Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Южно-Уральский государственный университет»

(национальный исследовательский университет)

Факультет «Математики, механики и компьютерных наук»

Кафедра «Вычислительная механика сплошных сред»

РАБОТА ПРОВЕРЕНА Допустить к Защите

Рецензент, Заведующий кафедрой,

к.ф.-м.н., доцент кафедры МиФА д. ф.-м. н., профессор

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/М.А. Корытова \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Ковалев Ю.М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 г.

**Детектирование ключевых точек на лице человека с использованием сверточных нейронных сетей**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ – 010800.62.2015.838.ВКР

Руководитель

к., ф.-м. н., доцент кафедры

ВМСС

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Н.Л. Клиначева

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г.

Автор работы

студент группы ММиКН-460

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/А.О. Литвинов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г.

Нормоконтролер – к.ф.-м.н.,

доцент кафедры ВМСС

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/В.К.Рябинин

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г.

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Южно-Уральский государственный университет»

(национальный исследовательский университет)

Факультет «Математики, механики и компьютерных наук»

Кафедра «Вычислительная механика сплошных сред»

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ Заведующий кафедрой,  д. ф.-м. н., профессор  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Ю.М. Ковалев  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г. |

**ЗАДАНИЕ**  
студенту группы ММиКН-473

Литвинову Андрею Олеговичу

на выполнение выпускной квалификационной работы

по направлению 011000.62.–Механика. Прикладная математика.

1. **Тема выпускной квалификационной работы** Детектирование ключевых точек на лице человека с использованием сверточных нейронных сетей

**(Утверждена приказом ректора от 27.04.2015 № 838)**

1. **Перечень подлежащих исследованию вопросов**

2.1 Обзор литературы по решению задачи по тематике нейронных сетей

2.2 Составление программы расчёта.

2.3 Решение задачи для заданных начальных условий.

1. **Календарный план подготовки выпускной квалификационной работы**

|  |  |  |
| --- | --- | --- |
| Наименование этапов выпускной квалификационной работы | Срок выполнения этапов работы | Отметка о выполнении |
| 1.Обзор литературы | 10.02.16-03.03.16 | выполнено |
| 2. Изучение численного метода для решения поставленной задачи. Решение тестовой задачи. | 04.03.16-25.04.16 | выполнено |
| 3.Получение результатов, формулировка выводов, структурирование текста | 25.04.16-15.05.16 | выполнено |
| 4.Подготовка текста | 16.05.15-01.06.15 | выполнено |
| 5. Проверка и рецензирование работы руководителем, исправление замечаний. Подготовка доклада и текста выступления. Внешнее рецензирование. | 02.06.15-13.06.15 | выполнено |
| 6. Защита выпускной квалификационной работы. | 22.06.16 |  |

**4. Дата выдачи задания 10 февраля 2016 г.**

**Руководитель работы**

к.ф.-м.н., доцент кафедры ВМСС \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ С. У. Турлакова

Задание принял к исполнению**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/**А.О. Литвинов

УДК 004.93'1

**Литвинов А.О.**

Детектирование ключевых точек на лице человека с использованием сверточных нейронных сетей. Литвинов А.О. – Челябинск, 2016. – **N**с.

В дипломной работе реализована программа детекции ключевых точек лица человека на RGB изображении с помощью сверточной нейронной сети. Исследованы альтернативные методы для решения этой задачи. Также в рамках данной работы были проведены предварительные действия для приведения данных к корректному виду, необходимому для подачи на вход нейронной сети.

В приложении приведен текст программы на языке python

Список лит. – **K** назв.

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 6](#_Toc452686982)

[ПОСТАНОВКА ЗАДАЧИ 7](#_Toc452686983)

[ДЕТЕКТИРОВАНИЕ ЧЕЛОВЕЧЕСКОГО ЛИЦА НА ИЗОБРАЖЕНИИ ДЛЯ ВХОДА СВЕРТОЧНОЙ СЕТИ 9](#_Toc452686984)

[Интегральное представление изображений 9](#_Toc452686985)

[Признаки Хаара 10](#_Toc452686986)

[Построения классификатора на основе алгоритма бустинга и построение каскадной структуры 11](#_Toc452686987)

[НАХОЖДЕНИЕ ТОЧЕК ЛИЦА ПО ДЕТЕКТИРОВАННОМУ СЕКТОРУ 13](#_Toc452686988)

[СТРУКТУРА СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ 13](#_Toc452686989)

[ПРИНЦИП РАБОТЫ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ 15](#_Toc452686990)

[Convolution слой 15](#_Toc452686991)

[Pooling слой 15](#_Toc452686992)

[Fully-connected 16](#_Toc452686993)

[ПРИНЦИП ОБУЧЕНИЯ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ ГРАДИЕНТНЫМИ МЕТОДАМИ 17](#_Toc452686994)

[Вычисление ошибки 17](#_Toc452686995)

[Вычисление градиента 18](#_Toc452686996)

[МЕТОДЫ ПРИМЕНЯЕМЫЕ ПРИ ОБУЧЕНИИ 19](#_Toc452686997)

[Обучение на GPU 19](#_Toc452686998)

[Увеличение тренировочной выборки 19](#_Toc452686999)

[Предотвращение переобучения 19](#_Toc452687000)

[Dropout 20](#_Toc452687001)

[Перемешивание обучающих примеров 21](#_Toc452687002)

[Паралич сети 21](#_Toc452687003)

[АРХИТЕКТУРА СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ В ДАННОЙ РАБОТЕ 23](#_Toc452687004)

[РЕЗУЛЬТАТЫ 24](#_Toc452687005)

[ЗАКЛЮЧЕНИЕ 25](#_Toc452687006)

[ЛИТЕРАТУРА 26](#_Toc452687007)

[ПРИЛОЖЕНИЕ 27](#_Toc452687008)

[Текст программы на языке python 27](#_Toc452687009)

ВВЕДЕНИЕ

Искусственные нейронные сети в настоящее время получили широкое распространение в различных сферах человеческой деятельности, таких как распознавание и классификация образов, прогнозирование, управление. Пример конкретных задач – автоматическое управление автомобилем, распознавание изображений, речи, текстовые чат боты. Это произошло благодаря тому, что они имеют возможность синтезировать какие-либо алгоритмы, вычленяя из больших выборок данных произвольные закономерности. Изначально они построены по примеру биологических нейронных сетей – сетей нервных клеток животного. Один из видов искусственных нейронных сетей - сверточные нейронные сети, в основном применяющиеся для работы с изображениями. Причину этого можно увидеть в предпосылках их зарождения. Сверточные сети построены по примеру обработки визуальной информации у живых существ – в мозге есть области, выделяющие какие-либо инвариантные закономерности в изображениях – например линии.

Целью выпускной квалификационной работы является исследование возможностей применения сверточных нейронных сетей в области детекции произвольных объектов на RGB изображении. Эта область стала актуальной, в связи с появлением задач, где данный инструмент является вспомогательным а порой ключевым. А именно: досмотр людей на таможне, проект «Умный дом», различные задачи киноиндустрии, 3D анимация человекоподобных моделей и т.д. На сегодняшний день в данной сфере не существует решения дающего результат, хоть как-то близкий к идеальному.

Для достижения поставленной цели необходимо исследовать предметную область, исследовать методы для решения аналогичных задач, исследовать вспомогательные методы.

Также требуется разработать архитектуру нейронной сети для конкретной задачи – для детекции ключевых точек лица человека. Этот тип задачи относиться к классу регрессионных задач, то есть восстановление каких-либо зависимостей. Аналогичные методы решения возможно применять к схожим проблемам – распознавание рукописного текста, номеров автомобилей, дорожных знаков.

ПОСТАНОВКА ЗАДАЧИ

Данные получаемые с видеокамеры(кадр) в момент времени обозначим . Каждый кадр есть матрица размерности , каждый элемент принадлежит множеству , где – ширина кадра, - высота кадра, R,G, B – множества в диапазоне от 0 до 255, обозначающие кодировки цветов в формате RGB. Видеопоследовательность есть упорядоченная последовательность кадров (), где – количество кадров в видеопоследовательности.

Сформулируем задачу детектирования точек на лице человека. По имеющемуся кадру предсказать положение точек лица человека, предполагая при этом, что в кадре находится произвольное количество людей, на расстоянии от 0.5 до 2 метров от камеры. Требуется предсказать положение 8 точек:

* Центр подбородка
* Левый край губ
* Правый край губ
* Центр носа
* Левую височную область
* Левый край левого глаза
* Правую височную область
* Правый край правого глаза

Процесс детекции происходит в два этапа:

1. Детектирование человеческих лиц на изображение с помощью каскада Хаара, для последующей передачи выделенных областей на вход сверточной нейронной сети,
2. Детектирование на выделенной области точек лица с помощью сверточной нейронной сети.

В работе необходимо из видеопоследовательности в режиме реального времени выделять точки лица. В качестве исходных данных выступает видеопоследовательность, полученная с неподвижной камеры, с разрешением пикселей в формате RGB глубиной 8 бит на один канал и частотой кадров 30 в секунду.

Заданные входные данные накладывают на возможные методы решения ограничения в плане оптимизации.

ДЕТЕКТИРОВАНИЕ ЧЕЛОВЕЧЕСКОГО ЛИЦА НА ИЗОБРАЖЕНИИ ДЛЯ ВХОДА СВЕРТОЧНОЙ СЕТИ

Для детектирования лица используется метод, называемый метод Виолы-Джонса, также называемый каскадом Хаара.

Метод был разработан в 2001 г. Майклом Джонсом и Полом Виолой. Этот метод до сих пор не потерял своей актуальности в связи с одним из лучших по соотношению показателей скорости работы и эффективности распознавания. Этот детектор обладает крайне низкой вероятностью ложного срабатывания, порядка 0.00007%. Алгоритм хорошо работает при ориентации детектируемого объекта в пределах 30 градусов. При большем угле, чем 30 градусов, процент обнаружения резко уменьшается. Это не позволяет в стандартной реализации, без каких-либо модификаций, производить детекцию человеческого лица, повернутого под произвольным углом.

Данный метод в общем виде ищет лица и черты лица по общему принципу сканирующего окна.

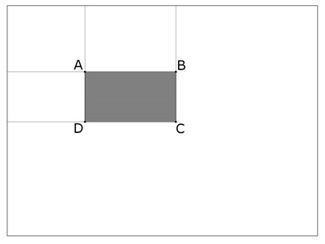
Интегральное представление изображений

Для расчета суммарной яркости прямоугольного участка изображения, используют интегральное представление. Это представление используется часто и в других методах, например, фильтрах Хаара, в вейвлет преобразованиях, SURF.

Интегральное представление позволяет быстро рассчитывать суммарную яркость произвольного прямоугольника на данном изображении, причем время расчета не зависит от площади прямоугольника. Оно записывается в виде матрицы, размеры которой совпадают с размерами исходного изображения I:

где – значение интегрального представления в точке , — яркость пиксела исходного изображения.

Каждый элемент матрицы представляет собой сумму интенсивностей пикселей в прямоугольнике от до . Расчет такой матрицы занимает линейное время. Для того чтобы вычислить сумму прямоугольной области в интегральном представлении изображения требуется всего 4 операции обращения к массиву и 3 арифметические операции. Это позволяет быстро рассчитывать признаки Хаара для изображения в процессе обучения и распознавания.



Для прямоугольного изображения ABCD ее интегральное значение можно выразить через суммы и разности смежных интегральных значений прямоугольников по формуле:

Признаки Хаара

Целью признаков Хаара является сравнения яркостей в двух прямоугольных областях изображения.

В стандартном методе Виолы–Джонса используются прямоугольные признаки, рис. 1. Эти признаки называются примитивами Хаара.

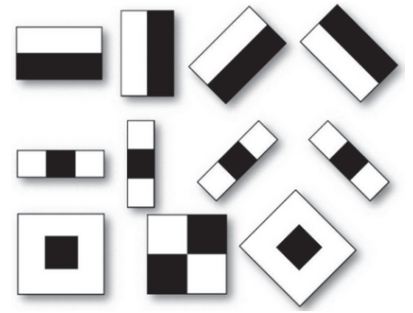


Рис. 1.

Примитивы признаков Хаара

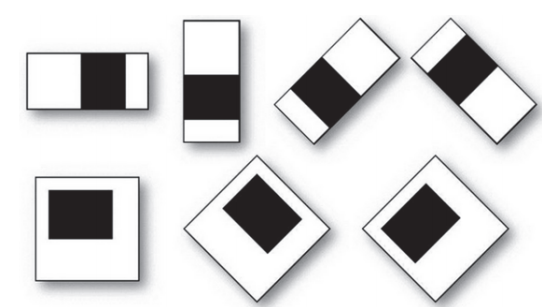


Рис. 2. Дополнительные признаки Хаара

В расширенном методе Виолы–Джонса, представленном в библиотеке OpenCV, используются дополнительные признаки (рис. 2).

Вычисляемым значением такого признака будет:

где – значение интегрального представления, закрываемое светлой частью признака, а – значение интегрального преставления, закрываемое темной частью признака. Признаки Хаара дают значение перепада яркости между и .

Построения классификатора на основе алгоритма бустинга и построение каскадной структуры

Бустинг (boosting) — это процедура последовательного построения композиции алгоритмов машинного обучения, когда каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих алгоритмов. Бустинг представляет собой жадный алгоритм построения композиции алгоритмов. Изначально понятие бустинга возникло в работах по вероятно почти корректному обучению в связи с вопросом: возможно ли, имея множество плохих (незначительно отличающихся от случайных) алгоритмов обучения, получить хороший.

В течение последних 10 лет бустинг остаётся одним из наиболее популярных методов машинного обучения, наряду с нейронными сетями и машинами опорных векторов. Основные причины — простота, универсальность, гибкость (возможность построения различных модификаций), и, главное, высокая обобщающая способность.

Простое голосование базовых классификаторов:

– классифицируемый объект, – базовые классификаторы

Композиция может быть лучше базовых если они лучше случайного классификатора и достаточно различны.

Каскадная структура повышает скорость обнаружения, фокусируя свою работу на наиболее информативных областях изображения.

Структура каскадного детектора приведена на рис. 3. Каскад состоит из слоев, которые представляют собой классификаторы, обученные с помощью процедуры бустинга.

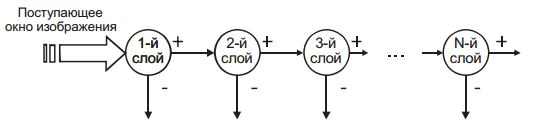


Рис. 3. Структура каскадного детектора

НАХОЖДЕНИЕ ТОЧЕК ЛИЦА ПО ДЕТЕКТИРОВАННОМУ СЕКТОРУ

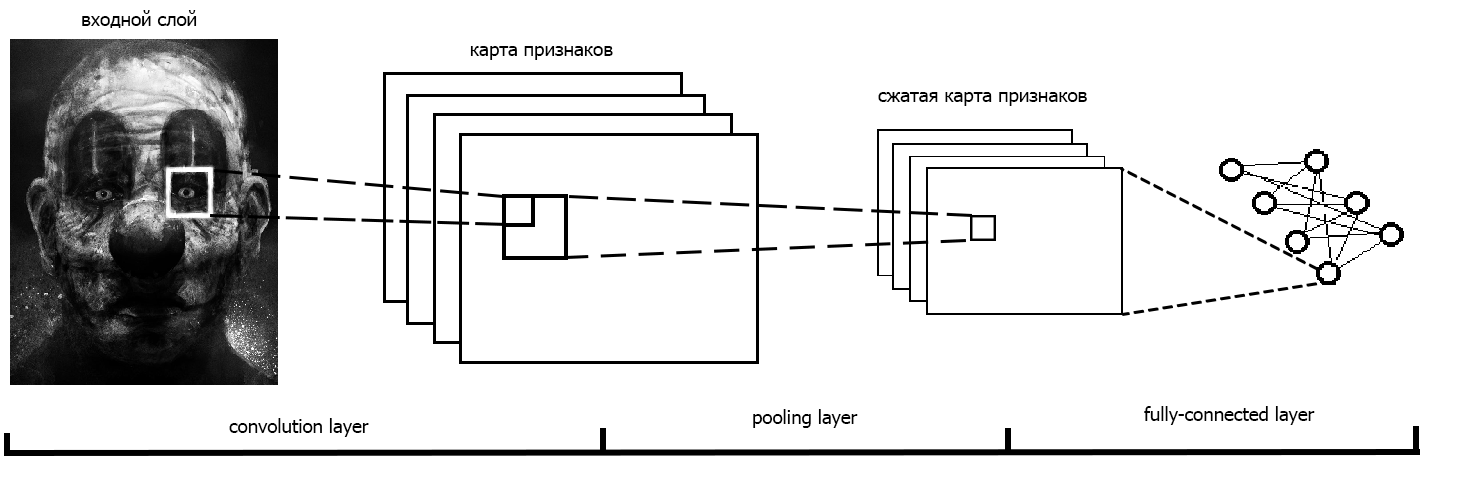
СТРУКТУРА СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

Сверточные нейронные сети содержат три типа слоев:

1. **Convolutional(сверточные).** В операции свертки используется лишь ограниченная матрица весов небольшого размера, которую «двигают» по всему обрабатываемому слою (в самом начале по входным данным), формируя после каждого сдвига сигнал активации для нейрона следующего слоя с аналогичной позицией. То есть для различных нейронов выходного слоя используются общие веса — матрица весов, которую также называют набором весов или ядром свертки. Она построена таким образом, что графически кодирует какой-либо один признак, например, наличие наклонной линии под определенным углом. Тогда следующий слой, получившийся в результате операции свертки такой матрицей весов, показывает наличие данной наклонной линии в обрабатываемом слое и ее координаты, формируя так называемую карту признаков (feature map). В сверточной нейронной сети количество ядер, как правило много: они кодируют признаки исходных данных.

При этом в зависимости от метода обработки краёв исходной матрицы результат может быть меньше исходного изображения , такого же размера или большего размера .

1. **Pooling, subsampling (операция субдискретизации).** После каждого сверточного слоя, может быть pooling слой. Pooling слой берет небольшие прямоугольные блоки из сверточного слоя и вычисляет один выход из этого блока. Есть несколько способов выполнить эту операцию, например, взяв среднее, максимальное или какую-либо линейную комбинацию нейронов в блоке.
2. **Fully-Connected(персептрон, MLP, Полносвязная сеть).** После нескольких convolutional и max-pooling слоев следуют полносвязные слои. Fully connected слой принимает все нейроны из предыдущего слоя (будь то fully connected, pooling или convolutional) и соединяет его с каждым нейроном, который он имеет. Fully connected слои не могут быть расположены где угодно, то есть не может быть сверточных слоев после того, как идет fully connected слой.



Первые два типа слоёв (convolutional, pooling), чередуясь между собой, формируют входной вектор признаков для fully-connected. Fully-connected слой по вектору признаков выдает обобщенный результат. Сеть можно обучать с помощью градиентных методов.

Применение нейронных сетей не ограничивается двумерным случаем. Возможно точно таким же образом построить одно или трехмерные сверточные нейронные сети; convolutional фильтры будут просто иметь размеры, определенные надлежащим образом, и pooling слои поменяют размерность. Например, возможно использовать одномерные сверточные сети на аудио данных.

ПРИНЦИП РАБОТЫ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

Convolution слой

Свёрточный слой реализует идею локальных рецептивных полей, то есть каждый выходной нейрон соединен только с определённой областью входной матрицы и таким образом моделирует некоторые особенности человеческого зрения.

Этот слой описывается следующий формулой:

Здесь - выход слоя , - функция активации, - коэффициент сдвига, символом ∗ обозначена операция свёртки входа с ядром .

При этом за счёт краевых эффектов размер исходных матриц изменяется.

Здесь - карта признаков (выход слоя ), - функция активации, - коэффициент сдвига для карты признаков , - ядро свёртки номер , - карты признаков предыдущего слоя.

Pooling слой

Слои этого типа выполняют уменьшение размера входной карты признаков. Это можно делать разными способами, рассмотрим метод выбора максимального элемента (max-pooling) - вся карта признаков разделяется на ячейки размера, из которых выбираются максимальные по значению. Формально слой может быть описан следующим образом.

Здесь - выход слоя , - функция активации, - коэффициенты, - операция выборки локальных максимальных значений.

Использование этого слоя позволяет улучшить распознавание образов с изменённым масштабом (уменьшенных или увеличенных).

Fully-connected

Последний из типов слоёв это слой многослойного перцептрона (MLP), его можно описать следующим соотношением.

Здесь - выход слоя , - функция активации, - коэффициент сдвига, - матрица весовых коэффициентов.

ПРИНЦИП ОБУЧЕНИЯ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ ГРАДИЕНТНЫМИ МЕТОДАМИ

Вычисление ошибки

**Для выходного (Fully-connected)** слоя ошибка рассчитывается следующим образом.

Здесь - ожидаемый (учебный) выход, - реальный выход, - производная функции активации по её аргументу

**Для скрытых слоёв Fully-connected** ошибка имеет следующий вид.

Здесь - ошибка слоя , - производная функции активации, - состояние (не активированное) нейронов слоя , - матрица весовых коэффициентов слоя .

**Ошибка на выходе convolutional слоя** формируется путём простого увеличения размера матриц ошибки следующего за ним pooling слоя.

Здесь - ошибка слоя , - производная функции активации, - состояние (не активированное) нейронов слоя , - операция увеличения размера матриц.

**Ошибка на выходе pooling** слоя рассчитывается путём выполнения ”обратной свёртки” карт признаков следующего за ним свёрточного слоя, то есть над каждой картой признаков выполняется свёртка с соответствующим ”перевернутым” ядром, при этом за счёт краевых эффектов размер исходных матриц изменяется. Далее над получившимися картами вычисляются несколько частичных сумм по числу ядер свертки, в соответствии с матрицей смежности polling и convolutional слоёв.

Здесь - ошибка слоя , - производная функции активации, - состояние (не активированное) нейронов слоя , - ядра свёртки.

Вычисление градиента

Процедура вычисления градиента функции ошибки. Обучение сводиться к её минимизации в пространстве весов сети.

* Градиент для ядра свёртки можно посчитать как свёртку матрицы входа convolutional слоя с "перевёрнутой" матрицей ошибки для выбранного ядра.

Здесь - ошибка слоя , - вход слоя , - ядра свёртки.

Градиент для сдвига для convolutional слоя вычисляется как сумма значений соответствующей матрицы ошибки.

Здесь - ошибка слоя

* Градиент для коэффициентов pooling слоя вычисляется следующим образом.

Здесь - выход слоя , - ошибка слоя , - операция выборки локальных максимальных значений.

Градиент для коэффициента сдвига для pooling слоя вычисляется как сумма значений соответствующей матрицы ошибки.

Здесь - ошибка слоя

* Градиент для весов MLP выглядит следующим образом.

Здесь - ошибка слоя , - вход слоя , - матрица весовых коэффициентов слоя .

МЕТОДЫ ПРИМЕНЯЕМЫЕ ПРИ ОБУЧЕНИИ

Обучение на GPU

Важную роль в процессе обучения нейронных сетей занимает время ее обучения, так как современные сети как правило очень сложны и без какой-либо оптимизации они могут обучаться годами. Один из способов ускорить обучение – это распараллеливание процессов.

В сверточных сетях есть независимые операции(например вычисление по различным фильтрам соответствующим им future map)

В глубоком обучении активно используется обучение на видеокартах, так как они обладают множеством ядер.

GPU используются для обучения глубоких нейронных сетей с помощью намного более крупных обучающих последовательностей в более сжатые сроки, с использованием меньшей инфраструктуры ЦОД. GPU также используются, чтобы воспроизводить эти учебные модели машинного обучения для выполнения задач классификации и прогнозирования на облаке. При этом графические процессоры позволяют работать с данными большего объема и с более высокой производительностью, потребляя меньше энергии и на базе меньшей инфраструктуры.

К числу тех, кто впервые применил графические ускорители для решения задач машинного обучения, относятся многие крупные веб-компании и социальные сетевые серверы, наряду с научно-исследовательскими институтами высокого ранга в области обработки и анализа данных и машинного обучения. Благодаря тысячам вычислительных ядер и увеличению производительности приложений в 10-100 раз по сравнению с CPU, GPU стали процессорами, которые выбирают специалисты по обработке данных для работы с данными большого объема.

Увеличение тренировочной выборки

Зачастую в глубоком обучении наибольшую выгоду приносит простое увеличение тренировочной выборки.

Предотвращение переобучения

Переподгонка, переобучение, (overfitting, overtraining,) —явление, возникающее при решении задач обучения “c учителем”, когда ошибка на тренировочной выборке оказывается значительно меньше, чем ошибка на тестовой выборке. Переобучение возникает при избыточно большой подгонке под тренировочную выборку или при использовании избыточно сложной модели.

Для того, чтобы избежать переобучения, требуется использовать дополнительные методы:

1. Перекрёстная проверка (*Cross-validation*)

Процедура оценивания обобщающей способности алгоритмов, обучаемых по какой-либо тренировочной базе данных.

Выбирается множество разбиений исходной выборки на две подвыборки исходных тренировочных данных: на новую тренировочную и тестовую. Для каждого разбиения выполняется обучение алгоритма на тренировочной подвыборке, затем оценивается его средняя ошибка на объектах тестовой подвыборки. Оценкой скользящего контроля называется средняя по всем разбиениям величина ошибки на контрольных подвыборках.

Если выборка независима, то средняя ошибка скользящего контроля даёт несмещённую оценку вероятности ошибки. Это отличает её от средней ошибки на обучающей выборке, которая может оказаться смещённой (заниженной) оценкой вероятности ошибки.

Перекрестная проверка является стандартной методикой предотвращения переобучения и тестирования алгоритмов регрессии, классификации и прогнозирования.

1. L1 и L2 регуляризация

Регуляризация в машинном обучении— метод добавления некоторой дополнительной информации к условию с целью предотвратить переобучение или решить некорректно поставленную задач. Эта информация представляет штраф за избыточную сложность модели. В случае L1 это добавление к штрафующей функции модуля весов, в L2 – добавление квадратов весов.

1. Ранняя остановка обучения

Dropout

При неограниченных вычислительных ресурсах лучшим способом регуляризации моделей фиксированного размера является усреднение предсказания по всем возможным настройкам параметров. Комбинация моделей почти всегда увеличивает качество работы, но в больших сетях использование комбинации нескольких сетей непомерно дорого. Комбинирование нескольких сетей наиболее полезно, когда сети имеют различную архитектуру или обучаются на различных тренировочных данных. Обучение сразу нескольких сетей довольно дорогостоящая и долгая процедура из-за очень большого количества параметров сети, требующая большого количества вычислительных ресурсов. Более того, обучение больших сетей требует большого количества обучающих данных, которых всегда достаточно. Но даже если натренировать большие сети, то их применение будет также требовать больших вычислительных ресурсов.

Dropout – технология позволяющая решить обе этих задачи. Она предотвращает переобучение и позволяет эффективно комбинировать сети с различными архитектурами. Термин «dropout» означает удаление некоторых узлов в нейронных сетях.

Под удалением узлов понимается временное исключение входящих и исходящих связей нейрона. Выбор нейрона, который будет удален, случаен. В простейшем случае вероятность исключения любого узла одинакова и выбирается либо, исходя из конкретной задачи.

Применение dropout к нейронным сетям создает несколько «разреженных» сетей внутри одной. Каждая разряженная сеть содержит только оставшиеся после отсечения узлы. Сеть, состоящая из n нейронов, может образовать разреженных сетей. Эти сети имеют общие веса, поэтому общее число параметров будет или меньше. Для каждого представления каждого тренировочного случая, выбирается разреженная сеть и обучается. Таким образом, обучение нейронной сети с использование dropout можно свести к обучению набора из разряженных сетей с общими весами, где каждая сеть обучается гораздо реже, чем все в совокупности. При тестировании сети невозможно усреднить выход каждой сети из-за их экспоненциального количества. Но на практике хорошо работает другой метод усреднения. Идея в том, чтобы использовать единственную сеть без dropout. Веса данной сети являются сжатыми весами тренированных сетей. Если какой-то нейрон имел вероятность включения в сеть p – при обучении то, во время тестирования вес будет умножен на данную вероятность.

Использование dropout в качестве усреднения ведет к улучшению качества работы сети, уменьшению ошибки на широком спектре задач классификации по сравнению с другими методами регуляризации.

Перемешивание обучающих примеров

Так как данные в выборке могут быть определены изначально в некоторой определенной последовательности(например в выборке данных мужчин и женщин, данные могут быть отсортированы по полу), то получая данные, по порядку их следования, модель лучше обучается по последним данным.

Паралич сети

Слишком большие значения весовых коэффициентов, вызывающие «паралич» нейронной сети или чрезмерное уменьшение шага обучения, увеличивающее время обучения до чрезмерно большого значения. Кроме того, при некоторых условиях обучение нейронной сети может превратиться в неустойчивый процесс, когда, выучивая одно, сеть при этом забывает другое

Способы избежать паралича сети:

1. Уменьшить размер шага градиента, соответственно увеличится время обучения.

2.В слоях с большими весами отказаться от зависимости

3.Эвристические правила для предотвращения роста весов.

АРХИТЕКТУРА СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ В ДАННОЙ РАБОТЕ

input (3, 100, 100) produces 30000 outputs

conv1 (32, 98, 98) produces 307328 outputs

pool1 (32, 49, 49) produces 76832 outputs

dropout1 (32, 49, 49) produces 76832 outputs

conv2 (64, 48, 48) produces 147456 outputs

pool2 (64, 24, 24) produces 36864 outputs

dropout2 (64, 24, 24) produces 36864 outputs

conv3 (128, 23, 23) produces 67712 outputs

pool3 (128, 12, 12) produces 18432 outputs

dropout3 (128, 12, 12) produces 18432 outputs

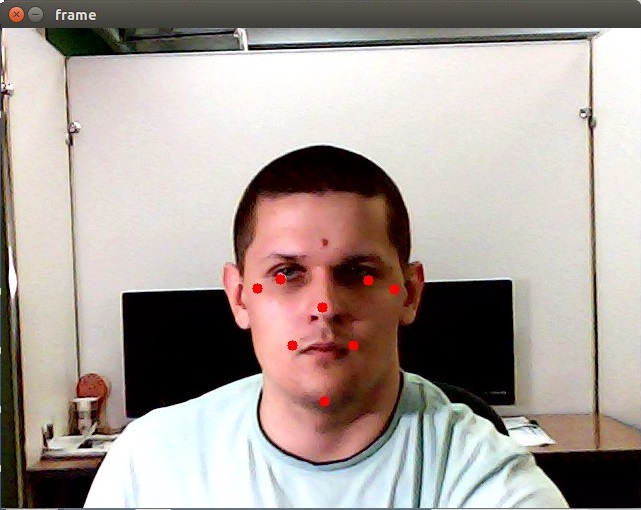
hidden4 (500) produces 500 outputs

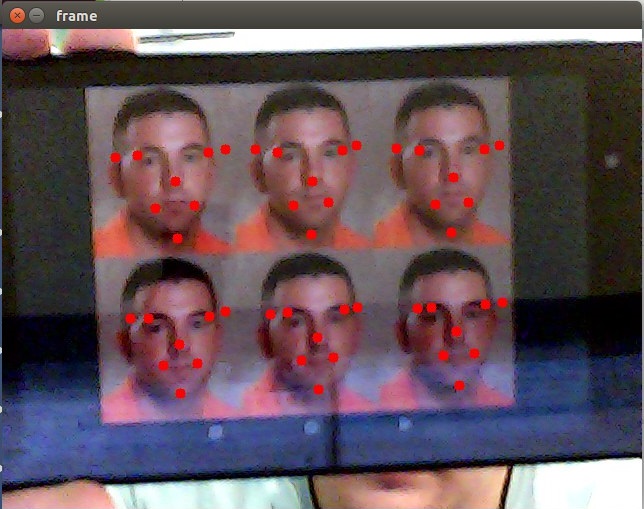
dropout4 (500) produces 500 outputs

hidden5 (500) produces 500 outputs

output (16) produces 16 outputs

РЕЗУЛЬТАТЫ





ЗАКЛЮЧЕНИЕ

Разработан и протестирован комплекс алгоритмов для распознавания точек лица человека. Тест проводился в режиме реального времени. Установлено, что приемлемый уровень качества возможен при присутствии перед камерой количества лиц не более шести. При шести лицах в кадре происходит обработка 7 кадров в секунду. При одном лице – 25 кадров в секунду. Возможно существенное улучшение работы алгоритма за счет следующих подходов:

1. Уменьшение “глубины” нейронной сети без потери качества.
2. Распараллеливание предсказания точек лица, по областям определенным методом Виолы-Джонса
3. Использование более качественных детекторов лиц, чем метод Виолы-Джонса.
4. Расширение обучающей выборки для получения лучшего результата.
5. Использование языков более низкого уровня для предсказания по предобученной модели определенной архитектуры

ЛИТЕРАТУРА

1. Hubel, D. H.; Wiesel, T. N. (1968-03-01)."Receptive fields and functional architecture of monkey striate cortex". *The Journal of Physiology* **195** (1): 215–243.ISSN 0022-3751.PMC 1557912. PMID 4966457.
2. LeCun, Yann; Léon Bottou; Yoshua Bengio; Patrick Haffner (1998)."Gradient-based learning applied to document recognition"
3. <http://www.deeplearningbook.org/>
4. Viola P., Jones M.J. Rapid object detection using a boosted cascade of simple features // IEEE Conf. on Computer Vision and Pattern Recognition. – Kauai, Hawaii, USA, 2001. – V. 1. – P. 511–518. 2.
5. Viola P., Jones M.J. Robust realtime face detection // International Journal of Computer Vision. – 2004. – V. 57. – № 2. – P. 137–154.
6. <http://cogprints.org/5869/1/cnn_tutorial.pdf>
7. <http://cs231n.github.io/>
8. <http://neuralnetworksanddeeplearning.com/chap6.html>
9. <http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/>
10. <http://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/>
11. <http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/>
12. <http://mechanoid.kiev.ua/ml-lenet.html>
13. Борисов Е. О методах обучения многослойных нейронных сетей прямого распространения.   
    – <http://mechanoid.kiev.ua/neural-net-backprop.html>
14. <http://mechanoid.kiev.ua/ml-lenet.html>
15. <http://oxozle.com/2015/04/11/metod-raspoznavaniya-lic-violy-dzhonsa-viola-jones>

ПРИЛОЖЕНИЕ

Текст программы на языке python