УДК 621.397

МОДИФИКАЦИЯ ДЕТЕКТОРА ОБЪЕКТОВ YOLO ДЛЯ РЕАЛИЗАЦИИ НА ПЛИС В РЕАЛЬНОМ ВРЕМЕНИ

Васильев Д.А., *МГТУ им. Н.Э. Баумана, e-mail: vasilev.bmstu@gmail.com*
Лёвкин Т.В., *АО «НПК «Альфа-М», e-mail: tim-12345@mail.ru*
Сконников П.Н., *АО «НПК «Альфа-М», e-mail: skonnikovpn@yandex.ru*
Трофимов Д.В., *АО «НПК «Альфа-М», e-mail: samael1978@rambler.ru*

A MODIFICATION OF YOLO OBJECT DETECTOR FOR REAL-TIME IMPLEMENTATION ON FPGA

Vasilev. D.A., Levkin T.V., Skonnikov P.N., Trofimov D.V.

The principle of YOLO object detector operation is considered. The most difficult operations implemented on FPGAs have been identified. It is proposed to replace the transformations that require significant computational costs with simpler ones. For the proposed transformations, formulas for calculating the loss backpropagation are analytically derived. Using the obtained formulas, a YOLO detector based on a modified neural network was trained. This detector is implemented on the FPGA board. The created model of the vision system for image recognition works with low latency in real time and shows high quality characteristics.

Key words: digital image processing, artificial neural networks, YOLO, backpropagation.

Ключевые слова: цифровая обработка изображений, искусственные нейронные сети, YOLO, обратное распространение ошибки.

Рассмотрен принцип работы детектора объектов на изображении YOLO. Выделены операции, используемые данным детектором, реализация которых на ПЛИС затруднительна. Предложена замена преобразований, требующих значительных вычислительных затрат, на более простые. Для предложенных преобразований аналитически выведены формулы для расчёта обратного распространения ошибки. С использованием полученных формул обучен детектор YOLO, основанный на модифицированной нейронной сети. Данный детектор аппаратно реализован на плате с ПЛИС производства Xilinx. Созданный макет системы технического зрения с распознаванием образов работает с малой задержкой в режиме реального времени и показывает высокие качественные характеристики.

Введение

В современных системах технического зрения для обнаружения и распознавания объектов используются детекторы на основе искусственных нейронных сетей (ИНС) [1]. В системах, работающих в режиме реального времени, наибольшее распространение получили однопроходные детекторы, поскольку они обладают высокими качественными характеристиками при сравнительно малых вычислительных и временны́х затратах. В число таких детекторов входят алгоритмы типа YOLO [2-4].

Несмотря на то, что данные детекторы могут быть основаны на ИНС различных архитектур, иметь различные размеры и количество каналов входных изображений, а также отличаться по количеству классов распознавания, они обладают общим принципом интерпретации массива, поступающего с выхода ИНС на постобработку. Данный трёхмерный массив $\mathbf{B}$ размера $X \times Y \times N(5+C)$ представляет собой упорядоченный набор параметров определённых заранее «якорных рамок» (в англоязычной литературе «anchor boxes»), где X и Y – количество исходных позиций «якорных рамок» по двум координатам, N – количество «якорных рамок» на каждой исходной позиции а C – объём алфавита классов [2]. Такие рамки изначально располагаются на изображении в узлах сетки размером $X \times Y$. В каждом узле сетки размещается N рамок с определёнными заранее размерами x_n и y_n. Для каждой $n = \overline{1, N}$ рамки ИНС предсказывает $5 + C$ параметров: точность наведения рамки на цель IoU (Intersection over Union [5]), смещения рамки относительно исходной позиции Δx и Δy, коэффициенты корректировки исходных размеров рамки δx и δy, а также C предсказаний вероятностей p_c принадлежности объекта, ограниченного текущей рамкой, к c-му классу, $c = \overline{1, C}$.

Для того чтобы элементы массива A, поступающего с выхода последнего свёрточного слоя сети, получили описанный выше смысл, проводится преобразование $B = \mathrm{f}(A)$, приводящее элементы $a \in A$ к требуемому диапазону значений. Данное преобразование содержит экспоненциальную функцию, расчёт значений которой требует существенных вычислительных затрат на программируемых логических интегральных схемах (ПЛИС). В связи с этим, представляется целесообразным проведение более простого преобразования с точки зрения реализации на ПЛИС.

Исходное преобразование элементов выходного массива

Алгоритм YOLO предполагает обработку трёх разных видов для различных элементов выходного массива A. Для предсказания IoU, Δx и Δy выполняется поэлементное нелинейное преобразование, имеющее вид логистической функции [2-4]:

$$b = \mathrm{f}_1(a) = 1/1 + e^{-a}, \quad (1)$$

где a и b – элементы массивов A и B, соответствующих величинам IoU, Δx и Δy.

Преобразование для δx и δy имеет экспоненциальную форму [2-4]:

$$b = \mathrm{f}_2(a) = e^a. \quad (2)$$

Значения элемента b массива B для предсказаний вероятностей p_c зависят не только от элемента a, находящегося на той же позиции в массиве A, но и от других элементов массива A, соответствующих остальным классам на текущей позиции сетки для текущей якорной рамки, то есть каждый элемент b_k, $k = \overline{1,C}$ определяется по формуле [2-4]:

$$b_k = \mathrm{f}_3(a_1, a_2 \ldots a_C) = e^{a_k} \Big/ \sum_{c=1}^{C} e^{a_c}. \quad (3)$$

При работе детектора необходимо выполнять $3XYN$ операций (1), $2XYN$ операций (2) и $CXYN$ операций (3) в каждом кадре, что требует значительных вычислительных затрат на ПЛИС.

При обучении ИНС методами, основанными на стохастическом градиентном спуске и его модификациях [6], обратное распространение ошибки через слой, выполняющий преобразования (1)-(3), производится с использованием цепного правила дифференцирования сложной функции:

$$\frac{\partial L}{\partial a} = \frac{\partial L}{\partial b}\frac{\partial \mathrm{f}(a)}{\partial a}, \quad (4)$$

где $\partial L / \partial a$ – составляющая искомого градиента функции потерь, подлежащая передаче на предыдущий слой, $\partial L / \partial b$ – составляющая градиента функции потерь, поступающая с последующего слоя, а производная $\partial \mathrm{f}(a) / \partial a$ выражается аналитически через величины a, b и $\partial L / \partial b$, известные на этапе проведения операции (4).

В частности,

$$\partial \mathrm{f}_2(a) / \partial a = e^a = \mathrm{f}_2(a) = b, \quad (5)$$

то есть на практике экспонирование не производится, поскольку оно приведёт к получению рассчитанного ранее значения b.

Для функции f_1

$$\frac{\partial \mathrm{f}_1(a)}{\partial a} = \frac{e^{-a}}{(1+e^{-a})^2}. \quad (6)$$

Несложно убедиться, что более простой расчёт произведения $b(1-b)$, где $b = \mathrm{f}_1(a)$, приведёт к тому же результату.

Известно [7], что для функции f_3 производная также может быть выражена через $b_k = \mathrm{f}_3(a_1, a_2 \ldots a_C)$:

$$\frac{\partial \mathrm{f}_3(a_k)}{\partial a_c} = \begin{cases} b_k(1-b_k), & \text{при } k = c; \\ -b_k b_c, & \text{при } k \neq c. \end{cases}$$

Предложенное преобразование элементов выходного массива

Как было отмечено выше, многократное выполнение преобразований (1)-(3), требуемое для работы детектора YOLO, требует расчёта экспоненциальной функции. Методы вычислительной математики позволяют получить достаточно точное приближение экспоненты при определённом количестве учитываемых членов соответствующего ряда. Тем не менее, для обеспечения необходимой точности соответствия операций, производимых на ПЛИС, преобразованиям (1)-(3), требуется неприемлемо высокая вычислительная мощность. В настоящей работе предлагается иной подход, не требующий точного приближения преобразований (1)-(3).

Логистическая функция (1) заменяется на рациональную сигмоиду:

$$\mathrm{f}_1(a) = \frac{1}{2} \cdot \frac{a}{1+|a|} + \frac{1}{2}. \quad (8)$$

Здесь коэффициенты масштабы и сдвига, равные $1/2$, введены для того, чтобы, так же как и для функции (1), выполнялись условия $\lim\limits_{a \to -\infty} \mathrm{f}_1(a) = 0$, $\lim\limits_{a \to +\infty} \mathrm{f}_1(a) = 1$, $\mathrm{f}_1(0) = 1/2$.

Экспоненциальная функция (2) заменяется на параболу:

$$\mathrm{f}_2(a) = (a+\beta)^\alpha / \beta^\alpha. \quad (9)$$

Значение сдвига параболы $\beta = 16$ выбрано таким образом, чтобы все возможные значения знаковых чисел с фиксированной запятой, которые используются в описанной далее реализации детектора на ПЛИС, приходились на монотонно возрастающую ветвь параболы. Показатель степени принят равным $\alpha = 4$, поскольку такое значение более точно по сравнению с другими целочисленными показателями преобразует входные величины a к диапазону значений b исходного преобразования (2).

Аналогичная замена экспоненты производится и в преобразовании (3), с тем отличием, что нормировка по β^α не требуется:

$$b_k = \mathrm{f}_3(a_1, a_2 \ldots a_C) = (a_k+\beta)^\alpha / \Sigma, \quad (10)$$

где $\Sigma = \sum\limits_{c=1}^{C} (a_c+\beta)^\alpha$.

Как было отмечено выше, выражения (8)-(10) не являются точными приближениями преобразований (1)-(3). В связи с этим работа ИНС с преобразованиями (8)-(10), обученной с использованием формул (5)-(7), приведёт к получению некорректных результатов. Это означает, что выполненные замены необходимо учитывать при обучении сети.

Для обратного распространения ошибки по правилу (4) необходимо рассчитать производные функций (8)-(10):

$$\frac{\partial \mathrm{f}_1(a)}{\partial a} = \frac{1}{2} \cdot \frac{1}{(1+|a|)^2}; \quad (11)$$

$$\frac{\partial \mathrm{f}_2(a)}{\partial a} = \frac{\alpha}{\beta^\alpha}(a+\beta)^{\alpha-1}; \quad (12)$$

$$\frac{\partial \mathrm{f}_3(a_k)}{\partial a_c} = \begin{cases} \dfrac{\alpha(a_k+\beta)^{\alpha-1}}{\Sigma} - \dfrac{\alpha(a_k+\beta)^{2\alpha-1}}{\Sigma^2}, & \text{при } k = c; \\ -\dfrac{\alpha(a_c+\beta)^{\alpha-1}(a_k+\beta)^\alpha}{\Sigma^2}, & \text{при } k \neq c. \end{cases} \quad (13)$$

После того как частные производные (13) рассчита-

ны, обратное распространение ошибки через преобразование (10) может быть рассчитано следующим образом [8]:

$$\frac{\partial L}{\partial a_k} = \sum_{c=1}^{C} \frac{\partial L}{\partial b_c} \frac{\partial \mathrm{f}(a_c)}{\partial a_k}. \quad (14)$$

Однако, допущение

$$\frac{\partial L}{\partial a_k} = \frac{\partial L}{\partial b_k} \frac{\partial \mathrm{f}(a_k)}{\partial a_k}, \quad (15)$$

принятое при обучении ИНС, также приводит к корректной работе детектора.

Обучение сети по формулам (11)-(13) обладает большей вычислительной сложностью, чем расчёт по формулам (5)-(7). Однако, следует учитывать, что обучение производится на высокопроизводительной видеокарте в приемлемые сроки (от нескольких часов до нескольких дней), а обучение ИНС на ПЛИС в режиме реального времени не требуется.

Экспериментальные исследования аппаратной реализации детектора

Детектор YOLO с предложенными преобразованиями C реализован на специально разработанной плате. Внешний вид данной платы показан на рис. 1.

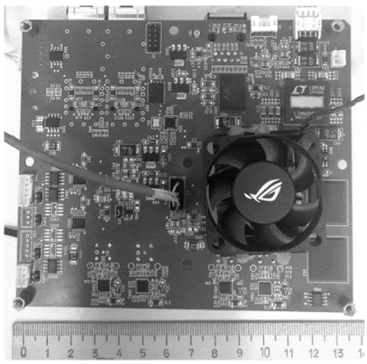

Рис. 1. Внешний вид платы распознавания образов

Вычисления производятся на ПЛИС производства Xilinx. Плата оснащена четырьмя цифровыми видеоинтерфейсами SDI, а также Ethernet и двумя SFP для подключения иных интерфейсов. Размеры платы не превышают $13 \times 11,5$ см, а средняя потребляемая мощность при работе детектора составляет не более 20 Вт.

Производится обработка цветных кадров размером до 1920×1080 пикселей, следующих с частотой 60 Гц. Размер области поиска объектов в проводимых экспериментальных исследованиях составлял 522×522 пикселя, причём данный размер при необходимости может быть увеличен.

На проход ИНС в прямом направлении затрачивается 8 мс. Дальнейшая постобработка, включающая фильтрацию, подавление немаксимумов (Non-Maximun Supression), отрисовку знакографической информации и передачу выходного кадра на отображение занимает от 3 до 6 мс, в зависимости от количества обнаруженных целей.

Распознавание производилось при помощи двух модифицированных детекторов YOLO одинаковой архитектуры на основе ИНС ResNet-18: одного – предназначенного для распознавания наземных объектов и одного – для воздушных.

По результатам валидации получено, что площадь под кривой Recall-Precision [9] составляет не менее 0,9 по каждому классу объектов, а площадь под рабочей характеристикой приёмника [10] – не менее 0,97. Такие же качественные показатели были получены при обучении ИНС по формулам (5)-(7) и обнаружении объектов с использованием формул (1)-(3).

При обработке типовых видеозаписей значения выходного массива чисел с плавающей запятой, полученного на видеокарте персонального компьютера, совпали с соответствующими значениями, рассчитанными аппаратно на плате с ошибкой менее 1 %.

Заключение

Предложенные преобразования (8)-(10) позволили создать аппаратную реализацию детектора объектов YOLO на плате. Созданный макет системы технического зрения с распознаванием образов работает с минимальной задержкой в режиме реального времени.

Выведенные аналитически формулы (11)-(13) использованы в программе, реализующей обучение модифицированной ИНС.

Эффективность предложенных преобразований подтверждена качественными характеристиками распознавания объектов, не уступающими характеристикам исходного детектора.

Литература

1. Aziz L. et al. Exploring deep learning-based architecture, strategies, applications and current trends in generic object detection: A comprehensive review. IEEE Access. 2020, vol. 8. pp. 170461-170495.

2. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE confer-ence on computer vision and pattern recognition. 2016. pp. 779-788.

3. Redmon J., Farhadi A. YOLO9000: Better, Faster, Stronger. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. pp. 7263-7271.

4. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement [Электронный ресурс]. arXiv. 8 апреля 2018. URL: https://arxiv.org/pdf/1804.02767.pdf (дата обращения: 22.07.2020).

5. Rezatofighi H. et al. Generalized intersection over union: A metric and a loss for bounding box regression. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019. pp. 658-666.

6. Shalev-Shwartz S., Ben-David S. Understanding machine learning: From theory to algorithms. Cambridge university press. 2014. 449 pp.

7. Backpropagation with Cross-Entropy and Softmax [Электронный ресурс]. ML Dawn. 2021. URL: https://www.mldawn.com/back- propagation-with-cross-entropy-and-softmax/ (дата обращения: 20.01.2022).

8. The Derivative of Softmax Function [Электронный ресурс]. ML Dawn. 2021. URL: https://www.mldawn.com/the-derivative-of-softmaxz-function-w-r-t-z/ (дата обращения: 20.01.2022).

9. Buckland M., Gey F. The relationship between recall and precision. Journal of the American society for information science. 1994, vol. 45, no. 1. pp. 12-19.

10. Van Trees H. L. Detection, estimation, and modulation theory, part I: detection, estimation, and linear modulation theory. John Wiley & Sons, 2004, 716 pp.