ПЕРСПЕКТИВНЫЕ НЕЙРОННЫЕ СЕТИ ДЛЯ РЕАЛИЗАЦИИ НА ПЛИС

А.П Береснев, И.В.Зоев
Научный руководитель: Н.Г. Марков
Национальный исследовательский Томский политехнический университет
apb3@tpu.ru

Введение

На сегодняшний день задача распознавания объектов на изображениях часто решается с использованием CPU и/или GPU. Однако в автономных системах такие решения не удовлетворяют требованиям по энергоэффективности в отличие от реализации на ПЛИС (FPGA). В задаче распознавания связка ПЛИС и нейросетевых алгоритмов показывают хорошие значения производительности и точности. Однако при решении задачи детектирования значения производительности ухудшаются. В силу ограниченности ресурсов ПЛИС необходимы исследования в области оптимизации имеющихся алгоритмов и поиска новых, решающих данную проблему.

Сравнение реализаций СНС

Ранее была разработана аппаратная нейронная сеть для реализации компьютерного зрения на ПЛИС [1]. Эффективность данной реализации необходимо сравнить с существующими решениями, например, программными реализациями на CPU и GPU. В сравнении использовалась архитектура сверточной нейронной сети (СНС) архитектуры подобной *LeNet5*, которая применяется для задачи распознавания рукописных цифр с базы *MNIST*.

Результаты исследования производительности представлены в табл. 1, где *Тоб* – среднее время, за которое происходит распознавание одного тестового изображения из базы *MNIST*, начиная с его загрузки в память устройства и заканчивая получением результата распознавания, а *Тпп* – среднее время работы СНС по распознаванию одной тестовой цифры.

Таблица 1. Результаты производительности

Тестируемые устройства	Тоб, мс	Тпп, мс
AMD Phenom II 925 (32 бита)	3,042	2,980
AMD Phenom II 925 (16 бит)	21,331	21,264
ARM Cortex A9 (32 бита)	11,489	11,200
ARM Cortex A9 (16 бит)	99,144	97,296
Nvidia GTX 1060 GTX (32 бита)	0,822	0,774
Nvidia GTX 1060 GTX (16 бит)	1,704	1,652
Altera Cyclone V (16 бит)	3,262	2,417

Видим, что производительность FPGA реализации показывает значения ниже чем у варианты GPU и близка к производительности CPU. Аппаратная реализация СПС показывает высокие результаты за счёт параллельной работы вычислительных блоков.

Полученные результаты показывают время распознавания только одного объекта. Если анализировать всё изображение, то необходимо использовать различные методики. Самый простой это метод скользящего окна с помощью которого возможно осуществить детектирование объекта и его дальнейшее распознавание. Например, из полученных результатов можно сказать, что время обработки изображения 400x400 с шагом 4 будет ровняться примерно 30 секунд. Что является не эффективным решением. Однако если использовать данный подход, то анализ всего изображения с искомыми объектами займет неудовлетворительно большое количество времени. Откуда вытекает задача исследования современных архитектур для детектирования и распознавания объектов на изображении.

Современные сети для детектирования и распознавания объектов на изображении

В статье [2], Росс Гиршик (Ross Girshick) с коллегами описывает систему для детектирования объектов и семантической сегментации, называемую *R-CNN*. Эта система состоит из нескольких частей, каждая из которых выполняет часть работы по детектированию и сегментации.

Из входного изображения выделяется 2000 возможных областей *(region proposals)* с использованием какого-либо алгоритма, например, селективный поиск *(Selective Search), objectness* и др. Каждая область масштабируется под входные размеры сверточной нейронной сети (СНС) и распознается с помощью этой сети. Затем, для каждой области используется специальная линейная машина опорных векторов *(SVM)* которая решает задачу регрессии, заменяя классификатор *softmax*. Средняя точность *(mean Average Precision)* распознавания, полученная для Pascal VOC 2010 составляет 53,7%.

Затем тем же автором представляется *Fast R-CNN* [3]. Эта сеть работает в 9 раз быстрее *R-CNN*. *R-CNN* достаточно медленная, так как необходимо применять СНС для каждой возможной области. Вместо этого одна СНС применяется для всего изображения. Всё изображение подаются на вход СНС, которая генерирует карту признаков *(feature map)*. Затем, предполагаемые области *(object proposals)* накладываются на полученную карту признаков для каждой предполагаемой области с помощью слоя пулинга *(pooling layer)* извлекается вектор признаков *(RoI feature vector)*. Каждый вектор признаков подается на вход нескольких

полносвязных слоёв, в результате получается два выхода, один содержит вероятности принадлежности к классам, а другой координаты ограничивающей области для объекта. mAP (mean Average Precision) с использованием этого классификатора на выборке Pascal VOC 2007 составляет 66,9%.

Росс Гиршик с коллегами в статье [4] представил сеть *Faster R-CNN*. Она состоит из двух частей. Для поиска предполагаемых областей предлагается использовать сети региональных предположений (*Region Proposal Networks – RPNs*), которая использует якорные значения (anchor boxes) для определения предполагаемых областей. А вторая часть это *Fast R-CNN* детектор, использующий полученные предполагаемые области. Эту сеть можно обучать как один компонент в отличие от предшественников. С использованием *RPN* для поиска предполагаемых областей и *VGG-16* в качестве детектора mAP для этой сети на наборе Pascal VOC 2007 составляет 73.2%.

Джозефа Редмона (Joseph Redmon) в статье [5] предложил сеть *YOLO*. Эта СНС из изображения одновременно извлекает ограничивающие области и вероятности принадлежности объекта к определенному классу. Такая модель имеет ряд преимуществ. Данный подход имеет крайне высокую скорость работы. Сеть использует всё изображение для детектирования поэтому ошибка срабатывания детектора на фоне изображения у *YOLO* меньше в сравнении с *R-CNN* подходом. В сравнении с *Fast R-CNN* и *Faster R-CNN*, для которых скорость работы составляет 0,5 и 7 кадров в секунду соответственно, *YOLO* обрабатывает 45 кадров. Однако, точность работы отстает от этих детекторов и составляет 63,4% mAP на выборке Pascal VOC 2007.

В [6] описывается сеть *SSD*. Для детектирования тоже, как и в подходе *YOLO* используется только одна сеть. Но она имеет иную топологию. Архитектура сети представлена в виде *VGG-16* сети в качестве детектора, над котором надстроены сверточные слои разных размеров фильтров, каждый из которых предугадывает различные параметры. Для предугадывания размеров объектов также используются *anchor boxes.* В качестве результатов, приводимых авторами, точность mAP составляет 74,3% на выборке Pascal VOC 2007. А скорость работы составляет 59 кадров в секунду на GPU Nvidia Titan X.

В статье [7] представлена улученная версия *YOLO*. Ряд нововведений позволил повысить точность и скорость работы сети. Так точность mAP теперь составляет 76,8% на выборке Pascal VOC 2007. Скорость работы при такой точности достигает 67 кадров в секунду. Одно из нововведений это батч-нормализация (batch normalization). Другое нововведение — это переход к полностью сверточной сети (*fully convolution network).* Из *YOLO* удалены два полносвязных слоя на выходе, вместо этого, на выходе имеем результаты сверточных слоев. Вместо координат, сеть предсказывает смещения относительно якорных значений. Для *YOLO* эти anchor boxes получаются с использованием метода *K*-means с использованием обучающей выборки.

Заключение

На основе полученной реализации СНС на ПЛИС для распознавания можно сказать, что эта идея является перспективной. Однако в настоящее время актуально решение задачи распознавания вместе с задачей детектирования объектов. Для этого были предложены архитектуры СНС, решающие данные задачи с высокой точностью и малыми вычислительными затратами. В данной работе были рассмотрены современные архитектуры, включая YOLO и SSD которые выглядят наиболее перспективными.

Список использованных источников

1. Зоев И. В. Разработка аппаратной нейросети для реализации компьютерного зрения на ПЛИС // МСИТ: сборник трудов XIV Международной научно-практической конференции студентов, аспирантов и молодых ученых, г. Томск, 7-11 ноября 2016 г. : в 2 т. — Томск : Изд-во ТПУ, 2016. — Т. 1. — [С. 45-46].
2. Rich feature hierarchies for accurate object detection and semantic segmentation. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik [Электронный ресурс] – URL: https://arxiv.org/pdf/1311.2524.pdf (дата обращения 18.09.2017)
3. Ross Girshick. Fast R-CNN [Электронный ресурс] – URL: https://arxiv.org/pdf/1504.08083.pdf (дата обращения 18.09.2017)
4. Ross Girshick. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [Электронный ресурс] – URL: https://arxiv.org/pdf/1506.01497.pdf (дата обращения 18.09.2017)
5. Joseph Redmon. You Only Look Once: Unified, Real-Time Object Detection [Электронный ресурс] – URL: https://arxiv.org/pdf/1506.02640.pdf (дата обращения 18.09.2017)
6. Wei Liu. SSD: Single Shot MultiBox Detector [Электронный ресурс] – URL: https://arxiv.org/pdf/1512.02325.pdf (дата обращения 18.09.2017)
7. Joseph Redmon. YOLO9000: Better, Faster, Stronger [Электронный ресурс] – URL: https://arxiv.org/pdf/1612.08242.pdf (дата обращения 18.09.2017)