|  |  |  |  |
| --- | --- | --- | --- |
| Министерство образования и науки РФ  ФГБОУ ВПО «Пермский государственный национальный исследовательский университет» | | | |
|  | | | |
|  | | Кафедра математического обеспечения вычислительных систем | |
| Отчёт по предмету  «Методы искусственного интеллекта» | | | |
| **Исследование изображений плевральных выпотов для ранней диагностики заболеваний** | | | |
|  | | | Работу выполнил студент  группы ПМИ-1,2 4 курса  механико-математического ф-та  Заманов М.Р. |
|  | | | Работу проверил:  Доктор технических наук, профессор Ясницкий Леонид Нахимович |
|  |  | | “\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019г |
| Пермь 2019 | | | |

**Цель работы:**

Создание программного продукта, который мог бы ставить диагноз по микроскопическому изображению плеврального выпота пациента. Необходимо это для того, чтобы медицинские работники могли чуть более точно определять онкологические заболевания.

**Для достижения поставленной цели необходимо:**

1. Изучить предметную область, нюансы, связанные с ней.
2. Изучить теоретические основы сверточных нейронных сетей.
3. Более подробно ознакомиться с предварительной обработкой изображения с использованием функций OpenCV, а именно:
   1. Устранение шумовых помех на изображении
   2. Бинаризация изображения

**Сверточные нейронные сети**

Сверточные нейронные сети (convolutional neural networks, CNN) — это широкий класс архитектур, основная идея которых состоит в том, чтобы переиспользовать одни и те же части нейронной сети для работы с разными маленькими, локальными участками входов. Как и многие другие нейронные архитектуры, сверточные сети известны довольно давно, и в наши дни у них уже нашлось много самых разнообразных применений, но основным приложением, ради которого люди когда-то придумали сверточные сети, остается обработка изображений.

Название архитектура сети получила из-за наличия операции свертки, суть которой в том, что каждый фрагмент изображения умножается на матрицу (ядро) свертки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения.

Сверточная нейронная сеть обладаем рядом критических преимуществ:

1. При архитектуре сверточных нейронных сетей возможно распараллеливание вычислительных процессов за счет использования графических процессоров.
2. По сравнению с полносвязной нейронной сетью (перцептрон), сверточная нейронная сеть обладает гораздо меньшим количеством настраиваемых весов, что подталкивает сеть к обогащению демонстрируемой информации при обучении. Что делает возможным распознавание изображений с большим количеством признаков при наименьших требованиях к техническим средствам.
3. Сверточная нейронная сеть обладает достаточной устойчивостью к повороту и сдвигу распознаваемого изображения.

Среди недостатков сверточной нейронной сети обычно отмечают сложную настройку параметров (количество слоев, количества ядер каждого из слоев, варианты реализации каждого из слоев и т.д.). При разработке архитектуры сети стоит учитывать специфику области применения и требования, выдвигаемые к сети.

Традиционно сверточная нейронная сеть содержит в себе следующие типы слоев:

1. Сверточный (convolutional). Используется для генерации «карт значений» при помощи фильтров (ядер свертки). Веса ядер свертки в начале обучения устанавливаются случайно, и корректировка происходит во время обучения.
2. Субдискретизирующий (подвыборка, pooling). Основной задачей этого типа слоев является уплотнение карты признаков посредством сжатия изображения (обычно группа пикселей 2x2 преобразуются в 1 пиксель). Слой подвыборки, как правило, идет сразу после сверточного слоя и выполняет функцию отсеивания ненужной информации.
3. Активационный. Функция активации, через которую проходят результаты свертки или пуллинга.
4. Полносвязный. После прохождения всех слоев свертки и пуллинга, остается большой набор каналов, хранящих абстрактные понятия, полученные из исходного изображения. Эти данные объединяются и передаются на полносвязную нейронную сеть, состоящую из одного или более слоев.

**Реализация приложения**

Программный продукт разрабатывался на ЯП Python с использование таких библиотек, как Keras и OpenCV.

Библиотека OpenCV использовалась для предварительной обработки изображений плевральных выпотов. Библиотека же Keras использовалась для реализации сверточной нейронной сети, обучения и распознавания.

**Модель**

Была построена следующая модель нейронной сети:

* На вход подается изображений 256×256 пикселей.
* Нейронная сеть содержит 4 идущих подряд сверхточных нейронных слоя. Во всех используются фильтры размерами 3×3 каждый. В первых двух слоях используется 256 фильтра в каждом, в последующих двух - 128. В каждом слое используется субдискретизирующий слой уменьшения размерности с использованием метода выбора максимального элемента (max pooling) с фильтром 2×2.
* Плоский слой.
* 3 полносвязных слоя (256 нейронов) + функция RELU.
* Результат – 2 класса.

Данная модель сверточной нейронной сети продемонстрирована на Рис. 1.

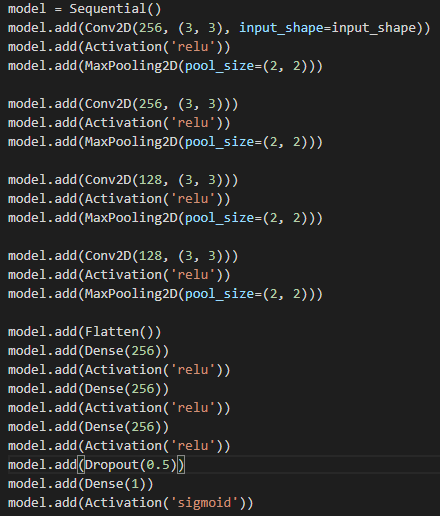


Рис. 1 Модель CNN

**Слой регуляризация (Dropout)**

Переобучение в нейронных сетях можно происходит, когда находящиеся рядом нейроны настраиваются на совместную работу, тем самым настраиваясь на особенности конкретной выборки, а не на общие закономерности, которые характерны для различных изображений.

Dropout – техника предотвращения переобучения, которая в процессе обучения, когда на вход нейронной сети подаётся новый объект, случайным образом выключается некоторое количество нейронов с заданной вероятностью. В нашей модели Dropout(0.5) означает, что нейрон будет отключаться с вероятностью 50%. Оставшимся в сети нейронам во время обучения приходится подбирать веса так, чтобы обнаруживать важные признаки самостоятельно, без участия соседних нейронов.

**Снижение погрешности посредством увеличения обучающего множества**

Также для улучшения результатов было принято решение увеличить обучающее множество посредством поворотов всех изображений на по 10 градусов от 0 до 360 градусов. Тем самым обучающее множество было увеличено в 36 раз.

На Рис. 2 можно наблюдать всю предварительную обработку изображений, которая производится перед обучением. Где Square() – преобразует изображение к квадратному виду, Squeeze – изменяет размеры изображения на указанные, Blur – устраняет шумы на изображении, Binary – отвечает за бинаризацию изображения, Turn – осуществляет поворот изображения на указанное число градусов.

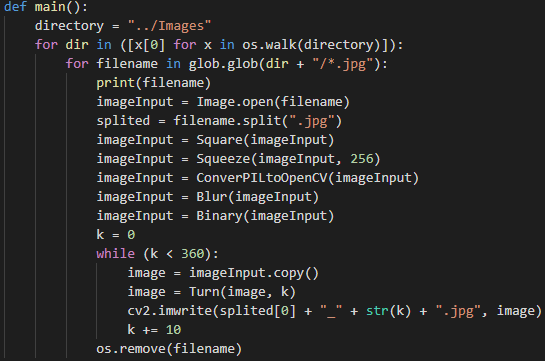


Рис. 2 Предварительная обработка изображения

**Компиляция и генераторы**

При компиляции в качестве функции ошибки указывается binary\_crossentropy, так как всего 2 класса (онкология и не онкология). Оптимизатор adam, метрика accuracy. Код на языке Python продемонстрирован на Рис. 3.



Рис. 3 Компиляция модели

Также создаем генератор изображений (Рис. 4), который основывается на классе ImageDataGenerator. Генератор делит значения всех пикселей изображения на 255.



Рис. 4 Генератор изображений

Затем создаем генераторы данных для обучения и тестирования на основе изображений из каталога. Генератор для обучения нейронной сети продемонстрирован на Рис. 5.

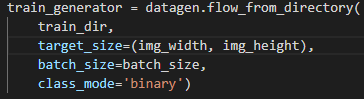


Рис. 5 Генератор для обучения

**Обучение нейронной сети**

При помощи метода fit\_generator (Рис. 6) происходит непосредственно обучение нейронной сети. Данному методу сообщается генератор данных для обучения и генератор данных для проверки, также необходимо указать необходимое количество обращений к генераторам и количество эпох обучения.

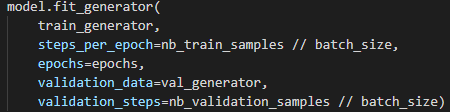


Рис. 6 Обучение нейронной сети

**Сохранение модели и оценка качества работы сети**

Сохранение модели осуществляется при помощи метода save. Для оценки же качества обученной сети используется метод evaluate\_generator, в который передается тестовый генератор и необходимое количество обращений к нему. Код оценки качества работы сети продемонстрирован на Рис. 7.



Рис. 7 Оценка качества работы сети

**Литература**

1. Ясницкий Л.Н. Введение в искусственный интеллект. 2005.
2. Ясницкий Л.Н. Интеллектуальные информационные технологии и системы. 2007.
3. Николаенко С., Кадурин А., Архангельская Е. Сверточные нейронные сети // Глубокое обучение. 2018. С. 176-231.
4. Сверточные нейронные сети // intellect.ml. 2016. URL: <https://intellect.ml/svertochnaya-nejronnaya-set-convolutional-neural-network-cnn-6013> (дата обращения: 15.12.2018).
5. Сверточная нейронная сеть// habr.com. 2016. URL: https://intellect.ml/svertochnaya-nejronnaya-set-convolutional-neural-network-cnn-6013 (дата обращения: 15.12.2018).