СОДЕРЖАНИЕ

[2. Разработка универсальной гибридной системы интеллектуальной обработки экспериментальных данных на основе методов интеллектуального анализа данных с помощью нейросетей различных типов 2](#_Toc57295252)

[2.1 Вводные замечания 2](#_Toc57295253)

[2.2 Разработка проекта методики обработки экспериментальных данных на основе гибридных нейронечетких сетей и сетей глубокого обучения для больших массивов данных: создание и верификация первой версии базы знаний гибридной интеллектуальной системы поддержки принятия решений и обработки экспериментальных данных, полученной на основе системы интеллектуального анализа данных с участием специалистов-радиологов. 3](#_Toc57295254)

[2.2.1. Применение нечетких нейронных сетей для идентификации и прогнозирования экспериментальных данных 3](#_Toc57295255)

[2.2.2. Применение глубоких нейронных сетей для идентификации и прогнозирования экспериментальных данных 15](#_Toc57295256)

[2.3 Разработка модуля предварительной обработки экспериментальных данных для ОВРН - альфа-версия. 20](#_Toc57295257)

[2.3.1. Модуль предварительной обработки экспериментальных данных на основе нейро-нечетких сетей 20](#_Toc57295258)

[2.3.2. Модуль предварительной обработки экспериментальных данных на нейросетей глубокого обучения 28](#_Toc57295259)

[2.4. Проведение тестовых расчетов. 30](#_Toc57295260)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ В ГЛАВЕ 2 43](#_Toc57295261)

[ЗАКЛЮЧЕНИЕ 45](#_Toc57295262)

# 2. Разработка универсальной гибридной системы интеллектуальной обработки экспериментальных данных на основе методов интеллектуального анализа данных с помощью нейросетей различных типов

### 2.1 Вводные замечания

В разделе решается задача: как на основе входных данных экспериментов по ОВРН, выраженных в числовой форме, например, доза облучения, интенсивность нагрева, и соответствующими выходными данными – процентом выживаемости клеток в экспериментах – восстановить или аппроксимировать общую функцию зависимости выходной переменной от входных переменных и вычислить априорно наиболее эффективные входные параметры.

На этапе 2019 года была проведена разработка методов обработки экспериментальных данных и моделирования выживания клеток при гипертермии и радиотерапии опухолей. Задача состояла в рассмотрении набора неструктурированных («сырых») данных из опытов по ОВРН и разработке средств по установлению истинной зависимости выходных данных от входных по визуализации этой зависимости и по поиску скрытых зависимостей в данных. Также был проведен анализ имеющихся экспериментальных данных о совместном действии гипертермии и облучения на клетки опухоли.

В 2020 году разрабатывался проект методики обработки экспериментальных данных на основе гибридных нейро-нечетких сетей и сетей глубокого обучения для больших массивов данных: создание и верификация первой версии базы знаний гибридной интеллектуальной системы поддержки принятия решений и обработки экспериментальных данных, полученной на основе системы интеллектуального анализа данных с участием специалистов-радиологов. Была разработана архитектура нейро-нечетких сетей и сетей глубокого обучения и на этой основе создана программная среда на языке Python 3.7 на основе нейро-нечетких сетей и сетей глубокого обучения для аппроксимации и прогнозирования результатов эксперимент.

### 2.2 Разработка проекта методики обработки экспериментальных данных на основе гибридных нейронечетких сетей и сетей глубокого обучения для больших массивов данных: создание и верификация первой версии базы знаний гибридной интеллектуальной системы поддержки принятия решений и обработки экспериментальных данных, полученной на основе системы интеллектуального анализа данных с участием специалистов-радиологов.

### 2.2.1. Применение нечетких нейронных сетей для идентификации и прогнозирования экспериментальных данных

На основе теоретических исследований и компьютерного моделирования, выполненных нами в 2019, можно утверждать, что нечеткие нейро-сетевые модели являются удобным инструментом для планирования и оптимизации режимов радиационного и гипертермического воздействия на опухолевые клетки. С другой стороны при увеличении объема экспериментальных данных для их обработки требуются более мощные сети глубокого обучения. Поэтому на их основе был разработан проект методики обработки экспериментальных данных на основе гибридных нейро-нечетких сетей и сетей глубокого обучения для больших массивов данных

Применение теории нечетких множеств для прогнозирования временных рядов стало активно развиваться с начала 90-х годов за рубежом. Сначала это были экспертные системы с использованием нечетких множеств, так называемые нечеткие экспертные системы. Последующее развитие привело к появлению гибридных систем (экспертная система и нейронная сеть) и нейронных сетей с нечеткой логикой.

Рассматривая архитектуры нейро-нечетких моделей, можно выделить три методики объединения искусственных нейронных сетей (ИНС) и нечетких моделей [1, 2]:

1. neuro-FIS, в которых ИНС используется как инструмент в нечетких моделях;

2. нечеткие ИНС, в которых классические модели ИНС фазифицированы;

3. нейро-нечеткие гибридные системы, в которых нечеткие системы и ИНС объединены в гибридные системы [3, 4].

Исходя из данных методик, нейро-нечеткие модели можно разделить на три [5, 6, 7] класса:

1. Кооперативные нейро-нечеткие модели. В данном случае, часть ИНС изначально используется для определения нечетких множеств и / или нечетких правил, где впоследствии выполняется только полученная нечеткая система. В процессе обучения определяются функции принадлежности, а также формируются нечеткие правила на основе обучающей выборки. Здесь, основная задача нейронной сети заключается в подборе параметров нечеткой системы.

2. Параллельные нейро-нечеткие модели. Нейронная сеть в данном типе модели работает параллельно с нечеткой системой, предоставляя входные данные в нечеткую систему или изменяя выходные данные нечеткой системы. Нейронная сеть здесь может являться также и постпроцессором выходных данных из нечеткой системы.

3. Гибридные нейро-нечеткие модели. Нечеткая система использует метод обучения, как это делает и ИНС, чтобы настроить свои параметры на основе обучающих данных. Среди представленных классов моделей, наибольшей популярностью пользуются модели именно данного класса, доказательством тому служит их применение в широком спектре реальных задач [8, 9, 10, 11].

Среди наиболее популярных гибридных моделей можно выделить такие архитектуры как:

• Fuzzy adaptive learning control network (FALCON) [12] имеет пятислойную архитектуру. На одну выходную переменную приходится по два лингвистических узла. Первый узел работает с обучющей выборкой (паттерном обучения), второй является входным для всей системы. Первый скрытый слой размечает входную выборку в соответствии с функциями принадлежности. Второй слой определяет правила и их параметры. Обучение происходит на основе гибридного алгоритма без учителя для определения функции принадлежности, базы правил и использует алгоритм градиентного спуска для оптимизации и подбора итоговых параметров функции принадлежности.

• ANFIS [13] - это хорошо известная нейронечеткая модель, которая использовалась во многих приложениях и исследовательских областях [14]. Более того, сравнение архитектур нейро-нечетких сетей показало (Таблица 2.1), что ANFIS показываем минимальную ошибку в задаче прогнозирования. Основным недостатком модели ANFIS является то, что она накладывает серьезные требования к вычислительной мощности [15].

• Generalized approximate reasoning-based intelligent control (GARIC) [16] представляет собой нейро-нечеткую систему, использующую два нейросетевых модуля, модуль выбора действия ( Action Selection Network) и модуль оценки состояния (Action State Evaluation Network), который отвечает за оценку качества выбора действий предыдущим модулем. GARIC является пятислойной сетью прямого распространения.

• Neural fuzzy controller (NEFCON) [17] был разработан для реализации системы нечеткого вывода типа Мамдани. Связи определяются с помощью нечетких правил. Входной слой является фаззификатором, а выходной решает задачу дефаззификации. Обучается сеть на основе гибридного алгоритма обучения с подкреплением и алгоритма обратного распространения ошибки.

• Fuzzy inference and neural network in fuzzy inference software (FINEST) [18] представляет собой систему настройки параметров. Производится настройка нечетких предикатов, функции импликации и комбинаторной функции.

• Self-constructing neural fuzzy inference network (SONFIN) [19] по своей сути аналогичен NEFCON контроллеру, но вместо реализации нечеткого вывода типа Мамдани, реализует тип Такаги-Сугено. В данной сети входная выборка обрабатывается на основе алгоритма кластеризации (aligned clustering). реализует модифицированный FIS Такаги-Сугено и проиллюстрирован на рисунке 10. При идентификации структуры части предварительного условия входное пространство разделяется гибким образом в соответствии с алгоритмом, основанным на выровненной кластеризации (aligned clustering). Настройка параметров системы частично реализована на основе метода наименьших квадратов, предварительные условия настраиваются с помощью метода обратного распространения ошибки.

• Dynamic evolving fuzzy neural network (dmEfuNN) и (EFuNN) [20]. В EFuNN все узлы создаются в процессе обучения. Первый слой передает обучающие данные на второй, который вычисляет степень соответствия с заранее определенной функцией принадлежности. Третий слой содержит в себе наборы нечетких правил, являющихся прототипами входных-выходных данных, которые можно представить в качестве гиперсфер нечеткого входного и выходного пространств. Четвертый слой рассчитывает степень, с которой выходная функция принадлежности разметила входные данные, а пятый слой производит дефаззификацию и подсчитывает числовые значения выходной переменной. DmEfuNN представляет собой модифицированную версию EFuNN. Основная идея состоит в том, что для всех входных векторов динамически подбирается набор правил, значения активации которых используются для расчета динамических параметров выходной функции. В то время как EFuNN реализует нечеткие правила типа Мамдани, dmEFuNN использует тип Такаги-Сугено.

В приведенном выше обзоре мы рассмотрели большинство типов известных нейро-нечетки моделей, среди них стоит особо отметить гибридный тип, который можно рассматривать шагом к разработке систем объяснительного искусственного интеллекта. В данном типе систем объединяются модели нечеткого вывода с искусственными нейронными сетями, что позволяет решить известную проблему нейронных сетей – проблему черного ящика. Целесообразно сравнить также производительность рассмотренных гибридных моделей. В Таблица 2.1 приведены результаты сравнения ряда моделей в задаче прогнозирования известного временного ряда Макки-Гласса [21]. Обучающая выборка состоит из 500 входных элементов, тестовая выборка содержала также 500 элементов.

Таблица 2.1. Сравнение гибридных нейро-нечетких моделей

|  |  |  |
| --- | --- | --- |
| Модель | Количество эпох | Средняя квадратичная ошибка |
| ANFIS | 75 | 0.0017 |
| NEFPROX | 216 | 0.0332 |
| EFuNN | 1 | 0.0140 |
| dmEFuNN | 1 | 0.0042 |
| SONFIN | 1 | 0.0180 |

В приведенном выше обзоре мы рассмотрели большинство типов известных нейро-нечетки моделей, среди них стоит особо отметить гибридный тип, который можно рассматривать шагом к разработке систем объяснительного искусственного интеллекта. В данном типе систем объединяются модели нечеткого вывода с искусственными нейронными сетями, что позволяет решить известную проблему нейронных сетей – проблему черного ящика. Имеющиеся на настоящий момент данные по совместному действия гипертермии и облучения на опухоль доказывают, что физиологический механизм воздействия этих методов на опухолевые клетки в целом известен и может быть успешно промоделирован. Можно утверждать, что нечеткие системы и глубокие нейронные сети могут быть эффективным инструментом для планирования и оптимизации режимов гипертермии и облучения. Планируется продолжение исследований в области моделирования выживания клеток при гипертермии и гамма-терапии опухолей.

Модель нечеткой нейронной сети для прогнозирования временных рядов представлена на рисунке 2.1.



Рисунок 2.1. Модель нечеткой нейронной сети для прогнозирования временных рядов

Сеть состоит из трех слоев: входного, скрытого (слоя правил вида ЕСЛИ – ТО) и выходного слоя. Нейроны, изображенные в виде кругов, представляют собой один узел, а нейроны в виде квадратов имеют в своем составе большое количество узлов. Каждая пара нейронов в скрытом слое составляет нечеткое правило. Выбрана следующая форма правил, в которой нечеткие множества включаются только во вступительную часть правил:

П*j*: ЕСЛИ  ТО ,

где П*j*: – *j*-е правило; *xi*– входные переменные;  – значение *k*-ого выхода, вычисленные по *j*-му правилу;  –  логические члены, характеризующиеся функциями принадлежности ;  – весовые коэффициенты.

В подслое «ЕСЛИ» вычисляется значение функции принадлежности для каждой входной переменной следующим образом:

,

где  – функция принадлежности;  – отклонение.

Эта функция имеет колоколообразную форму с центром  и отклонением . Результирующее значение функции принадлежности по каждому правилу определяется как:



или

.

Нейроны *j* в подслое «ТО» представляют собой двухслойную нейронную сеть, в которой весовыми коэффициентами являются коэффициенты , а  – коэффициент смещения. Выход *j* нейрона определяется выражением:

.

В выходном слое на основании результатов, полученных в слое правил, формируется четкий вывод как взвешенное среднее по формуле:

,

где *Pk* – прогнозируемое значение временных рядов; *M* – количество правил.

В отличие от типовых ИНС здесь начальные значения параметров сети выбираются не произвольно, а задаются максимально близкими к их реальным возможным значениям на основании специальных расчетов, что дает существенное преимущество по скорости обучения. Основным достоинством нечетких нейросетевых методов для прогнозирования временных рядов является возможность заполнения базы правил экспертом с последующим изменением. Однако не существует стандартной методики проектирования и расчета нечетких нейронных систем, что в некоторых случаях усложняет процесс прогнозирования временных рядов.

Временной ряд представляет собой последовательность упорядоченных в равноотстоящие моменты времени пар. Нечетким временным рядом называют упорядоченную в равноотстоящие моменты времени последовательность наблюдений некоторого процесса, состояния которого изменяются во времени, если значение состояния процесса в некоторый момент времени может быть выражено с помощью нечеткой метки временных рядов В отличие от традиционного временного ряда значениями нечеткого временного ряда являются нечеткие множества, а не действительные числа.

Математическую основу нечетких временных рядов составляют нечеткие модели и теоретические выводы, в частности, теорема о нечеткой аппоксирмации, согласно которой функциональная зависимость, заданная на компактном множестве, может быть аппроксимирована нечеткой моделью. В 1992 году Ванг показал, что нечеткая модель вида «ЕСЛИ-ТО» является универсальным аппроксиматором, то есть может приближать любую непрерывную функцию с произвольной точностью. В 1995 году Кастро доказал справедливость свойства универсальной аппроксимации для широкого спектра нечетких моделей. Таким образом, произвольная взаимосвязь может быть бесконечно точно описана с помощью формализованных высказываний естественного языка «ЕСЛИ-ТО». Структура нейро-нечеткой продукционной модели ANFIS, которая реализует описанный выше функционал, представлена на рисунке 2.2.



Рисунок 2.2. Структура нейронечеткой продукционной модели типа ANFIS

Программные средства прогнозирования временных рядов на основе нечетких временных рядов реализуют предложенный способ с учетом коррелирующих факторов. В основу решения положена модель программных средств, которая характеризуется модульностью в соответствии с нижеперечисленными этапами работы.

Этап 1. Установление зависимостью между параметрами многомерных временных рядов на основе источников данных и экспертных заключений.

Этап 2. Настройка прогнозной модели на основе нечетких временных рядов.

Этап 3. Инициализация и обучение прогнозной модели на этапах 1, 2.

Этап 4. Выполнение прогнозирования временных рядов в соответствии с заданной глубиной прогнозирования.

Этап 5. Вывод прогнозных данных и, если возможно, сравнение прогнозных данных с эталонными значениями.

Структура программных средств в соответствии с вышеперечисленными этапами , прогнозирования на основе нечетких временных рядов представлена на рисунке 2.3.



Рисунок 2.3. Структура программных средств прогнозирования на основе нечетких временных рядов

Источником данных может являться база данных, таблицы данных, специализированные файлы данных и т.д. Данные загружаются в виде временных рядов. Из множества ВР составляется зависимость, после построения которой формируется обучающая выборка и «горизонт» возможного прогнозирования.

Модуль установления зависимостей прогнозных моделей устанавливает зависимость между предыдущими значениями временных рядов, корреляционными временными рядами, таких параметров как температура, влажность, а также со значением временного ряда на следующем этапе. После применения установленной зависимости к модели прогнозирования выполняется формирование обучающей выборки на основе ретроспективных данных, взятых из источников данных.

Модуль настройки прогнозных моделей осуществляет настройку прогнозных моделей для прогнозирования временных рядов на основе нечетких временных рядов.

Для формирования базы нечетких правил и нечетких множеств лингвистических переменных используется алгоритм «горной» кластеризации, входными параметрами которого являются входные временные ряды обучающей выборки прогнозной модели и параметры кластеризации.

Для настройки параметров обучения прогнозной модели необходимо указать коэффициент обучения и его изменение на каждой итерации обучения, а также методы пересечения и объединения нечетких множеств нечетких переменных. Корректировка лингвистических термов нечетких переменных и базы нечетких правил доступна после обучения прогнозной модели.

После осуществления прогноза происходит верификация данных, если есть эталонные данные, и вывод полученных результатов.

Модуль установления зависимостей работает на основе временных рядов, хранимых в базе данных. Временные ряды преобразуются в переменные, которые содержат имя, значения временного ряда и численные параметры: максимальное, минимальное и среднее значение ряда.

Эксперт устанавливает зависимость между входными и выходными переменными. Если значения временных рядов зависят от предыдущих элементов, в таком случае эксперт задает параметры авторегрессии, который указывается в зависимости от параметров сезонности и может быть получен на основе автокорреляционного анализа. Параметр авторегрессии выходной переменной не может быть меньше параметров авторегрессии, указанных для этой переменной во множестве входных переменных.

Выходные переменные зависят от входных переменных, количество которых определяется на основании предварительных исследований и задаётся экспертом. После установления зависимости формируются параметры обучающей выборки.

На основании авторегрессионных сдвигов и количества элементов переменных, используемых в установленной зависимости, формируются границы обучающей выборки и «горизонт» прогнозирования. В случае, если в зависимости участвует единственная переменная, «горизонт» прогнозирования не ограничивается и задается экспертом. После определения «горизонта» прогнозирования и границ обучающей выборки формируется векторы входных и выходных данных для обучения прогнозной модели.

Горная кластеризация или кластеризация без задания количества кластеров не является нечеткой, однако часто используется при генерации лингвистических термов нечетких переменных и нечетких правил. Алгоритм горной кластеризации выполняется в несколько этапов.

На первом этапе алгоритма «горной» кластеризации происходит поиск максимальных значений элементов на каждом кластере. По умолчанию этот элемент выбирается как первый найденный кластер.

На втором этапе осуществляется нормализация данных, т.е. приведение всех значений исходных данных к единичному гиперкубу.

На третьем этапе формируются потенциальные центры кластеров. Их количество должны быть конечным. В качестве центров могут выступать объекты кластеризации. Потенциал кластера зависит от расстояния между потенциальным кластером и другими объектами кластеризации.

В качестве центра кластера назначают точку с наивысшим потенциалом, которая в большинстве случаев окружена другими достаточно высокими пиками. В связи с этим, назначение центром следующего кластера с максимальным потенциалом среди оставшихся вершин привело бы к размножению кластеров. Для поиска следующего кластера необходимо исключить его влияние на остальные вершины. Поэтому из потенциала кластеров вычитают потенциал найденного кластера.

На четвертом этапе происходит перерасчет потенциалов оставшихся кластеров.

Этапы 3, 4 происходят итеративно, пока максимальное значение кластера превышает заданный порог. После окончания формирования центров кластеров исходные данные и центры кластеров денормализуются.

UML-диаграмма деятельности алгоритма «горной» кластеризации представлена на рисунке 2.4. Найденные центры кластеров являются нечеткими термами лингвистических переменных. Ширина лингвистических термов нечетких лингвистических переменных зависит от радиуса найденных кластеров. Обучение нейро-нечеткой модели происходит при фиксированных лингвистических переменных и базы нечетких правил, полученных на основе алгоритма горной кластеризации.



Рисунок 2.4. UML-диаграмма деятельности обучения нейро-нечеткой модели  
для прогнозирования временных рядов

Представленный алгоритм является гибридным и одним из наиболее эффективных способов обучения нечетких продукционных моделей.

UML-диаграмма деятельности гибридного алгоритма обучения представлена на рисунке выше. Его главная особенность состоит в разделении процесса обучения на два обособленных во времени этапа. На каждом этапе уточняется только часть параметров модели. Если принять во внимание, что вычислительная сложность каждого алгоритма зависит от числа параметров, то уменьшение размерности задачи оптимизации существенным образом сокращает количество операций и увеличивает скорость выполнения алгоритма. Благодаря этому гибридный алгоритм обучения значительно эффективней, чем обычный градиентный алгоритм фронтального типа, согласно которому уточнение всех параметров сети проводится параллельно и одновременно

На основе модели временного ряда предлагается способ анализа прогнозирования временного ряда, который предполагает разделение ряда на локальные области и дальнейшую работу с текущей локальной областью. Обобщенная структура способа анализа и прогнозирования ВР представлена на рисунке ниже.

Предлагаемый способ анализа и последующего прогнозирования значений временного обобщенно можно разбить на несколько этапов.

На первом этапе осуществляется нормировка данных, которая влияет на точность прогнозирования временных рядов. Нормировка необходима для адекватного применения математических моделей и компьютерных расчетов при вычислениях, связанных с большими и малыми величинами. Для равномерного распределения элементов временного ряда их значения представляются в диапазоне [0, 1].

На втором этапе выполняется фаззификация входных переменных. Процесс фаззификации заключается в соотнесении входной переменной и функции принадлежности из некоторого набора термов лингвистической переменной. Лингвистическая переменная задается непосредственно экспертом или адаптивно при обучении на основе обучающей выборки. Для фаззификации используются колоколообразная, треугольная или трапециевидная функции принадлежности. Такие функции широко используются при анализе факторов.

В случае, если измеряемые данные были искажены шумом, функции принадлежности следует значительно расширить, чтобы уменьшить чувствительность. Степень истинности определяется путем соотношения прогнозного значения и эталонного значения. На третьем этапе осуществляется агрегирование подусловий. На этом этапе осуществляется определение степени истинности условий для каждого правила модели нечеткого вывода. Набор правил нечеткого вывода в базе знаний задается экспертом или адаптивно при обучении на основе обучающей выборки.

На четвертом этапе выполняется переход от условий к подзаключениям (активизация подзаключений). Для каждого подзаключения находится степень истинности.

На пятом этапе осуществляется аккумуляция заключений. На этом этапе получается нечеткое множество (или их объединение) для каждой из входных переменных.

На шестом этапе выполняется дефаззификация выходных переменных. На этапе дефаззификации выполняет преобразование нечеткого значения, полученного при нечетком выводе, в четкое. Этап дефаззификации считается законченным, когда для каждой из выходных лингвистических переменных будут определены итоговые количественные значения в виде некоторого действительного числа. Для дефаззификации могут быть использованы следующие методы: метод центра тяжести, метод биссектрисы площади, метод правого модального значения функции принадлежности и другие.

### 2.2.2. Применение глубоких нейронных сетей для идентификации и прогнозирования экспериментальных данных

Современные поколения искусственных нейронных сетей представляют собой гораздо более сложные и гибкие архитектуры, чем нейронные сети первых поколений. Все ведущие научные организации мира активно занимаются развитием искусственного интеллекта, расширением областей его применения, усовершенствованием существующих искусственных нейронных сетей. Решая задачу развития искусственных нейронных сетей, следует решать ряд сложных математических задач, задач оптимизации, разрабатывать наиболее быстрые и гибкие алгоритмы обучения.

Область применения искусственных нейронных сетей (ИНС) колоссальна, и она продолжает расширяться. Сегодня ИНС применяются в задачах оптимизации, в задачах анализа и прогнозирования, в задачах распознавания образов, распознавании эмоций и даже анализе поведения объекта, с целью прогнозирования его дальнейших действий. От банковской сферы до медицины и обороны, везде ИНС применяются довольно широко и показывают впечатляющие результаты.

Среди наиболее активно развивающихся моделей искусственных нейронных сетей стоит выделить глубинные модели нейронных сетей (Deep Learning Neural Networks). Подобные модели нейронных сетей уже доказали свою эффективность в задачах распознавания и классификации изображений. Наиболее популярной архитектурой является так называемая сверточная нейронная сеть (Convolutional Neural Network). Технологические гиганты выделяют миллиарды долларов на исследования и разработки в области искусственного интеллекта. Примером может служить сверточная нейронная сеть. я сеть, разработанная компанией Google – GoogLeNet. ,

Глубинные нейронные сети или Deep Learning NN представляют собой класс, основанный на методах машинного обучения. Глубокое обучение основано на обучении признакам, а не на специализированных алгоритмах, разработанных для конкретных задач. Многие методы глубокого обучения были известны ещё в 1980-е, но результаты были невпечатляющими, пока продвижения в теории искусственных нейронных сетей (предобучение нейросетей с помощью специального случая ненаправленной графической модели, так называемой ограниченной машины Больцмана) и вычислительные мощности середины 2000-х (прежде всего, графических процессоров Nvidia, а в настоящее время и тензорных процессоров Google) не позволили создавать сложные технологические архитектуры нейронных сетей, обладающие достаточной производительностью и позволяющие решать широкий спектр задач, не поддававшихся эффективному решению ранее, например, в компьютерном зрении, машинном переводе, распознавании речи, причём качество решения во многих случаях теперь сопоставимо, а в некоторых случаях превосходит эффективность специалистов-экспертов. В отличие от машинного обучения, глубинное обучение требует гораздо большего объема обучающей выборки, нежели в случае с машинным обучением. Также, в отличие от машинного обучения, глубинная нейронная сеть может иметь тысячи слоев. Все это помогает глубинным нейронным сетям достигать достаточно высокой точности в задачах анализа, классификации и распознавания изображений.

Наиболее известным примером сети глубокого обучения является сверточная нейронная сеть (convolutional neural network). Данная архитектура искусственных нейронных сетей, предложенная Яном Лекуном в 1988 году и нацеленная на эффективное распознавание изображений.

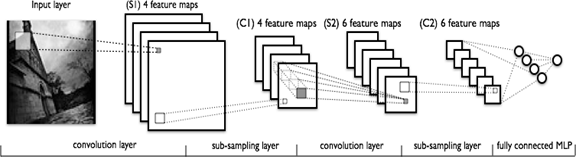


Рис.2.5. Сверточная нейронная сеть

Пожалуй, наиболее знаменитым примером сверточной нейронной сети можно назвать сеть AlexNet и GoogLeNet. Последняя выиграла конкурс ImageNet recognition challenge в 2014-м году с результатом 6.67% top 5 error. Напомним, что top 5 error — метрика, в которой алгоритм может выдать 5 вариантов класса картинки, и ошибка засчитывается, если среди всех этих вариантов нет правильного. Всего в тестовой выборке 150000 картинок и 1000 категорий, то есть задача крайне нетривиальна.

Способность CNN изучать и автоматически извлекать особенности из необработанных входных данных может успешно применяться к задачам прогнозирования временных рядов. Последовательность наблюдений можно рассматривать как одномерное изображение, которое модель CNN может с легкостью обрабатывать. Нейронные сети с глубоким обучением могут автоматически изучать произвольные сложные сопоставления от входов к выходам и поддерживать несколько входов и выходов.

Сверточная нейронная сеть является частью технологии глубокого обучения. Это многослойные и однонаправленные (без обратных связей) сети. Идея сверточных нейронных сетей заключается в том, чтобы чередовать сверточные и субдискретизирующие слои.

Сверточная нейронная сеть – это нейронная сеть прямого распространения, значит, нейроны в этой сети разбиты на группы, называемые слоями. Активация слоев в подобных многослойных нейронных сетях происходит по следующему сценарию: вычисление значений этих переменных (активаций слоев) выполняется последовательно: от первого до последнего слоя. Выходами нейронной сети как раз и являются значения активации последнего слоя. При этом нейронная сеть обладает множеством параметров, каждый слой имеет некоторые свои параметры, от которых зависит активация последующего слоя. Однако, внутри слоев активации подсчитываются независимо и параллельно , из-за этого, современные процессоры и графические сопроцессоры могут очень эффективно проводить вычисления с такими нейронными сетями.

Отличительной чертой сверточной нейронной сети является то, что нейроны первых уровней имеют такую структуру, где нейроны первичных слоев разбиты на изображения определенного размера (их еще иногда называют картами). Разные карты внутри одного слоя соответствуют нейронам разного типа, которые реагируют на разные особенности изображений. Существует два основных метода вычислений активации следующего слоя в сверточных нейронных сетях.

При первом типе вычисления выполняются в как линейная комбинация активации нейронов, расположенных на предыдущем слое, веса не зависят от положения нейрона на карте, на значение активации нейрона оказывает влияние только тип нейрона и их взаимное положение.

При втором типе вычислений активация нейронов на следующем слое копирует значения предыдущих, при этом изображение уменьшается за счет того, что активация ближних друг к другу нейронов заменяется на их максимальное или среднее значения, этот процесс называется пулингом. Благодаря этой особенности сверточные нейронные сети являются очень эффективными в работе с изображениями. К примеру, если два изображения различаются небольшим сдвигом, то она обеспечит получение на выходе очень похожего результата. Кроме того, сверточные нейронные сети обладают небольшим количеством параметров в сравнении с количеством нейронов. Если сравнить количество параметров с обычными, не сверточными нейронными сетями, то это сравнение будет не в пользу последних, так как при аналогичном количестве нейронов, сверточные нейронные сети используют на порядок меньше параметров. Например, в современных сверточных сетях может использоваться миллионы или даже десятки миллионов параметров, в то время как обычные нейронные сети, при примерно том же количестве нейронов, использовали бы сотни миллиардов параметров. В связи с этим, возникает проблема поиска обучающих данных для обучения. Сверточным нейронным сетям с таким же количеством нейронов требуется для обучения гораздо меньше наборов данных, они могут обучаться по существующим выборкам, именно это делает их такими успешными.

Существенной особенностью архитектур сверточной нейронной сети является то, что размеры ввода становятся все меньше и меньше от начала до конца сети, а количество каналов становится больше. При этом базовая архитектура сверточных нейронных сетей не сильно изменилась со времен первых моделей. Сверточные нейронные сети могут иметь множество типов слоев, основными из которых являются:

Входной слой: входное изображение, включая несколько цветовых каналов.

Сверточный слой: все нейроны слоя, в отличие от персептрона, связаны только с частью нейронов предыдущего слоя.

Слой субдискретизации: выделение наиболее значимых признаков предыдущего слоя и значительное сокращение размерности последующих слоев сети.

Полносвязный слой: представляет собой скрытый слой искусственной нейронной сети типа персептрон.

После сверточных и полносвязных слоев возможно применение функции активации нейронов, преобразующей сигнал нейрона и формирующей выходной сигнал.

Рассмотрим типовую структуру свёрточной нейронной сети более подробно. Сеть состоит из большого количества слоёв. После начального слоя (входного изображения) сигнал проходит серию свёрточных слоёв, в которых чередуется собственно свёртка и субдискретизация (пулинг). Чередование слоёв позволяет составлять «карты признаков» из карт признаков, на каждом следующем слое карта уменьшается в размере, но увеличивается количество каналов. На практике это означает способность распознавания сложных иерархий признаков. Обычно после прохождения нескольких слоёв карта признаков вырождается в вектор или даже скаляр, но таких карт признаков становятся сотни. На выходе свёрточных слоёв сети дополнительно устанавливают несколько слоёв полносвязной нейронной сети (перцептрон), на вход которому подаются оконечные карты признаков.

Различают два концептуальных подхода к обучению искусственных нейронных сетей:

обучение с учителем, предполагающее, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе эти вектора представляют обучающую пару.

Обучение без учителя, при котором обучающее множество состоит из входных векторов, а алгоритм обучения нейронной сети подстраивает веса сети так, чтобы получались согласованные выходные векторы, чтобы предъявление достаточно близких входных векторов давало одинаковые выходы.

Хотя CNN традиционно разрабатывались анализа двумерных изображений, их можно использовать для моделирования задач прогнозирования одномерных временных рядов. Модель CNN будет изучать функцию, которая отображает последовательность прошлых наблюдений в качестве входных данных для выходных наблюдений. Таким образом, последовательность наблюдений должна быть преобразована в несколько примеров, на которых модель может учиться.

Одномерная CNN - это модель CNN, которая имеет сверточный скрытый слой, который работает над одномерной последовательностью. За этим, возможно, следует, в некоторых случаях, второй сверточный слой, например, очень длинные входные последовательности, а затем слой объединения, задача которого заключается в дистилляции выходных данных сверточного слоя до наиболее значимых элементов. Сверточный и объединяющий слои сопровождаются плотным полностью связанным слоем, который интерпретирует признаки, извлеченные сверточной частью модели. Сглаженный слой используется между сверточными слоями и плотным слоем, чтобы свести карты функций к одному одномерному вектору.

### 2.3 Разработка модуля предварительной обработки экспериментальных данных для ОВРН - альфа-версия.

### 2.3.1. Модуль предварительной обработки экспериментальных данных на основе нейро-нечетких сетей

Разработанная программная библиотека реализована на языке программирования *python*3+. *UML*-диаграммы классов разработанной программной библиотеки представлены на рисунках 2.6-2.11

На рисунке 2.6 отображена диаграмма классов функций принадлежности нечетких переменных.

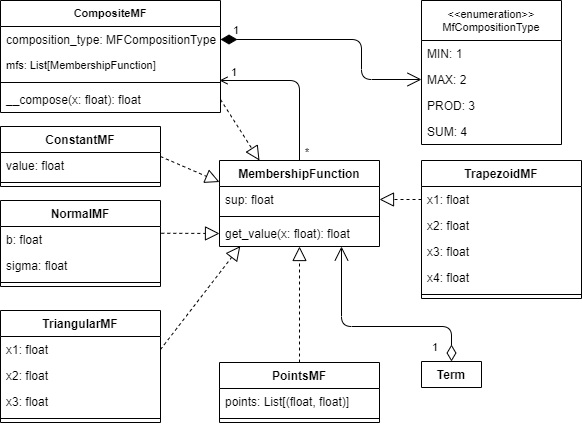


Рисунок 2.6– Диаграмма классов функций принадлежности

Абстрактный класс *MembershipFunction* содержит свойство *sup* типа *float* и метод *get\_value(x)*, который возвращает значения функции принадлежности в зависимости от указанного входного числа. В библиотеке программных средств реализованы самые распространенные лингвистические термы. В случае необходимости можно лингвистические термы можно расширить за счет частных реализаций, которые являются наследниками абстрактного класса *MembershipFunction.*

Стоит заметить, что на диаграмме классов функций принадлежности представлен класс *CompositeMF*, который содержит в себе массив элементарных функций принадлежности и объединяет их по методу пересечения, представленному в классе перечислений *MFCompositionType*.

На рисунке 2.7 представлена диаграмма классов нечетких продукционных правил.

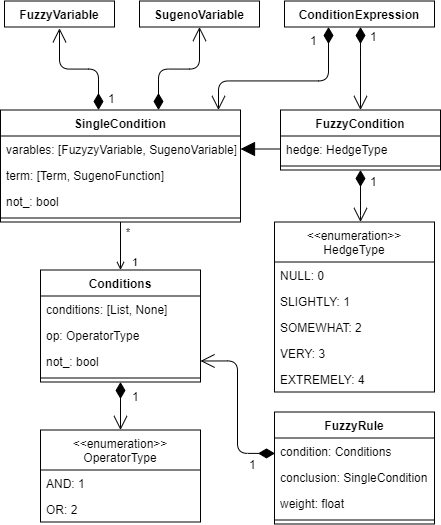


Рисунок 2.7– Диаграмма классов нечетких продукционных правил

Реализация нечетких продукционных классов начинается с класса, который осуществляет обобщенное представление нечеткого продукционного правила *SingleCondition*. В обобщенном представлении содержатся входные нечеткие переменные *variables*, лингвистический терм *term* и показатель инверсии *not\_*. Кроме того, нечеткое правило может быть модифицировано усилением, реализованным в классе *FuzzyCondition*. Параметр модификации представлен в классе перечисления *HedgeType*. В случае, если нечеткое правило имеет сложную структуру, включающую в себя несколько элементарных правил, используется класс *Conditions* с модификатором перечисления этих правил, который отображен классом перечислений *OperatorType*. Нечеткие продукционные правила нейро-нечетких продукционных моделей реализуются с помощью класса *FuzzyRule*, который состоит из условия *condition*, результата правила *conclusion* и веса правила *weight*, который в общем случае равен единице.

Диаграмма классов парсера нечетких правил показана на рисунке 2.8.

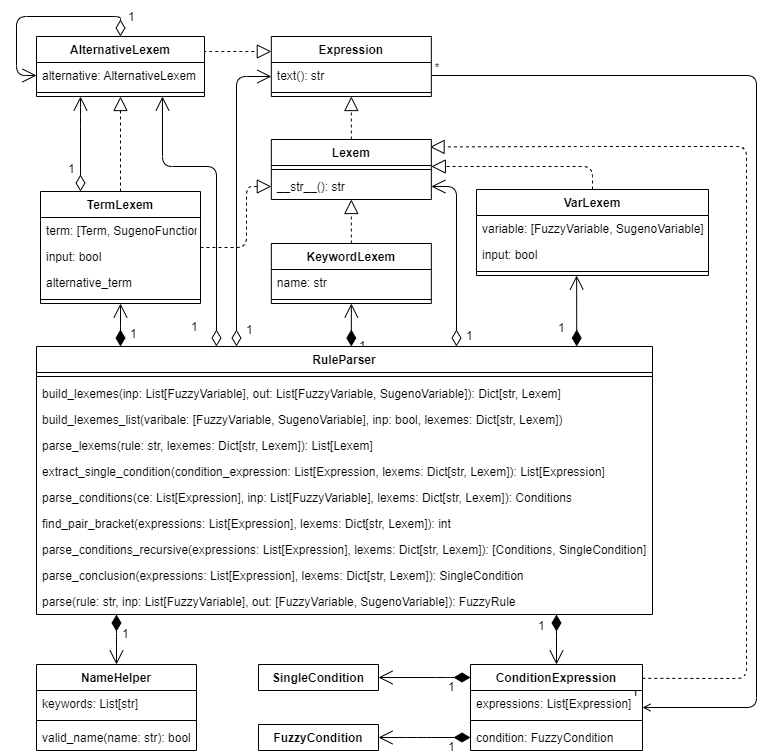


Рисунок 2.8 – Диаграмма классов парсера нечетких продукционных правил

Для выявления нечеткого продукционного правила из строки реализован класс *RuleParser*, который включает в себя классы абстрактных лексем *Expression*, *AlternativeLexem*, *Lexem* и их наследников *TermLexem*, *KeywordLexem*, *VarLexem*. Точкой входа разбора нечеткого продукционного правила является функция *parse* из класса *RuleParser*, которая принимает строковое обозначение правила и отвечает за построение абстрактного синтаксического дерева. Ключевые слова парсера правил отображены в классе *NameHelper*: *if*, *then*, *is*, *and*, *or*, *not*, «(», «)», *slightly*, *somewhat*, *very*, *extremely*.

Обобщенное представление нечетких продукционных моделей показано на диаграмме классов, изображенной на рисунке 2.9.

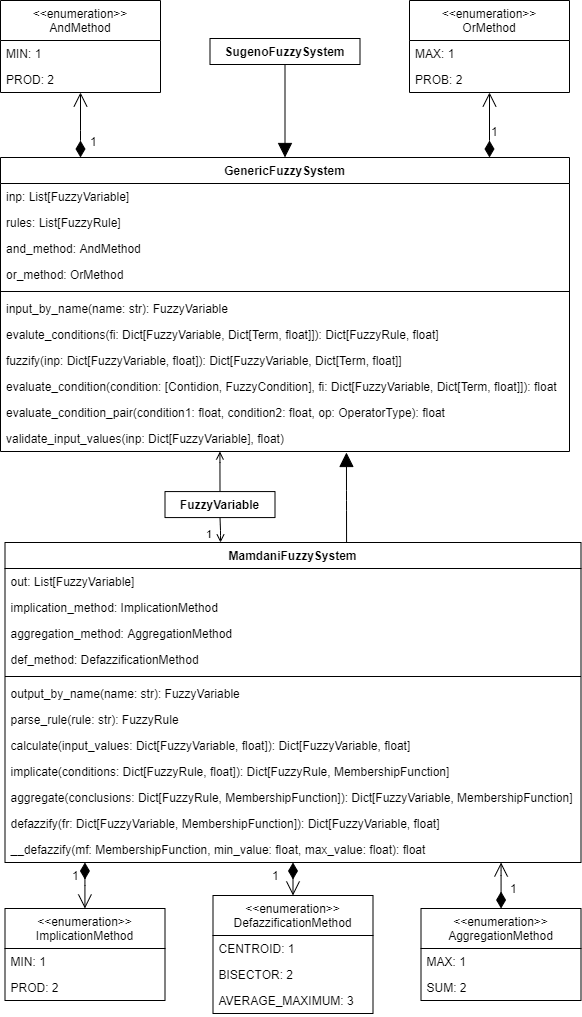


Рисунок 2.9 – Диаграмма классов обобщенного представления нечетких продукционных моделей

Обобщенный функционал нейро-нечетких моделей описан в абстрактном классе *GenericFuzzySystem*. Данный класс содержит входные значения *inp*, нечеткие продукционные правила *rules*, метод объединения *or\_mehtod* и метод пересечения *and\_method*, реализуемые классами перечислений *AndMethod* и *OrMethod*, соответственно. Среди важных методов, представленных в обобщенном абстрактном классе, разработаны следующие: *fuzzify* – фаззификация входных нечетких переменных; *evaluate\_condition* – вычисление условия нечеткого продукционного правила; *evaluate\_conditions* – расчет заключения нечеткого продукционного правила.

На основе обобщенного абстрактного метода может быть реализована любая нейро-нечеткая продукционная модель, в частности нейро-нечеткая модель на основе логического вывода Мамдани, который описан в классе *MamdaniFuzzySystem*. Кроме основных полей, расположенных в классе родителя, добавляются выходные переменные, содержащиеся в классе *FuzzyVariable*. Класс нечетких переменных содержит название лингвистической переменной и ее термы.

Ниже представлен листинг использования нейро-нечеткой продукционной модели Мамдани, реализованного с помощью класса *MamdaniFuzzySystem*, разработанного на языке программирования *python*.

На рисунке 2.10 представлена диаграмма классов нейро-нечеткой продукционной модели типа Такаги-Сугено.

Класс наследует свойства и методы обобщенного абстрактного класса *GenericFuzzySystem* и расширяет его свойством, которое содержит массив выходных переменных, описанных в классе *SugenoVariable*. Класс *SugenoVariable* содержит имя, отображенное свойством *name*, а также набор функций, которые являются массивом полиномов первого порядка, включая постоянные составляющие – константы и коэффициенты входных четких значений, реализованных с помощью класса *LinearSugenoFunction*. Он наследует функционал, описанный в обобщенной абстрактной модели *SugenoFunction*.

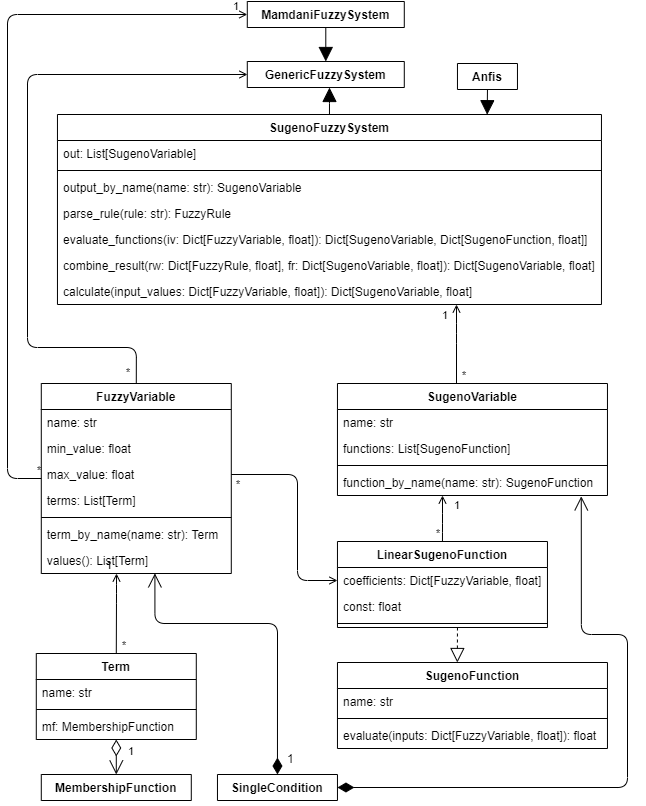


Рисунок 2.10– Диаграмма классов нейро-нечеткой продукционной модели типа Такаги-Сугено

Фрагмент использования нейро-нечеткой продукционной модели Такаги-Сугено, описанного в классе *SugenoFuzzySystem*, представлен ниже.

На рисунке 2.11 изображена диаграмма классов адаптивной нейро-нечеткой продукционной модели (*ANFIS*). Класс *Anfis* является наследником нечеткой продукционного модели *SugenoFuzzySystem*, однако расширяет его методами алгоритма автоматизированного обучения за счет содержания следующих свойств: *x* – вектор входа, *y* – вектор выхода, *rules\_text* – правила в строковых представлениях, а также параметрами горной кластеризации, описанных в классе *SubtractClustering*.

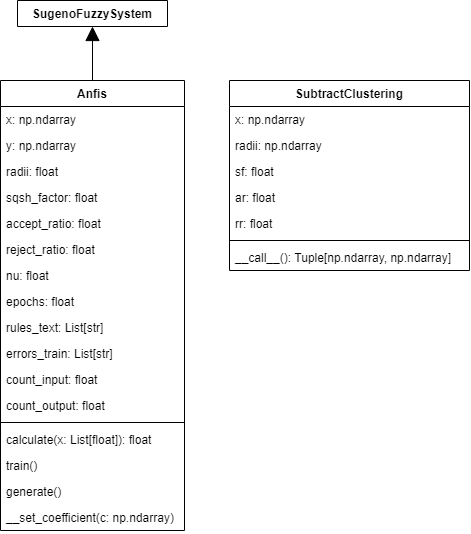


Рисунок 2.11 – Диаграмма классов адаптивной нейро-нечеткой продукционной модели (*ANFIS*)

Кроме того, класс *Anfis* содержит метод генерации нечетких продукционных правил *generate* и метод обучения *train*. Ниже представлен листинг использования адаптивной нейро-нечеткой продукционной модели *ANFIS*, реализованного с помощью класса *Anfis*.

### 2.3.2. Модуль предварительной обработки экспериментальных данных на нейросетей глубокого обучения

Имеющиеся на настоящий момент данные по совместному действия гипертермии и облучения на опухоль доказывают, что физиологический механизм воздействия этих методов на опухолевые клетки в целом известен и, приводит к следующему списку явлений, которые и используются на практике:

* бóльшая уязвимость снабжённых кислородом клеток к облучению и меньшая – к гипертермии;
* бóльшая уязвимость гипоксических (испытывающих недостаток кислорода) клеток к гипертермии и меньшая – к облучению;
* слабое возрастание кровотока в опухолевых клетках при гипертермии по сравнению с нормальными клетками (что вкупе с традиционно высоким уровнем метаболизма опухолевой ткани приводит к высокой вероятности гибели опухолевой клетки от гипоксии/ацидоза при гипертермии, как результат предыдущего пункта);
* бóльшая уязвимость клеток, находящихся в фазе деления, к облучению и химиотерапии, и меньшая – к гипертермии;
* бóльшая уязвимость клеток в интерфазе (этапе развития, сравнительно далеко отстоящем от фаз деления и гибели) к гипертермии, и меньшая – к облучению и химиотерапии;

Список можно продолжить, но уже по приведённым пунктам видно, что ОВРН представляет собой набор методов, использующих для повреждения опухолевых клеток сложное сочетание методик и факторов, имеющих, во-первых, вероятностный характер, а, во-вторых, как следствие, имеющих основную. проблему селективности воздействия гипертермии и облучения. Обработать и, особенно, выявить все факторы, влияющие на воздействие гипертермии и облучения могут только столь сложные аппроксиматоры, как глубокие нейронные сети. Они позволяют выявить иерархию признаков в десятки и сотни слоев. В отличие от признаков, созданных вручную, эти признаки не имеют прямого отношения к тому, что радиологи могут легко интерпретировать. Дополнение глубоких нейросетей информацией, уже известной как полезная, может улучшить характеристики этих моделей и их интерпретируемость

Для решения задачи прогнозирования выживаемости клеток злокачественных новообразований, дозами облучения и интенсивностью нагрева была выбрана сверточная модель глубокой нейронной сети. Схема модели представлена на рисунке 2.12

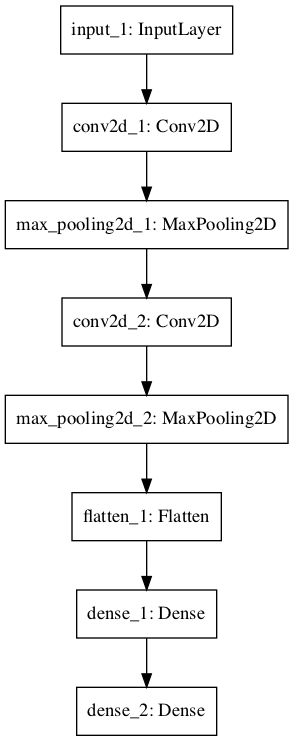


Рисунок 2.12. Модель CNN для прогнозирования

Рассмотрим программный код на языке Python 3.7, на котором реализована модель, а также параметры искусственной нейронной сети. На данном этапе происходит подключение необходимых для работы библиотек:

Далее происходит загрузка набора данных для обучения, а также его разбиение на тестовую и обучающую выборки, нормализация данных для корректной работы нейронной сети, построение модели CNN, подсчет среднеквадратичной ошибки обучения и вывод на экран параметров нейронной сети: Модель сверточной нейронной сети имеет 3 входных нейрона, 20 скрытых слоев и 1 выходной нейрон.

Для обучения нейронной сети было установлено 100 эпох. В результате экспериментального тестирования, данное количество нейрона дало наилучший результат в условиях имеющихся данных.

## 2.4. Проведение тестовых расчетов.

В качестве языка программирования для разработки программных продуктов для оценки выживаемости клеток опухоли при совместном действии гипертермии и облучения на базе нечетких, нейронных и нейро-нечетких нейронных моделей использовался язык программирования *python3+*. Разработана библиотеки программных функций для ключевых алгоритмов обработки данных ОВРН (нейро-нечетких контроллер). С помощью алгоритмов анализа данных с использованием разработанной библиотеки программных функций построены функции аппроксимации выживаемости клеток опухоли в зависимости от температуры нагрева и интенсивности облучения.

Имеющиеся на настоящий момент данные по совместному действия гипертермии и облучения на опухоль доказывают, что физиологический механизм воздействия этих методов на опухолевые клетки в целом известен и может быть успешно промоделирован.

Можно утверждать, что нечеткие системы и глубокие нейронные сети могут быть эффективным инструментом для планирования и оптимизации режимов гипертермии и облучения. Планируется продолжение исследований в области моделирования выживания клеток при гипертермии и гамма-терапии опухолей.

Рассмотрим пример использования разработанной программной библиотеки на основе исходных данных, которые отображены в таблице 2.2.

Таблица 2.2. Исходные данные для обучения адаптивной нейро-нечеткой продукционной модели *ANFIS*.

|  |  |  |  |
| --- | --- | --- | --- |
| Вектор входа | | | Вектор выхода |
| Радиация | Температура | Время | Процент выживших клеток |
| 0 | 42 | 45 | 0.53 |
| 0 | 43 | 45 | 0.18 |
| 0 | 44 | 30 | 0.021 |
| 2 | 42 | 45 | 0.26 |
| 2 | 43 | 45 | 0.051 |
| 2 | 44 | 30 | 0.008 |
| 4 | 42 | 45 | 0.26 |
| 4 | 43 | 45 | 0.051 |
| 4 | 44 | 30 | 0.008 |
| 6 | 42 | 45 | 0.022 |
| 6 | 43 | 45 | 0.012 |
| 6 | 44 | 30 | 0.0003 |
| 8 | 42 | 45 | 0.0007 |
| 8 | 43 | 45 | 0.00004 |
| 8 | 44 | 30 | 0 |

Обучающая выборка состоит из трех входных временных рядов и одного выходного вектора. В каждом одномерном ряду представлено 15 отсчетов, что является ключевым фактором, определяющим задачу как задачу аппроксимации в условиях малых выборок.

Выбор языка программирования *python*3 обусловлен наличием большого числа библиотек для анализа данных, простотой использования и широким функционалом графической визуализации. Кроссплатформенность язык позволяет выполнять исследования на любой операционной системе.

Для работы программы, написанной на языке программирования *python*, необходимо выполнить загрузку модулей .

Изначально временные ряды могут быть представлены различными источниками данных, например, базами данных, *excel*-, *csv-*файлами. На примере, показанном в листинге 6, временные ряды данных загружаются из *csv*-файла. Также показан код на языке программирования *python*, отвечающий за визуализацию данных в системе. Графическое представление обучающей выборки изображено на рисунке 2.13.

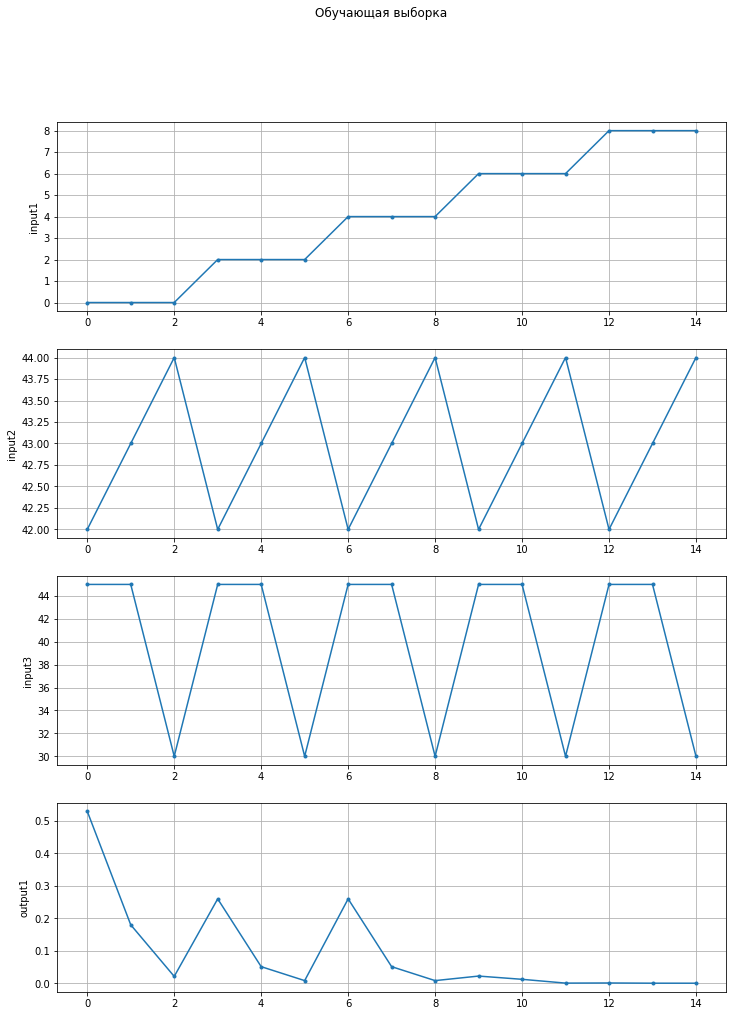


Рисунок 2.13 – Графическое представление обучающей выборки. для радиации (input 1), температуры (input 2) и времени (input 3) и доли выживших клеток

Автоматизированное обучение *ANFIS* предполагает автоматическое изъятие нечетких продукционных правил из обучающей выборки. В текущей версии разработанной программной библиотеки это выполнено на основе горной кластеризации. Особенностью горной кластеризации является то, что не требуется изначальной установки количества кластеров. Регулируемыми параметрами горной кластеризации являются:

* *radii* – допустимый радиус кластеров;
* *sqshFactor* – коэффициент подавления кластеров;
* *acceptRatio* – коэффициент принятия кластеров;
* *rejectRatio* – коэффициент отторжения кластеров.

Горная кластеризация, или кластеризация без задания количества кластеров, не является нечеткой, однако, часто используется при генерации лингвистических термов нечетких переменных и нечетких правил. Алгоритм «горной» кластеризации выполняется в несколько этапов.

На первом этапе алгоритма «горной» кластеризации происходит поиск максимальных значений элементов на каждом кластере. По умолчанию этот элемент выбирается как первый найденный кластер.

На втором этапе осуществляется нормализация данных, т.е. приведение всех значений исходных данных к единичному гиперкубу.

На третьем этапе формируются потенциальные центры кластеров. Их количество должны быть конечным. В качестве центров могут выступать объекты кластеризации. Потенциал кластера зависит от расстояния между потенциальным кластером и другими объектами кластеризации. В евклидовом пространстве это расстояние находится по следующей формуле:

,

где  – евклидово расстояние; *Z* – потенциальный центр кластера; *X* – центр объекта кластеризации.

В качестве центра кластера назначают точку с наивысшим потенциалом, которые в большинстве случаев окружен другими достаточно высокими пиками. В связи с этим, назначение центром следующего кластера с максимальным потенциалом среди оставшихся вершин привело бы размножению кластеров. Для поиска следующего кластера необходимо исключить его влияние на остальные вершины. Поэтому из потенциала кластеров вычитают потенциал найденного кластера.

На четвертом этапе происходит перерасчет потенциалов оставшихся кластеров.

Этапы 3, 4 происходят итеративно, пока максимальное значение кластера превышает заданный порог. После окончания формирования центров кластеров исходные данные и центры кластеров денормализуются.

UML-диаграмма деятельности алгоритма «горной» кластеризации представлена на рисунке ниже.



Рисунок – UML-диаграмма деятельности алгоритма «горной» кластеризации

Найденные центры кластеров являются нечеткими термами лингвистических переменных. Ширина лингвистических термов нечетких лингвистических переменных зависит от радиуса найденных кластеров.

Каждому найденному центру кластера соответствует одно правило тип [40]:

,

где  – лингвистический терм колоколообразной формы входной переменной;  – лингвистический терм выходной переменной.

Алгоритм горной кластеризации является частью разработанной программной библиотеки. Визуализация кластерного анализа показана на рисунке 2.14.

|  |  |
| --- | --- |
| C:\Users\Victor\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\77F0A86A.tmp  а) | C:\Users\Victor\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\7AEAC08.tmp  б) |
| C:\Users\Victor\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\31FBD756.tmp  в) | |

Рисунок 2.14 – Визуализация горной кластеризации представленной обучающей выборки: а) для *radiation* и *temperature*; б) для *radiation*  и *time*; в) для *temperature* и *time.*

Предварительно загруженные данные из *csv*-файла разбивают на входные и выходные вектора. Ниже представлен листинг разнесения обучающей выборки на вектор входа и вектор выхода с инициализацией экземпляра класса *ANFIS*, реализованной в разработанной программной библиотеке.

Формирование лингвистических термов входных переменных также выполняется на основе горной кластеризации при автоматизированном обучении *ANFIS*. Результирующие вид функций принадлежности может реализовываться различными представлениями. Наиболее часто применяемые являются колоколообразная, треугольная или трапециевидная функции принадлежности

Колоколообразная функция (обобщенная функция Гаусса) принадлежности задается следующим выражением:

,

где  – определяет центр нечеткого множества; ,  – определяет форму функции.

Треугольная функция принадлежности задается следующие выражением:



где  – определяет моду нечеткого множества; *d* – определяет носитель нечеткого множества.

Трапециевидная функция принадлежности задается следующей зависимостью:



где s – некоторый числовой параметр, принимающий произвольное значение;  
; .

Визуализация лингвистических термов нечетких входных переменных представлена на рисунке 2.15.

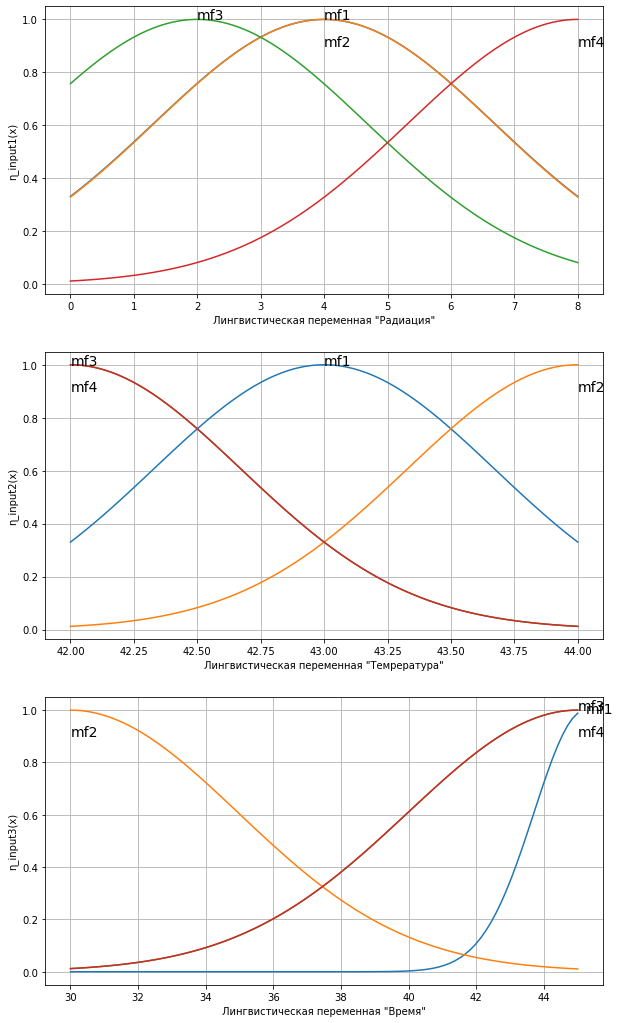


Рисунок 2.15. Функции принадлежности лингвистических термов входных переменных *ANFIS* «Радиации», «Температура» и «Время».

Проведен анализ имеющихся экспериментальных данных о совместном действии гипертермии и облучения на клетках опухоли. Обработка входных экспериментальных данных реализована в нечетком контроллере, построенном на базе разработанной библиотеки программных функций, которая реализует нейро-нечеткую адаптивную модель *ANFIS*. Нейро-нечеткая модель имеет три входа с Гауссовыми функциями принадлежности по 4 на каждую переменную. Единственный выход нейро-нечеткой модели представлен переменной типа Сугено, которая имеет 4 лингвистических терма:



где *mf*1, *mf*2, *mf*3, *mf*4 – название лингвистических термов выходной переменной «процент выживших клеток» *radiation* – интенсивность оказываемого излучения; *temperature* – температура в месте воздействия; *time* – время оказываемого излучения.

Трёх эпох обучения сети оказалось достаточно для достижения погрешности аппроксимации сложной зависимости, которая составила . На основе окончательного варианта модели адаптивной нейро-нечёткой системы, полученной в результате многократного анализа, проведено исследование интенсивности разрушения опухолевых клеток под воздействием радиационного излучения и нагрева во временной развертке.

База нечетких продукционных правил может быть построена экспертом предметной области или сгенерирована автоматизированных способом на основе алгоритма горной кластеризации. В процессе автоматизированного построения нечеткой продукционной базы правил было выявлено 4 нечетких правила.

Нечеткие продукционные правила, сформированные на основе горной кластеризации, имеют следующий вид:

***if*** (*radiation* ***is*** *mf*1) ***and*** (*temperature* ***is*** *mf*1) ***and*** (*time* ***is*** *mf*1) ***then*** (*sf* ***is*** *mf*1),

***if*** (*radiation* ***is*** *mf*2) ***and*** (*temperature* ***is*** *mf*2) ***and*** (*time* ***is*** *mf*2) ***then*** (*sf* ***is*** *mf*2),

***if*** (*radiation* ***is*** *mf*3) ***and*** (*temperature* ***is*** *mf*3) ***and*** (*time* ***is*** *mf*3) ***then*** (*sf* ***is*** *mf*3),

***if*** (*radiation* ***is*** *mf*4) ***and*** (*temperature* ***is*** *mf*4) ***and*** (*time* ***is*** *mf*4) ***then*** (*sf* ***is*** *mf*4),

где *sf* (*survival\_fraction*) – доля выживаемости опухолевых клеток.

На рисунке 2.16 показано сравнение результатов прогнозной модели с эталонными значениями обучающей выборки

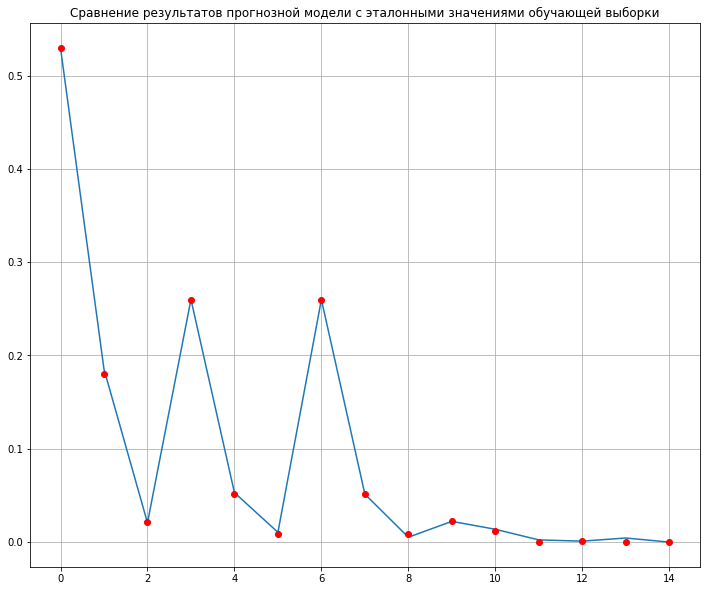


Рисунок 2.16. Сравнение результатов прогнозной модели с эталонными значениями обучающей выборки. Красными точками обозначены выходные значения обучающей выборки, а синей линией обозначены результаты аппроксимации, с использованием обученной адаптивной нейро-нечеткой продукционной модели ANFIS.

Полученный график подтверждает адекватность обученной модели, аппроксимируя зависимость в рамках обучающей выборки с достаточной степенью точности.

Автоматизированное обучение адаптивной нейро-нечеткой заключается в подаче множества обучающей выборки при создании экземпляра класса ANFIS. После инициализации нейро-нечеткой модели выполняется нормализация данных, а именно приведение данных к единичному гиперкубу с целью нивелирования разнофакторности временных рядов в составе вектора обучающей выборки. За этапом нормализации следует этап горной кластеризации, в результате которого происходит выявление нечетких знаний, и как следствие, генерация нечеткой продукционной базы правил. На основе выявленных нечетких правил происходит итерационная корректировка ядер нечетких лингвистических термов входных и выходной переменных. Результатом автоматизированного обучения адаптивной нейро-нечеткой продукционное модели является структурно-параметрически настроенная модель, способная к аппроксимации входных данных с выдержкой заданной точности. Диаграмма последовательности обучения адаптивной нейро-нечеткой продукционной модели представлена на рисунке 2.17.

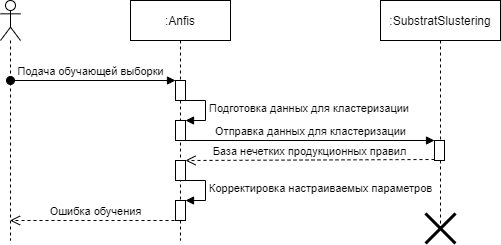


Рисунок 2.17– Диаграмма последовательности автоматизированного обучения нейро-нечеткой продукционной модели ANFIS

После автоматизированного обучения адаптивная нейро-нечеткая продукционная модель может использоваться для решения задач предметной области.

Разработанная глубокая модель искусственной нейронной сети имеет сверточную архитектуру. В данном примере модель обучалась на 100 эпохах, функция активации выбрана линейная.

Рассмотрим результаты обучения глубокой нейронной сети в таблице 2.3 Рассмотрим каждый показатель подробнее. Столбец loss показывает простое отклонение результатов, полученных нейронной сетью в процессе обучения. Столбцы MAE и MSE показывают нам среднюю абсолютную ошибку обучения и среднюю квадратичную ошибку обучения соответственно. Исходя из полученных данных обучения можно сделать вывод о том, что глубокая нейронная сеть имеет достаточно высокую точно обучения. Столбцы VAL\_LOSS, VAL\_MAE, VAL\_MSE соответственно для тех же показателей показывают ошибки валидации, то есть, проверка соответствия даны, получаемых в процессе обучения, реальному ходу явлений, для описания которых создана модель искусственной нейронной сети.

Таблица 2. Таблица ошибок обучения глубокой нейронной сети

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Epoch | loss | mae | mse | val\_loss | val\_mae | val\_mse |
| **95** | 0,002394 | 0,029637 | 0,002394 | 0,005726 | 0,072946 | 0,005726 |
| **96** | 0,002363 | 0,030244 | 0,002363 | 0,00491 | 0,0666 | 0,00491 |
| **97** | 0,002321 | 0,029459 | 0,002321 | 0,005746 | 0,073128 | 0,005746 |
| **98** | 0,002281 | 0,029907 | 0,002281 | 0,004893 | 0,066524 | 0,004893 |
| **99** | 0,002254 | 0,029472 | 0,002254 | 0,005755 | 0,073271 | 0,005755 |

Также, рассмотрим график средней абсолютной ошибки обучения глубокой нейронной сети, представленный на рисунке 2.18. Из графика видно, что среднее значение ошибки обучения на обучающей выборке в процессе обучения падает.

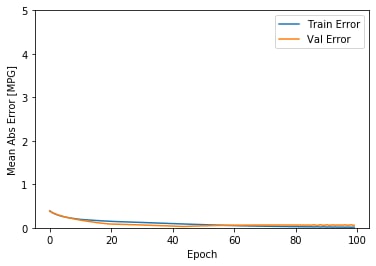


Рисунок 2.18. Средняя абсолютная ошибка обучения глубокой нейронной сети

В результате обучения мы протестировали глубокую нейронную сеть на тестовой выборке. Необходимо спрогнозировать процент выживших при облучении клеток. На рисунке 2.19 представлены результаты прогнозирования. Из результатов прогнозирования видно отклонение прогнозного значения процента выживших клеток от значения в обучающей выборке. По оси Predictions у нас располагается процент выживших клеток, полученный в результате работы нейронной сети, по оси True Values мы видим процент выживших клеток из тестовой выборки. Идеальный случай представляет собой полное совпадение этих значений (спрогнозированных и фактических) и должны располагаться на диагональной прямой y=x. Таким образом, чем меньше отклонение расчетных значений от фактических, тем ближе точки графика находятся к диагональной теоретической кривой. Столь высокое отклонение результатов прогнозирования от фактических значений обусловлен тем, что пока обучающая выборка слишком мала, чтобы получать высокое качество прогнозирования.

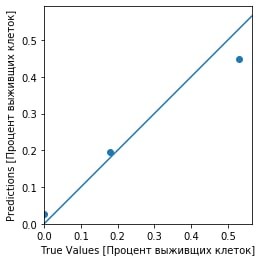


Рисунок 2.19. Результаты прогнозирования на основе Deep Learning

В таблице 2.3 представлена верификация полученных результатов прогнозирования. В данной задаче у нас имеются короткие ряды данных, так как в обучающей выборке всего 12 значений. Для коротких рядов данных результат MAPE до 20% считается приемлемым. В результате решения нашей задачи ошибка варьируется от 7 до 9 процентов, что видно из таблицы верификации прогнозов. Для подобного набора данных можно считать данный результат вполне приемлемым. 

Таблица 2.3 Верификация результатов прогноза

|  |  |  |  |
| --- | --- | --- | --- |
| Номера тестовых точек | Процент выживших клеток | Прогноз процента  выживших клеток | Квадратичное отклонение |
| 0 | 0.53 | 0.41 | 0,0512 |
| 5 | 0,18 | 0,20 | 0,0023 |
| 12 | 0,001 | 0,0015 | 0,0041 |

Глубокая искусственная нейронная сеть, в отличие от нейро-нечеткой, способна работать с миллионами строк входных данных, что позволит выявлять сотни новых признаков и зависимостей, которые, возможно, были до сих пор неизвестны врачам радиологам, от которых будет зависеть результат терапии. В том числе, как показывают результаты исследования предметной области, глубокие нейронные сети показывают превосходные результаты в анализе снимков клеток для нахождения скрытых зависимостей, незаметных человеческому глазу признаков и факторов.

На данный момент нами ведутся разработки методов объяснительного искусственного интеллекта, при помощи которого планируется построить систему объяснений и извлечения правил для глубоких нейронных сетей, в которых полученные глубокой нейронной сетью признаки и зависимости будут использоваться в правилах.

В дальнейшем возможна аппаратная реализация данных методов для встраивания в медицинскую аппаратуру облучения и нагрева клеток.

Таким образом, нами предложены два методов конструирования моделей выживания клеток при действии двух факторов поражения, основанных на экспериментальных данных, полученных для нескольких вариантов действия каждого из них в отдельности, а также в нескольких вариантах их совместного применения.

1. Первый медод использует класс нейро-нечетких сетей, изоморфный классу нечетких продукционных систем, основанный, на ЕСЛИ - ТО правилах, которые позволяют использовать лингвистическое представление информации. Правила вывода могут быть скорректированы экспериментальной обработкой данных с использованием опыта с участием специалистов-радиологов.

2. Второй метод основан на нейронных сетях глубокого обучения и использует их универсальную способность к аппроксимации функции многих переменных и нахождение иерархии свойств объекта, в том числе при работе с большим объёмом экспериментальных данных.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ В ГЛАВЕ 2

1 X.-H. Jin Neuro-fuzzy decision support system for efficient risk allocation in public-private partnership infrastructure projects // J. Comput. Civ. Eng., 24 (6) (2010), pp. 525-538, 10.1061.

2 X.-H. Jin Model for efficient risk allocation in privately financed public infrastructure projects using neuro-fuzzy techniques // J. Constr. Eng. Manag., 137 (11) (2011), pp. 1003-1014, 10.1061.

3 S. Rajab, V. Sharma A review on the applications of neuro-fuzzy systems in business// Artif. Intell. Rev., 49 (2018), pp. 481-510, 10.1007.

4 S. Mitra, Y. Hayashi Neuro-fuzzy rule generation: survey in soft computing framework// IEEE Trans. Neural Netw., 11 (3) (2000), pp. 748-768, 10.1109.

5 J. Vieira, F. Morgado-Dias, A. Mota Neuro-fuzzy systems: a survey WSEAS // Transactions on Systems, 3 (2) (2004), pp. 414-419.

6 S. Rajab, V. Sharma A review on the applications of neuro-fuzzy systems in business Artif. Intell. Rev., 49 (2018), pp. 481-510.

7 K.V. Shihabudheen, G.N. Pillai Recent advances in neuro-fuzzy system: a survey Knowl.-Based Syst., 152 (2018), pp. 136-162.

8 J. Vieira, F. Morgado-Dias, A. Mota Neuro-fuzzy systems: a survey WSEAS Transactions on Systems, 3 (2) (2004), pp. 414-419.

9 Z.J. Viharos, K.B. Kis Survey on neuro-fuzzy systems and their applications in technical diagnostics and measurement Measurement, 67 (2015), pp. 126-136.

10 Lin C. T. & Lee C. S. G., Neural Network based Fuzzy Logic Control and Decision System // IEEE Transactions on Comput. (40(12): pp. 1320-1336, 1991.

11 Jang J.-S. R. ANFIS: Adaptive-Network-Based Fuzzy Inference System // IEEE Trans. Systems & Cybernetics. – 1993. - Vol. 23. - Pp. 665 - 685.

12 H. Naderpour, M. Mirrashid Shear failure capacity prediction of concrete beam-column joints in terms of ANFIS and GMDH // Pract. Period. Struct. Des. Constr., 24 (2) (2019), Article 04019006, 10.1061.

13 K.V. Shihabudheen, G.N. Pillai Recent advances in neuro-fuzzy system: a survey // Knowl.-Based Syst., 152 (2018), pp. 136-162.

14 Bherenji H R and Khedkar P. Learning and Tuning Fuzzy Logic Controllers through Reinforcements // IEEE Transactions on Neural Networks, Vol (3), pp. 724-740, 1992.

15 Nauck D., Kruse R, Neuro-Fuzzy Systems for Function Approximation // 4th International Workshop Fuzzy-Neuro Sy stems, 1997.

16 Tano S, Oyama T., Arnould T., Deep combination of Fuzzy Inference and Neural Network in Fuzzy Inference // Fuzzy Sets and Systems, 82(2) pp. 151-160, 1996.

17 Juang Chia Feng, Lin Chin Teng, An Online Self Constructing Neural Fuzzy Inference Network and its Applications // IEEE Transactions on Fuzzy Systems, Vol 6, No.1, pp. 12-32, 1998.

18 Kasabov N. and Qun Song, Dynamic Evolving Fuzzy Neural Networks with 'm-out-of-n' Activation Nodes for On-line Adaptive Systems // Technical Report TR99/04, Department of information science, University of Otago, 1999.

19 Mackey M.C., Glass L. Oscillation and Chaos in Physiological Control Systems // Science Vol 197, pp.287-289, 1977.

20 Jang J.-S. R. ANFIS: Adaptive-Network-Based Fuzzy Inference System // IEEE Trans. Systems & Cybernetics. – 1993. - Vol. 23. - Pp. 665 - 685.

21 Mackey MC, Glass L, Oscillation and Chaos in Physiological Control Systems // Science Vol 197, pp.287-289, 1977.

## ЗАКЛЮЧЕНИЕ

Микроволновая гипертермия является одной из форм . терапии, которая значительно улучшает результаты лечения рака другими методами. Раковые опухоли вырастают из мутированных клеток и для выживания нуждаются в большем количестве энергии по сравнению с нормальными клетками. Так как тело избавляется от излишнего тепла с помощью циркуляции крови, страдающие от неправильного кровоснабжения раковые клетки становятся более чувствительными к тепловому разрушению, чем окружающие их клетки с нормальным кровоснабжением. Кроме того, повышенная температура увеличивают циркуляцию крови в опухоли, как ответ организма на тепловой стимул. Эта циркуляция увеличивает приток крови в ткани опухоли, что увеличивает эффективность ионизирующей радиации. Ионизирующая радиация разрушает с помощью кислородных радикалов ДНК раковых клеток, так как клетки, страдающие от недостатка кислорода в три раза более устойчивы к ионизирующей радиации, чем нормальные клетки.

В разделе решается задача, как с помощью нейро-нечетких сетей и сетей глубокого обучения на основе набора входных данных экспериментов по ОВРН, например доза облучения, интенсивность нагрева, и соответствующими выходными данными – процентом выживаемости клеток в экспериментах – восстановить или аппроксимировать общую функцию зависимости выходной переменной от входных переменных, с целью вычисления априорно наиболее эффективных входных параметров.

В качестве языка программирования для разработки программных продуктов для оценки выживаемости клеток опухоли при совместном действии гипертермии и облучения на базе нечетких, нейронных и нейро-нечетких нейронных моделей использовался язык программирования *python3+*. Завершена работа над реализацией библиотеки программных функций для ключевых алгоритмов обработки данных ОВРН (нейро-нечеткий контроллер). С помощью алгоритмов анализа данных с использованием разработанной библиотеки программных функций построены функции аппроксимации выживаемости клеток опухоли в зависимости от температуры нагрева и интенсивности облучения

Специфика глубоких архитектур нейронных сетей накладывает ограничения на входную выборку для обучения. А именно, для качественных результатов прогноза необходимо большое количество данных. Для решения данной проблемы необходимы дальнейшие исследования в области автоматической генерации обучающей выборки на основе имеющейся. В частности, возможно использовать генеративно-состязательные сети, технику Lean data. Но для этого необходимо проводить дополнительные исследования.