Возможная постановка задачи – достичь улучшения качества и скорости работы нерйонной сети, решающей в режиме онлайн задачу классификации изображений | детектирования объектов | сегментация изображений в привязке к платформе. В интернете уже можно найти как описание современных архитектур сетей, так и их код.

Разработанную модель можно хотябы в теории привязать к конкретному устройству. Если брать устройство на linux с процессором arm (raspberiPi, Jetson Nano), то провести замер скорости работы будет просто, т.к. для arm`ов есть откопмилированные библиотеки нейронных сетей.

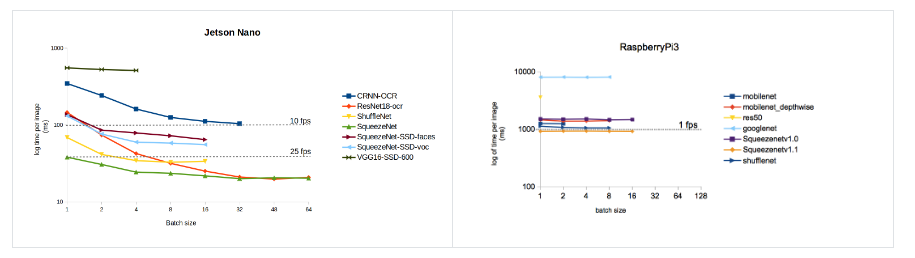
Я прочитал обзор платформ для запуска нерйосетей (<https://habr.com/ru/company/recognitor/blog/468421/>). Самые популярные из них для DIY – Jetson Nano («RaspberriPy» c GPU от nVidia) , RaspberriPi.

Тестирование

<https://github.com/jolibrain/dd_performances>

В этом репозитории представлены результаты тестирования различных нейронных сетей для обработки изображений или видео на скорость работы (по числу обрабатываемых кадров в секундку) на различных вычислительных устройствах (декстоп GPU, мини-GPU с микрокомпьютером, RaspberriPi).

На данных рисунках отображены графики FPS для относительно слабых вычислителей. У них fps не поднимается выше 25. Можно поставить задачу преобдоления или приближения к этому барьеру путем квантования нейросети. Нужно поискать данные о запуске квантованных сетей для аналогичных задач.

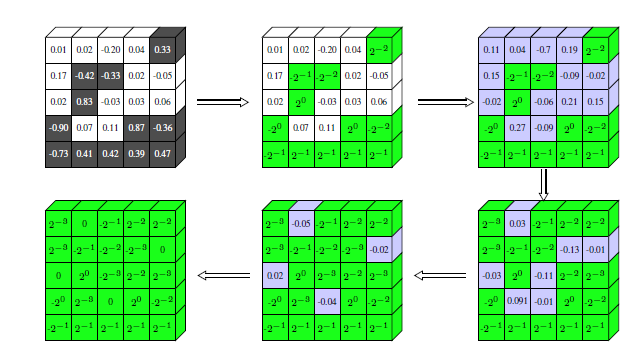
Начало обзора новых алгоритмов квантования ( возможно отсюда можно взять новые идеи)

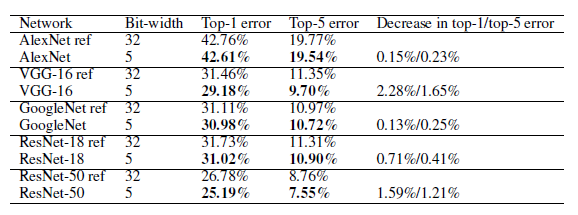
INCREMENTAL NETWORK QUANTIZATION: TOWARDS

LOSSLESS CNNS WITH LOW-PRECISION WEIGHTS

В статье предложен алгоритм квантования произвольной нейронной сети методом итеративного квантования и переобучения: на каждой итерации веса делятся на две группы при помощи эвристического правила. Веса первой группы сразу квантуются по уровням степени двух, веса второй группы оптимизируются на исходном наборе данных. Процедура деления на группы и квантования первой половины повторяется рекурсивно, пока не будет получена неделимая группа весов. Веса наименьшей группы обращаются в 0.

На рисунке изображен массив весов слоя в процессе квантования: Зеленым обозначены квантованные на данном или предудыщих шагах веса, серым – откорректированные на данном шаге веса. Группа из 4 весов на предпоследнем шаге обнуляется.



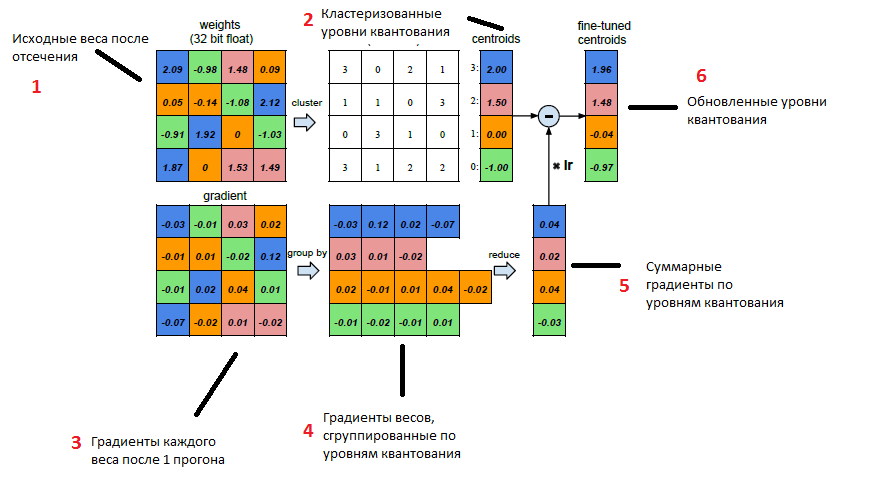


DEEP COMPRESSION: COMPRESSING DEEP NEURAL

NETWORKS WITH PRUNING, TRAINED QUANTIZATION

AND HUFFMAN CODING

В этой статье представлен метод сжатия нейронной сети при помощи квантования с переменным шагом и последующей корректировкой: на первом этапе выпоняется усечение весов (превращение в 0) по итеративному адаптивному алгоритму. После этого выполняется квантование оставшихся коэффициентов с переменным шагом. Уровни квантования выбираются на основе кластеризации K-средних. Веса квантуются по полученным k-уровням, после этого выполняется дополнительные итерации обучения методом градиентного спуска по исходной обучающей выборке. k-й уровень квантования корректируется на сумму градиентов весов, квантованных до этого уровня.



Вверху слева изображени исходный массив весов после отсечения. На втором схерху рисунке изображены квантованные веса, но в самом массиве весов лежат индексы, по которым в словаре можно найти сами веса. Это еще одна оптимизация. Сами уровни квантования, полученные в результате кластеризации изображены в столбце справа.

В нижней строке изображен процесс обновления уровней квантования градиентами весов.

Задачу можно поставить так:

1 вариант: Квантование нейросети архитектуры \_\_\_\_\_ (выбрать существующую модель) для решения задачи детектирования объектов/классификации, пригодной к работе в режиме онлайн на микрокомпьютере

2 вариант: разработка алгоритма квантования нейросети в виде плагина для библиотеки pytorch

3 вариант: (продолжение 1 варианта) разработка системы распознавания (нужно взять конкретную задачу) в реальном времени с использованием нейронной сети на базе микрокомпьютера