СТРАТЕГИЯ СЖАТИЯ НЕЙРОННЫХ СЕТЕЙ СЕМЕЙСТВА YOLOV5 ДЛЯ ВСТРАИВАЕМЫХ РЕШЕНИЙ

Г. А. Ковбаса

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Беларусь
E-mail: g.kovbasa@gmail.com

Технология обнаружения объектов всегда была одним из важных направлений исследований в области компьютерного зрения. Решения на основе искусственных нейронных сетей становятся все более совершенными и популярными. В данной работе произведена оценка результатов детектирования различных нейронных сетей на датасете MS COCO 2017. В качестве основы для дальнейшей оптимизации была выбрана сеть YOLOv5 и объединена с ShuffleNet V2 для ее облегчения. Производится оптимизация параметров сети, в результате которой количество параметров снижается более чем на 30%, но точность снижается всего на 4%. Далее проводится квантование полученной модели на основе методов, предоставленных фреймворком PyTorch. После квантования точность снижается от 2% до 5%.

Ключевые слова*: YOLOv5, обрезка нейронных сетей, квантование.*

Производительность нейронных сетей для детектирования объектов остается невысокой для использования в режиме реального времени на встраиваемых решениях в связи с большим количеством параметров сети и высокой вычислительной сложностью модели. В данной работе предлагается облегченная модель на основе YOLOv5 [1]. YOLOv5— это семейство моделей обнаружения объектов с составным масштабированием, обученных и протестированных на наборе данных MS COCO [2]. Представленная в данной статье модель использует ShuffleNet V2 в качестве основной сети. ShuffleNet V2 [3] — чрезвычайно эффективная CNN для встраиваемых и мобильных устройств. Она заимствует сетевую архитектуру быстрого доступа, аналогичную DenseNet [4]. Использование ShuffleNet V2 дает увеличение скорости, не оказывая значительного влияния на точность детектирования, как указано в таблице 1.

Методы сжатия нейронной сети. Облегченная модель YOLOv5 на основе ShuffleNet V2 все еще имеет пространство для оптимизации и сжатия размеров. В этой работе используются методы обрезки каналов и слоев в целях снижения требований к оборудованию. Все операции производятся на предварительно обученной модели и их последовательность продемонстрирована на рис.1.

Рис. 1. Последовательность операций

Согласно рис. 1 процесс уменьшения размеров YOLOv5 состоит из следующих шагов:

1. Разреженное обучение. Разреженность слоев сети для дальнейшей обрезки каналов обеспечивается применением L1 регуляризации к коэффициентам масштабирования [5]. Целевая функция потерь представлена в формуле

$$Loss = \sum l(f(x, W), y) + \lambda \sum g(\gamma), \quad (1)$$

где *Loss* - потери сети при обучении; второй член представляет собой регулярный ограничивающий член L1 коэффициента γ слоя BN; x и y — вход и выход обучения соответственно; W — обучающий параметр в сети; λ - коэффициент регуляризации.

2. Обрезка каналов. Обрезка каналов значительно уменьшает количество параметров и размер файла весов. Процедура производится итеративно до достижения наилучшего результата в соотношении количества параметров и точности детектирования.

3. Обрезка слоев. Сочетание обрезки слоев и каналов существенно уменьшает вычислительную сложность модели.

4. Тонкая подстройка весовых коэффициентов сети. Производится для оценки модели и восстановления точности.

5. Квантование полученной модели. Дальнейшее квантование позволяет уменьшить размер файлов и увеличить скорость обработки кадров в режиме реального времени при выполнении на CPU.

Процесс разреженного обучения. На рис. 2 отображен процесс разреженного обучения. Во время разреженного обучения mAP_0.5 будет сначала постепенно уменьшаться, а затем медленно повышаться обратно после того, как скорость обучения уменьшится на более позднем этапе обучения. По мере обучения коэффициент γ постепенно приближается к 0, что указывает на то, что γ постепенно становится разреженным.

Обрезка каналов и слоев. Обрезку можно начинать после достаточной разреженности нейронной сети. Обрезку можно разделить на обрез-

ку каналов и обрезку слоев. Обе процедуры оценивают γ, поэтому степень разреженности после обучения напрямую влияет на результат.

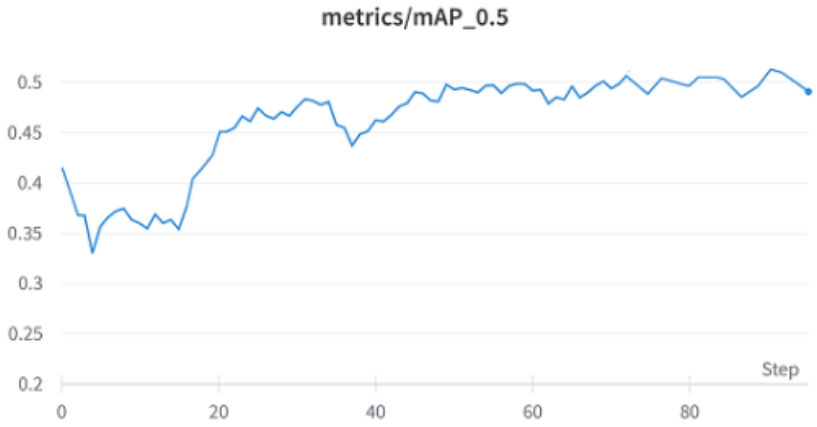

Рис. 2. Разреженное обучение

Веса, значения которых близки к нулю (в соответствии с заданным порогом) будут удалены, так как мало влияют на конечную точность модели и не повлияют на окончательный результат обнаружения.

Обрезка слоев является производной от предыдущей стратегии обрезки канала. Производится сортировка среднего значения γ для каждого слоя и выбирается слой с наименьшим значением для сокращения.

Таблица 1

Результаты обрезки каналов и слоев

Модель	Размер изображения	mAP@0.5	mAP@0.5:0.95	Параметры	Размер файла
YOLOv5s	640*640	56.0	37.2	7.23M	14M
YOLOv5-Shuffle	640*640	51.6	35.7	5.39M	10.9M
YOLOv5-Shuffle-pruned&tuned	640*640	50.1	32.5	3.58M	6.4M

В таблице 1 указаны изменения количества параметров и точность детектирования до и после нескольких итераций обрезки и тонкой подстройки. Размер модели уменьшился почти на 55%, но точность осталась практически неизменной. Суммируя вышеперечисленные характеристики, определено, что обрезка каналов и слоев была эффективна.

Квантование. После процедур обрезки и тонкой подстройки было применено статическое квантование при помощи средств фреймворка PyTorch. Квантование приводит к снижению точности, поэтому вместо конфигурации квантования по умолчанию, где используется наблюдатель MinMax, целесообразно использовать наблюдатель гистограммы,

чтобы улучшить показатели [6]. Результаты квантования приведены в таблице 2.

Результаты. Экспериментальные результаты, представленные в таблице 2, показывают, что точность распознавания сети YOLOv5-Shuffle-q16 остается на уровне 87,1% от уровня YOLOv5s на датасете MS COCO 2017. При этом количество параметров было уменьшено на 78%. Время детектирования стало на 56,8% меньше, чем у YOLOv4-Tiny, что соответствует требованиям для выполнения детектирования в реальном времени.

Таблица 2

Результаты тестирования моделей на MS COCO 2017

Модель	mAP@ 0.5	mAP@ 0.5:0.95	Параметры	Размер (M)	Inference	GFlops
YOLOv5l	67.3	49.0	46.5M	91.4	320ms	109.1
YOLOv5s	56.8	37.4	7.2M	14.5	131ms	16.5
YOLOv4-tiny	42.0	**22.0**	5.9M	23.1	125ms	6.9
YOLOv5-Shuffle-P&T	50.1	32.5	3.6M	6.4	65ms	3.4
YOLOv5-Shuffle-q16	49.1	32.0	-	6.4	54ms	2.6
YOLOv5-Shuffle-q8	47.4	27.6	-	3.2	37ms	1.7

Все тестирование было произведено на устройстве с Intel i7-8550U. В дальнейших исследованиях эффективность алгоритмов обрезки и квантования будет продолжать изучаться для снижения влияния на точность обнаружения.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

1. YOLOv5. [Электронный ресурс]. – Режим доступа: https://github.com/ultralytics/yolov5. – Дата доступа: 02.01.2022.
2. COCO – Common Objects in Context. [Электронный ресурс]. – Режим доступа: https://cocodataset.org/. – Дата доступа: 15.02.2022.
3. Ma N., Zhang X., Zheng H.-T., Sun J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design // arXiv:1807.11164v1 [cs.CV] 30 Jul 2018. DOI: 10.48550/ARXIV.1807.11164
4. Review: DenseNet — Dense Convolutional Network (Image Classification). [Электронный ресурс]. – Режим доступа: https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803. – Дата доступа: 02.01.2022.
5. Университет ИТМО. Регуляризация. [Электронный ресурс]. – Режим доступа: https://neerc.ifmo.ru/wiki/index.php?title=Регуляризация. – Дата доступа: 05.01.2022.
6. PyTorch. Quantization. [Электронный ресурс]. – Режим доступа: https://pytorch.org/docs/stable/quantization.html. – Дата доступа: 08.02.2022.