

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное
учреждение высшего образования
ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Институт математики, механики и компьютерных наук
имени И. И. Воровича

Направление подготовки 02.03.02 — "Фундаментальная
информатика и информационные технологии"

Выпускная квалификационная работа
на степень бакалавра

Ивахненко Д.И., 4 курс, 9 группа

СЕМАНТИЧЕСКИЙ АНАЛИЗ ФОТОГРАФИЙ С ПОМОЩЬЮ
ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ

Научный руководитель:
к.т.н, ст.преп. М. В. Юрушкин

Допущено к защите:
Заместитель директора ИММКН
по направлению ФИИТ

_____ В.С. Пилиди

Ростов-на-Дону
2020

Содержание

Введение	3
1. Постановка и описание задачи	4
1.1. Задача сегментации	4
1.2. Выделение неба в рамках задачи сегментации	6
2. Обзор предметной области	8
2.1. Сегментация изображений	8
2.2. Способы регуляризации	9
3. Метод решения	9
3.1. Глубокая сверточная сеть	10
3.2. Регуляризация на уровне архитектуры	15
3.3. Выделение признаков, downsampling	17
3.4. Генерализация признаков, upsampling	18
4. Данные	18
4.1. Датасет SkyFinder	18
4.2. Датасет с синтетической разметкой	19
Заключение	19
Список литературы	19

Введение

В рамках данной работы освещается вопрос семантического анализа изображений путем применения глубоких сверточных нейронных сетей. Под семантическим анализом понимается получение из изображения какой-либо интерпретируемой информации: расположение объектов на сцене, принадлежность объектов к заранее заданным классам, наличие на изображении объектов определённого типа и т.п.. Данная тема будет рассмотрена на примере задачи обнаружения и выделения неба на изображениях. Входными данными задачи являются фотографии, сделанные на камеры мобильных устройств. Специализированный домен изображений был выбран с целью упрощения адаптация потенциального решения к применению в конечных продуктах, таких как пользовательское программное обеспечение для смартфонов. Решением задачи выступают сгенерированные для входных фотографий полутоновые изображения. Такое изображение называется сегментационной маской и для каждого пикселя исходной фотографии выражает его принадлежность к региону неба: белый цвет интерпретируется как положительный результат, черный - как отрицательный. В ходе разработки алгоритма решения задачи была исследована эффективность применения различных подходов к генерации подобного рода масок - сегментации. Сравнения эффективности проходило по индексу Жаккара - в западной литературе также встречается название *intersection over union*, IoU. Для решения задачи сегментации из наиболее эффективных подходов был составлен стек алгоритмов: применение к входному изображению глубокой сверточной сети с последующей корректировкой методами компьютерного зрения полученной маски. В ходе обучения модели искусственной нейронной сети, ИНС, для предотвращения переобучения и улучшения сходимости применялись техники регуляризации, такие как *learning rate reduce* и *one cycle policy*. Влияние данных подходов на решение также отражено в результатах работы.

В заключительной части хода разработки была рассмотрена возможность адаптация модели, обученной на данных датасета SkyFinder, к выбранному домену изображений. Данная техника имеет название Domain Adaptation и используется для улучшения качества на практических данных.

1. Постановка и описание задачи

Цифровая обработка изображений являются комплексной темой. В нее входят задачи фильтрации, преобразования цвета, яркости и контрастности, морфологической обработка, распознавания и выделения объектов на сцене [1]. Семантический анализу - подход, целью которого является получение интерпретируемой информации высших порядков об изображении. К подзадачам семантического анализа можно отнести классификацию объектов на сцене, детекцию объектов и сегментацию изображений на семантические регионы. В рамках текущей работы будет проведен разбор подзадачи сегментации.

1.1. Задача сегментации

Для возможности цифровой обработки и анализа будем рассматривать представление изображения как трёхмерного массива чисел, имеющего ширину, количество столбцов, и высоту, количество строк, равными ширине и высоте изображения соответственно. На каждой позиции по ширине и высоте будет находится вектор из трех целых чисел, из промежутка $[0, 255]$, что соответствует RGB модели представления цвета пикселя изображения. В таком случае сегментацией изображения будет являться отображение каждого RGB вектора в некоторое целое число. Это число соответствует идентификатору некоторого класса. При применение такого отображения ко всему

изображению получается двумерный массив, ширина и высота которого соответствуют таковым у исходного изображения. Подобный двумерный массив будет разделять изображение на регионы по обозначенному признаку и называться сегментационной маской. Сегментацию возможно рассматривать как попиксельную классификацию объектов.

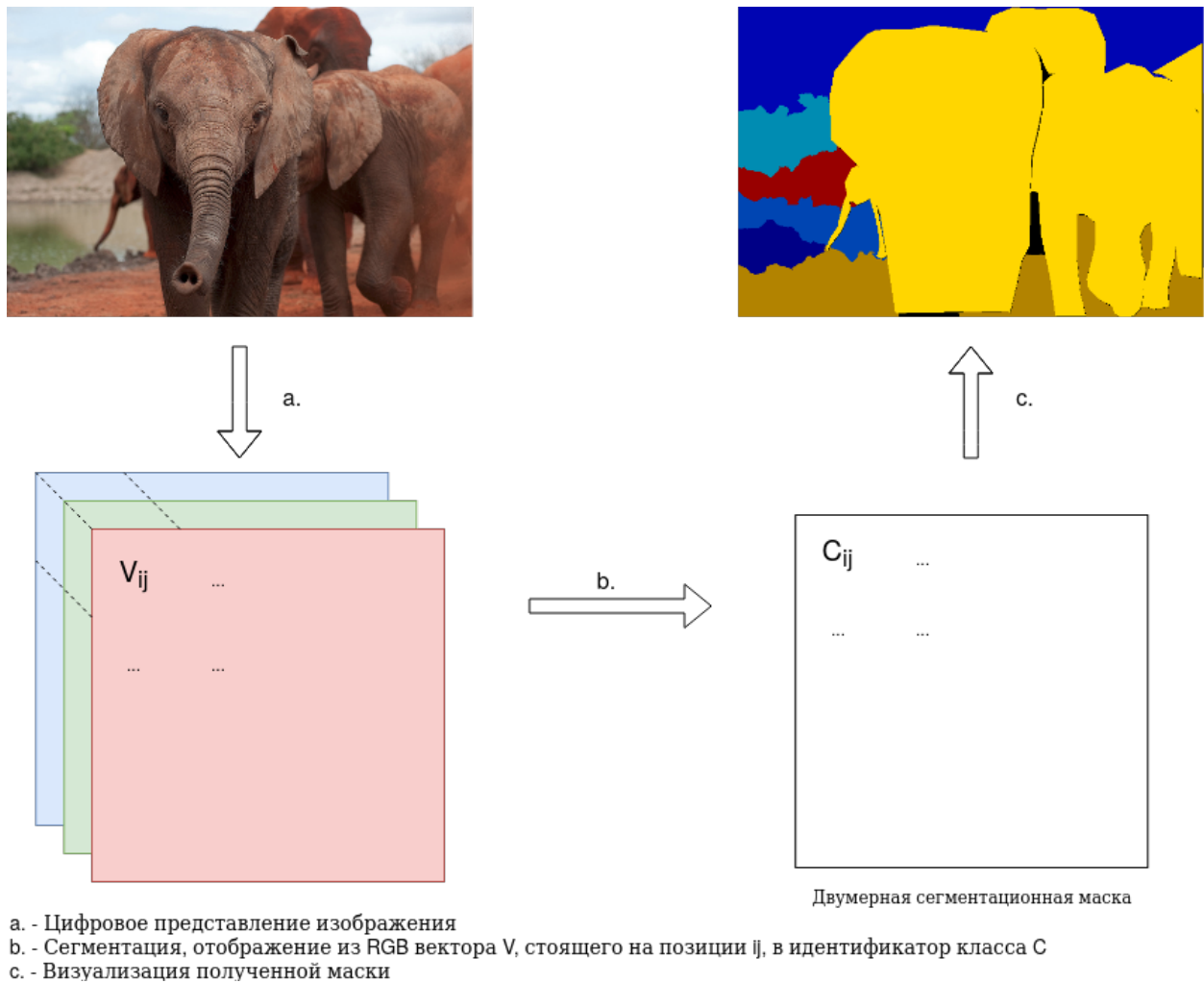


Рисунок 1 — Общий вид сегментации на примере данных датасета COCO

Отметим, что описанное выше отображение может обладать относительно простой природой и учитывать только значение пикселя в конкретной позиции, так и более сложной структурой, исполь-

зующей информацию о распределении цветов во всем изображении, положении пикселя на изображении и свойствах соседних пикселей, непосредственно соседствующих с обозреваемым значением или отступающих от него на заданное смещение [2]. На 1 показан общий вид задачи сегментации изображения.

1.2. Выделение неба в рамках задачи сегментации

Описанная в введение задача выделения региона неба на входном изображении может быть рассмотрена как задача сегментации. Так как в данном случае результирующих классов всегда два - класс принадлежности и обратный ему -, то имеется более узкий случай сегментации - бинарная. Итоговая маска будет содержать только значения 0 и 1, представляя собой однобитовое бинарное изображение, что можно считать вырожденным полутоновым.

Таким образом, решение задачи выделения неба сводится к нахождению отображения из вектора цветов для каждого пикселя в целое число из промежутка $[0, 1]$. Данное отображение возможно получить как алгоритмами машинного зрения, так и с помощью использования моделей глубоких сверточных нейронных сетей. В данной работе приводится решение методом нейронных сетей, при этом алгоритмы компьютерного зрения используются для корректировки полученной маски. Под корректировкой здесь понимается обнаружение и удаление ложноположительных регионов неба небольшой площади.



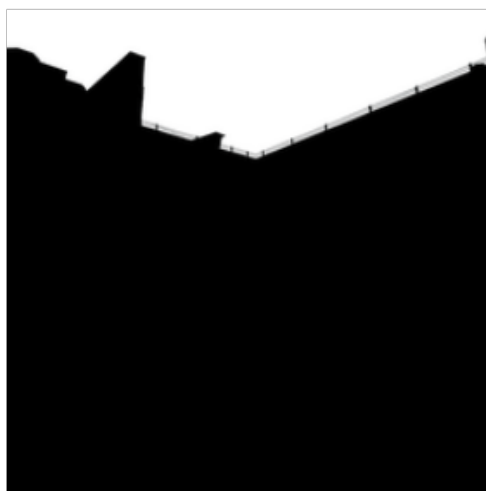
1. Результат работы сети



2. Исходное изображение



3. Коррекция результата



4. Размеченная маска

Рисунок 2 — Пример, демонстрирующий работу алгоритма

Практическое применение подобного решения можно найти в пользовательских приложениях эстетической обработки пейзажных фотографий для смартфонов, в автоматических системах мониторинга воздушного пространства, при извлечении семантической информации высшего порядка для использования в иных методах анализа и обработки изображений [3].

2. Обзор предметной области

Задача семантического анализа изображений имеет широкое применение при решении различных прикладных и исследовательских проблем [4] [5] [6] [7], в связи с чем активно изучается. Сегментации изображений, в частности, нашли применения в таких областях как проведение хирургических операций, автопилотируемый транспорт, автоматизированное картографирование местности [2]. Ниже приведен краткий обзор исследований, темы которых связаны с поставленной задачей. Результаты этих исследований в разной степени использовались для построения решения.

2.1. Сегментация изображений

До начала активного применения глубоких нейронных сетей в задачах сегментации использовались методы компьютерного зрения, основанные на применении порогов бинаризации для полутоновых изображений, выявлении признаков, кластеризации методом k-средних [8] [9] [10]. Каждый из этих подходов имеет свои преимущества, метод бинарной сегментации полутоновых изображений до сих пор успешно применяется в области анализа медицинских данных [11]. Но применение данных подходов к задаче выделения границы между объектами путем сегментации показало худшие результаты в сравнении с FCN, полностью сверточными глубокими сетями [12].

Современные решения задачи сегментации в различных областях зачастую опираются на применение нейронных сетей [13]. Наиболее распространёнными архитектурами являются Unet, DeepLab, RefineNet [14] [15] [16]. Имеются исследования применения архитектуры RefineNet для определения региона неба на датасете SkyFinder [17].

2.2. Способы регуляризации

Помимо специализированных решений задачи сегментации, были рассмотрены также общие методики, применяемые для обучения глубоких нейронных сетей. При высокой сложности модели, в процессе обучения она может начать отражать в ответах шум в тренировочных данных [18] [19]. Данное явление называется переобучением, *overfitting*. С целью снизить вероятность его проявления применяются разнообразные техники регуляризации: *dropout*, нормализация значений между слоями сети, настройка гиперпараметров процесса обучения [20] [21] [22].

3. Метод решения

Конечное решение задачи сегментации имеет комплексное строение. Изображение, поступающее на вход решающему алгоритму, обрабатывается ИНС. Результатом такого применения является двумерный массив пар. Каждое значение в паре обозначает вероятность принадлежности одному из двух классов. Для получения сегментационной маски индекс максимального из двух значений рассчитывается как идентификатор класса: 1 для региона неба, 0 - обратный ему. В ходе экспериментов было выявлено наличие артефактов в результатах классификации. Для части изображений имелись ложноположительные регионы неба небольшой площади. Для их устранения последовательно применялись алгоритмы *FindCounters* и *DrawCounters* из состава библиотеки *OpenCV*.

Для обучения сети использовался датасет *SkyFinder*. Для улучшения показателя *IoU* на целевом домене изображений использовалась техника *Unsupervised Domain Adaptation*. Это позволило уменьшить количество ложноположительных и ложноотрицательных регионов без дополнительной разметки обучающей выборки из фотографий, сделанных на мобильные устройства.

3.1. Глубокая сверточная сеть

Как было отмечено выше, текущие исследования указывают на преимущества глубоких сверточных сетей в задачах семантического анализа перед классическими алгоритмами компьютерного зрения. Глубокими сверточными сетями называют подмножество искусственных нейронных сетей, все полносвязные слои которых заменены сверточными.

Архитектура сети для решения задачи сегментации представляется двумя частями. Отбор признаков, *downsampling path* или энкодер, и генерализация результатов отбора, *upsampling path* или декодер. Для решения проблемы потери информации в сверточных слоях, карты признаков с некоторых уровней нисходящей части объединяют с противоположными входами восходящей части. На 3 приведена общая схема сети для задач сегментации.

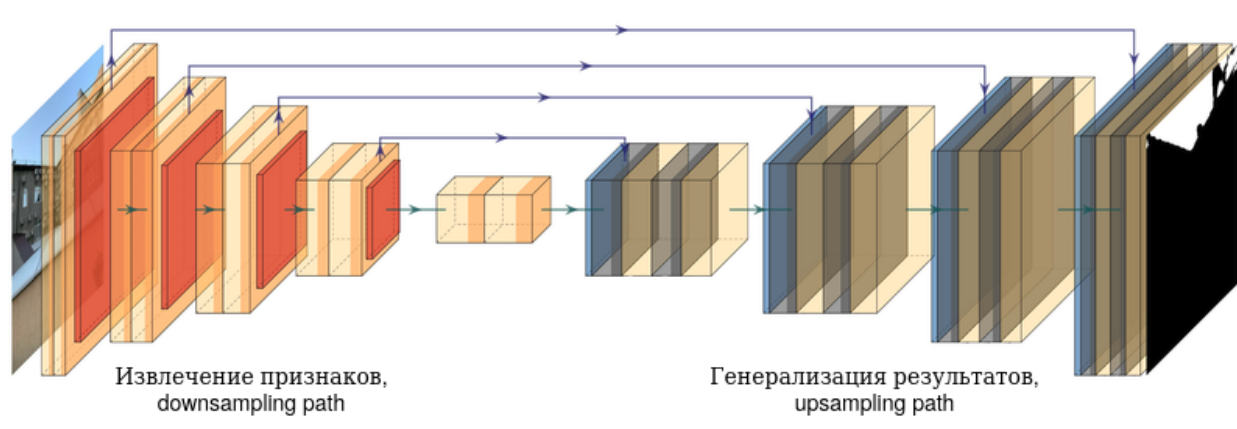


Рисунок 3 — Общий вид глубокой сверточной сети для задачи сегментации

В нисходящей части используются сверточные слои (отмечены желтым цветом на 3), слои с выбором наибольшего значения по ядру - *max pooling* (отмечены красным цветом на 3). На сверточных слоях к входным данным применяются ядра свертки. Количество ядер определяет для слоя количество выходных карт признаков. Выходные кар-

ты признаков подаются на вход следующему слою. К выходным картам применяется функция активации (отмечены оранжевым цветом на 3). Значения в ядрах свертки являются обучаемыми параметрами сети. Ниже приведена формула для простого случая двумерного входного массива и двумерного ядра свертки (1) при единичном шаге.

$$G_{m,n} = (f * h)[m, n] = \sum_j \sum_k h_{j,k} f_{m+j, n+k} \quad (1)$$

где f - входной массив данных, h - ядро свертки, G - выходная карта признаков.

Размеры каждой карты определяются размерами входных данных, размерами ядра, размером добавочного отступа по краям массива данных и сдвигом, на которое ядро смещается по массиву (2).

$$output = \frac{input + 2 * p - k}{s} + 1 \quad (2)$$

где $output$ - размеры выходной карты признаков, $input$ - размеры входного массива данных, p - величина отступа, добавляемого к краям массива ($padding$), k - размеры ядра свертки, s - сдвиг ядра.

В слоях с выбором наибольшего значения ядро определяет только размеры окна, в котором выбирается максимальный элемент (3). Операции сложения и умножения на этом слое не используются.

$$G_{i,j} = \max_{(k,l) \in N} F_{i+k, j+l} \quad (3)$$

где

$$N = \{0, \dots, \text{kernel size}\} \quad (4)$$

G - выходная карта признаков, F - входной массив данных.

В качестве функции активации используется rectified linear unit, ReLU (5).

$$g(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases} \quad (5)$$

В upsampling части сети используются, помимо сверточных, транспонированные сверточные слои (отмечены синим цветом на 3). Прежде чем описать оператор транспонированной свертки, необходимо рассмотреть применение сверточного ядра к входным данным, как произведение матриц. Рассмотрим на примере ядра свертки размером 3×3 и входного массива размером 4×4 . Значение сдвига равно 1, дополнение отступом не используется.

$$(6) \quad \begin{pmatrix} K_{0,0} & K_{0,1} & K_{0,2} & 0 & K_{1,0} & K_{1,1} & K_{1,2} & 0 & K_{2,0} & K_{2,1} & K_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & K_{0,0} & K_{0,1} & K_{0,2} & 0 & K_{1,0} & K_{1,1} & K_{1,2} & 0 & K_{2,0} & K_{2,1} & K_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_{0,0} & K_{0,1} & K_{0,2} & 0 & K_{1,0} & K_{1,1} & K_{1,2} & 0 & K_{2,0} & K_{2,1} & K_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & K_{0,0} & K_{0,1} & K_{0,2} & 0 & K_{1,0} & K_{1,1} & K_{1,2} & 0 & K_{2,0} & K_{2,1} & K_{2,2} \end{pmatrix}$$

$$(7) \quad (W_{0,0} \ W_{0,1} \ W_{0,2} \ W_{0,3} \ W_{1,0} \ W_{1,1} \ W_{1,2} \ W_{1,3} \ W_{2,0} \ W_{2,1} \ W_{2,2} \ W_{2,3} \ W_{3,0} \ W_{3,1} \ W_{3,2} \ W_{3,3})$$

В (6) введена матрица K, ненулевые элементы которой представлены значениями из ядра свертки. $K_{i,j}$ - элемент, стоящий в ядре на позиции i, j. В (7) введена матрица I, полученная путем перекомпоновки входного массива данных в вектор по строкам. Тогда карту при-

знаков G размером 2×2 можно получить путем умножения матрицы K на I^T .

$$K * I^T = \begin{pmatrix} \sum_j \sum_k h_{j,k} f_{j,k} \\ \sum_j \sum_k h_{j,k} f_{j,1+k} \\ \sum_j \sum_k h_{j,k} f_{1+j,k} \\ \sum_j \sum_k h_{j,k} f_{1+j,1+k} \end{pmatrix} \quad (8)$$

Чтобы из (8) получить (1) необходимо записать матричное произведение, избавиться от нулевых слагаемых и объединить оставшиеся под знаками сумм.

$$K * I^T = \begin{pmatrix} G_{0,0} \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \end{pmatrix} \quad (9)$$

Введем транспонированную свертку для входных данных F размером 2×2 , ядра свертки H 3×3 и выходной карты признаков G 4×4 ,

как произведение K^T , полученной из Н аналогично (6), на F_0^T , полученный из F аналогично (7).

$$K^T * F_0^T = \begin{pmatrix} G_{0,0} \\ G_{0,1} \\ G_{0,2} \\ G_{0,3} \\ G_{1,0} \\ G_{1,1} \\ G_{1,2} \\ G_{1,3} \\ G_{2,0} \\ G_{2,1} \\ G_{2,2} \\ G_{2,3} \\ G_{3,0} \\ G_{3,1} \\ G_{3,2} \\ G_{3,3} \end{pmatrix} \quad (10)$$

Таким образом (9) позволяет восстанавливать исходное количество пикселей для их классификации по выделенным признакам [23]. Отметим здесь, что сверточные слои в восходящей части не уменьшают размеры данных по высоте и ширине за счёт использования паддинга.

После выходного слоя размеры массива данных равняются $N \times M \times C$, где N, M - размеры исходного изображения, а C - количество классов. К полученным данным применяется логистическая функция, обобщённая на многомерный случай, softmax (10). Вычисления осуществляются для каждой позиции $[n, m]$ по размерности C .

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad (11)$$

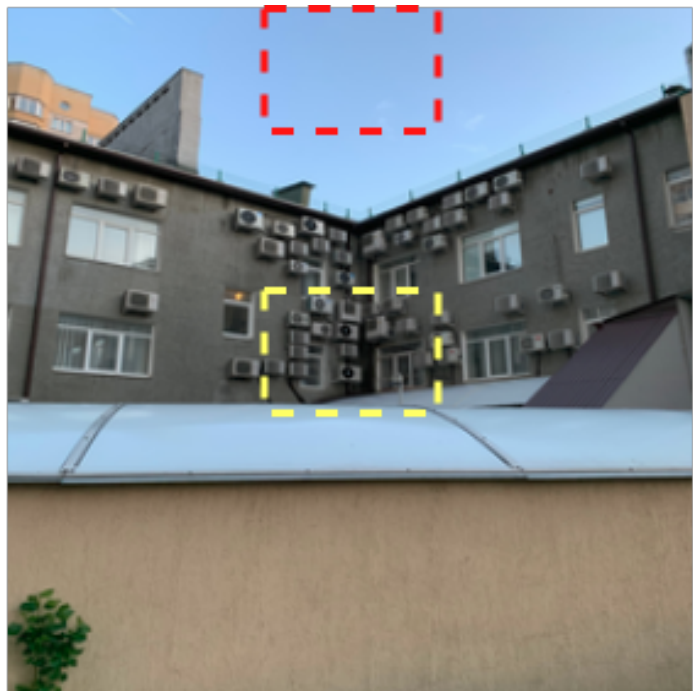
В результате каждый вектор \bar{v} из массива данных по размерности C обладает следующими свойствами:

- $v_i \in [0, 1] \forall i \in [0, C]$
- $\sum_{i=0}^C v_i = 1$

Ниже приведен (4) отладочный вывод данных до и после softmax для программы, реализующей обучения глубокой сверточной сети с ResNet34 блоком в качестве downsampling части и LinkNet в качестве upsampling.

Красным отмечен регион, из которого случайным образом брался пример 1.

Желтым - пример 2.



```
Pre-softmax state; Size: torch.Size([2, 256, 256]),
1. tensor([-9.2658939, 10.0882759], dtype=torch.float64),
2. tensor([ 4.1472898, -4.4603243], dtype=torch.float64)

Post-softmax state; Size: torch.Size([2, 256, 256]),
1. tensor([ 0.0000000, 1.0000000], dtype=torch.float64),
2. tensor([ 0.9998173, 0.0001827], dtype=torch.float64)
```

Рисунок 4 — Отладочная информация с визуализацией

3.2. Регуляризация на уровне архитектуры

Как было отмечено в части обзора решений, при обучении сети может возникнуть overfitting. Состояние модели, в котором она демонстрирует значительно более высокие метрики качества на тренировочных данных, чем на тестовых. Причиной такого поведения может быть запоминание шумов данных. К ним могут быть отнесены статистические выбросы, неравномерное представление классов в выборке, фактическая природа данных. В частности, при преобладании определённого признака в данных вероятно проявление ковариантного сдвига по признаку (Covariate Shift) [24]. Подобное смещение приводит к смещению на уровне внутренних слоев. Для предотвращения влияния данные между слоями приводят к нормализованному состоянию (11). Нормирование происходит по пакету, batch.

1. $\mu_B = \frac{1}{m} \sum_{i=1}^m x_i$
2. $\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$
3. $\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$
4. $\text{BN}_{\gamma, \beta}(x_i) = \gamma \hat{x}_i + \beta$ (12)

Здесь $x_i \in X$, где X - текущий пакет входных данных. Этапы 11.1, 11.2, 11.3 применяются только на этапе обучения сети. Параметры γ и β являются обучаемыми. При инференсе сети они обеспечивают соответствие среднего и смещения для входных данных аналогичными показателями тренировочных пакетов. На 5 продемонстрировано сравнение процессов обучения с и без описанной техники.



График значения функции потерь к количеству пройденных образцов

Графики:

1. Оранжевый - без нормализации по пакету

2. Серый - с нормализацией по пакету

Рисунок 5 — Улучшение сходимости нормализацией

3.3. Выделение признаков, downsampling

Для выделения признаков рассматривались сети архитектуры ResNet. Отличительной чертой данной архитектуры являются residual block и наличие skip connection. В состав одного блока последовательно входят:

1. Сверточный слой с паддингом
2. Нормализация по пакету
3. Активационная функция
4. Сверточный слой с паддингом
5. Нормализация по пакету

Результат блока объединяется с входными данными. Такое решение позволяет предотвратить затухание градиента при обратном распространении ошибки [25].

Для снижения размеров массива данных в архитектуре ResNet используются блоки, сочетающие в себе сверточный слой с ядром 1×1 , без паддинга и сдвигом в 2 и нормализацию по пакету.

В итоговое решение в качестве downsampling path вошла архитектура ResNet34. Она обладает меньшим количеством тренируемых параметров в сравнении с ResNet50, ResNet101, ResNet152. При этом демонстрируется достаточно высокое значение метрики качества.

3.4. Генерализация признаков, upsampling

В задачах сегментации в качестве

4. Данные

Эксперименты с обучением модели ИНС проводились на двух наборах данных: SkyFinder и набор из фотографий, сделанных на камеры смартфонов, с синтетической разметкой в объединении с данными из SkyFinder.

4.1. Датасет SkyFinder

SkyFinder представляет собой набор из 90.000 фотографий [26]. Все фотографии сделаны на статичные веб-камеры, расположенные вне зданий. В верхней части изображений преобладает регион неба. Средний процент пикселей относящихся к классу принадлежности равен 41 со стандартным отклонением в 16 процентов. Изображения покрывают широкий диапазон освещённости и погодных условий, что препятствует переобучению на конкретных значениях. Для каждого изображения имеется размеченная бинарная маска с описанным в постановке задачи свойством: 1 обозначает класс принадлежности, 0 - обратный ему.

4.2. Датасет с синтетической разметкой

Заключение

Список литературы

1. *Gonzalez R. C., Woods R. E.* Digital image processing. — Upper Saddle River, N.J. : Prentice Hall, 2008. — ISBN 9780131687288 013168728X 9780135052679 013505267X. — URL: <http://www.amazon.com/Digital-Image-Processing-3rd-Edition/dp/013168728X>.
2. *Liu X., Deng Z., Yang Y.* Recent progress in semantic image segmentation. — 2018. — arXiv: 1809.10198 [cs.CV].
3. *Wang C., Ding J., Chen P.* An efficient sky detection algorithm based on hybrid probability model // 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA). — 2015. — С. 919—922.
4. A Gentle Introduction to Deep Learning in Medical Image Processing / A. Maier [и др.]. — 2018. — arXiv: 1810.05401 [cs.CV].
5. Image processing in DNA / C. Pan [и др.]. — 2019. — arXiv: 1910.10095 [eess.IV].
6. *Stabinger S., Piater J., Rodríguez-Sánchez A.* Evaluating the Progress of Deep Learning for Visual Relational Concepts. — 2020. — arXiv: 2001.10857 [cs.CV].
7. *Li K.* A Brief Survey of Image Processing Algorithms in Electrical Capacitance Tomography. — 2015. — arXiv: 1510.04585 [cs.CV].

8. Adaptive and Generic Corner Detection Based on the Accelerated Segment Test / E. Mair [и др.] // Proceedings of the 11th European Conference on Computer Vision: Part II. — Heraklion, Crete, Greece : Springer-Verlag, 2010. — С. 183—196. — (ECCV'10). — ISBN 3642155510.
9. *Hartigan J. A.* Clustering Algorithms. — 99th. — USA : John Wiley, Sons, Inc., 1975. — ISBN 047135645X.
10. Detecting People Using Mutually Consistent Poselet Activations / L. Bourdev [и др.] //. — 12.2010. — С. 168—181. — DOI: 10.1007/978-3-642-15567-3_13.
11. *Ogiela M., Tadeusiewicz R.* Modern Computational Intelligence Methods for the Interpretation of Medical Images. Т. 84. — 01.2008. — С. 91. — ISBN 978-3-540-75399-5. — DOI: 10.1007/978-3-540-75402-2.
12. Comparison of semantic segmentation approaches for horizon/sky line detection / T. Ahmad [и др.] // 2017 International Joint Conference on Neural Networks (IJCNN). — 2017. — С. 4436—4443.
13. Deep Multi-modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges / D. Feng [и др.]. — 2019. — arXiv: 1902.07830 [cs.R0].
14. *Ronneberger O., Fischer P., Brox T.* U-Net: Convolutional Networks for Biomedical Image Segmentation. — 2015. — arXiv: 1505.04597 [cs.CV].
15. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs / L.-C. Chen [и др.]. — 2016. — arXiv: 1606.00915 [cs.CV].
16. RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation / G. Lin [и др.]. — 2016. — arXiv: 1611.06612 [cs.CV].

17. *Place C. L., Khan A. U., Borji A.* Segmenting Sky Pixels in Images. — 2017. — arXiv: 1712.09161 [cs.CV].
18. *Salman S., Liu X.* Overfitting Mechanism and Avoidance in Deep Neural Networks. — 2019. — arXiv: 1901.06566 [cs.LG].
19. *Ghojogh B., Crowley M.* The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial. — 2019. — arXiv: 1905.12787 [stat.ML].
20. *Smith L. N.* A disciplined approach to neural network hyperparameters: Part 1 – learning rate, batch size, momentum, and weight decay. — 2018. — arXiv: 1803.09820 [cs.LG].
21. *Labach A., Salehinejad H., Valaee S.* Survey of Dropout Methods for Deep Neural Networks. — 2019. — arXiv: 1904.13310 [cs.NE].
22. *Ioffe S., Szegedy C.* Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. — 2015. — arXiv: 1502.03167 [cs.LG].
23. *Dumoulin V., Visin F.* A guide to convolution arithmetic for deep learning. — 2016. — arXiv: 1603.07285 [stat.ML].
24. *Shimodaira H.* Improving predictive inference under covariate shift by weighting the log-likelihood function // Journal of Statistical Planning and Inference. — 2000. — Окт. — Т. 90. — С. 227–244. — DOI: 10.1016/S0378-3758(00)00115-4.
25. Deep Residual Learning for Image Recognition / К. Хе [и др.]. — 2015. — arXiv: 1512.03385 [cs.CV].
26. Sky Segmentation in the Wild: An Empirical Study / R. P. Mihail [и др.] // IEEE Winter Conference on Applications of Computer Vision (WACV). — 2016. — С. 1–6.