

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего
образования «Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки/специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Исследование параметрических функций активации в нейронных сетях

Обучающийся

Е.С. Мельников

(Инициалы Фамилия)

(личная подпись)

Руководитель

к. ф.-м. н. О.В. Лелонд

(ученая степень (при наличии), учёное звание (при наличии), Инициалы Фамилия)

Консультант

Е. В. Косс

(ученая степень (при наличии), учёное звание (при наличии), Инициалы Фамилия)

Тольятти 2022

Аннотация

Тема бакалаврской работы - «Исследование параметрических функций активации в нейронных сетях».

В данной бакалаврской работе исследуются параметрические функции активации, применяемые в нейронных сетях.

Структура бакалаврской работы представлена введением, двумя разделами, заключением, списком используемой литературы и используемых источников.

Во введении описывается актуальность проводимого исследования, даётся краткая характеристика выполненной работы.

В первом разделе рассматриваются основы нейронных сетей, типовые решаемые задачи, области применения, сбор данных для нейронных сетей, обучение нейронных сетей, а также функции активации.

Во втором разделе рассматриваются параметрические функции активации.

В заключении представлены выводы по проделанной работе.

В работе использовано 19 рисунков, 34 формулы, список используемой литературы и используемых источников, содержит 54 пункта.

Объём бакалаврской работы составляет 59 страниц.

Abstract

The topic of the given graduation work is «Research of parametric activation functions in neural networks»

The relevance of the topic of this work is to study the difference in practical application between the parametric and non-parametric activation functions in neural networks and to determine the influence of parameters in activation functions on accuracy and error in artificial neural networks.

The relevance of the topic of this work is to study the difference in practical application between the parametric and non-parametric activation functions in neural networks and to determine the influence of parameters in activation functions on accuracy and error in artificial neural networks.

The object of the graduation work is the comparative analysis of various types of activation functions, in which two identical models of convolutional neural networks were used, with the use of different activation functions.

In order to get more accurate results of a comparative analysis, in one model of the neural network, the non-parametric activation function of the Rectified Linear Unit (ReLU) was used, and in the other model its analogue with the trained parameter Parametric Rectified Linear Unit (PReLU) was used.

The graduation work consists of an explanatory note on 1 page, introduction, including 19 figures, 34 formulae, the list of 54 references including 5 foreign sources.

The total volume of the work is 59 pages.

Содержание

Введение.....	5
1 Теоретические основы нейронных сетей.....	8
1.1 Определение нейронной сети	8
1.2 Применение нейронных сетей.....	9
1.3 Сбор данных для нейронных сетей.....	12
1.4 Обучение нейронных сетей.....	15
1.5 Функции активации	16
1.6 Ступенчатая функция (функция пороговой активации).....	17
1.7 Сигмоидная функция.....	18
1.8 Линейная функция	20
1.9 Функция ReLU (линейный выпрямитель).....	21
2 Параметрические функции активации.....	25
2.1. PReLU.....	25
2.2. Параметрический синтез нейромоделей на основе эволюционного подхода.....	27
2.3. Настройка параметрической оптимизации для одноконтурных автоматизированных систем.....	29
2.4. Задание коэффициентов ONC для решения задач параметрической оптимизации	32
2.5. Эволюционный метод параметрического синтеза модели нейронной сети с использованием априорной информации.....	37
2.8. Параметрический синтез управления стабилизацией методом градиента скорости	49
Заключение	53
Список используемой литературы	55

Введение

Первое упоминание о нейронных сетях произошло в 1940-х годах. Полагается, что теоретические основы нейронных сетей, как научного направления, были определены в работе У. Маккаллока и У. Питса в 1943 г. В ней говорилось, что любую арифметическую или логическую функцию можно осуществить с помощью простой нейронной сети.

В 1948 - Н. Виннер вместе с единомышленниками публикует работу о кибернетике, в которой сложные биологические процессы представлялись математическими моделями.

«В 1949 г. Д. Хэбб предложил закон обучения, явившийся стартовой точкой для алгоритмов обучения искусственных нейронных сетей, а также теоремы М. Минского и исследования им ряда типовых задач, в том числе, популярной задачи «Исключающего ИЛИ»» [31].

«В 1958 г. Ф. Розенблатт предложил нейронную сеть, названную персептроном, и построил первый нейрокомпьютер «Марк-1». Персептрон был предназначен для классификации объектов и получал на этапе обучения от «учителя» сообщение, к какому классу принадлежит предъявляемый объект. Обученный персептрон был способен классифицировать объекты, в том числе, не использовавшиеся при обучении, делая при этом очень мало ошибок» [38].

В 1960 г. Уидроу совместно со своим студентом Хоффом на основе дельта-правила (формулы Уидроу) разработали Адалин, который сразу же начал использоваться для задач предсказания и адаптивного управления. В настоящее время Адалин (адаптивный сумматор) - стандартный элемент многих систем обработки сигналов.

В 1963 г. А. П. Петровым в Институте проблем передачи информации АН СССР, было проведено обширное исследование задач «сложных» для персептрона.

В 1969 г. М. Минский опубликовал не отражающее существа дела доказательство ограниченности перцептрона и показал его неспособность решать некоторые задачи (задача «чётности» и «один в блоке»), связанные с инвариантностью представлений. После данной публикации интерес к нейронным сетям упал.

«В 1972 г. Т. Кохонен и Дж. Андерсон обособлено друг от друга предложили новый тип нейронных сетей, которые были готовы функционировать в качестве памяти» [21].

«В 1973 г. Б. В. Хакимов представил нелинейную модель с синапсами на основе сплайнов и запустил её для решения задач в медицине, геологии, экологии» [21].

«В 1974 г. Пол Дж. Вербос и А. И. Галушкин независимо друг от друга изобрели алгоритм обратного распространения ошибки для обучения многослойных перцептронов» [21].

«В 1975 г. - Фукусима представляет когнитрон - самоорганизующуюся сеть, предназначенную для инвариантного распознавания образов, достигающую за счёт запоминания фактически всех состояний образа» [21].

«В 1982 г. - после некоторого периода затишья, интерес к нейронным сетям снова возрастает. Дж. Хопфилд обнаружил, что нейронная сеть с обратными связями может являть собой систему, уменьшающую энергию (так называемая сеть Хопфилда). Кохоненом были показаны модели сети, обучающейся без учителя (нейронная сеть Кохонена), решающей задачи кластеризации, визуализации данных (самоорганизующаяся карта Кохонена) и другие задачи предварительного анализа данных» [21].

«В 1985-1986 гг. теория нейронных сетей получила «технологический импульс», который был вызван возможностью моделирования нейронных сетей на появившихся в то время доступных и высокопроизводительных персональных компьютерах» [21].

В 2007г. Джеффри Хинтоном в университете Торонто созданы алгоритмы глубокого обучения многослойных нейронных сетей. Успех

обусловлен тем, что Хинтон при обучении нижних слоев сети использовал ограниченную машину Больцмана (RBM — Restricted Boltzmann Machine).

В последнее время, открыты новые возможности нейронных сетей, а работы в этой области приносят огромный вклад в промышленность, науку, технологии, экономику и медицину и т.д.

Несмотря на многообразие научных публикаций, нерешенным остается ряд вопросов, относящихся к выбору функций активации, в том числе, использование параметрических функций активации в нейронных сетях, а также обучения нейронных сетей, что показывает актуальность проводимого мною исследования.

Цель работы – рассмотреть основные теоретические основы нейронных сетей и области их применения.

Объект исследования - параметрические функции активации в нейронных сетях.

Предмет исследования - применение параметрических функций активации в нейронных сетях.

Задачи исследования:

1. исследовать теоретические основы построения нейронных сетей.
2. изучить параметрические функции активации.
3. произвести сравнительный анализ параметрических функций активации с другими функциями активации.

1 Теоретические основы нейронных сетей

1.1 Определение нейронной сети

Нейронная сеть является крайне эффективным примером машинного обучения, математическим, программным, и аппаратным её олицетворением. Нейронная сеть построена по принципу организации и функционирования биологических нейронных сетей - сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы.

Нейроны используют числа в диапазоне $[0,1]$ или $[-1,1]$. Возникает резонный вопрос, как же избавиться от чисел в этом диапазоне? На данном этапе самый простой ответ - разделить 1 на это число. Этот процесс называется нормализацией, и он очень часто используется в нейронных сетях.

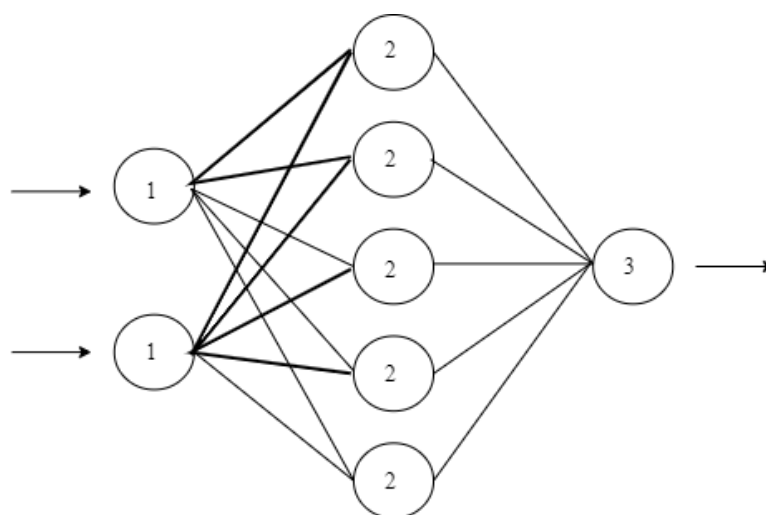
«ИНС - это система связанных и коммуницирующих между собой простых процессоров (искусственных нейронов). Подобные процессоры, как правило, довольно простые, особенно по сравнению, например, с процессором, используемым в персональном компьютере. Каждый процессор такой сети обрабатывает только сигналы, которые он периодически получает, и сигналы, которые он периодически отправляет другим процессорам. И, несмотря на это, будучи подключенным к достаточно большой сети с контролируемым взаимодействием, этот единый и простой процессор может совместно выполнять не простые задачи» [27].

«С позиции искусственного интеллекта, ИНС является одним из подходов коннекционистской философии и основным направлением в структурном методе, по исследованию возможности построения или моделирования естественного интеллекта с помощью компьютерных алгоритмов» [19].

«Также нужно затронуть как с точки зрения других моделей представляется нейронная сеть:

- в аспекте машинного обучения, нейронная сеть является частным случаем методов распознавания образов, дискриминантного анализа;
- с математической точки зрения, обучение нейронной сети представляет собой многопараметрическую задачу нелинейной оптимизации;
- с точки зрения кибернетики, нейронные сети используются для задач адаптивного управления и в качестве алгоритмов для роботов;
- в ракурсе развития вычислительной техники и программирования, нейронные сети - это способ решения проблемы «эффективного параллелизма» [74].

На рисунке 1 представлена схема простой нейронной сети.



1 - входные нейроны, 2 - скрытые нейроны, 3 - выходные нейроны

Рисунок 1 - Схема простой нейронной сети

1.2 Применение нейронных сетей

Нейронные сети применяются в тогда, когда человеческий интеллект не эффективен, а вычисления требуют больших затрат труда или не отражают реальные физические процессы и объекты.

Значимость применения нейронных сетей увеличивается тогда, когда необходимо решить плохо формализованные задачи.

«Основными областями применения нейронных сетей являются: промышленность, высокие технологии, оборона, наука и техника, здравоохранение, бизнес и финансы.

Типовые задачи, решаемые с помощью нейронных сетей:

- автоматизация процесса классификации;
- автоматизация прогнозирования;
- автоматизация процесса предсказания;
- автоматизация процесса принятия решений;
- управление;
- кодирование и декодирование информации;
- аппроксимация зависимостей и др» [36].

Ниже приведено несколько примеров таких задач:

- «прогноз фондового рынка. Учитывая цену акций на прошлой неделе и значение индекса FTSE сегодня, можно спрогнозировать цену акций на завтра» [20];
- «предоставлять кредиты. Необходимо выяснить, высок ли риск выдачи кредитов физическим лицам, которые обращаются с такими запросами. В результате общения с ним, изучив его доход и кредитную историю и т.д. это становится известно» [20];
- «руководству компании необходимо определить, что должен делать робот (поворачиваться направо или налево, двигаться вперед или назад и т.д.), для того, чтобы выполнить поставленную задачу, если есть только изображение, передаваемое видеокамерой, установленной на роботе» [20].

«Очевидно, что не каждую задачу можно решить при помощи нейронной сети. Например, спрогнозировать результаты розыгрыша, тираж которого будет через неделю, зная свой размер одежды, то вряд ли это получится, поскольку эти критерии не связаны между собой. На самом деле,

если тираж проводится честно, то нет такой информации, на основании которой можно было бы спрогнозировать результат. Многие финансовые структуры уже используют нейронные сети, или экспериментируют с ними с целью прогнозирования ситуации на фондовом рынке, и похоже, что любой тренд, прогнозируемый с помощью нейронных методов, каждый раз уже бывает "дисконтирован" рынком, и поэтому (к сожалению) эту задачу тоже вряд ли удастся решить» [20].

«Вторым условием применения нейронных сетей является понимание связи между известными исходными значениями и неизвестными значениями на выходе. Эта связь может быть искажена шумом (так, едва ли можно ожидать, что по данным из примера с предсказанием цен акций можно построить абсолютно точный предсказание, поскольку на цену влияют и другие факторы, не представленные во входных данных, и кроме того в задаче присутствует элемент случайности), но она должна существовать» [20].

«Обычно, когда неизвестен истинный тип взаимосвязи между входами и выходами, используется нейронная сеть. Если бы указанный выше тип взаимосвязи была известен, то можно было бы эту связь смоделировать» [20].

«Еще одной отличительной особенностью нейронных сетей является то, что взаимосвязь между вводом и выводом находится в процессе обучения сети» [20].

Для обучения нейронных сетей используются два типа алгоритмов (разные типы сетей используют разные типы обучения):

- управляемые ("учитесь с учителями");
- неуправляемые ("без учителей").

Обучение с учителями является наиболее часто используемым.

«Для управляемого алгоритма обучения сети пользователь должен подготовить набор обучающих данных. Такие данные являются примером ввода и соответствующего вывода. Сеть обучена устанавливать соединение между ними, как правило, обучающие данные берутся из исторической информации. В примере, рассмотренном выше, это могут быть предыдущие

значения цен на акции и индекса FTSE, знания о прошлых заемщиках - их персональные данные и то, как отреагировали они на выполнение своих обязательств» [20].

«Нейронная сеть обучается, используя тот или иной алгоритм управляемого обучения (наиболее известным из которых является метод обратного распространения, предложенный Румельхартом и др.), 1986), в котором доступные данные используются для корректировки весов и пороговых значений сети таким образом, чтобы минимизировать ошибку прогнозирования на обучающем наборе. Если сеть хорошо обучена, она получает возможность моделировать (неизвестные) функции, которые связывают значения входных и выходных переменных, а затем такую сеть можно использовать для прогнозирования, когда выходное значение неизвестно» [20].

1.3 Сбор данных для нейронных сетей

«Для точного представления принципов работы нейронных сетей, необходимо знать, как они собирают данные» [20].

«Выбор данных (по крайней мере первоначальный) осуществляется интуитивно. Личный опыт работы в данной области поможет определить, какие данные являются важными. Например, при работе с пакетом ST Neural Networks, можно случайно подбирать данные и аннулировать предыдущий выбор, кроме того, система ST Neural Networks может сама опытным путем отбирать полезные данные. Сначала есть смысл включить все данные, которые, могут оказывать влияние на результат - на следующих этапах уменьшить это множество» [20].

«Нейронные сети могут обрабатывать числовые данные в определенном ограниченном диапазоне. Это может вызвать проблемы, когда данные имеют нестандартные пропорции, в них отсутствуют значения, а данные являются нечисловыми. Пакет ST neural network содержит инструменты для решения

всех этих трудностей. Числовые данные масштабируются в диапазон, подходящий для сети. Отсутствующие значения могут быть заменены на средние значения (или другими статистическими данными) этой величины для всех доступных обучающих примеров (Bishop, 1995)» [20].

«Гораздо более сложной задачей является работа с данными нечислового характера. Чаще всего нечисловые данные бывают приведены в виде номинальных переменных, например, Пол = {Муж, Жен}. Переменные с номинальными значениями можно показать в числовом виде, и в системе ST Neural Networks имеются инструменты для работы с такими данными. Но, нейронные сети не дают хороших результатов при работе с номинальными переменными, которые могут иметь много разных значений» [20].

«Например, поставлена задача, научить нейронную сеть анализировать рынок недвижимости. Цена недвижимости во многом зависит от того, в каком районе города она расположена. Город может быть разделен на районы, каждый из этих районов, имеет своё собственное название. Можно, в данных для нейронной сети ввести переменную номинального значения для обозначения района. Но тогда, обучение нейронных сетей будет очень сложным. Напротив, лучше всего присвоить каждому району определенный рейтинг (основанный на экспертной оценке)» [20].

«Нечисловые значения других типов данных могут быть изменены на числовую форму или объявлены как несущественные. При необходимости значения даты и времени могут быть преобразованы в числовые значения путем вычитания из них начальной даты (времени). Нетрудно конвертировать название валюты в сумму. Нельзя использовать произвольные текстовые поля (например, фамилии людей), их необходимо сделать тривиальными» [20].

«Вопрос о том, сколько наблюдений требуется для обучения сети, часто является сложным. Известно много эвристических правил, которые связывают количество необходимых наблюдений с размером сети (простейшее из которых предусматривает, что количество наблюдений должно более чем в десять раз превышать количество подключений в сети). Фактически, это число

также зависит от (ранее неизвестной) сложности отображения, которое нейронная сеть стремится воспроизвести. По мере увеличения числа переменных, количество требуемых наблюдений растет нелинейно, поэтому даже при довольно небольшом (например, пятьдесят) количестве переменных может потребоваться большое количество наблюдений. Эта трудность называется "Проклятием Измерения"» [20].

«Для большинства реальных задач достаточно нескольких сотен или нескольких тысяч наблюдений. Для особо сложных задач может потребоваться большее число, но очень редко встречаются задачи (даже тривиальные), в которых достаточно менее ста наблюдений» [20].

«Если данных меньше, то для обучения сети будет недостаточно информации, тогда можно будет попытаться вписать в данные некоторые линейные модели. Пакет ST Neural Networks включает в себя инструменты для коррекции линейных моделей» [20].

«Во многих реальных задачах приходится иметь дело с данными, которые не являются полностью надежными. Значения некоторых переменных могут быть искажены или частично отсутствовать из-за шума. Программный пакет ST Neural Networks имеет специальные инструменты для обработки пропущенных значений (их можно заменить средним значением этой переменной или другими статистическими данными), поэтому, если у вас не так много данных, вы можете включить случай пропущенных значений (Кроме того, нейронные сети обычно устойчивы к шуму. Однако у этой стабильности есть свои ограничения. Например, значения выбросов, то есть значения, далекие от нормального диапазона значений переменной, могут исказить результаты обучения. В этом случае лучше всего попытаться обнаружить и удалить эти значения выбросов (путем удаления связанных наблюдений или преобразования значений выбросов в отсутствующие значения). Если выбросы трудно обнаружить, то можно использовать возможности, доступные в пакете ST Neural Networks, чтобы сделать процесс

обучения устойчивым к выбросам, но этот вид обучения устойчивости к выбросам обычно не так эффективен, как стандартный» [20].

1.4 Обучение нейронных сетей

Как правило, в качестве обучающих данных берутся данные за прошлые периоды. Если положения изменились, то закономерности, имевшие место ранее, могут не функционировать.

Нейронная сеть обучается, используя данные, которыми она обладает. Например, лица с годовым доходом более 300 000 долларов имеют высокий кредитный риск, а обучающее множество не включает лиц с годовым доходом более 30 000 долларов в год. В данной ситуации, компетентное решение от нейронной сети не может быть корректным.

«Онлайн-обучение - это самая простая вещь в освоении. Например, система машинного зрения, для автоматической идентификации танков. Нейронную сеть обучили с использованием картинок, на ста из них, содержались изображения танков, а на ста других - танков не было» [25].

Результат был 100%. Но при вводе новых данных, результат уже не был достигнут. Фотографии с танками были сделаны в ненастную погоду, а фотографии без танков были сделаны в солнечные дни. Нейронная сеть научилась обнаруживать различия в общей освещенности. Чтобы сеть работала продуктивно, данные должны быть обучены, и содержать все погодные условия, типы освещения, информацию о местности, ракурсе, расстоянии съемки и т.д.

Несогласованная база данных. Поскольку сеть сводит к минимуму общую ошибку, корреляция различных видов, представленных данных становится актуальной. Сеть, которая обработала 850 положительных и 100 отрицательных данных, деформирует результаты их обработки в пользу положительных, так как алгоритм стремится уменьшить общую ошибку. Если в совокупности положительные и отрицательные данные представлены в

разных пропорциях, то результаты, которые выдает сеть, с достаточной долей вероятности, могут быть неверными.

Например, нейронной сети поставлена задача выявления заболеваний. Во время плановых обследований в среднем 90% людей здоровы. Сеть проводит обучение на основе имеющихся данных, из которых соотношение здоровых и больных составляет в процентном соотношении 90/10. После этого, он применяется для исследования пациентов с определенными жалобами, соотношение которых уже составляет 50/50. В подобной ситуации, сеть будет ставить диагноз излишне внимательно и не распознает заболевание у определённых пациентов. Если сеть проводит обучение на "жалующихся" данных, а затем проверяет "нормальные" данные, то это приведет к увеличению числа неправильных диагнозов. В этом случае обучающие данные необходимо скорректировать с учетом различий в распределении данных (например, редкие наблюдения могут повторяться или частые наблюдения могут быть удалены), или решение, выданное сетью, может быть изменено с помощью матрицы потерь (Бишоп, 1995). Лучше всего попытаться обеспечить равномерное представление различных типов наблюдений и соответствующее объяснение результатов, полученных сетью.

1.5 Функции активации

Нейронная сеть - это математическое устройство, которое получает входные значения и преобразует их в соответствии с указанными в нем правилами. Так же нейронную сеть можно продемонстрировать, как группу нейронов, через которые проходит входной сигнал.

«Ненатуральный нейрон или искусственный нейрон представляет из себя узел искусственной нейронной сети, представляющий из себя упрощённый естественный нейрон. Математический нейрон представляется в виде функции. Если утрировать, то нейрон производит расчёт взвешенной

суммы значений на своих вводах, добавляет смещение и определяет, нужно ли исключить данное значение или нет, и использовать дальше» [40].

«При прохождении данных через сети, эти данные преобразуются в соответствии с заданным математическим алгоритмом, то есть, на входе нейрона определяется взвешенная сумма в совокупности с коэффициентом смещения, а самом нейроне полученный результат анализируется функцией активации» [40].

«Функция, в виде которой представляются нейроны, называется функцией активации. Именно функции активации определяет входное значение нейрона, в зависимости от решения взвешенной суммы входов и порогового значения» [40].

1.6 Ступенчатая функция (функция пороговой активации)

«Ступенчатая функция активации или пороговая функция активации, является такой из-за того, что для её активации требуется установить пороговое значение, то есть, если x больше этого значения, тогда считать нейрон активным в противном случае, нейрон неактивен» [40].

Для ступенчатой функции активации используется следующая математическая формула:

$$f(x) = \begin{cases} 1, & x \geq a \\ 0, & x < a \end{cases} \quad (1)$$

где a – пороговый параметр.

На рисунке 2 изображена ступенчатая функция или функция пороговой активации.

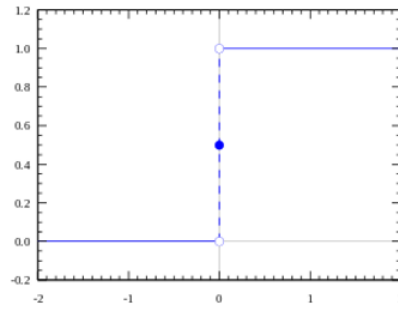


Рисунок 2 - Ступенчатая функция или функция пороговой активации.

Ступенчатая функция или функция пороговой активации, является самой простой из функций, но у нее много недочётов. В задачах классификации часто используется концепция нечеткой логики, и при исследовании, нельзя полагаться что наш пример на 100%, принадлежит к тому или иному классу, но при применении пороговых функций, на выходе возможны только два варианта решения: 0% или 100%, это не годится для задач множественной классификации, потому что, есть возможность превышения двумя нейронами пороговых значений.

«Поэтому для обучения нейронной сети пороговые функции как правило, не используются» [28].

1.7 Сигмоидная функция

Сигмоидная функция частично аналогична ступенчатой функции.

Для сигмоидной функции используется следующая математическая формула:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

На рисунке 3 представлена сигмоидная функция.

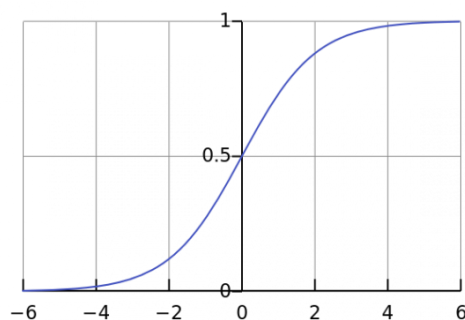


Рисунок 3 - Сигмоидная функция.

Преимущества:

- сигмоидная функция по своей сути не линейна, и комбинация таких функций также создает нелинейную функцию;
- сигмоидная функция не является бинарной, что делает активацию аналоговой, в отличие от ступенчатой функции. Также для сигмоиды характерен гладкий градиент.

«Если в диапазоне значений X от -2 до 2 значение Y меняется очень быстро, это указывает на то, что любое небольшое изменение значения X в данной области приведет к значительному изменению значения Y . Такое поведение функции указывает на то, что Y имеет тенденцию прижиматься к одному из краёв кривой» [40].

«Сигмоида кажется подходящей функцией для задач классификации. Она пытается привести значение к одной стороне кривой (например, к верхнему при $x = 2$ и нижнему при $x = -2$). Такое поведение способствует к нахождению четких границ при составлении прогнозов» [40].

«Еще одно достоинство сигмоидной функции перед линейной функцией заключается в следующем. В первом случае, имеем фиксированным диапазоном значений функции $[0,1]$, в то время, как линейная функция изменяется в пределах $(-\infty, \infty)$. Данное свойство сигмоиды очень выгодно, поскольку оно не вызывает ошибок при больших значениях активации» [40].

«Сегодня сигмоидная функция является одной из наиболее распространенных функций активации в нейронных сетях. Но у него имеются недостатки, которые заслуживают внимания» [40].

«При приближении к концам сигмоиды значения Y имеет тенденцию слабо отвечать на изменения в X . Это указывает на то, что градиент в этих областях принимает небольшие значения. Это, в свою очередь, приводит к проблемам с градиентом исчезновения. Что происходит, когда функция активации приближается к почти горизонтальной части кривой на обеих сторонах?» [40].

«В этом случае значение градиента мало или исчезает (поскольку значение очень мало, его нельзя существенно изменить). Нейронная сеть отказывается от обучения или делает это очень медленно (в зависимости от используемого метода или до тех пор, пока градиент / вычисление не начнет ограничиваться значениями с плавающей запятой). Существуют варианты решения этих проблем, и сигмоидная функция по-прежнему очень популярна для задач классификации» [40].

1.8 Линейная функция

Следующая упрощённая математическая кривая - это линейная функция. Данная линейная функция представляет собой прямую линию, которая пропорциональна входу (взвешенной сумме на этом нейроне).

Для линейной функции используется следующая математическая формула:

$$f(x) = c * x \quad (3)$$

где c - коэффициент пропорциональности.

На рисунке 4 представлена линейная функция.

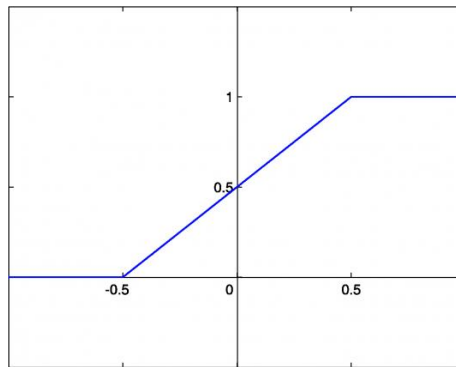


Рисунок 4 - Линейная функция

Линейная функция кажется более сложной, чем пороговая функция, но ее линейный характер разрушает всю картину обучения.

Во-первых, для обучения нейронной сети применяется метод обратного распространения ошибки. Это означает, что используется производная для функции активации. Ее производная будет равна коэффициенту пропорциональности.

«Градиент является "двигателем" весовых коэффициентов, если он постоянен и не зависит от изменений входных значений» [42].

Во-вторых, как упоминалось выше, нейронная сеть - это математическое устройство, а это значит, что ее возможно описать математическими операциями. Линейный характер функции приводит к тому, что многослойный персептрон становится эквивалентным однослойному персептрону.

1.9 Функция ReLU (линейный выпрямитель)

ReLU или линейный выпрямитель - нелинейная функция.

Любую другую функцию можно аппроксимировать комбинацией ReLU.

Функция ReLU строится по формуле:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (4)$$

На рисунке 5 представлена функция ReLU или линейный выпрямитель.

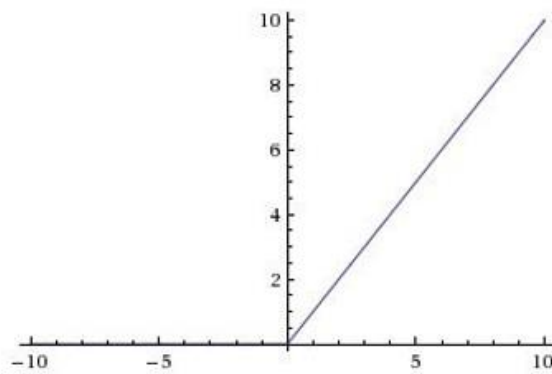


Рисунок 5 - Функция ReLU или линейный выпрямитель

ReLU возвращает значение x , если x положительно, и 0 в противном случае.

Преимущества ReLU:

- математическая форма, которая требует использование меньшего количества вычислительных ресурсов, что в свою очередь, приближает срок подготовки нейронных сетей, она также очень подходит для создания глубоких нейронных сетей;
- используя ReLU, возможно инактивировать некоторые нейроны, что сделает нейронную сеть более скоростной с точки зрения времени каждого цикла обучения.

Недостатки ReLU:

- он линеен в первом квадранте. В действительности ReLU нелинейен, и комбинация ReLU также нелинейна. Это означает, что слои можно складывать;
- проблема умирающего ReLU. Минусовое входное значение даст нулевой градиент на выходе, что окажет крайне негативное влияние на процесс обучения нейронной сети, из-за отсутствия обновления весового коэффициента. ReLU - это горизонтальная прямая (для отрицательного значения x), поэтому градиент этой части равен 0. Поскольку уклон равен нулю, вес не будет регулироваться во время

спуска. Это означает, что нейроны в этой позиции не будут реагировать на изменения ошибок / входных данных (просто потому, что градиент равен нулю, изменений нет). Это явление называется проблемой умирающего ReLU. Из-за этой проблемы некоторые нейроны просто отключатся и не будут реагировать, делая большую часть нейронной сети пассивной. Эта проблема частично решается модификацией ReLU, где для отрицательных значений входных параметров присваивается определенная линейная функция с малым масштабным коэффициентом, что позволяет избавиться от нулевого градиента и отрегулировать вес в процессе обучения.

Выводы по первому разделу:

- дано определение нейронной сети;
- представлены необходимые сведения о применении нейронных сетей, рассказано о том, какие проблемы решают нейросети;
- описаны общие сведения о сборе данных для нейронных сетей;
- дан обзор алгоритмов обучения нейронных сетей;
- раскрываются основные функции активации, а именно: ступенчатая, сигмоидная, линейная и функция ReLU или линейный выпрямитель, рассматриваются их отличительные особенности, достоинства и недостатки, также перечислены основные области применения функций активации.

Конечно, нельзя дать однозначного ответа, в каком случае, какую функцию активации нужно использовать. Но зная характеристики нейронной сети, которую необходимо аппроксимировать, нужно выбирать именно ту функцию активации, которая лучше всего аппроксимирует нейронную сеть и процесс обучения сети пройдет быстрее.

Ступенчатая функция хорошо работает для бинарной классификации, но не работает тогда, когда для классификации требуется большое количество нейронов и количество возможных классов больше двух.

Линейная функция позволяет получить диапазон значений на выходе, а не только бинарные 0 и 1, решая таким образом, проблему классификации с большим количеством кластеров.

Сигмоида показывает высокий результат при решении задач классификации, благодаря её преимуществу нормализации выходного сигнала.

ReLU используется в тех случаях, когда нет особых требований для выходного значения нейрона, но, если после обучения нейронной сети результат не оптимален, то стоит использовать другие функции активации. ReLU функционирует, как аппроксиматор, и она не так предвзята к вычислительным ресурсам, так как более проста в математических операциях.

2 Параметрические функции активации

2.1. PReLU

PReLU или параметризованный ReLU, является логическим развитием обычной функции активации ReLU. Основное различие заключается в том, что PReLU имеет модифицированные и обучаемые параметры. Также PReLU улучшает настройку модели почти без дополнительных затрат на вычислительную мощность и уменьшает риск переобучения. Получаем надежный метод инициализации, учитывая нелинейность выпрямителя. При этом данный метод позволяет обучать модели очень глубокой коррекции с нуля и изучать более глубокие или более широкие сетевые архитектуры.

PReLU определяется следующей формулой:

$$f(y_i) = \begin{cases} y_i, & \text{если } y_i > 0 \\ a_i y_i, & \text{если } y_i \leq 0, \end{cases} \quad (5)$$

где y_i - входная функция активации канала i ;
 a_i - коэффициент наклона отрицательной части.

Индекс параметра i показывает на то, что данная нелинейная активация может быть разной для разных каналов. В то же время, если a_i равно нулю, функция активации станет простым ReLU.

Указанную выше формулу можно записать следующим образом:

$$f(y_i) = \max(0, y_i) + a_i \min(0, y_i) \quad (6)$$

На рисунке 6 представлена функция PReLU, при параметре $a_i = 0,25$.

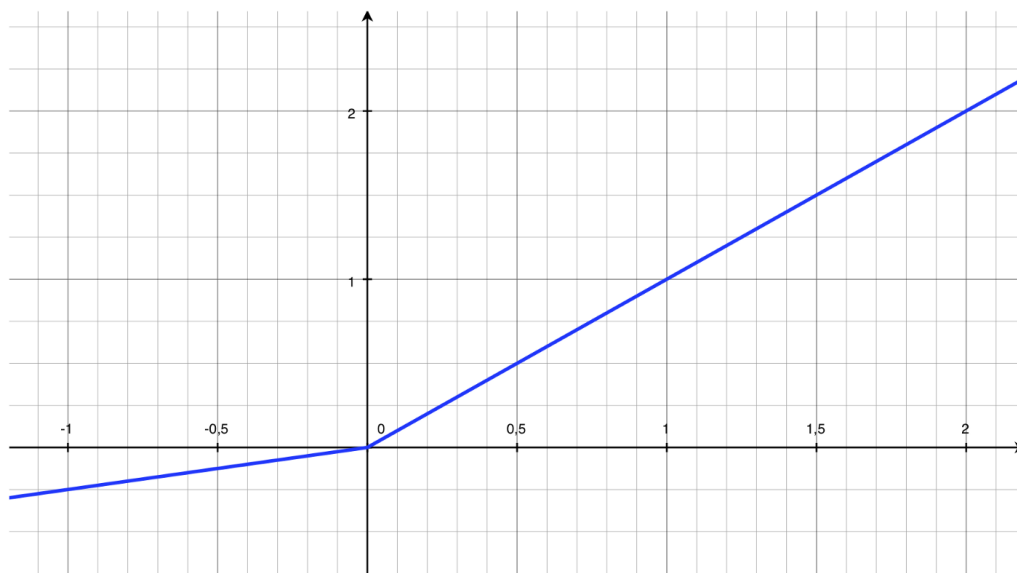


Рисунок 6 - Функция PReLU, при параметре $a_i = 0,25$

Если значение параметра a_i мало и фиксировано, например, 0,01, то PReLU становится Leaky ReLU (LReLU).

«Основная идея создания LReLU заключается в устранении нулевого градиента» [52].

Однако на самом деле, по сравнению с традиционным ReLU, увеличение точности LReLU незначительно, поэтому говорить об этом бессмысленно.

Функция PReLU очень хороша, потому что с ее добавлением количество сетевых параметров немного увеличилось: общее количество новых параметров для каждого уровня равно такому же количеству каналов в этом слое.

«PReLU может быть channel-shared. Это означает, что значение параметра a будет общим для всех каналов. В этом случае в формуле у параметра не будет индекса» [52].:

$$f(y_i) = \max(0, y_i) + a \min(0, y_i) \quad (7)$$

Кроме того, для параметрического ReLU (parametric ReLU, PReLU) угловой коэффициент на отрицательном интервале не задается заранее, а определяется на основе данных.

2.2. Параметрический синтез нейромоделей на основе эволюционного подхода

Для применения эволюционного поиска к параметрическому синтезу нейронных сетей необходимо определить целевую функцию (фитнесс-функцию) и выбрать способ представления значений весов в хромосоме.

«Как правило, среднеквадратичная ошибка используется в качестве функции приспособленности, при синтезе параметров нейронной модели» [23].

«Хромосомы в параметрическом синтезе состоят из K генов, которые содержат веса и значения смещения всех нейронов в сети. В то же время реальное кодирование используется для представления значения весового коэффициента в хромосоме» [23].

Размер хромосомы определяется по формуле:

$$K = N_i(L + 1) + \sum_{\mu=2}^M N_{\mu} + 1 \quad (8)$$

где N_{μ} - количество нейронов на μ -м слое;
 L – количество признаков в обучающей выборке;
 M – количество слоёв нейросети.

Эволюционная оптимизация значения синаптического веса нейронной сети может быть выражена в виде последовательности шагов:

«Шаг 1. Начать исходную популяцию хромосомой, которая содержит информацию о значении сетевого веса данной структуры»[3]-[6].

«Шаг2. Оценить приспособленность хромосом текущей популяции (функции приспособленности)» [7]-[12].

Шаг 2.1. Любая популяция хромосом кодируется в наборе весовых коэффициентов нейронной сети.

«Шаг 2.2. Построить нейронную сеть, соответствующую предполагаемой хромосоме»[13]-[18].

Шаг 2.3. Вычислить значение предполагаемой функции приспособленности хромосом, с учетом ошибки и сложности сети.

Шаг 3. «Проверить критерии завершения поиска (достигнуть допустимого значения ошибки модели синтетической нейронной сети, превзойти максимально допустимое количество итераций и превзойти допустимое время выполнения метода). Если хотя бы один из критериев остановки удовлетворен, переходите к шагу 7» [44].

Шаг 4. «На основе значения функции приспособленности, выбирается индивидум для создания нового решения» [23].

Шаг 5. «Использовать операторы кроссовера и мутации к хромосомам, выбранным на предыдущем шаге» [23].

Шаг 6. Для формирования нового поколения элитных хромосом и хромосом потомства, полученных с помощью скрещиваний и мутаций. Перейдите к шагу 2.

«Шаг 7. Остановить»[22].

На рисунке 7 приведено схематическое представление хромосомы при параметрическом синтезе нейромоделей.

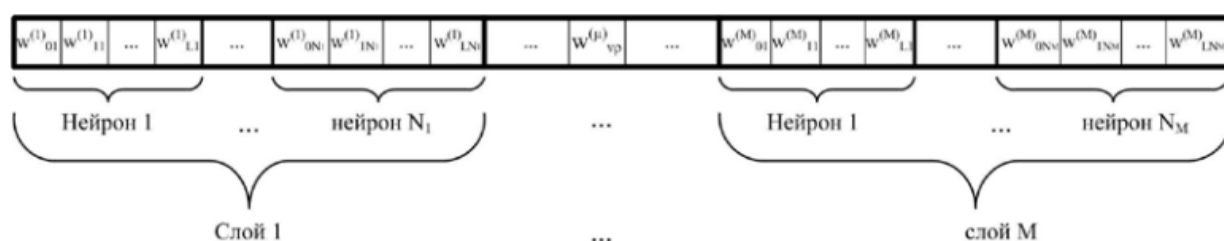


Рисунок 7 - Схематическое представление хромосомы при параметрическом синтезе нейромоделей

Таким образом, обучение нейромоделей, основанное на эволюционном подходе, не нуждается в вычислении градиента целевой функции и позволяет найти значения глобальных оптимумов синаптических весов для многомерных, полимодальных и не дифференцируемых целевых функций.

Еще одним преимуществом эволюционного поиска к параметрическому синтезу нейромоделей является возможность применения одного метода синтеза к построению различных моделей нейронных сетей (прямого распространения, рекуррентных и др.).

2.3. Настройка параметрической оптимизации для одноконтурных автоматизированных систем

«Для оптимизации параметров одноконтурной непрерывной автоматической системы, состоящей из объекта управления и ПИД-нейронного регулятора, необходимо определить такие значения в векторе настраиваемых параметров (синоптическая весовая матрица), чтобы критерий качества мог принимать экстремальные значения» [37].

Структура такой системы представлена на рисунке 8.

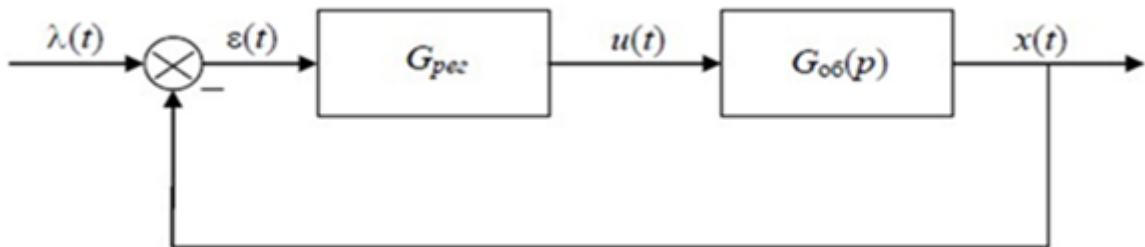


Рисунок 8 - Схема непрерывной автоматической системы, одноконтурная

«Процессы, протекающие в системе, можно описать при помощи следующих выражений:

$$\varepsilon(t) = \lambda(t) - x(t) \quad (9)$$

$$u(t) = G_{pes} \cdot \varepsilon(t) \quad (10)$$

$$x(t) = G_{ob}(p) \cdot u(t) \quad (11)$$

где $\lambda(e) = i(t)$ - задающее воздействие;
 $u(t)$ - регулирующее воздействие;
 $p = d/dt$ - оператор дифференцирования;
 $G(p, W)$ - оператор регулятора;
 $G_{об}(p)$ – оператор объекта регулирования» [37].

«Пусть оператор объекта регулирования $G_{об}(p)$ задан инерционным звеном второго порядка с запаздыванием:

$$G_{об}(p) = \frac{k}{(T_{об1}p + 1) \cdot (T_{об2}p + 1)} \cdot e^{-\tau} \quad (12)$$

где k – коэффициент передачи объекта регулирования;
 $T_{об1}$ и $T_{об2}$ – постоянные времени;
 τ – время запаздывания, причём $\frac{\tau}{T_{max}} \approx 1$, где $T_{max} = \max[T_{об1}, T_{об2}]$ » [37].

Рассмотрим работу алгоритма ONS при настройке ПИД-нейронного регулятора.

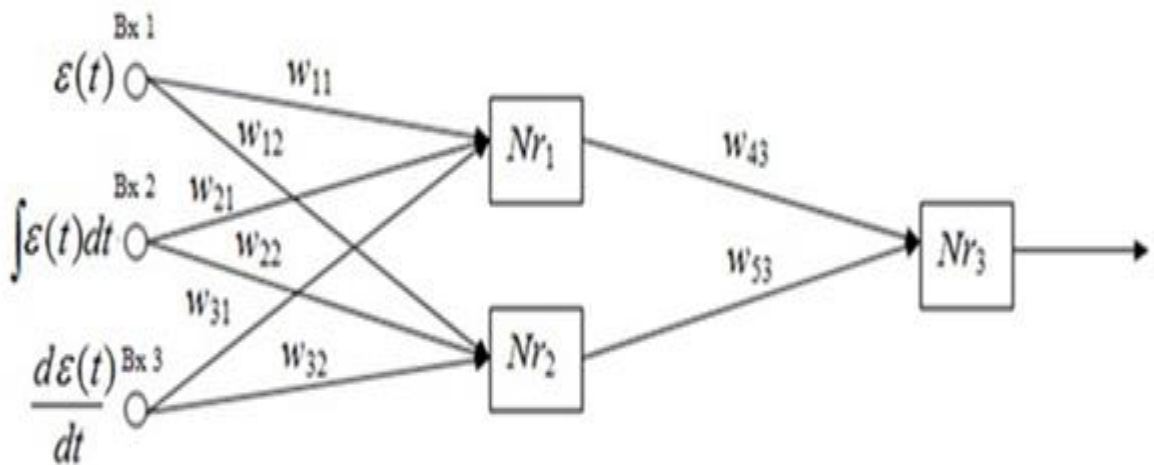


Рисунок 9 - Архитектура нейронной сети, входящей в состав ПИД-нейрорегулятора

«На рисунке 9 w_{11} в первой строке и первом столбце, показывает связь между первым входом (Vx_1) и первым нейроном (Nr_1). А элемент w_{32} показывает связь между третьим входом (Vx_3) и вторым нейроном (Nr_2)» [24].

«Из предложенных результатов этого исследования можно сделать вывод, что при использовании интегрального квадратичного критерия его минимум обеспечивается функцией активации в виде гиперболического тангенса» [24].

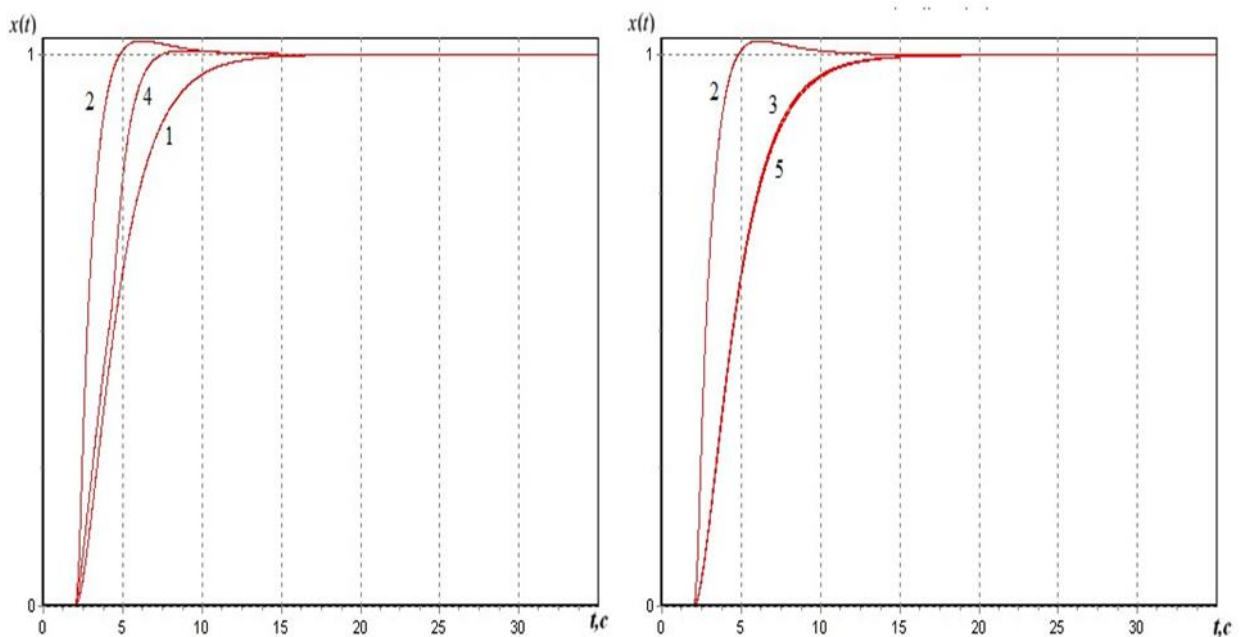


Рисунок 10 - Переходные процессы, определяющие минимальное значение

В соответствии с полученными данными исследования, при применении интегрального критерия модульности минимальное значение обеспечивается функцией ступенчатой активации.

2.4. Задание коэффициентов ONS для решения задач параметрической оптимизации

«Решим задачу конкретизации коэффициентов алгоритма ONS для определения параметрической задачи используя одноконтурную непрерывную автоматическую систему, состоящую из объекта управления и ПИД-нейронного регулятора» [32].

«Изучим структуру проводимых исследований, используя два ПИД-нейрорегулятора:

- ПИД-нейрорегулятор;
- дискретный ПИД-нейрорегулятор» [24].

«Алгоритм ONS использует эти нейрорегуляторы для настройки системы на минимизацию одного из трех критериев:

- интегральный квадратичный критерий;
- интегральный модульный критерий;
- критерий устойчивости по Ротачу» [24].

«При этом нейронная сеть применяет одну из пяти функций активации:

- логическую;
- ступенчатую;
- гиперболическую касательную;
- сигмовидную (рациональную);
- синусоидальную» [24].

«Определим, что для первого варианта нейронной сети в зависимости от количества синаптических весов, настроенных алгоритмом ONS, выбирают от 15 до 50 начальных симплексов, а для второго – от 15 до 75 начальных симплексов» [24].

«Скорее всего, для того, чтобы корректировать коэффициенты алгоритма ONS, каждая особь должна содержать четыре хромосомы» [24].

Длина каждой хромосомы на основе рекомендуемой границы:

- верхняя граница, каждого настраиваемого параметра равна $S_j = 5$;
- шаг изменений $\delta_j = 0,001$.

Тогда длина L_j каждой из хромосом равна:

$$L_j = \left\lceil \log_2 \frac{5}{0,001} \right\rceil + 1 = \lceil 12,28 \rceil + 1 = 13 \quad (13)$$

«Параметры всех генетических алгоритмов, назначенных в качестве предварительных результатов исследования, достигают следующих значений:

- количество особей в поколении $N = 50$;
- возможность скрещивания $p_c = 0,65$;
- шанс мутации $p_m = 0,25$ » [24].

На всех типах ПИД-нейронных регуляторов, ввиду сходства полученных результатов, с учетом всех функций активации нейронов и стандартов, параметры алгоритма ONS могут быть представлены в виде рисунков (рисунки 11-14 для ПИД-нейронного регулятора, 15-17 - для дискретного ПИД-нейронного регулятора).

Рисунки 11-17 показывают полученные зависимости параметров алгоритмов ONS от количества настраиваемых синаптических весов - K .

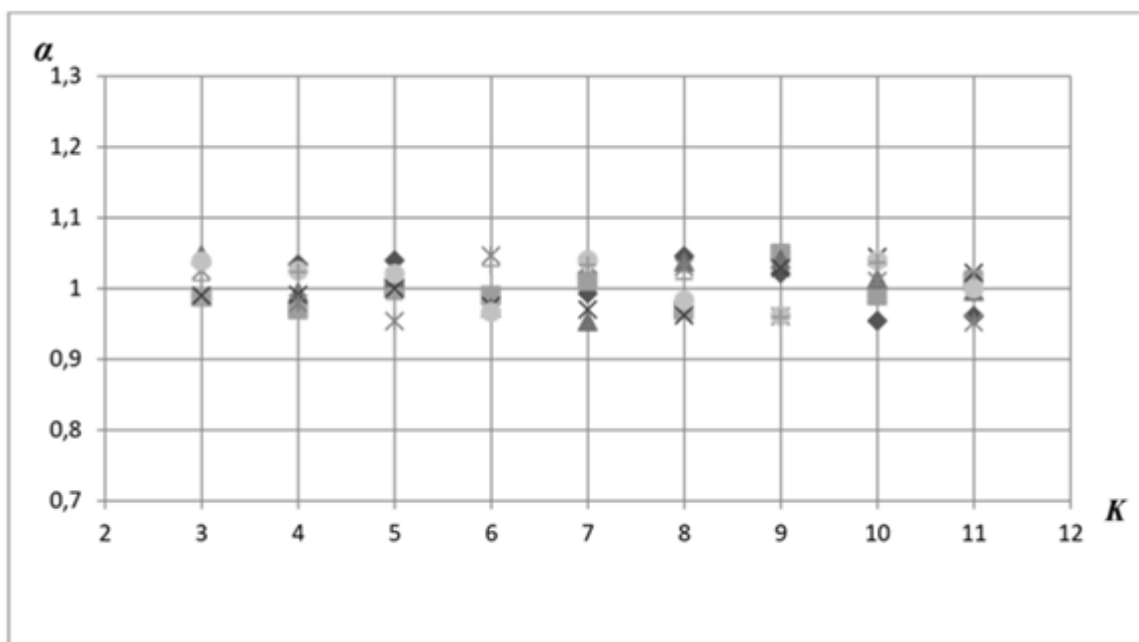


Рисунок 11 – Зависимость коэффициентов отражения от количества настраиваемых синаптических весов

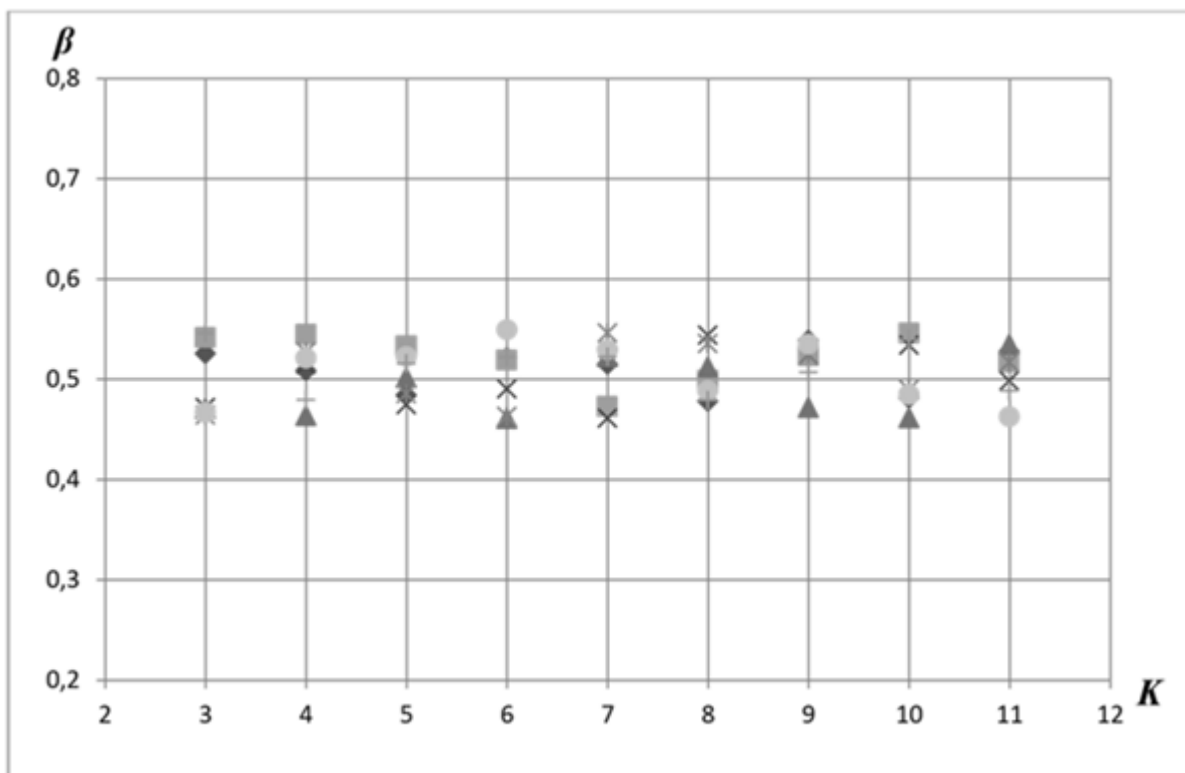


Рисунок 12 – Зависимость коэффициентов сжатия от количества настраиваемых синаптических весов

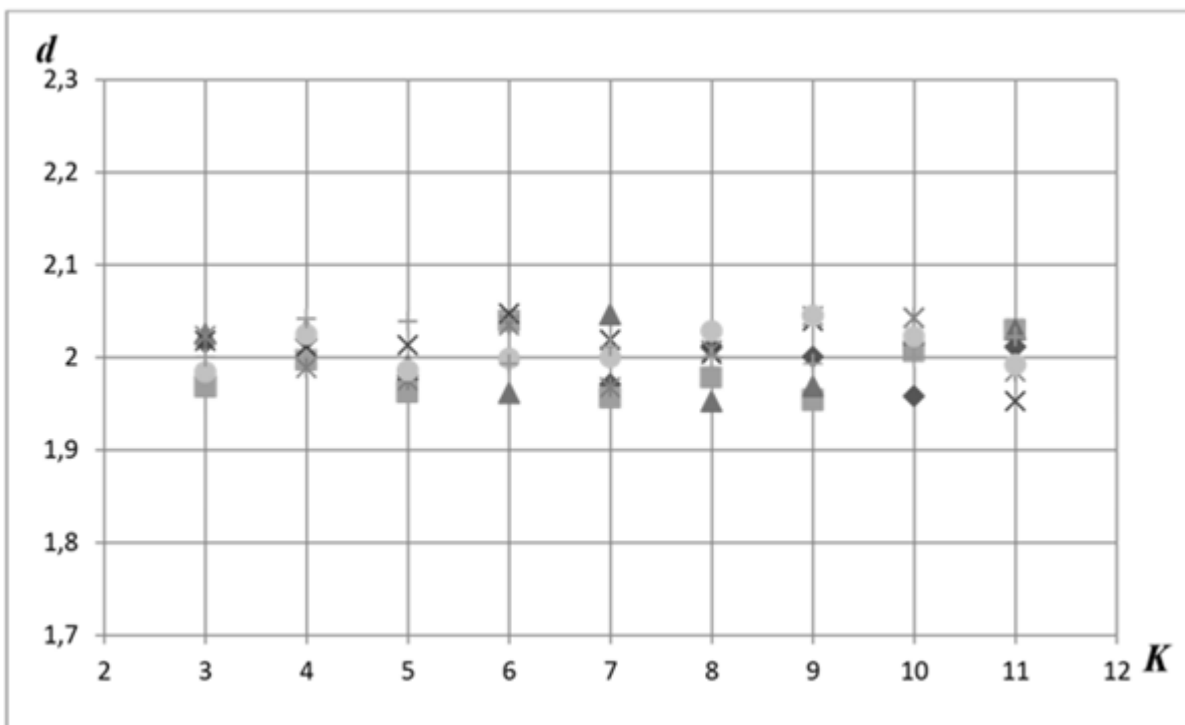


Рисунок 13 – Зависимость коэффициентов усечения от количества настраиваемых синаптических весов

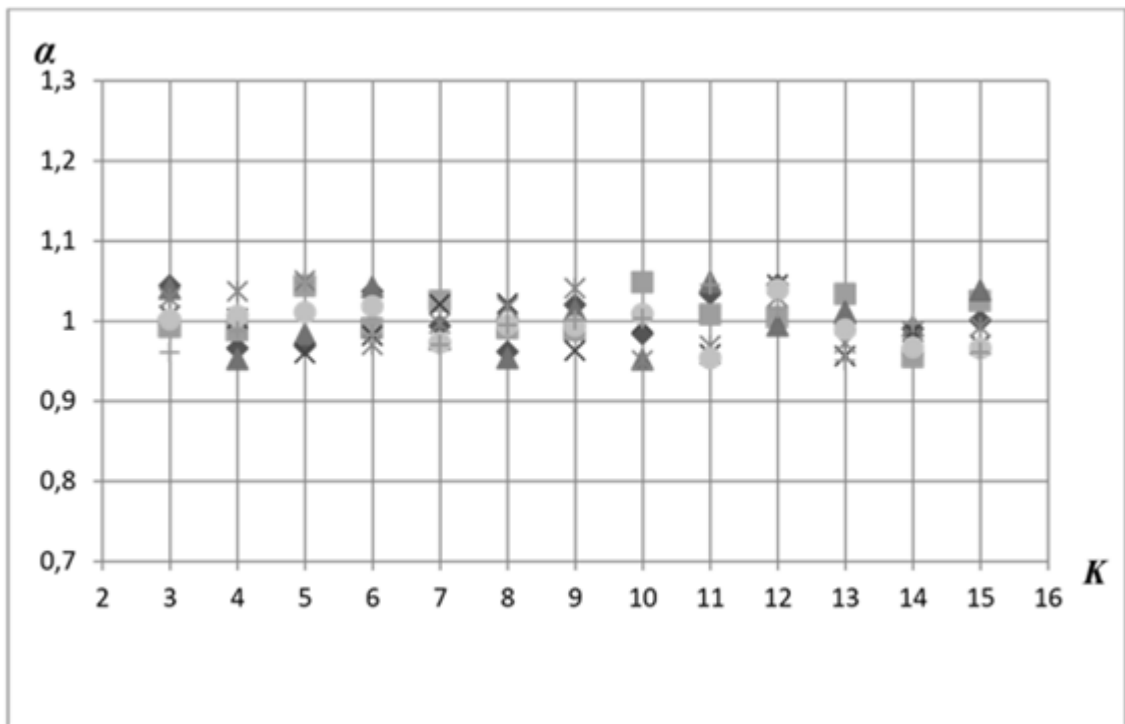


Рисунок 14 – Зависимость коэффициентов отражения от количества настраиваемых синаптических весов для дискретных ПИД-нейромодуляторов

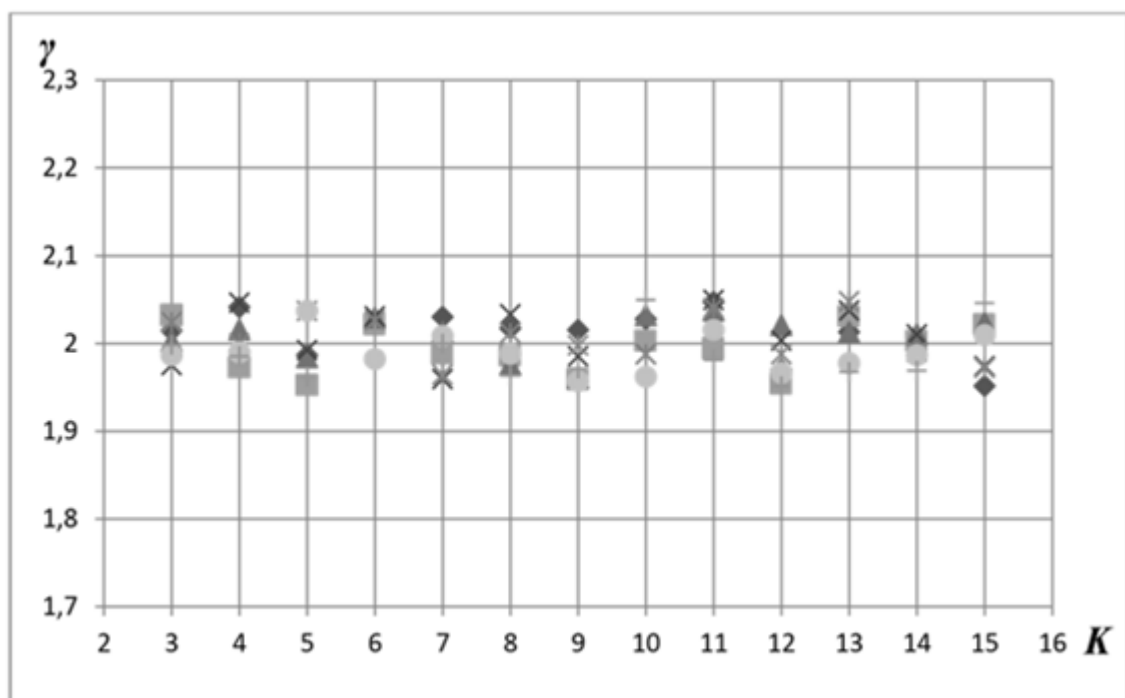


Рисунок 15 – Зависимость коэффициентов растяжения от количества настраиваемых синаптических весов

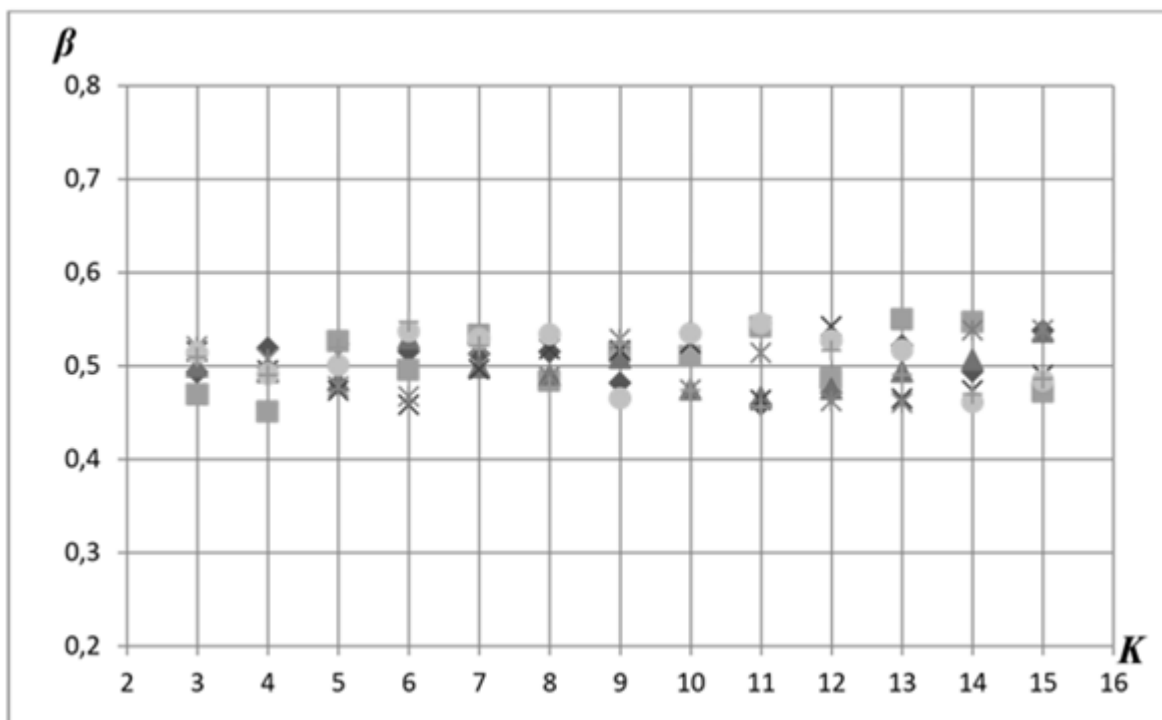


Рисунок 16 – Зависимость коэффициентов сжатия от количества настраиваемых синаптических весов

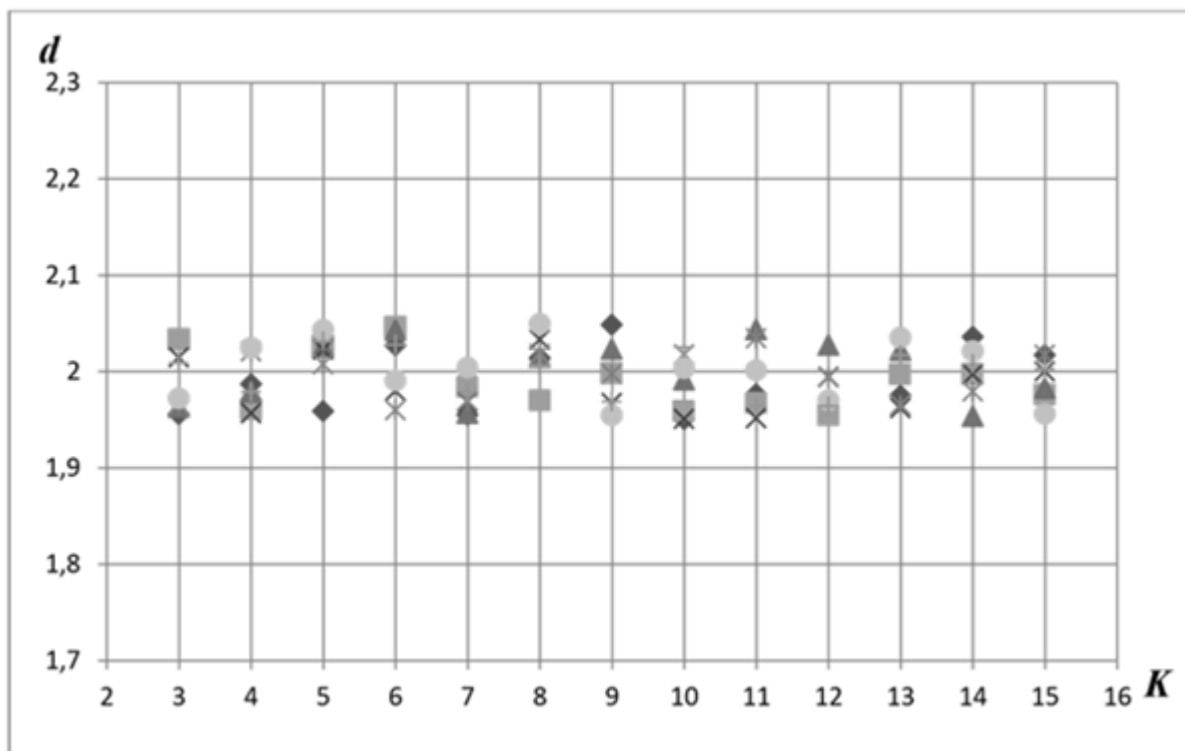


Рисунок 17 – Зависимость коэффициентов усечения от количества настраиваемых синаптических весов

где K – количество настраиваемых синаптических весов.

Результаты исследования показывают, что стандартное среднеквадратическое отклонение находится в допустимом диапазоне (0,03–0,07).

«Следовательно, в экспериментах, проведенных с достаточным диапазоном параметров, можно рекомендовать параметры алгоритма ONS при настройке ПИД-нейрорегулятора: коэффициент отражения $\alpha = 1$; коэффициент растяжения $\gamma = 2$; коэффициент сжатия $\beta = 0,05$; коэффициент усечения $d = 2$ » [24].

2.5. Эволюционный метод параметрического синтеза модели нейронной сети с использованием априорной информации

При использовании вышеуказанного метода эволюционной оптимизации хромосомы исходной популяции формируются случайным образом, а случайное создание хромосом исходной популяции во время параметрического синтеза новой модели нейрона со значениями весов и смещений, генерируемых любым способом, приводит к снижению эффективности эволюционной оптимизации в процессе поиска оптимальных значений весовой матрицы и снижению эффективности эволюционной оптимизации.

В разработанном эволюционном методе параметрического синтеза нейронных моделей обсуждается важность особенностей в операторах инициализации и мутации, используемых при эволюционном поиске.

«Создание нейронных сетей во время инициализации хромосом исходной популяции предлагается с использованием модифицированного метода Нгуена-Уидроу, который вычисляет весовой коэффициент в слое и начальное значение смещения нейрона таким образом, чтобы активные области, определяющие функцию активации каждого нейрона, были равномерно распределены по пространству входных переменных» [23].

«Активная область, которая определяет функцию активации нейрона, означает ограниченный диапазон значений входных параметров, в котором происходит значительное изменение значения функции активации» [23].

«В логистической сигмоидной функции b в качестве активной области решения предлагается интервал $[-4;4]$, и функция принимает значения в интервале $(0,018; 0,982)$, что составляет 96,4% от всего диапазона значений, а также для тангенциальной сигмоидной функции и радиальной базисной функции, интервал $[-2;2]$ указывает на то, что эти функции принимают значения в интервале $(-0,964; 0,964)$ и $(0,0183; 1]$ соответственно. Он предлагается в качестве активной области принятия решения» [23].

«Для пороговых значений и линейных функций активная область не определена, так что метод Нгуена-Уидроу для нейронов с такими функциями активации может быть использован для определения значений весовых коэффициентов и смещений» [23].

«Предлагается использовать максимальный интервал значения нервного ввода в качестве активной области определения такой функции. Однако, когда параметры модели нейронной сети инициализируются с использованием метода Нгуена-Уидроу, предполагается, что входные признаки имеют одинаковое значение. При решении реальных практических задач полезные значения характеристик обучающих выборок не совпадают, и в результате известно, что использование метода Нгена Уидроу для инициализации матрицы весовых коэффициентов нейромодели недостаточно» [23].

«Поэтому в разработанном эволюционном методе параметрического синтеза нейронных сетей было предложено инициализировать хромосомы исходной популяции с использованием модифицированного метода Нгуена-Уидроу, который учитывает априорную информацию о важности признаков во время инициализации нейронной сети» [23].

«В разработанном способе индекс индивидуальных полезных признаков также предлагается использовать для операторов точечной мутации путем

снижения вероятности мутации генов, соответствующих значению весового коэффициента связи, исходящего от входного признака, выше среднего» [23].

«Разработанный эволюционный метод параметрического синтеза моделей нейронных сетей с использованием априорной информации может быть представлен в виде следующей серии действий.

«1 этап»[25]-[26].

Устанавливается счетчик итераций (времени): $t = 0$.

2 этап.

«Для оценки индивидуальной значимости признака используется коэффициент парной корреляции, коэффициент корреляции признаков, коэффициент корреляции Фехнера, коэффициент дисперсии, коэффициент связи, информационный критерий, энтропию признака, критерий, основанный на стохастическом подходе, или критерий, основанный на вероятностном и статистическом подходе» [23].

3 этап.

Устанавливается счетчик сгенерированных хромосом: $j = 1$.

«4 этап»[29].

Генерируется j -ю хромосома, выполнив действия 4.1–4.12.

Действие 4.1.

Устанавливается счетчик слоев нейросети, соответствующей j -й хромосоме начальной популяции: $\mu = 1$.

Действие 4.2.

Вычисляется:

$$\alpha = \begin{cases} \gamma^L \sqrt{N_\mu}, & \text{если } \mu = 1; \\ \gamma^{\mu-1} \sqrt{N_\mu}, & \text{если } \mu \neq 1, \end{cases} \quad (14)$$

где N_μ - количество нейронов на μ -м слое;

L – количество признаков в обучающей выборке;

γ – коэффициент, задаваемый пользователем, $\gamma \in (0;1)$, по умолчанию предлагается устанавливать: $\gamma = 0,7$.

Действие 4.3.

Вычисляется количество входов для слоя V_μ μ -го слоя:

$$V_\mu = \begin{cases} L, & \text{если } \mu = 1; \\ N_{\mu-1}, & \text{если } \mu \neq 1, \end{cases} \quad (15)$$

Действие 4.4.

Определяется минимальное $x_{vmin}^{(\mu)}$ и максимальное $x_{vmax}^{(\mu)}$ () max значения v -го входа нейронов μ -го слоя сети:

$$x_{vmin}^{(\mu)} = \begin{cases} x_{vmin}, & \text{если } \mu = 1; \\ \psi_{vmin}^{\mu-1}, & \text{если } \mu \neq 1, \end{cases} \quad x_{vmax}^{(\mu)} = \begin{cases} x_{vmax}, & \text{если } \mu = 1; \\ \psi_{vmax}^{\mu-1}, & \text{если } \mu \neq 1, \end{cases} \quad (16)$$

где x_{vmin} и x_{vmax} – минимальное и максимальное значения v -го признака обучающей выборки:

$\psi_{vmin}^{\mu-1}$ и $\psi_{vmax}^{\mu-1}$ – минимальное и максимальное значения функции активации v -го нейрона ($\mu-1$)-го слоя.

Действие 4.5.

Определяется для каждого r -го нейрона μ -го слоя минимальное $x_{рактmin}^{(\mu)}$ и максимальное $x_{рактmax}^{(\mu)}$ значение активной области определения функции активации.

Действие 4.6.

Устанавливается счетчик нейронов в μ -м слое: $r=1$.

Действие 4.7.

Генерируется вес связи для r -го нейрона в слое M .

Действие.7.1.

Устанавливается входной счетчик r -го нейрона в μ -м слое: $v=1$.

Действие 4.7.2.

Сформируется случайное число r :

$$r = \begin{cases} rand[-I_v; I_v], & \text{если } \mu = 1; \\ rand[-1; 1], & \text{если } \mu \neq 1, \end{cases} \quad (17)$$

где I_v - значение оценки индивидуальной информативности v -го признака в обучающей выборке;

$rand[a;b]$ – случайно сгенерированное число в интервале $[a;b]$.

Действие 4.7.4.

Устанавливается: $v=v+1$.

Действие 4.7.5.

Производится проверка наличия расчета значения всех весовых коэффициентов нейронов p -го ($v > V_\mu$) μ -го слоя.

«При выполнении условия $v > V$, производится переход к действию 4.8, в противном случае осуществляется переход к действию 4.7.2» [23].

Действие 4.8.

Вычисляется значение смещения $w_{0\rho}$ μ р - это p -й нейрон μ -го слоя:

$$w_{0\rho}^{(\mu)} = \sum_{v=1}^{V_\mu} \frac{x_{vmax}^{(\mu)} + x_{vmin}^{(\mu)}}{x_{рактmax}^{(\mu)} - x_{рактmin}^{(\mu)}} w_{v\rho}^{(\mu)} + b_\rho, \quad (18)$$

$$\text{где } b_\rho = \begin{cases} \frac{1}{2} (x_{рактmax}^{(\mu)} + x_{рактmin}^{(\mu)}) + \frac{1}{2} \alpha (x_{рактmax}^{(\mu)} - x_{рактmin}^{(\mu)}) \times \\ \quad \times \left(-1 + \frac{2(\rho-1)}{N_\mu-1} \right), & \text{если } N_\mu \neq 1; \\ \frac{1}{2} (x_{рактmax}^{(\mu)} + x_{рактmin}^{(\mu)}), & \text{если } N_\mu = 1. \end{cases}$$

Действие 4.9.

Устанавливается: $\rho = \rho + 1$.

Действие 4.10.

Производится переход к действию 4.11, при вычисленном весовом коэффициенте и значении смещения в $(p > N\mu)$, для всех нейронов в μ -м слое, в противном случае, производится переход к действию 4.7.

Действие 4.11.

Увеличивается счетчик уровня нейронной сети: $\mu = \mu + 1$.

Действие 4.12.

Рассчитать весовые коэффициенты и значения смещения для всех нейронов во всех слоях нейронной сети ($\mu > M$, где M - количество нейронов в сети), то перейти к этапу 5; в противном случае перейти к действию 4.3 [5].

«Этап 5»[33].

Увеличивается счетчик, чтобы получить для генерации хромосом: $j = j + 1$.

«Этап 6»[35].

Определяется полнота формирования первой популяция ($j > N$). Если все хромосомы в исходной популяции активированы, производится переход к этапу 7, в противном случае к этапу 4.

«Этап 7»[41].

Просмотреть хромосомы текущей популяции, декодируя каждую хромосому в набор и преобразуя их в набор весовых коэффициентов нейронной сети.

«Этап 8»[43].

Просмотреть критерии для завершения поиска (достигнут допустимого значения в синтетической модели, превышены максимально допустимые итерации и превышено время, отведенное для операций метода). Если условие все еще выполняется, перейдите к этапу 13.

«Этап 9»[45]-[51].

На основе значения функции приспособленности выбираются индивидуум для создания новых решений.

«Этап 10»[53]-[54].

Использовать оператор скрещивания к выбранным на предшествующем шаге.

Этап 11.

Запустить оператор точечной мутации на выбранной хромосоме.

Результаты этого исследования основаны на формуле, h_i -хромосомная мутация i -го гена этого мутационного исследовательского представителя P_i :

$$P_i = \begin{cases} \gamma \frac{\bar{I}}{I_i}, & \text{если } h_i = w_{\nu\rho}^l; \\ \gamma, & \text{если } h_i \neq w_{\nu\rho}^l, \end{cases} \quad (19)$$

где \bar{I}_i – значение оценки индивидуальной информативности входного признака в связи, определяемой весом $w_{\nu\rho}^l$, которому соответствует ген h_i ;

$\bar{I} = \frac{1}{L} \sum_{i=1}^L I_i$ - среднее значение,

L – количество признаков в обучающей выборке;

γ – вероятность мутации генов, которым соответствуют веса связи нейронов, расположенных во 2-м и более поздних слоях нейронной сети (рекомендуется установить $\gamma=0,01K$);

K - количество генов в мутантной хромосоме).

«В результате, используя предложенный оператор мутации, вероятность мутации гена, соответствующего значению весового коэффициента от входного признака, снижается до вышеуказанного среднего значения» [23].

Этап 12.

Увеличивается счетчик итераций (время): $t=t+1$, тем самым создается будущее поколение из лучших и потомственных хромосом, составленных благодаря использования операторов скрещивания и мутации.

Этап 13. Остановка.

Данный метод эволюционной оптимизации для параметризованного синтеза нейронных сетей даёт возможность корректировать значения весов и смещений без наложения ограничений на типы функций активации нейронов.

«В отличие от классических методов эволюционного поиска для обучения нейронных сетей, разработанные методы используют специальные операторы инициализации и мутации, которые учитывают априорную информацию о важности характеристик, повышая эффективность поиска и сокращая время эволюционной оптимизации» [23].

2.6 Постановка задачи нейронного управления

«Исследуем технологический процесс (ТП), складывающийся из 1 – влияющих друг на друга технологических модулей (ТМ). ТМ обладают общеизвестной структурой и отличаются только переходами от j -го входа к i -му выходу ТП. Предположим, что каждый ТМ осуществляет некоторую функцию сложного ТП и описывается последовательным соединением линейной дифференциальной системы (линеаризованной на интервале дискретизации (наблюдения) нелинейной системы обыкновенных дифференциальных уравнений) объекта и нелинейного безынерционного преобразователя (НБП) – однослойного персептрона (искусственного нейрона). Подобные ТМ создают класс абсолютно устойчивых нелинейных систем» [34].

Уравнения j -го ТМ $j = (\overline{1, L})$ выглядят следующим образом:

$$\dot{x} = Ax + B(u + \sigma(t, z, u)) + \xi, \quad (20)$$

$$z = c^T x, \quad (21)$$

$$u = Lz \quad (22)$$

«Индекс j для лаконичности записей переменных и параметров системы уравнений (19)–(21) исключаем. В последствии считаем, что система (19) должна удовлетворять требованиям существования и единственности решения; n - мерный вектор неконтролируемых возмущений $\xi = \vec{0}$

В формулах (19)-(21) определено, что:

$x = (x_1, x_2, \dots, x_n)$ - n -мерный вектор состояния j -го ТМ, $x \in R^n$;
 u – управление j -м ТМ (измеряемый скалярный вход системы (19)), $u \in U_e \subset C^1$;
 z – наблюдение j -м ТМ (измеряемый скалярный выход системы (19)), $z \in R^1$;
 A – постоянная параметрически неопределенная матрица Якоби размерности $n \times n$;
 $B = (B_1, B_2, \dots, B_n)$ – постоянный вектор настраиваемых параметров НБП размерности $n \times 1$;
 $c = (c_1, c_2, \dots, c_n)$ – вектор-столбец задаваемых параметров наблюдения (регрессии) размерности $1 \times n$;
 L – скалярный параметрический регулятор;
 $\sigma(t, z, u)$ – скалярная функция активации (ФА) однослойного персептрона, получаемая нелинейным преобразованием j -го управления u и j -го наблюдения z .

Структура однослойной нейронной сети, параметрически адаптированной под j -й ТМ, представлена на рис. 18.

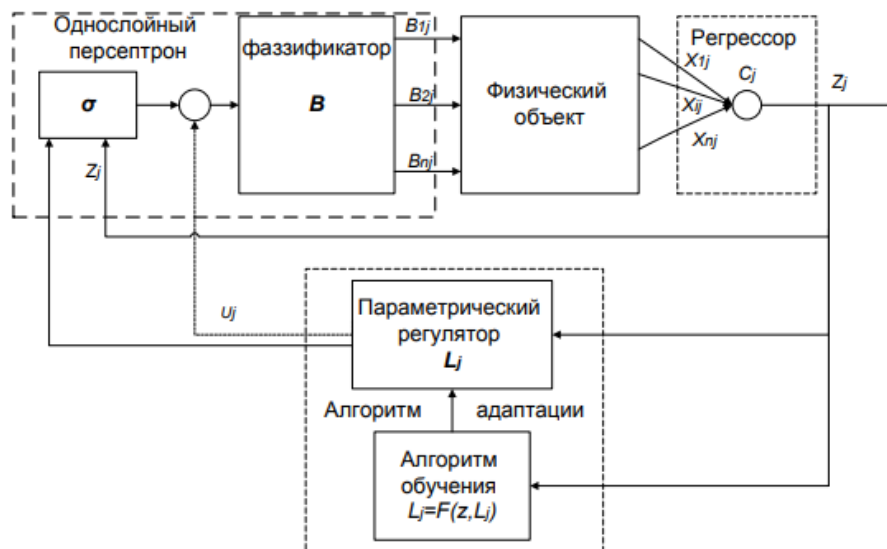


Рисунок 18 - Структура однослойной нейронной сети, параметрически адаптированной под j -й ТМ

Сплошными линиями обозначена структура однослойной НС с нелинейностью в «прямой цепи». Представленная структура отличается от классической схемы адаптивного регулятора (АР) наличием НБП и положительной жесткой обратной связи по скалярному выходу z . Данная структура НС добавлена пунктирной линией и сумматором – «цепью внутренней обратной связи» по скалярному входу u » [34].

2.7 Свойства функции активации

«ФА определяет архитектуру НС. Чётких требований по выбору ФА в настоящий момент не существует. В режиме обучения off-line более результативен алгоритм обратного распространения ошибки сигнала и следующие ФА: гиперболическая, тангенциальная, линейная и логическая сигмоидальная функции активации» [30].

«В режиме обучения on-line (оперативное обучение и управление), когда НС работает в реальном времени и осуществляет функции адаптивного регулятора, полагаем, что ФА отвечает условиям» [34].

$$0 \leq z u \sigma \leq \tilde{q}_1 \alpha z^2 + \tilde{q}_2 \beta u^2 \quad (23)$$

«где $z \in (-\infty; \infty)$, $\tilde{q}_2 \in [0,1]$, $\tilde{q}_1 = 1 - \tilde{q}_2$, $\alpha \in]0, \alpha]$, $\beta \in [0, \beta]$:
 \tilde{q}_1, \tilde{q}_2 – корректируемые на интервалах нечёткие коэффициенты, определяющие перераспределение сигналов от выхода к входу и от входа к выходу (режим работы НС);
 α, β – весовые коэффициенты соответствующей физической размерности (ед.) при переменных в части ограничений σ .

НБП $\sigma(\cdot)$ определяется правой частью выражения (22)» [1].

$$\sigma(t, z, u) = \tilde{q}_1 \alpha \frac{z}{u} + \tilde{q}_2 \beta \frac{z}{u} \quad (24)$$

Функция (23) имеет две особенные точки: $z = 0; u = 0$. На первых порах будем предполагать их изолированность [39].

В связи с тем, что наблюдение z и уравнение u – скалярные функции, то с учетом (21) НБП (23) можно запечатлеть в виде

$$\sigma(t, z, u) = \sigma(L) = \tilde{q}_1 \frac{\alpha}{L} + \tilde{q}_2 \beta L \quad (25)$$

Зависимость $\sigma(\cdot) = \sigma(L)$ при фиксированных \tilde{q}_1, \tilde{q}_2 представлена на рисунке 19.

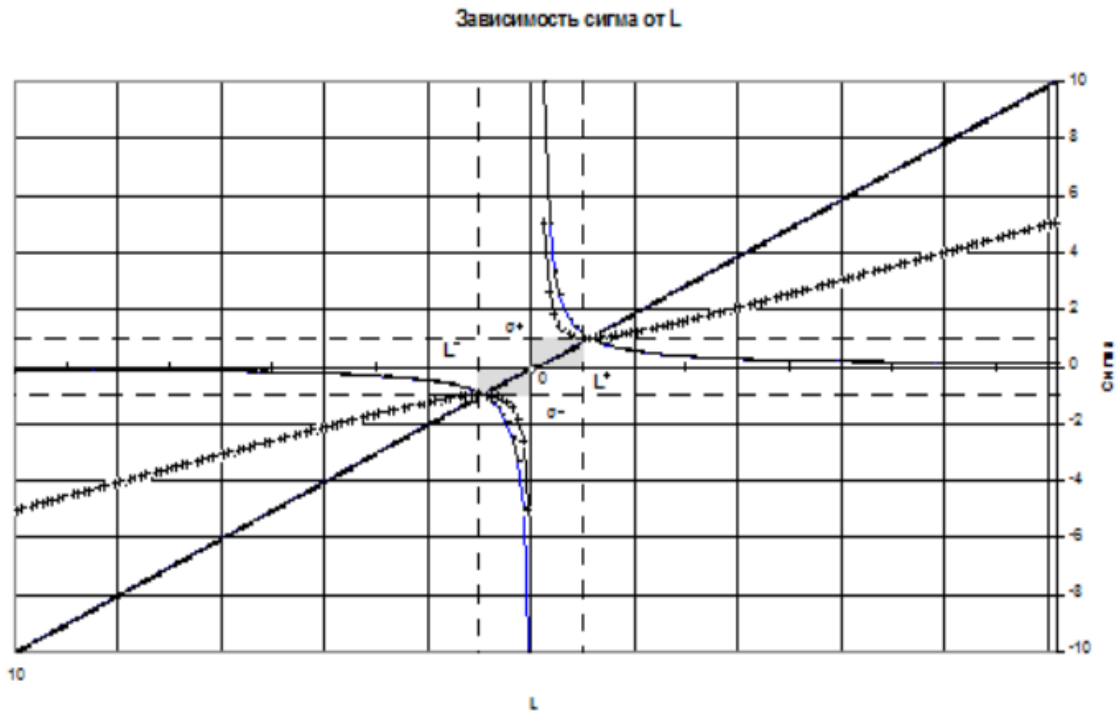


Рисунок 19 - Зависимость функции активации от значений регулируемого параметра при различных \tilde{q}_1, \tilde{q}_2 , q и $\alpha = 1, \beta = 1$ (ед.)

В свойствах функции $\sigma(\cdot)$ выражения (23) и (24) обнаруживается, что функция $\sigma(\cdot)$:

– знакоопределенная:

$\sigma(\cdot) > 0$ при $zu > 0$ ($z < 0, u < 0$ или $z > 0, u > 0$);

$\sigma(\cdot) < 0$ при $zu < 0$ ($z > 0, u < 0$ или $z < 0, u > 0$);

– содержит особенную точку $z = 0, L = 0$, в которой происходит разрыв первого рода:

$$\begin{aligned} \lim_{L \rightarrow 0^-} \sigma(L) = -\infty, \lim_{L \rightarrow 0^+} \sigma(L) = \infty, \text{ или} \\ \lim_{z \rightarrow 0^-} \sigma(t, z, u) = -\infty, \lim_{z \rightarrow 0^+} \sigma(t, z, u) = \infty. \end{aligned} \quad (26)$$

– насчитывает две точки экстремума: точку максимума $L = L^- = -\sqrt{\frac{\tilde{q}_1 \alpha}{\tilde{q}_2 \beta}}$ и точку минимума $L = L^+ = \sqrt{\frac{\tilde{q}_1 \alpha}{\tilde{q}_2 \beta}}$. В точках экстремума функция σ обретает следующие значения:

$$\begin{aligned} \sigma_{max} = \sigma^- = -2\sqrt{\tilde{q}_1 \tilde{q}_2 \alpha \beta}, \\ \sigma_{min} = \sigma^+ = 2\sqrt{\tilde{q}_1 \tilde{q}_2 \alpha \beta}. \end{aligned} \quad (27)$$

– носит нечётно-симметричную форму: $L\sigma > 0$ при $L \neq 0$. На самом деле, эта функция меняет знак только при условии $zu \neq 0$. При $L < 0$ ($zu < 0$) функция $\sigma(\cdot)$ определяется в III квадранте рисунка 19 и отрицательна $\sigma < 0$. При $L > 0$ ФА вычисляется в I квадранте и положительна: $\sigma > 0$. Параметр $L \neq 0$, так как $z = 0, u = 0$ особенные точки;

– при $L > L^+$ и $L > L^-$ функция $\sigma(\cdot)$ близка к линейной с коэффициентами наклона прямых $\pm \sqrt{\frac{\tilde{q}_1 \alpha}{\tilde{q}_2 \beta}}$ в I и III квадранте рисунок 19.

«При адаптивном подходе видно, что имеет место устойчивая внутренняя структура физического объекта, но неизвестны его параметры (коэффициенты матрицы А). В отношении нелинейности показаны частичные свойства характеристики $\sigma(\cdot)$. К тому же, неизвестен вектор B настраиваемых весовых коэффициентов однослойного персептрона. Необходимо осуществить оперативное управление j -м ТМ с помощью адаптивного параметрического нейрорегулятора L . В настоящем исследовании положена

параллель со схемой адаптивного управления с самонастройкой: НС настраивает параметры управления, задающие работу обычного контроллера, так, чтобы выходной сигнал j -го ТМ поддерживался как можно ближе к желаемому: $\lim_{t \rightarrow \infty} x(t) = x_* = \vec{0}$. Такого типа управление j -м ТМ именуется стабилизирующим. Вопрос синтеза адаптивного параметрического нейрорегулятора определяется в три этапа:

- рассматривается выбранный класс нелинейных систем на стабильность;
- синтезируется стабилизирующее управление u , обеспечивающее цель адаптации: $\lim_{t \rightarrow \infty} x(t) = \vec{0}$;
- по условиям устойчивости инициализируются параметры НС» [39].

2.8. Параметрический синтез управления стабилизацией методом градиента скорости

Постановка задачи синтеза стабилизирующего управления с использованием МСГ, сводится к следующему.

«Условие j -го ТМ - $x \in R^n$, управление u – скаляр, рассматривается скалярная величина $z = c^T x \in R^1 = c^t x \in R^1$. Эволюция j -го ТМ описывает дифференциальная система (1*), которая измеряет кусочно-дифференцируемая функция $\sigma(t, z, u)$. Она представляет собой, аддитивную добавку к скалярному управлению u и подвластна ограничению вида: B – постоянный, n -мерный вектор, для настройки входных параметров. Линейная зависимость управления от наблюдаемой величины выглядит следующим образом: $u = Lz$ » [2].

Матрица A и вектор B неизвестны. Вектор c , учитывающий "вклад" переменных состояния x для наблюдения z , задаётся.

Необходимо синтезировать параметрический регулятор $L = L(t)$ [2].

Задача управления состоит в том, чтобы соответствовать условиям $\lim_{t \rightarrow \infty} x(t) = \vec{0}$ - это соответствует минимизации в пределе локального функционала $J(t) = 0,5x^T Hx$, где H - положительно определенная, симметричная матрица размерами $n \times n$.

Воспользуемся схемой МСГ в дифференциальной форме для синтеза нелинейного параметрического регулятора:

$$\sigma(t, z, u) = \sigma(L) = \tilde{q}_1 \frac{\alpha}{L} + \tilde{q}_2 \frac{\beta}{L} \quad (28)$$

С этой целью вычислим полную производную от локального функционала:

$$\begin{aligned} \dot{J}(t) = 0,5(x^T H \dot{x} \\ + \dot{x}^T Hx) = x^T H \left[(A + LBc^T)x + B\left(\frac{\tilde{q}_1 \alpha}{L} + \tilde{q}_2 \beta L\right) \right] \end{aligned} \quad (29)$$

Определим градиент по параметру L :

$$\nabla_L \varphi(x, L) = x^T HB \left(z - \frac{\tilde{q}_1 \alpha}{L^2} + \tilde{q}_2 \beta \right) \quad (30)$$

Согласно схеме МСГ, нелинейный скалярный регулятор, синтезированный в дифференциальной форме, имеет вид:

$$\dot{L} = -\gamma \nabla_L \varphi(x, L) \quad (31)$$

Для системы (19), (20), (21) при ограничениях на ФА типа (22) алгоритм адаптации (обучения) для j -го ТМ, записывается так:

$$\dot{L} = -\gamma_L x^T HB \left(z - \frac{\tilde{q}_1 \alpha}{L^2} + \tilde{q}_2 \beta \right) \quad (32)$$

где $\gamma > 0$ - положительное число, оно определяет скорость убывания градиента по параметру L .

С учётом того, что по постановке задачи, процесс управления должен зависеть только от наблюдаемого значения z рисунок 18. Следовательно, для соблюдения равенства:

$$NB = c \quad (33)$$

Учитывая, что $c^T = B^T N$, в конечном итоге получим алгоритм адаптации j -го ТМ:

$$\dot{L} = \gamma \left(z - \frac{\tilde{q}_1 \alpha}{L^2} + \tilde{q}_2 \beta \right) z, \quad (34)$$

где $\tilde{q}_2 \in [0, 1]$, $\tilde{q}_1 = 1 - \tilde{q}_2$ – параметры, определяющие режим работы НС.

Исходя из выше сказанного, на основе метода градиента скорости, разработан эффективный алгоритм адаптации, который позволяет:

- форсировать сходимость нейро-нечёткого управления за счёт обучения нейронной сети в настоящем времени;
- использовать гибридные сети, в которых искусственные нейронные сети объединяются со структурами адаптивного управления на основе классических технологий;
- выполнить предварительное обучение и осуществить результативные процедуры инициализации параметров нейронной сети.

В отличие от классического адаптивного подхода, когда область устойчивых решений линейной части системы (19) определяется условием $L < 0$, для организации нейроподобных структур областью допустимых значений параметра L регулятора выступает всё множество действительных чисел $L \in R^1$.

Выводы по второму разделу

В данном разделе определение PReLU или параметрической функции.

Представлены необходимые сведения о применении параметрической функции.

Представлен обзор существующих методов и методик обучения параметрических функций.

С общей точки зрения рассмотрен задача параметрической оптимизации автоматических систем с использованием INS, и благодаря данной постановке достигается базовая концепция применения широкого класса алгоритмов, на основе метода Нелдера-Мида.

Проводится краткое изложение процесса формирования первоначальных симплексов. Так же дана структурная схема ОНС

Параметрическая функция активации на крайне эффективные, параметрическая функция позволяет более эффективно обрабатывать полученные данные, так же те же самые данные в неё можно поместить гораздо больше, из-за того, что количество сетевых параметров увеличивается в двое. Но при этом нагрузка на систему увеличивается незначительно. Так же из названия можно понять, что PReLU имеет обучаемые и модифицированные параметры.

Кроме того, для параметрического ReLU (parametric ReLU, PReLU) угловой коэффициент на отрицательном интервале не задается заранее, а определяется на основе данных

Во втором разделе, были приведены графики, которые показывают, как параметрическая функция активации себя введёт при вводе разных параметров. Так же были приведены математические формулы, которые показывают разные методы параметрической функции активации и расчёты с помощью их.

Заключение

В бакалаврской работе получены теоретические результаты, позволяющие выяснить основные преимущества и недостатки параметрической системы активации перед другими системами активаций.

Хотел бы добавить, что в последнее время нейронные сети развиваются с ошеломляющей скоростью, и также за ними и функции активации которых с каждым годом становится всё больше и больше, и одна из последних как раз исследуемая функция PReLU.

Подход к изучению нейронных сетей, а также методов как они получают данные, обрабатывают их для обучения показал, что с развитием данного направления, увеличивается и спектр, их применения, от обычного обрабатывания фотографии или построения простейшего математического графика, до более масштабного, как воссоздание с минимальными данными структур или сложных вычислений таких, как, вычисления квантовой физики.

Модели, основанные на нейронных сетях, могут быть успешно использованы для решения задач практически в любом направлении и разнообразии. В ходе исследования полученные результаты доказывают эффективность этих моделей по сравнению с ранее использовавшимися решениями.

Сравнивая на один взгляд похожих функций активации, а именно (ReLU и PReLU) показал, крайне эффективность второй, а именно, с его добавлением количество сетевых параметров немного увеличилось: общее количество новых параметров для каждого уровня и равно только количеству каналов для этого уровня, и это позволяет ей работать намного стабильнее, также данный метод позволяет обучать модели очень глубокой коррекции с нуля и изучать более глубокие или более широкие сетевые архитектуры. И ещё одно преимущество, что данный метод может обучить более глубокому, без практически дополнительных затрат на ресурс системы.

И помимо достигнутой изначальной цели исследования, были также получены новые данные о эффективности использования PReLU представлений на регуляризацию моделей. Полученные результаты могут быть использованы при выборе подхода к разработке новых функций активации или при выборе существующей модели представлений для решения задач обработки естественного языка.

Список используемой литературы

1. Адаптивное управление на нейроморфных структурах технологических модулей. [Электронный ресурс] – Режим доступа: <http://elibrary.ru>
2. Адаптивный подход к нейронному управлению одним классом абсолютно устойчивых систем. [Электронный ресурс] – Режим доступа: <http://elibrary.ru>
3. Абрамов Н.А. Ситуационный подход к управлению режимами систем тягового электроснабжения / Н.А. Абрамов, В.П. Закарюкин, А.В. Крюков // Современные технологии. Системный анализ. Моделирование. – 2010. – № 1 (25). – С. 186–191.
4. Александров А.Г. Состояние и перспективы развития адаптивных ПИДрегуляторов в технических системах / А.Г. Александров, М.В. Паленов // Труды 3-й Всерос. конф. с междунар. участием «Технические и программные средства систем управления, контроля и измерения». – М.: ИПУ РАН, 2012. – С. 1577–1587.
5. Ануфриенко С.Е. Нейронные модели на основе импульсного нейрона: учеб. пособие / С.Е. Ануфриенко, Е.В. Коновалов. – Ярославль: Ярославский университет им. Демидова, 2012. – 78 с.
6. Бесекерский, В.А. Теория автоматического управления / В.А. Бесекерский, Е.П. Попов. – 3-е изд., испр. – М.: Наука, 2007. – 768 с.
7. Булатов Ю.Н. Применение сетевых кластеров (microgrid) в системах электроснабжения железных дорог / Ю.Н. Булатов, А.В. Крюков, Хынг Чан Зюй. – Братск: БрГУ, 2015. – 173 с.
8. Булатов Ю.Н. Автоматические регуляторы для установок распределенной генерации / Ю.Н. Булатов, А. В. Крюков, Хынг Чан Зюй // Системы. Методы. Технологии. – 2014. – № 3 (23). – С. 108–116.
9. Булатов Ю.Н. Автоматические регуляторы установок распределенной генерации в системах электроснабжения железных дорог /

Ю.Н. Булатов // Транспортная инфраструктура сибирского региона. – 2017. – № 1 (66). – С. 15–25.

10. Булатов Ю.Н. Автопрогностический регулятор частоты вращения ротора генератора установки распределенной генерации / Ю.Н. Булатов, А.В. Крюков, Нгуен Ван Хуан // Научный вестник Новосибирского государственного технического университета. – 2017. – № 1(66). – С. 15–25.

11. Булатов Ю.Н. Интеллектуальная настройка регуляторов установок распределенной генерации / Ю.Н. Булатов, А.В. Крюков // Информационные и математические технологии в науке и управлении. – 2017. – № 3 (7). – С. 122–135.

12. Булатов Ю.Н. Применение генетических алгоритмов для настройки автоматических регуляторов установок распределенной генерации / Ю.Н. Булатов, А.В. Крюков // Системы. Методы. Технологии. – 2016. – № 2. – С. 30–45.

13. Булатов Ю.Н. Согласованная настройка регуляторов установок распределенной генерации, работающих в системе электроснабжения железной дороги / Ю.Н. Булатов, А.В. Крюков, Хынг Чан Зюй // Системы. Методы. Технологии. – 2015. – № 1 (25). – С. 94–102.

14. Бураков М.В. Нейронные сети и нейроконтроллеры / М.В. Бураков. – СПб.: ГУАП, 2013 – 213 с.

15. Бураков М.В. Синтез нечетких логических регуляторов / М.В. Бураков, А.С. Коновалов // Информационно-управляющие системы. – 2011. – № 1. – С. 14–19.

16. Бураков М.В. Синтез дискретного ПИД-нейрорегулятора / М.В. Бураков // Вестник Казанского технологического университета. – 2014. – № 3. – С. 286–288.

17. Бураков М.В. Нейронный супервизор для управления нелинейным объектом / М.В. Бураков, Т.Г. Полякова / Сб. докл. «Завалишинские чтения». – СПб.: ГУАП, 2010. – С. 23–27.

18. Высоцкая О.В. Формирование алгоритмов параметрической оптимизации систем с широтно-импульсной модуляцией второго рода / О.В. Высоцкая, Н.Н. Куцкий // Вестник ИрГТУ. – Иркутск, 2003. – № 5. – С. 8–13.
19. Вестник Иркутского Государственного Технического Университета. [Электронный ресурс] – Режим доступа: <http://bibliorossica.com>
20. Геометрическое моделирование и конструкторские базы данных [Электронный ресурс] – Режим доступа: <http://diplomba.ru>
21. Горбань А., Россиев Д. Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996. 276 с.
22. Горбачевская, Е. Н. Классификация нейронных сетей [Электронный ресурс] / Е. Н. Горбачевская // Вестник ВУиТ. – 2012. – № 2 (19). – Режим доступа: <https://cyberleninka.ru/article/n/klassifikatsiya-neyronnyh-setey>.
23. Единое окно [Электронный ресурс] – Режим доступа: <http://window.edu.ru>
24. Игумнов И.В. Метод Нелдера - Мида для настройки регулятора, функционирующего на основе искусственных нейросетей [Электронный ресурс] – Режим доступа: <http://dlib.rsl.ru/>.
25. Информационные технологии в экономике [Электронный ресурс] – Режим доступа: <https://book.ru>
26. Искусственные нейронные сети [Электронный ресурс] – Режим доступа: <https://bestreferat.ru>
27. Искусственная нейронная сеть. [Электронный ресурс] – Режим доступа: <http://ru.wikipedia.org>
28. Искусственные нейронные сети для распознавания графических образов. [Электронный ресурс] – Режим доступа: <http://elibrary.ru>
29. Крылов А.А., Сизых В.Н., Чумак А.Г. Методика структурно-параметрического синтеза нейросетевой модели продольного движения воздушного транспортного средства // Современные технологии. Системный анализ. Моделирование. 2011. № 1 (29). С. 129–134.

30. Крылов А.А., Сизых В.Н., Чумак А.Г. Методика структурно-параметрического синтеза нейросетевой модели продольного движения воздушного транспортного средства // Современные технологии. Системный анализ. Моделирование. 2011. № 1 (29). С. 129–134.
31. Минский М., Пейперт С. Перцептроны. М.: Мир, 1971.
32. Метод Нелдера-Мида для настройки регуляторов, функционирующих на основе нейронных сетей [Электронный ресурс] – Режим доступа: <http://arhvak.minobrnauki.gov.ru>
33. Модель математической нейронной сети. [Электронный ресурс] – Режим доступа: <http://elibrary.ru>
34. Мухопад Ю.Ф., Пашков Н.Н., Сизых В.Н. Адаптивный подход к нейронному управлению одним классом абсолютно устойчивых систем. М.: Изд-во РАЕ, 2011. №8. Ч. 1. С. 139–147.
35. Нейронные сети - Plants and People [Электронный ресурс] – Режим доступа: <https://sites.google.com>
36. Нейросетевые технологии. [Электронный ресурс] – Режим доступа: <http://elibrary.ru>
37. Параметрическая оптимизация автоматических систем стабилизации с помощью генетического алгоритма (2/4) [Электронный ресурс] – Режим доступа: <http://irgups.ru>
38. Розенблатт Ф. Принципы нейродинамики. Перцептроны и теория механизмов мозга. М.: Мир, 1965.
39. Срагович В.Г. Адаптивное управление. М.: Наука, 1981. 384 с.
40. Функции активации нейросети: сигмоида, линейная, ступенчатая, relu, tahn. [Электронный ресурс] – Режим доступа: <http://elibrary.ru>
41. Функции активации: ступенчатая, линейная, ReLu, Tahn, сигмоида [Электронный ресурс] – Режим доступа: <https://neurohive.io>
42. Функции активации в искусственных нейронных сетях [Электронный ресурс] – Режим доступа: <https://aisimple.ru>

43. Функции активации в искусственных нейронных сетях [Электронный ресурс] – Режим доступа: <https://aisimple.ru>
44. Эволюционный метод структурно-параметрического синтеза нейронных сетей. [Электронный ресурс] – Режим доступа: <http://elibrary.ru>
45. Lu W. The PID Controller Based on the Artificial Neural Network and the Differential Evolution Algorithm / W. Lu, J. Yang, X. Liu // Journal of computers. – 2012. – Vol. 7. – P. 2368–2375.
46. Madhusudhana Rao G.A. Neural Network Based Speed Control for DC motor / G.A. Madhusudhana Rao // International Journal of Recent Trends in Engineering. – 2009. – Vol. 2, № 6. – P. 121–124.
47. Metaheuristic Applications to Speech Enhancement / P. Kunche, K.V.V.S. Reddy // SpringerBriefs in Speech Technology. – 2016. – P. 122.
48. Mora M.C. Hand Posture Prediction using Neural Networks within a Biomechanical Model / M.C. Mora, J.L. Sancho-Bru, A. Pérez-González // International Journal of Advanced Robotic Systems. – 2012. – Vol. 9. – P. 1–11.
49. Omatu S. Neuro-PID Control for Electric Vehicle / S. Omatu, M. Yoshioka, T. Fujinaka // Journal of Advanced Computational Intelligence and Intelligent Informatics. – 2010. – Vol. 15, № 7. – P. 82–91.
50. Origins and evolution of enactive cognitive science: Toward an enactive cognitive architecture / L.L. Carvalho, D.J. Pereira, S.A. Coelho // Science direct. – 2016. – № 16. – P. 169–178.
51. Paviani D. Dissertation, The Univ. of Texas, Austin, Tex. / D. Paviani, D. Ph. – 1969.
52. PReLU или Parametric ReLU. [Электронный ресурс] – Режим доступа: <https://congyuzhou.medium.com>
53. Yuguang N. Decentralized coordinated neural control of doubly fed induction generator based wind farm for power system stability support / N. Yuguang, L. Xiaoming, L. Zhongwei // Journal of renewable and sustainable energy. – 2014. – Vol. 6. – P. 217–229.

54. Zeng S. Nonlinear Adaptive PID Control for Greenhouse Environment Based on RBF Network / S. Zeng, H.Hu, L. Xu, G. Li // Sensors. – 2012. – № 12. – P. 5328–5348.