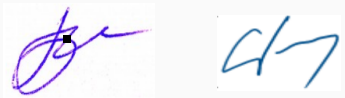
Московский Авиационный Институт

# Квантизатор нейронных сетей с эффективным матричным перемножением

Студент: Бронников Максим Андреевич

Руководитель: Ревизников Дмитрий Леонидович

Москва
2021



# Мотивация

На текущий момент различные embedded устройства окружают нас повсюду. Такие устройства характеризуются:

- Слабым процессором (тактовая частота 180 MHz у STM32F446).
- Небольшим количеством оперативной памяти (128Kb RAM в STM32F446)
- Низкой стоймостью ($\approx 650p$ для STM32F446), что делает их настолько популярными.
- Отстутсвием hardware поддержки float-pointing вычислений в некоторых моделях (процессоры Cortex Arm M3 и ниже).

# Мотивация

**Нейронные сети** - мощный и незаменимый инструмент при работе с данными из окружающей среды.

Работа полносвязного слоя описывается формулой:

$$o = \sigma(xW + b)$$

Сверточные сети могут быть приведены к данной формуле с помощью im2col преобразования, а рекуррентные сети содержат полносвязные слои внутри себя.

Операция перемножения матриц вычислительно трудная, что является ограничением для использования глубоких моделей в множестве задач.

# Мотивация

Существуют такие ускорители, как:

- Нейронные процессоры (Huawei, Samsung, Qualcomm).
- Мемристорные кроссбары (Hewlett-Packard).

Эти устройства отличаются высокой производительностью и малым энергопотреблением, однако имеют проблемы:

- Ограниченное число представимых чисел.
- Небольшой объем памяти (менее 1Mb).

# Квантизация

Один из инструментов оптимизации нейронных сетей - **квантизация** параметров слоев модели, а именно замена типа данных с *float32* на *intK*, где *K* - количество бит на каждый из весов или входов сети.

Преимущества:

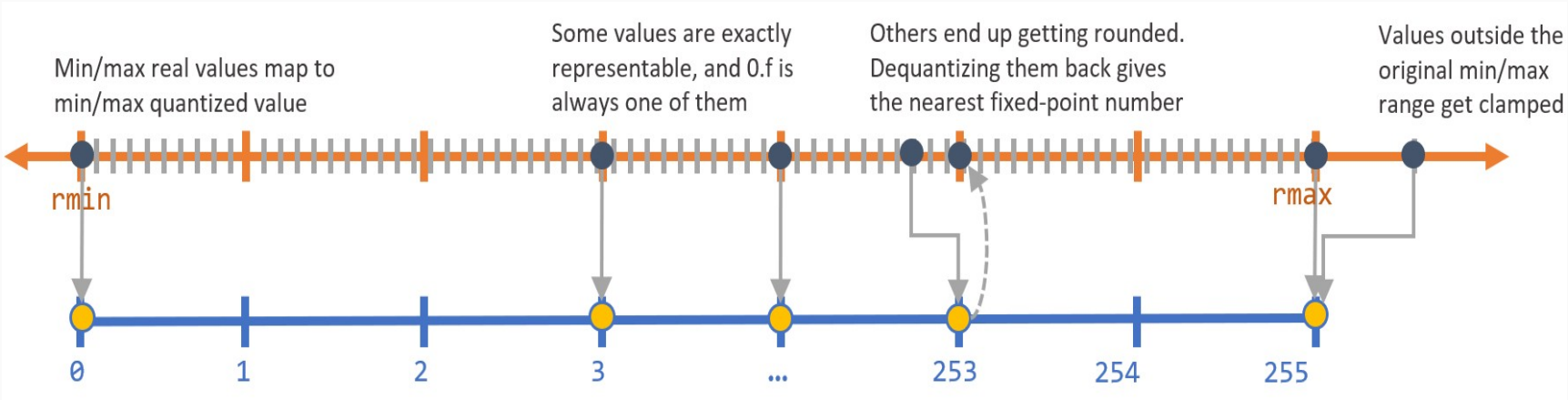
- Значительное снижение занимаемого объема как в долговременной, так и в оперативной памяти.
- Увеличение производительности за счет использования целых чисел малого размера.

Недостатки:

- Потеря точности результатов.

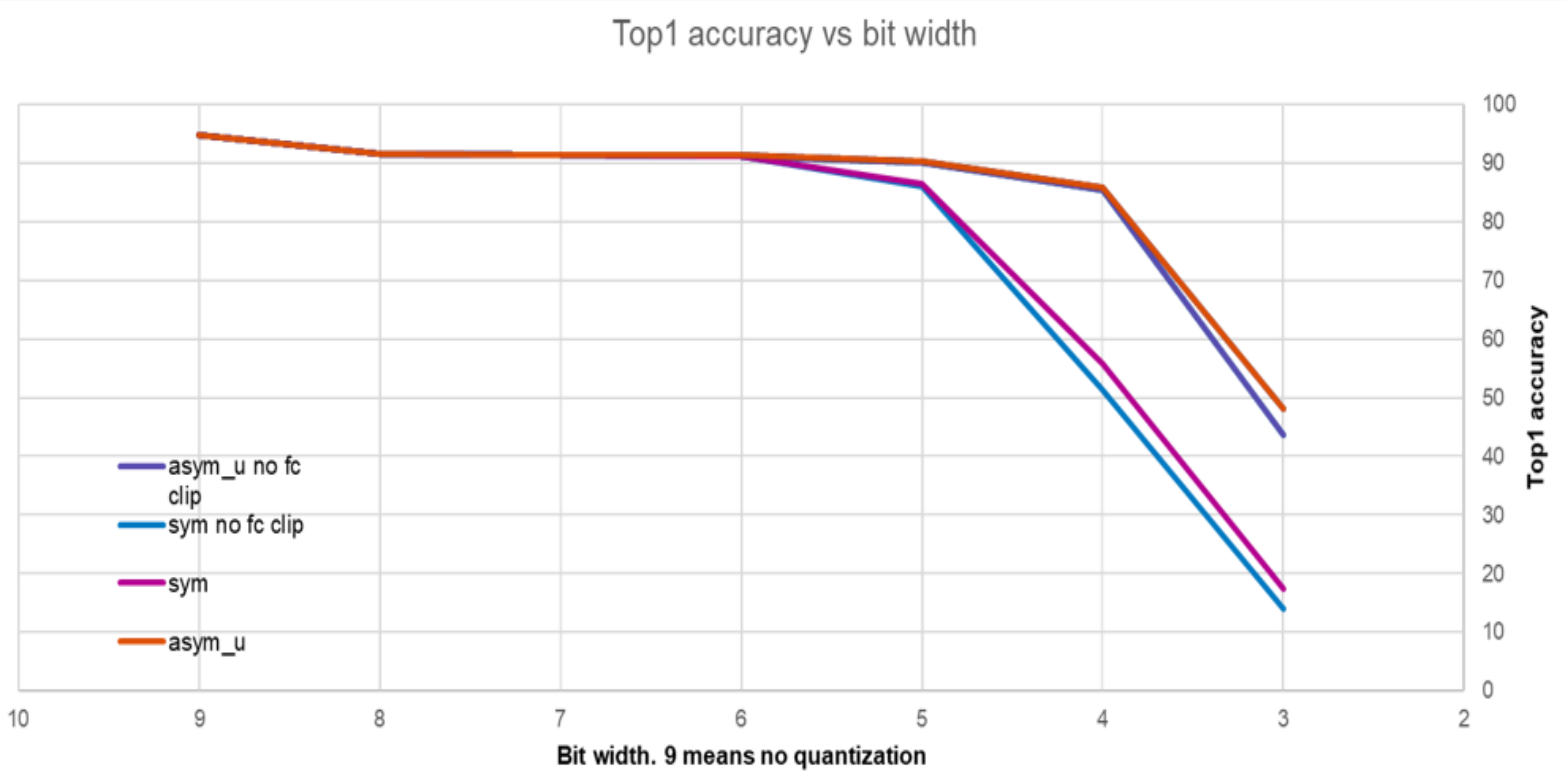
# 8-битная квантизация

Для квантизации весов(активаций) определяется диапазон возможных значений, на который равномерно отображаются $256$ уровней квантизации. Чтобы квантизовать значение, находится ближайший к нему уровень и берется соответствующий *int8*:



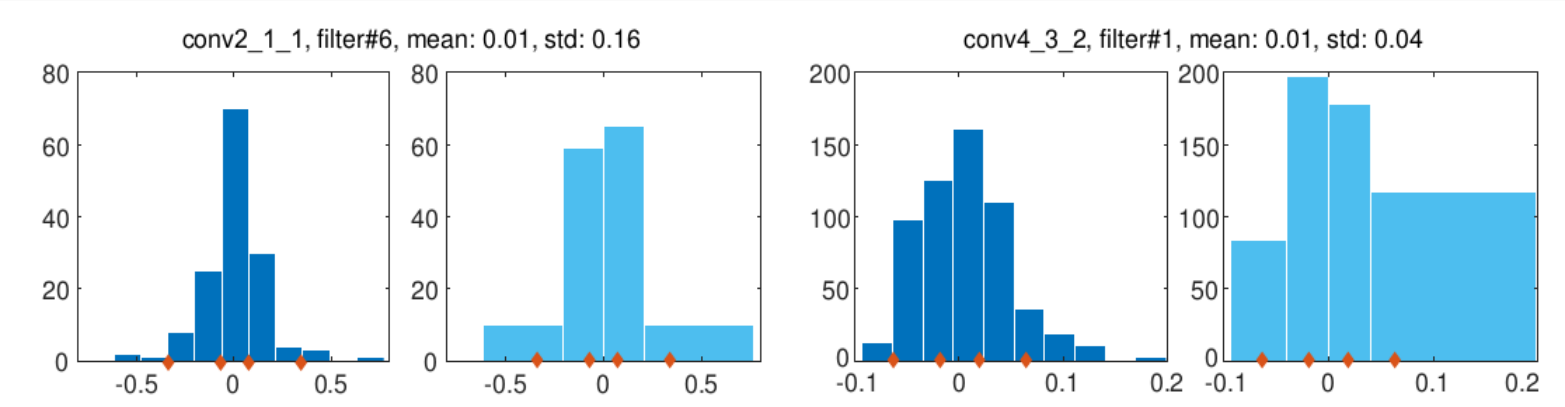
# Попытки снизить размер данных

Попытки использовать классический метод с меньшим количеством бит на параметр зачастую неудачны:



# Проблема

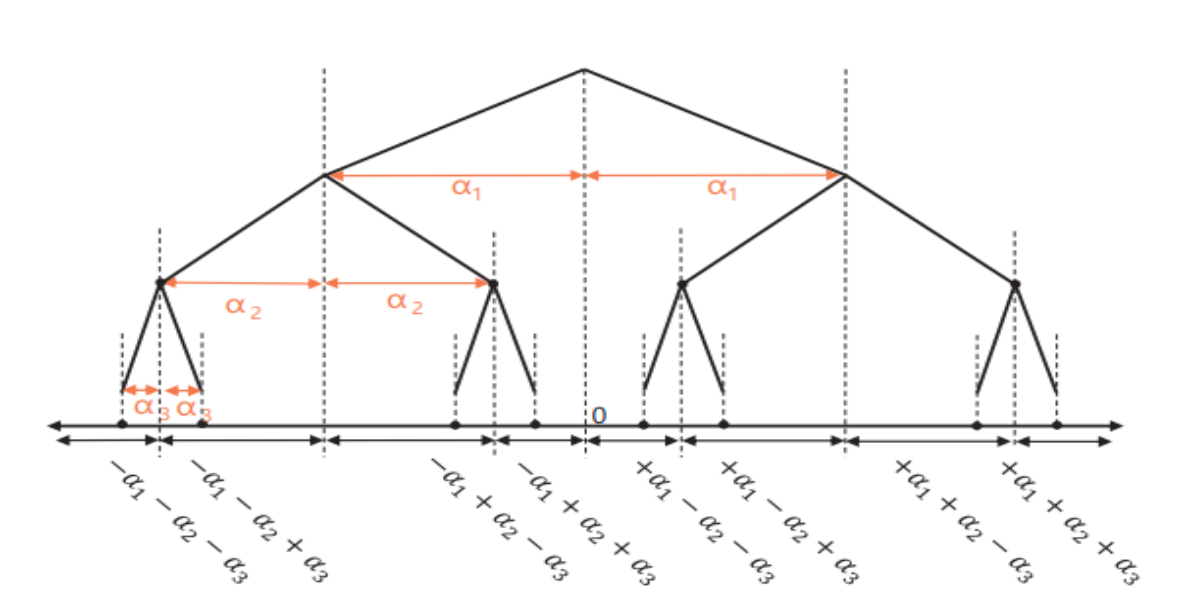
Веса и входные значения каждого слоя имеют своё собственное распределение, что не учитывает классический алгоритм. При использовании уровней квантизации, которые находятся ближе к плотным местам распределения, можно точнее представлять значения параметра.



# Предлагаемый метод

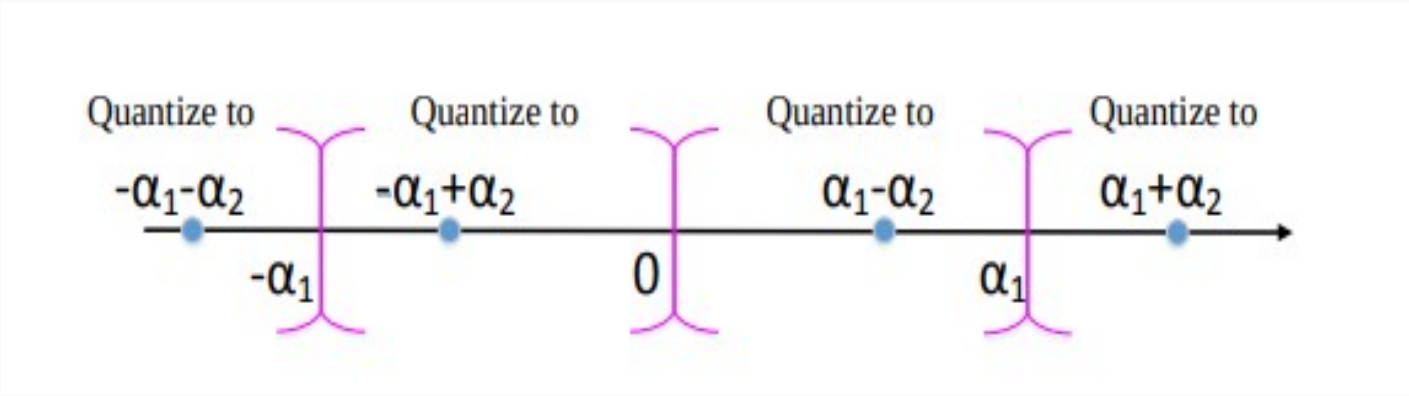
Уровни квантизации распределяются не равномерно, а с помощью обучаемого базис-вектора $a$. Уровни квантизации $q^l$ определяются скалярным произведением со всевозможными кодировками $e^l \in \{-1, 1\}^K$, $1 \leq l \leq 2^K$:

$$q^l =< e^l, a >= e^l a^T = \sum_{i=1}^{K} a_i e_i^l$$



# Предлагаемый метод

Сама же квантизация и деквантизация производятся тем же образом, что и в классическом методе с использованием уровней квантизации:




Для этого находятся интервалы, для которых все значения будут иметь одинаковую целочисленную кодировку. Границы таких интервалов определяются как середины между уровнями квантизации.

# Работа слоя

Битовая матрица кодировок значений (как весовых, так и входных) для одного нейрона:

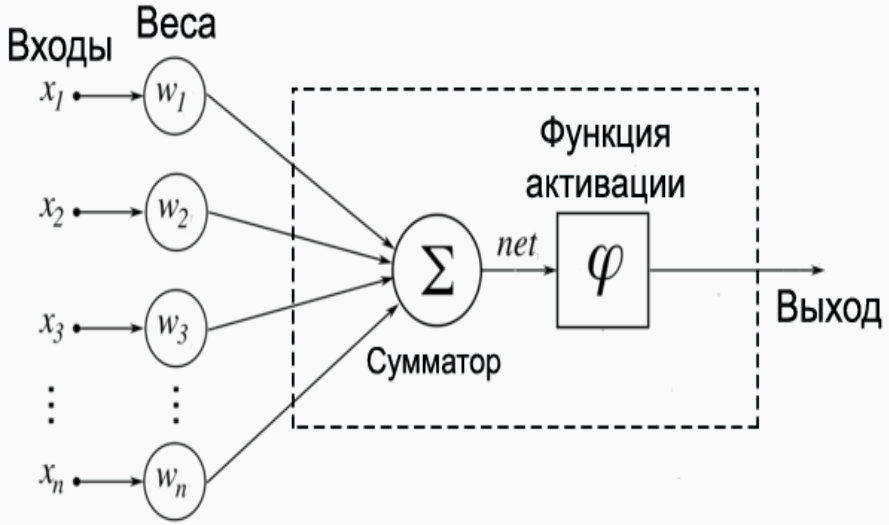
$$B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{K1} & b_{K2} & \dots & b_{Kn} \end{pmatrix} \in \{-1, 1\}^{K \times n}$$

Обозначим каждую строку матрицы $b_j = (b_{j1}\ b_{j2}\ \dots\ b_{jn})$.

# Работа слоя

Обозначим $x$ - входной вектор активаций, а $w$ - вектор весов, тогда выход сумматора нейрона расчитывается:

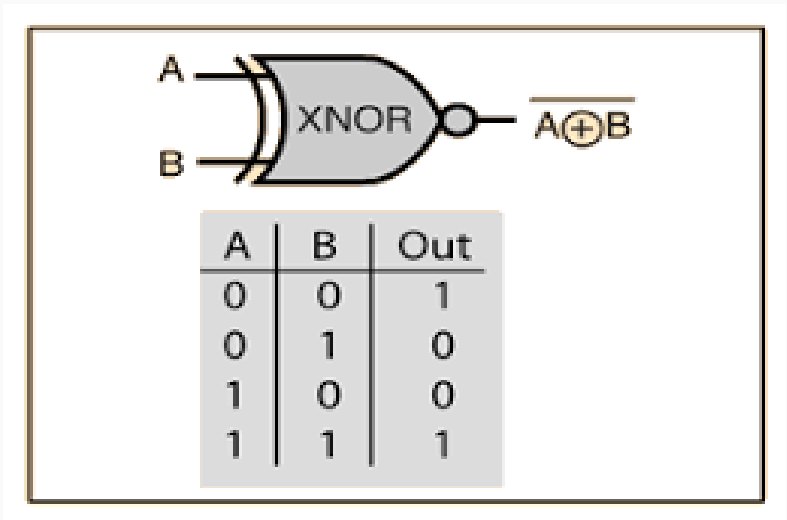
$$net = \sum_{m=1}^{n} x_m w_m \approx \sum_{m=1}^{n} \sum_{i=1}^{K^x} (b_{im}^x a_i^x) \sum_{j=1}^{K^w} (b_{jm}^w a_j^w) = \sum_{i=1}^{K^x} \sum_{j=1}^{K^w} a_i^x a_j^w (b_i^x \odot b_j^w)$$



# Работа слоя

Побитовое скалярное произведение $\odot$ может быть получено с помощью следующих побитовых операций:

XNOR:




| A | B | Out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

POPCNT:

Количество единиц в бинарном представлении числа.

# Обучение

Для подбора оптимальных параметров квантизации $a^x$, $a^w$ предлагается использование алгоритма **Quantization Error Minimization**, который минимизирует ошибку между реальным значением параметра и тем, к которому он преобразуется путем квантизации.

Метод позволяет проводить квантизацию уже обученных сетей (*post-training квантизация*).

# Обучение

**Quantization Error Minimization:**

**Вход:** данные $x$ и базис-вектор $a_0$, количество иттераций $N$.
**Выход:** обновленный базис-вектор $a$.

1. Квантизовать $x$ в матрицу $B$ с помощью $a_0$.
2. Для $\forall i \in \{1, ..., N\}$:
   2.1 $a_i = (BB^T)^{-1}Bx$
   2.2 Обновить $B$ с помощью $a_i$;
3. Вернуть $a = a_N$.

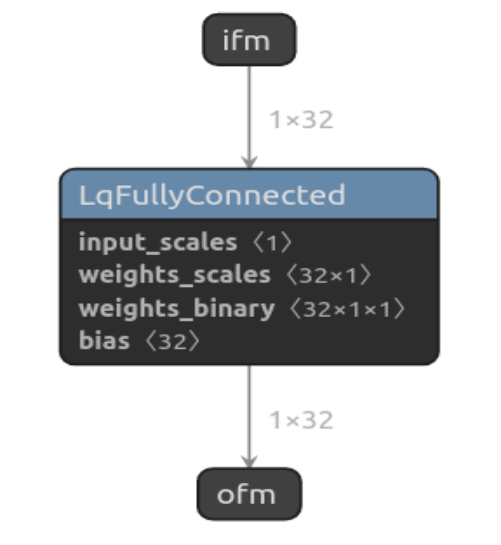
Поскольку $BB^T$ может быть неопределена и расчет обратной матрицы вычислительно сложен, для 2.1 в данной работе используется градиентный спуск с L2 регуляризацией!

# Результаты

1. Реализован новый нейронный слой в фреймворке Samsung ONE.
2. Написано вычислительное ядро в интерпретаторе с использованием побитовых операций.
3. Создан инструмент, который переводит обученные нейронные сети в квантизованный формат.

# Нейронный слой

Новый квантизованный нейронный слой реализван в графовых представлениях фреймворка с учетом оптимального способа хранения бинарных весов, параметров квантизации и вектора смещений.




Описана структура взаимодействия слоя с соседними операциями - вершинами графа модели.

# Ядро интерпретатора

Для того, чтобы было возможно исполнять модели с квантизованным слоем, реализовано вычислительное ядро в интерпретаторе luci_interpreter фреймворка.

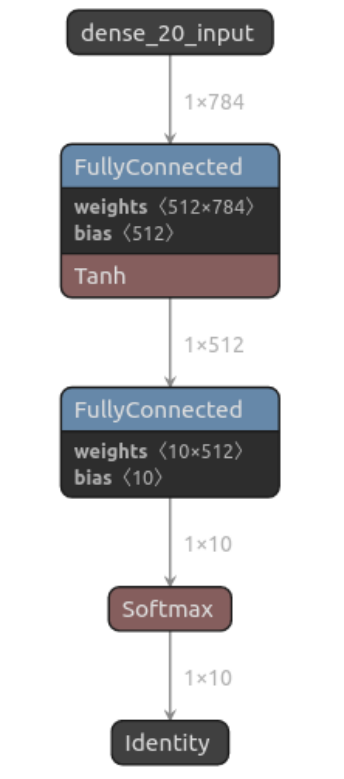
Перед каждым выполнением расчетов необходимо проводить квантизацию входных значений в бинарный формат. Поиск оптимальных кодировок для каждого значения выполняется бинарным поиском.

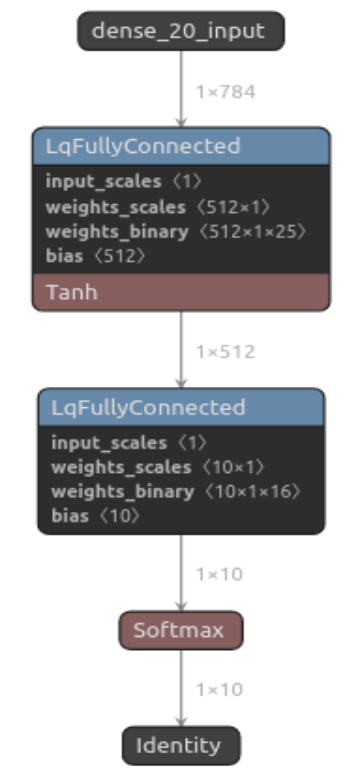
Основные расчеты выполняются с использованием побитовых операций, что ускорило производительность.

# Квантизатор

Добавлен новый инструмент - квантизатор lquantizer.

Он проходит по графу и заменяет все FullyConnected слои на квантизованные, обученные при помощи QEM путем прогонки реальных данных через сеть.



# Полученные показатели

Для сравнения результатов была спроектирована сеть из 2-х слоёв и обучена на датасете MNIST.

**Таблица 1:** Показатели исходной и квантизованных моделей

| - | Время работы | Размер | Accuracy |
|---|---|---|---|
| fp32 | **584.829 мкс** | 2.1 Mb | 97.19 % |
| int3 | **152.743 мкс** | 218.3 Kb | 95.59 % |
| int2 | **84.7845 мкс** | 147.2 Kb | 88.86 % |
| int1 | **44.7829 мкс** | 76.2 Kb | 75.11 % |