



ДИАГНОСТИКА ПНЕВМОНИИ ПО РЕНТГЕНОВСКИМ СНИМКАМ С ПОМОЩЬЮ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Макиров Хаджимамед Байрамович

Университет инженерных технологий Туркменистана имени Огуз хана

г. Ашхабад Туркменистан

Аннотация. Поднимается проблема диагностики заболеваний, для решения которой предлагается использовать искусственный интеллект или методы машинного обучения, а именно разработать нейронную сеть (**Нейронные сети — это подразделение искусственного интеллекта, которое моделирует работу человеческого мозга. Они состоят из слоев искусственных нейронов, которые работают вместе для анализа данных**) для диагностики пневмонии по рентгеновским снимкам. Для распознавания пневмонии на рентгеновских снимках применялась сверточная нейронная сеть. Разработка программы диагностики пневмонии проводилась на языке Python. В программе использовались функции из трех библиотек: matplotlib, keras_metrics и Keras. В данной работе была применена аугментация данных и метод регуляризации dropout для преодоления эффекта переобучения нейронной сети. Результаты. Обучение нейронной сети проводилось в течение 20 эпох и время ее обучения составило 35 мин. В результате тестирования модели сверточной нейронной сети, в выходном слое которой для активации нейронов использовалась сигмоидальная функция активации, а в качестве метода оптимизации – алгоритм RMSProp, были получены наилучшие значения показателей точности и полноты: precision – 89 %, accuracy – 82 % и recall – 93 %. Выводы. При тестировании нейронная сеть показала точность распознавания пневмонии, равную 89 %. И этот результат превышает значения показателя precision, которые были получены в известных работах.

Ключевые слова: искусственный интеллект, нейронные сети, сверточные нейронные сети, глубокое обучение, машинное обучение, диагностика пневмонии, распознавание изображений

Введение

В связи с информатизацией общества в современном мире очень активно развивается сфера информационных технологий. IT-технологии применяются во всех областях жизнедеятельности человека. Особенно сейчас популярны искусственный интеллект и машинное обучение. Они являются инструментом для решения огромного количества задач. Например, методы искусственного интеллекта и машинного обучения можно применять для диагностики заболеваний. В настоящее время создается очень много медицинских систем на основе машинного обучения, в том числе и в России, которые позволят облегчить работу врачей.

Главными проблемами применения нейронных сетей являются повышение точности и обобщающей способности сетей, т.е. способности показывать высокие результаты не только на данных, на которых обучалась сеть, но и на новых данных. Целью исследования

является разработка сверточной нейронной сети для диагностики пневмонии по рентгеновским снимкам, обладающей высокой точностью и обобщающей способностью.

В основном архитектура сверточных нейронных сетей включает в себя слои трех основных видов [4–6]: сверточные слои, слои субдискретизации (пуллинга) и полносвязные слои. Но иногда сверточные нейронные сети могут состоять из слоев четырех видов, к уже перечисленным видам добавляются слои прореживания. Сверточные слои работают с тензорами. Тензоры в сверточных сетях – это трехмерные массивы матриц чисел. Рентгеновское изображение на входе сети (входная карта) имеет размеры 150×150 пикселей. Пиксель описывается целым числом от 0 до 255. Входные значения пикселей нормализуются в диапазон от 0 до 1. Сверточный слой представляет собой набор ядер (фильтров). Ядра являются небольшими матрицами весовых коэффициентов, настраиваемых в процессе обучения сети. Ядро с заданным шагом обходит соответствующую карту предыдущего сверточного слоя, поэлементно умножая и суммируя элементы карты и ядра. Результат записывается в карту текущего слоя. Количество карт признаков слоя называется глубиной слоя. За сверточным слоем располагается слой субдискретизации (пуллинга). Чаще всего применяется MaxPooling. Для этого карта признаков разделяется на ячейки размером ядра субдискретизирующего слоя (обычно, 2×2), из каждой ячейки выбирается максимальное значение. К полученным картам применяется функция активации нейронной сети, как правило ReLU. В процессе преобразования количество карт признаков увеличивается, а их размер уменьшается. В результате формируется вектор признаков, который обрабатывается полносвязными слоями, решающими задачу классификации.

В процессе обучения нейронная сеть может переобучиться. Это означает, что сеть хорошо распознает примеры из обучающей выборки, но плохо определяет примеры, на которых она не обучалась (сеть теряет обобщающую способность). В сверточных сетях для преодоления переобучения применяют разновидность регуляризации – прореживание сети (dropout). При использовании dropout при обучении сети связи между слоями удаляются с некоторой заданной вероятностью удаления, обычно 0,5. При обученной сети связи не удаляются, но соответствующие выходы умножаются на единицу минус вероятность удаления.

Материал и методика

В данной работе для обучения, тестирования и проверки сверточной нейронной сети использовалась база рентгеновских снимков Chest X-Ray Images (Pneumonia). Данная база данных размещена на сайте Kaggle [7]. Рентгеновские снимки в базе Chest X-Ray Images (Pneumonia) представлены в виде изображений разного размера в формате JPEG, поэтому, чтобы привести эти снимки к одному размеру, использовался метод `flow_from_directory`. Медицинские изображения из этой базы данных разделены на три набора. Обучающий набор данных используется для настройки нейронной сети, тестовое множество нужно для того, чтобы оценить качество обучения модели нейронной сети, а проверочная выборка – для обнаружения эффекта переобучения (табл. 1).

Структура базы рентгеновских снимков

Название набора	Количество снимков NORMAL	Количество снимков PNEUMONIA	Всего
Обучающий	1340	3875	5215
Тестовый	234	390	624
Контрольный	8	8	16
Всего	1582	4273	5855

Пример рентгеновских снимков из базы Chest X-Ray Images (Pneumonia) приведен на рис. 1.

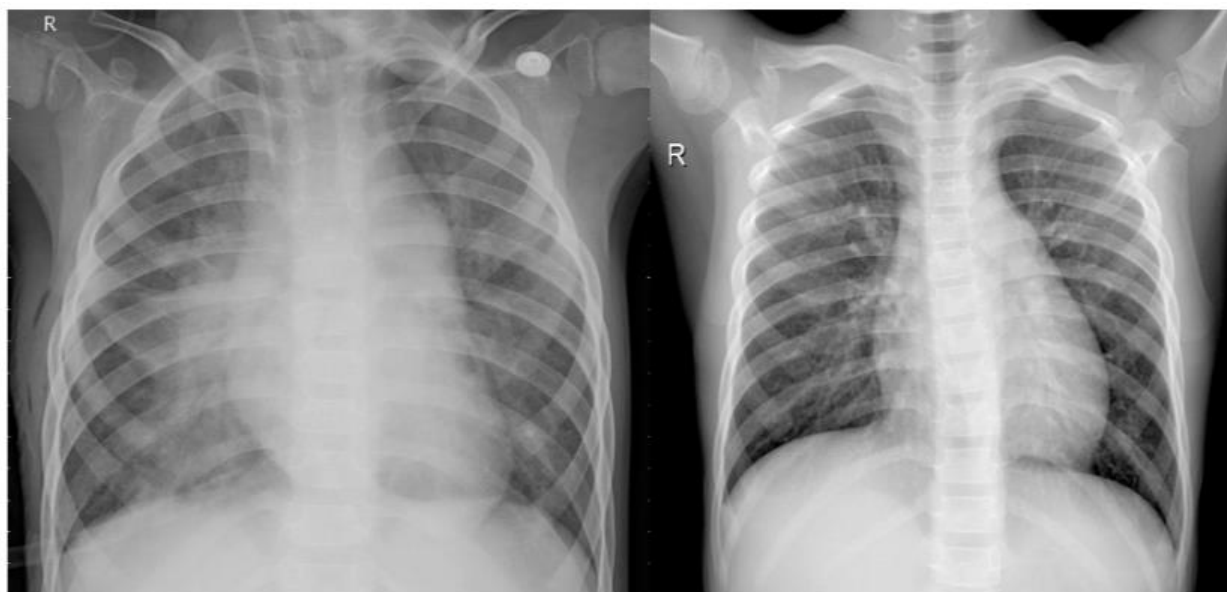


Рис. 1. Пример изображений из обучающего набора

С целью разработки программы для выявления пневмонии на рентгеновских снимках использовался язык программирования Python, нейросетевая библиотека Keras версии 2.3.1 [8], а также библиотеки matplotlib и keras_metrics. Функции из библиотеки Keras применялись для создания сверточной нейронной сети. Из библиотеки keras_metrics использовались такие метрики, как binary_precision и binary_recall. Функция binary_recall служит для вычисления процента положительных примеров, правильно определенных классификатором, а метрика binary_precision – для вычисления точности предсказания положительного исхода. Библиотека matplotlib использовалась для вывода графиков точности и потерь.

Сверточная нейронная сеть состоит из одиннадцати слоев, это значение было выбрано исходя из результатов проведенной серии экспериментов. Структура модели этой нейронной сети представлена в табл. 2.

Структура сверточной нейронной сети

Номер слоя	Тип слоя нейронной сети	Параметры слоя нейронной сети
1	Сверточный слой	1. Функция активации – ReLU. 2. Глубина выходных карт признаков – 32. 3. Размер ядра – 3×3
2	Субдискретизирующий слой	1. Операция – MaxPooling. 2. Размер ядра – 2×2
3	Сверточный слой	1. Функция активации – ReLU. 2. Глубина выходных карт признаков – 64. 3. Размер ядра свертки – 3×3
4	Субдискретизирующий слой	1. Операция – MaxPooling. 2. Размер ядра – 2×2
5	Сверточный слой	1. Функция активации – ReLU. 2. Глубина выходных карт признаков – 128. 3. Размер ядра свертки – 3×3
6	Субдискретизирующий слой	1. Операция – MaxPooling. 2. Размер ядра – 2×2
7	Сверточный слой	1. Функция активации – ReLU. 2. Глубина выходных карт признаков – 128. 3. Размер ядра свертки – 3×3
8	Субдискретизирующий слой	1. Операция – MaxPooling. 2. Размер ядра – 2×2
9	Слой dropout	Вероятность отключения нейрона предыдущего слоя – 0,5
10	Полносвязный слой	1. Функция активации – ReLU. 2. Количество нейронов в слое – 512
11	Полносвязный слой	1. Функция активации – sigmoid. 2. Количество нейронов в слое – 1

В одиннадцатом (выходном) слое в качестве функции активации нейрона применяется сигмоидальная функция активации (антисимметричная логистическая функция):

$$f(s) = \frac{1 - e^{-s}}{1 + e^{-s}}.$$

Функция активации для выходного слоя была подобрана экспериментально.

В качестве функции потерь применяется функция бинарной перекрестной энтропии (binary_crossentropy)

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i))$$

Эта функция была выбрана, так как она лучше, чем другие функции потерь, подходит для решения задач бинарной классификации. А в данной работе как раз решается такая задача (нужно определить, болен пациент или здоров).

Для настройки сверточной нейронной сети был выбран алгоритм оптимизации первого порядка, основанный на применении коэффициента ослабления Root Mean Square Propagation (RMSProp).

При тестировании сверточной нейронной сети были построены графики точности и потерь (рис. 2), на которых видно, при росте точности распознавания растут ошибки на проверочном множестве. Это говорит о том, что нейронная сеть переобучилась.

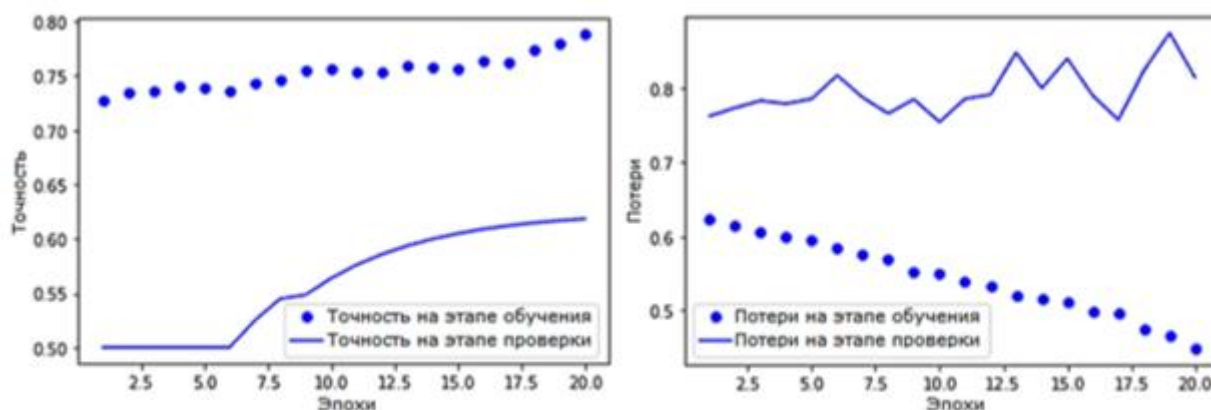


Рис. 2. Графики точности и потерь

Для того, чтобы преодолеть эффект переобучения нейронной сети, было увеличено количество входных данных и применен метод регуляризации dropout [8]. Увеличение количества исходных изображений было проведено благодаря применению аугментации данных [9]. Аугментация данных – это метод, который использует различные преобразования для увеличения размера выборки. В случае с изображениями в качестве таких преобразований применяются поворот, сдвиг, сжатие и растяжение вдоль осей, масштабирование и другие преобразования. В данной работе использовались эти четыре преобразования. А для проведения этих преобразований применялась функция ImageDataGenerator.

Результаты

При тестировании сверточной нейронной сети были рассчитаны три показателя: precision, accuracy и recall. Показатель precision (точность) позволяет увидеть, сколько процентов рентгеновских снимков из тестовой выборки, которые были отнесены к классу PNEUMONIA, действительно ему принадлежат. Показатель accuracy (точность) определяется как число правильно классифицированных снимков, отнесенное к их общему числу. С помощью третьего показателя (полноты) можно определить процент людей, которые больны пневмонией и которые были распознаны правильно.

Было проведено четыре эксперимента по подбору оптимальной функции активации для выходного слоя нейронной сети и для выбора алгоритма оптимизации. Результаты этих экспериментов представлены в табл. 3.

Результаты экспериментов

Параметры		Результаты эксперимента			
Функция активации	Метод оптимизации	Время обучения нейронной сети, мин	Accuracy	Precision	Recall
1. Softmax	Adam (adaptive moment estimation)	40	62,35	62,4	69,4
2. Sigmoid	Adam	42	82,12	89,7	82,35
3. Sigmoid	RMSProp	35	82,03	89,9	93,6
4. Softmax	RMSProp	40	62,6	62,7	62,5

Исходя из результатов этих экспериментов был сделан вывод, что в качестве функции активации выходного слоя сверточной нейронной сети лучше использовать функцию sigmoid, а в качестве метода оптимизации алгоритм RMSProp. Лучшие значения показателей точности, полученные в результате обучения нейронной сети с выбранными параметрами, равны: precision – 89 %, accuracy – 82 % и recall – 93 %, т.е. диагнозы 89 % людей из тестовой выборки были определены правильно и данный результат превышает показатели, которые были получены в работах.

Заключение

В результате проведенного исследования была разработана программа диагностики пневмонии при помощи рентгеновских снимков. Был проведен ряд экспериментов по подбору параметров (количество слоев, функция активации выходного слоя сети и алгоритм оптимизации) для обучения нейронной сети. Обучение сверточной нейронной сети проводилось на расширенном, с помощью аугментации данных, обучающем наборе рентгеновских снимков. Время обучения этой сети составило 35 мин. А количество эпох, в течение которых нейронная сеть обучалась, равно 20. Наилучшие результаты распознавания нейронная сеть показала при проведении третьего эксперимента. В результате тестирования третьей модели была получена точность распознавания пневмонии – 89 %. И этот результат превышает показатели, полученные в известных работах.