Московский государственный технический университет им. Н.Э.

Баумана

Кафедра «Системы обработки информации и управления»

Домашнее Задание

по дисциплине

«Методы машинного обучения»

Выполнил:

студент группы ИУ5И-21М

Ли Лююй

Москва — 2024 г.

Задание

Домашнее задание по дисциплине направлено на анализ современных методов машинного обучения и их применение для решения практических задач. Домашнее задание включает три основных этапа:

- выбор задачи;
- теоретический этап;
- практический этап.

Этап выбора задачи предполагает анализ ресурса paperswithcode. Данный ресурс включает описание нескольких тысяч современных задач в области машинного обучения. Каждое описание задачи содержит ссылки на наиболее современные и актуальные научные статьи, предназначенные для решения задачи (список статей регулярно обновляется авторами ресурса). Каждое описание статьи содержит ссылку на репозиторий с открытым исходным кодом, реализующим представленные в статье эксперименты. На этапе выбора задачи обучающийся выбирает одну из задач машинного обучения, описание которой содержит ссылки на статьи и репозитории с исходным кодом. Теоретический этап включает проработку как минимум двух статей, относящихся к выбранной задаче. Результаты проработки обучающийся излагает в

теоретической части отчета по домашнему заданию, которая может включать:

- описание общих подходов к решению задачи;

конкретные топологии нейронных сетей, нейросетевых ансамблей или других моделей машинного обучения, предназначенных для решения задачи;

- математическое описание, алгоритмы функционирования, особенности обучения используемых для решения задачи нейронных сетей, нейросетевых ансамблей или других моделей машинного обучения;

- описание наборов данных, используемых для обучения моделей;

- оценка качества решения задачи, описание метрик качества и их значений;

- предложения обучающегося по улучшению качества решения задачи. Практический этап включает повторение экспериментов авторов статей на основе представленных авторами репозиториев с исходным кодом и возможное улучшение обучающимися полученных результатов. Результаты проработки обучающийся

излагает в практической части отчета по домашнему заданию, которая может включать:

- исходные коды программ, представленные авторами статей, результаты документирования программ обучающимися с использованием диаграмм UML, путем визуализации топологий нейронных сетей и другими способами;

- результаты выполнения программ, вычисление значений для описанных в статьях метрик качества, выводы обучающегося о воспроизводимости экспериментов авторов статей и соответствии практических экспериментов теоретическим материалам статей;

- предложения обучающегося по возможным улучшениям решения задачи, результаты практических экспериментов (исходные коды, документация) по возможному улучшению решения задачи.

Выбранная задача: «Семантическая сегментация»

1. Выбор задачи

Семантическая сегментация, или сегментация изображения, - это задача объединения в кластеры частей изображения, принадлежащих к одному классу объектов. Это форма прогнозирования на уровне

пикселей, поскольку каждый пиксель в изображении классифицируется в соответствии с категорией.

2. Теоретический этап

Часть I

Тема《Полностью конволюционные сети для семантической сегментации》

Конволюционные сети способствуют прогрессу в распознавании. Конволютные сети не только улучшают классификацию всего изображения, но и добиваются успехов в решении локальных задач со структурированным выходом. К ним относятся успехи в обнаружении объектов в ограниченной области, предсказание частей и ключевых точеки локальное соответствие.

Естественным следующим шагом в продвижении от грубого к естественным следующим шагом в переходе от грубого к тонкому прогнозированию является предсказание по каждому пикселю. В предыдущих подходах для семантической сегментации использовались конвентные сети, в которых каждый пиксель помечался классом окружающего его объекта или области, но с недостатками, которые устраняются в данной работе.

Мы показываем, что полностью конволюционная сеть (FCN), обученная из конца в конец, от пикселя к пикселю на семантической сегментации, превосходит современную без дополнительных механизмов. Насколько нам известно, это первая работа по обучению FCN из конца в конец (1) для предсказания по пикселям и (2) из предварительного обучения под наблюдением. Полностью конволюционные версии существующих сети предсказывают плотные выходы из входов произвольного размера. Как обучение, так и выводы выполняются по всему изображению одновременно с помощью плотного прямого вычисления и обратного распространения. Внутрисетевые слои апсемплинга позволяют предсказывать и обучаться попиксельно в сетях с субдискретизацией.

Этот метод эффективен, как асимптотически, так и абсолютно, и исключает необходимость в усложнениях, которые встречаются в других работах. работах. Обучение по участкам широко распространено, но не обладает эффективностью полностью конволюционного обучения. Наш подход не использует усложнения до и после обработки, включая суперпиксели, предложения, или пост-специальное уточнение с помощью случайных полей или локальных классификаторов. Наша модель переносит недавний успех в классификациина плотное предсказание, переосмысливая классификационные сети как полностью конволюционные и

выполняя тонкую настройку из их выученных представлений. В отличие от этого, предыдущие работы применяли небольшие сверточные сети без контролируемого предварительного обучения.

Семантическая сегментация сталкивается с внутренним напряжением между семантикой и местоположением: глобальная информация определяет, что, а локальная - где. Глубокие иерархии признаков совместно кодируют местоположение и семантику в локально-глобальной пирамиде.

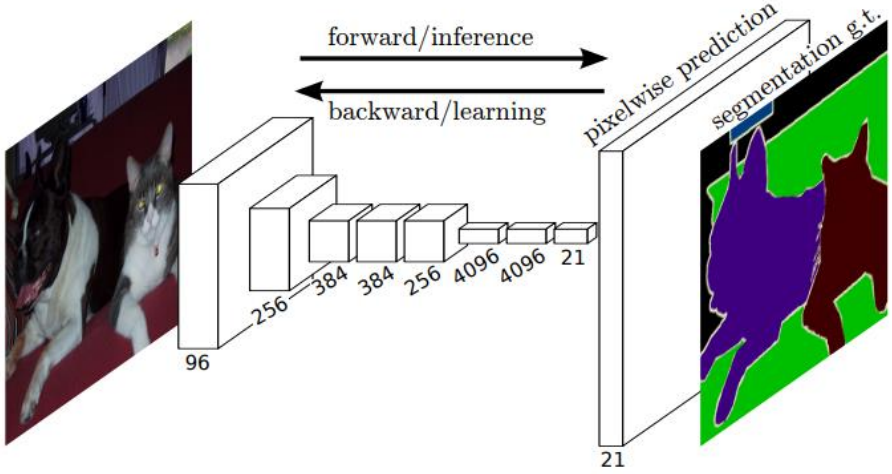

Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

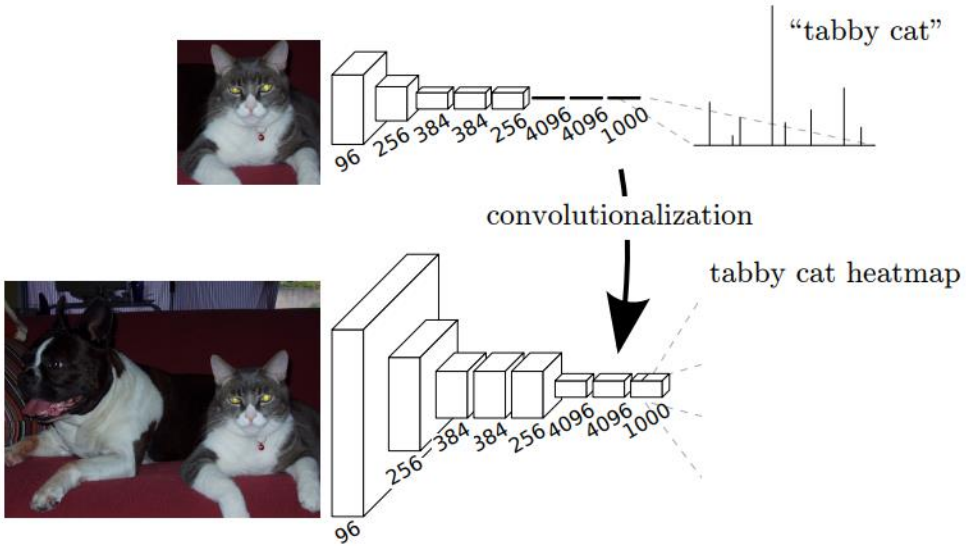

Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

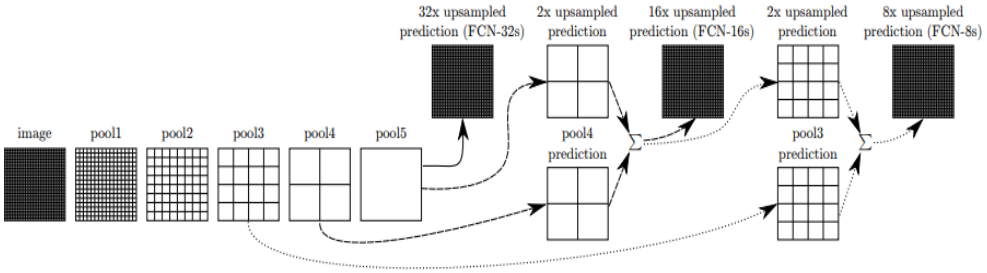

Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Layers are shown as grids that reveal relative spatial coarseness. Only pooling and prediction layers are shown; intermediate convolution layers (including our converted fully connected layers) are omitted. Solid line (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Dashed line (FCN-16s): Combining predictions from both the final layer and the `pool4` layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Dotted line (FCN-8s): Additional predictions from `pool3`, at stride 8, provide further precision.

Полностью конволюционные сети

Каждый слой данных в конвсети представляет собой трехмерный массив размером h × w × d, где h и w - пространственные размеры, а d - размерность признака или канала. Первый Первый слой - это изображение с размером пикселя h × w и d цветовыми каналами. Места в более высоких слоях соответствуют местам в изображении, с которыми они связаны, и которые называются их рецептивными полями.

Конвентные сети построены на основе инвариантности перевода. Их основные компоненты (функции свертки, объединения и активации) работают на локальных входных областях и зависят только от относительных пространственных координат. Записывая xij для вектора данных в месте (i, j) в определенном слое, и yij для следующего слоя, эти функции вычисляют выходы yij по:

$$\mathbf{y}_{ij} = f_{ks}\left(\{\mathbf{x}_{si+\delta i, sj+\delta j}\}_{0 \le \delta i, \delta j \le k}\right)$$

где k - размер ядра, s - коэффициент страйда или поддискретизации, а fks определяет тип слоя: матричное умножение для свертки или среднего объединения, пространственный max для объединения max, или элементарная нелинейность для функции активации, и т.д. для функции активации. активационная функция, и так далее для других типов слоев.

Эта функциональная форма сохраняется при композиции, с размером ядра и страйдом, подчиняющимся правилу преобразования

$$f_{ks} \circ g_{k's'} = (f \circ g)_{k'+(k-1)s', ss'}.$$

В то время как общая глубокая сеть вычисляет общую нелинейную функцию, сеть с единственными слоями такой формы вычисляет нелинейный фильтр, который мы называем глубоким фильтром или полностью сверточной сетью. FCN естественно работает на входе любого размера и производит выход соответствующего (возможно пространственных размеров.

A real-valued loss function composed with an FCN defines a task. If the loss function is a sum over the spatial dimensions of the final layer,

$\ell(\mathbf{x};\theta) = \sum_{ij} \ell'(\mathbf{x}_{ij};\theta)$, its gradient will be a sum over the gradients of each of its spatial components. Thus stochastic gradient descent on ` computed on whole images will be the same as stochastic gradient descent on , ℓ' taking all of the final layer receptive fields as a minibatch.

Когда эти рецептивные поля значительно перекрываются, как вычисления с прямой передачей, так и обратное распространение намного эффективнее, если вычислять послойно по всему изображению, а не независимо участок за участком.

Часть II

Тема 《 ENet: Архитектура глубокой нейронной сети для Семантическая сегментация в реальном времени》

Недавний интерес к носимым устройствам с дополненной реальностью, устройствам домашней автоматизации и самодвижущимся автомобилям вызвал острую потребность в алгоритмах семантической сегментации (или визуального понимания

сцены). которые могут работать в режиме реального времени на мобильных устройствах с низким энергопотреблением. Эти алгоритмы маркируют каждый пиксель на изображении с одним из классов объектов. В последние годы доступность больших наборов данных и мощные вычислительные машины помогли глубоким сверточным нейронным сетям (CNN) превзойти производительность многих обычных алгоритмов компьютерного зрения. Даже Несмотря на то, что CNN все успешнее справляются с задачами классификации и категоризации, они дают грубые пространственные результаты при применении к попиксельной маркировке изображений. Поэтому их часто объединяют в каскад с другими алгоритмами для уточнения результатов, такими как сегментация на основе цветаили условные случайные поля, чтобы назвать другие алгоритмы. поля, и др.

Для пространственной классификации и тонкой сегментации изображений было предложено несколько архитектур нейронных сетей, таких как SegNet или полностью сверточные сети. были предложены такие архитектуры, как SegNet или полностью конволюционные сети. Все эти работы основаны на архитектуре VGG16, которая является очень большой моделью, предназначенной для многоклассовой классификации. В этих работах предлагаются сети с огромным количеством параметров и длительным временем вычисления. время вывода. В таких условиях они становятся

непригодными для многих мобильных или работающих от аккумуляторов приложений, которые требуют обработки изображений со скоростью более 10 кадров в секунду.

В данной работе мы предлагаем новую архитектуру нейронной сети, оптимизированную для быстрого вывода и высокой точности. Примеры изображений, сегментированных с помощью ENet, показаны на рисунке 1. В нашей работе мы решили не использовать никаких шагов постобработки, которые, конечно, могут быть объединены с нашим методом, но это может ухудшат производительность сквозного подхода CNN.

Архитектура сети

Table 1: ENet architecture. Output sizes are given for an example input of 512×512.

Name	Type	Output size
initial		$16 \times 256 \times 256$
bottleneck1.0	downsampling	$64 \times 128 \times 128$
$4\times$ bottleneck1.x		$64 \times 128 \times 128$
bottleneck2.0	downsampling	$128 \times 64 \times 64$
bottleneck2.1		$128 \times 64 \times 64$
bottleneck2.2	dilated 2	$128 \times 64 \times 64$
bottleneck2.3	asymmetric 5	$128 \times 64 \times 64$
bottleneck2.4	dilated 4	$128 \times 64 \times 64$
bottleneck2.5		$128 \times 64 \times 64$
bottleneck2.6	dilated 8	$128 \times 64 \times 64$
bottleneck2.7	asymmetric 5	$128 \times 64 \times 64$
bottleneck2.8	dilated 16	$128 \times 64 \times 64$
Repeat section 2, without bottleneck2.0		
bottleneck4.0	upsampling	$64 \times 128 \times 128$
bottleneck4.1		$64 \times 128 \times 128$
bottleneck4.2		$64 \times 128 \times 128$
bottleneck5.0	upsampling	$16 \times 256 \times 256$
bottleneck5.1		$16 \times 256 \times 256$
fullconv		$C \times 512 \times 512$

Архитектура нашей сети представлена в таблице 1. Она разделена на несколько этапов, выделенных горизонтальными линиями в таблице и первой цифрой после названия каждого блока. Размеры выходных данных приведены для входного изображения с разрешением 512 × 512. Мы придерживаемся взгляда на ResNets, который описывает Они имеют одну главную ветвь и расширения с конволюционными фильтрами, которые отделяются от нее, а затем объединяются обратно с помощью поэлементного сложения, как

показано на рисунке 2b. Каждый блок состоит из трех сверточных слоев: проекции 1 × 1, которая уменьшает размерность, основного сверточного слоя (conv на рисунке 2b) и расширения 1 × 1. Мы помещаем пакетную нормализациюи PReLU между всеми свертками. Как и в оригинальной статье, мы называем их "узкими местами". модули. Если узким местом является понижающая выборка, то к основной ветви добавляется слой максимального пула

Кроме того, первая проекция 1 × 1 заменяется на сверткой 2 × 2 с шагом 2 в обоих измерениях. Мы обнуляем активации, чтобы соответствовать количество карт признаков. Свертка - это либо обычная, расширенная или полная свертка (также известная как как деконволюция или дробная свертка) с фильтрами 3 × 3. Иногда мы заменяем ее асимметричной сверткой, т.е. последовательностью из 5 × 1 и 1 × 5 сверток. В качестве регуляризатора мы используем Spatial Dropout, с p = 0.01 до узкого места2.0 и p = 0.1 после него.

Начальный этап содержит один блок, который представлен на рисунке 2a. Стадия 1 состоит из 5 узких блоков, в то время как стадии 2 и 3 имеют такую же структуру, за исключением того, что этап 3 не понижает дискретизацию входного сигнала в начале (мы опускаем 0-е узкое место). Эти три первых являются кодером. Этапы 4 и 5 относятся к декодеру.

Мы не использовали условия смещения ни в одной из проекций, чтобы сократить количество вызовов ядра и операций с памятью. вызовов ядра и общих операций с памятью, поскольку cuDNN использует отдельные ядра для свертки и добавления смещения. Этот выбор не оказал никакого влияния на точность. Между каждым сверточным слоем и последующей нелинейностью мы используем Пакетную нормализацию. В декодере макс. pooling заменяется на max unpooling, a padding заменяется на пространственную свертку без смещения. Мы не использовали индексы пулинга в последнем модуле апсемплинга, поскольку исходный блок оперировал на 3 каналах входного кадра, в то время как конечный выход имеет C карт признаков (количество классов объектов). классов). Также, по соображениям производительности, мы решили поместить в качестве последнего модуля сети только "голую" полную свертку. Также из соображений производительности мы решили поместить в качестве последнего модуля сети только голую полную свертку, которая сама по себе занимает значительную часть времени обработки декодера.

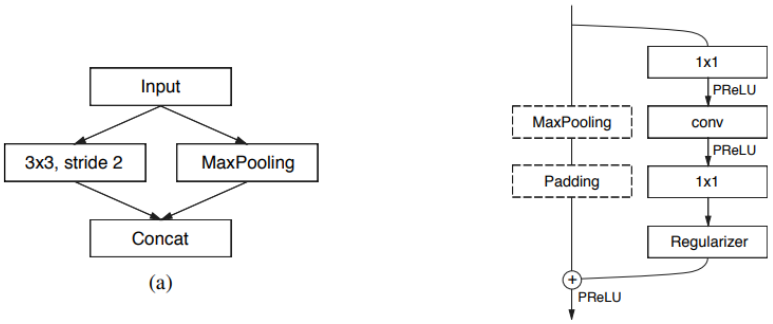

Figure 2: (a) ENet initial block. MaxPooling is performed with non-overlapping 2×2 windows, and the convolution has 13 filters, which sums up to 16 feature maps after concatenation. This is heavily inspired by [28]. (b) ENet bottleneck module. `conv` is either a regular, dilated, or full convolution (also known as deconvolution) with 3×3 filters, or a 5×5 convolution decomposed into two asymmetric ones.

3. Практическая часть

Практическая часть выложена в Gitlab.

4 Заключение

Мы предложили новую архитектуру нейронной сети, разработанную с нуля специально для семантической сегментации. Наша главная цель - эффективно использовать ограниченные ресурсы, доступные на встроенных платформах, по сравнению с полноценными рабочими станциями глубокого обучения. Наша работа обеспечивает значительный значительный выигрыш в решении этой задачи, при этом соответствуя, а иногда и превосходя существующие базовые модели, которые имеют на порядок более высокие требования к вычислениям и памяти. Применение ENet на

NVIDIA TX1 является примером портативных встраиваемых решений реального времени.

Несмотря на то, что основной целью была работа сети на мобильных устройствах, мы обнаружили, что она также очень эффективна на высокопроизводительных графических процессорах, таких как NVIDIA Titan X. Это может оказаться полезным в центрах обработки данных приложениях, где требуется обработка большого количества изображений высокого разрешения. ENet позволяет выполнять крупномасштабные вычисления гораздо быстрее и эффективнее, что может привести к значительной экономии.

5 Список использованных источников

[1] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” The handbook of brain theory and neural networks, pp. 255–258, 1998.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems 25, 2012, pp. 1097–1105.

[3] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014. 9

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan,

V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[5] J. Shotton, J. Winn, C. Rother, and A. Criminisi, “Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context,” Int. Journal of Computer Vision (IJCV), January 2009.

[6] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, “Large-scale image retrieval with compressed fisher vectors,” in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010, pp. 3384–3391.

[7] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, “Segmentation as selective search for object recognition,” in IEEE International Conference on Computer Vision, 2011.

[8] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning hierarchical features for scene labeling,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1915–1929, Aug 2013.

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected crfs,” arXiv preprint arXiv:1412.7062, 2014.

[10] V. Badrinarayanan, A. Handa, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling,” arXiv preprint arXiv:1505.07293, 2015.