

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
АННОТАЦИЯ.....	3
ВВЕДЕНИЕ	4
ПОСТАНОВКА ЗАДАЧИ	6
1 ОБЗОР ИЗВЕСТНЫХ МЕТОДОВ И СРЕДСТВ РЕШЕНИЯ	
ПРОБЛЕМЫ.....	7
1.1 ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ.....	8
1.2 МЕТОД ОПОРНЫХ ВЕКТОРОВ	8
1.3 БАЙЕСОВСКИЙ КЛАССИФИКАТОР	10
1.4 ДЕРЕВО РЕШЕНИЙ.....	11
1.5 МЕТОДЫ РАБОТЫ С ИЗОБРАЖЕНИЯМИ МРТ.....	11
2 ПОСТРОЕНИЕ РЕШЕНИЯ	12
2.1 РАБОТА С КАТЕГОРИАЛЬНЫМИ ПРИЗНАКАМИ	12
2.2 НЕЙРОННЫЕ СЕТИ	12
2.3 АВТОКОДИРУЮЩИЕ НЕЙРОННЫЕ СЕТИ	14
2.4 АРХИТЕКТУРА АВТОКОДИРУЮЩИХ НЕЙРОННЫХ СЕТЕЙ	16
2.5 ПРИМЕНЕНИЕ АВТОКОДИРУЮЩИХ НЕЙРОННЫХ СЕТЕЙ	19
3 ОПИСАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ.....	22
ЗАКЛЮЧЕНИЕ.....	24
СПИСОК ЛИТЕРАТУРЫ.....	25

АННОТАЦИЯ

При проведении операций по удалению глиом головного мозга вводится вещество, которое заставляет опухоль излучать свет, флуоресцировать. Данный эффект позволяет врачам проще и точнее удалять опухоль, так как она лучше контрастирует на фоне остального мозга во время операции. Флуоресценция не всегда проявляется, что приводит к бесполезной трате препарата.

Данная работа посвящена исследованию методов предсказания флуоресценции глиом головного мозга после ввода 5-аминолевулиновой кислоты. Для этого используются методы машинного обучения и нейросети, а в качестве данных для модели снимки магнитно-резонансной томографии и дополнительные атрибуты.

ВВЕДЕНИЕ

На данный момент рак является одной из самых опасных и трудноизлечимых болезней. Удаление раковой опухоли головного мозга является крайне сложной задачей, так как любая ошибка может привести к повреждению центральной нервной системы, вплоть до летального исхода. Для более точного удаления опухоли используется эффект флуоресценции опухоли после ввода 5-аминолевулиновой кислоты.

В случае если эффект не проявится во время операции, задача точного удаления опухоли многократно усложнится. Не знание сработает ли данный метод вводит врача в стрессовую ситуацию, которой стоит избежать. Для этого требуется предсказать свечение опухоли до операции по снимкам магнитно-резонансной томографии.

Для решения данной задачи используются методы машинного обучения и нейросети. Данная задача является задачей бинарной классификации. Задано множество объектов, для которых известна целевая переменная – свечение опухоли. Так же для этих объектов имеется информация в виде снимков магнитно-резонансной томографии, а также дополнительных атрибутов. Требуется построить модель, которая разделит объекты на классы.

Процесс разработки модели включает в себя разбор исходных данных, нахождение закономерностей, выбор алгоритма, подбор параметров, выбор метрики, определяющей качество модели, обучение моделей и выбор лучшей.

При разборе исходных данных изучаются признаки. Проводится нормализация, очистка данных от аномалий, заполнение пропущенных значений. Так же рассматриваются особенности признаков и возможно проводятся специальные преобразования.

Алгоритмы выбираются в соответствии с задачей и особенностями данных. Алгоритмы имеют гиперпараметры, которые подбираются для достижения оптимального результата в конкретной задаче, как правило экспериментальным путем.

Метрика качества определяется в соответствии с задачей и особенностями распределения целевой переменной. После этого модели обучаются – настраиваются для решения конкретной задачи. Для обученных моделей вычисляется метрика и выбирается модель с лучшим качеством решения задачи.

Для решения задач машинного обучения как правило используется язык программирования python. Для данного языка существует множество фреймворков для машинного обучения таких как numpy, sklearn. Для создания нейросети существует фреймворк pytorch. Данные фреймворки использовались в этой работе.

ПОСТАНОВКА ЗАДАЧИ

Целью данной магистерской диссертации является разработка подхода для предсказания флуоресценции глиом головного мозга по данным снимка магнитно-резонансной томографии пациента с использованием данных обучающей выборки.

Для достижения поставленной цели необходимо решить следующие задачи:

1. исследовать существующие методы и алгоритмы в области анализа данных, оценить исследованные методы на предмет возможности их использования в отношении снимков МРТ;
2. разработать подход для предсказания флуоресценции глиом головного мозга по набору в основном категориальных признаков, описывающих пациента;
3. разработать подход для предсказания флуоресценции глиом головного мозга по данным снимка магнитно-резонансной томографии;
4. реализовать на высокоуровневом языке программирования python полученный подходы.

1 ОБЗОР ИЗВЕСТНЫХ МЕТОДОВ И СРЕДСТВ РЕШЕНИЯ ПРОБЛЕМЫ

В данной работе решается задача классификации. Дано множество объектов, описываемых признаками, которое представляет собой набор данных. Это множество является выборкой из некоторой генеральной совокупности, например, людей больных раком мозга. Этот набор называют обучающей выборкой. Для каждого объекта известна целевая переменная – класс к которому принадлежит данный объект. Требуется выявить закономерность между объектами выборки и целевой переменной – классифицировать объекты.

Для решения данной задачи наиболее популярными являются методы машинного обучения. Машинное обучение — обширный подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться. Для решения задачи обучения в первую очередь фиксируется модель восстанавливаемой зависимости. Затем вводится функционал качества, значение которого показывает, насколько хорошо модель описывает наблюдаемые данные. Алгоритм обучения ищет такой набор параметров модели, при котором функционал качества на заданной обучающей выборке принимает оптимальное значение. Процесс настройки модели по выборке данных в большинстве случаев сводится к применению численных методов оптимизации.

Задача классификации относится к классу задач обучения с учителем. Особенностью задачи обучения с учителем является наличие правильных ответов для объектов обучающей выборки. Задача классификации отличается конечным множеством допустимых ответов. В данной работе рассматривается задача бинарной классификации, то есть ответов может быть всего два.

Далее будет рассмотрен ряд алгоритмов, решающих задачу классификации.

1.1 Логистическая регрессия

Логистическая регрессия — это алгоритм машинного обучения, вычисляющий вероятность принадлежности объекта к конкретному классу. В классической версии дано два класса 1 и 0. Идея логистической регрессии заключается в том, чтобы вычислить апостериорную вероятность попадания в класс 1 для объектов. Для каждого объекта вычисляется функция $z = \sum_i \theta_i x_i$, где θ_i — параметры модели [1]. Если значение больше нуля, то вероятность принадлежности объекта к классу 1 больше 0.5. Если значение функции меньше 0, то вероятность принадлежности к классу 0 больше 0.5. Иначе вероятность принадлежности к обоим классам равновероятна.

Чем дальше объект находится от разделяющей гиперплоскости, тем выше его вероятность нахождения в данном классе. $f(z) = \frac{1}{1 + e^{-z}}$ — логистическая функция определяющая вероятность попадания объекта в класс 1. Соответственно вероятность попадания в класс 0 равна $1 - f(z)$. Подбор параметров θ осуществляется методом максимального правдоподобия. Требуется максимизировать логарифм функции правдоподобия $L(\theta) = \prod_i P(y = y_i | x_i)$, то есть $\ln(L(\theta)) = \sum_i \ln(P(y = y_i | x_i))$. Максимизация функции правдоподобия производится методом градиентного спуска.

В итоге получается модель, предсказывающая вероятность принадлежности объекта x к классу 1.

1.2 Метод опорных векторов

Метод опорных векторов — это алгоритм машинного обучения, вычисляющий гиперплоскость, разделяющую признаковое пространство на классы. Дано два класса -1 и 1. Строится классификатор $a(x) = \text{sign}(\sum_i \theta_i x_i - \theta_0)$. Нужно вычислить параметры θ , определяющие оптимальную разделяющую гиперплоскость.

В случае если выборка линейно разделима, то существует множество гиперплоскостей, которые разделяют данную выборку. Нужно выбрать оптимальную гиперплоскость [2] Гиперплоскость является оптимальной если расстояние до ближайших объектов каждого из классов максимально.

В случае если выборка линейно разделима задача сводится к минимизации квадратичной формы при l ограничениях неравенства:

$$\langle \theta, x_i \rangle - \theta_0 \begin{cases} \leq -1 & \text{при } y_i = -1 \\ \geq 1 & \text{при } y_i = 1 \end{cases}$$

Где $x \in X_l$ множество пограничных объектов относительно переменных θ :

$$\begin{cases} \langle \theta, \theta \rangle \rightarrow \min \\ y_i(\langle \theta, x_i \rangle - \theta_0) \geq 1 \end{cases}$$

По теореме Куна-Таккера эта задача эквивалентна двойственной задаче поиска седловой точки функции Лагранжа [2]:

$$\begin{cases} \mathcal{L}(\theta; \lambda) = \frac{1}{2} \langle \theta, \theta \rangle - \sum_{i=1}^l \lambda_i (y_i(\langle \theta, x_i \rangle - \theta_0) - 1) \rightarrow \min_{\theta} \max_{\lambda} \\ \lambda_i \geq 0 \end{cases}$$

Необходимым условием седловой точки является равенство нулю производных Лагранжиана. Искомый вектор весов θ является линейной комбинацией векторов обучающей выборки, причём только тех, для которых $\lambda_i \neq 0$. Согласно условию дополняющей нежесткости на этих векторах x_i ограничения-неравенства обращаются в равенства: $\langle \theta, x_i \rangle - \theta_0 = y_i$, следовательно, эти векторы находятся на границе разделяющей полосы. Данные вектора являются опорными.

Сведем задачу к эквивалентной:

$$\left\{ \begin{array}{l} -\mathcal{L}(\lambda) = -\sum_{i=1}^l \lambda_i + \sum_{i=1}^l \sum_{j=1}^l \lambda_i \lambda_j y_i y_j (\langle x_i x_j \rangle) \rightarrow \min_{\lambda} \\ \lambda_i \geq 0 \\ \sum_{i=1}^l \lambda_i y_i = 0 \end{array} \right.$$

Допустим, мы решили эту задачу. Тогда вектор w вычисляется по формуле [2]:

$$\theta = \sum_{i=1}^l \lambda_i y_i x_i$$

В итоге алгоритм классификации может быть представлен в следующем виде:

$$a(x) = \text{sign}\left(\sum_{i=1}^l \lambda_i y_i \langle x_i, x \rangle - \theta_0\right)$$

В случае линейной неразделимости можно прибегнуть к ядерной функции. Данный метод позволяет перевести объекты из исходного признакового пространства X к новому пространству H с помощью некоторого преобразования $\psi: X \rightarrow H$. Функция $K: X \times X \rightarrow R$ называется ядром, если она представима в виде $K(x, \acute{x}) = \langle \psi(x), \psi(\acute{x}) \rangle$, где H — пространство со скалярным произведением. Ядро должно быть симметричным и неотрицательным. Алгоритм будет иметь следующий вид:

$$a(x) = \text{sign}\left(\sum_{i=1}^l \lambda_i y_i K(x_i, x) - \theta_0\right)$$

1.3 Байесовский классификатор

Байесовский классификатор — алгоритм, решающий задачу классификации, основанный на максимизации апостериорной вероятности. Для классифицируемого объекта вычисляются функции правдоподобия

каждого из классов, по ним вычисляются апостериорные вероятности классов [3]. Объект относится к тому классу, для которого апостериорная вероятность максимальна. Наиболее распространенным вариантом является наивный байесовский классификатор.

Наивный байесовский классификатор работает на предположении о независимости признаков. Тогда функции правдоподобия для классов из множества классов Y выглядят так:

$$P_y(x) = p_{y1}(x_1)p_{y2}(x_2) \dots p_{yn}(x_n)$$

Где $p_{in}(x_i)$ плотность распределения i -го признака для класса y . Тогда алгоритм классификации будет выглядеть следующим образом:

$$a(x) = \underset{y \in Y}{argmax} \lambda_y P_y P_y(x)$$

Где P_y – это апостериорная вероятность класса y , а λ_y – штраф за отнесение объекта класса y к другому классу.

1.4 Дерево решений

Дерево решений — это алгоритм машинного обучения решающий задачу классификации. Алгоритм строит дерево в листьях которого находятся классы, в остальных вершинах атрибуты, по которым принимаются решения, а в ребрах значения атрибутов по которым идет идентификация объекта [4]. При классификации производится спуск по дереву до листа в котором и содержится ответ. Дерево решений может быть построено следующим образом. В качестве корня выбирается признак с наименьшей энтропией. Для каждого значения этого признака в обучающей выборке строится ребро. В качестве вершины этого ребра выбирается либо значение класса, если все объекты выборки принадлежат этому классу, либо рекурсивно строится новое поддерево по тому же принципу, выбирается другой признак с минимальной энтропией и подмножество объектов, удовлетворяющих данному узлу.

1.5 Методы работы с изображениями МРТ

При анализе снимков магнитно-резонансной томографии часто используются нейронные сети.

Наиболее популярными видами нейронных сетей для работы со снимками магнитно-резонансной томографии являются сверточные нейронные сети [5,6] и автокодировщики нейронные сети [6,7,8,9]. Так же используются более специфичные модели, например, нейронные сети Кохонена [10]. В данной работе используются автокодировщики нейронные сети.

2 ПОСТРОЕНИЕ РЕШЕНИЯ

Для решения задачи предсказания флуоресценции глиом головного мозга институт нейрохирургии имени Н. Н. Бурденко предоставил таблицу с параметрами 320 пациентов. Так же они предоставили снимки МРТ 78 пациентов. Каждый снимок представляет собой набор из двумерных срезов головного мозга, которые вместе образуют трехмерное изображение. Для различных пациентов разрешение снимков различное. Срезы представляют собой квадратную матрицу, элементы которой являются яркостью соответствующего пикселя. С каждым снимком так же идет некоторая метаинформация, например, пол, возраст, вес.

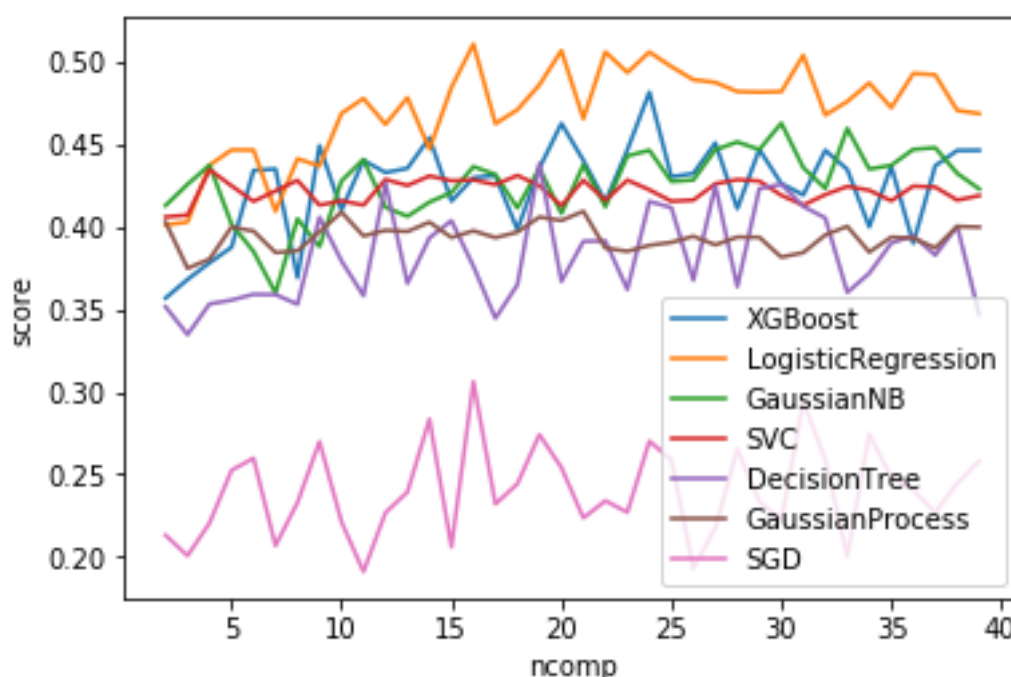
Сперва данные нужно привести к общему виду. Изображения сжимаются к единому разрешению, например, 64x64. Так же данные нужно нормализовать: значение элементов матрицы привести к интервалу от 0 до 1. Для этого все элементы матрицы делятся на значение максимального элемента этой матрицы. В данном виде каждый снимок представляет собой 4096 не осмысленных признаков.

Использовать стандартные алгоритмы машинного обучения на данных в таком виде неэффективно, поэтому предлагается снизить размерность. В данной работе для снижения размерности использовались автокодирующие нейронные сети [11].

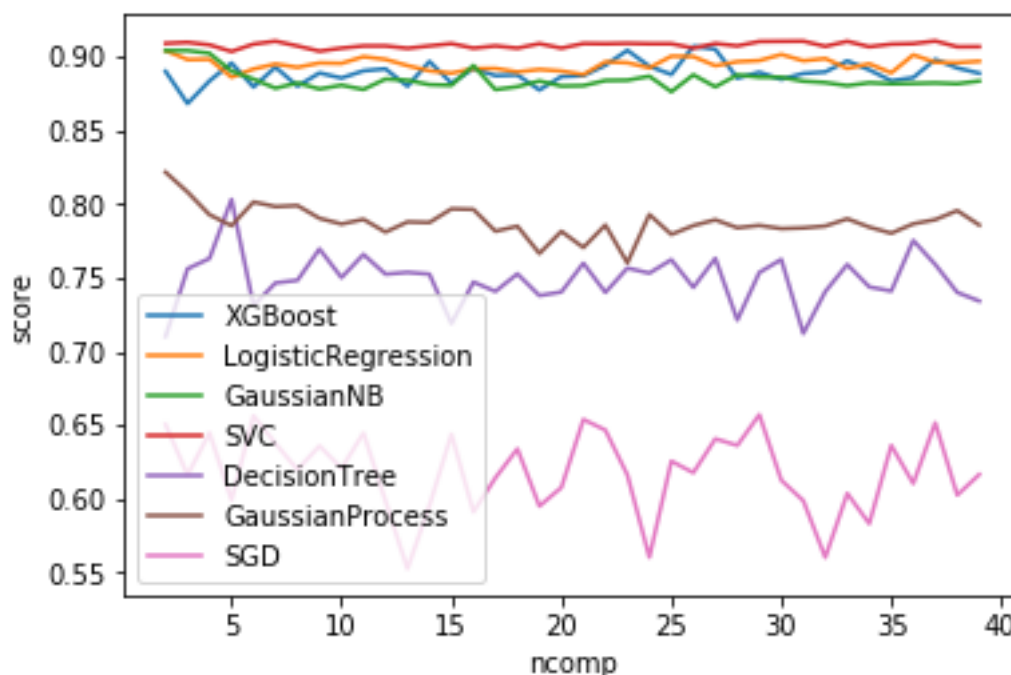
2.1 Работа с категориальными признаками.

Институт нейрохирургии имени Н. Н. Бурденко предоставил таблицу с категориальными признаками, описывающими 320 пациентов. Данные для многих пациентов были не до конца заполнены, поэтому многие столбцы пришлось исключить из исследования. Были выделены наиболее релевантные столбцы, и заменены пропущенные значения для них на 0, а также значения были приведены к числам, например, пол вместо «м» и «ж» 0 и 1.

В качестве целевой переменной был категориальный признак, принимающий 4 значения: ярко розовое свечение, розовое свечение, бледно розовое свечение, нет видимого свечения. У полученной выборки была снижена размерность с помощью метода главных компонент и к выборки сниженной размерности были применены алгоритмы машинного обучения. Наилучший результат показала логистическая регрессия с точность равной примерно 0.51.



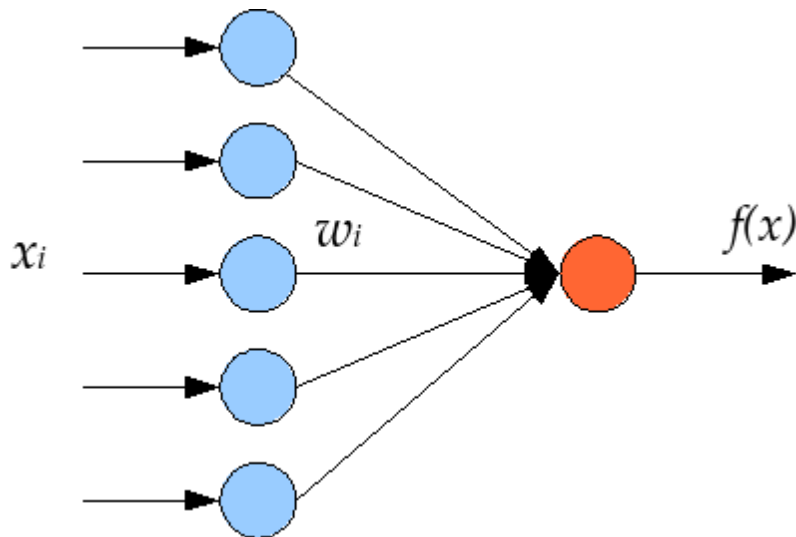
Так как целью работы было выяснить будет ли свечение, а коллег из института нейрохирургии устраивало и бледно розовое свечение, то было решено заменить целевую переменную на принимающую 2 значения: светилась, объединяющая все виды свечения и не светилась. То же самое что и ранее было проделано с новой целевой переменной. Однако классы стали не сбалансированными и в качестве меры качества использовалась мера f1.



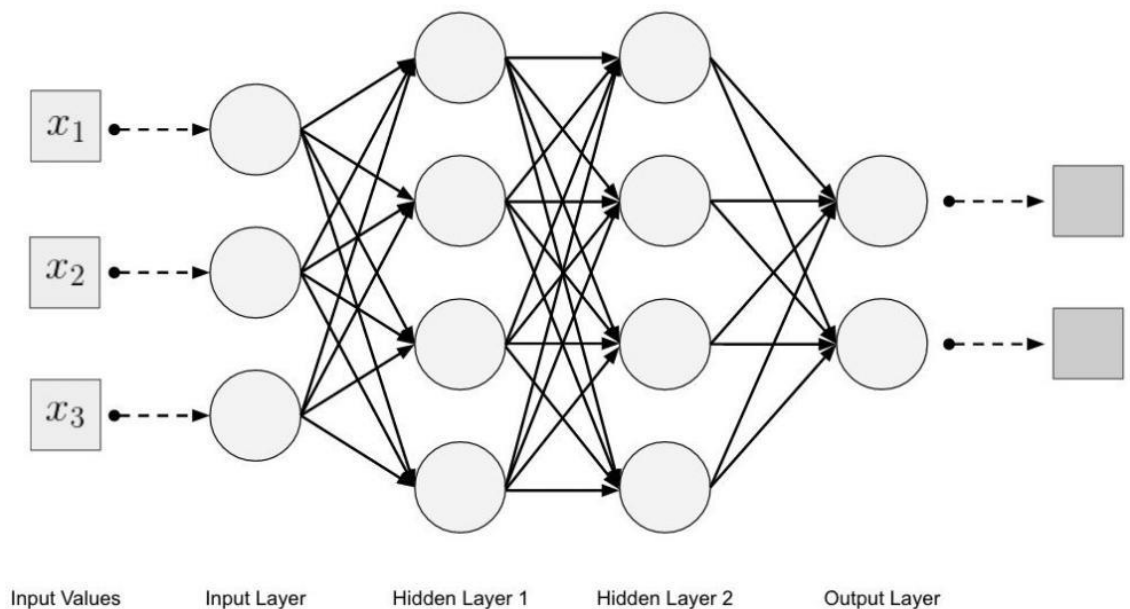
Лучший результат показал Метод опорных векторов с качеством примерно равным 0.91 по f1 мере. С помощью корреляционного теста Пирсона были выявлены наиболее значимые признаки. Одним из таких оказался возраст, использовавшийся в дальнейшем при построении модели по снимкам МРТ. Коэффициент Пирсона для этого признака составил 0.26. Еще одним значимым признаком с коэффициентом Пирсона равным 0.45 оказалась категория опухоли, но к сожалению, данный признак добавить в модель не удалось, так как он не был известен для пациентов со снимками.

2.2 Нейронные сети

Нейронная сеть это математическая модель, построенная по подобию мозга. Нейроны в мозгу получают на вход некоторое количество сигналов и выдает на выходе некоторое, возможно другое количество сигналов.



С точки зрения математики к исходному вектору параметров длины n применяется некоторая функция в результате получается результирующий вектор длины m . К каждому элементу результирующего вектора применяется функция активации $f(x)$. Функция активации является возрастающей. Таким образом простейшая нейронная сеть представляется уравнением $y = f(\langle \omega, x \rangle)$. На практике используют глубокие нейронные сети, состоящие из множества слоев.



Обучение нейронных сетей представляет собой настройку параметров. Для этого используется алгоритм обратного распространения ошибок [12].

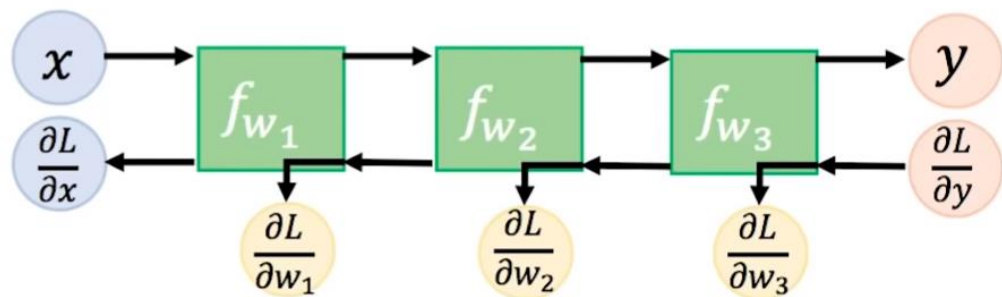
Пусть L это некая функция потерь. Функция потерь определяет насколько предсказанные значения отличаются от реальных. Задача минимизировать функцию потерь. Тогда для каждого слоя можно вычислить градиент:

$$\frac{dL}{d\omega} = \frac{dL}{dy} \frac{dy}{d\omega}$$

Для этого нужно знать производную функции потерь для следующего слоя. Поэтому так же нужно вычислить эту производную для предыдущего слоя:

$$\frac{dL}{dx} = \frac{dL}{dy} \frac{dy}{dx}$$

И передать его на предыдущий слой.

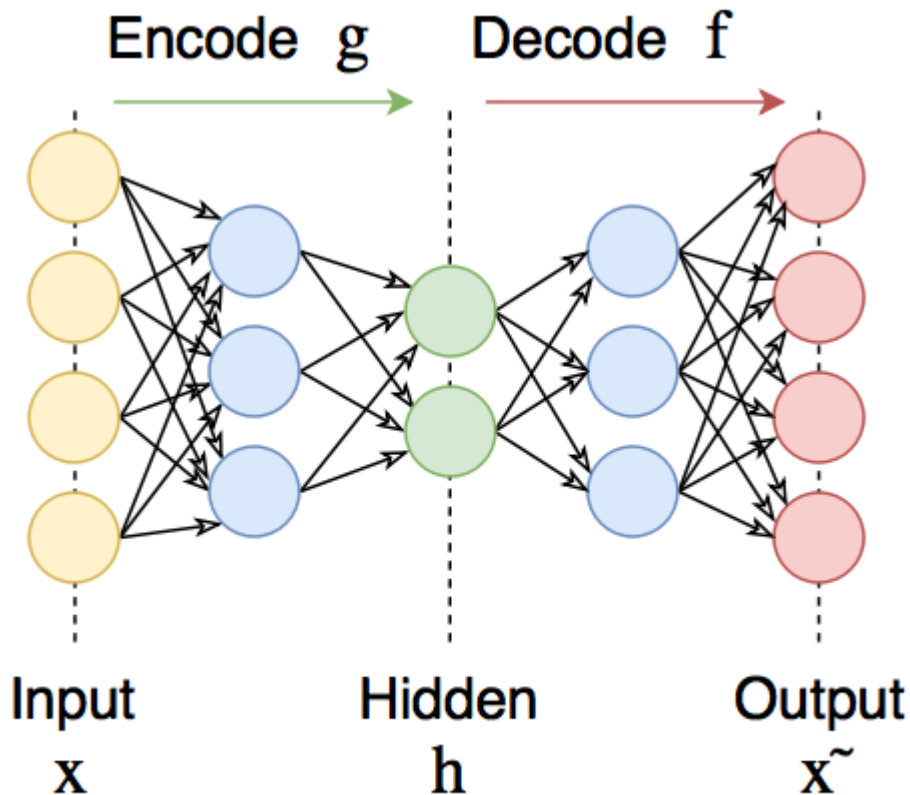


Имея градиент на каждом слое можно методом градиентного спуска обновить веса.

2.3 Автокодирующие нейронные сети

Автокодирующая нейронная сеть — это разновидность нейронных сетей, использующихся для снижения размерности [11]. Данная нейронная сеть состоит из двух компонентов: кодировщика и декодировщика, соединённых последовательно. Кодировщик сжимает данные до некоторой размерности, в то время как декодировщик восстанавливает данные из сжатых на предыдущем шаге. Данная сеть при обучении в качестве выходных данных использует

входные данные. Обучив всю сеть можно использовать только часть — кодировщик как функцию для снижения размерности. В то же время декодировщик используется только для обучения.



Так схематически выглядит автокодировщик.

Автокодирующие нейронные сети можно разделить на 4 вида:

1. Глубокие
2. Регуляризованные
3. Сверточные
4. Шумоподавляющие

Глубокие автокодирующие нейронные сети отличаются наличием нескольких скрытых слоев. В основном используются линейные операторы. Регуляризованные автокодирующие нейронные сети используют специальные приемы, штрафующие за переобучение и заставляющие

выделять более уникальные признаки. Сверточные используют операции свертки и пулинга. В шумоподавляющих к исходному изображению добавляется шум, таким образом нейронная сеть так же избавляется от шумов.

2.4 Архитектура автокодирующих нейронных сетей

Для снижения размерности были применены глубокие, регуляризованные и сверточные автокодирующие нейронные сети.

Сперва снимки из обучающей выборки были приведены к размеру 64x64.

Для начала была использована шестислойная сеть: три слоя кодировщик и три декодировщик. Архитектура представляет собой линейный слой, преобразующий размерность 4096 в 500 затем функция активации, затем снова линейный слой из 500 в 64, функция активации и снова линейный слой из 64 в целевую размерность, являющуюся параметром модели. Линейный слой имеет следующий вид:

$$linear(x) = \langle \omega, x \rangle + \omega_0$$

Декодировщик представляет собой зеркальное отражение кодировщика. Функция активации для кодировщика была взята сигмоида. Для декодировщика тангенсоида.

$$sigm(x) = \frac{1}{1 + e^{-x}}, tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Далее была использована более глубокая сеть. В ней было 12 слоев по 6 на кодировщик и декодировщик. Архитектура кодировщика имеет следующий вид: линейный слой 4096x2000, функция активации, линейный слой 2000x1100, функция активации, линейный слой 1100x500, функция активации, линейный слой 500x300, функция активации, линейный слой 300x100, функция активации, линейный слой 100 – целевая размерность. Декодировщик также представляет собой зеркальное отражение кодировщика. Функции активации используются те же что и в предыдущей сети.

Далее были применены регуляризованные автокодирующие нейронные сети. В качестве регуляризованной автокодирующей нейронной сети использовалась k -разряженная сеть [13,14]. Особенность данной сети заключается в том, что на последнем слое кодировщика остаются только k признаков, имеющих наибольшее значение, остальные зануляются. Данный прием заставляет нейросеть выучивать более уникальные признаки и снижает риски переобучения, когда нейросеть выучивает среднее изображение.

Сперва была построена шестислойная k -разряженная нейросеть. Архитектура кодировщика имеет следующий вид: линейный слой 4096×1000 , функция активации, линейный слой 1000×500 , функция активации, линейный слой 500×100 , далее зануляются все кроме k наибольших значений. Декодировщик представляет собой зеркальное отражение кодировщика, за исключением зануляющего слоя. Функции активации для кодировщика – сигмоиды, для декодировщика тангенсоиды.

Далее была построена более глубокая десятислойная k -разряженная нейросеть. Архитектура кодировщика имеет следующий вид: линейный слой 4096×2000 , функция активации, линейный слой 2000×1100 , функция активации, линейный слой 1100×500 , функция активации, линейный слой 500×300 , функция активации, линейный слой 300×100 , далее зануляются все кроме k наибольших значений. Декодировщик представляет собой зеркальное отражение кодировщика, за исключением зануляющего слоя. Функции активации для кодировщика – сигмоиды, для декодировщика тангенсоиды.

Также были построены похожие нейросети для снимков более высокой размерности 128×128 . Для этого снимки были приведены к размерности 128×128 .

Сперва десятислойная сеть. Архитектура кодировщика имеет следующий вид: линейный слой 16384×5000 , функция активации, линейный слой 5000×1000 , функция активации, линейный слой 1000×300 , функция активации, линейный слой 300×60 , функция активации, функция активации,

линейный слой 60 – целевая размерность. Декодировщик также представляет собой зеркальное отражение кодировщика. Функции активации используются те же что и в предыдущей сети.

Далее была использована более глубокая сеть. В ней было 14 слоев по 7 на кодировщик и декодировщик. Архитектура кодировщика имеет следующий вид: линейный слой 16384x10000, функция активации, линейный слой 10000x6000, функция активации, линейный слой 6000x2000, функция активации, линейный слой 2000x800, функция активации, линейный слой 800x300, функция активации, линейный слой 300x100, функция активации, линейный слой 100 – целевая размерность. Декодировщик также представляет собой зеркальное отражение кодировщика. Функции активации используются те же что и в предыдущей сети.

Далее была построена восьмислойная k-разряженная нейросеть. Архитектура кодировщика имеет следующий вид: линейный слой 16384x5000, функция активации, линейный слой 5000x1000, функция активации, линейный слой 1000x300, линейный слой 300x100, далее зануляются все кроме k наибольших значений. Декодировщик представляет собой зеркальное отражение кодировщика, за исключением зануляющего слоя. Функции активации для кодировщика – сигмоиды, для декодировщика тангнсоиды.

Далее была построена более глубокая двенадцатислойная k-разряженная нейросеть[10]. Архитектура кодировщика имеет следующий вид: линейный слой линейный слой 16384x10000, функция активации, линейный слой 10000x6000, функция активации, линейный слой 6000x2000, функция активации, линейный слой 2000x800, функция активации, линейный слой 800x300, функция активации, линейный слой 300x100, далее зануляются все кроме k наибольших значений. Декодировщик представляет собой зеркальное отражение кодировщика, за исключением зануляющего слоя. Функции активации для кодировщика – сигмоиды, для декодировщика тангнсоиды.

Так же была построена сверточная автокодирующая нейронная сеть. Сверточная сеть отличается использованием операции свертки и пулинга.

Архитектура сверточного кодировщика имеет следующий вид: свертка 3x3, макспулинг 4x4, свертка 3x3, макспулинг 2x2, свертка 3x3, макспулинг 4x4, линейный слой 4x4, макспулинг 4. Декодировщик имеет зеркальный вид, за исключением того, что пулинг заменен аналогичным анпулингом.

2.5 Применение нейронных сетей

После того как данные сети были обучены снимки были сжаты с помощью кодировщиков данных сетей. В результате получился новый набор данных. К данному набору были добавлены дополнительные признаки: возраст, вес и пол. На полученном наборе данных были обучены алгоритмы машинного обучения.

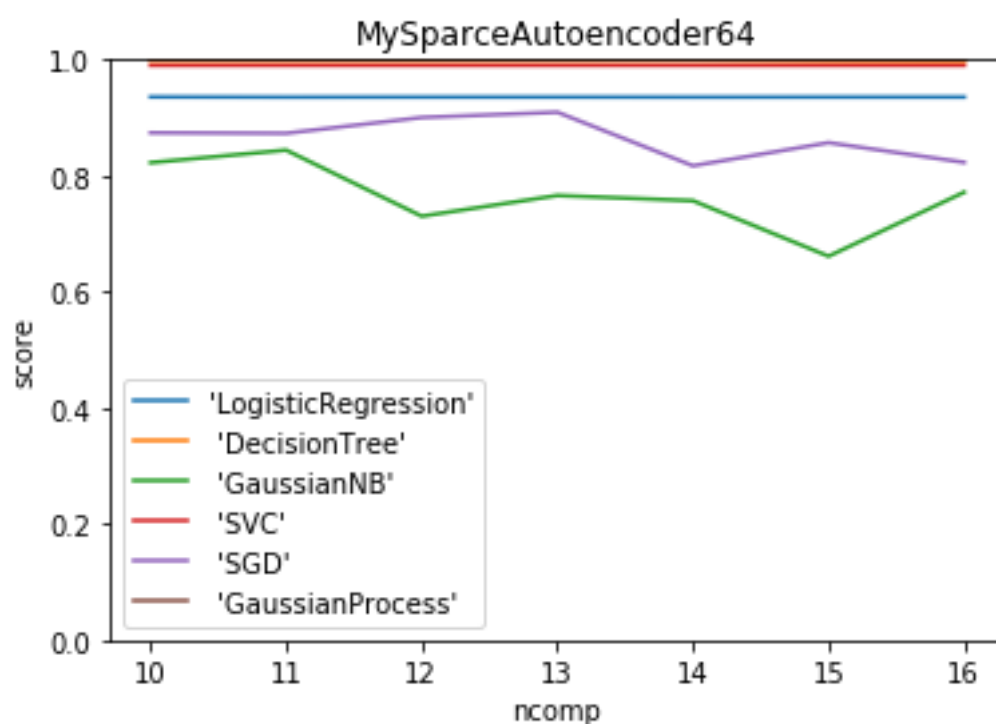
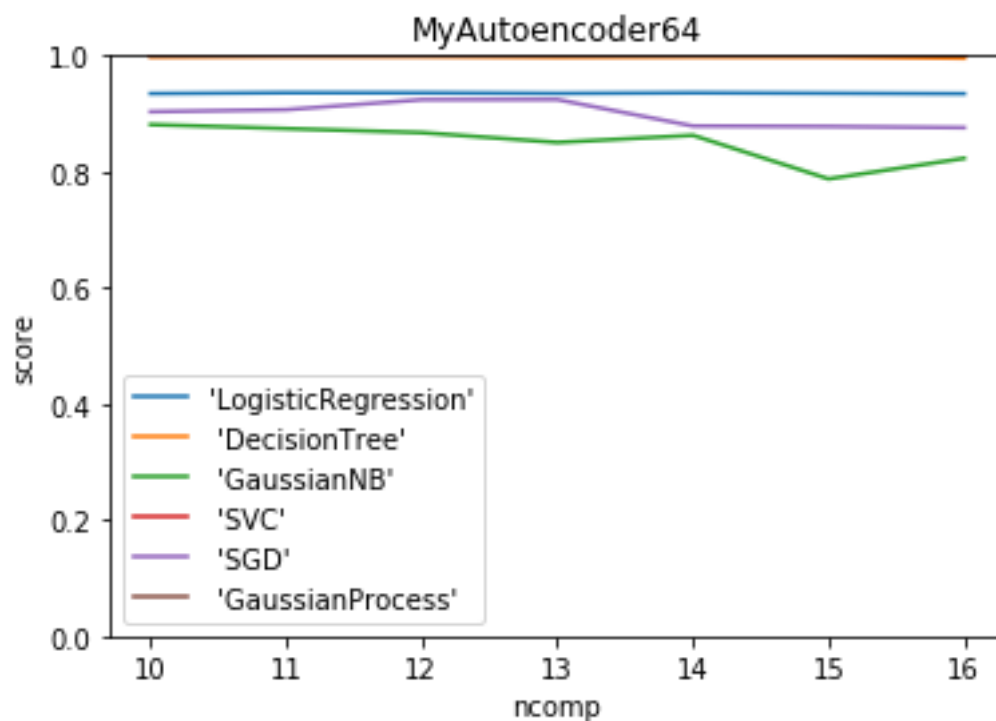
Для наиболее релевантной оценки качества моделей использовался метод кроссвалидации. При кроссвалидации выборка перемешивается и разбивается на несколько частей. Каждая часть используется для проверки, в то время как остальные для обучения. Средняя оценка качества по всем тестовым частям считается общей оценкой качества.

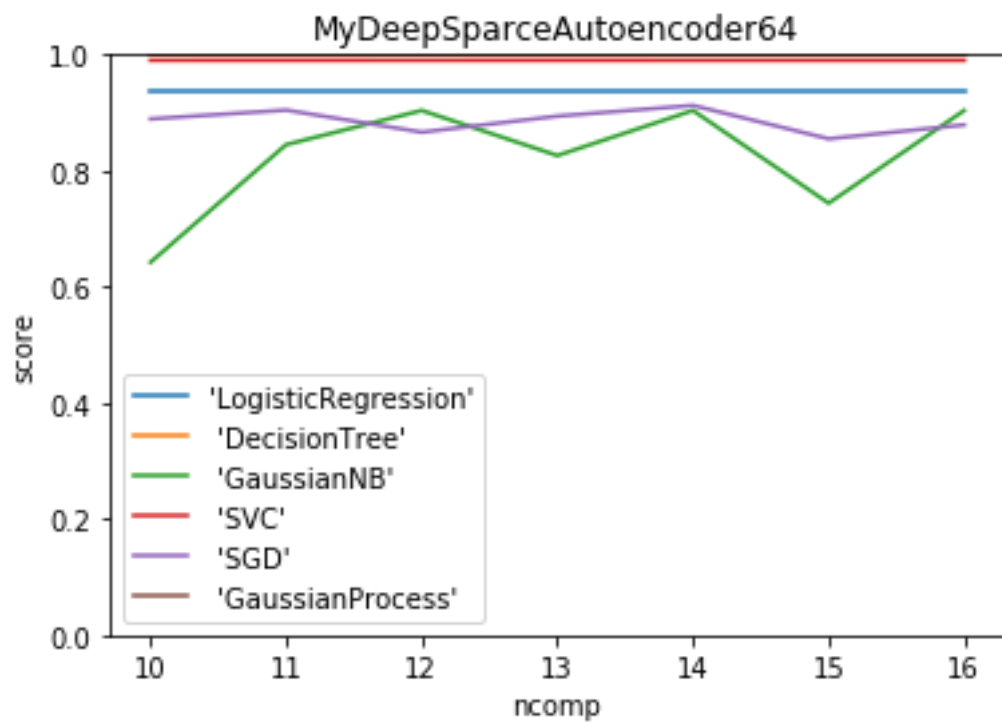
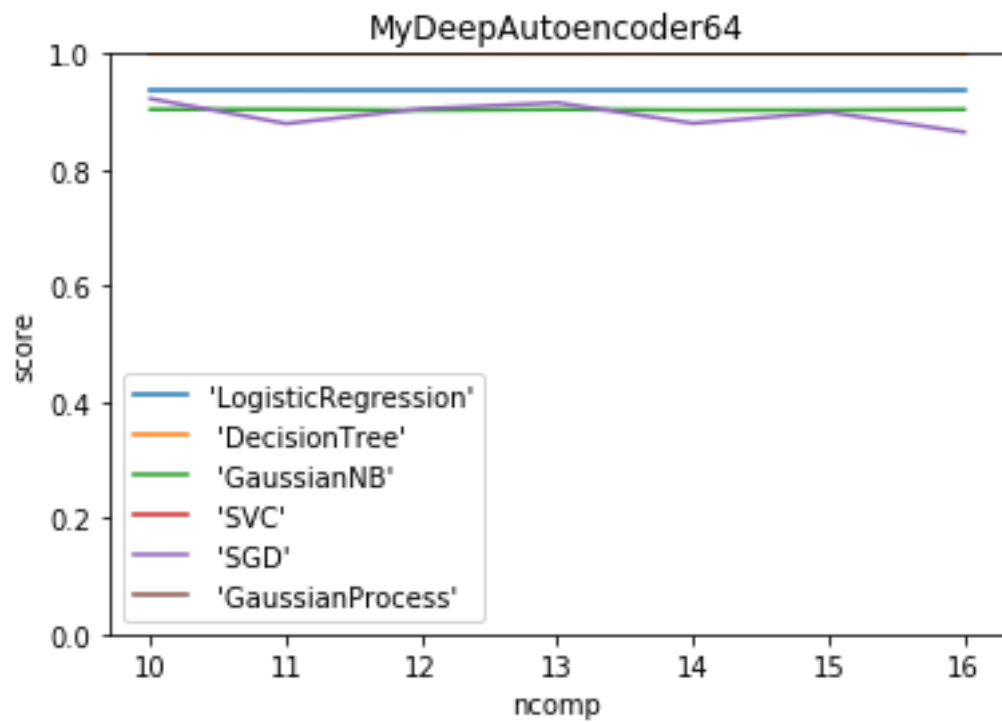
Для оценки качества простая точность предсказания, вычисляемая как доля правильных ответов не подходит для данной задачи, так как наблюдается дисбаланс классов: доля класса 1 составляет 87% и если модель будет плохо предсказывать меньший по объему класс, то в данной метрике это не будет отражено. Мерой качества в данном случае использовалась f1 мера. F1 мера вычисляется по следующей формуле:

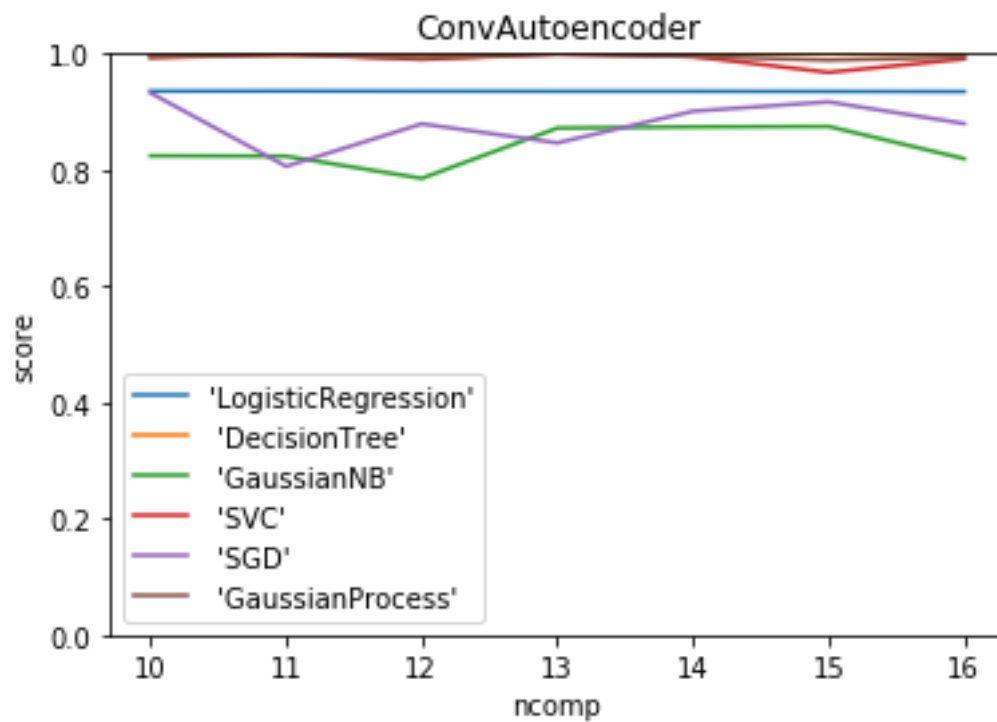
$$f1 = 2 \frac{precision \cdot recall}{precision + recall}, precision = \frac{TP}{TP + FP}, recall = \frac{TP}{TP + FN}$$

Где TP – кол-во правильно определенных объектов одного из классов, FP – количество объектов, принадлежащих этому классу, но определенных неправильно, FN – количество объектов, принадлежащих другому классу, но ошибочно определенных к данному. Данный подход позволяет учесть дисбаланс классов.

В результате применения нейронных сетей и обучения алгоритмов машинного обучения на сжатых данных были получены следующие результаты:







В результате можно сделать вывод, что метод опорных векторов отлично показал себя на всех данных и добился почти идеального предсказания.

3 ОПИСАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ

Данное исследование было проведено с использованием языка программирования python. Для создания и обучения нейронных сетей использовался фреймворк pytorch [15]. Алгоритмы машинного обучения были взяты из библиотеки sklearn.

Снимки были представлены в формате dicom – медицинский отраслевой стандарт хранения изображений. Трехмерные изображения представляют собой набор dicom файлов, каждый из которых является двухмерным изображением, срезом головного мозга. Каждый файл сопровождается набором атрибутов, таких как пол, возраст, вес, тип изображения и ряд других, не значимых для данного исследования. Во всем наборе файлов для каждого пациента имеются не только снимки МРТ. Поэтому стояла задача отфильтровать ненужные изображения и выбрать только снимки МРТ. Для работы с данным форматом данных использовалась библиотека pydicom [16]. Фильтровались снимки по атрибуту типа снимка.

Изображения приводились к нужному размеру с помощью библиотеки PIL, позволяющей менять размер изображений.

Обучение нейронных сетей проходило в 300 эпох. Эпохи – это итерации обратного распространения ошибок по тестовой выборке. Обратное распространение ошибок производится партиями по 50 объектов. Выборка предварительно перемешивается и делится на тестовую, составляющую 25% от общего объема и тренировочную выборку. В качестве функции потерь использовалась среднеквадратичная ошибка (MSE).

$$MSE = \frac{\sum_{i=1}^n (y_i - \bar{y}_i)^2}{n}$$

В качестве оптимизатора использовался ADAM – метод стохастической оптимизации [17].

Исследование было проведено на кластере, доступ к которому предоставил Институт проблем информатики ФИЦ ИУ РАН. Расчеты

проводились на видеокартах Nvidia с поддержкой технологии cuda. Полное обучение всех моделей составило примерно 3 дня. Код, который использовался для обучения и построения моделей доступен на репозитории github [18].

ЗАКЛЮЧЕНИЕ

В рамках данной работы проведено исследование об эффективности применения автокодирующих нейронных сетей при предсказании флуоресценции глиом головного мозга по снимкам магнитно-резонансной томографии. Некоторые из полученных моделей показали высокое качество предсказания свечения опухолей головного мозга. Лучшее всего себя показал метод опорных векторов, давший 99% результат для всех вариантов автокодирующих нейронных сетей. Данный метод в комбинации с разряженной нейронной сетью и предлагается использовать на практике.

Были решены следующие задачи:

1. исследованы существующие методы и алгоритмы в области анализа данных, оценены исследованные методы на предмет возможности их использования в отношении снимков МРТ
2. разработан подход для предсказания флуоресценции глиом головного мозга по набору в основном категориальных признаков, описывающих пациент.
3. разработан подход для предсказания флуоресценции глиом головного мозга по данным снимка магнитно-резонансной томографии.
4. реализован на высокоуровневом языке программирования python полученный подходы.

СПИСОК ЛИТЕРАТУРЫ

1. Kleinbaum D. G. et al. Logistic regression. – New York : Springer-Verlag, 2002.
2. Воронцов К. В. Лекции по методу опорных векторов (<http://www.ccas.ru/voron/download/SVM.pdf>).
3. Kononenko I. Semi-naive Bayesian classifier //European Working Session on Learning. – Springer, Berlin, Heidelberg, 1991. – С. 206-219.
4. Quinlan J. R. Induction of decision trees //Machine learning. – 1986. – Т. 1. – №. 1. – С. 81-106.
5. Pereira S. et al. Brain tumor segmentation using convolutional neural networks in MRI images //IEEE transactions on medical imaging. – 2016. – Т. 35. – №. 5. – С. 1240-1251.
6. Vu T. D. et al. Multimodal learning using convolution neural network and Sparse Autoencoder //2017 IEEE International Conference on Big Data and Smart Computing (BigComp). – IEEE, 2017. – С. 309-312.
7. Shin H. C. et al. Autoencoder in time-series analysis for unsupervised tissues characterisation in a large unlabelled medical image dataset //2011 10th International Conference on Machine Learning and Applications and Workshops. – IEEE, 2011. – Т. 1. – С. 259-264.
8. Chen X., Konukoglu E. Unsupervised Detection of Lesions in Brain MRI using constrained adversarial auto-encoders //arXiv preprint arXiv:1806.04972. – 2018.
9. Majumdar A. Real-time dynamic MRI reconstruction using stacked denoising autoencoder //arXiv preprint arXiv:1503.06383. – 2015.
10. Chuang K. H. et al. Model-free functional MRI analysis using Kohonen clustering neural network and fuzzy C-means //IEEE transactions on medical imaging. – 1999. – Т. 18. – №. 12. – С. 1117-1128.
11. Wang Y., Yao H., Zhao S. Auto-encoder based dimensionality reduction //Neurocomputing. – 2016. – Т. 184. – С. 232-242.
12. Hecht-Nielsen R. Theory of the backpropagation neural network //Neural networks for perception. – Academic Press, 1992. – С. 65-93.

13. Makhzani A., Frey B. K-sparse autoencoders //arXiv preprint arXiv:1312.5663. – 2013..
14. DEEP LEARNING: SPARSE AUTOENCODERS. URL: <http://www.ericlewilkinson.com/blog/2014/11/19/deep-learning-sparse-autoencoders>.
15. Pytorch documentation. URL: <https://pytorch.org/docs/stable/index.html>.
16. Pydicom documentation. URL: <https://pydicom.github.io/>.
17. Kingma D. P., Ba J. Adam: A method for stochastic optimization //arXiv preprint arXiv:1412.6980. – 2014.
18. Github. URL: <https://github.com/LavrVV/Fluorescence-prediction>