

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
КАФЕДРА МОДЕЛИРОВАНИЯ ЭЛЕКТРОМЕХАНИЧЕСКИХ И  
КОМПЬЮТЕРНЫХ СИСТЕМ

Селихова Анастасия Владимировна

Магистерская диссертация

Нейросетевая обработка разнородных данных  
экологических происшествий

Направление 01.04.02

"Прикладная математика и информатика"

Магистерская программа "Прикладные информационные технологии.

Информационные экспертные системы"

Научный руководитель,  
кандидат физ.-мат. наук,  
доцент  
Степенко Н. А.

Рецензент,  
«Национальный исследовательский  
университет ИТМО» —  
ITMO University  
доктор физ.-мат. наук,  
доцент,  
Холодова С. Е.

Санкт-Петербург

2020

# Содержание

<b>Введение</b>	<b>3</b>
<b>Постановка задачи</b>	<b>5</b>
<b>Обзор литературы</b>	<b>6</b>
<b>Глава 1. Предварительная обработка данных о пожарах в РФ</b>	<b>9</b>
§1. Источник данных о пожарах . . . . .	9
§2. Описание полученных данных . . . . .	13
§3. Обработка данных по сетке . . . . .	18
<b>Глава 2. Нейросетевая обработка данных</b>	<b>19</b>
§1. Типы нейронных сетей . . . . .	19
§2. Функция потерь . . . . .	23
§3. Выбор параметров модели . . . . .	24
§4. Обучение нейронной сети . . . . .	25
<b>Глава 3. Добавление данных о погоде и улучшение предсказаний нейронной сети</b>	<b>28</b>
§1. Загрузка и обработка данных . . . . .	28
§2. Нахождение расстояния между двумя точками . . . . .	31
§3. Нейросетевая обработка дополненных данных . . . . .	33
<b>Выводы</b>	<b>35</b>
<b>Заключение</b>	<b>36</b>
<b>Список литературы</b>	<b>37</b>
<b>Приложение</b>	<b>40</b>

# Введение

Лесные пожары — неизбежная часть жизни на Земле. Ежегодно повреждаются огнем более 340 млн. га природных территорий (включая леса). Наибольшие площади ежегодно сгорающих лесов находятся в Австралии и странах Африки. По общей площади лесов, уничтоженных пожарами, Россия занимает 8 место среди стран мира [1]. Ознакомиться со статистикой по лесным пожарам в Российской Федерации можно, используя Единую Межведомственную Информационно-Статистическую систему (ЕМИСС) [?], где представлены данные от Федерального агентства лесного хозяйства. Сформированные данные по площади сгоревших лесных насаждений за 2015–2018 года графически представлены на рис. 1 а).

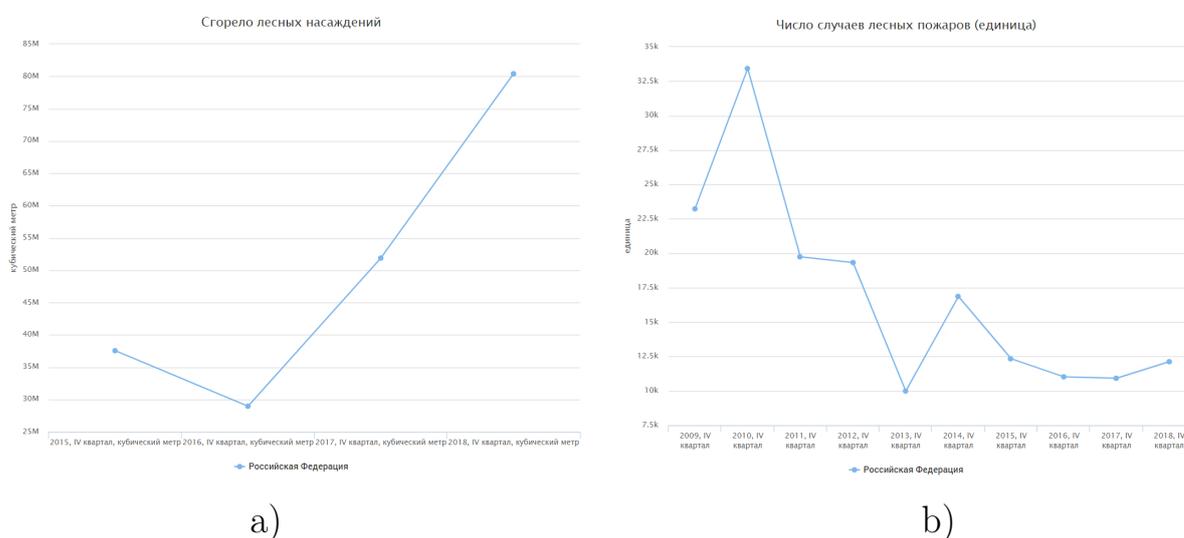


Рис. 1: а) Количество сгоревших лесных насаждений по данным Федерального агентства лесного хозяйства за 2015–2018 гг.; б) Число случаев лесных пожаров по данным Федерального агентства лесного хозяйства

Здесь виден высокий рост общей площади сгоревших лесов, несмотря на то, что, как видно на рис. 1 б), показатель количества пожаров меняется не столь значительно. В 2018 году по сравнению с 2017 число случаев лесных пожаров увеличилось на 11%, тогда как площадь сгоревших насаждений — на 55%.

По данным Европейской информационной системы лесных пожаров, похожая ситуация наблюдается в ряде других стран. В частности, в Швеции количество пожаров возросло на 55% в 2018 году по сравнению с 2017 годом (8181 — в 2018 и 5276 — в 2017) в то время, как площадь сго-

ревшей территории оказалась выше почти в 16 раз (1433 га — в 2017 г. и 24310 га — в 2018 г.) [2].

Проанализировав эти данные, можно сделать вывод, что задачи прогнозирования возникновения и распространения пожаров, противодействие их развитию, а также выявление факторов риска и причин их возникновения для недопущения самих возгораний, являются важными не только для Российской Федерации, но и для других стран.

В настоящее время обнаружение пожаров не является сложной задачей. Даже в самых отдаленных уголках земного шара их тепловые сигнатуры могут быть обнаружены датчиками на борту спутников наблюдения Земли. Возможность быстро предоставлять информацию о приблизительном местоположении и перемещении отдельных лесных пожаров с помощью приборов на борту спутников наблюдения Земли — это глобальная история успеха, которая помогает спасти жизни и имущество.

Большинство существующих в настоящее время моделей [3, 4, 5] распространения лесных пожаров подразделяется на:

- физические и квазифизические (используют методы фундаментальной физики);
- эмпирические и квазиэмпирические (статистический анализ данных, полученных в ходе экспериментов);
- имитационные и математические модели.

Однако все эти методы анализа процессов связанных с пожарами осложняются в следствие влияния множества факторов окружающей среды, неочевидных зависимостей между ними, а также трудностей проведения практических экспериментов для оценки качества и достоверности модели. Зачастую эти проблемы помогает решить построение моделей с широким использованием методов машинного обучения, в частности применение нейронных сетей. С помощью них можно решать задачи даже с противоречивыми и зашумленными или даже недостоверными данными.

## Постановка задачи

Целями настоящей работы является проведение, методами машинного обучения, моделирования развития пожароопасной обстановки. На основе работы разработанной нейронной сети производится оценка развития картины пожарных очагов, выявление критических направлений движения огня и оценка оптимального противодействия пожарам.

Для реализации заявленных целей требуется осуществить следующие задачи:

1. Рассмотреть различные типы данных о пожароопасной обстановке и произвести их предварительную обработку и систематизацию;
2. Найти и обосновать определяющие факторы динамики пожарной обстановки в рассмотренных данных;
3. Разработать многослойную нейронную сеть с прямыми связями, предсказывающую развитие пожароопасной обстановки;
4. Продемонстрировать полученные результаты разработанной нейронной сети.

## Обзор литературы

При написании данной работы были использованы научная и учебно-методическая литература, статьи периодических изданий и интернет-ресурсы.

- В первую очередь были изучены уже имеющиеся методы моделирования и прогнозирования распространения пожаров. Гришин А. М. в статье “О математическом моделировании природных пожаров и катастроф” [8] кратко описывает уже полученные результаты в данной сфере.
- В монографии Кузнецова Г. В. и Барановского Н. В. [9] излагаются основные результаты, полученные отечественными и зарубежными исследователями в области прогноза возникновения лесных пожаров и их экологических последствий. Приводятся данные об основных процессах и факторах, влияющих на возникновение лесных пожаров. Уделяется внимание таким важным факторам лесной пожарной опасности, как метеоусловия, антропогенная нагрузка и грозовая активность. Описываются зарубежные и отечественные методики прогноза лесной пожарной опасности. Анализируется тематика численного прогноза погоды, высокопроизводительных вычислений на суперкомпьютерах, разработки геоинформационных систем. Обсуждаются перспективы развития подходов и технологий прогнозирования лесной пожарной опасности.
- Важным источником в данной области также являются книги Салливана А. Л. “Моделирование распространения огня на поверхности диких земель” [3, 4, 5]. В этой серии обзорных статей предпринимается попытка критически и всесторонне рассмотреть все типы моделей распространения поверхностного огня, разработанных с 1990 года. В первом томе рассматриваются физические и квазифизические методы. Они основаны на фундаментальной химии и физике горения и распространения огня. Во втором томе рассматриваются модели эмпирического или квазиэмпирического характера. Эти модели основаны исключительно на статистическом анализе экспериментально полученных данных с использованием или без использования некоторой

физической основы для построения соотношений. Основные соотношения эмпирических моделей — это соотношение скорости ветра и влажности топлива со скоростью прямого распространения. Сравниваются различные функциональные зависимости, выбранные различными авторами для этих переменных. И, наконец, в третьем томе рассматриваются модели имитационного или математического аналогового характера. Большинство имитационных моделей являются реализациями существующих эмпирических или квазиэмпирических моделей, и их основная функция заключается в преобразовании этих, как правило, одномерных моделей в двухмерные, а затем распространении периметра пожара по моделируемому ландшафту.

В этом исследовании наиболее важными являлись источники, предлагающие методы, основанные на информационных технологиях. Приведём также и обзор некоторых из них.

- Так, в книге “Лесные пожары: методы и исследования” [10] даётся общий обзор существующих методов моделирования поведения лесных пожаров, а также оценки лесопожарной опасности. Некоторые из самых известных пожаров подробно описаны такими системами оценки рисков, как американская и канадская. Авторами предложена имитационная модель на основе о законах горения химического огня и формально-логическом аппарате. Особое внимание уделяется использованию современных информационных технологий в борьбе с лесными пожарами. Например, рассмотрен подход к построению модели лесного пожара с помощью искусственных нейронных сетей, обрабатывающих большие наборы характеристик различной природы.
- Ясинским Ф. Н. и др. [6] был предложен новый подход к оценке качества прогнозирования возникновения лесных пожаров с помощью искусственных нейронных сетей. Была показана необходимость применения многопроцессорной техники для реализации предложенного алгоритма.
- В работе Станкевича Т. С. [11] описывается задача улучшения оперативного прогнозирования динамики развития лесных пожаров. В

качестве решения данной проблемы разрабатывается метод оперативного прогнозирования динамики развития лесного пожара в случае нестационарности и неопределенности на основе быстро развивающихся алгоритмов ИИ и глубокого обучения (сверточной нейронной сети). Для данного метода прогнозирования построена общая логическая схема, основная особенность которой - создание дерева сверточных нейронных сетей. В целях улучшения качества обучения нейронной сети, отвечающей за функцию предсказания распространения лесного пожара, предлагается создание базы данных о распространении лесных пожаров.

Для построения нейронной сети были изучены источники о Deep Learning на языке Python, в частности с применением модуля PyTorch.

- Книга [12], написанная Сэмюэлем Бернсом, представляет собой введение в методы глубокого обучения. Автор не уделяет слишком много внимания математике, так как это руководство предназначено для разработчиков, которые являются новичками в области глубокого обучения. Книга была сгруппирована по главам, в каждой из которых рассматриваются различные функции библиотек глубокого обучения (Tensorflow, Keras и PyTorch), которые могут быть использованы в языке программирования Python. Каждая глава содержит уникальную архитектуру нейронных сетей, включая свёрточные нейронные сети.
- В книге [13] объясняются причины выбора модуля PyTorch, даются фундаментальные сведения по линейной алгебре, теории вероятностей и методам оптимизации и уже на основе этого рассматриваются ключевые компоненты и функциональность Pytorch. В книге представлены все ключевые архитектуры глубокого обучения, включая прямые сети, сверточные нейронные сети, рекуррентные нейронные сети, сети долговременной кратковременной памяти, автокодеры и генеративные состязательные сети.

# Глава 1. Предварительная обработка данных о пожарах в РФ

## §1. Источник данных о пожарах

Предоставление информации о горячих точках, обнаруженных приборами на борту спутников наблюдения Земли в режиме, близком к реальному времени, является возможностью, которая была доступна в течение более чем десятилетия через систему управления ресурсами пожарной информации NASA (Fire Information for Resource Management System, FIRMS).

FIRMS была разработана Университетом Мэриленда в 2007 году на средства программы прикладных наук NASA и Продовольственной и сельскохозяйственной Организации Объединенных Наций (UNFAO). NASA начало предлагать FIRMS данные, близкие к реальному времени, в 2007 году, а UNFAO — в 2010 году через свою глобальную систему управления информацией о пожарах (GFIMS).

По данным лесной службы США, сухое древесное топливо воспламеняется при достижении температуры около 540° F (при условии наличия достаточного количества кислорода), а установленный лесной пожар горит при температурах, которые могут превышать 1500° F. Это тепло является маяком для орбитальных датчиков, таких как MODIS и VIIRS, которые обнаруживают излучаемую энергию.

MODIS вращается вокруг Земли на борту спутников Terra и Aqua, а VIIRS — на борту совместного спутника NASA / NOAA Suomi National Polar-orbiting Partnership (Suomi-NPP).

Продукт MODIS Fire and Thermal Anomalies является основой продуктов FIRMS. Когда MODIS обнаруживает тепловую аномалию, которая может указывать на лесной пожар или любой значительный источник тепла, компьютерный алгоритм определяет центр области площадью 1 км<sup>2</sup>, в которой обнаружена аномалия. Это местоположение строится и доступно для пожарных команд и менеджеров диких земель (и всех, кто имеет доступ к карте пожаров фирм) в течение трех часов наблюдения и обеспечивает приблизительное местоположение потенциального лесного пожара

или точки доступа.

Важно отметить, что геолокация обнаруженной точки доступа основана на прогнозной, а не окончательной информации о спутниковой орбите. Это означает, что могут быть небольшие различия — обычно менее 100 метров — между указанной точкой доступа и ее фактическим местоположением на земле.

VIIRS I-band (375 m) был добавлен в коллекцию FIRMS в 2016 году и был разработан с использованием продукта MODIS Fire and Thermal Anomalies. Это означает, что продукты VIIRS и MODIS fire дополняют друг друга, и как продукт VIIRS 375 m, так и продукт MODIS fire показывают хорошую согласованность в обнаружении горячих точек. VIIRS, однако, обеспечивает лучшую реакцию для более малых пожаров и обеспечивает улучшенное отображение больших периметров пожара. В добавление, продукт VIIRS 375 m показывает лучшую реакцию в ночных наблюдениях. В то время как продукт VIIRS 750 m active fire также доступен, фирмы решили распространять продукт 375 m из-за его повышенного пространственного разрешения и увеличения количества обнаруженных пожаров.

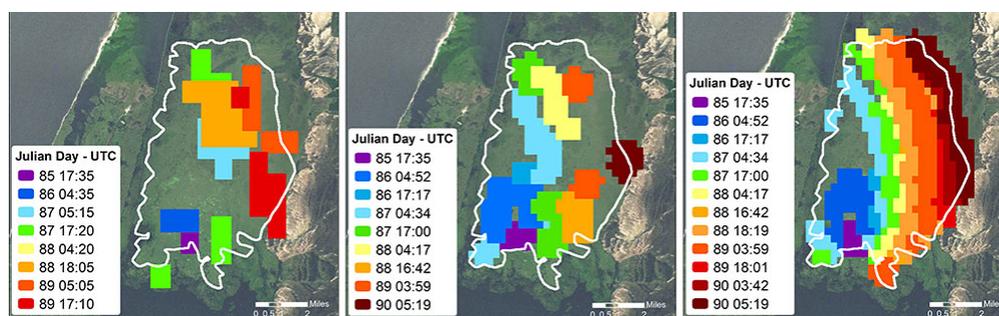


Рис. 2: Сравнение суточного распространения огня, нанесенного на карту 1 км Aqua / MODIS (слева), 750 м VIIRS (в центре) и 375 м VIIRS (справа) в Таимском экологическом заповеднике на юге Бразилии в период с 26 по 31 марта 2013 года.

На рис. 2 белый контур представляет собой выжженную площадь, нанесенную на карту с использованием 30-метровых данных Landsat-7 31 марта. Стоит обратить внимание на более когерентное распространение огня и отличное пространственное согласование данных VIIRS 375 m (правое изображение).

## VIIRS 375 m

VIIRS 375 m описан в работе Schroeder et al [14]. Продукт основан на наследии пожарной продукции MODIS [15, 16]. Алгоритм использует все пять 375-метровых каналов VIIRS для обнаружения пожаров и разделения пикселей земли, воды и облаков на изображении.

Некоторые ключевые сведения о VIIRS.

- Датчик VIIRS на борту спутника АЭС Суоми пересекает экватор приблизительно в 13:30 (восходящий узел) и 1:30 (нисходящий узел).
- VIIRS имеет 5 каналов изображений с высоким разрешением (I), 16 каналов с умеренным разрешением (M) и дневной/ночной диапазон (DNB).
- Детекторы VIIRS имеют постоянное угловое разрешение, что приводит к увеличению размера пиксельного следа по мере удаления сканирования от надира (см. Рис. 3). Это означает, что фактическая область каждого сканирования имеет форму галстука-бабочки, поскольку последовательные сканирования перекрываются от надира. Эффект галстука-бабочки уменьшается во время обработки за счет комбинации агрегации и удаления перекрывающихся пикселей.

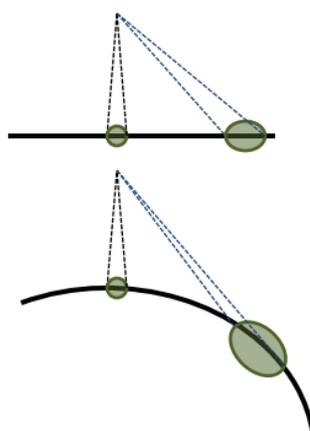


Рис. 3: Размер пиксельного следа VIIRS, проецируемого на Землю, увеличивается вдали от надира.

Обнаружение пожара осуществляется с помощью контекстного алгоритма, который использует сильное излучение среднего инфракрасного излучения от пожаров. Алгоритм MODIS исследует каждый пиксель полосы

MODIS и в конечном счете присваивает каждому из следующих классов: отсутствующие данные, облако, вода, не-огонь, огонь или неизвестный.

В соответствии с MODIS, алгоритм VIIRS представляет собой гибридный пороговый и контекстный алгоритм, использующий радиометрические сигналы из полос 4 мкм и 11 мкм (M13 и M15 соответственно) и дополнительных полос и набор тестов для внутренней облачной маски и отбраковки ложных тревог. Продукт состоит из простого файла, содержащего в основном данные широты и долготы для тех пикселей, которые классифицируются как тепловые аномалии.

## §2. Описание полученных данных

Данные содержат информацию о 9239612 возгораниях на территории Российской Федерации в период с 20 января 2012 г. по 30 июня 2019 г. Они были получены по запросу на официальном сайте NASA FIRMS[20] и представляют собой таблицу  $9239612 \times 14$  (см. Рис. 2).

:

	latitude	longitude	bright_ti4	scan	track	acq_date	acq_time	satellite	instrument	confidence	version	bright_ti5	frp	type
0	66.95284	56.64498	349.0	0.40	0.60	2012-01-20	104	N	VIIRS	n	1	259.8	4.4	2
1	66.95133	56.64341	337.6	0.40	0.60	2012-01-20	104	N	VIIRS	n	1	257.7	3.6	2
2	67.65945	60.50869	322.9	0.42	0.61	2012-01-20	104	N	VIIRS	n	1	252.2	2.0	2
3	67.17742	56.62468	321.1	0.39	0.59	2012-01-20	104	N	VIIRS	n	1	255.4	1.6	2
4	67.18079	56.63607	320.4	0.39	0.59	2012-01-20	104	N	VIIRS	n	1	257.7	1.3	2

Рис. 4: Первые пять строк данных о возгораниях на территории РФ

Рассмотрим подробнее, что из себя представляют колонки данной таблицы.

- Поля "latitude" и "longitude" — широта и долгота соответственно центра огневого пикселя. Стоит отметить, что центр данного пикселя может не совпадать в точности с местом возгорания.
- "Bright\_ti4" и "Bright\_ti5" — VIIRS I-4 и I-5 каналные яркостные температуры огненного пикселя соответственно. Измеряются в Кельвинах.
- Поля "scan" и "track" отражают фактический размер пикселя.
- "Acq\_Date" и "Acq\_Time" — дата и время получения данных от VIIRS.
- "satellite" — спутник, с которого получены данные, N = Suomi National Polar-orbiting Partnership (Suomi NPP).
- Значение "instrument" в данной таблице для всех обнаруженных возгораний одинаково (VIIRS).
- Значение "confidence" основано на наборе промежуточных величин алгоритма, используемых в процессе обнаружения. Этот параметр

предназначен для того, чтобы помочь пользователям данных оценить качество отдельных огневых пикселей. Доверительные значения устанавливаются на низкий ("l"), номинальный ("n") и высокий ("h") уровень. Пиксели дневного огня с низкой достоверностью обычно ассоциируются с областями солнечного блеска и более низкой относительной температурной аномалией ( $<15\text{K}$ ) в среднем инфракрасном канале I-4. Номинальные доверительные пиксели — это пиксели, свободные от потенциального загрязнения солнечным блеском в течение дня и отмеченные сильной ( $>15\text{K}$ ) температурной аномалией в дневных или ночных данных. Пиксели огня высокой достоверности связаны с пикселями дня или ночи насыщенными. Также стоит отметить, что ночные пиксели с низкой достоверностью наблюдаются только в географической области, простирающейся от  $11^\circ\text{E}$  до  $110^\circ\text{W}$  и от  $7^\circ\text{N}$  до  $55^\circ\text{S}$ . Эта область описывает область влияния Южно-Атлантической магнитной аномалии, которая может вызывать ложные яркостные температуры в среднем инфракрасном канале I-4, приводящие к потенциальным ложноположительным сигналам тревоги. Они были удалены из данных NRT, распространяемых FIRMS.

- "version" — идентифицирует коллекцию (например, VIIRS Collection 1) и источник обработки данных: почти в реальном времени (суффикс NRT) или стандартная обработка (только коллекция).
- "frp" отображает пожарную излучательную мощность в МВт. Учитывая уникальное пространственное и спектральное разрешение данных, алгоритм обнаружения пожара VIIRS 375 m был настроен таким образом, чтобы оптимизировать его реакцию на небольшие пожары, балансируя возникновение ложных тревог. Частое насыщение среднего инфракрасного канала I4 ( $3,55\text{-}3,93\text{ мкм}$ ), приводящее к обнаружению активных пожаров, требует дополнительных испытаний и процедур, позволяющих избежать ошибок классификации пикселей. В результате субпиксельная характеристика огня (например, поиск мощности излучения огня FRP) является жизнеспособной только при небольших и/или низкоинтенсивных пожарах.
- "type" - предполагаемый источник возгорания. Тип 0 — предполагае-

мый пожар растительности, 1 — активный вулкан, 2 — другой статический источник земли, 3 — обнаружение на шельфе (включает все обнаружения над водой).

Сделаем предварительную обработку данных. В последующих расчетах нам не понадобятся значения полей спутника, версии и инструмента. Поэтому удалим колонки, содержащие эти значения. Также нас интересуют только пожары растительности, следовательно, сделаем выборку строк со значением `type = 0`. И, наконец, удалим пожары с низким уровнем достоверности. Все преобразования проводились с использованием модуля `Pandas` на языке программирования `Python`. Первые строки полученной таблицы представлены на Рис. 16.

	<code>latitude</code>	<code>longitude</code>	<code>bright_ti4</code>	<code>scan</code>	<code>track</code>	<code>acq_date</code>	<code>acq_time</code>	<code>confidence</code>	<code>bright_ti5</code>	<code>frp</code>	<code>type</code>
<b>6</b>	70.38039	68.23817	335.6	0.38	0.43	2012-01-20	600	n	256.6	5.8	0
<b>11</b>	70.37950	68.23682	340.1	0.49	0.65	2012-01-20	922	n	256.9	3.9	0
<b>27</b>	61.34594	112.85989	335.8	0.40	0.44	2012-01-20	1944	n	238.3	2.8	0
<b>35</b>	51.73400	111.95381	322.3	0.40	0.60	2012-01-20	1946	n	240.3	2.4	0
<b>57</b>	64.29448	76.17104	326.7	0.39	0.36	2012-01-20	2125	n	244.9	7.0	0

Рис. 5: Первые пять строк обработанной таблицы

Таким образом, в нашей таблице осталось 7645504 строк.

Ниже представлено распределение возгораний 1 и 2 июля 2012 года.

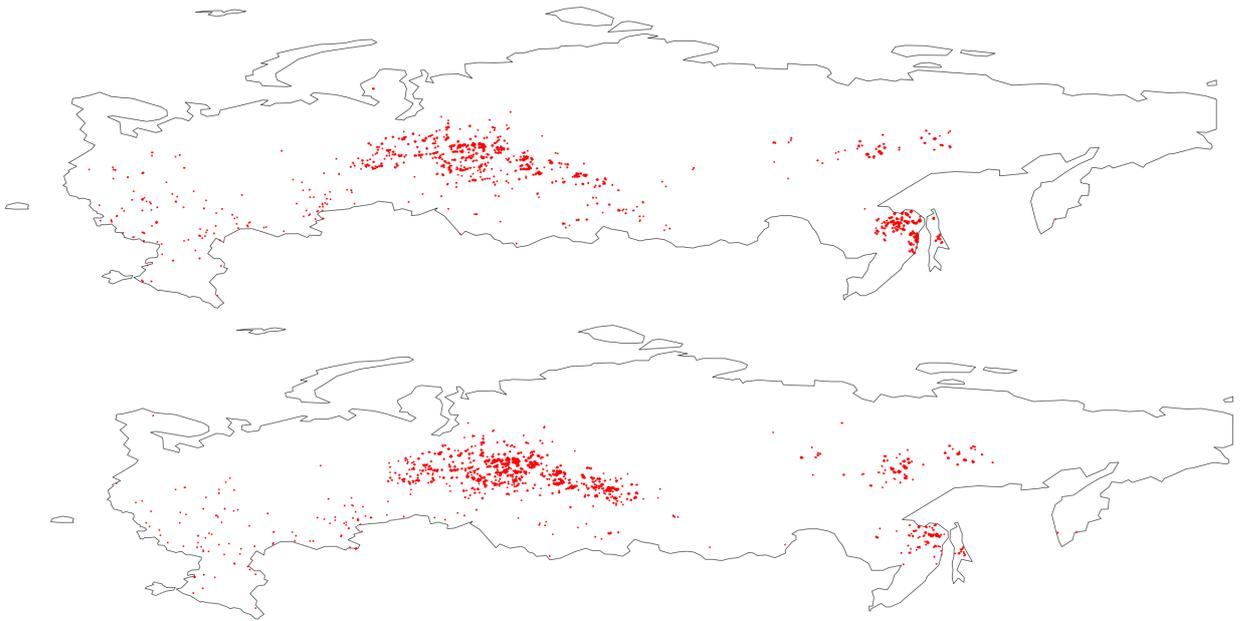


Рис. 6: Визуальное представление данных за 1 и 2 июля 2012 года.

Сгруппируем данные по годам и месяцам.

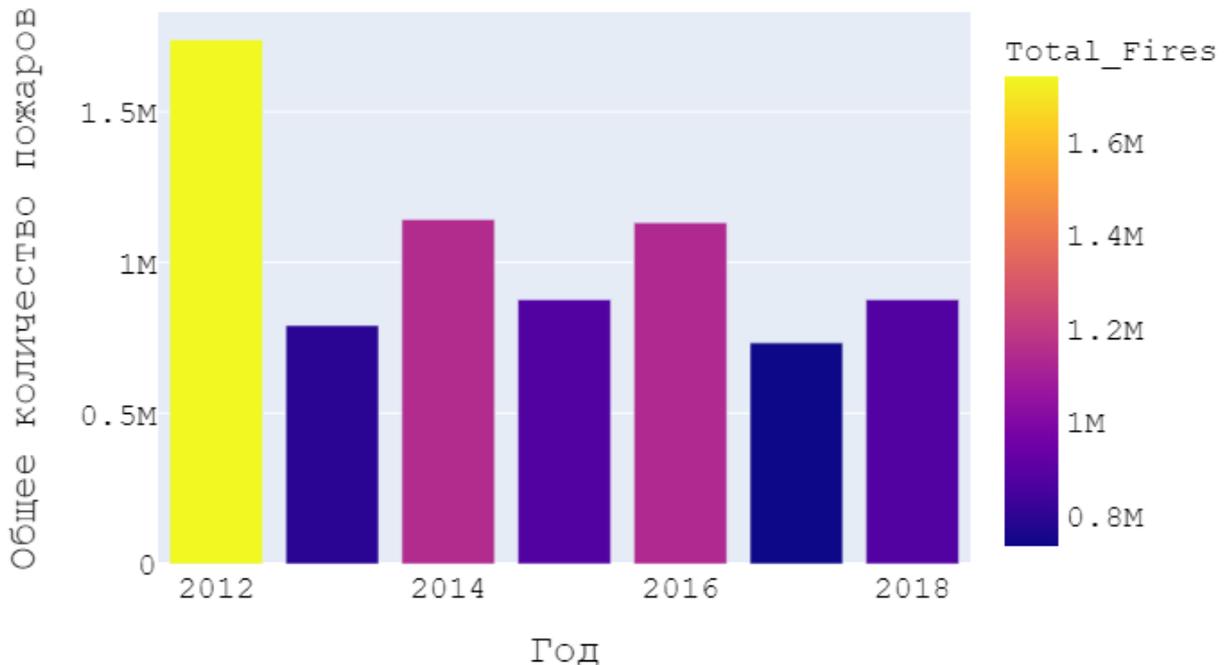


Рис. 7: Количество пожаров за год с 2012 по 2018

Эти данные отражают ту динамику, о которой уже говорилось во введении. Не наблюдается сильного роста общего числа пожаров, однако, как уже говорилось, количество сгоревших насаждений увеличивается, в

данных с NASA это не отражено.

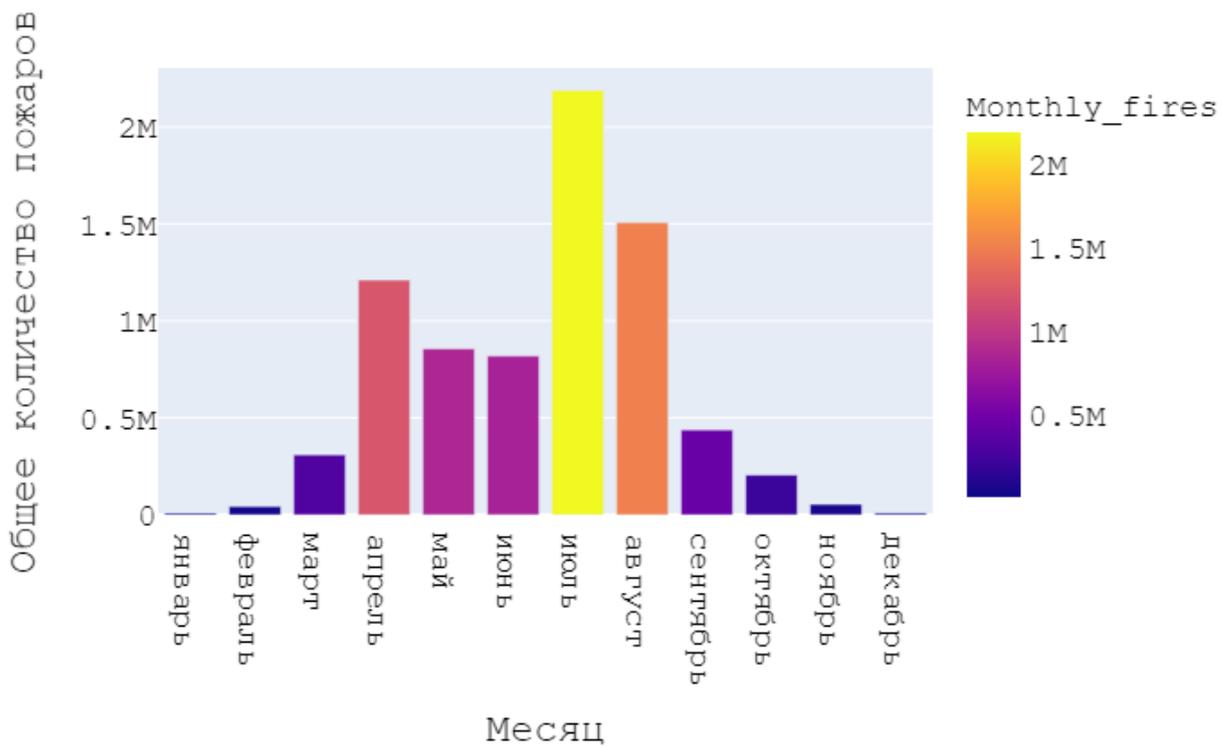


Рис. 8: Количество пожаров по месяцам

Вполне ожидаемо, в связи с климатом РФ большая часть пожаров приходится на период с апреля по август.

### §3. Обработка данных по сетке

Для решения задачи прогнозирования возгораний на определенной территории необходимо разделить данную территорию на некоторое количество областей с шагом 0,1 по широте и 0,2 по долготе. Затем для каждого дня и каждой области необходимо вычислить количество произошедших там пожаров, среднюю и максимальную яркости пожаров, а так же определить переменную, отвечающую за то, произошел ли там хоть один пожар, ее мы и будем предсказывать. Пример строк полученной таблицы представлен на рис. 9.

	day	lat	lon	mo	station	year	is_fire	num_fires	max_ti4	mean_ti4
<b>225036</b>	1	58.6	80.0	7	291220	2012	1	1	309.4	309.40
<b>225042</b>	1	59.2	80.0	7	291220	2012	1	5	336.8	310.18
<b>225044</b>	1	59.4	80.0	7	291220	2012	1	10	336.1	308.98
<b>225045</b>	1	59.5	80.0	7	291220	2012	1	1	332.0	332.00
<b>225060</b>	1	61.0	80.0	7	238670	2012	1	10	333.1	310.68

Рис. 9: Обработанная таблица

Затем для каждой строки определим данные о пожарах в этой и соседних областях в предыдущий день. Ниже представлен пример строк окончательной таблицы.

	day	lat	lon	mo	year	is_fire	num_fires	max_ti4	mean_ti4	num_neigh_fires_y	num_sq_neigh_fires_y	sum_mean_ti4_neigh_y
<b>7641</b>	2	59.1	80.2	6	2012	1	8	307.7	302.375000	13	1	328.369231
<b>7740</b>	2	59.0	80.4	6	2012	1	3	313.1	302.200000	13	1	328.369231
<b>8370</b>	2	62.0	81.6	6	2012	1	3	353.1	330.666667	4	1	339.375000
<b>8601</b>	2	55.1	82.2	6	2012	1	2	343.7	342.600000	21	1	324.009524
<b>8634</b>	2	58.4	82.2	6	2012	1	52	367.0	340.961538	12	1	332.283333

Рис. 10: Обработанная таблица

# Глава 2. Нейросетевая обработка данных

## §1. Типы нейронных сетей

В каждой нейронной сети обязательно есть входной слой - первый слой нейронов. Задача этого слоя не вычисления, а прием и распределение входных данных по нейронам. И этот слой единственный, являющийся общим для всех типов нейросетей, а критерием для классификации является уже дальнейшее строение[12]:

1. Однослойная нейронная сеть. В ней сигналы со входного слоя сразу движутся на выходной, который, преобразовав их, выдает ответ.
2. Многослойная нейронная сеть. В ней, кроме входного и выходного слоев, есть ещё несколько скрытых. Их количество зависит от сложности нейронной сети. Именно эта сеть больше всего похожа на биологическую нейронную сеть. Такие сети появились сравнительно недавно, до них использовались только однослойные сети.

Помимо количества слоёв, нейронные сети можно разделить по направлению распределения данных по синапсам между нейронами:

1. Нейронные сети прямого распространения (однонаправленные, с прямой связью). В них сигнал движется строго от входного слоя к выходному. Движения в обратном направлении нет и для данных сетей оно невозможно. Такие сети успешно решают задачи распознавания образов, прогнозирования и кластеризации.
2. Рекуррентные нейронные сети (с обратными связями). В них сигнал перемещается в обоих направлениях. Таким образом, результат выхода может возвращаться на вход. Выход нейрона определяется весовыми характеристиками и входными сигналами, плюс дополняется предыдущими выходами, снова вернувшимися на вход. Этим нейросетям присуща функция кратковременной памяти, на основании чего сигналы восстанавливаются и дополняются во время их обработки.
3. Радиально-базисные функции.
4. Самоорганизующиеся карты.

В данной работе представлена модель нейронной сети прямого распространения.

Нейронная сеть прямого распространения — искусственная нейронная сеть, в которой соединения между узлами не образуют цикл. Такая сеть отличается от рекуррентной нейронной сети. Нейронная сеть с прямой связью была первым и самым простым типом искусственной нейронной сети. В этой сети информация перемещается только в одном направлении вперед от входных узлов, через скрытые узлы (если они есть) и к выходным узлам. В сети нет циклов или петель обратных связей.

Самым простым типом нейронной сети с прямой связью является персептрон, нейронная сеть с прямой связью без скрытых слоев. То есть, персептрон имеет только входной слой и выходной. Выходные единицы вычисляются непосредственно из суммы произведений их весов с соответствующими входными единицами плюс некоторое смещение.

Исторически выход персептрона был двоичным, то есть он выводил значение 0 или 1. Это достигается путем передачи вышеупомянутой суммы произведений в ступенчатую функцию  $H(x)$ . Это определяется как

$$H(x) = \begin{cases} 1 & \text{если } x \geq 0 \\ 0 & \text{если } x < 0 \end{cases}$$

Для двоичного персептрона с  $n$ -мерным входом  $\vec{x}$ ,  $n$ -мерным вектором весов  $\vec{w}$  и смещением  $b$  выходной сигнал  $o$  получается из

$$o = \begin{cases} 1 & \text{если } \vec{w} \cdot \vec{x} + b \geq 0 \\ 0 & \text{если } \vec{w} \cdot \vec{x} + b < 0 \end{cases}$$

Поскольку персептрон делит входное пространство на два класса, 0 и 1, в зависимости от значений  $\vec{w}$  и  $b$ , он известен как линейный классификатор.

Более общие однослойные персептроны могут использовать функции активации, отличные от ступенчатой функции  $H(x)$ . Типичными вариантами выбора являются тождественная функция  $f(x) = x$ , сигмовидная функция  $\sigma(x) = (1 + e^{-x})^{-1}$  и гиперболический тангенс  $\tanh(x) = \frac{e^x + e^{-x}}{e^x - e^{-x}}$ . Использование любой из этих функций гарантирует, что выходные данные

являются непрерывным числом (в отличие от двоичного), и, следовательно, не каждая функция активации дает линейный классификатор.

Вообще говоря, персептрон с функцией активации  $g(x)$  имеет выход

$$o = g(\vec{w} \cdot \vec{x} + b)$$

Персептроны могут обучаться с помощью простого алгоритма обучения, который обычно называется дельта-правилом. Он вычисляет ошибки между вычисленными выходными данными и образцами выходных данных и использует их для создания корректировки весов, реализуя таким образом форму градиентного спуска.

Более сложным видом сетей является многослойный персептрон (MLP). Он представляет собой искусственную нейронную сеть, состоящую из множества персептронов. В отличие от однослойных персептронов, MLP способны учиться вычислять нелинейно разделимые функции. Поскольку они могут изучать нелинейные функции, они являются одним из основных методов машинного обучения для регрессии и классификации в контролируемом обучении. MLP обычно организованы в нечто, называемое слоями. Обобщенная искусственная нейронная сеть состоит из входного слоя, некоторого числа (возможно, нуля) скрытых слоев и выходного слоя. В случае однослойного персептрона скрытых слоев не существует, поэтому общее количество слоев равно двум. С другой стороны, MLP имеют по меньшей мере один скрытый слой, каждый из которых состоит из множества персептронов. Ниже приведен пример нейронной сети с двумя скрытыми слоями.

В многослойных сетях используется множество методов обучения, наиболее популярным из которых является обратное распространение. Здесь выходные значения сравниваются с правильным ответом для вычисления значения некоторой предопределенной функции ошибки. Затем различными способами ошибка возвращается по сети. Используя эту информацию, алгоритм корректирует вес каждого соединения, чтобы уменьшить значение функции ошибки на некоторое небольшое значение. После повторения этого процесса для достаточно большого количества тренировочных циклов сеть обычно будет сходиться к некоторому состоянию, в котором погрешность вычислений мала. В этом случае можно сказать, что сеть изучила определенную целевую функцию. Для правильной настройки весов

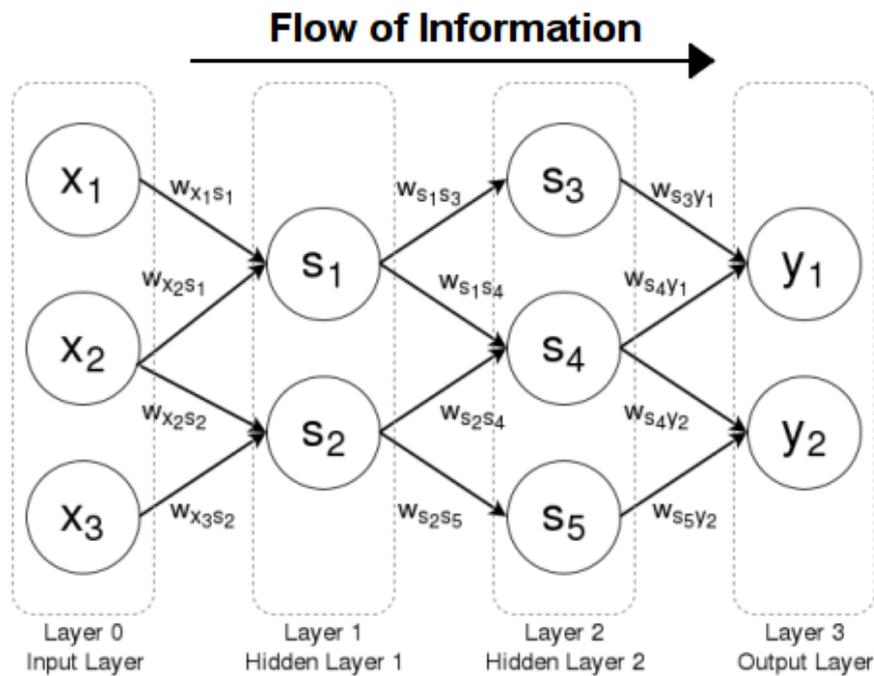


Рис. 11: График функции потерь в процессе обучения

применяется общий метод нелинейной оптимизации, который называется градиентным спуском. Для этого сеть вычисляет производную функции ошибки по весам сети и изменяет веса, чтобы ошибка уменьшалась (таким образом, спускаясь вниз по поверхности функции ошибки). По этой причине обратное распространение может применяться только в сетях с дифференцируемыми функциями активации.

## §2. Функция потерь

В задачах бинарной классификации удобно использовать в качестве функции потерь бинарную кросс-энтропию.

Выход модели можно интерпретировать как вероятности, которые служат основанием для классификации наблюдения. Вероятность моделируется с использованием логистической функции  $g(z) = 1/(1+e^{-z})$ , где  $z$  — некоторая функция входного вектора  $x$ , обычно просто линейная функция. Вероятность выхода  $y = 1$  определяется из

$$q_{y=1} = \hat{y} \equiv g(\mathbf{w} \cdot \mathbf{x}) = 1/(1 + e^{-\mathbf{w} \cdot \mathbf{x}}),$$

где вектор весов  $\mathbf{w}$  оптимизируется с помощью некоторого соответствующего алгоритма. Аналогично, дополнительная вероятность нахождения выхода  $y = 0$  определяется из

$$q_{y=0} = 1 - \hat{y}$$

Используя обозначения,  $p \in \{y, 1 - y\}$ , а также  $q \in \{\hat{y}, 1 - \hat{y}\}$ , мы можем использовать кросс-энтропию, чтобы получить меру различия между  $p$  и  $q$ :

$$H(p, q) = - \sum_i p_i \log q_i = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

Логистическая регрессия обычно оптимизирует потери логарифма для всех наблюдений, на которых она обучается, что аналогично оптимизации средней кросс-энтропии в выборке. Например, предположим, у нас есть  $N$  образцы с каждым образцом, проиндексированным  $n = 1, \dots, N$ . Среднее значение функции потерь определяется следующим образом:

$$J(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N H(p_n, q_n) = -\frac{1}{N} \sum_{n=1}^N \left[ y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n) \right],$$

где  $\hat{y}_n \equiv g(\mathbf{w} \cdot \mathbf{x}_n) = 1/(1 + e^{-\mathbf{w} \cdot \mathbf{x}_n})$ .

В модуле PyTorch определен метод BCEWithLogitsLoss для бинарной кросс-энтропии.

### §3. Выбор параметров модели

Самым важным фактором, влияющим на результаты работы нейронной сети, является несомненно выбор параметров. Для задач машинного обучения, использующего нейронные сети, нет универсального алгоритма этого выбора, чаще всего это делается методом проб и ошибок. Требуется опробовать несколько комбинаций различных значений параметров, чтобы выбрать оптимальные. Для этого был использован метод Grid Search (поиск по сетке). Была сгенерирована сетка значений параметров и при помощи кросс-валидации из всех комбинаций данных значений была выбрана наилучшая модель.

Всего были проверены 27 моделей с разбиением обучающей выборки на 3 части для кросс-валидации. Для всех моделей было получено среднее значение accuracy. Лучшие модели представлены ниже.

№	Количество скрытых слоев	Количество узлов	learning rate	accuracy
1	3	10	0.001	0.9737
2	10	3	0.001	0.9313
3	3	15	0.001	0.9248

Таблица 1

Итак, следует построить сеть с тремя скрытыми слоями с 10 узлами на них и значением  $learning\_rate = 0.001$ . Для оптимизации выбран алгоритм Адама.

## §4. Обучение нейронной сети

Известно, что для задач с неравными классами обучение по метрике ассигасу может показывать плохие результаты. Обучим нейронную сеть и оценим работу модели на тестовой выборке (см. Приложение).

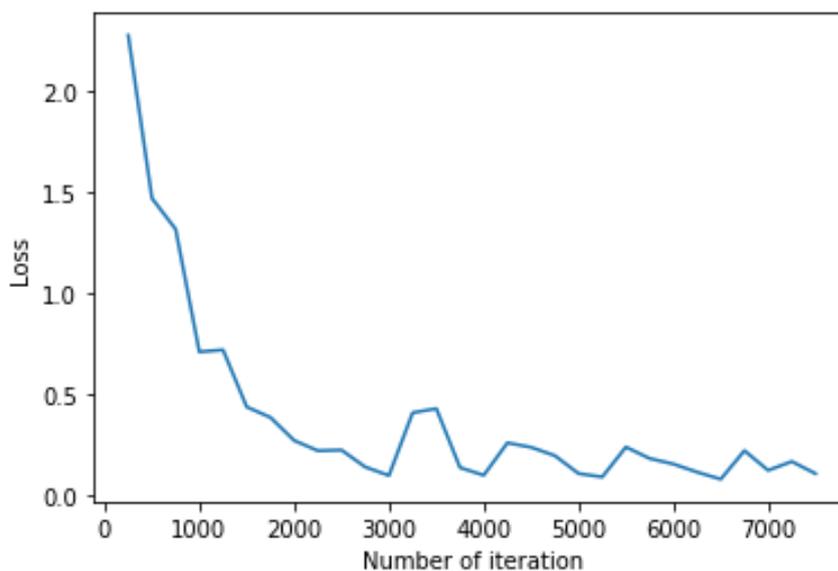


Рис. 12: График функции потерь в процессе обучения

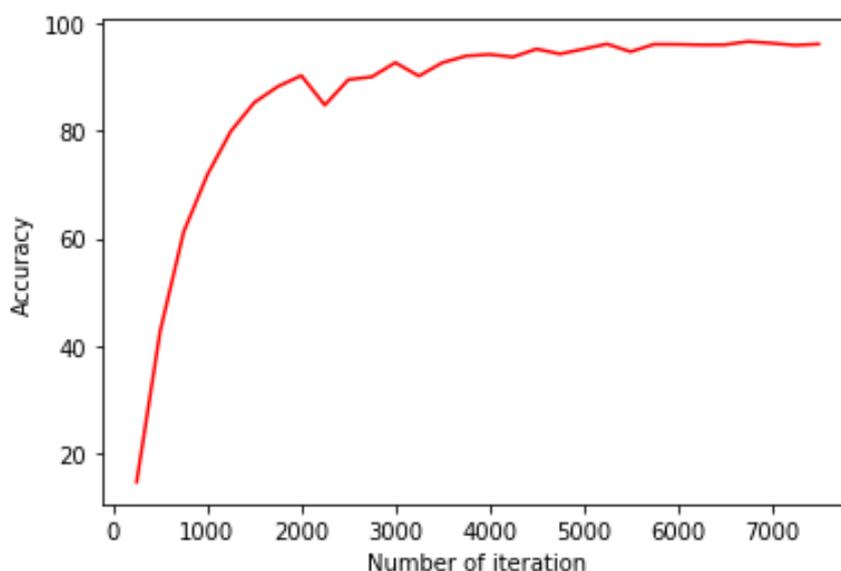


Рис. 13: График метрики ассигасу в процессе обучения

На тестовых данных значение метрики ассигасу составляет 0,98. Может показаться, что это точное предсказание, однако, так как классы в

данной задаче являются неравными (количество областей без пожаров значительно больше), делать вывод, опираясь только на эту метрику, нельзя. Чтобы оценить точность предсказанных значений следует использовать метрику F-мера.

Для начала введем понятия True Positive (TP), True Negative (TN), False Positive (FP) и False Negative (FN).

Пусть  $y$  — реальное значение объекта, а  $\hat{y}$  — предсказанное.

	$y = 1$	$y = 0$
$\hat{y} = 1$	True Positive (TP)	False Positive (FP)
$\hat{y} = 0$	False Negative (FN)	True Negative (TN)

Таблица 2

Введем понятия precision и recall.

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

Таким образом, precision показывает, какая часть от объектов, объявленных классификатором положительными, на самом деле является положительными, а recall — какая часть от объектов, реально положительных, классификатором объявлена положительными.

Данные метрики не дают классификатору объявить для всех объектов один и тот же класс, т.е. в нашем случае не позволят записать все области в области без пожаров.

Как правило для оптимизации параметров модели (например, в случае перебора по сетке, как в данной работе) используется только одна метрика, улучшение которой ожидается увидеть на тестовой выборке. Существует несколько способов объединить precision и recall в единый критерий качества. F-мера (в общем случае) — среднее гармоническое precision и recall :

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

F-мера достигает максимума, если precision и recall равны единице, и близка к нулю, если один из аргументов близок к нулю.

Вычислим значение F-меры для нашей оптимальной по ассигасу модели. Оно составляет 0,6602, что является неплохим результатом, если учесть, что при обучении использовались только данные о количестве и мощности пожаров. Встает вопрос, можно ли улучшить данную модель, если делать перебор по сетке, вычисляя F-меру, а не ассигасу, а также проводить обучение по метрике precision.

Чтобы проверить это, необходимо будет заново запустить перебор по сетке, однако сначала добавим к нашим данным информацию о погоде.

# Глава 3. Добавление данных о погоде и улучшение предсказаний нейронной сети

## §1. Загрузка и обработка данных

В данной работе использовались данные о погоде от Национальной метеорологической службы Национального управления океанических и атмосферных исследований.

Эти данные хранятся в сервисе BigQuery.

Данные хранятся в базе, состоящей из таблиц для каждого года с 1929 по 2020, а также таблицы, содержащей информацию о станциях.

Как правило, доступны данные более 9000 станций.

Ежедневными элементами, включенными в набор являются:

- средняя температура
- средняя скорость ветра
- среднее давление на уровне моря
- средняя видимость
- максимальная устойчивая скорость ветра
- максимальный порыв ветра
- количество осадков

Однако, для каждой станции этот набор может различаться. Для российских станций в основном доступны только данные о средней температуре и количестве осадков.

Чтобы загрузить данные с сервиса, необходимо подключить модуль `bq_helper` и составить запросы в формате SQL.

Для начала загрузим из таблицы со станциями все станции, находящиеся на территории РФ (в данных NOAA страна имеет обозначение RS).

```
Query = '''  
SELECT * from 'bigquery-public-data.noaa_gsod.stations'
```

```
WHERE country="RS"
```

```
'''
```

Начало полученной таблицы представлено ниже.

	usaf	wban	name	country	state	call	lat	lon	elev	begin	end
0	200340	99999	NAGURSKOYE	RS	None	None	80.817	47.417	+0018.0	19550305	20030314
1	200490	99999	TIKHAYA BAY	RS	None	None	80.367	52.917	+0046.0	19320202	20040930
2	200660	99999	USHAKOVA ISLAND	RS	None	None	80.833	79.700	+0047.0	19550305	20040118
3	200850	99999	OSTROV PIONEER	RS	None	None	79.883	91.233	+0030.0	19591101	19591225
4	202740	99999	OSTROV UEDINENIJA	RS	None	None	77.500	82.200	+0023.0	19350202	19961121
5	202910	99999	SOLNECHNAYA BAY	RS	None	None	78.233	103.117	+0030.0	19550305	20050901
6	206470	99999	OSTROV GOLETS	RS	None	None	73.050	53.100	+0029.0	19550305	19621210
7	206490	99999	MAMOCKIN SAR	RS	None	None	73.267	56.400	+0043.0	19320102	19580916
8	206790	99999	TAREYA (STREAM)	RS	None	None	73.250	90.917	+0021.0	19550304	20020807
9	206960	99999	CAPE KASISTY	RS	None	None	73.667	109.750	+0020.0	19550305	19910418

Рис. 14: Первые строки таблицы с российскими станциями

Даже по первым строкам видно, что не все станции работали в период с 2012 по 2019 гг. Необходимо убрать из таблицы те, которые прекратили свою работу до 2012 года.

	usaf	wban	name	country	state	call	lat	lon	elev	begin	end
15	209820	99999	VOLOCHANKA	RS	None	None	70.967	94.500	+0040.0	19370101	20180105
39	221010	99999	YANISKOSKI	RS	None	None	68.967	28.783	+0101.0	19590101	20160614
40	221050	99999	NIVANKYUL	RS	None	None	68.350	30.733	+0090.0	19590101	20160614
43	221330	99999	OSTROV HATLOV	RS	None	None	68.817	37.333	+0064.0	19590101	20141112
56	223050	99999	ZASHEEK	RS	None	None	66.250	31.117	+0119.0	19630613	20160614
60	223610	99999	MORZHOVEC	RS	None	None	66.717	42.483	+0016.2	19590102	20180228
69	226380	99999	KALGACHIHA	RS	None	None	63.333	36.717	+0182.9	19590101	20141231
88	228290	99999	VOSNESENE	RS	None	None	61.017	35.483	+0041.0	19590101	20160614
93	228910	99999	SOSNOVO	RS	None	None	60.550	30.317	+0070.0	19590101	20160614
104	230740	99999	DUDINKA	RS	None	None	69.400	86.167	+0019.0	19480101	20141221

Рис. 15: Первые строки таблицы с российскими станциями

Далее нужно, объединяя таблицу со станциями с сервиса и таблицы с погодой 2012-2019 гг, создать новую таблицу, содержащую все данные о погоде в России за 2012-2019 гг. Составим запрос с параметром года.

```
weatherAtRS = '''
```

```
SELECT stn, lat,lon, name, year, mo, da, temp, prcp
FROM
```

```
‘bigquery-public-data.noaa_gsod.gsod{’ a
```

```

INNER JOIN `bigquery-public-data.noaa_gsod.stations` b
ON a.stn = b.usaf
WHERE
    country = 'RS'
'''

```

	stn	lat	lon	name	year	mo	da	temp	prcp
0	304050	55.383	101.033	TANGUJ	2012	02	13	-0.6	0.00
1	301420	58.883	111.250	VIZIRNYJ	2012	07	15	60.3	0.44
2	301420	58.883	111.250	VIZIRNYJ	2012	12	28	-21.5	0.00
3	236990	62.717	101.117	KERBO	2012	12	21	-26.1	0.00
4	236990	62.717	101.117	KERBO	2012	10	28	-8.5	0.00

Рис. 16: Первые строки полученной таблицы

Для каждой области сетки с данными о пожарах необходимо определить, с какой станции брать данные о погоде, а для этого нужно найти ближайшие станции. Так как Земля не является плоскостью, для расчета расстояния требуется особенная формула.

## §2. Нахождение расстояния между двумя точками

Длину ортодромии между двумя точками на сфере с учетом их долготы и широты можно вычислить при помощи формулы гаверсинусов.

Первая таблица гаверсинусов на английском языке была опубликована Джеймсом Эндрю в 1805 году, но известно более раннее использование Хосе де Мендоса и Риосом в 1801 году [17]. Термин "гаверсинус" был придуман в 1835 году Джеймсом Инманом [18].

Это название формулы следует из того факта, что она обычно записывается в терминах функции гаверсинуса, заданной  $hav(\theta) = \sin^2(\frac{\theta}{2})$ . Формулы в равной степени могут быть написаны в терминах любого кратного гаверсинуса, например более старой функции синус-верзуса (дважды гаверсинуса).

Пусть центральный угол  $\Theta$  между любыми двумя точками на сфере равен:

$$\Theta = \frac{d}{r}$$

где:

$D$  — расстояние между двумя точками вдоль ортодромии сферы,  $R$  — радиус сферы.

Формула гаверсинусов позволяет гаверсинусу из  $\Theta$  (то есть,  $hav(\Theta)$ ) вычисляться непосредственно из широты и долготы двух точек:

$$hav(\Theta) = hav(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) hav(\lambda_2 - \lambda_1)$$

где  $\phi_1, \phi_2$  — широта точки 1 и широта точки 2 (в радианах),  $\lambda_1, \lambda_2$  — долгота точки 1 и долгота точки 2 (в радианах).

И, наконец, функция гаверсинусов  $hav(\Theta)$  является

$$hav(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2}$$

Функция гаверсинусов вычисляет половину синус-верзуса угла  $\theta$ .

Чтобы рассчитать расстояние  $d$ , применим  $archaversine$  (обратный гаверсинус) в  $h = hav(\Theta)$  или используем арксинус:

$$d = r \text{ archav}(h) = 2r \arcsin\left(\sqrt{h}\right)$$

или более явно:

$$\begin{aligned}d &= 2r \arcsin \left( \sqrt{\text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1)} \right) \\ &= 2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right)\end{aligned}$$

Аналогичная формула может быть написана с использованием косинусов вместо гаверсинусов, но если две точки находятся близко друг к другу (например, в километре друг от друга), мы можем получить  $\cos(d/R) = 0.99999999$ , что приводит к неточному ответу. Поскольку формула гаверсинуса использует синус, она избегает этой проблемы.

Любая из этих формул является лишь приближением применительно к Земле, которая не является совершенной сферой: "радиус Земли"  $R$  варьируется от 6356,752 км на полюсах до 6378,137 км на экваторе.

### §3. Нейросетевая обработка дополненных данных

Подход об использовании F-меры в качестве функции ошибки при обучении нейронной сети в задачах с неравными классами был предложен в [21]. Используем этот подход при обучении и прогоне по сетке для поиска оптимальных параметров модели.

Снова обучим 27 моделей и оценим их с помощью кросс-валидации, чтобы выбрать оптимальную, но теперь уже не по ассурасу, а по F-мере. Обучающую выборку так же делим на три части. Лучшие модели представлены ниже.

№	Количество скрытых слоев	Количество узлов	learning rate	F-score
1	3	15	0.001	0.76
2	10	3	0.001	0.74
3	3	10	0.001	0.68

Таблица 3

На рис.17 представлен график функции потерь при обучении оптимальной модели.

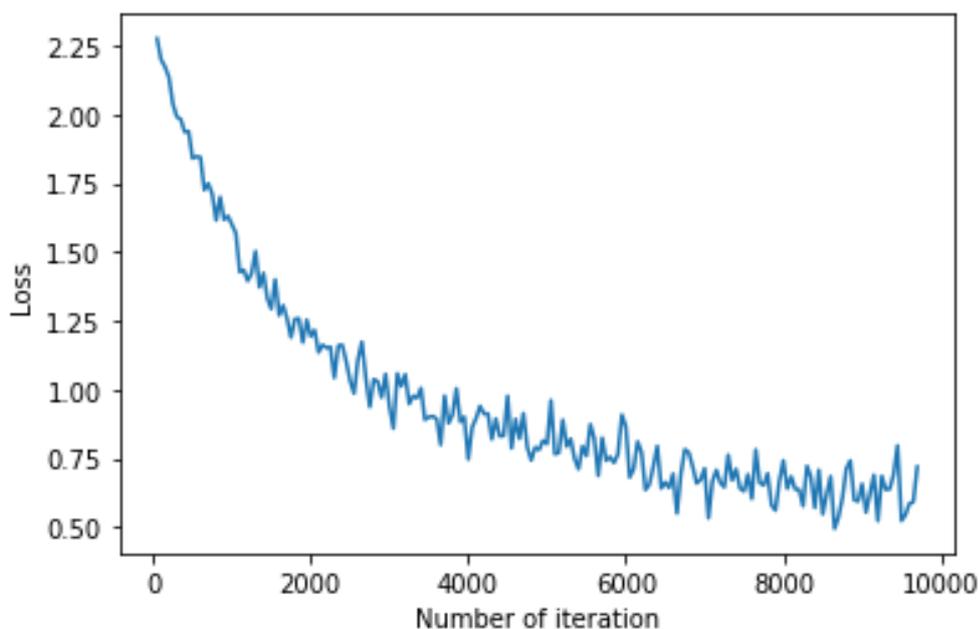


Рис. 17: График функции потерь в процессе обучения

На тестовых данных модель дает значение F-меры 0,76, что значительно лучше, чем модель, обученная без данных о погоде и на основе метрики ассигасы.

## Выводы

Предложенный подход совмещения данных о пожарах и данных о погоде позволяет добиваться лучших результатов предсказания нейронной сети, чем по отдельности. В рамках данной задачи, для неравных классов, использование метрики ассигасу признано необоснованным, так как не позволяет гарантировать нужную степень точности работы нейронной сети. Была предложена замена на F-меру, решающая задачу достижения требуемой степени точности.

## Заключение

Целью данной работы являлась разработка модели нейронной сети, предсказывающей развитие пожароопасной обстановки по выбранному региону. Разумно было предположить совмещение данных о пожарах с космических снимков с данными о погоде за соответствующий период времени.

Основные результаты представленной дипломной работы:

- Рассмотрены данные о пожарной обстановке в России от NASA FIRMS, полученные с помощью датчиков VIIRS в период с 2012 по 2019 год, а также данные о погоде от Национальной метеорологической службы Национального управления океанических и атмосферных исследований (NOAA);
- Разработана многослойная нейронная сеть с прямыми связями с выбором оптимальных параметров модели методом поиска по сетке;
- Произведено нейросетевое предсказание развития пожароопасной обстановки на следующий интервал времени по данным космических снимков от NASA FIRMS;
- На основе совмещения данных с космических снимков от NASA FIRMS с данными состояния погоды от NOAA произведена коррекция нейронной сети и получена лучшая точность предсказания пожароопасной обстановки;
- Показано, что для задач с неравными классами использование метрики ассигасу не позволяет в достаточной мере оценить точность предсказания на следующий интервал времени;
- Предложена замена метрики ассигасу на F-меру, которая решает задачу требуемой точности предсказания, с учетом совмещения данных о погоде.

Исходя из всего вышеперечисленного, можно утверждать, что поставленная задача была решена полностью.

## Список литературы

- [1] Филипчук А. Н., Моисеев Б. Н., Медведева М. А., Кинигопуло П. С., Югов А. Н. Сравнительная характеристика лесов по данным глобальной оценки лесных ресурсов 2015 // Лесохозяйственная информация, № 3, 2017, с.: 75-91
- [2] Jesús San-Miguel-Ayanz, Tracy Durrant, Roberto Boca, Giorgio Libertà, Alfredo Branco, Daniele de Rigo, Davide Ferrari, Pieralberto Maianti, Tomàs Artés Vivancos, Duarte Oom, Hans Pfeiffer, Daniel Nuijten, Thaïs Leray; Forest Fires in Europe, Middle East and North Africa 2018. EUR 29856 EN, ISBN 978-92- 76-11234-1, doi:10.2760/1128
- [3] Sullivan A.L. (2009) Wildland surface fire spread modelling, 1990–2007. 1: Physical and quasi-physical models. International Journal of Wildland Fire, no. 18, pp. 349–368.
- [4] Sullivan A.L. (2009) Wildland surface fire spread modelling, 1990–2007. 2: Empirical and quasi-empirical models. International Journal of Wildland Fire, no. 18, pp. 369–386.
- [5] Sullivan A.L. (2009) Wildland surface fire spread modelling, 1990–2007. 3: Simulation and mathematical analogue models. International Journal of Wildland Fire, no. 18, pp. 387–403.
- [6] Ясинский Федор Николаевич, Потемкина Ольга Владимировна, Сидоров Сергей Георгиевич, Евсеева Анна Владимировна Прогнозирование вероятности возникновения лесных пожаров с помощью нейросетевого алгоритма на многопроцессорной вычислительной технике // Вестник ИГЭУ. 2011. №2
- [7] Vahidnia M.H., Alesheikh A.A., Behzadi S., Salehi S. (2013) Modeling the spread of spatio-temporal phenomena through the incorporation of ANFIS

and genetically controlled cellular automata: a case study on forest fire. *International Journal of Digital Earth*, vol. 6, no. 1, pp. 51–75.

- [8] Гришин А.М. Моделирование и прогноз катастроф. - Томск: Изд-во Том. ун-та, 2002.
- [9] Кузнецов Г.В. Прогноз возникновения лесных пожаров и их экологических последствий. - Новосибирск: Изд-во СО РАН, 2009.
- [10] Ходаков В.Е., Жарикова М.В. Лесные пожары: методы и исследования // Херсон: Гринь Д.С., 2011, с.: 458
- [11] Станкевич Т.С. Применение сверточных нейронных сетей для решения задачи оперативного прогнозирования динамики распространения лесных пожаров // Бизнес-информатика. 2018. № 4 (46). С. 17–27. DOI: 10.17323/1998-0663.2018.4.17.27
- [12] Samuel Burns. Python Deep learning // Kindle Edition, 2019, p. 178
- [13] Nihkil Ketkar. Deep Learning with Python. Learn Best Practices of Deep Learning Models with PyTorch // Apress, 2020. p. 271
- [14] Schroeder, W., Oliva, P., Giglio, L., Csiszar, I. A. (2014). The New VIIRS 375m active fire detection data product: algorithm description and initial assessment. *Remote Sensing of Environment*, 143, 85-96.
- [15] Giglio, L., Descloitres, J., Justice, C. O. and Kaufman, Y. 2003. An enhanced contextual fire detection algorithm for MODIS. *Remote Sensing of Environment* 87:273-282. doi: 10.1016/S0034-4257(03)00184-6.
- [16] Kaufman, Y. J., C. O. Justice, L. P. Flynn, J. D. Kendall, E. M. Prins, L. Giglio, D. E. Ward, W. P. Menzel, and A. W. Setzer. "Potential global fire monitoring from EOS-MODIS." *Journal of Geophysical Research: Atmospheres* (1984–2012) 103, no. D24 (1998): 32215-32238.
- [17] Joseph de Mendoza y Ríos (1795). Memoria sobre algunos métodos nuevos de calcular la longitud por las distancias lunares: y application de su teórica á la solucion de otros problemas de navegacion (in Spanish). Madrid, Spain: Imprenta Real.

- [18] James Inman(1835) [1821]. Navigation and Nautical Astronomy: For the Use of British Seamen (3 ed.). London, UK: W. Woodward, C. J. Rivington. Retrieved 2015-11-09. (Fourth edition: [1].)
- [19] Акимов В. А., Соколов Ю. И. Пожарные риски России. - М., Всероссийский научно-исследовательский институт по проблемам гражданской обороны и чрезвычайных ситуаций МЧС России (Москва), 2016. - с. 174-230
- [20] <https://firms.modaps.eosdis.nasa.gov/download/>
- [21] Pastor-Pellicer J., Zamora-Martínez F., España-Boquera S., Castro-Bleda M.J. (2013) F-Measure as the Error Function to Train Neural Networks. In: Rojas I., Joya G., Gabestany J. (eds) Advances in Computational Intelligence. IWANN 2013. Lecture Notes in Computer Science, vol 7902. Springer, Berlin, Heidelberg

# Приложение

## Приложение № 1

```
#Обработка данных за текущий день
for row in mydf.itertuples():
    lat = int(row.latitude*10)/10
    lon = int(row.longitude*10)/10
    index = round((int(row.year)-2012)*for_year +
(int(row.day)-1)*num_squares + int((lat - startlat)/steplat)+
int((lon - startlon)/steplon)*num_lat_in_lon - 1)
    if (row.mo == '07'):
        index += for_june
    elif (row.mo == '08'):
        index += for_june + for_july
    full_df.loc[index, 'num_fires'] += 1
    full_df.loc[index, 'sum_ti4'] += row.bright_ti4
    if (full_df.loc[index, 'max_ti4'] < row.bright_ti4):
        full_df.loc[index, 'max_ti4'] = row.bright_ti4
full_df['is_fire'] = np.where(full_df['num_fires'] > 0, 1, 0)
full_df['mean_ti4'] = np.where(full_df['num_fires'] > 0,
full_df['sum_ti4'] / full_df['num_fires'], 0)
```

## Приложение № 2

```
#Обработка данных за предыдущий день
for index, row in full_df.iterrows():
    if (index >= num_squares):
        cur_index = index - num_squares
        full_df.loc[index, 'num_fires_yesterday'] =
full_df.loc[cur_index, 'num_fires'].copy()
        full_df.loc[index, 'mean_ti4_yesterday'] =
full_df.loc[cur_index, 'mean_ti4'].copy()
        full_df.loc[index, 'max_ti4_yesterday'] =
full_df.loc[cur_index, 'max_ti4'].copy()
        if (full_df.loc[index, 'lat'] == startlat):
```

```

if (full_df.loc[index, 'lon'] == startlon):
    indeces = [cur_index+numlat, cur_index+numlat+1,
               cur_index+1]
elif (full_df.loc[index, 'lon'] == endlon):
    indeces = [cur_index+1, cur_index-numlat+1,
               cur_index-numlat]
else:
    indeces = [cur_index-numlat, cur_index+numlat+1,
               cur_index+1, cur_index-numlat+1, cur_index+numlat]
elif (full_df.loc[index, 'lat'] == endlat):
    if (full_df.loc[index, 'lon'] == startlon):
        indeces = [cur_index-1, cur_index+numlat,
                   cur_index+numlat-1]
    elif (full_df.loc[index, 'lon'] == endlon):
        indeces = [cur_index-1, cur_index-numlat-1,
                   cur_index-numlat]
    else:
        indeces = [cur_index-numlat, cur_index-numlat-1,
                   cur_index-1, cur_index+numlat-1, cur_index+numlat]
elif (full_df.loc[index, 'lon'] == startlon):
    indeces = [cur_index+1, cur_index+numlat+1, cur_index-1,
               cur_index+numlat-1, cur_index+numlat]
elif (full_df.loc[index, 'lon'] == endlon):
    indeces = [cur_index-numlat, cur_index-numlat-1,
               cur_index-1, cur_index-numlat+1, cur_index+1]
else:
    indeces = [cur_index+numlat-1, cur_index-numlat,
               cur_index+numlat+1, cur_index+1, cur_index-numlat+1,
               cur_index+numlat, cur_index-numlat-1, cur_index-1]
for i in indeces:
    if (full_df.loc[i, 'is_fire']):
        full_df.loc[index, 'num_neigh_fires_y'] +=
        full_df.loc[i, 'num_fires'].copy()
        full_df.loc[index, 'num_sq_neigh_fires_y'] += 1

```

```

full_df.loc[index, 'sum_mean_ti4_neigh_y'] +=
full_df.loc[i, 'mean_ti4'].copy()
if (full_df.loc[index, 'max_ti4_neigh_yesterday'] <
full_df.loc[i, 'max_ti4']):
    full_df.loc[index, 'max_ti4_neigh_yesterday'] =
    full_df.loc[i, 'max_ti4'].copy()

```

### Приложение № 3

# Архитектура сети

```

def net(d_in, d_hid, n_hid, d_out):
    if d_in < 1 or d_hid < 1 or d_out < 1:
        raise ValueError("Размерность слоев должна быть больше
или равна 1")
    if n_hid < 0:
        raise ValueError("Число скрытых слоев не должно быть
отрицательным ")

    # Если число скрытых слоев равно нулю, то у нас однослойная сеть
    if n_hid == 0:
        return torch.nn.Linear(d_in, d_out)

    # Главные блоки
    hidlayer_first = [torch.nn.Linear(d_in, d_hid), torch.nn.ReLU()]
    hlayers = [torch.nn.Linear(d_hid, d_hid), torch.nn.ReLU()]
    out_layer = [torch.nn.Linear(d_hid, d_out)]

    all_layers = torch.nn.ModuleList()

    # Добавление слоев
    all_layers.extend(hidlayer_first)
    for i in range(n_hid - 1):
        all_layers.extend(hlayer)

    # Output layer

```

```
all_layers.extend(out_layer)

return torch.nn.Sequential(*all_layers)
```

## Приложение № 4

#Обучение

```
def fit(model, X_train, y_train, epochs, optim, lr):
    #Функция потерь
    loss_fun = torch.nn.BCEWithLogitsLoss()

    # Метод оптимизации
    optim = optim.lower()
    if optim == 'sgd':
        optimizer = torch.optim.SGD(model.parameters(), lr=lr)
    elif optim == 'rmsprop':
        optimizer = torch.optim.RMSprop(model.parameters(), lr=lr)
    elif optim == 'adam':
        optimizer = torch.optim.Adam(model.parameters(), lr=lr)

    for t in range(epochs):
        logits = model(X_train)

        loss = loss_fun(logits, y_train)

        # Обнуляем градиенты
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        pred = torch.sigmoid(logits)

        acc = torch.eq(y_train, pred.round_()).cpu().float().mean().item()
    return {'loss': loss.item(), 'acc': acc}
```

## Приложение № 5

#Кросс-валидация

```
def cross_valid(model, X_train, y_train, cv, epochs, optim, lr):
    kfolds = KFold(cv, False).split(X_train)

    score = []
    for fold, (train_idx, val_idx) in enumerate(kfolds):
        X = X_train[train_idx]
        y = y_train[train_idx]
        X_val = X_train[val_idx]
        y_val = y_train[val_idx]

        X = Variable(torch.Tensor(X), requires_grad=True)
        y = Variable(torch.Tensor(y), requires_grad=False).unsqueeze_(1)
        X_val = Variable(torch.Tensor(X_val), requires_grad=False)
        y_val = Variable(torch.Tensor(y_val), requires_grad=False).unsqueeze_(1)

        model_train = copy.deepcopy(model)

        metrics = fit(model_train, X, y, epochs=epochs, optim=optim,
                      lr=lr, verbose=0)

        y_val_pred = torch.sigmoid(model_train(X_val))
        acc = torch.eq(y_val, y_val_pred.round_()).cpu().float().mean()
        score.append(acc)
    return score
```

## Приложение № 6

```
def my_grid_search(X, y, param_grid, cv, epochs):
    grid = list((dict(zip(param_grid, param))
                 for param in itertools.product(*param_grid.values())))

    # Кросс-валидация для каждой комбинации гиперпараметров
```

```

best_score = 0
for candidate, params in enumerate(grid):
    d_in = X_train.shape[-1]
    d_hidden = params['d_hidden']
    n_hidden = params['n_hidden']
    d_out = 1
    model = titanic_net(d_in, d_hidden, n_hidden, d_out)

    cv_score = cross_val_score(model, X_train, Y_train, cv = n_folds,
                               use_cuda=use_cuda, verbose=0)
    cv_mean_acc = np.mean(cv_score)
    if verbose > 1:
        print("Mean CV accuracy: {0:.4f}".format(cv_mean_acc))
        print()

    if cv_mean_acc > best_score:
        best_params = params
        best_model = model
        best_score = cv_mean_acc

return {'best_model': best_model, 'best_params': best_params, 'best_score': best_score}

```

## Приложение № 7

```

#формула гаверсинусов
def haver(lat1, lon1, lat2, lon2):
    lat1 = float(lat1)
    lon1 = float(lon1)
    lat2 = float(lat2)
    lon2 = float(lon2)

    rad = float(pi / 180.0)

    lat = (lat2 - lat1) * rad
    lon = (lon2 - lon1) * rad

```

```
a = pow(sin(lat / 2), 2) + cos(lat1 * rad) * cos(lat2 * rad) * pow
c = 2 * atan2(sqrt(a), sqrt(1 - a))
km = 6367 * c
return km
```

## Приложение № 8

```
min_dist = 1000
station = 0
for index, row in df.iterrows():
    for r in need_stations.itertuples():
        dist = haversine(row.lat, row.lon, r.lat, r.lon)['km']
        if (dist < min_dist):
            min_dist = dist
            station = r.usaf
df.loc[index, 'station'] = station
min_dist = 1000
station = ''
```