

ДВА ПОДХОДА К РАСПОЗНАВАНИЮ ПНЕВМОНИИ ПО РЕНТГЕНОВСКИМ СНИМКАМ: ОПИСАНИЕ, РЕАЛИЗАЦИЯ И СРАВНЕНИЕ

Печников А. А.¹,

*(Карельский научный центр Российской академии наук,
Петрозаводск)*

Богданов Н. А.²

*(Санкт-Петербургский государственный университет,
Санкт-Петербург)*

Рассматриваются два подхода к задаче классификации рентгеновских снимков грудной клетки для диагностики пневмонии. Первый, на сегодня широко распространенный подход, основан на применении нейронных сетей. Вторым подходом используется теоретическое понятие колмогоровской сложности, на основе которого вводится нормализованное расстояние сжатия, позволяющее определять расстояния между объектами различной природы, включая изображения. Дается изложение обоих подходов, описание основанных на них алгоритмов и сведения о программных реализациях. Рентгеновские снимки для экспериментов взяты из базы социальной сети по обработке данных и машинному обучению Kaggle. Проведенная серия экспериментов позволяет подобрать значения параметров для успешного решения поставленной задачи. Высокие значения метрик качества классификации в обоих случаях убедительно свидетельствуют о надежном разграничении рентгеновских снимков грудной клетки у здоровых людей от больных пневмонией. Преимущества первого подхода проявляются при больших массивах данных обучающего множества. Вторым подходом позволяет решать эту же задачу при наличии малого числа классифицированных снимков, когда первый подход не работает. Полученные результаты дают хорошие перспективы разработки чувствительных и надежных методов диагностики пневмонии, сочетающих оба этих подхода.

Ключевые слова: классификация изображений, сверточная нейронная сеть, нормализованное расстояние сжатия, обработка рентгеновских снимков, пневмония.

¹ Андрей Анатольевич Печников, д.т.н., доцент (pechnikov@krc.karelia.ru).

² Николай Андреевич Богданов, выпускник факультета прикладной математики-процессов управления СПбГУ 2021 года (bogdanov699@gmail.com).

1. Введение

Пневмония является одной из форм острой респираторной инфекции, воздействующей на легкие. Симптомы вирусной и бактериальной пневмонии схожи [3]. Оперативная диагностика заболевания затруднена, поскольку методы на основе полимеразной цепной реакции [10], обладающие наибольшей достоверностью, дают результат через несколько часов при обеспечении высоких требований к соблюдению технологии анализа и профессионализму персонала. Для распознавания пневмонии применяется компьютерная томография [8], однако компьютерные томографы пока есть далеко не во всех лечебных учреждениях. Рентгенологическая диагностика обладает меньшей разрешающей способностью и наличием шумов, но является более распространенной из-за большей доступности рентгеновских аппаратов [5]. Поэтому актуальна разработка быстрых и надежных компьютерных программ, помогающих врачам оперативно обнаруживать заболевание по рентгеновским снимкам грудной клетки.

В данной работе для решения задачи распознавания по рентгеновским снимкам такого заболевания как пневмония рассматриваются два подхода. Первый из них, активно изучаемый в последнее время, основан на применении нейронных сетей. В работах [1, 4, 12, 14, 22] показана перспективность данного подхода. При этом отмечается, что точность классификации сильно зависит от количества представленных образцов, и уделяется особое внимание предварительной обработке изображений.

Второй подход, предлагаемый авторами, использует нормализованное расстояние сжатия [13, 19], являющееся практической реализацией такого теоретического понятия как колмогоровская сложность [2, 6, 18]. Сама идея классификации изображений с использованием колмогоровской сложности не нова (например, [21]). Представляется важным сравнить распространенный сегодня подход, основанный на нейронных сетях, и подход, базирующийся на колмогоровской сложности, с целью выявления достоинств и недостатков каждого из них, что

в дальнейшем можно будет учитывать при разработке программ для диагностики пневмонии.

В следующем разделе описано используемое в работе множество изображений, источник, из которого оно было получено и основные этапы предварительной обработки этих изображений.

Далее в двух разделах описаны основные теоретические аспекты каждого из алгоритмов, реализующих два подхода, и наиболее важные моменты их программной реализации. Затем следует раздел с описанием и результатами проведенных экспериментов, сравнением и основными преимуществами каждого из подходов. На основе этого сделаны выводы и предложены практические рекомендации.

2. Множество рентгеновских снимков и предварительная обработка изображений

Рентгеновские снимки для данной работы взяты из базы социальной сети по обработке данных и машинному обучению Kaggle [11]. Здесь в формате JPEG хранятся в виде отдельных файлов 5856 рентгеновских снимков, из которых 1583 принадлежат здоровым людям, а 4273 – людям, у которых выявлена пневмония. В базе Kaggle снимки классифицированы и им присвоены метки классов Normal и Pneumonia.

Примеры рентгеновских снимков, взятые из [11], приведены на рис. 1.

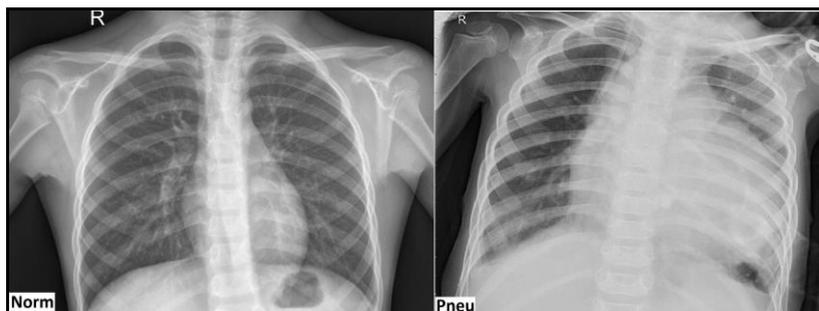


Рис. 1. Примеры рентгеновских снимков

Левый снимок относится к человеку, не болеющему пневмонией, а правый – соответственно, к больному пневмонией.

Реализация обоих алгоритмов распознавания выполнена на языке Python в среде разработки Pycharm с использованием открытой нейросетевой библиотеки Keras, поэтому далее мы будем использовать соответствующую терминологию.

Рентгенограммы легких являются черно-белыми растровыми изображениями, причем в Kaggle они имеют разный размер. Поэтому необходима предварительная обработка изображений в некоторую матричную (векторную) форму, используемую в обоих наших методах. Для этого изображения разбиваются на пиксели. И в случае нейронной сети, и в случае колмогоровской сложности количество пикселей варьируется по ширине и высоте в зависимости от проводимых экспериментов. Изображение с минимальным количеством пикселей имеет 16384 пикселей (128 по ширине и 128 по высоте), а с максимальным – 1048576 пикселя (1024 по ширине и 1024 по высоте). Цвет каждого пикселя представлен целым числом от 0 до 255. По шкале «оттенки серого» (grayscale) значение 0 – это чёрный цвет, а значение 255 – белый. Преобразование изображений .jpeg в представление в виде вектора выполняется с помощью пакета `opencv-python` [20], часто используемого для таких целей.

Для работы с программой, реализующей подход, основанный на нейронных сетях, необходимо иметь три непересекающихся множества данных: обучающее, проверочное и тестовое. Обучающее множество (обозначим его T) предназначено собственно для обучения сети. Проверочное (или валидационное) множество V служит для настройки параметров классификатора. В нашем случае мы используем его для контроля за ходом обучения. На тестовом множестве осуществляется проверка построенной модели; обозначим его D . Каждое множество, в свою очередь, разбивается на два непересекающихся подмножества: снимки здоровых людей и снимки больных пневмонией.

В процессе обучения производилась так называемая *real-time-аугментация* данных с использованием класса `ImageDataGenerator` [23]. Суть данного метода в том, что перед очередной эпохой обучения каждый снимок исходного множества T изменяется

посредством серии трансформаций, таких как поворот/сдвиг на небольшой случайный угол/смещение над снимками и умножение масштаба изображения на небольшой случайный множитель (zoom in / zoom out). В результате описанных действий нейронная сеть в процессе обучения не встречает один и тот же снимок из T дважды, что значительно улучшает обобщающую способность сети.

Для программы, основанной на колмогоровской сложности, требуется два множества рентгеновских снимков: множество снимков Y , которое мы будем называть контрольным, и опять-таки тестовое множество, которое в этом случае обозначим X . Соответственно, множества Y и X также разбиваются на два непесекающихся подмножества.

Мощности множеств T , V , Y , D и X и их подмножеств варьируются в зависимости от цели обучения и/или эксперимента.

3. Подход, основанный на сверточных нейронных сетях

Сверточная нейронная сеть (СНС) как специальная архитектура нейронных сетей впервые была описана в работе [17] и на сегодняшний день являются одной из самых популярных и эффективных для решения задач распознавания изображений.

Работа СНС представляет собой поэтапный переход от конкретных особенностей изображения к более абстрактным деталям, затем к ещё более абстрактным деталям и так далее вплоть до выделения понятий высокого уровня. Эти абстрактные признаки формируют так называемые карты признаков (фактически матрицы). При этом сеть самонастраивается и вырабатывает необходимую иерархию абстрактных признаков (последовательности карт признаков), отбрасывая маловажные детали и выделяя существенные.

В общем виде СНС состоит из большого количества слоёв. Исходное изображение подается на входной слой. Далее сигнал проходит последовательность свёрточных слоёв, в которых чередуется собственно свёртка и субдискретизация.

Формально свёртка представляет собой специфическую операцию умножения исходной матрицы на матрицу меньшего размера, называемую ядром свертки. Ядро как бы «двигается» по исходной матрице слева направо и сверху вниз, в каждом положении считается скалярное произведение фильтра и той части исходной матрицы, на которую он наложен. Получившееся число запоминается как соответствующий элемент результата. Сверточный слой представляет собой применение операции свертки к выходам с предыдущего слоя, а веса ядра свертки являются обучаемыми параметрами.

Размерность матрицы-ядра и шаг сдвига подбираются в зависимости от ряда условий [9]. По аналогии с работой [4] мы использовали слои с 32, 64, 128 и 256 ядрами, одинаковые размеры ядра (3×3) и шаг сдвига, равный 1.

Скалярный результат каждой свёртки попадает на функцию активации (функция срабатывания, передаточная функция), представляющую собой некую нелинейную функцию, которая обычно встроена в слой свёртки. Как правило, функция активации «зажимает» полученный результат в требуемые ограничения. Например, функция ReLU (Rectified Linear Unit) возвращает 0, если аргумент отрицательный, а в случае положительного аргумента функция возвращает само число. Мы используем функцию ReLU как и в [4], поскольку здесь показаны хорошие результаты нейронной сети для классификации рентгенографических изображений больных пневмонией и COVID-19.

Используемый нами метод регуляризации Dropout заключается в том, что в одном или нескольких слоях нейронной сети каждый нейрон может быть отключен (выходной сигнал этого нейрона равен 0 вне зависимости от входных сигналов) с заданной вероятностью p . Состояние нейрона (включен/отключен) определяется на каждой итерации процесса обучения. Это позволяет имитировать большое количество сетей с разной структурой сети и, в свою очередь, делать узлы в сети, как правило, более устойчивыми к входам для предотвращения ее переобучения (overfitting) [24].

Субдискретизация (или пулинг, от английского “pooling”) предназначена для уменьшения размерности матрицы. Исход-

ная матрица делится на блоки и для каждого из них вычисляется некоторая функция. Например, используется функция максимума, когда блок заменяется на единственное значение максимального элемента этого блока.

Чередование слоёв позволяет составлять очередные карты признаков из предыдущих карт признаков, на каждом следующем слое карта уменьшается в размере, но увеличивается их количество. Обычно после прохождения нескольких слоёв карта признаков вырождается в вектор или даже скаляр, но таких карт признаков становятся сотни.

На выходе последовательности свёрточных слоёв (свёртки и субдискретизации) получается набор окончательных карт признаков. Он, в свою очередь, подается на вход обычной полносвязной нейронной сети, которая тоже может состоять из нескольких слоёв. Иногда эта конструкция называется общим термином «скрытый слой». После скрытого слоя сигнал подается на выходной слой, на выходах которого формируется результат (отклик сети на входное воздействие).

Для обучения СНС разработан ряд стандартных методов, из которых довольно часто используется метод обратного распространения ошибки. В математическом смысле это итеративный градиентный алгоритм, который используется с целью минимизации ошибки работы нейронной сети и получения желаемого выхода. Для этого строится функция оценки, зависящая от выходных сигналов и требуемых (известных) их значений. К примеру, может использоваться метод наименьших квадратов. Для минимизации функции оценки следует изменять веса после каждого обучающего примера, «двигаясь» в многомерном пространстве весов в сторону, противоположную градиенту, который указывает на правление наибольшего возрастания ошибки.

Для решения задачи распознавания пневмонии по рентгеновским снимкам нами была разработана СНС, имеющая следующую структуру: входной слой, 4 свёрточных слоя, скрытый слой и выходной слой. Структура изображена на рис. 2 с использованием терминологии Python и Keras.

Conv2D(32, (3,3), strides = 1, padding = 'same', activation = 'relu', input_shape = (150,150,1)) BatchNormalization() MaxPool2D((2,2), strides = 2, padding = 'same')	I
Conv2D(64, (3,3), strides = 1, padding = 'same', activation = 'relu') Dropout(0.1) BatchNormalization() MaxPool2D((2,2), strides = 2, padding = 'same')	II
Conv2D(64, (3,3), strides = 1, padding = 'same', activation = 'relu') Dropout(0.1) BatchNormalization() MaxPool2D((2,2), strides = 2, padding = 'same')	III
Conv2D(128, (3,3), strides = 1, padding = 'same', activation = 'relu') Dropout(0.2) BatchNormalization() MaxPool2D((2,2), strides = 2, padding = 'same')	IV
Conv2D(256, (3,3), strides = 1, padding = 'same', activation = 'relu') Dropout(0.2) BatchNormalization() MaxPool2D((2,2), strides = 2, padding = 'same')	V
Flatten() Dense(units = 128, activation = 'relu') Dropout(0.2)	VI
Dense(units = 1, activation = 'sigmoid')	VII

Рис. 2. Архитектура и основные параметры СНС

Первый блок содержит Conv2D – это сверточный 2D слой, который принимает на входе матрицу размерности 150x150 пикселей, а на выход выдает 32 матрицы той же размерности. В качестве функции активации используется функция ReLU. Здесь input_shape – дополнительный аргумент, содержащий размерность входного изображения: первые две компоненты задают высоту и ширину входного изображения, а третья компонента – количество цветовых каналов (1 так как изображение черно-белое). MaxPool2D – это слой субдискретизации. Операция BatchNormalization реализует метод пакетной нормализации, предложенный в [15] и, по устоявшемуся мнению, делающий нейронные сети более быстро обучаемыми и стабильными за счет нормализации входных данных.

Второй блок принимает на входе 32 сформированные матрицы, свертывает, субдискретизирует и нормализует их. На выходе имеем 64 матрицы. Здесь регуляризация Dropout выполняется при заданном значении $p = 0,1$, как и в третьем блоке, а в следующих блоках $p = 0,2$.

Третий блок подает на выход столько же карт признаков, сколько и принимает. Однако за счет слоя субдискретизации дополнительно уменьшается размерность карт признаков, подающихся на выход, что впоследствии влияет на уменьшение размерности вектора, подающегося на вход полносвязному участку сети, в 4 раза.

Четвертый и пятый блоки почти аналогичны третьему, но на выходе имеем соответственно 128 и 256 матриц, и операция Dropout, как уже сказано выше, выполняется с вероятностью 0,2.

Шестой блок реализует собственно полносвязную нейронную сеть («скрытый слой»), в свою очередь, состоящую из нескольких слоев. Слой Flatten преобразует 2D-данные в 1D-данные и в нашем случае формирует из 256 матриц размерности 5×5 вектор из 2560 элементов. Слой Dense на выходе генерирует вектор из 128 элементов с использованием функции ReLU.

Седьмой блок является выходным слоем, формирующим результат. В качестве функции активации используется функция sigmoid, вычисляющая значения сигмоидальной функции от аргумента x :

$$\sigma(x) = 1/(1 + e^{-x})$$

Обучение проводилось с использованием оптимизатора RMSprop [16]. В качестве функции потерь была выбрана бинарная перекрестная энтропия, реализованная в библиотеке Keras как `binary_crossentropy` и имеющая следующий вид:

$$Loss(y_j, y_j^*) = -y_j * \log(y_j^*) - (1 - y_j) * \log(1 - y_j^*)$$

Для n наблюдений $j \in 1, \dots, n$, x_j – результат (выходной сигнал) j -го наблюдения, а y_j – требуемое (известное) значение. Ответ нейронной сети для объекта x_j представляет собой единственное вещественное число y_j^* , равное вероятности принадлежности объекта x_j классу 1, а вероятность принадлежности к классу 2 равно $1 - y_j^*$.

Для оценки качества обучения использовалась метрика ассигасу (достоверность), значение которой равно отношению верно классифицированных объектов к общему количеству объектов.

При обучении нейросетей термин «эпоха» (англ. "epoch") используется для обозначения повторяющегося процесса, который заключается в подаче на вход всех примеров из обучающего множества, обновления весов нейронной сети после каждой итерации и, возможно, оценку качества обучения на валидационном множестве. Эффективность классификации, проводимой СНС, замеряется после каждой эпохи, и результаты этой проверки могут влиять на гиперпараметры обучения (например, коэффициент скорости обучения, оптимальное число эпох обучения и т.д.).

В нашем случае необходимо было провести серию экспериментов для различных обучающих и валидационных множеств и разного количества эпох. В качестве основного примера можно привести обучение СНС, которое проводилось в течение 40 эпох для обучающего множества T , содержащего 6375 обработанных снимков (2500 снимков здоровых людей и 3875 снимков больных пневмонией) с использованием валидационного множества V (200: 100 и 100 снимков соответственно). После каждой обучающей итерации на множестве T проводилась оценка качества на множестве V . Графики изменения функции ассигасу в зависимости от эпохи для обоих множеств показаны на рис. 3.

На рисунке видно, что точность на валидационном множестве достигает максимума на 20-й эпохе обучения, после чего несколько снижается, что может свидетельствовать о «переобучении» сети. Таким образом, имеет смысл остановить процесс обучения после 20-й эпохи.

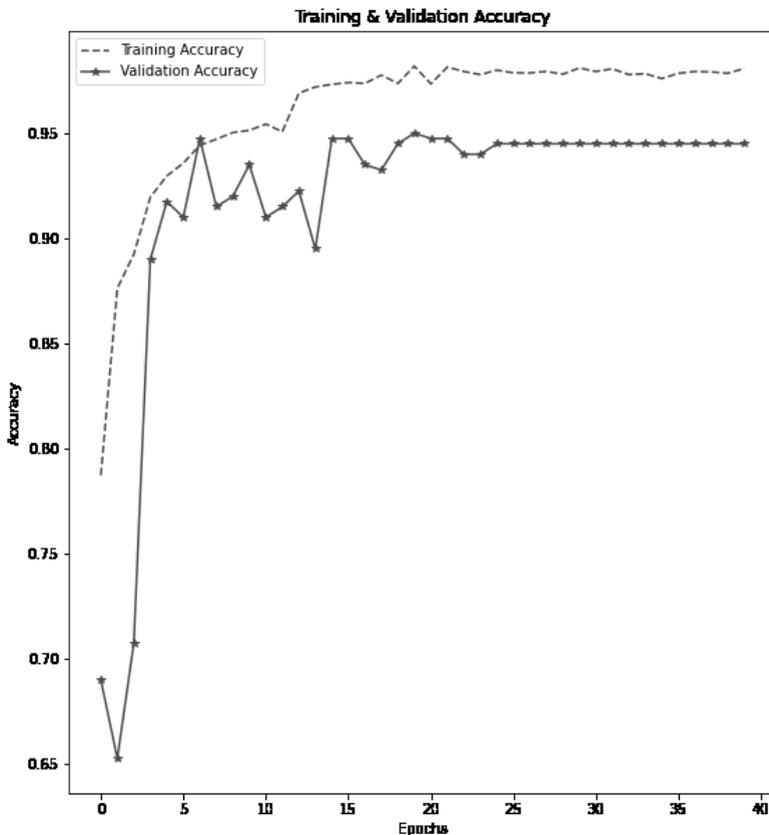


Рис. 3. Графики функции точности для множеств T и V

4. Подход, основанный на колмогоровской сложности

В двух следующих абзацах без кавычек дается определение колмогоровской сложности, в точности следуя [2, стр. 23].

Способом описания, или декомпрессором, мы называем произвольное вычислимое частичное отображение D из множества двоичных слов Ξ в себя. (Вычислимость отображения D

означает, что есть алгоритм, который применим к словам из области определения отображения D и только к ним; результат применения алгоритма к слову x есть $D(x)$.) Если $D(y) = x$, говорят, что y является описанием x при способе описания D . Для каждого способа описания D мы определяем сложность относительно этого способа описания, полагая её равной длине кратчайшего описания: $KSD(x) = \min\{l(y) \mid D(y) = x\}$.

При этом минимум пустого множества считается равным ∞ . Говорят, что способ описания D_1 не хуже способа описания D_2 , если найдётся такая константа c , что $KSD_1(x) \leq KSD_2(x) + c$ для всех слов x . Способ описания называют оптимальным, если он не хуже любого другого способа описания.

Теперь зафиксируем некоторый (не обязательно оптимальный) способ описания, и сложность слова x относительно этого способа описания обозначим $K(x)$. В дальнейшем мы будем ее считать равной числу битов в сжатой версии x , а в качестве отображения D будем использовать программы-архиваторы. Такой подход используется, например, в [13, стр. 1529]: «...Колмогоровская сложность файла – это, по существу, длина конечной сжатой версии файла».

Пусть y – еще одно двоичное слово. Следуя [13], обозначим $K(x/y)$ – минимальное количество битов, необходимых для восстановления x из y . Для любой пары x и y можно определить нормализованное расстояние сжатия как

$$NCD(x, y) = \max\{K(x/y), K(y/x)\} / \max\{K(x), K(y)\}.$$

Доказано, что расстояние $NCD(x, y)$ симметрично, аксиомы тождества и треугольника также выполняются. Теорема о симметрии алгоритмической информации доказывается в [18], а следствием из нее является примерное равенство $K(x|y) \approx K(yx) - K(y)$, здесь yx обозначает конкатенацию y и x . Поэтому с учетом того, что на практике $K(xy) \approx K(yx)$, для приближенных вычислений NCD можно записать как

$$NCD(x, y) = K(xy) - \min\{K(x), K(y)\} / \max\{K(x), K(y)\}.$$

Пусть имеется «контрольное» множество снимков $Y = Y_{Pneu} \cup Y_{Norm}$, где Y_{Pneu} – подмножество снимков лёгких с признаками пневмонии, а Y_{Norm} – подмножество снимков без

признаков пневмонии; $|Y_{Pneu}| = |Y_{Norm}| = M$. Имеется также тестовое множество X , снимки которого должны быть классифицированы как здоровые или как с признаками пневмонии; $|X| = N$. Каждый элемент множеств Y и X представляет собой файл с соответствующим снимком грудной клетки. Обозначим элементы множеств $y^i_{Pneu} \in Y_{Pneu}$, $y^i_{Norm} \in Y_{Norm}$, $x_i \in X$.

Основная идея алгоритма заключается в том, чтобы для каждого тестируемого элемента множества X было вычислено расстояние до Y_{Pneu} и Y_{Norm} , после чего снимок признается либо имеющим признаки пневмонии, либо нормальным по критерию минимума расстояния до соответствующего множества. Расстояние от элемента до множества может быть определено различными способами, в нашем случае это минимум из расстояний NCD от тестируемого элемента множества X до каждого элемента Y_{Pneu} и Y_{Norm} .

Пусть имеются программа-конвертер, преобразующая содержимое файла со снимком в бинарный файл, и программа-архиватор, предназначенная для сжатия содержимого бинарного файла. Операцию конвертирования обозначим $Conv$, а операцию сжатия – $Arch$. Последовательное выполнение двух операций над файлом снимка x обозначим $Arch(Conv(x))$. Соответственно, колмогоровская сложность $K(x)$ файла x равна размеру файла, полученного в результате $Arch(Conv(x))$, т.е. $K(x) = Size(Arch(Conv(x)))$.

Для обозначения операции конкатенации файлов будем использовать символ \oplus .

С учетом введенных понятий и определений в общих чертах алгоритм классификации рентгеновских снимков грудной клетки для диагностики пневмонии можно описать следующим образом:

Подготовительный этап алгоритма (вычисление K для контрольного множества):

для $\forall y^i_{Pneu} \in Y_{Pneu}$ вычислить
 $K(y^i_{Pneu}) = Size(Arch(Conv(y^i_{Pneu})));$

для $\forall y^i_{Norm} \in Y_{Norm}$ вычислить
 $K(y^i_{Norm}) = Size(Arch(Conv(y^i_{Norm}))).$

Основной этап алгоритма (классификация элементов тестируемого множества):

для $\forall x_i \in X$:

вычислить $K(x_i) = \text{Size}(\text{Arch}(\text{Conv}(x_i)))$;

для $\forall y^j_{Pneu} \in Y_{Pneu}$:

выполнить $x_i \oplus y^j_{Pneu}$;

вычислить $K(x_i \oplus y^j_{Pneu}) = \text{Size}(\text{Arch}(\text{Conv}(x_i \oplus y^j_{Pneu})))$;

вычислить $NCD(x_i, y^j_{Pneu}) = \frac{K(x_i \oplus y^j_{Pneu}) - \min\{K(x_i), K(y^j_{Pneu})\}}{\max\{K(x_i), K(y^j_{Pneu})\}}$;

для $\forall y^j_{Norm} \in Y_{Norm}$:

выполнить $x_i \oplus y^j_{Norm}$;

вычислить $K(x_i \oplus y^j_{Norm}) = \text{Size}(\text{Arch}(\text{Conv}(x_i \oplus y^j_{Norm})))$;

вычислить $NCD(x_i, y^j_{Norm}) = \frac{K(x_i \oplus y^j_{Norm}) - \min\{K(x_i), K(y^j_{Norm})\}}{\max\{K(x_i), K(y^j_{Norm})\}}$;

найти $sign_{Pneu} = \min_{j=1...M} NCD(x_i, y^j_{Pneu})$ и $sign_{Norm} = \min_{j=1...M} NCD(x_i, y^j_{Norm})$;

если $sign_{Pneu} \leq sign_{Norm}$, то отнести x_i к классу с признаками пневмонии, иначе отнести x_i к классу без признаков пневмонии.

Принадлежность снимка грудной клетки к одному из классов в данном описании алгоритма определяется по критерию минимума из минимальных расстояний NCD до каждого класса. Архивация проводилась с помощью программы 7z.

5. Анализ результатов

Достаточно часто результаты бинарной классификации оцениваются по матрице ошибок (confusion matrix) [7], которую представим в виде таблицы 1.

Таблица 1. Матрица ошибок

	Истинная классификация	
	пневмония	нет пневмонии
Предсказанный результат	$TP:FN$	$TN:FP$

Здесь TP (True Positive) – снимок классифицирован как принадлежащий человеку, у которого выявлена пневмония, и таковым и является; FP (False Positive) – снимок классифицирован как принадлежащий человеку, у которого выявлена пневмония, но принадлежит здоровому человеку; TN (True Negative) – снимок классифицирован как принадлежащий здоровому человеку, и снимок принадлежит здоровому человеку; FN (False Negative) – снимок классифицирован как принадлежащий здоровому человеку, но снимок принадлежит человеку с пневмонией.

В этих обозначениях метрика *precision* (точность), показывающая, какая часть положительно классифицированных примеров предсказана корректно, описывается следующей формулой:

$$precision = TP / (TP + FP).$$

Соответственно, метрика *recall* (полнота) показывает, какая часть положительных примеров классифицирована корректно:

$$recall = TP / (TP + FN),$$

а метрика *F-measure* представляет гармоническое среднее между точностью и полнотой, позволяя оптимизировать сразу две эти метрики:

$$F\text{-measure} = 2 \cdot precision \cdot recall / (precision + recall).$$

Была проведена серия экспериментов для разработанной СНС и программы, реализующей алгоритм, основанный на колмогоровской сложности (назовем ее ПКС). Наша задача заключается в сравнении двух программ, поэтому для каждой из них использовалось одно и то же сгенерированное случайным образом тестовое множество. Далее мы приведем в статье наиболее существенные результаты.

Для СНС в качестве варьируемых параметров была взята размерность обучающего множества T . В таблице 2 приведены значения для случая 20 эпох обучения сети, представленной на рис. 2, для этого сеть была обучена заново с теми же инициализационными весами и другими псевдослучайными величинами, что и в первый раз. В колонке T указано количество снимков

здоровых людей и больных пневмонией в обучающем множестве через знак «+».

Тестовое множество D содержит 50 снимков здоровых людей (для их обозначения будем употреблять слово «норма») и 50 снимков больных пневмонией (обозначение «пневмо»). Валидационное множество V для СНС во всех случаях содержит по 100 тех и других снимков.

Таблица 2. Результаты экспериментов и значения метрик для СНС

T	норма	пневмо	<i>precision</i>	<i>recall</i>	<i>F-measure</i>
200+200	0:50	50:0	0,500	1,000	0,667
250+250	12:38	49:1	0,563	0,980	0,715
300+300	46:4	45:5	0,918	0,900	0,909
400+400	49:1	46:4	0,979	0,920	0,948

Для ПКС в качестве варьируемого параметра было принято разбиение изображений на пиксели. В колонке «пиксели» таблицы 3 приводятся значения разбиения на пиксели по ширине и высоте. Контрольное множество Y содержит 100 снимков больных пневмонией и 100 снимков здоровых людей.

Таблица 3. Результаты экспериментов и значения метрик для ПКС

пиксели	норма	пневмо	<i>precision</i>	<i>recall</i>	<i>F-measure</i>
128x128	0:50	50:0	0,500	1,000	0,667
350x350	9:41	50:0	0,549	1,000	0,709
512x512	42:8	44:6	0,846	0,880	0,863
800x800	50:0	22:38	1,000	0,367	0,537

Основной этап алгоритма ПКС требует временных затрат на классификацию каждого элемента тестируемого множества, существенно возрастающих с увеличением размерности контрольного множества. Варьирование размерности показало, что существенного улучшения качества классификации при увели-

чении размерности свыше множества из 100 снимков больных пневмонией и 100 снимков здоровых людей не происходит.

В качестве тестового множества X используется множество D для СНС.

Очевидно, что с увеличением обучающего множества улучшаются значения метрик СНС. Для ПКС картина несколько иная: увеличение количества пикселей в разбиении снимка ведет к улучшению результатов до некоторого значения, после чего наступает ухудшение. В нашем случае наилучшие результаты получены в случае 512×512 . Заметим, что на одном и том же тестовом множестве эти результаты находятся между результатами $250 + 250$ и $300 + 300$ для СНС, лучше первых и хуже вторых.

Как сказано во введении, нейронные сети на сегодняшний день представляют перспективное направление для решения задачи распознавания пневмонии по рентгеновским снимкам. Приведённое исследование подтверждает это утверждение, как и тот факт, что точность классификации сильно зависит от количества представленных образцов.

В то же время предложенный подход, основанный на колмогоровской сложности, показывает возможность достаточно точной классификации в случаях, когда база снимков с истинной классификацией еще находится в стадии формирования. В случае, когда СНС обучена на большом количестве данных, ПКС уступает, хотя и незначительно, по всем метрикам. Это видно в таблице 3, вторая строка. Здесь тестовое множество $X = D$, а контрольное множество Y содержит 100 снимков больных пневмонией и 100 снимков здоровых людей.

6. Выводы

В работе рассматриваются два принципиально различных подхода к задаче классификации рентгеновских снимков грудной клетки для диагностики пневмонии. Первый, на сегодня широко распространенный подход, основан на применении нейронных сетей. Второй подход использует теоретическое понятие колмогоровской сложности, на основе которого вводится

нормализованное расстояние сжатия, позволяющее определять расстояния между объектами различной природы. Дается изложение обоих подходов, описание основанных на них алгоритмов и сведения о программных реализациях.

Проведенная серия экспериментов позволяет подобрать значения параметров для успешного решения поставленной задачи. СНС показывает высокие значения метрик качества классификации для обучающего множества начиная с 300 снимков здоровых людей и 300 снимков больных пневмонией. Для меньшего количества снимков метрики качества существенно ухудшаются. В случае ПКС мы имеем достаточно высокие значения метрик уже для случая 100 снимков здоровых людей и 100 снимков больных пневмонией, что недостижимо в случае СНС.

Отсюда можно сделать следующий вывод. Преимущества первого подхода проявляются при больших массивах данных обучающего множества. Второй подход позволяет решать эту же задачу при наличии малого числа классифицированных снимков, когда первый подход не работает. Полученные результаты дают хорошие перспективы разработки чувствительных и надежных методов диагностики пневмонии, сочетающих оба этих подхода.

Литература

1. АРБУЗОВА А.А. *Диагностика легочных заболеваний с помощью нейронных сетей* // Математическое и компьютерное моделирование естественно-научных и социальных проблем: материалы XIV Международной научн.-техн. конф. молодых специалистов, аспирантов и студентов (Россия, г. Пенза, 1–4 июня 2020 г.) – 2020. – С. 185–189.
2. ВЕРЕЦАГИН Н.К., УСПЕНСКИЙ В.А., ШЕНЬ А. *Колмогоровская сложность и алгоритмическая случайность* – М.: МЦНМО, 2013. – 576 с.
3. *Всемирная организация здравоохранения. Пневмония.* – URL: <https://www.who.int/ru/news-room/factsheets/detail/pneumonia> (дата обращения: 25.04.2022).

4. ЕФРЕМЦЕВ В.Г., ЕФРЕМЦЕВ Н.Г., ТЕТЕРИН Е.П., ТЕТЕРИН П.Е., БАЗАВЛУК Е.С. *Классификация рентгеновских изображений грудной клетки больных вирусной пневмонией и COVID-19 с помощью нейронных сетей* // Компьютерная оптика. – 2021. – Т. 45, №1. – С. 149–153.
5. ИВАНОВ В.В., ХАРИТОНОВ М.А., ДАНЦЕВ В.В., ЧУМАК Б.А., И ДР. *Дифференциальная диагностика внебольничной пневмонии и диссеминированного заболевания* // Вестник Российской военно-медицинской академии. – 2015. – №2. – С. 113–117.
6. КОЛМОГОРОВ А.Н. *Три подхода к определению понятия «количество информации»* // Проблемы передачи информации. – 1965. – Т. 1, вып. 1. – С. 3–11.
7. СТАРОВОЙТОВ В.В., ГОЛУБ Ю.И. *Сравнительный анализ оценок качества бинарной классификации* // Информатика. – 2020. – Т. 17, №1. – С. 87–101.
8. СТРУТЫНСКИЙ А.В., ЮДИН А.Л., СЕМЕНОВ Д.В., АФАНАСЬЕВА Н.И., СЕМЕНОВ В.А. *Опыт применения компьютерной томографии высокого разрешения легких в диагностике и оценке результатов лечения внебольничной пневмонии* // Пульмонология. – 2014. – №4. – С. 68–70.
9. ШОЛЛЕ Ф. *Глубокое обучение на Python*. – СПб.: Питер, 2018. – 400 с.
10. BARTLETT J.M.S., STIRLING D. *A short history of the polymerase chain reaction* // PCR protocols. – Humana Press, 2003. – P. 3–6.
11. *Chest X-Ray Images (Pneumonia)*/Kaggle. – URL: <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia> (дата обращения 26.04.2022).
12. СННИКАРА Р., SINGH Р., GUPTA Р., BHATIA Т. *Deep convolutional neural network with transfer learning for detecting pneumonia on chest X-rays* // Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals. – Springer, Singapore, 2020. – P. 155–168.
13. CILBRASI R., VITANYI P. *Clustering by compression* // IEEE Transactions on Information Theory. – 2005. – Vol. 51, Iss. 4. – P. 1523–1545.

14. IMRAN A. *Training a CNN to detect Pneumonia* – URL: <https://medium.datadriveninvestor.com/training-a-cnn-to-detect-pneumonia-c42a44101deb> (дата обращения: 14.04.2022).
15. IOFFE S., SZEGEDY C. *Batch normalization: Accelerating deep network training by reducing internal covariate shift* // Proc. of the 32nd Int. Conf. on Machine Learning. – 2015. – P. 448–456.
16. *Keras: the Python deep learning AP | RMSprop.* – URL: <https://keras.io/api/optimizers/rmsprop> (дата обращения: 26.04.2022).
17. LECUN Y., BOTTOU L., BENGIO Y., HAFNE P. *Gradient-Based Learning Applied to Document Recognition* // Proc. of the IEEE. –1998. – Vol. 86, Iss. 11. – P. 2278–2324.
18. LI M., VITÁNYI P. *An Introduction to Kolmogorov Complexity and Its Applications.* – 3rd ed. – New York: Springer-Verlag, 2008. – 809 p.
19. LI M., VITÁNYI P. *The Similarity Metric* // IEEE Trans. on Information Theory. – 2004. – Vol. 50, No. 12. – P. 3250–3264.
20. *Opencv-python PyPI.* – URL: <https://pypi.org/project/opencv-python> (дата обращения: 26.04.2022).
21. QUISPE-AYALA M.R., ASALDE-ALVAREZ K., ROMAN-GONZALEZ A. *Image classification using data compression techniques* // IEEE 26-th Convention of Electrical and Electronics Engineers in Israel. – 2010. – P. 000349–000353.
22. RAHMAN T., CHOWDHURY M.E.H., KHANDAKAR A., ISLAM K.R., ISLAM K.F., ET AL. *Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray* // Applied Sciences. – 2020. – Vol. 10, No. 9. – P. 3233. – <https://doi.org/10.3390/app10093233>.
23. SHIJIE J., PING W., PEIYI J., SIPING H. *Research on data augmentation for image classification based on convolution neural networks* // 2017 Chinese Automation Congress (CAC). – 2017. – P. 4165–4170.
24. SRIVASTAVA N., HINTON G., KRIZHEVSKY A., SUTSKEVER I., SALAKHUTDINOV R. *Dropout: a simple way to prevent neural networks from overfitting* // The Journal

of Machine Learning Research. – 2014. – Vol. 15, No. 1. – P. 1929–1958.

TWO APPROACHES FOR DETECTING PNEUMONIA IN X-RAY IMAGES: DESCRIPTION, IMPLEMENTATION AND COMPARISON

Andrey Pechnikov, Karelian Research Centre of the Russian Academy of Sciences, Petrozavodsk, Dr.Sc., Chief Researcher (pechnikov@krc.karelia.ru).

Nikolay Bogdanov, Saint Petersburg University, Saint Petersburg, Faculty of Applied Mathematics and Control Processes, Master's Degree 2021 (bogdanov699@gmail.com).

Abstract: The paper investigates two approaches for classification of x-ray images for presence of pneumonia. The first, widely used approach relies on neural networks (NN). The second approach utilises the theoretical concept of Kolmogorov complexity. The latter approach further enables the use of normalised compression distance (NCD) which defines a distance measure for arbitrary data objects, including images. Both approaches and their underlying algorithms are described in detail and implemented programmatically. The X-rays for this work are taken from the database of the Kaggle social network for data processing and machine learning. Optimal model parameters are found empirically. Experimental results show high accuracies for both approaches. This is a clear indication of reliability and applicability of either method for identifying the presence of pneumonia in x-ray images. The NN approach performs well when ample training data is available. The NCD method is turn applicable when training data is limited and the NN approach fails. These results provide a solid foundation for developing precise and reliable diagnostics of pneumonia, using a combination of the two approaches.

Keywords: classification of images, convolutional neural network, normalised compression distance, X-ray processing, pneumonia.

УДК 004.032.26 + 616.24

ББК 16.632

DOI: 10.25728/ubs.2022.99.5

Статья представлена к публикации членом редакционной коллегии А.И. Михальским.

Поступила в редакцию 28.04.2022.

Опубликована 30.09.2022.