

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных
систем

Технология программирования

Ткачева Дарья Алексеевна

Поиск аномалий изображений при
профилактическом флюорографическом
исследовании

Бакалаврская работа

Научный руководитель:
к. ф.-м. н., ст. преп. СПбГУ Салищев А. А.

Рецензент:
к. т. н. доцент ГУАП Ронжин А. Л.

Санкт-Петербург
2018

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems
Technology in Programming

Tkacheva Daria

Anomaly detection in preventive fluorographic

Bachelor's Thesis

Scientific supervisor:
Senior lecturer Sergey Salishchev

Reviewer:
Assoc. prof. Alexander Ronzhin

Saint-Petersburg
2018

Оглавление

Введение	4
1. Постановка задачи	6
2. Обзор	7
2.1. Существующие решения	7
2.2. Автоэнкодеры	8
2.2.1. Сверточные автоэнкодеры	9
2.2.2. Состязательные автоэнкодеры	9
2.3. Метод опорных векторов	10
2.4. Методы оценки качества	11
3. Обработка данных	13
4. Примененные модели	14
4.1. Состязательный автоэнкодер	14
4.2. Сверточный автоэнкодер	16
4.3. Метод опорных векторов	18
5. Результаты	19
Заключение	22
Приложение А. Конфигурация сервера	23
Список литературы	24

Введение

Флюорографическое исследование является одним из наиболее распространенных методов рентгенодиагностики, и хотя считается, что флюорография легких не достаточно информативный осмотр, полученные с его помощью данные позволяют выявить изменения в структуре легочной ткани, которые могут стать причиной дальнейшего более детального обследования. В нашей стране по-прежнему сохраняется достаточно высокий уровень заболеваемости туберкулезом, поэтому существует необходимость ежегодного прохождения флюорографического обследования.

Таким образом, создается большая нагрузка на врачей-рентгенологов, которые вынуждены исследовать каждый снимок неоднократно для снижения рисков человеческого фактора. В связи с этим растет спрос на вспомогательные системы для врачей-рентгенологов, разработанные для поиска аномалий на флюорографических снимках.

В данной работе рассматривается возможность применения генеративного подхода машинного обучения в области поиска аномалий на рентгенограммах. Идея заключается в том, чтобы обучить сверточный автоэнкодер генерировать похожие изображения, и в результате этого обучения заставить выделять его наиболее важные признаки входного изображения, после чего извлекать из скрытого пространства автоэнкодера эти признаки и пытаться классифицировать по ним нормальные рентгенограммы и рентгенограммы с патологиями методом опорных векторов.

Новизна такого подхода заключается именно в применении автоэнкодера для сокращения размерности входных данных в задаче классификации медицинских снимков, а также использование неразмеченных данных для обучения автоэнкодера. Кроме того, обучение проводилось на базе данных, предоставленных СПб НИИ фтизиопульмонологии.

В главе 3 описаны примененные методы предобработки изображений. Детали реализации сверточного автоэнкодера можно найти в главе 4.2, классификатора SVM в главе 4.3.

Применение данного подхода оценивалось по двум метрикам: AUC-ROC и AUC-PR. Обе метрики показали хорошие результаты: площадь под ROC-кривой составила 0.92, под PR-кривой 0.94. Более подробно с результатами можно ознакомиться в главе 5.

1. Постановка задачи

Целью данной работы является применение генеративного подхода машинного обучения в области поиска аномалий в легких по флюорографическим снимкам и проведение соответствующих исследований. Для достижения этой цели были поставлены следующие задачи:

1. Подготовить данные для машинного обучения: выбрать наиболее подходящие снимки и предобработать их для ускорения процесса обучения;
2. Подобрать оптимальную конфигурацию для модели машинного обучения таким образом, чтобы поиск аномалий мог быть осуществлен наиболее эффективно;
3. Провести эксперименты с реализованной моделью машинного обучения: обучить и протестировать модель, произвести оценку качества работы модели по выбранной метрике.

2. Обзор

2.1. Существующие решения

Авторы статьи [1] предложили следующую методологию обнаружения заболеваний легких, таких как туберкулез, пневмония и рак, на основе рентгенограмм грудной клетки:

1. Предварительная обработка изображений
2. Сегментация легких
3. Извлечение признаков
4. Классификация изображений.

Целью предварительной обработки изображений является удаление нерелевантных данных на рентгеновском снимке. Метод выравнивания гистограммы был использован для того, чтобы сделать изображение более подходящим для последующих этапов обработки. Для удаления нежелательного шума, присутствующего на снимке, выполняется фильтрация изображений.

Обнаружение границ легких производится с помощью метода на основе интенсивности (преобразование изображения в оттенках серого в черно-белое изображение (двоичное)) и метода на основе разрыва (оператор Собеля).

При извлечении признаков учитываются геометрические особенности, такие как площадь, периметр, эквивалентный диаметр, индекс неравномерности и статистические функции (среднее, стандартное отклонение и энтропия).

На последнем этапе проводится классификация изображений с использованием нейронной сети прямого и обратного распространения для выявления основных заболеваний легких.

База данных составлена из рентгенограмм грудных клеток 80 пациентов Sassoon General Hospital, Pune.

Авторы статьи утверждают, что искусственная нейронная сеть прямого распространения дает результаты, которые на 92% точны.

В статье [2] для классификации моделей интерстициальных заболеваний легких используется сверточная нейронная сеть. Предлагаемая сеть состоит из 5 сверточных слоев с ядрами 2×2 и функции активации LeakyReLU, за которыми следует средний пул с размером, равным размеру окончательных карт признаков и трех плотных слоев. Последний плотный слой имеет 7 выходов, что эквивалентно 7 рассмотренным классам разных болезней.

Для обучения и оценки сверточной нейронной сети использован набор данных из 14696 патчей изображений, полученных с разных сканеров и больниц из 120 КТ-сканирований. Получена точность 85%.

2.2. Автоэнкодеры

Неконтролируемое машинное обучение (или обучение без учителя) [3] ориентировано на внутреннюю структуру данных. При таком подходе начальные веса задаются небольшими случайными значениями, а затем обновляются с помощью алгоритмов прямого и обратного распространения ошибки. В отличие от контролируемого подхода (обучением с учителем) неконтролируемый подход может давать хорошие результаты, обучаясь на неразмеченных входных данных.

Автоэнкодеры [4] являются искусственными нейронными сетями, которые используют подход обучения без учителя. Автоэнкодер представляет из себя две нейронные сети: кодирующую и декодирующую. На вход подаются данные, которые кодирующая нейронная сеть (энкодер) преобразует в скрытое пространство меньшей размерности, после чего декодирующая нейронная сеть (декодер) пытается восстановить их из скрытого пространства в пространство размерности входных данных. $Code = f(Input)$ — скрытый код, полученный в результате работы энкодера. $Output = f'(Code)$ — результат работы декодера.

Входные данные сравниваются с выходными, и, таким образом, после многочисленных итераций значение функции ошибки автоэнкодера

достигает своего оптимального значения, а это означает, что восстановленные входные данные могут максимально приближать исходные.

2.2.1. Сверточные автоэнкодеры

Сверточные автоэнкодеры [5] основаны на стандартной архитектуре автоэнкодера и включают в себя уровни сверточного кодирования и декодирования. По сравнению с классическими автоэнкодерами, сверточные автоэнкодеры больше подходят для обработки изображений, поскольку они используют все возможности сверточных нейронных сетей для изучения структуры изображения.

В сверточных автоэнкодерах веса распределяются между всеми входными точками, что помогает сохранить локальную пространственность. Представление i -й карты признаков определяется как

$$h^i = s(x * W^i + b^i)$$

где b^i – смещение i -й карты признаков, $*$ обозначает свертку (2D), s – активация. Используется однократное смещение на скрытую карту, а восстановление – как

$$y = s\left(\sum_{i \in H} h^i * \hat{W}^i + c\right)$$

где c – смещение на входной канал, H – группа скрытых карт признаков, \hat{W} – флип-операция над обоими весами размерностей.

Метод обратного распространения ошибки используется для вычисления градиента функции ошибки при обновлении весов.

2.2.2. Состязательные автоэнкодеры

Генеративно-состязательная сеть [6] состоит из двух нейронных сетей: генерирующей и дискриминирующей. Генерирующая нейронная сеть (генератор) обучается генерировать «поддельные» изображения, а дискриминирующая – отличать реальные изображения от «поддельных». Их обучение итеративно чередуется. Генератор пытается обра-

нуть дискриминатор так, чтобы он ошибался при классифицировании искусственных образцов, принимая их за подлинные. Процесс обучения продолжается до тех пор, пока генератор не станет настолько хорош в создании «поддельных» изображений, что дискриминатор больше не сможет отличать реальные изображения от «поддельных».

Состязательный автоэнкодер — классический автоэнкодер в комбинации с генеративно-состязательной нейронной сетью.

2.3. Метод опорных векторов

Идея метода опорных векторов [7] заключается в поиске оптимальной гиперплоскости, которая делит набор входных данных таким образом, чтобы все точки с одинаковой меткой класса находились по одну сторону от этой гиперплоскости. Если (x_i, y_i) — набор входных данных, где x_i сопоставляется метка класса $y_i = \{-1, 1\}$, то требуется найти w и b такие, что

$$y_i(w \cdot x_i + b) \geq 1, \quad i = 1 \dots N.$$

Множество входных данных должно быть линейно разделимым для существования такой гиперплоскости. Ближайшие к разделяющей гиперплоскости вектора называются опорными векторами, а расстояние между опорными векторами разных классов равно $\frac{2}{\|w\|}$ и называется отступом. Оптимальной гиперплоскостью является та, от которой расстояние до ближайшей точки максимально. Тогда требуется минимизировать $\|w\|^2$.

В случае линейной неразделимости, входные данные отображаются в высокоразмерное пространство с помощью некоторого нелинейного отображения так, чтобы в новом пространстве данные стали линейно разделимыми. Это отображение задает ядро классификатора

$$K(x, x') = \Phi(x) \cdot \Phi(x').$$

Тогда классифицирующая функция имеет вид

$$F = \text{sign}(w \cdot \Phi(x) + b).$$

2.4. Методы оценки качества

В качестве метрики для оценки эффективности работы модели применялась интегральная метрика качества — площадь под Receiver Operator Characteristic (ROC)-кривой (Area Under ROC Curve, AUC-ROC) и площадь под Precision-Recall (PR)-кривой [8].

Решение, принятое классификатором, может быть представлено матрицей ошибок. Матрица ошибок имеет четыре категории:

- TP — классификатор обозначил объект как положительный, при этом он действительно является положительным
- TN — классификатор обозначил объект как отрицательный, при этом он действительно является отрицательным
- FP — классификатор обозначил объект как положительный, но он не является положительным
- FN — классификатор обозначил объект как отрицательный, но он не является отрицательным.

Тогда матрица ошибок имеет вид:

$$\begin{array}{cc} & y = 1 & y = 0 \\ \hat{y} = 1: & \text{True Positive (TP)} & \text{False Positive (FP)} \\ \hat{y} = 0: & \text{False Negative (FN)} & \text{True Negative (TN)} \end{array}$$

Здесь y — метка класса на объекте, \hat{y} — ответ классификатора на объекте.

Используя матрицу ошибок можно определить метрики в пространстве ROC и PR-пространстве.

Метрика AUROC определяется как площадь под кривой на графике зависимости False Positive Rate (FPR) — доли объектов, ошибочно классифицированных как положительные, и True Positive Rate (TPR) — доли объектов, верно классифицированных как положительные, при варьировании некоторого порога, используемого классификатором для предсказания.

$$FPR = \frac{FP}{FP + TN} \quad (1)$$

$$TPR = \frac{TP}{TP + FN} \quad (2)$$

Также, в рамках медицинских исследований выделяют такие показатели как чувствительность и специфичность [9]. Чувствительностью определяется способность алгоритма идентифицировать заболевание у пациента, а специфичностью – способность идентифицировать здоровых пациентов.

В рамках данной работы особое внимание уделяется чувствительности, т.к. приоритетной задачей стоит сведение к минимуму пропусков пациентов с патологиями. TPR определяет показатель чувствительности. Специфичность определяется как $1 - FPR$.

Метрика AUC-PR определяется как площадь под кривой на графике зависимости Precision (доля объектов, верно классифицированных как положительные, которые при этом действительно являются положительными) и Recall (доля объектов, верно классифицированных как положительные из всех положительных объектов) при варьировании некоторого порога, используемого классификатором для предсказания.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

3. Обработка данных

Для обучения модели использовался набор снимков, предоставленный СПб НИИ фтизиопульмонологии. Из этих снимков для датасета были отобраны снимки наилучшего качества, на которых не присутствуют посторонние предметы (например, медицинские аппараты) и вспомогательные надписи. В конечном итоге датасет для обучения автоэнкодера составил 12871 неразмеченный снимок. Предполагается, что неразмеченные данные могут содержать как нормальные снимки, так и снимки с патологиями. Также были отобраны 194 снимка без патологий, 3395 снимков с различными патологиями для тестирования обученной модели.

Для улучшения качества каждого флюорографического снимка применялось выравнивание гистограммы и уменьшение изображения до размера 128×128 , в силу ограниченных возможностей обучать модель на снимках более высокого разрешения, т.к это повлекло бы за собой значительное увеличение вычислений процессора. Методы выравнивания гистограммы и уменьшения размера изображения были реализованы с помощью библиотеки Open-CV [10] для Python [11].

4. Примененные модели

В рамках данной работы было решено применять модель сверточного автоэнкодера [5] для получения наиболее важных свойств входных изображений, после чего отобранные энкодером признаки классифицировать методом опорных векторов [7], а также была опробована модель состязательного автоэнкодера [6].

4.1. Состязательный автоэнкодер

Модель была реализована при помощи фреймворка TensorFlow [12] с разными конфигурациями состязательного автоэнкодера с 12-ю полносвязными слоями с количеством нейронов на каждом слое не больше 4000. Состязательный автоэнкодер включал в себя энкодер, декодер и дискриминатор. После каждого полносвязного слоя в энкодере и декодере применялась функция активации ReLU (так называемый «выпрямитель») с формулой:

$$f(x) = \max(0, x).$$

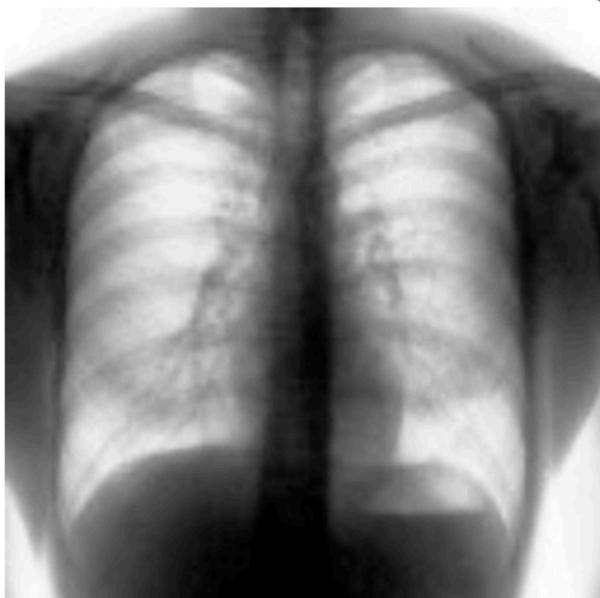
, кроме последнего слоя в декодере, где использовалась функция активации Sigmoid с формулой:

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

Дискриминатор получает вектор признаков, отобранных энкодером, которые подаются как «поддельные» изображения, и нормально распределенный вектор как реальное изображение.

Наилучшим результатом в условиях ограниченных ресурсов для машинного обучения при скорости обучения $1e-3$ были изображения на рис. 1.

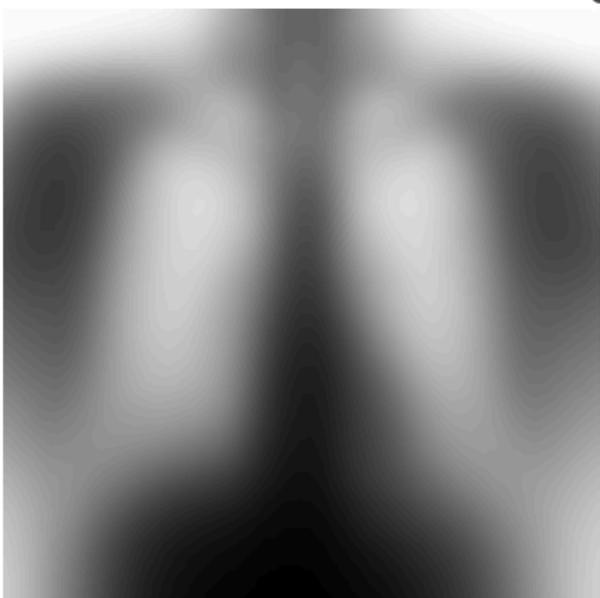
Input_Images/image/0
step 12 791 Sat Apr 14 2018 07:06:01 GMT+0300 (MSK)



Input_Images/image/1
step 12 791 Sat Apr 14 2018 07:06:01 GMT+0300 (MSK)



Generated_Images/image/0
step 12 791 Sat Apr 14 2018 07:06:01 GMT+0300 (MSK)



Generated_Images/image/1
step 12 791 Sat Apr 14 2018 07:06:01 GMT+0300 (MSK)

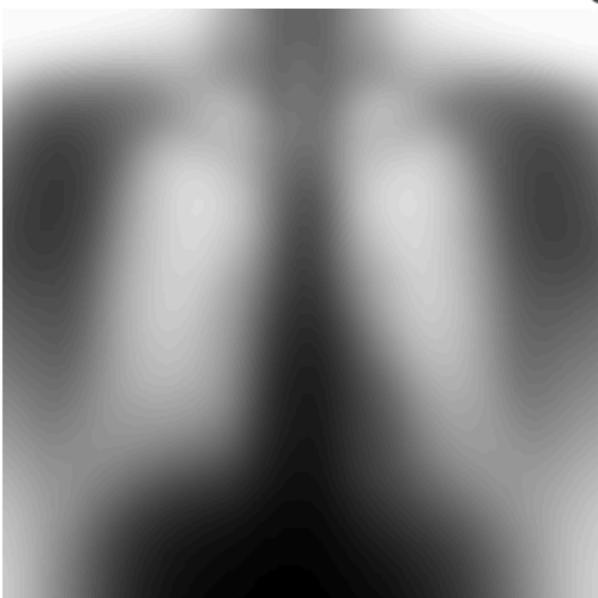


Рис. 1: Пример изображений на входе и выходе состязательного автоэнкодера

При этом можно видеть, что потеря состязательного автоэнкодера не имеет нисходящего тренда и составляет в среднем 0.027 (рис. 2).

Autoencoder_Loss

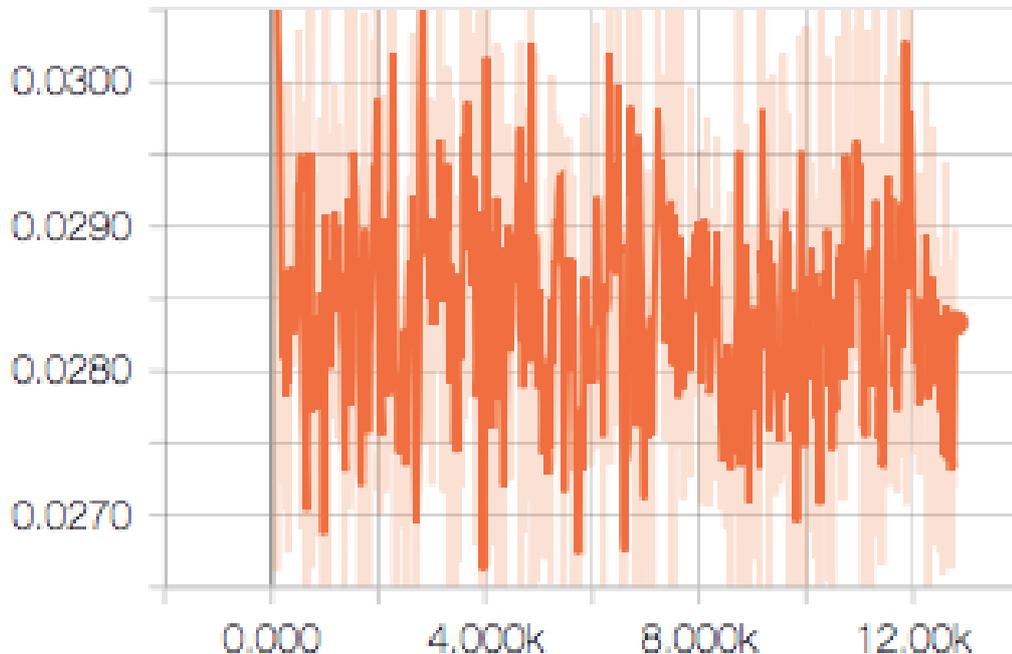


Рис. 2: График изменения функции потерь состязательного автоэнкодера

4.2. Сверточный автоэнкодер

Модель сверточного автоэнкодера реализована на языке программирования Python с помощью фреймворка PyTorch [13]. Для обучения были взяты неразмеченные снимки в количестве 12871.

Весь процесс обучения состоит из эпох. На каждой итерации эпохи выбирается 32 изображения из перемешанного в произвольном порядке датасета. Эпоха завершается, когда обучение проведено на всем наборе данных. На вход кодирующей нейронной сети подается блок из изображений размера 128×128 .

На рис. 3 представлена архитектура реализованной модели. Под каждым сверточным слоем подписано до какого размера сжимается изображение и какое количество фильтров при этом используется.

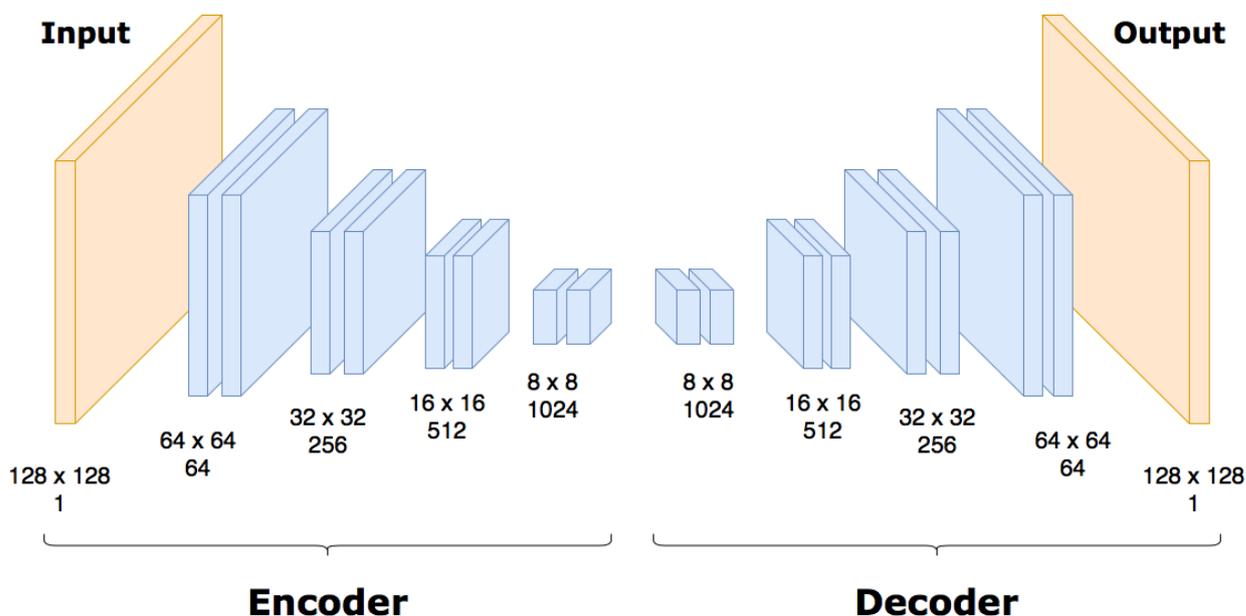


Рис. 3: Архитектура сверточного автоэнкодера

Энкодер состоит из 4 сверточных слоев с ядром размера 4, чередующихся со слоями нормализации для ускорения обучения и активационными слоями с функцией активации ReLU (так называемый «выпрямитель») с формулой:

$$f(x) = \max(0, x).$$

Декодер и энкодер похожи по своей архитектуре: декодер состоит из 4 групп слоев, состоящих из обратного сверточного слоя с ядром размера 4, слоя нормализации и активационного слоя с функцией активации ReLU. После последнего сверточного слоя используется функция активации гиперболический тангенс, который принимает на вход произвольное вещественное число, а на выходе выдает вещественное число в интервале от -1 до 1 .



Рис. 4: Пример изображений на входе сверточного автоэнкодера

Изображение, полученное на выходе декодирующей нейронной сети

имеет размерность соответствующую входному изображению.

Ошибка работы автоэнкодера рассчитывалась как средняя квадратическая ошибка между исходным изображением и обработанным нейросетью изображением. Скорость обучения модели составляла $1e-2$.

При обработке обученным автоэнкодером флюорографических снимков были получены изображения, на которых вполне различимы легкие (рис. 5).



Рис. 5: Пример изображений на выходе сверточного автоэнкодера

4.3. Метод опорных векторов

Для обучения модели классификатора SVM на языке программирования Python использовались библиотеки scikit-learn [14].

Классификация флюорографических снимков реализована методом опорных векторов (SVM). Идея состоит в том, чтобы подавать на вход автоэнкодеру снимки, помеченные как аномалии, и снимки, помеченные как здоровые легкие, после чего пытаться классифицировать методом опорных векторов данные, полученные из скрытого пространства автоэнкодера.

В качестве ядра классификатора была выбрана радиальная базисная функция (RBF), т.к данные не являются линейно разделимыми. Штрафной параметр C определяет отступ между двумя ближайшими точками разных классов, а параметр γ определяет диаметр сферы RBF [15]. Чтобы подобрать оптимальные C и γ были проведены эксперименты с разными значениями. Поиск C осуществлялся среди значений 0.1, 1, 10, γ – среди значений 0.0001, 0.001, 0.01, 0.1, 1. Наилучшие результаты при обучении SVM показал классификатор с параметрами $C = 10$ и $\gamma = 0.01$.

5. Результаты

Применение состязательного автоэнкодера в генеративном подходе себя не оправдало, поскольку по рис. 1 можно видеть, что модель не обучилась генерировать похожие изображения. Также из рис. 2 можно заключить, что модель состязательного автоэнкодера явно недообучена, и для ее обучения нужно в разы больше времени и ресурсов памяти для увеличения количества нейронов на полносвязных слоях.

Обучение сверточного автоэнкодера требует меньше ресурсов и времени. На рис. 5 видно, что изображения, полученные на выходе, чуть хуже по качеству, чем те, что подаются на вход.

При обучении применялся метод оптимизации Адама со скоростью обучения $1e-2$ и было проведено всего 10 эпох обучения, что составляет около 400 итераций. Обучение проводилось на сервере Microsoft Azure (с конфигурацией сервера можно ознакомиться в Приложении А) и заняло 3 дня. График изменения функции потерь автоэнкодера в зависимости от шага обучения представлен на рис. 6. На последних итерациях потеря автоэнкодера составила 0.0022.

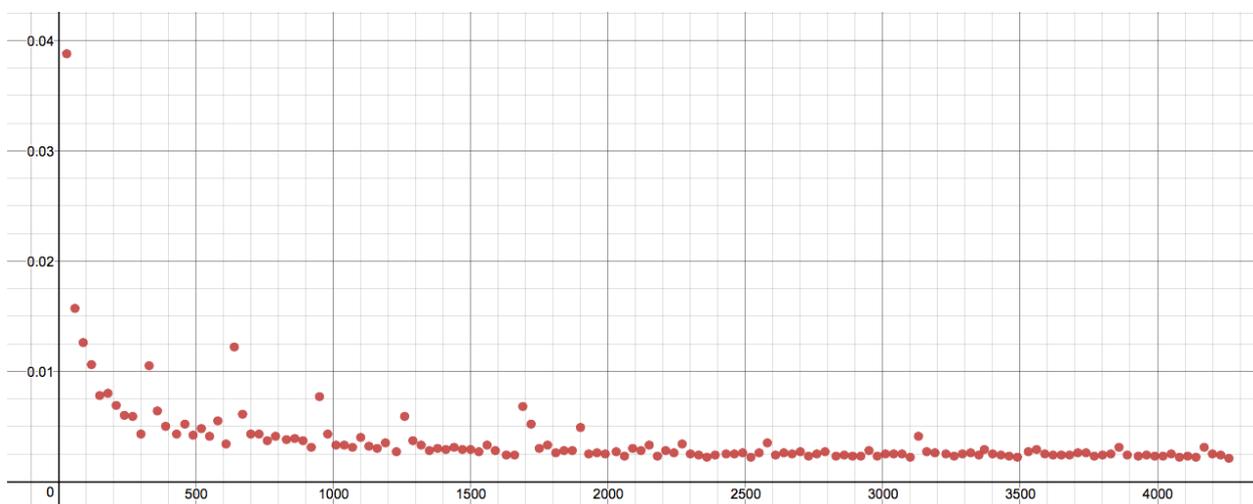


Рис. 6: График изменения функции потерь сверточного автоэнкодера

Данные для обучения классификатора SVM были разделены в одинаковой пропорции: 2716 снимков с аномалиями и 2716 неразмеченных снимков, которые были помечены как снимки, не содержащие патологий. Предполагается, что среди неразмеченных снимков мало таких, на

которых могут обнаружиться аномалии. Для тестирования были взяты оставшиеся 679 снимков с аномалиями, 193 снимка, которые точно не содержат аномалий, и 486 неразмеченных с меткой «здоровые легкие».

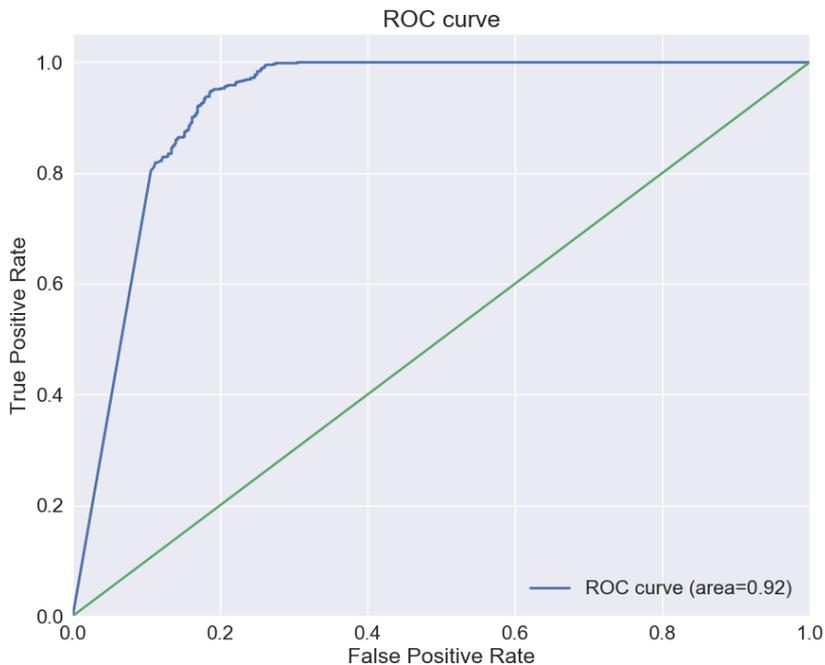


Рис. 7: ROC-кривая

Для оценки качества работы обученных моделей использовалась метрика AUC-ROC (рис. 7) и AUC-PR (рис. 8), которые показывают, что данный подход хорошо применим в поиске аномалий на флюорографических снимках. Площадь под ROC-кривой составила 0.92. Под PR-кривой 0.94. Важно, что при такой оценке мы имеем дело именно со сбалансированными классами (679 – нормальных снимков, 679 – аномалий).

Для проверки того, как SVM классифицирует неразмеченные данные, было проведено дополнительное тестирование, которое показало, что против 10098 нормальных находится всего 56 рентгенограмм с аномалиями.

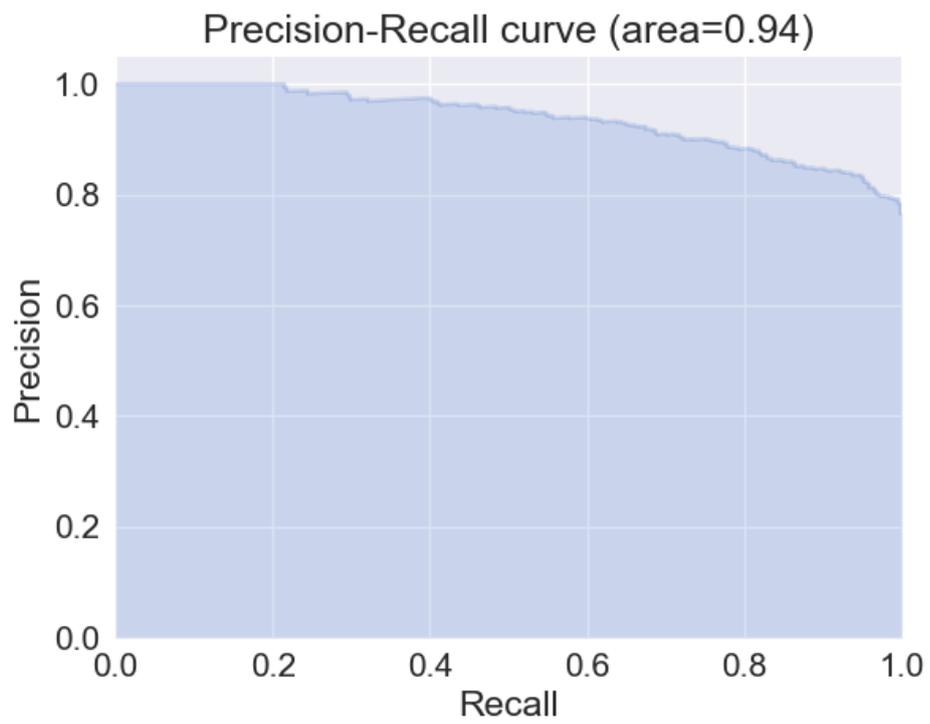


Рис. 8: PR-кривая

Заключение

В рамках выпускной квалификационной работы были достигнуты следующие результаты:

- Были подготовлены данные для обучения выбранной модели: выбраны наиболее подходящие снимки, которые не содержат ничего лишнего, что могло бы помешать в процессе обучения, в качестве предобработки изображений применялось выравнивание гистограммы и уменьшение изображений до размера 128×128 ;
- Путем проб и ошибок была подобрана оптимальная конфигурация для реализации модели сверточного автоэнкодера, а также подобраны оптимальные параметры C и γ классификатора SVM с ядром RBF. Модели реализованы на языке программирования Python с помощью фреймворка PyTorch и библиотеки scikit-learn;
- Проведены эксперименты с реализованными моделями: проведено обучение и тестирование на предварительно обработанных данных, для оценки качества работы модели использовалась метрика AUC-ROC и AUC-PR. Наилучший результат составил 0.92 по метрике AUC-ROC и 0.94 по метрике AUC-PR.

A. Конфигурация сервера

Обучение моделей провизводилось на сервере Microsoft Azure со следующей конфигурацией:

- Model name: Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz
- L1d cache: 32K
- L1i cache: 32K
- L2 cache: 256K
- L3 cache: 51200K
- RAM: 16G

Список литературы

1. Shubhangi Khobragade Aditya Tiwari C.Y Pati1, Narke Vikram. Automatic Detection of Major Lung Diseases Using Chest Radiographs and Classification by Feed-forward Artificial Neural Network // IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES-2016). — 2016.
2. Marios Anthimopoulos Member IEEE Stergios Christodoulidis Member IEEE Lukas Ebner Andreas Christe, Stavroula Mougiakakou Member IEEE. Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network // IEEE Transactions on Medical Imaging. — 2016. — Vol. 35.
3. Zhao Zheng, Liu Huan. Spectral Feature Selection for Supervised and Unsupervised Learning // Proceedings of the 24th International Conference on Machine Learning. — ICML '07. — New York, NY, USA : ACM, 2007. — P. 1151–1157. — URL: <http://doi.acm.org/10.1145/1273496.1273641>.
4. Min Chen Senior Member IEEE Xiaobo Shi Yin Zhang Senior Member IEEE Di Wu Mohsen Guizani Fellow IEEE. Deep Feature Learning for Medical Image Analysis with Convolutional Autoencoder Neural Network // IEEE Transactions on Big Data. — 2017.
5. Gondara Lovedeep. Medical image denoising using convolutional denoising autoencoders // CoRR. — 2016. — Vol. abs/1608.04667. — 1608.04667.
6. Barone Antonio Valerio Miceli. Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders // CoRR. — 2016. — Vol. abs/1608.02996. — 1608.02996.
7. Olivier Chapelle Patrick Haffner, Vapnik Vladimir N. Support Vector Machines for Histogram-Based Image Classification // IEEE Transactions on Neural Networks. — 1999. — Vol. 10.

8. Davis Jesse, Goadrich Mark. The Relationship Between Precision-Recall and ROC Curves // Proceedings of the 23rd International Conference on Machine Learning. — ICML '06. — New York, NY, USA : ACM, 2006. — P. 233–240. — URL: <http://doi.acm.org/10.1145/1143844.1143874>.
9. Lalkhen Abdul Ghaaliq, McCluskey Anthony. Clinical tests: sensitivity and specificity // Continuing Education in Anaesthesia Critical Care & Pain. — 2008. — Vol. 8, no. 6. — P. 221–223. — URL: <http://dx.doi.org/10.1093/bjaceaccp/mkn041>.
10. Bradski G. The OpenCV Library // Dr. Dobb's Journal of Software Tools. — 2000.
11. Официальный сайт Python. — URL: <https://www.python.org>.
12. Abadi Martín, Agarwal Ashish, Barham Paul et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. — 2015. — Software available from tensorflow.org. URL: <https://www.tensorflow.org/>.
13. Automatic differentiation in PyTorch / Adam Paszke, Sam Gross, Soumith Chintala et al. — 2017.
14. Scikit-learn: Machine Learning in Python / F. Pedregosa, G. Varoquaux, A. Gramfort et al. // Journal of Machine Learning Research. — 2011. — Vol. 12. — P. 2825–2830.
15. Xueying Zhang Yueling Guo. Optimization of SVM Parameters Based on PSO Algorithm // Natural Computation, 2009. ICNC '09. — 2009.