

Петрушов Андрей Александрович

# КВАНТИФИКАЦИЯ МОДЕЛЕЙ НЕЙРОННЫХ СЕТЕЙ

Московский государственный университет им.  
М.В.Ломоносова

Физический факультет, 2020 г.



# Терминология

- ❖ Квантификация – переход на менее объемные типы данных



# Терминология

- ❖ Бинаризация – переход на 1-битовые данные, частный случай квантификации
- ❖ В контексте нейронных сетей квантифицировать можно входные данные, веса и выходы

# Что это дает?

- ❖ Уменьшение размера модели
- ❖ Снижение энергопотребления
- ❖ Увеличение скорости работы сети
- ❖ Аппаратное целочисленное ускорение

# Сфера применения

- ❖ Компьютерное зрение – автопилот Tesla. 8 камер. Важно быстрое принятие решений!



# Сфера применения

- ❖ Автономные роботы – энергия ограничена



# Методы квантификации

Квантификация



# Методы квантификации

## Квантификация



```
graph TD; A[Квантификация] --> B[Post-training quantization]; A --> C[Quantization-aware training]
```

### Post-training quantization

Необходима полностью обученная модель на максимальной точности



# Методы квантификации

## Квантификация



```
graph TD; A[Квантификация] --> B[Post-training quantization]; A --> C[During-training binarization];
```

### Post-training quantization

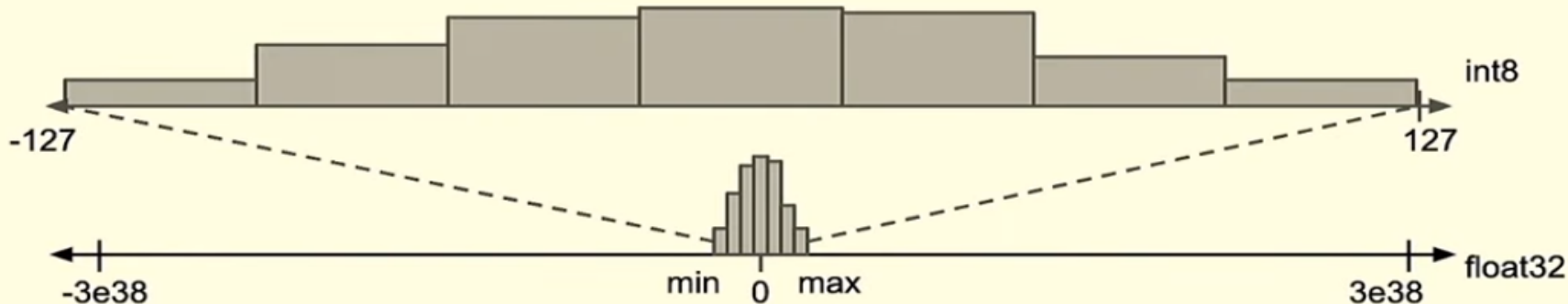
Необходима полностью обученная модель на максимальной точности

### During-training binarization

Методы квантификации используются непосредственно в процессе обучения. Выбор целевого типа данных производится до начала обучения

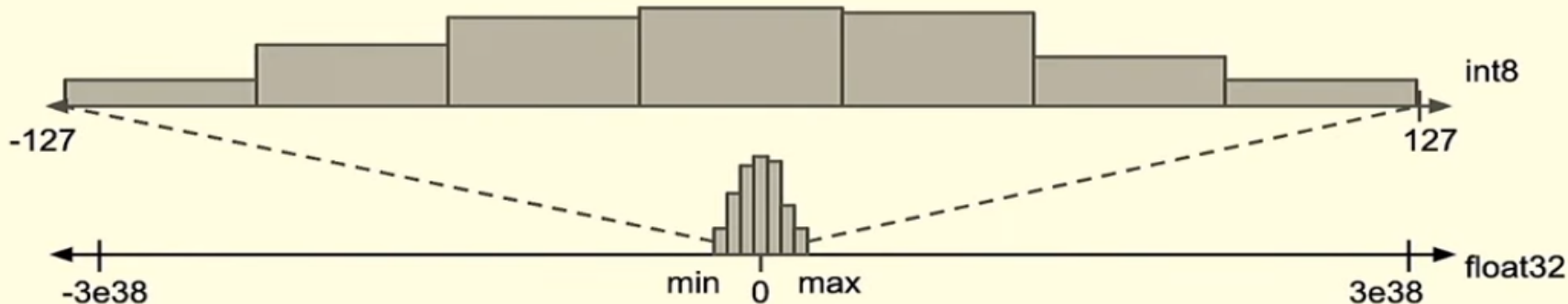
# Методы квантификации

## Общий принцип отображения



# Методы квантификации

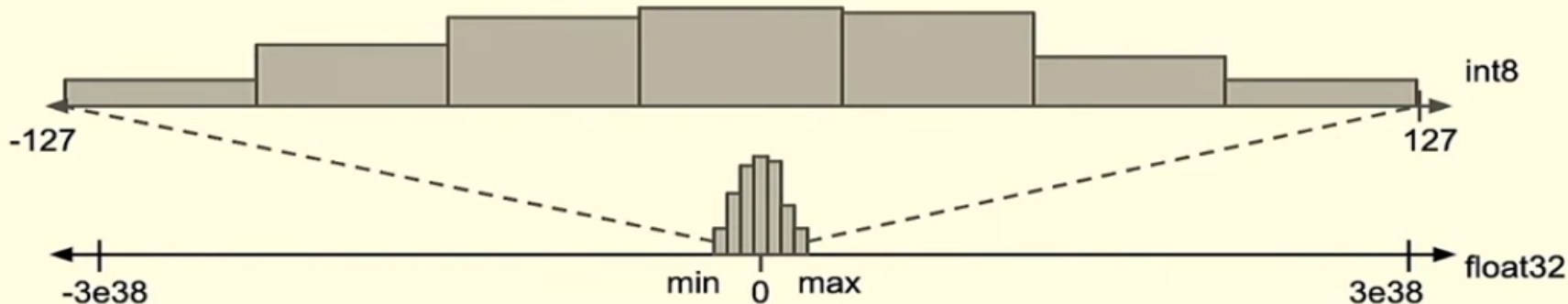
## Общий принцип отображения



$$\begin{pmatrix} -1270 \\ -10 \\ 0.1 \\ 1270 \end{pmatrix} \rightarrow 10 * \begin{pmatrix} -127 \\ -1 \\ 0 \\ 127 \end{pmatrix}$$

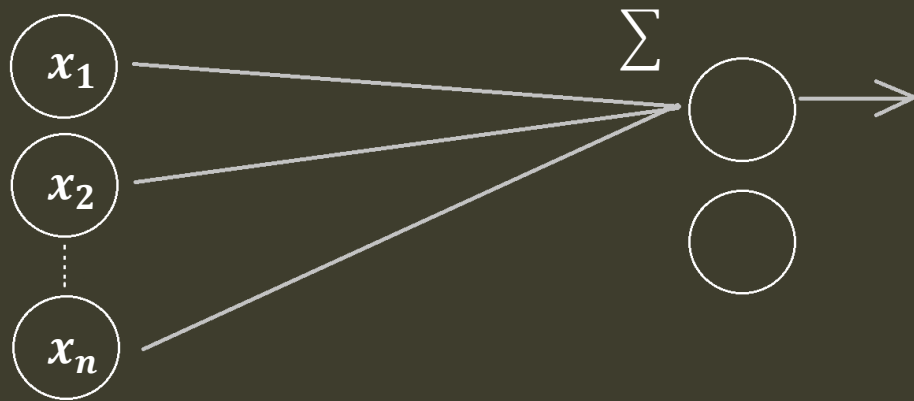
# Методы квантификации

## Общий принцип отображения

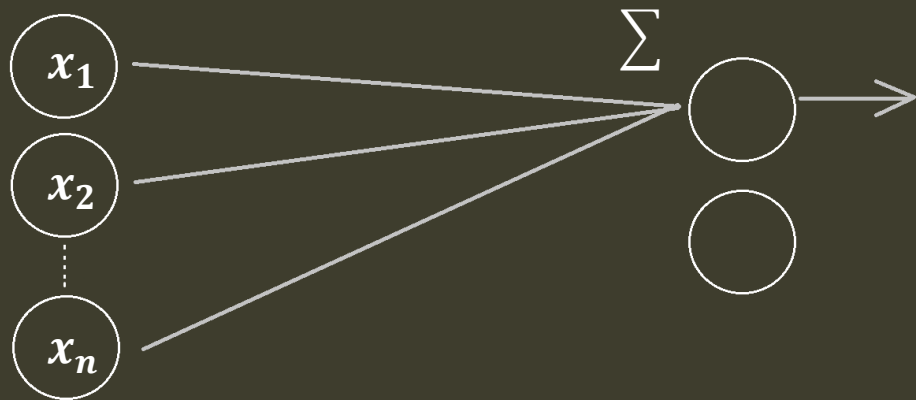


$$\begin{pmatrix} -1270 \\ -10 \\ 0.1 \\ 1270 \end{pmatrix} \rightarrow 10 * \begin{pmatrix} -127 \\ -1 \\ 0 \\ 127 \end{pmatrix}$$

# Post-training quantization

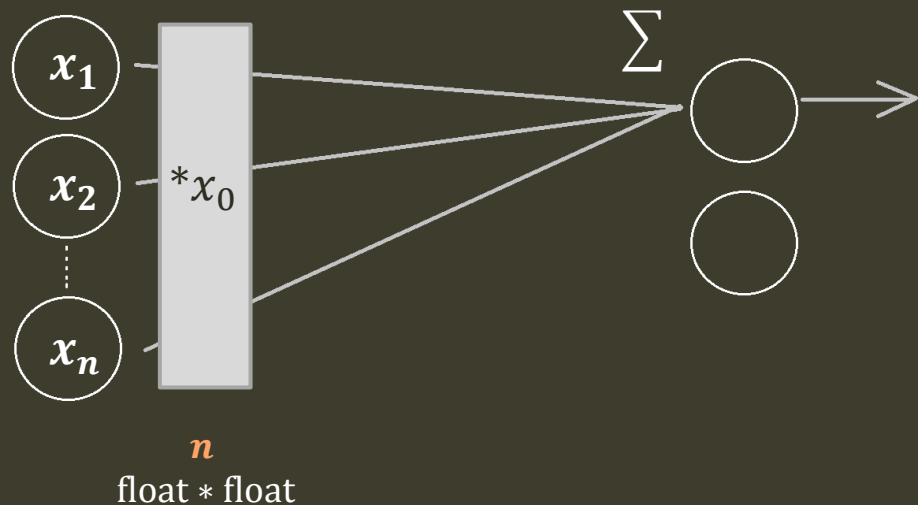


# Post-training quantization



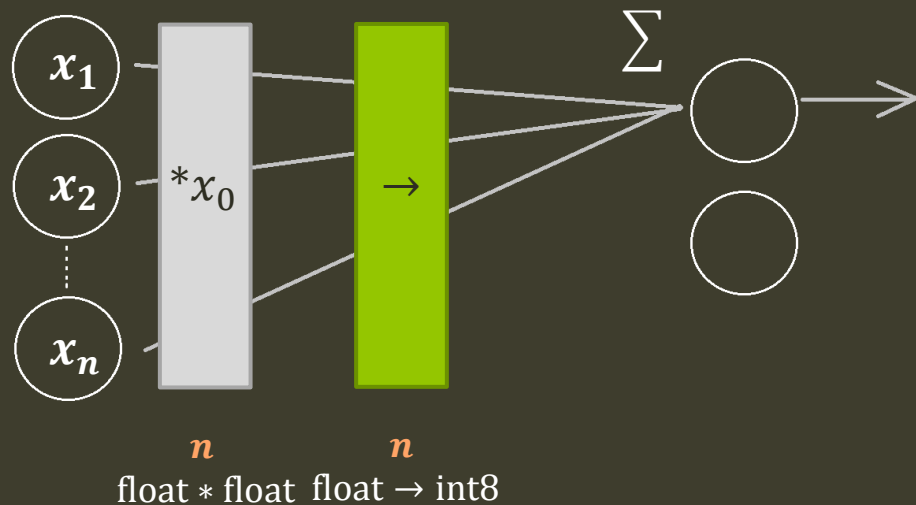
$$input = \sum_i x_i w_i$$

# Post-training quantization



$$\text{input} = \sum_i x_i w_i$$

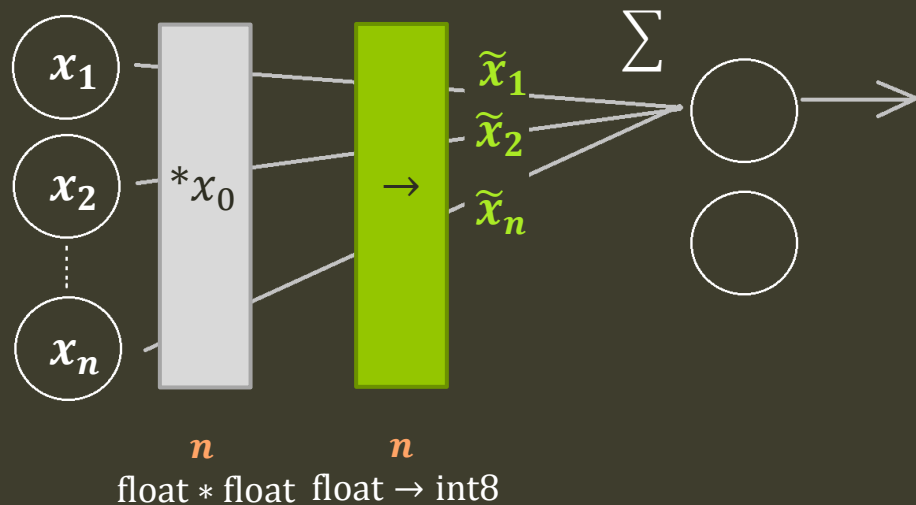
# Post-training quantization



$$\text{input} = \sum_i x_i w_i$$

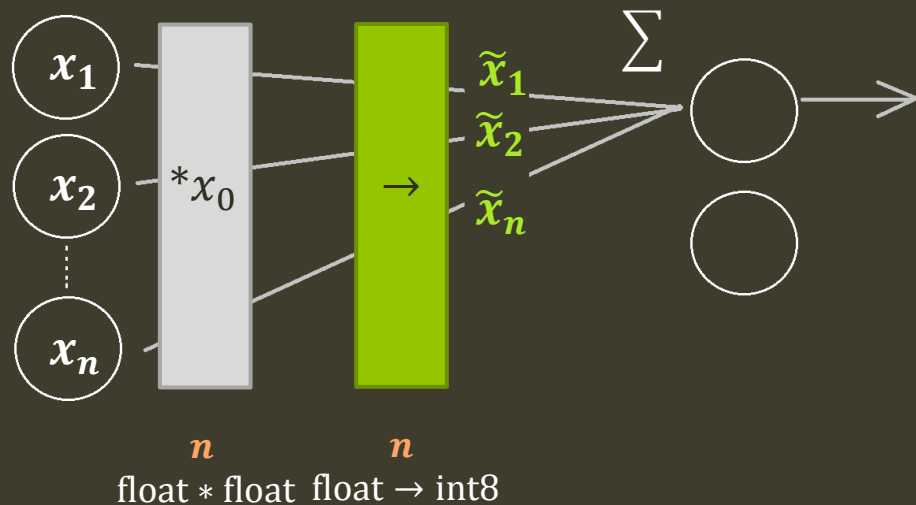


# Post-training quantization



$$\text{input} = \sum_i x_i w_i$$

# Post-training quantization

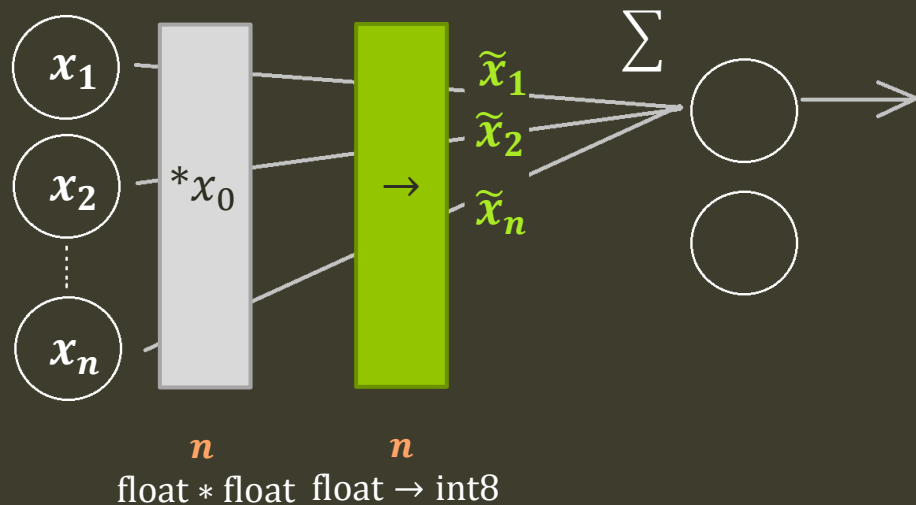


$$x_i \rightarrow \tilde{x}_i$$

**float**      **int 8**

$$input = \sum_i x_i w_i$$

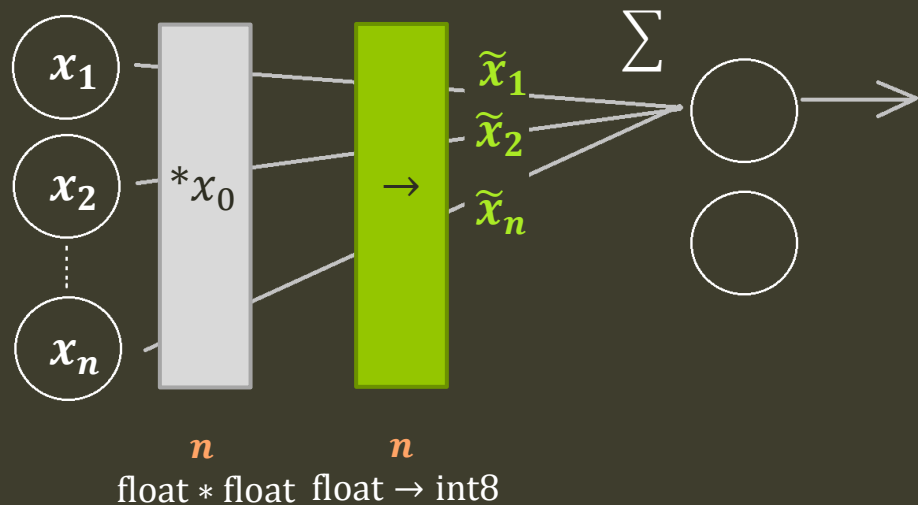
# Post-training quantization



$$\begin{aligned} x_i &\rightarrow \tilde{x}_i \\ \text{float} &\quad \text{int 8} \\ w_i &\rightarrow \tilde{w}_i \end{aligned}$$

$$\text{input} = \sum_i x_i w_i$$

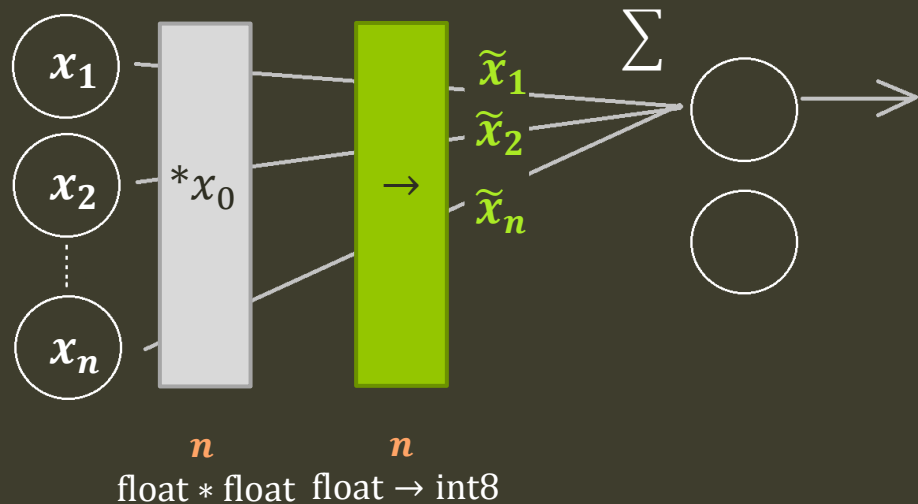
# Post-training quantization



$$\begin{aligned} x_i &\rightarrow \tilde{x}_i \\ \text{float} &\quad \text{int 8} \\ w_i &\rightarrow \tilde{w}_i \end{aligned}$$

$$\text{input} = \sum_i x_i w_i = \sum_i \frac{\tilde{x}_i}{x_0} \frac{\tilde{w}_i}{w_0} = (x_0 w_0)^{-1} \sum_i \tilde{x}_i \tilde{w}_i$$

# Post-training quantization

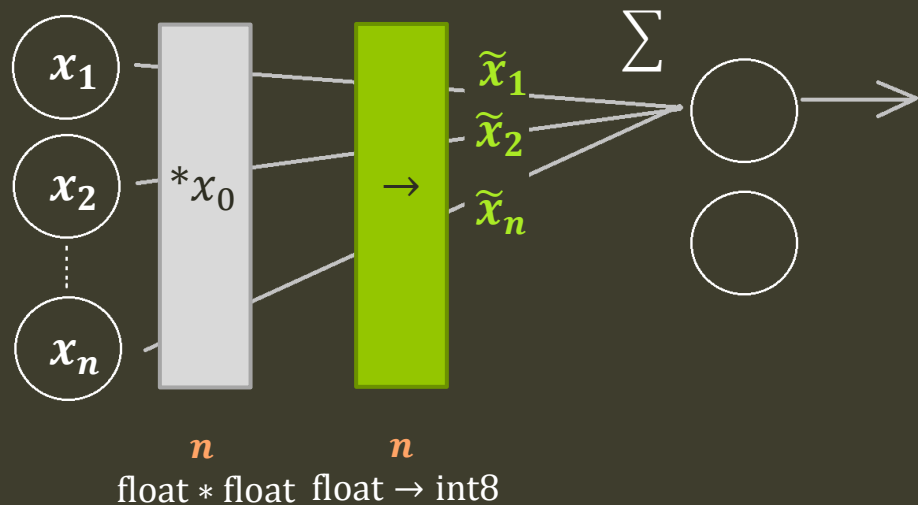


$$\begin{aligned} x_i &\rightarrow \tilde{x}_i \\ \text{float} &\rightarrow \text{int 8} \\ w_i &\rightarrow \tilde{w}_i \end{aligned}$$

$$\text{input} = \sum_i x_i w_i = \sum_i \frac{\tilde{x}_i}{x_0} \frac{\tilde{w}_i}{w_0} = (x_0 w_0)^{-1} \sum_i \tilde{x}_i \tilde{w}_i$$

$i$  медленно                       $i$  быстро

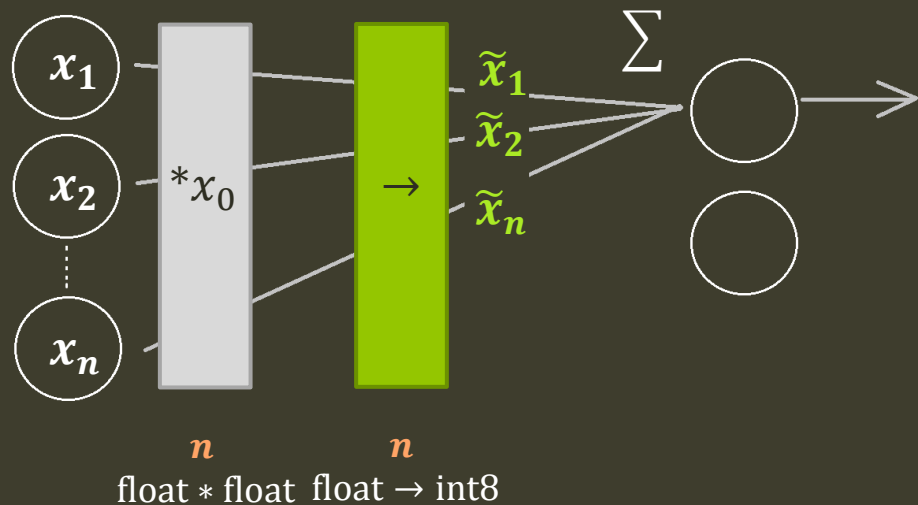
# Post-training quantization



$n * m$  произведений *float*

$$\text{input} = \sum_i x_i w_i = \sum_i \frac{\tilde{x}_i}{x_0} \frac{\tilde{w}_i}{w_0} = (x_0 w_0)^{-1} \sum_i \tilde{x}_i \tilde{w}_i$$

# Post-training quantization



$n * m$  произведений *float*



$n$  произведений *float*

$n * m$  произведений *int 8*

$1$  деление *float*

$$\text{input} = \sum_i x_i w_i = \sum_i \frac{\tilde{x}_i}{x_0} \frac{\tilde{w}_i}{w_0} = (x_0 w_0)^{-1} \sum_i \tilde{x}_i \tilde{w}_i$$

# During-training binarization

1. Создать обычную модель
2. Добавить квантификаторы в каждый слой
3. Обучить, бинаризуя веса и активации на лету
4. Готово! У модели **бинаризованные** веса



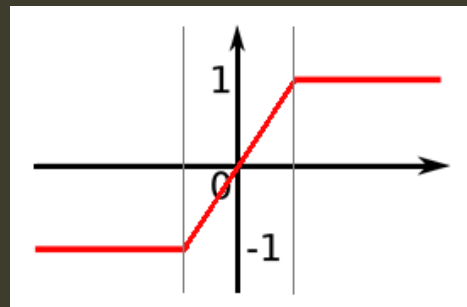
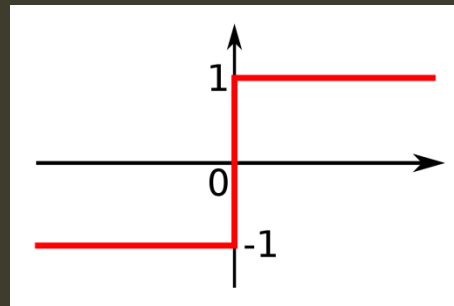
# During-training binarization

Пример квантификатора:

-используется для бинаризации активаций и весов

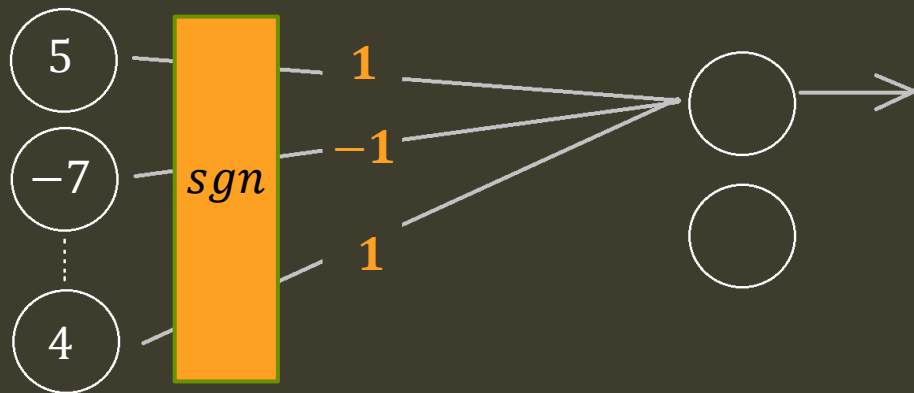
Для обучения нужен градиент.  
Как считать градиент у разрывной функции?

**А градиент будем считать для этой функции:**



зона обучения

# During-training binarization



Не нужно хранить никакие константы

Нет деления, в отличие от post-training!

$$input = \sum_i x_i w_i = \sum_i (\pm 1) * (\pm 1)$$

# Модель распознавания изображений

32



32

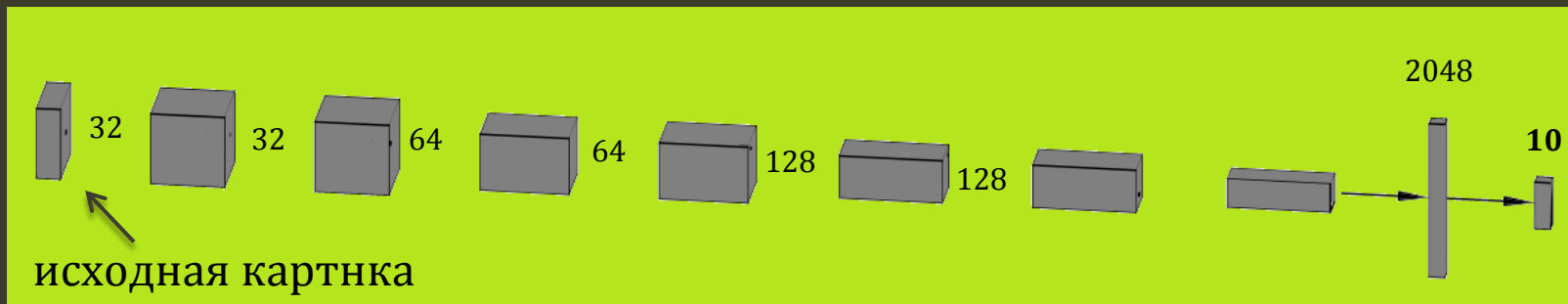
Dataset – CIFAR-10

**10 классов:**

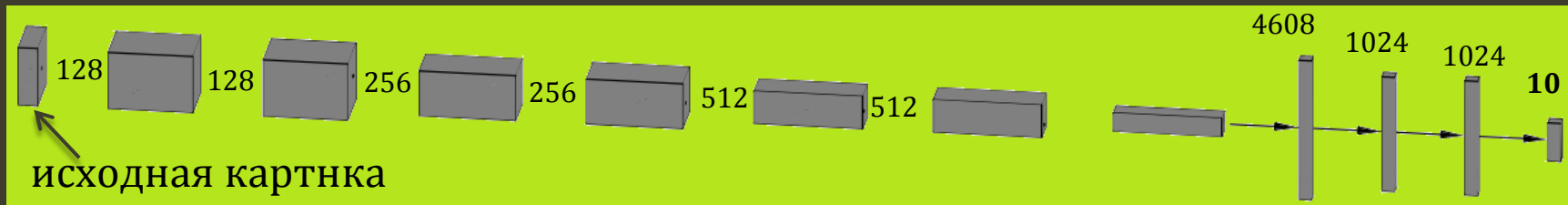
- Самолет
- Машина
- Птичка
- ...
- Грузовик

# Модель распознавания изображений

## Для Post-training



## Для During training



# Модель распознавания изображений



1.18 Мбайт

88.14 %

# Модель распознавания изображений

1.18 Мбайт

88.14 %

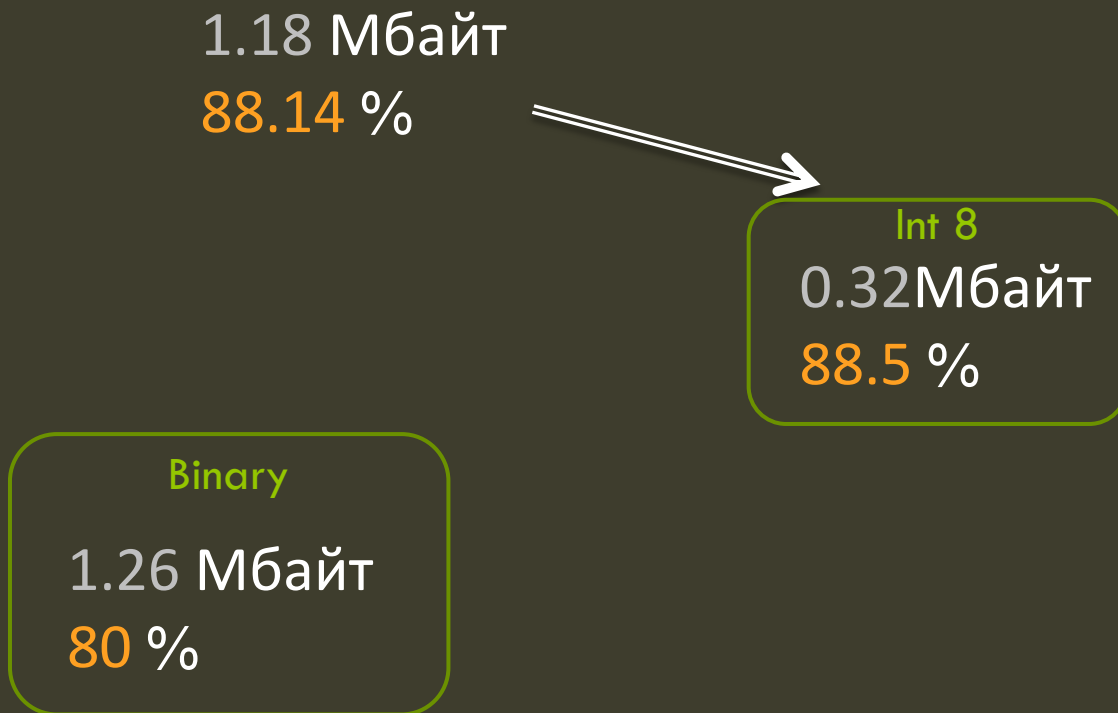


Int 8

0.32 Мбайт

88.5 %

# Модель распознавания изображений



# Итоги





# Итоги

## Post-training **int 8**

- + Точность
- + Приличное ускорение
- Деление  $\frac{\text{float}}{\text{float}}$
- + Сжатие x4

# Итоги


## Post-training **int 8**

- + Точность
- + Приличное ускорение
- Деление  $\frac{\text{float}}{\text{float}}$
- + Сжатие x4

## During-training **1 bit**

- Точность похуже
- + Дикое ускорение
- + Только бинарное сложение и умножение

Не вошло в выпуск, но очень интересно



# Модель распознавания изображений

