



РАЗРАБОТКА СППР ДЛЯ ДИАГНОСТИКИ РАКА ЛЕГКОГО

В статье рассматриваются вопросы разработки системы поддержки принятия решения для диагностики рака легкого на основе снимков компьютерной томографии. СППР строится на базе сверточной нейронной сети.

СППР, сверточная нейронная сеть, новообразование.

Без медицинских знаний тяжело определить, злокачественное или доброкачественное новообразование у человека. Даже врач при визуальном анализе компьютерной томографии (КТ) снимков может ошибаться. Зачастую проблема ошибок в анализе возникает из-за большого количества признаков новообразований. Одной из главных задач становится необходимость уменьшения количества ложных срабатываний для того, чтобы повысить точность предсказания ракового заболевания.

Для системы используется база данных LUNA16 [1]. LUNA16 – анонимизированная база данных снимков КТ с возможным раком. Все снимки из данной базы были помечены врачами вручную.

Цель работы – разработать систему, которая будет по снимку КТ распознавать и показывать положение новообразования в легких. Результат работы должен быть визуализирован в 2D. Таким образом, конечным продуктом будет являться ПО, которое должно сегментировать, находить расположение рака и визуализировать результат. Ориентировочная цель – получить результат с точностью не менее 70 %.

Система помощи принятия решения была написана на языке Python с использованием библиотек: numpy – является фундаментальным пакетом для научных вычислений на языке Python; SimpleITK – это набор инструментов для анализа изображений с большим количеством компонентов, поддерживающих общие операции фильтрации, сегментации и регистрации изображений. Он построен на основе инструментария сегментации и регистрации Insight ITK с целью предоставления упрощенного интерфейса для ITK. Сам SimpleITK написан на языке C++, но досту-

пен для большого количества языков программирования; openCV – это библиотека программного обеспечения для компьютерного зрения и машинного обучения с открытым исходным кодом; tensorflow – это библиотека с открытым исходным кодом для численных вычислений и крупномасштабного машинного обучения; pandas – библиотека для обработки и анализа данных; scikit-learn – это библиотека на Python, которая предоставляет множество неконтролируемых и контролируемых алгоритмов обучения [2].

На входе нейросети – объемное изображение, полученное с помощью компьютерной томографии.

На первом этапе происходит сегментация легких, то есть нахождение зоны интереса, чтобы не искать новообразования в костных и жировых тканях человека. После сегментации алгоритм отбрасывает те воксели, которые не вошли в маску интереса, чтобы не производить над ними вычислений.

Следующий этап – поиск раковых узелков на объемных сегментированных легких. По изображению передвигается окно размером 64×64. Если некоторые условия выполняются, то классификатор проверяет данную область. Классификацию выполняет сверточная нейронная сеть.

Когда на вход сети попадает изображение 64×64, нейронная сеть определяет «рак» и «не рак».

Сегментация делится на этапы: сглаживание; бинаризация; морфологическое замыкание; заполнение однородных областей; морфологическое размыкание.

На рисунке 1 слева представлен исходный срез легких, а справа отброшены те пиксели, которые не попали в маску области интереса.

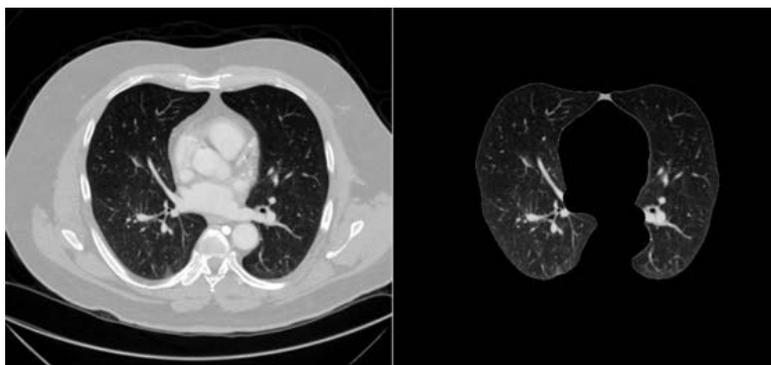


Рис. 1. Срез легких: 1 – исходный срез; 2 – срез под маской области интереса

Причина сегментации – поиск маски интереса для уменьшения вычислений и ускорения процесса обучения и распознавания.

Для одного z-среза алгоритм сегментации строится следующим образом:

1. Сглаживание: используется фильтр Гаусса с $r = 1, \sigma = 1$.

2. Бинаризация.

Получаем бинаризованное изображение легкого (рис. 2).

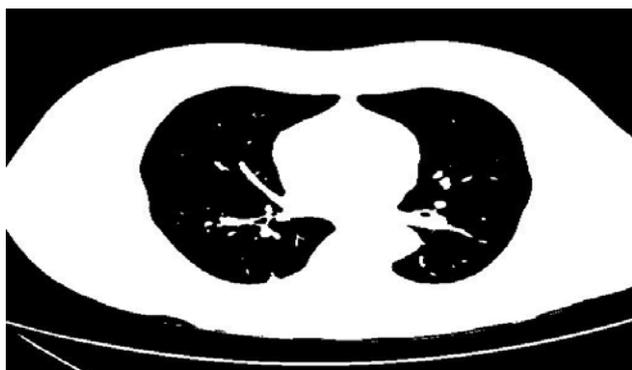


Рис. 2. Бинаризованное изображение легкого

3. Морфологическое замыкание $R=2$ (рис. 3).

Сложно подобрать идеальный радиус. Этот шаг делается для того, чтобы легкие не подверглись заливке на следующем этапе. Для этого мы в больших белых зонах вокруг легких должны убрать как можно больше черных пятен. Если использовать радиус больший чем нужно, то маска будет неточной.

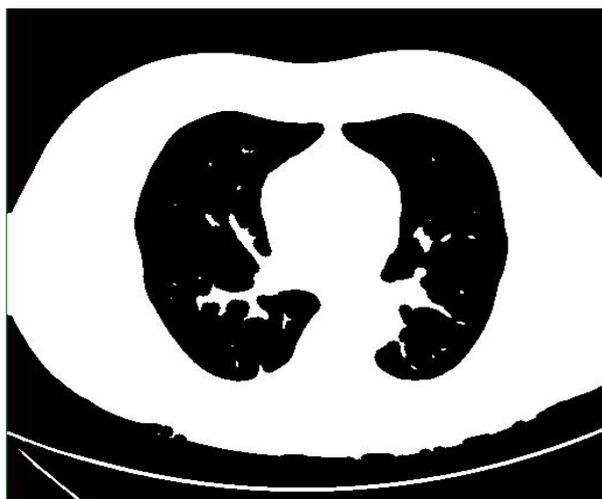


Рис. 3. Бинаризованное изображение легкого после морфологического замыкания

4. Заполнение однородных областей (рис. 4).

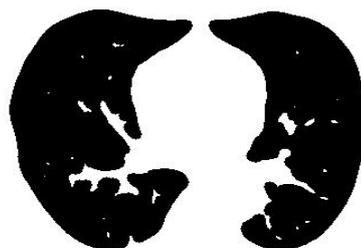


Рис. 4. Изображение после заполнения однородных областей

5. Морфологическое размыкание: производится для получения самой маски (рис. 5)



Рис. 5. Маска сегментации

Опишем принцип построения слоев сверточной нейронной сети (CNN). Сверточный слой – основная часть нейронной сети, где у каждого канала есть свой фильтр, ядро которого обрабатывает предыдущий слой по фрагментам, при этом коэффициенты веса ядра свертки каждого канала определяются на этапе обучения.

Пусть у нас есть d фильтров и $D = \{D_1, D_2, \dots, D_d\}$ – размерности фильтров. Пусть $E = \{E_1, E_2, \dots, E_d\}$ – размерности выхода предыдущего слоя ($\forall i: D_i \leq E_i$), C – количество каналов. $A \in R^{E_1 \times \dots \times E_d \times C}$ – выходные нейроны предыдущего слоя, $W \in R^{D_1 \times \dots \times D_d \times C}$ – фильтр, а $b \in R$ – биас (аддитивная компонента) ядра W . Тогда значение сверточного слоя Z_p для точки $p = (p_1, \dots, p_d)$ следующее:

$$Z_p = b + \sum_{c=0}^{C-1} \left(\sum_{d_1=0}^{D_1-1} \dots \sum_{d_d=0}^{D_d-1} A_{p_1+d_1, \dots, p_d+d_d, c} \cdot W_{d_1, \dots, d_d, c} \right). \quad (1)$$

Если A^l – это выход на слое под номером l , σ – соответственно функция активации для текущего слоя, тогда получается:

$$A^l = \sigma(Z^l). \quad (2)$$

«Слой пуллинга представляет собой нелинейное уплотнение карты признаков, при этом группа пикселей (обычно размера 2×2) уплотняется до одного пикселя, проходя нелинейное преобразование. Наиболее употребительна при этом функция максимума» [3].

Слой, в котором нейроны обладают соединениями со всеми активациями в предыдущем слое, называется полносвязным (рис. 6). Их активации вычисляются умножением матриц со смещением. Формула вычисления значений слоя:

$$Z_j^l = \sum_i Z_j^{l-1} \cdot W_{ij}^l + b_j^l. \quad (3)$$

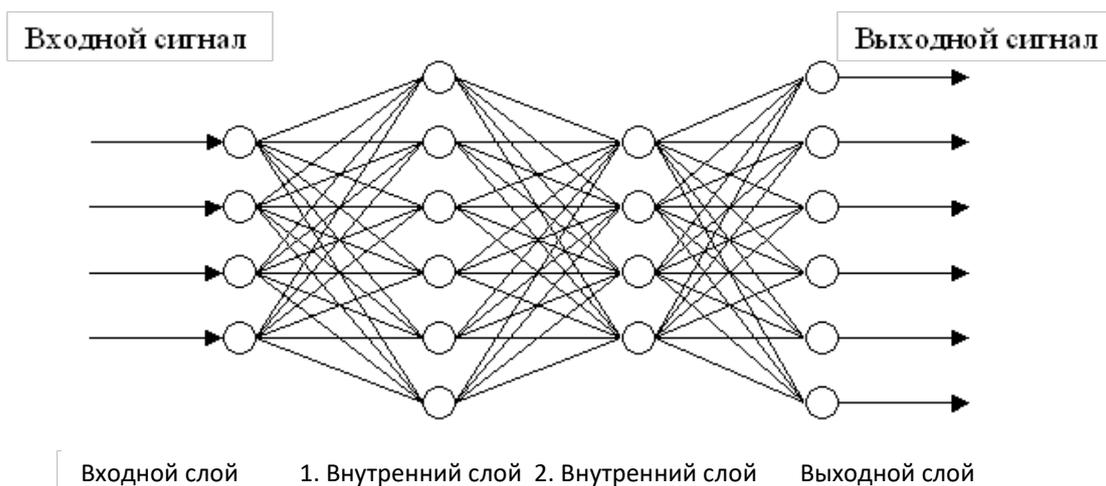


Рис. 6. Полносвязные слои (внутренние слои и выходной слой)

«Функция активации нейрона определяет выходной сигнал, который определяется входным сигналом или набором входных сигналов. Стандартная компьютерная микросхема может рассматриваться как цифровая сеть функций активации, которые могут принимать значения «ON» (1) или «OFF» (0) в зависимости от входа» [4]. Приведем несколько примеров:

$$\text{relu}(x) = \max(0, x), \quad (4)$$

$$\text{lrelu}(x) = \max(\alpha x, x), \alpha < 1, \quad (5)$$

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}. \quad (6)$$

Также существуют функции активации, которые применяют не к одному значению слоя, а ко всем. «Функция Softmax применяется в машинном обучении для задач классификации, когда количество возможных классов больше двух (для двух классов используется логистическая функция)» [5].

$$\text{softmax}(x)_k = \frac{e^{x_k}}{\sum_j e^{x_j}}, \quad (7)$$

Обучение нейронной сети – это настройка ее весовых параметров ядра. В данной системе обучение происходит с учителем, так как результаты выборки известны.

«Обучение – это процесс, в котором свободные параметры нейронной сети настраиваются посредством моделирования среды, в которую эта сеть встроена» [6]. В качестве функции ошибки для нейронных сетей обычно используется среднеквадратичная ошибка (MSQE):

$$E = \frac{1}{2} \sum_j (y_j - c_j)^2, \quad (8)$$

где y_j – выход сети номер j , c_j – правильный ответ для выхода j .

Для сетей softmax обычно используют среднюю кросс-энтропию по всем учебным данным.

$$E = \frac{1}{n} \sum_i (-\log(p_{y_i})). \quad (9)$$

Таким образом, обучение нейронной сети сводится к задаче минимизации функции ошибок E .

Для решения задачи будет использован градиентный метод оптимизации. «Градиентный спуск – метод нахождения локального экстремума (минимума или максимума) функции с помощью движения вдоль градиента. Для минимизации функции в направлении градиента используются методы одномерной оптимизации, например метод золотого сечения. Также можно искать не наилучшую точку в направлении градиента, а какую-либо лучше текущей» [7]. Исходя из этого предположения, схема обучения строится следующим образом, при задании критериев остановки обучения:

1. Инициализируются веса W , а также биасы b .
2. Вычисляется значение функции ошибок E .
3. Если выполнены критерии остановки, то алгоритм прекращает работу. Критериями остановки обучения могут быть следующие случаи: если функция ошибок E меньше заданного числа или в ходе работы сделано заданное количество шагов.
4. Вычисляем ∇E .
5. Вычисляем изменение параметров ΔW и Δb :
6. $\Delta W = \eta \nabla_w E$, (10)
7. $\Delta b = \eta \nabla_b E$. (11)
8. Корректируем параметры: $W = W - \Delta W$, $b = b - \Delta b$.
9. Повторить с п. 2.

Пользуясь методом обратного распространения ошибки, решаем задачу методом градиентного спуска, вычисляя для этого градиент. «Минимизация любой функции означает поиск самой глубокой впадины в этой функции. Функция используется, чтобы контролировать ошибку в прогнозах модели машинного обучения. Поиск минимума означает получение наименьшей возможной ошибки или повышение точности модели. Мы увеличиваем точность, перебирая набор учебных данных при настройке параметров нашей модели (весов и смещений)» [8]. Каким образом происходит вычисление градиента для всех видов слоев в CNN?

Исходя из того, что CNN заканчивается полносвязными слоями, определим следующие формулы. Вначале вычислим ошибку δ для выходного слоя сети:

$$\delta_j^l = \frac{\partial E}{\partial A_j^l} \sigma'(Z_j^l), \quad (12)$$

где σ – функция активации этого слоя, A – активация слоя.

Соответственно для скрытого полносвязного слоя будет:

$$\delta_j^l = \sum_k \frac{\partial Z_k^{l+1}}{\partial Z_j^l} \delta_k^{l+1}, \quad (13)$$

где Z – значения слоя до активации.

На следующем шаге происходит вычисление градиентов для весов и смещений:

$$\frac{\partial E}{\partial b_j^l} = \delta_j^l, \quad (14)$$

$$\frac{\partial E}{\partial W_{jk}^l} = A^{l-1} \delta_j^l. \quad (15)$$

LUNA16 – это список кандидатов на раковые узелки. Среди них есть как не раковые, так и раковые узлы. Из-за огромного количества кандидатов пришлось поменять соотношение раковых кандидатов, так как был сильный дисбаланс между раковыми и нераковыми кандидатами.

Выборка делилась на 3 набора: обучающую, для валидации, для тестирования. Соотношение наборов по объему 65:15:20. В итоге соотношение положительных и отрицательных кандидатов было 1:5.

По результатам контрольной проверки и результатам тестирования, можно сказать, что нейронная сеть работает корректно и находит расположение новообразований. Нейронная сеть находит более 82 % новообразований и тратит на это не более 30 секунд. Чем больше обучаемая выборка, тем алгоритм более точный.

Перспективы дальнейшей работы: разработать алгоритм классификации новообразований; перейти от 2D-среза к 3D; уменьшить время обучения и работы программы; увеличить точность выявления новообразований.

Литература

1. Lung Nodule Analysis (LUNA16) : сайт. – URL: <https://luna16.grand-challenge.org/> (дата обращения: 06.05.20). – Текст : Electronic.

2. Уткин, Д. Е. Разработка системы сегментации и обнаружения новообразований для диагностики рака легкого / Д. Е. Уткин // XIII Ежегодная научная сессия аспирантов и молодых ученых. материалы межрегиональной научной конференции : в 2 томах. – 2019. – С. 221–225.

3. Сверточная нейронная сеть : сайт – URL: <https://ru.wikipedia.org/wiki/> (дата обращения: 06.05.20). – Текст : электронный.

4. Функция активации : сайт – URL: <https://ru.wikipedia.org/wiki/> (дата обращения 06.05.20). – Текст : электронный.

5. Softmax : сайт – URL: <https://ru.wikipedia.org/wiki/Softmax> (дата обращения: 06.05.20). – Текст : электронный.

6. Обучение нейронной сети : сайт – URL: <http://www.aiportal.ru/articles/neural-networks/learning-neunet.html#:~:text=Обучение%20-%20это%20процесс%2C%20в%20котором,определяется%20способом%20подстройки%20этих%20параметров> (дата обращения: 06.05.20). – Текст : электронный.

7. Градиентный спуск : сайт – URL: <https://ru.wikipedia.org/wiki/> (дата обращения: 06.05.20). – Текст : электронный.

8. Neurohive : сайт – URL: <https://neurohive.io/ru/osnovy-data-science/gradient-descent/> (дата обращения 6.05.20). – Текст : Electronic.

A.A. Sukonshchikov, D.E. Utkin
Vologda State University

DSS DEVELOPMENT FOR LUNG CANCER DIAGNOSIS

The article deals with the development of a decision support system for diagnosing lung cancer based on computed tomography images. DSS is built on the basis of a convolutional neural network.

DSS, convolutional neural network, neoplasm.