В марте 2020 года ВОЗ (всемирная организация здравоохранения) объявила пандемию короновирусной инфекции. Пандемия каснулась миллионов людей по всему миру. Самым важным симптомом является пневмония.

Было обнаружено на ранних стадиях исследований, что поражение можно увидеть на снимках КТ.

ПРЦ долгий, трудозатратный,сложный. Кроме того неточный.

Альтернативный метод скрининга - КТ грудной клетки. На снимке рентгенолог визуально определяет маркеры, соответствующие заболеванию. На ранних этапах исследований было установлено, что наличие на снимке аномалий является признаком инфицирования. Также в пользу применения в диагностике такого подхода говорит возможность

-быстрой диагностики в условиях большого количества пациентов в больших городах

-наличие необходимого рентген оборудования для проведения снимка

-возможность передачи снимков по сети для консультаций.

Следовательно, анализ рентгеновского изображения может быть проведен быстрее и является хорошим дополнением к ПРЦ тестированию. Так же предполагается, что сканирование может являться необходимостью, если у пациента отрицательный результат ПЦР-теста.

Таким образом, рентгеновское исследование является частью разработанной стратегии диагностики заболевания. Поскольку визуальные индикаторы могут быть нечеткими, у рентгенологов возникает необходимость в помощи от более точных и быстрых цифровых системах.

Рентгеновский снимок представляет собой изображение темных областей внутренних органов с белыми «тенями». Рентген – один из самых известных и старых методов диагностики. В сравнении с другими методами диагностики COVID-19, рентген изображение имеет несколько «+»ов – низкая стоимость, широкая доступность, неинвазивный метод исследования. Таким образом, этот метод диагностики может являться лучшим кандидатом на массовую, быструю и простую диагностику в условиях панемии и глобального кризиса системы здравоохранения, как с точки зрения необходимого количества медицинских работников, так и с точки зрения диагностического оборудования.

-1.1Описание развития ИИ.

Глубокое обучение продемонстрировало большое влияние за последнее десятилетие на решение многих сложных задач.

Модели глубокого обучения состоят из глубоких нейронных сетей, которые автоматически изучают важные функции из данных. Обучение глубоких моделей обычно выполняется с использованием оптимизации **стохастического градиентного спуска**.

За последнее десятилетие машинное обучение и ИИ получили значительное развитие. Нейронные сети превзошли традиционные модели по многим важным показателям. Зарекомендовали себя в широком спектре областей – распознавание речи, обработка изображений, распознавание речи.

Качество анализа и классификации медицинских изображений уже можно де факто признать стандартом. СNN исользовались для различных задач классификации медицинских изображений, таких как: болезни легких, определение переносчика малярии в образцах крови, рак груди, беспроводная эндоскопия, диагностик рака кожи по классификации изображения.

Another recent and relevant study has been conducted on validation and adaptability of Decompose-, Transfer-, and Compose-type deep CNN for COVID-19 detection using chest X-ray image classiﬁcation (A. Abbas, M. M. Abdelsamea, and M. M. Gaber, “Classiﬁcation of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network,” 2020, http://arxiv.org/abs/2003.13815). The authors have reported the results of the study with an accuracy of 95.12%, sensitivity of 97.91%, and speciﬁcity of 91.87%.

На невысокие качественные характеристики ранних модель влияли недостаточность снимков с COVID-19 и низкое качество некоторых изображений в датсетах.

Искусственный интеллект, усиленный методами глубокого обучения может играть важную роль для скринига COVID-19.

Сверточная нейронная сеть

Нейронная сеть имитирует работу части человеческого мозга, отвечающую за распознавание образов. НС используется для распознавания и анализа изображений, классификации изображений, а так же распознавания речи. СНН это тип глубоких нейронных сетей, содержащий сверточные, объединяющие уровни, и уровень активации. Основным является уровень свертки, в котором расположены ядра, на входы которым подается изображение. Все выходы сверточного слоя составляют карту объектов.

Applications of two-dimensional CNN

Image Classification

Классификация изображений - это задача отнесения изображения к классовой категории. CNN представляет собой прорыв в этой области.

LeNet-5 [10] считается первым приложением, используемым для классификации рукописных цифр. AlexNet [11] заставил подходы к классификации, основанные на CNN, сдвинуться с мертвой точки. Затем Симонян [32] и др. подчеркивают важность глубины, но эти примитивные CNN состоят не более чем из десяти слоев. Впоследствии появились более глубокие сетевые структуры, такие как GoogLeNet [33] и VGGNets [32], которые значительно повышают точность в задачах классификации.

-1.2Описание существующих подходов и реализаций

**1.2.1Одна из первых опенсорсных - COVID ‑Net(2020)**

Сверточная нейронная сеть, приспособленная для выявления КОВИДА на снимках КТ. Открытый проект.

Открыла новое применение искусственного интеллекта для раннего, эффективного и масштабного выявления вирусов среди людей.

Использоаван датасет COVIDx 13 975 изображений от 13 870 пациентов. Датасет является компиляцией пяти открытых источников данных:

COVID-19 Image Data Collection

COVID-19 Chest X-ray Dataset Initiative

ActualMed COVID-19 Chest X-ray Dataset Initiative

RSNA Pneumonia Detection Challenge

COVID-19 radiography.

-Принципы разработки сети.

Детали реализации:

Предобучена на ImageNet, а затем обучена на COVIDx датасете с использованием Adam оптимизатора. learning rate = 2e−4, number of epochs = 22, batch size = 64, factor = 0.7, patience = 5. Finally, we introduce a batch re-balancing strategy to promote better distribution of each infection type at a batch level. The initial COVID-Net prototype was built and evaluated using the Keras deep learning library with a Tensor-

Flow backend. The proposed COVID-Net architecture was built using generative synthesis 41 , as described in

“Machine-driven design exploration”

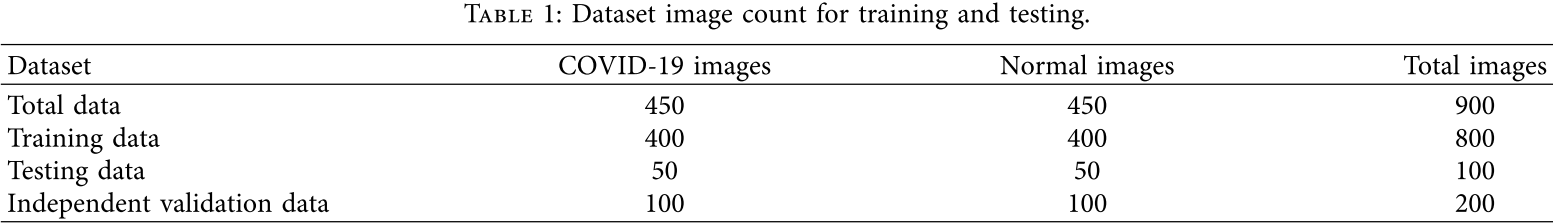
**1.2.2CNN(2021)**

ИИ и сверточные сети в частности достигли успеха в анализе медицинских изображений и классификации. В этой работе представлена нейронная сеть для диагностики COVID-19 по классификации рентгеновских снимков.

Модель тестировалась на 100 снимках для получения точности 100%. Вторым этапом модель была протестирована на несвязанном датасете и показала 99,5% точности.

Для обучения модели использовался датасет “Github Covid19 X-ray dataset” из репозитрия https://github.com/ieee8023/covid-chestxray-dataset, 2020. Online. Поскольку размер датасета мал и несбалансирован, то для решения этой проблемы была применена аугментация. Для генерации широкого набора вариантов применялись поворот изображения на разные углы. Для балансировки датасета по классам были добавлены изображения минорных классов.

Первый этап сбор первичного набора изображений двух классов: с COVID-19 и здоровых пациентов. Вторым этапом некачественные и недостаточно информативные для постановки диагноза изображения исключались медицинскими специалистами. Таким образом был сформирован качественный датасет. На третьем этапе датасет был аугментирован стандартными техниками для увеличения его размера. Результирующий датасет был использован для обучения сети на следующем этапе. После обучения модель была протестирована на качество распознавания на первичном датасете, а также на обособленном датасете. Каждый датасет имел свои характетиристики по количеству изображений, тестовый набор, сверочный набор, и разное соотношение изображений, принадлежащих одному классу.



--Материалы и методы.

Датасет

В первичный набор состоял из 178 рентгеновских изображений грудной клетки. Из которых 136 были снимками с подтвержденным заболеванием, и 42 изображения - обычные или с различными заболеваниями, такими как пневмония (J. P. Cohen, “Github Covid19 X-ray dataset,” 2020, https://github.com/ieee8023/covid-chestxray-dataset, 2020.). Таким образом, датасет состоял из изображений двух классов – COVID-19 и другие. Такое распределение показывает несбалансированность датасета. Следовательно, для достижения лучшего результата необходим предподготовка. На изначальном наборе данных модель показала недостаточную точность в 54%.

Техническое оснащение для обучения модели Core i7 7th поколения, 8 GB RAM, Microsoft Windows 10 операционная система,написана на языке программирования Python с использованием библиотеки Anaconda 3 и среды Jupyter Notebook.

Препроцессинг данных

Балансировка классов

Для балансировки набора данных использовались изображения 136 изображений без признаков заболевания из набора данных (P. Mooney, “Kaggle X rays dataset,” 2020, https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia Online.)

После балансировки датасета модель показала точность в 69%, что еще не достаточно эффективной системы диагностики COVID-19.

Анализ экспертами

Тщательный анализ изображений медицинскими специалистами 135 изображений выделил 90 изображений, подходящих для обучения модели. Результирующий датасет содержал 90 изображений COVID-19 и 90 обычных изображений. Полученный результат показал улучшение качества распознавания заболевания в 72%. Однако, невысокий прирост качественной характеристики обусловлен небольшим количеством изображений.

Аугментация

Аугментация данных, это способ существенно увеличить объем данных для обучения модели. Для изображений используются базовые операции такие как переворачивание, вращение, обрезка или дополнение. Первоначальный набор данных дополняется измененными таким образом изображениями, существенно увеличивая размер датасета для обучения нейронной сети. Изменение изображений дает дополнительные данные для обучения модели.

Поскольку изображения являются нелинейными по природе, то применяются также нелинейные функции активации.

Пример нелинейной функции активации – функция ReLU, где для всех неположительных аргументов, значение функции равно 0. График функции изображен на рисунке 1.

(1)

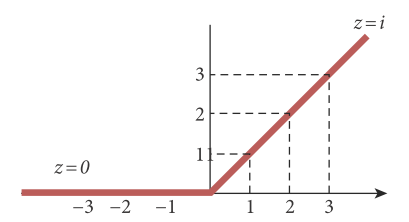


Рис График функции ReLU

Уровень объединения или уровень подвыборки также важный слой в рассматриваемом типе нейронной сети. Каждый объект, извлеченный сверточным слоем, обрабатывается независимо в этом слое. Чтобы минимизировать переобучение и количество выделенных объектов, размер карты уменьшается, и возвращаются только важные объекты. Значения объединения могут быть максимальными (max), средними (average) и суммированными (sum).

Эта сеть использует MAX пуллинг, более подходящий для определения границ контраста.

Дополнительно используется слой пакетной нормализации (batch normalization) для включения глубокой нейронной сети в обучение.

Модель настраивает масштабирование и активацию, чтобы нормализовать входной слой и ускорить процедуру обучения между скрытыми блоками.

Для минимизации переобучения использовался слой дропаута с параметром 20%.

Выравнивающий слой создает вектор из пикселей изображения, полученного после сверточных слоев, и передает на вход полносвязных слоев.

В полносвязном слое каждый нейрон предыдущего слоя напрямую связан с каждым нейроном последующего слоя. Основная задача полносвязного слоя это классификация изображения по полученным сглаженным результатам от предыдущего слоя. Каждое значение сглаженного набора объектов представляет вероятность принадлежности объекта к определенному классу. Таким образом, на основе этих вероятностей, полносвязная сеть в конечном итоге определяет решение о классификации.

Архитектура

Модель состоит из 38 слоев, в каждом из которых 6 сверточных, 6 объединяющих, 6 дропаута, 8 слоев с функциями активации, 8 слоев батч нормализации, 1 сглаживающий и 3 полносвязных слоя. Размер входных изображений 150\*150\*3 – 150\*150 в цвете RGB…

После обработки изображения получается 3-размерная матрица, которая преобразуется в вектор, который передается на вход 3 полносвязным слоям.

Используется бинарная классификация, поэтому используется бинарная кроссэнтропия (BCE) функции потерь. В двоичной классификации, поскольку для отнесения данных к одному из двух заданных классов требуется только один выходной узел, поэтому в случае функции потерь BCE выходное значение присваивается сигмовидной функции активации. Выходное значение, выдаваемое сигмовидной функцией активации, находится в диапазоне от 0 до 1. Он находит ошибку между предсказанным классом и фактическим классом. Был использован оптимизатор “Adam”, который изменяет вес атрибута и скорость обучения, чтобы уменьшить потери модели обучения.

Подход к определению числа слоев состоял в постоянном увеличении количества слоев, до достижения необходимых результатов, начиная с одного слоя.

**1.2.3AC-CovidNet(2022)**

Метод изучает устойчивые и дискриминационные признаки с помощью контрастивных потерь. Более того, предлагаемый метод придает большее значение инфицированным регионам, управляемым механизмом внимания.

attention guided contrastive CNN

Отличия:

1 Новый фреймворк для распознавания заболеванияна рентгеновских снимках

2 Используется модуль внимания, помогающий обучению.

3 Контрастные потери увеличивают способность к классификации изображения и надежность модели.

Использовались возможности механизма внимания и контрастного обучения, чтобы справиться с обучением с ограниченными данными для распознавание Covid-19 по изображениям.

Механизм внимания в глубоком обучении облегчает изучение локализованных признаков, что более важно в контексте проблемы распознавания Covid-19 на рентгенограммах.

Также обнаружено, что модель, основанная на внимании, может превзойти модели простой нейронной сети (Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.,Polosukhin, I.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)).

Сравнительное обучение - это свежая тенденция к изучению сходства и различия между похожими и непохожими образцами в абстрактном пространстве признаков для визуальных представлений. Как правило, контрастивное обучение зависит от сходства признаков между положительными и отрицательными парами.

Описание модели

Основана на модели COVID-Net. В модели широко используется легкий механизм расширения проекции остаточного проецирования (PEPX). В этой архитектуре также используется селективная связь на большие расстояния в модели, что улучшает репрезентативность. Это также облегчает обучение модели. Однако широкое использование этих соединений дальнего действия может привести к появлению множества избыточных низкоуровневых функций. Для решения этой проблемы в предлагаемой модели используется механизм внимания. Внимание помогает модели расставить приоритеты в важных областях. Внимание также полезно для подавления активации избыточных функций из начальных слоев и помогает сосредоточиться на важных функциях, необходимых для решения задачи. Используются шлюзы внимания в предлагаемой архитектуре на различных уровнях архитектуры COVID-Net, где используется много соединений дальнего действия. Это повышает чувствительность, поскольку модель лучше учитывает важные визуальные особенности зараженных областей на изображениях CXR из-за Covid-19. Поскольку разница между признаками Covid-19 и пневмонии очень тонкая, предлагается использовать контролируемую контрастную потерю. Контрастная потеря помогает сети максимально увеличить расстояние между выученными представлениями классов.

Архитектура

Архитектура модели представляет собой расширенную архитектуру модели COVID-Net. Используются ворота внимания, где сходятся несколько соединений дальнего действия. Используются уровни PEPX и шлюзы внимания. Также используется функция контрастных потерь во время обучения. Модель сначала обучается с использованием контролируемого контрастного обучения, а затем завершается точной настройкой с использованием контролируемого обучения.

PEPX слой

Архитектура проекции-расширения-проекции-расширения (PEPX). Задача модуля состоит в том, чтобы проецировать объекты в более низкое измерение, используя первые два слоя conv1x1, затем расширять эти объекты с помощью слоя свертки по глубине (DWConv3x3) и снова проецировать в более низкое измерение, используя два слоя conv1x1. Таким образом, уровень PEPX приводит к эффективной модели за счет уменьшения количества параметров и операций.

Ворота внимания

Используются на различных уровнях. Функции из нескольких слоев проходят через conv1x1 и складываются вместе.Затем агрегированные функции передаются через функцию активации Relu, за которой следует conv1x1, а затем функция активации сигмовидной формы. Затем выходные данные карты объектов сигмовидного слоя проходят через ресемплер. Выходные данные ресемплера добавляются к функциям из ближайшего входного слоя к модулю внимания, чтобы получить выход ворот внимания (рисунок 2).



Рис Схема модуля Ворота внимания

Детали нейронной сети. Описан алгоритм

Используется контролируемый контрастный метод обучения для обучения сети кодировщика извлечению признаков. Сеть классификатора обучается с использованием функции кросс-энтропийных потерь после замораживания сети кодировщика.

Контрастная потеря чаще всего используется в неконтролируемом и самоконтролируемом обучении.

\*\*формулы и описание

Использованный датасет

Сеть обучалась на данных датасета COVIDx.Это саамы большой публичный набор рентгенограмм специализированный под COVID-19. Этот датасет является компиляцией нескольких открытых наборов изображений. Содержит изображения трех классов: COVID-19, пневмония, и без патологий. В наборе 14 258 изобрежений, распределенных следующим образом:

7966-без патологий

5475 –пневмония

517 – COVID-19

в обучающем наборе и по 100 изображений каждого класса для тестирования.

На основе датасета было сформировано три датасета с различным распределением изображений по классам (рис 3)

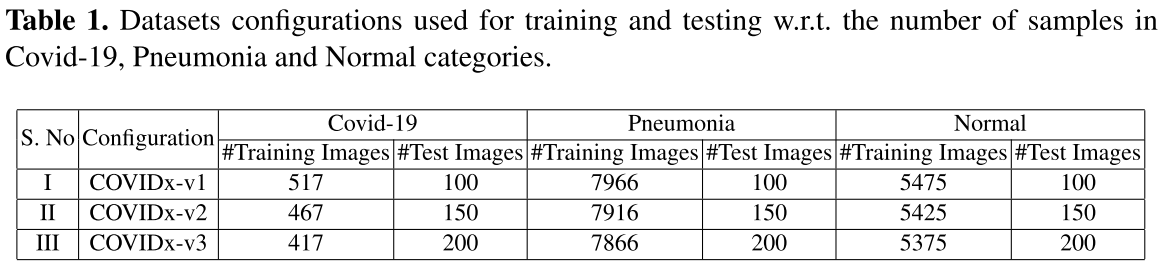


Рис Конфигурации наборов данных

Обучающие настройки

Модель была обучена на датасете. Последующее обучение модели состоит из двух этапов. На первом этапе сеть обучается с использованием функции потерь, для извлечения объектов. На втором этапе экстрактор признаков фиксируется и обучается путем добавления классификатора с функцией перекрестной потери энтропии.

Модель обучалась на трех созданных вариантах датасета. Использовался оптимизатор Adam. Значение уровня обучения составляло 1,7\*10^-4. Размер батча – 64. На каждом слое использовалась функция активации ReLU и софтмакс на последнем слое. Пулнг с параметром max используется после каждого PEPX слоя.

Модель обучалась с использованием вычислительных ресурсов Google Colab. Использовалась библиотека глубокого обучения Keras с tensorﬂow.

Использование ансамбля.

В основном исследователи используют одну архитектуру нейронной сети и с помощью подбора количества слоев и настройки коэффициентов пытаются достичь необходимых показателей качества распознавания (классификации) изображения. Некоторые исследователи создают новые архитектуры для решения задачи. Ни одно из ранних исследований не пыталось комбинировать модели с целью улучшения качества классификации модели в целом. Несколько исследований показывают, что ансамблевые модели лучше предсказывают, чем индивидуальные модели, а также помогают предотвратить переобучение. Одним из методов ансамблирования является вычисление средневзвешенного значения выходной вероятности каждой нейронной сети. Такой метод является более качественным, по сравнению с основанными на простом среднем значении.

Описание метода

Для принятия взвешенного решения в реальной жизни во многих сферах основываются на несколько мнений. Основываясь на этот подход, были адаптированы несколько моделей нейронных сетей. Каждая модель обучалась независимо. После обучения модели были скомбинированы, с использованием средневзвешенной средней для предсказания класса.

В модели используются три сверточные нейронные сети - DenseNet201, Resnet50V2, Inceptionv3.

Особенностью DenseNet является меньшее количество настраиваемых параметров, чем в других сверточных нейронных сетях. Каждый слой сети принимает на вход карты объектов от всех предыдущих слоев. Такая передача позволяет качественней передавать объекты между слоями.

Сеть ResNet50V2 с помощью остаточных блоков решает проблему исчезновения градиента. (что за поблема?). В резидентной сети блоки составляют стэк.

Inception мощная модель, с высокой точностью извлечения признаков и классификации изображения на основе этих признаков. Модель состоит из 48 слоев, в каждом из которых 11 модулей. Каждый модуль состоит из сверточных фильтров, пулинг слоев, и модуля функции активации ReLU. Занечение дропаута – 0.6.

При распределении весов в ансамбле больший вес присваивается модели с меньшей ошибкой валидации.

-предоработка изображений

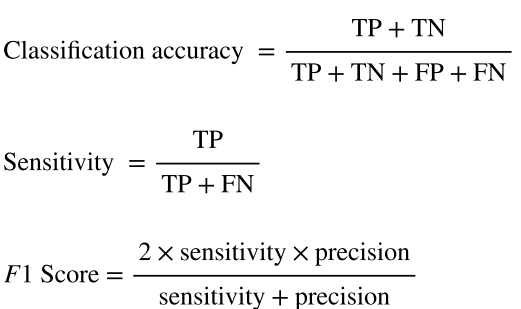
Датасет собран из других открытых наборов данных. Используются фронтальные снимки легких, поскольку на такой профиль снимка может быть лучше изучен. Собранные набор данных содержал три класса изображений – Ковид, пневмония, нормальный. Собранный датасет был разделен на два класса – Ковид+ и Ковид-. В итоге изображений класса Ковид+ в датасете 538 и 468 класа Ковид-.

-техническое описание

Обучение и тестирование проведено на платформе Google Colab, с использованием языка Python, библиотеки глубокого обучения Tensorflow.

-метрики качества сети

Оценка эффективности модели основывается по следующим параметрам: точность классификации, чувствительность и цена F1.



where TP stands for True Positive, FP for False Positive, FN for False Negative and TN for True Negative.

-сравнение с бенчмарком

Результат эксперимента

Модель принятия решения, основанная на технике средневзвешенного решения показала точность 91,62%, и чувствительность 95%.

Развитие разработки нейронных сетей осуществляется по направлениям создания новых архитектур под конкретные задачи, или же путем комбинирования и настройки нескольких моделей, и выработки результата на основе взвешенного подхода.

1.3Проблематика данных

Один из этапов обучения нейронной сети является предобработка имеющихся данных. Необходимость предобработки связана с недостатком качественных, размеченных и подтвержденных данных.

При разработке каждой модели нейронной сети требуется размеченный тренировочный набор данных.

Общей проблемой данных является их доступный объем. Имеющиеся открытые датасеты имеют разные характеристики по объему, по качеству и размеру изображений, сбалансированность по классам.

Также существенным барьером для развития искусственного интеллекта является наличие качественной верификации данных. Процесс разработки качественной разметки тренировочного набора данных требует вовлечение экспертов в своей области. Разметка набора данных трудоемкий процесс, поэтому многие публично доступные наборы данных небольшого объема.

Каждый медицинский центр может создавать датасеты различного качества разметки и формата, что затрудняет процесс обучения.

Следовательно, качество обучения алгоритмов анализа медицинских изображений связано с качеством тренировочного датасета.

Таким образом, существует проблема сбора, верификации, накопления и предоставления данных.

В России для решения проблемы качественных данных в 2020 году приступили к формированию федеральных (общероссийских) датасетов экспертами Центра диагностики и телемедицины. Первый датасет будет ориентирован на рентгенограммы и снимки компьютерной томографии пациентов с признаками COVID-19. Целью проекта является создание эталонного датасета, с помощью которого специалисты смогут оценивать корректность и точность работы сервисов по выявлению патологий в исследуемых областях.

Важной особенностью датасета является полная деперсонализация данных, то есть полное исключение персональных сведений о пациенте. Такой подход снимает вопрос защиты персональных данных при передаче накопленного массива исследователям и разработчикам.

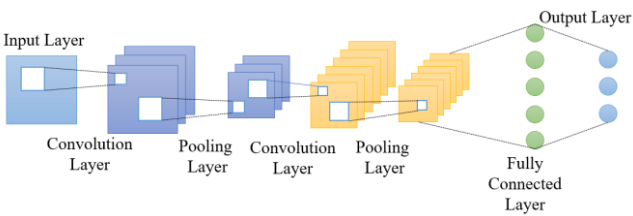
В период пандемии исследователи Центра диагностики и телемедицины собрали самую большую в мире базу данных пациентов с признаками коронавирусной пневмонии. База собиралась в сжатые сроки, что было вызвано необходимостью автоматизации процесса распознавания патологических изменений в легких при коронавирусной инфекции. На основе датасета, представленного в открытом доступе, разработчики всего мира могли "дообучать" и тестировать алгоритмы своих сервисов.

Выводы

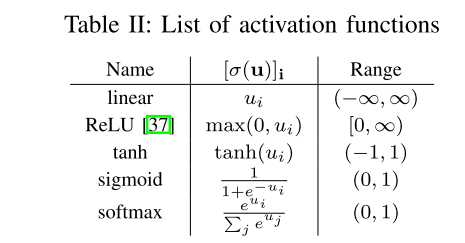
Создание больших по объему, качественных данных со стандартными характеристиками, предоставление свободного доступа к имеющимся наборам данных, позволит ускорить разработку и тестирование новых моделей. Качественный набор данных должен подкрепляться разметкой изображений экспертами. Применение единых стандартов к качеству и размеру изображений, позволит избежать трудоемкой разработки методик предварительной обработки изображений, а разрабатывать архитектуру нейронной сети.

1.4 Архитектура сверточных нейронных сетей

Фактически, архитектура сверточной нейронной сети представляет собой нейронную сеть, содержащую несколько скрытых слоев, каждый из которых имеет несколько двумерных плоскостей, состоящих из нескольких нейронов. Кроме того, все нейроны предполагаются независимыми. Его входные данные можно рассматривать как двумерное изображение, а модульизвлечения объектов встроен в архитектуру CNN. На рисунке представлена базовая архитектура сверточной нейронной сети.

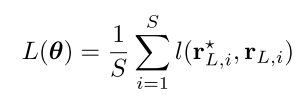
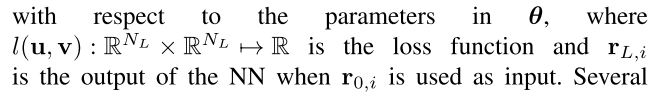


Функция активации вводит нелинейность в нейронную сеть, имитирующую функцию, при которой только нейронные электрические сигналы, превышающие определенный порог, могут быть переданы следующему нейрону. Без этой нелинейности не было бы большого преимущества в соединении нескольких слоев друг с другом. Как правило, функция активации применяется индивидуально к каждому элементу его входного вектора. Некоторые часто используемые функции активации перечислены в таблице 1

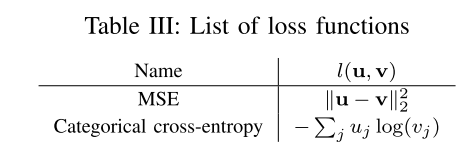


Нейронную сеть обычно обучают на размеченных данных, то есть, пар векторов ввод-вывод  

желаемый выходной сигнал нейронной сети, когда  используется в качестве входных данных. Цель обучения минимизировать значение функции потерь.

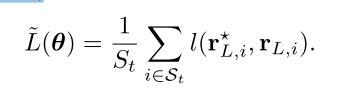
Примеры функций потерь указаны в таблице 2



Различные нормы (например, L1, L2) параметров или активаций могут быть добавлены к функции потерь в пользу решений с малыми или разреженными значениями (форма регуляризации). Наиболее популярным алгоритмом для поиска хороших наборов параметров θ является стохастический градиентный спуск (SGD), который начинается с некоторых случайных начальных значений θ = θ 0, а затем итеративно обновляет θ как



где η > 0 - скорость обучения и ˜ L(θ) - это аппроксимация функции потерь, которая вычисляется для случайной мини- партии обучающих примеров S t ⊂ {1, 2, . . . , S} размера S t на каждой итерации, т.е.,



Выбирая S t малым по сравнению с S, сложность вычисления градиента значительно снижается при одновременном снижении дисперсии обновления веса. Cуществует множество вариантов алгоритма SGD, которые динамически адаптируют скорость обучения для улучшения сходимости.

Определения и обучение нейронных сетей различных архитектур реализованы во многих существующих библиотек глубокого обучения.

В настоящее время существует множество инструментов для развертывания и обучения нейронных сетей. Такими инструментами являются библиотеки TensorFlow, Torch, MXNet и многие другие. В реестр отечественного программного обеспечения внесена библиотека PuzzleLib. Среди преимуществ PuzzleLib - высокая скорость работы, автоматическая оптимизация вычислений, большой набор нейросетевых модулей, которых насчитывается более 60 видов и ряд других.

Входной слой

Входной набор необработанных данных может быть непосредственно передан во входной слой. Одно изображение фактически вводится по его пиксельному значению во входной слой.

Сверточный слой

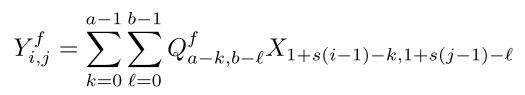
Также известен как уровень с расширенной выборкой, предназначен для извлечения объектов из входных данных. Свертка - это ключевой шаг для извлечения объектов. Каждый сверточный слой имеет свое собственное сверточное ядро, которое извлекает различные объекты из входных данных. Количество извлеченных объектов растет по мере увеличения числа сверточных ядер, включенных в слой. Выходные данные свертки называются картами объектов.

Слои сверточной нейронной сети были описаны в работе Y. LeCun et al., “Generalization and network design strategies,” Connectionism in perspective, pp. 143–155, 1989., чтобы обеспечить эффективный метод обучения для двумерных изображений. Связывая смежные сдвиги одних и тех же весов вместе способом, аналогичным тому, как фильтр скользит по входному вектору, сверточные слои способны изучать объекты с инвариантностью к сдвигам во входном векторе. Они также значительно снижают сложность модели, необходимую для представления эквивалентных объектов, инвариантных к сдвигу, с использованием полностью связанных слоев, уменьшая сложность оптимизации SGD и улучшая обобщение на соответствующих наборах данных.

В общем случае сверточный слой состоит из набора весов F фильтров

F называется глубиной

которые генерируют для каждого так называемую карту объектов из входной матрицы  исходя из следующей свертки:



Где s ≥ 1 это целое значение, называемое **шаг** число пикселей, на которое смещается окно после каждой операции, n’ = 

****, и предполагается что Х дополняется нулями  

Выходные данные свертки можно назвать картами объектов.

Выходные размеры могут быть уменьшены либо путем увеличения шага s, либо путем добавления объединяющего слоя (Слой пуллинга).

При установке ядра свертки определенного размера может потеряться информация на краю изображения. Следовательно, вводится заполнение для увеличения входных данных с нулевым значением, которое может косвенно регулировать размер. Кроме того, для контроля плотности свертывания используется шаг. Чем больше шаг, тем ниже плотность. После свертки карты объектов состоят из большого количества объектов, что может привести к проблеме переобучения.

Существует множество различных методов свертки, таких как разделяемые свертки, групповые свертки и многомерные свертки.

Слой пуллинга

Его основная функция состоит в завершении второго извлечения данных объектов, и передачи следующему сверточному слою. При нормальных условиях архитектура СНН содержит, по меньшей мере, два сверточных слоя и два слоя с понижающей дискретизацией соответственно. Чем больше уровней архитектуры задано, тем больше вероятность того, что извлечение объектов из входных данных поможет решению задачи классификации.

Объединяющий слой разбивает карту объектов на области p × p, для каждой из которых он вычисляет одно выходное значение, например, максимальное или среднее значение или L2-норму.

Полносвязный слой

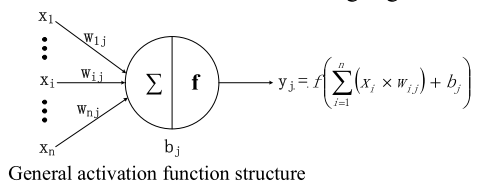
Все карты объектов соединяются вместе в качестве входных данных и представляют вектор размерности n\*1, где n это количество фильтров финального слоя. В общем случае узлы нейронов в более позднем слое соединены с узлами нейронов в предыдущем слое, но узлы в каждом слое разъединены. Этот слой интегрирует и нормализует абстрагированные функции предыдущих сверток, чтобы получить вероятность для различных условий.

Выходной слой

Количество нейронов в этом слое задается в соответствии с требуемыми условиями. Если требуется классификация, то количество нейронов обычно соотносится с количеством категорий, подлежащих классификации.

Функция активации

Сверточные нейронные сети используют различные функции активации для описания сложных объектов. Аналогично функции нейронной модели человеческого мозга, функция активации здесь представляет собой блок, который определяет, какая информация должна быть передана следующему нейрону. Каждый нейрон в нейронной сети принимает выходное значение нейронов из предыдущего слоя в качестве входных данных и передает обработанное значение следующему слою. В многослойной нейронной сети существует функция между двумя слоями. Это функция активации, её структура представлена на рисунке



На этом рисунке x i представляет входной признак; n признаков одновременно вводятся в нейрон j; w ij представляет значение веса связи между входным признаком x i и нейроном j; b j представляет внутреннее состояние нейрона j, которое является значением смещения; и yj - выходной сигнал нейрона j. 𝑓(∙) - функция активации, которая может быть сигмовидной функцией, функцией tanh (x) [10], Выпрямленной линейной единицей [52] и т.д. [53]

Если функция активации не используется или используется линейная функция, входные данные каждого слоя будут линейной функцией выходных данных предыдущего слоя. В этом случае Он и др. [38] проверяют , что независимо от того, сколько слоев имеет нейронная сеть, выходные данные всегда представляют собой линейную комбинацию входных данных, что означает, что скрытые слои не имеют никакого эффекта. Эта ситуация и есть примитивный персептрон [2], [3], который обладает ограниченной способностью к обучению. По этой причине нелинейные функции вводятся как функции активации. Теоретически, глубокие нейронные сети с нелинейной функцией активации могут аппроксимировать любую функцию, что значительно повышает способность нейронных сетей подгонять данные.

В этом разделе мы в основном сосредоточимся на нескольких часто используемых функциях активации. Начнем с того, что сигмовидная функция является одной из наиболее типичных нелинейных функций активации с общим S-образной формы (см. рис. 22(а)). При приближении значения x к 0 градиент становится более крутым. Сигмовидная функция может сопоставлять действительное число с (0, 1), поэтому ее можно использовать для задач двоичной классификации. Кроме того, SENet [48] и MobileNet v3 [46] необходимо преобразовать выходное значение в (0, 1) для механизма внимания, в котором сигмоид является хорошим способом реализации.

Отличается от сигмовидной, функция tanh [10] (см. рис. 22(б)) может сопоставить действительное число с (-1, 1). Поскольку среднее значение выходного сигнала tanh равно 0, он может достичь своего рода нормализации. Это облегчает освоение следующего уровня.

Кроме того, Выпрямленный Линейный блок (ReLU) [52] (см. Рис. 22 (c)) является еще одной эффективной функцией активации. Когда x меньше, чем 0, его функциональное значение равно 0; когда x больше или равно 0, его функциональное значение равно самому x. По сравнению с сигмовидной функцией и функцией tanh, значительным преимуществом использования функции ReLU является то, что она может ускорить обучение. Сигмоид и tanh участвуют в экспоненциальных операциях, которые требуют деления при вычислении производных, тогда как производная ReLU является константой. Более того, в сигмовидной и тангенциальной функциях, если значение x слишком велико или слишком мало, градиент функции довольно мал, что может привести к медленной сходимости функции. Однако, когда x меньше 0, производная ReLU равна 0, а когда x больше 0, производная равна 1, поэтому можно получить идеальный эффект сходимости. AlexNet [11], лучшая модель в ILSVRC- 2012, использует ReLU в качестве функции активации модели на основе CNN, которая устраняет проблему исчезновения градиента, когда сеть глубокая, и проверяет, что использование ReLU превосходит sigmoid в глубоких сетях.

Из того, что обсуждалось выше, мы можем сделать вывод, что ReLU не учитывает верхний предел. На практике мы можем установить верхний предел, например ReLU6 [54].

Однако, когда x меньше 0, градиент ReLU равен 0, что означает, что ошибка обратного распространения будет умножена на 0, в результате чего ошибка не будет передана предыдущему слою. В этом сценарии нейроны будут рассматриваться как инактивированные или мертвые. Поэтому предлагаются некоторые улучшенные версии. Негерметичный ReLU (см. рис. 22 (d)) может уменьшить инактивацию нейронов. Когда x меньше 0, выходной сигнал негерметичного ReLU равен 𝑥/𝑎 вместо нуля, где ‘a’ - это фиксированный параметр в диапазоне (1, +∞).

Другим вариантом ReLU является PReLU [38] (см. рис. 22(е)). В отличие от протекающего ReLU, наклон отрицательной части PReLU основан на данных, а не на предопределенном. Он и др. [38] считают , что PReLU является ключом к превышению уровня классификации человека в наборе данных классификации ImageNet 2012.

Функция экспоненциальных линейных единиц (ELU) [55] (см. Рис. 22 (f)) - это еще одна улучшенная версия ReLU. Поскольку ReLU активирован неотрицательно, среднее значение его выходного сигнала больше 0. Эта проблема приведет к смещению следующего блока слоя. Функция ELU имеет отрицательное значение, поэтому среднее значение ее выходного сигнала близко к 0, что делает скорость сходимости быстрее, чем ReLU. Однако отрицательная часть - это кривая, которая требует множества сложных производных.

Влияние функции активации

-Линейная функция активации действительно приводит к наихудшей производительности. Поэтому при построении глубокой нейронной сети (более одного слоя) нам нужно добавить нелинейную функцию. Если нет, то несколько слоев, теоретически, равны одному слою.

- Среди этих функций активации скорость конвергенции сигмовидной мышцы самая низкая. Обычно конечная производительность сигмовидной мышцы не так уж и хороша. В результате, если мы ожидаем быстрой конвергенции, sigmoid - не лучшее решение.

-С точки зрения точности, ELU обладает наилучшей точностью, но лишь немного лучше, чем ReLU, Leaky ReLU и ПРеЛЮ. Что касается времени обучения, то, судя по таблице I, ELU , как правило, занимает больше времени, чем ReLU и Leaky ReLU.

-ReLU и Leaky ReLU обладают лучшей стабильностью во время тренировки, чем PReLU и ELU.

-Функции активации Tanh, PReLU и ELU с большей вероятностью приведут к колебаниям в конце тренировки.

-При обучении глубокой модели CNN с предварительно обученными весами трудно сходиться с помощью функций активации sigmoid и tanh.

- Модели, обученные Leaky ReLU и ELU, имеют лучшую точность, чем другие в экспериментах. Но иногда ELU может привести к тому, что сети ничему не научатся. Чаще всего нет, Leaky ReLU обладает лучшими характеристиками с точки зрения точности и скорости обучения.

Правила выбора функции активации

Для задач двоичной классификации последний уровень может использовать сигмоид; для задач с несколькими классификациями последний уровень может использовать softmax.

Сигмовидные и танг-функции иногда следует избегать из-за исчезновения градиента. Обычно, в скрытых слоях, ReLU или Дырявый ReLU - хороший выбор.

Если вы понятия не имеете о выборе функций активации, не стесняйтесь попробовать ReLU или Leaky ReLU.

Если в процессе обучения инактивируется много нейронов, пожалуйста, попробуйте использовать Leaky ReLU, PReLU и т.д.

Отрицательный наклон в Leaky ReLU можно установить равным 0,02, чтобы ускорить обучение.

Функция потерь

Функция потерь или функция затрат используется для вычисления расстояния между прогнозируемым значением и фактическим значением. Функция потерь обычно используется в качестве критерия обучения в задаче оптимизации. Функция потерь может быть использована со сверточными нейронными сетями для решения задач регрессии и классификации, целью которых является минимизация функции потерь. Общие функции потерь включают Среднее абсолютное Ошибка (MAE), Среднеквадратичная ошибка (MSE), Перекрестная энтропия и т.д.

Функция потерь для задач классификации

В сверточных нейронных сетях, когда дело доходит до задач классификации, приходится обрабатывать множество функций потерь.

Наиболее типичный из них, называемый потерей перекрестной энтропии, используется для оценки разницы между распределением вероятностей , полученным в результате текущего обучения, и фактическим распределением. Эта функция сравнивает прогнозируемую вероятность с фактическим выходным значением (0 или 1) в каждом классе и вычисляет штрафное значение на основе расстояния от них. Штраф является логарифмическим, поэтому функция выдает меньший балл (0,1 или 0,2) для меньших различий и больший балл (0,9 или 1,0) для больших различий.

Потеря перекрестной энтропии также называется потерей softmax, что указывает на то, что она всегда используется в CNN со слоем softmax. Например, AlexNet [11], Inception v1 [33] и ResNet [37] используют кросс-энтропийные потери в качестве функции потерь в своей оригинальной статье, что помогло им достичь самых современных результатов.

Однако перекрестная потеря энтропии имеет некоторые недостатки. Потеря перекрестной энтропии заботится только о правильности классификации, а не о степени компактности внутри одного и того же класса или о разнице между разными классами. Следовательно, для решения этой проблемы предлагается множество функций потерь.

Контрастные потери [59] увеличивают расстояние между различными категориями и сводят к минимуму расстояние внутри одних и тех же категорий. Он может быть использован для уменьшения размерности в сверточных нейронных сетях. После уменьшения размерности две выборки, которые изначально похожи, все еще похожи в пространстве объектов, в то время как две выборки, которые изначально отличаются, все еще различаются. Кроме того, контрастная потеря широко используется со сверточными нейронными сетями при распознавании лиц. Впервые он был использован в SiameseNet [60], а позже был развернут в DeepID2 [61], DeepID2+ [62] и DeepID3 [63].

После контрастивной потери триплетная потеря была предложена Schroff et al. в FaceNet [64], с помощью которой модель CNN может лучше изучать вложения лиц. Определение функции триплетных потерь основано на трех изображениях. Эти три образа - якорный образ, позитивный образ и негативный образ. Позитивный образ и якорный образ принадлежат одному и тому же человеку, в то время как негативный образ и якорный образ принадлежат разным людям. Минимизация триплетных потерь заключается в том, чтобы уменьшить расстояние между якорем и положительным якорем и увеличить расстояние между якорем и отрицательным якорем. Потеря триплетов обычно используется со сверточными нейронными сетями для детальной классификации на индивидуальном уровне, что требует , чтобы модель обладала способностью различать разных людей из одной и той же категории. Сверточные нейронные сети с потерей триплетов или их варианты могут быть использованы в задачах идентификации, таких как идентификация лиц. [65], [66], [64], повторная идентификация личности [67], [68] и повторная идентификация транспортного средства [69].

Другим является потеря центра [70], которая является улучшением , основанным на перекрестной энтропии. Цель потери центра состоит в том, чтобы сосредоточиться на равномерности распределения внутри одного и того же класса. Чтобы сделать его равномерно распределенным вокруг центра класса, потеря центра добавляет дополнительное ограничение, чтобы минимизировать разницу внутри класса. Потеря центра использовалась с CNN при распознавании лиц [70], извлечении изображений [71], повторной идентификации человека. [72], распознавание динамиков [73] и т.д.

Другим вариантом перекрестной энтропии является потеря softmax с большой маржой [74]. Целью этого также является внутриклассовое сжатие и межклассовое разделение. Потеря softmax с большой маржой добавляет разницу между различными классами и вводит регулярность маржи через угол матрицы весовых ограничений. Крупномасштабные потери softmax использовались при распознавании лиц [74], распознавании эмоций [75], проверке говорящего [76] и т.д.

Правила выбора

• При использовании моделей CNN для решения проблем регрессии мы можем выбрать потери L1 или потери L2 в качестве функции потерь. • При решении задач классификации мы можем выбрать остальные функции потерь. • Перекрестная потеря энтропии - самый популярный выбор, обычно появляющийся в моделях CNN со слоем softmax в конце. • Если речь идет о компактности внутри класса или о разнице между различными классами, можно рассмотреть улучшения, основанные на потере перекрестной энтропии, такие как потеря центра и потеря softmax с большим запасом. • Выбор функции потерь в CNNs также зависит от сценария применения. Например, когда дело доходит до распознавания лиц, в настоящее время наиболее часто используются контрастные потери и триплетные потери.

Оптимизатор

В сверточных нейронных сетях нам часто приходится оптимизировать невыпуклые функции. Математические методы требуют огромных вычислительных мощностей, поэтому в процессе обучения используются оптимизаторы для минимизации функции потерь для получения оптимальных параметров сети за приемлемое время. Распространенными алгоритмами оптимизации являются Momentum, RMSProp, Adam и др.

Существует три вида методов градиентного спуска, которые мы можем использовать для обучения наших моделей CNN: Пакетный градиентный спуск (BGD), Стохастический градиентный спуск (SGD) и Мини-пакетный градиент Спуск (MBGD).

BGD указывает , что для получения градиента для каждого обновления необходимо рассчитать целый пакет данных, который может обеспечить сходимость к глобальному оптимуму выпуклой плоскости и локальному оптимуму невыпуклой плоскости. Однако использование BGD довольно медленно, поскольку необходимо рассчитать средний градиент для всей партии образцов. Кроме того, это может быть сложно для данных, которые не подходят для вычисления в памяти. Следовательно, BGD практически не используется при обучении моделей на основе CNN на практике.

Напротив, SGD использует только один образец для каждого обновления. Очевидно, что время SGD для каждого обновления значительно меньше , чем BGD, поскольку для вычисления требуется только один градиент выборки . В этом случае SGD подходит для онлайн-обучения [77]. Однако SGD быстро обновляется с высокой дисперсией, что приведет к серьезным колебаниям целевой функции. С одной стороны, колебание вычисления может привести к тому, что вычисление градиента выйдет за пределы локального оптимума и, наконец , достигнет лучшей точки; с другой стороны, SGD может никогда не сходиться из-за бесконечных колебаний.

На основе BGD и SGD был предложен MBGD, который сочетает в себе преимущества BGD и SGD. MBGD использует небольшую партию выборок для каждого обновления, так что он может не только выполнять более эффективный градиентный анализ, чем BGD, но и уменьшать дисперсию, делая сходимость более стабильной.

Среди этих трех методов MBGD является наиболее популярным. Многие классические модели CNN используют его для обучения своих сетей в оригинальных статьях, таких как AlexNet [11], VGG [32], Inception v2 [34], ResNet [37] и DenseNet [78]. Он также был использован в FaceNet [64], DeepID [79] и DeepID2 [61].

Другим часто используемым оптимизатором является Adaptive Moment Оценка (Adam) [91] По сути, это алгоритм, сформированный путем объединения импульса и RMSProp. Adam хранит как среднее значение экспоненциального спада прошлых квадратных градиентов, как алгоритм Adadelta, так и среднее среднее значение экспоненциального спада прошлых градиентов, как алгоритм Momentum. Практика доказала, что алгоритм Adam хорошо работает во многих задачах и применим ко многим различным структурам сверточных нейронных сетей [92], [93], [88].

Почти все оптимизаторы, которые мы тестировали, могут заставить модель на основе CNN сходиться в конце обучения. • Скорость сходимости мини-пакетного градиентного анализа является самой низкой, даже если она может сходиться в конце.

Правила

Мини-пакет следует использовать для того, чтобы найти компромисс между вычислительными затратами и точностью каждого обновления.

Производительность оптимизаторов тесно связана с распределением данных, поэтому, пожалуйста, не стесняйтесь пробовать различные оптимизаторы , упомянутые выше.

Если возникают чрезмерные колебания или расхождения, хорошим выбором может быть снижение скорости обучения.

1.5 Методы построения ансамбля

БУстинг

Бэггинг

….

1.6 Техническая реализация модели

Модель реализована на языке Python, поскольку используемые библиотеки и алгоритмы машинного обучения реализованы также на этом языке программирования.

Модель использует библиотеки: tensorflow для построения и обучения алгоритмов машинного обучения.

Вычисления производились на графической карте Nvidia GeForce GTX 1660 TI С 6 гигабайтами видеопамяти и на удалённом сервере предоставляемого Google Colaboratory.