32	FP16	Int8
MobileNet v1	1	1.91	2.49	1	3.03	5.50	1	3.03	6.21
MobileNet v2	1	1.50	1.90	1	2.34	3.98	1	2.33	4.58
ResNet50 (v1.5)	1	2.07	3.52	1	4.09	7.25	1	4.27	7.95
VGG-16	1	2.63	2.71	1	4.14	6.44	1	3.88	8.00
VGG-19	1	2.88	3.09	1	4.25	6.95	1	4.01	8.30
Inception v3	1	2.38	3.95	1	3.76	6.36	1	3.91	6.65
Inception v4	1	2.99	4.42	1	4.44	7.05	1	4.59	7.20
ResNext101	1	2.49	3.55	1	3.58	6.26	1	3.85	7.39

* По данным Nvidia

Примеры результатов*

Качество:

Model	FP32	Int8 (max)	Int8 (entropy)	Rel Err (entropy)
MobileNet v1	71.01	69.43	69.46	2.18%
MobileNet v2	74.08	73.96	73.85	0.31%
NASNet (large)	82.72	82.09	82.66	0.07%
NASNet (mobile)	73.97	12.95	73.4	0.77%
ResNet50 (v1.5)	76.51	76.11	76.28	0.30%
ResNet50 (v2)	76.37	75.73	76.22	0.20%
ResNet152 (v1.5)	78.22	5.29	77.95	0.35%
ResNet152 (v2)	78.45	78.05	78.15	0.38%
VGG-16	70.89	70.75	70.82	0.10%
VGG-19	71.01	70.91	70.85	0.23%
Inception v3	77.99	77.7	77.85	0.18%
Inception v4	80.19	1.68	80.16	0.04%

* По данным Nvidia

Quantization aware training



Тренировка моделей для квантования

Иногда квантование модели не даёт желаемой точности

Идея: научить модель корректировать ошибки, полученные в результате квантования

Шаги:

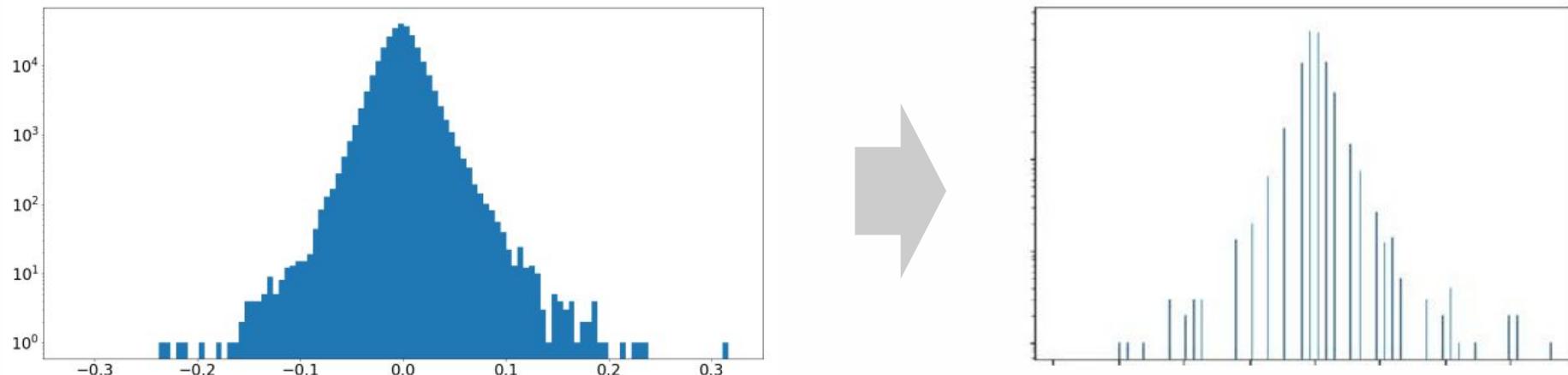
- Обучить модель во Float32
- Дообучить модель, подменяя выходы слоёв на квантованные значения
(изменяется только forward pass, backward во Float32)
- Квантовать полученную модель для inference

Тренировка моделей для квантования

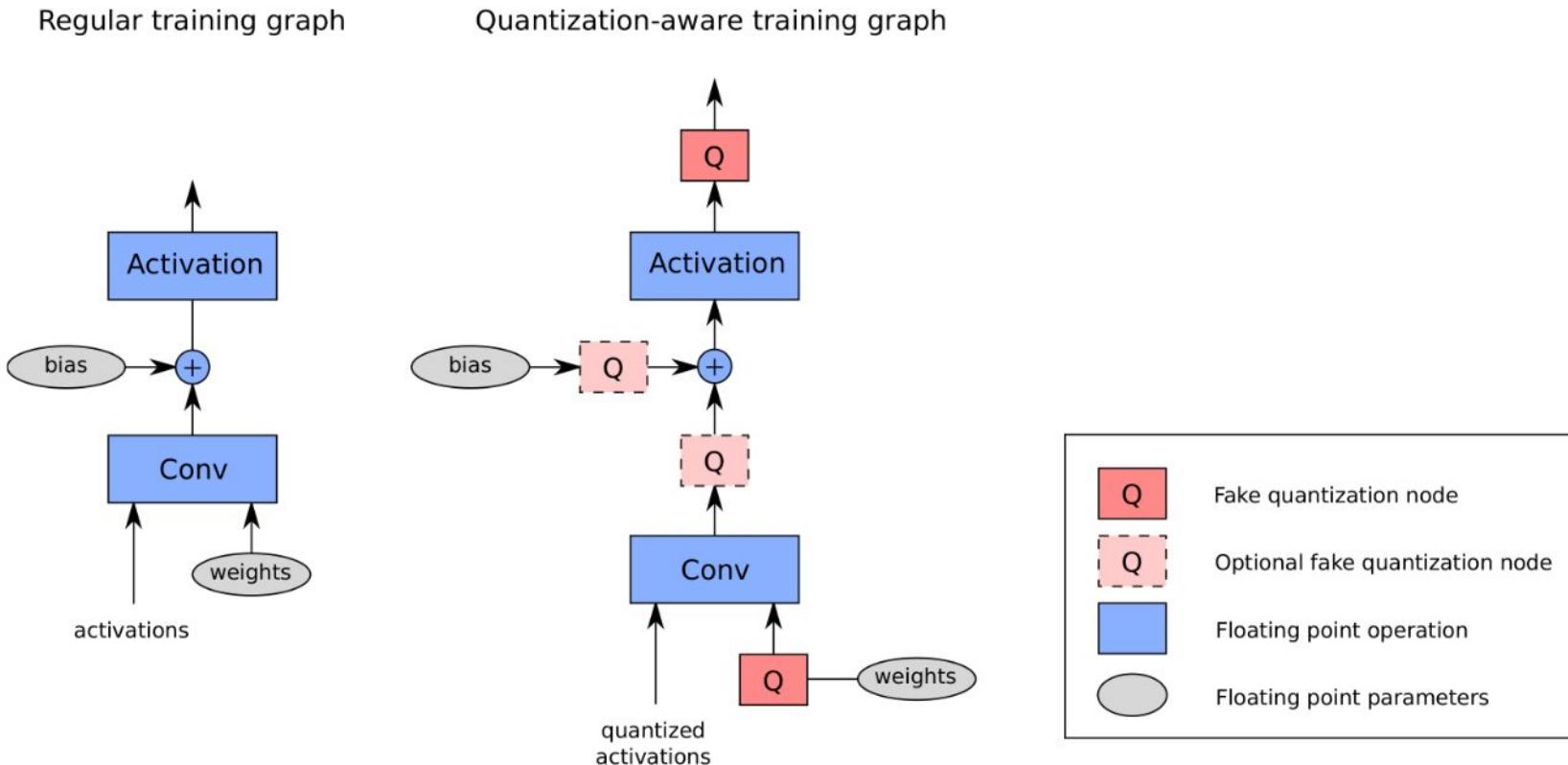
Fake quantization

Forward: $Q_f(X) = \text{Dequantize}(\text{Quantize}(X))$

Backward: $Q_f(X) = X$



Тренировка моделей для квантования



Во время дообучения собираются статистики распределения значений тензоров



PyTorch QAT*

PyTorch поддерживает QAT слои:

- Linear
- Conv2d

Можно добавлять собственные используя `torch.quantization.FakeQuantize`

* <https://pytorch.org/docs/stable/quantization.html#quantization-aware-training>

Резюме



Данные и метрики

Данные

- Train/Dev/Test
- Иногда нужен Train-Dev
- Посмотрели, как оценить размер testset

Метрики

- Несколько метрик можно усреднить
- Можно зафиксировать требования к части параметров

См. Machine Learning Yearning by Andrew Ng: <https://wwwdeeplearning.ai/programs/>

Ускорение

- Полезно использовать специализированные фреймворки для inference и квантования (TensorRT, OpenVINO)
- Во Float 16 обычно квантуется без существенных потерь качества
- Для Int8 обычно нужно специально дообучать сеть в режиме QAT

Ссылки:

<https://towardsdatascience.com/speeding-up-convolutional-neural-networks-240beac5e30f>

<https://iiw.kuleuven.be/onderzoek/eavise/starttodeeplearn/aspects-and-best-practices-of-quantization-aware-t.pdf>

<https://heartbeat.fritz.ai/practical-tips-for-better-quantization-results-613a3538c1a8>

Спасибо!