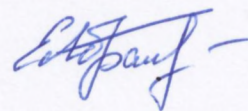


Федеральное государственное бюджетное образовательное учреждение
высшего образования «Владимирский государственный университет имени
Александра Григорьевича и Николая Григорьевича Столетовых» (ВлГУ)

На правах рукописи



Абрамова Елена Сергеевна

АЛГОРИТМЫ ИНКРЕМЕНТНОГО ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ С
ЧЕРЕДОВАНИЕМ РЕЖИМОВ ФУНКЦИОНИРОВАНИЯ И КОНСОЛИДАЦИИ
ЗНАНИЙ ДЛЯ НЕСТАЦИОНАРНЫХ ПОТОКОВЫХ ДАННЫХ

Специальность 2.3.1 – «Системный анализ, управление и обработка информации,
статистика» (технические науки)

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
доктор технических наук, доцент
Орлов Алексей Александрович

Владимир – 2026

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ГЛАВА 1. ОБЗОР И АНАЛИЗ СОВРЕМЕННЫХ МЕТОДОВ ИНКРЕМЕНТНОГО ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ.....	13
1.1 Особенности функционирования нейронных сетей в нестационарных средах	13
1.2 Классификация сценариев и методов инкрементного обучения	21
1.3 Машина экстремального обучения как основа быстрого инкрементного обучения	31
1.4 Анализ инкрементных версий машины экстремального обучения.....	33
Выводы по главе 1.....	38
ГЛАВА 2. РАЗРАБОТКА АЛГОРИТМОВ ПОСМЕННОГО ИНКРЕМЕНТНОГО ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ НА ОСНОВЕ МАШИНЫ ЭКСТРЕМАЛЬНОГО ОБУЧЕНИЯ.....	42
2.1 Разработка модели данных	42
2.2 Математическая модель машины экстремального обучения	43
2.3 Алгоритм инкрементного последовательного и пакетного обучения нейронных сетей на основе машины экстремального обучения с мемоизацией матриц системы нормальных уравнений	46
2.4 Математическая модель метода обратного распространения ошибки	49
2.5 Разработка алгоритма двухфазной консолидации знаний нейронной сети в режиме «сна».....	51
2.6 Разработка алгоритма посменного инкрементного обучения нейронной сети на основе машины экстремального обучения с циклическим чередованием состояний функционирования и «сна»	55
2.7 Разработка алгоритма динамической адаптации архитектуры нейронной сети на основе метрик обученности и упругости нейронов	58
Выводы по главе 2.....	63
ГЛАВА 3. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ РАЗРАБОТАННЫХ АЛГОРИТМОВ НА СИНТЕТИЧЕСКИХ ДАННЫХ	66
3.1 Методика проведения экспериментальных исследований на синтетических данных	66

3.2 Исследование алгоритма посменного инкрементного обучения на задаче аппроксимации	70
3.3 Экспериментальные исследования по настройке гиперпараметров модели	74
3.4 Исследование устойчивости алгоритма посменного обучения к катастрофическому забыванию на синтетических данных	77
3.5 Исследование эффективности алгоритма структурной адаптации нейронной сети в условиях инкрементного обучения	82
Выводы по главе 3.....	90
ГЛАВА 4. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ ПОЛУЧЕННЫХ АЛГОРИТМОВ ПРИ РЕШЕНИИ ЗАДАЧИ РАСПОЗНАВАНИЯ АКТИВНОСТИ ЧЕЛОВЕКА	92
4.1 Постановка задачи распознавания физической активности человека и описание экспериментальных данных.....	92
4.2 Методика проведения экспериментальных исследований	97
4.3 Сравнительное исследование базовых моделей обучения	105
4.4 Исследование эффективности разработанного алгоритма инкрементного обучения	109
4.5 Абляционный анализ разработанного алгоритма.....	115
4.6 Сравнительное исследование с современными методами инкрементного обучения	118
4.7 Исследование вычислительной эффективности и влияния ключевых гиперпараметров	124
4.8 Практическое применение разработанных алгоритмов при распознавании действий персонала.....	130
Выводы по главе 4.....	133
ЗАКЛЮЧЕНИЕ	136
СПИСОК ЛИТЕРАТУРЫ	139
Приложение А Акты о внедрении.....	152
Приложение Б Свидетельства о государственной регистрации программ для ЭВМ	154

ВВЕДЕНИЕ

Актуальность темы исследования

Традиционная парадигма машинного обучения исходит из стационарного распределения данных и полной доступности обучающей выборки на этапе обучения. Она противоречит условиям, в которых работают современные интеллектуальные системы. Во многих прикладных задачах объем поступающей информации непрерывно растет, а ее статистические свойства меняются во времени, поэтому классическое пакетное обучение при ограниченных ресурсах вычислительно неэффективно, а нередко и невозможно.

Главное препятствие для систем инкрементного обучения – это катастрофическое забывание. Оно возникает из-за распределенного характера представлений в нейронных сетях и отсутствия механизмов, которые явно разделяют прежние и новые знания. Особенно остро проблема стоит при нестационарности признакового пространства, типичной для прикладных задач.

Существующие подходы (регуляризация параметров, воспроизведение ранее изученных примеров, модификация структуры сети) частично повышают устойчивость моделей, но требуют значительных вычислений и хранения дополнительных данных либо снижают пластичность модели при освоении новой информации.

Перспективной вычислительной основой для инкрементного обучения служит машина экстремального обучения: она быстро обучается, потому что выходные параметры вычисляются аналитически. Но в базовых инкрементных модификациях этого метода нет специальных механизмов защиты от катастрофического забывания, и это ограничивает их применение при длительной работе на нестационарных данных.

Поэтому актуальна научная задача разработки алгоритмов инкрементного обучения нейронных сетей на основе машины экстремального обучения, которые

адаптируются к изменяющимся данным, сохраняют ранее приобретенные знания и экономно расходуют ограниченные вычислительные ресурсы.

Объект исследования: искусственные нейронные сети, функционирующие в условиях последовательного поступления нестационарных потоковых данных.

Предмет исследования: алгоритмы инкрементного обучения нейронных сетей с чередованием режимов функционирования, консолидации знаний и динамической адаптацией архитектуры, обеспечивающие сохранение ранее приобретенных знаний и адаптацию к изменению статистических свойств данных.

Степень разработанности темы исследования

Фундаментальные основы теории обучения искусственных нейронных сетей и адаптивных систем в условиях нестационарности заложены в работах Галушкина А.И., Цыпкина Я.З., Вапника В.Н., Червоненкиса А.Я., Журавлева Ю.И. и др. В данных исследованиях сформированы теоретические положения обучения нейронных сетей, оценки обобщающей способности, устойчивости алгоритмов и обработки информации в условиях неопределенности. Развитие архитектур глубоких нейронных сетей связано с работами Hinton G.E., LeCun Y., Bengio Y. и др.

Развитию методов самоорганизации, адаптации и обучения интеллектуальных систем в изменяющихся условиях посвящены работы Ивахненко А.Г., Загоруйко Н.Г., Горбаня А.Н., Поспелова Д.А., Синявского О.Ю., а также Gama J., Bifet A., Aljundi R., Parisi G.I. и др. Полученные результаты сформировали основу для разработки методов адаптации моделей к изменяющимся условиям, обработки потоковых данных и постепенного накопления знаний.

Значительный вклад в развитие методов быстрого обучения нейронных сетей внесли Huang G.B., Zhu Q.Y., Siew C.K., предложившие концепцию машины экстремального обучения. Вместе с тем применение метода в условиях непрерывного поступления информации требует дополнительных механизмов сохранения ранее приобретенных знаний, предотвращения катастрофического забывания и адаптации к изменению распределения данных.

Исследования в области инкрементного и непрерывного обучения нейронных сетей получили развитие в работах McCloskey M., Cohen D.E.,

Kirkpatrick J., Li Z., Hoiem D., Rusu A.A., Rebuffi S.-A., Lopez-Paz D., Ranzato M., Fernando C., Robins K., Gepperth A. и др. Предложены методы регуляризации, расширения архитектуры, эпизодической памяти, дистилляции знаний и псевдорепетиции, направленные на снижение катастрофического забывания.

Таким образом, сохраняется противоречие между необходимостью устойчивой адаптации моделей к изменению потока данных и ограниченными возможностями существующих методов. Большинство подходов требуют дополнительных вычислительных ресурсов, хранения информации о предыдущих задачах либо априорного выбора стратегии адаптации. Недостаточно исследованы методы, обеспечивающие совместное использование быстрого обучения, консолидации знаний и структурной адаптации нейронной сети в рамках единого алгоритма инкрементного обучения при изменении характеристик входного потока данных.

Выявленное противоречие определяет актуальную научную задачу, состоящую в разработке алгоритмов инкрементного обучения нейронной сети на основе машины экстремального обучения, обеспечивающих адаптацию модели к нестационарным потокам данных с сохранением ранее приобретенных знаний.

Цель и задачи исследования

Целью диссертационной работы является повышение устойчивости нейронных сетей к катастрофическому забыванию и обеспечение их адаптации к нестационарным потоковым данным за счет разработки алгоритмов инкрементного обучения, предусматривающих чередование режимов функционирования и консолидации знаний, а также динамическую адаптацию архитектуры.

Для достижения поставленной цели необходимо решить следующие **задачи исследования**:

- 1) анализ современных методов инкрементного обучения нейронных сетей, исследование особенностей их функционирования в условиях нестационарных потоковых данных и определение основных причин возникновения катастрофического забывания;

2) разработка комплекса алгоритмов инкрементного обучения нейронной сети на основе машины экстремального обучения, обеспечивающих последовательное обновление модели с сохранением ранее приобретенных знаний, консолидацию накопленной информации при поступлении новых данных и адаптацию архитектуры сети к изменению статистических свойств входного потока;

3) проведение экспериментальных исследований разработанных алгоритмов на синтетических и реальных данных, сравнительный анализ с методами инкрементного обучения и оценка эффективности предложенных решений по показателям качества обучения, устойчивости к катастрофическому забыванию и вычислительной эффективности.

Научная новизна полученных в ходе исследования результатов заключается в следующем:

1) разработан алгоритм посменного инкрементного обучения нейронной сети на основе машины экстремального обучения, отличающийся тем, что процесс обучения организован в виде циклического чередования режима функционирования, основанного на аналитическом обновлении выходных весов, и режима консолидации знаний («сна») с адаптацией параметров модели, что позволяет обеспечить непрерывное инкрементное обучение нейронной сети при снижении влияния катастрофического забывания;

2) разработан алгоритм двухфазной консолидации знаний нейронной сети в режиме «сна», отличающийся тем, что адаптация параметров скрытого слоя и аналитический пересчет выходных весов выполняются последовательно с использованием генерируемых сетью псевдопримеров без хранения полного набора исходных данных, что позволяет повысить сохранность ранее приобретенных знаний при инкрементном обучении;

3) разработан алгоритм динамической адаптации архитектуры нейронной сети в процессе инкрементного обучения, отличающийся тем, что динамическое изменение структуры скрытого слоя выполняется на основе оценки обученности, упругости и значимости нейронов посредством добавления новых нейронов и

функционального разделения существующих, что позволяет повысить адаптивность модели при изменении статистических характеристик входных данных.

Теоретическая значимость исследования заключается в развитии методов инкрементного обучения нейронных сетей в условиях нестационарных потоковых данных. Систематизированы и критически проанализированы современные подходы к последовательному обучению нейронных моделей, включая методы на основе машины экстремального обучения и обратного распространения ошибки; выявлены ограничения существующих методов, связанные с сохранением ранее приобретенных знаний, устойчивостью при длительной работе и возникновением катастрофического забывания.

Обоснована возможность совместного применения быстрого обучения на основе ELM и адаптивной коррекции параметров методом обратного распространения ошибки в едином процессе посменного инкрементного обучения. Это расширяет представления о способах организации последовательного обновления нейросетевых моделей при сохранении накопленных знаний. Предложенный подход к двухфазной консолидации знаний в режиме «сна» развивает научные основы биологически инспирированных механизмов сохранения информации в нейронных сетях: процессы активного обучения и стабилизации параметров модели разделены, и обращение к исходным обучающим примерам не требуется. Введение и формализация критериев обученности, пластичности и значимости нейронов для управления динамической адаптацией архитектуры сети прямого распространения развивают теоретический аппарат методов структурной адаптации в задачах инкрементного обучения.

Практическая значимость работы заключается в разработке программно реализованных алгоритмов инкрементного обучения нейронных сетей, которые повышают устойчивость и качество обработки динамически изменяющихся данных. Предложенные решения адаптируют модель к новым данным с сохранением ранее приобретенных знаний, автоматически наращивают емкость сети при изменении характеристик входного потока и снижают влияние

катастрофического забывания без хранения полного набора исходных обучающих примеров, что сокращает требования к памяти. Эффективность алгоритмов подтверждена экспериментально на синтетических и реальных данных.

Сформулированы рекомендации по применению алгоритмов в адаптивных системах обработки потоковых данных в реальном времени. Результаты внедрены в учебный процесс и научно-исследовательскую работу кафедры физики и прикладной математики МИ ВлГУ, а также в производство ООО «СТРОЙТАЙМ ПЛЮС» (Приложение А).

Методология и методы исследования. Для решения поставленных задач применены методы системного анализа и теории управления для исследования процессов инкрементного обучения нейронных сетей в условиях нестационарных потоковых данных; методы машинного обучения для построения и обучения нейросетевых моделей; методы математической статистики и теории планирования эксперимента для оценки достоверности полученных результатов; методология компьютерного моделирования для экспериментального исследования разработанных алгоритмов на синтетических и реальных данных.

Положения, выносимые на защиту

1. Алгоритм посменного инкрементного обучения нейронной сети на основе машины экстремального обучения обеспечивает снижение катастрофического забывания при непрерывной адаптации модели к поступающим данным посредством циклического чередования режима функционирования, основанного на аналитическом обновлении выходных весов, и режима консолидации знаний («сна»), что подтверждено снижением относительного забывания до 0,23 при сохранении средней точности 0,80 на последовательности синтетических задач и 0,77 при распознавании физической активности человека (п. 4 паспорта специальности 2.3.1).

2. Алгоритм двухфазной консолидации знаний нейронной сети в режиме «сна» обеспечивает снижение забывания ранее приобретенных знаний без хранения полного набора исходных данных посредством последовательной адаптации параметров скрытого слоя и аналитического пересчета выходных весов

с использованием генерируемых сетью псевдопримеров, что подтверждено снижением абсолютного забывания до 69,8% и ограничением роста ошибки на ранее изученных данных в 3 раза по сравнению с OS-ELM (п. 4 паспорта специальности 2.3.1).

3. Алгоритм динамической адаптации архитектуры нейронной сети обеспечивает повышение адаптивности модели при изменении статистических характеристик данных посредством изменения структуры скрытого слоя на основе оценки обученности, упругости и значимости нейронов путем добавления новых нейронов при превышении порога ошибки и функционального разделения существующих нейронов, что подтверждено снижением среднеквадратической ошибки на 65,2% при обучении на новых данных и сохранением устойчивости к забыванию на уровне, сопоставимом с EWC, при превосходстве по устойчивости к забыванию на 10,8% (п. 4 паспорта специальности 2.3.1).

4. Разработанная программная система распознавания физической активности человека по данным инерциальных датчиков, реализующая алгоритмы посменного обучения, консолидации знаний и адаптации архитектуры, обеспечивает сохранение ранее приобретенных знаний при непрерывной обработке потока данных, что подтверждается достигнутым в стрессовом сценарии с пермутацией признаков значением забывания 0,005 при 0,017 у EWC и 0,023 у ER (п. 5 паспорта специальности 2.3.1).

Степень достоверности и апробация результатов

Достоверность полученных результатов обеспечивается использованием общепринятых методик экспериментальной оценки алгоритмов инкрементного обучения, применением математического аппарата, включающего аналитические методы решения систем линейных уравнений и рекуррентные алгоритмы наименьших квадратов, а также обоснованностью принятых допущений. Вычислительные эксперименты проводились с многократными независимыми запусками при фиксированной методике проведения экспериментов, что позволило вычислить средние значения и доверительные интервалы оцениваемых показателей.

Сравнение разработанных алгоритмов с современными подходами (EWC, SI, ER, OS-ELM) и с результатами пакетного обучения выполнялось на синтетических данных с контролируемой сложностью, а также на открытом наборе данных, содержащем 630 тыс. наблюдений, в единых условиях эксперимента.

Воспроизводимость результатов подтверждена на различных конфигурациях разработанного алгоритма и сценариях поступления данных, включая стрессовый сценарий с пермутацией признаков, моделирующий радикальную нестационарность. Разработанное программное обеспечение прошло верификацию в ходе тестирования. Достоверность результатов также подтверждается согласованностью полученных выводов с известными теоретическими положениями и результатами исследований отечественных и зарубежных авторов в области инкрементного обучения нейронных сетей.

Методологическое и программное обеспечение разработаны в ходе выполнения научно-исследовательской работы в составе научного коллектива кафедры физики и прикладной математики Муромского института (филиала) ВлГУ по гранту Российского научного фонда (РНФ) № 24-21-00386 «Исследование процессов посменного применения машины экстремального обучения (как режима функционирования) и глубокого обучения (как режима «сна») для создания инкрементных методов обучения искусственных нейронных сетей», а также проекта, поддержанного Министерством молодежной политики Владимирской области «Методы и алгоритмы инкрементного обучения нейронных сетей для интеллектуального анализа производственной активности персонала в условиях потоковых данных».

Основные результаты диссертационного исследования докладывались и обсуждались на международных и всероссийских научных конференциях, в том числе: Международной научно-технической конференции «Распознавание» (Курск, 2018, 2019, 2021, 2023, 2025); Международной мультидисциплинарной конференции FarEastCon (Владивосток, 2019); Международных научно-технических конференциях ICIEAM (Сочи, 2021–2022), SmartIndustryCon (Сочи, 2023–2024) и RusAutoCon (Сочи, 2023–2025); Международной научно-технической

конференции «Динамика систем, механизмов и машин» (Dynamics) (Омск, 2023); Всероссийской конференции с международным участием «Математические методы распознавания образов» (Муром, 2025); а также на Всероссийских научных конференциях «Зворыкинские чтения» (Муром, 2018–2026).

Основные теоретические и практические результаты диссертационной работы опубликованы в 35 научных работах, среди которых 6 статей в рецензируемых научных изданиях из перечня ВАК Минобрнауки РФ и 9 публикаций, индексируемых в международных базах данных Scopus и Web of Science. Имеется 2 свидетельства о государственной регистрации программ для ЭВМ (Приложение Б).

Личный вклад. Основные результаты диссертационной работы получены автором лично. Постановка цели и задач, направления исследований и анализ результатов обсуждались совместно с научным руководителем. Автором лично разработаны алгоритм посменного инкрементного обучения нейронной сети с циклическим чередованием состояний функционирования и «сна» [80; 81], алгоритм двухфазного обучения нейронной сети в режиме «сна» [82; 109], а также алгоритм динамической адаптации архитектуры нейронной сети на основе метрик обученности и упругости нейронов [84; 93]. Для каждого из разработанных алгоритмов выполнены программная реализация, проведение экспериментальных исследований и анализ полученных результатов. Во всех совместных публикациях соискатель принимал непосредственное участие и внес существенный вклад на всех этапах работы.

Структура и объем диссертации

Диссертационная работа состоит из введения, 4 глав с выводами и заключения. Диссертационная работа изложена на 151 странице машинописного текста и содержит 25 рисунков, 27 таблиц, 2 приложения общим объемом 4 страницы. Список литературы состоит из 112 источников.

ГЛАВА 1. ОБЗОР И АНАЛИЗ СОВРЕМЕННЫХ МЕТОДОВ ИНКРЕМЕНТНОГО ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

1.1 Особенности функционирования нейронных сетей в нестационарных средах

Традиционная парадигма машинного обучения исторически опирается на строгое статистическое предположение, что обучающая и тестовая выборки независимы и одинаково распределены (independent and identically distributed, i.i.d.) [1], а также на то, что все обучающие данные доступны одновременно до начала обучения [2].

При таком подходе применяют пакетное обучение (batch learning): модель обучается на полном наборе данных, затем ее параметры фиксируются, и модель развертывается для решения прикладных задач [3].

Жизненный цикл классической модели машинного обучения, как правило, включает несколько последовательных этапов: сбор и разметку данных, их предобработку, итеративное обучение модели до сходимости функции потерь, валидацию и финальное развертывание. После развертывания модель считается статичной: она работает в режиме инференса и генерирует прогнозы для новых входных данных.

Если в ходе эксплуатации накапливается новая информация или изменяются статистические свойства среды, модель не способна адаптироваться к этим изменениям автономно. Тогда требуется полный перезапуск жизненного цикла: сбор новой выборки, ее объединение с историческими данными и переобучение модели «с нуля» (retraining from scratch).

Пакетные методы хорошо работают в стационарных условиях, но цифровизация, развитие Интернета вещей (IoT) и широкое внедрение носимых устройств делают их применение неэффективным или невозможным [4]. В реальной эксплуатации интеллектуальных систем (мобильные приложения, системы промышленного мониторинга, робототехнические комплексы,

медицинские датчики) данные длительное время поступают непрерывным потоком.

В таких условиях классическое пакетное обучение сталкивается с фундаментальными ограничениями, которые делятся на три основные группы.

Во-первых, это проблема вычислительной сложности и энергопотребления. Переобучение модели «с нуля» на объединенной выборке требует значительных вычислительных ресурсов и времени [5]. Для современных глубоких нейронных сетей с миллионами параметров этот процесс может занимать часы или даже дни работы высокопроизводительных серверов с графическими ускорителями (GPU) [6]. При этом целевой платформой для многих приложений (например, систем распознавания активности на смартфоне или умных часах) служат ресурсоограниченные Edge-устройства. Итеративно переобучать модель на таких устройствах физически невозможно из-за жестких ограничений по энергопотреблению, объему оперативной памяти и вычислительной мощности процессора.

Во-вторых, это проблема хранения данных и конфиденциальности. При потоковом поступлении информации ее объем может быть практически неограничен, а хранить полную историю наблюдений в оперативной памяти устройства невозможно. Кроме того, во многих прикладных областях (в частности, в медицине и персональном мониторинге здоровья) непрерывное хранение сырых пользовательских данных нарушает требования конфиденциальности и нормативные акты (например, GDPR) [7]. Следовательно, модель не может иметь доступа ко всей истории наблюдений для пакетного переобучения.

В-третьих, это проблема концептуальной устарелости (conceptual staleness). Даже если у системы достаточно вычислительных ресурсов для периодического переобучения, между циклами обновления накапливается значительный объем новых данных. Все это время модель работает с устаревшими знаниями, из-за чего ее качество падает, а в изменившихся условиях среды она принимает некорректные решения [8].

Классический подход, при котором обучение и эксплуатация разделены, не удовлетворяет требованиям современных интеллектуальных систем, работающих в динамически изменяющихся средах (Рисунок 1). Поэтому необходим переход от пакетного к инкрементному обучению: модель усваивает новые знания по мере поступления отдельных примеров или небольших пакетов (батчей) данных, непрерывно адаптируется к изменениям, не требует доступа к полной истории наблюдений и не останавливает эксплуатацию ради ресурсоемкого переобучения [9].

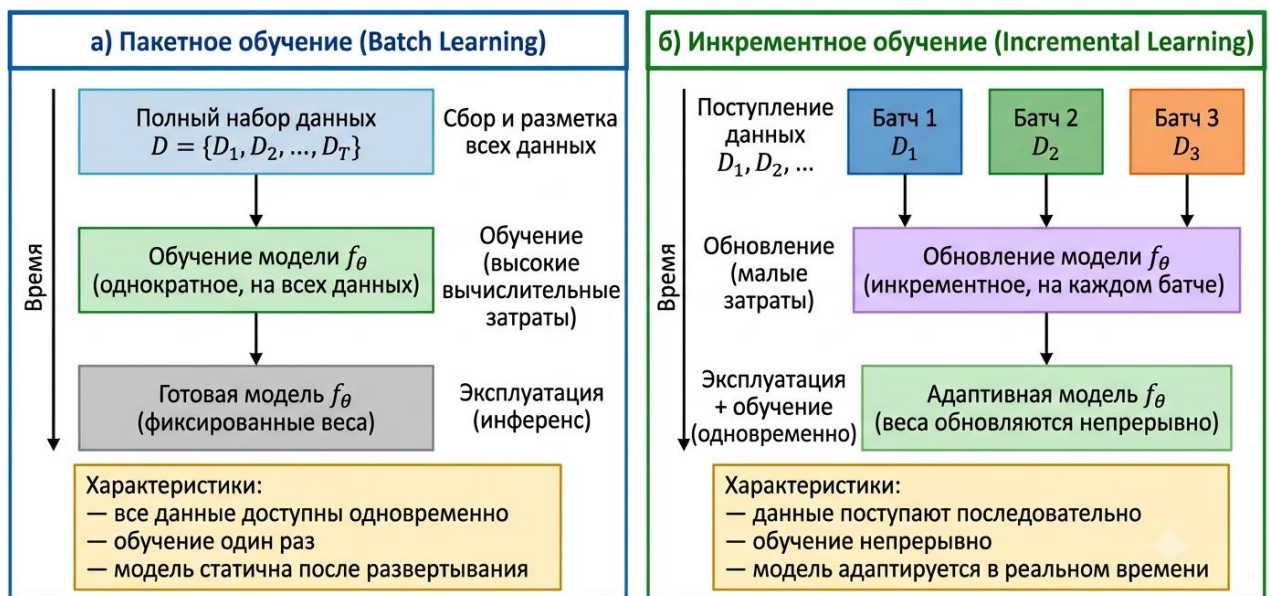


Рисунок 1 – Сравнение парадигм обучения нейронных сетей

Переход от пакетного обучения к обработке потоковых данных неизбежно сталкивает исследователей с нарушением базового статистического допущения о стационарности распределений. В реальной эксплуатации интеллектуальных систем в динамически изменяющейся среде статистические свойства входных данных и характер зависимостей между признаками и целевыми переменными не остаются постоянными во времени. Это явление называют нестационарностью данных. Оно проявляется в двух фундаментальных формах: концептуальном дрейфе и нестационарности самого признакового пространства (сдвиге ковариат) [10].

Концептуальный дрейф (concept drift) определяется как изменение во времени условного распределения целевой переменной при заданных входных

признаках. Формально, если в момент времени t зависимость описывается условной вероятностью $P_t(Y | X)$, то концептуальный дрейф имеет место при $P_t(Y | X) \neq P_{t+1}(Y | X)$. То есть меняются сами скрытые закономерности или правила, связывающие входные воздействия с выходными откликами.

Концептуальный дрейф может быть внезапным (например, резкое изменение режима работы технического оборудования), постепенным (сезонные изменения в поведении пользователей) или инкрементальным (медленная деградация характеристик сенсоров) [11]. Для нейросетевых моделей он представляет серьезную угрозу: модель, обученная на исторических данных, начинает систематически ошибаться на новых примерах, поскольку ее внутренние представления перестают соответствовать актуальной структуре данных.

Однако для задач, решаемых с помощью носимых устройств и мобильных сенсоров, не менее критична, а зачастую и сложнее нестационарность признакового пространства (feature space non-stationarity), или сдвиг ковариат (covariate shift).

В отличие от концептуального дрейфа, при сдвиге ковариат условная зависимость $P(Y | X)$ может оставаться неизменной, но радикально меняется маргинальное распределение входных признаков $P_t(X) \neq P_{t+1}(X)$. Для нейросетевого моделирования это означает, что статистические свойства, масштаб, дисперсия или даже физический смысл входных признаков изменяются в ходе работы системы.

Характерный пример нестационарности признакового пространства: задачи распознавания физической активности человека (Human Activity Recognition, HAR) по данным инерциальных сенсоров. Статистические характеристики сигналов акселерометра и гироскопа критически зависят от положения смартфона на теле пользователя [12]. Если переложить устройство из кармана брюк на пояс или на запястье, амплитудно-частотные характеристики сигналов для одного и того же класса активности (например, ходьбы) меняются настолько сильно, что в пространстве признаков они начинают перекрываться с другими классами.

Кроме того, на нестационарность признаков влияют индивидуальная усталость пользователя (изменение биомеханики движений), температурные дрейфы калибровки сенсоров и появление специфических аппаратных шумов. В итоге один и тот же класс объектов в разные моменты времени описывается принципиально различными векторами признаков.

Совокупность концептуального дрейфа и нестационарности признакового пространства создает для алгоритмов машинного обучения экстремально сложную среду. Чтобы адаптироваться к новым данным, модель вынуждена постоянно обновлять свои параметры. Если у архитектуры сети и алгоритма обновления весов нет специальных механизмов защиты, подстройка под новые статистические распределения неизбежно «перезаписывает» синаптические связи, отвечающие за ранее усвоенные закономерности.

Классические методы регуляризации, которые фиксируют важные веса сети, оказываются неэффективными при радикальной нестационарности признакового пространства. Веса, критически важные для распознавания паттернов в одном распределении признаков, при сдвиге ковариат могут стать совершенно бесполезными или даже вредными. Сеть перестает обучаться, ее пластичность снижается.

Нестационарность данных и признакового пространства не только снижает абсолютную точность классификации, но и запускает более глубокие деградационные процессы в нейронных сетях. Необходимость непрерывно адаптироваться к изменяющимся распределениям, не теряя способности к обобщению на ранее изученных паттернах, напрямую подводит к проблеме катастрофического забывания (*catastrophic forgetting*) и формулирует центральную дилемму инкрементного обучения [13; 14].

Катастрофическое забывание заключается в резкой и практически необратимой деградациии качества модели на ранее изученных задачах при обучении на новых данных. В отличие от постепенного ухудшения обобщающей способности при классическом переобучении, оно проявляется скачкообразно: после предъявления всего нескольких примеров из нового распределения

нейронная сеть может полностью утратить способность корректно решать задачи, которые ранее выполняла с высокой точностью.

Механизм катастрофического забывания обусловлен самой архитектурой нейронных сетей прямого распространения. Синаптические веса в такой сети представляют собой распределенное представление знаний: одна и та же группа нейронов и связей одновременно участвует в кодировании различных паттернов и классов [15]. При обновлении весов методом градиентной оптимизации на новых данных меняются те же параметры, которые отвечали за воспроизведение ранее усвоенных зависимостей. В результате новые знания фактически «перезаписывают» старые, поскольку градиентный спуск не различает, какие синаптические связи критически важны для решения предыдущих задач, а какие можно свободно менять.

Катастрофическое забывание тесно связано с одной из центральных концепций когнитивной науки и теории обучения, дилеммой стабильности-пластичности (stability-plasticity dilemma). Чтобы работать в динамически изменяющейся среде, система должна сочетать два взаимоисключающих свойства. С одной стороны, ей необходима пластичность, то есть способность быстро и эффективно усваивать новую информацию, адаптироваться к концептуальному дрейфу и нестационарности признакового пространства. С другой стороны, системе требуется стабильность, то есть способность защищать ранее сформированные внутренние представления от разрушения при обучении на новых данных.

Избыточная пластичность делает модель восприимчивой к каждому новому примеру и приводит к систематической потере накопленных знаний. Избыточная стабильность, напротив, делает сеть функционально зафиксированной: она сохраняет ранее усвоенные паттерны, но не способна усвоить принципиально новую информацию и приспособиться к изменившимся условиям среды. Поиск баланса между этими двумя свойствами составляет ключевую задачу при разработке алгоритмов инкрементного обучения, и от его результата зависит практическая применимость модели в реальных сценариях эксплуатации.

Дилемма стабильности-пластичности имеет глубокие биологические корни. В нейробиологии давно установлено, что человеческий мозг решает эту проблему, разделяя функции между разными структурами: гиппокамп обеспечивает быстрое запоминание отдельных эпизодов (высокая пластичность), а неокортекс отвечает за медленную консолидацию и обобщение накопленных знаний (высокая стабильность) [16]. Консолидация памяти, идущая преимущественно во время «сна», позволяет переносить информацию из кратковременного хранилища в долговременное и обеспечивает ее устойчивое хранение без интерференции с новыми впечатлениями.

В искусственных нейронных сетях дилемма стабильности-пластичности проявляется особенно остро, потому что механизмы быстрого и медленного обучения явно неразделены. Все параметры сети обновляются в едином контуре градиентной оптимизации, поэтому изменить одни знания, не затронув другие, невозможно. Попытки искусственно разделить эти функции, например, введением дополнительных слоев или специализированных модулей памяти, существенно усложняют архитектуру модели и увеличивают вычислительные затраты, что зачастую неприемлемо для ресурсограниченных приложений.

Нестационарность признакового пространства усугубляет катастрофическое забывание. Когда статистические свойства входных данных радикально меняются от задачи к задаче, модель вынуждена существенно перестраивать свои внутренние представления. В таких условиях даже незначительные обновления весов могут привести к полной утрате ранее усвоенных знаний, поскольку новые данные проецируются в пространство признаков, структурно несовместимое с тем, в котором были сформированы старые представления.

Градиентные методы оптимизации, и в первую очередь метод обратного распространения ошибки (backpropagation, BP), составляют математический фундамент современного машинного обучения и глубоких нейронных сетей [17]. Суть этих методов сводится к итеративному вычислению градиентов функции потерь по синаптическим весам сети и последующей корректировке параметров в направлении антиградиента. В условиях пакетного обучения на стационарных

наборах данных градиентные подходы очень эффективны и точно аппроксимируют сложные нелинейные зависимости. Именно благодаря им были достигнуты основные прорывы в компьютерном зрении, обработке естественного языка и распознавании образов.

Однако перенос этой парадигмы на потоковую обработку данных и инкрементное обучение выявляет фундаментальные ограничения, связанные как с математической природой градиентного спуска, так и с архитектурными особенностями нейронных сетей.

Главное ограничение при переходе к последовательному обучению состоит в отсутствии встроенных механизмов разделения знаний. Поскольку одни и те же синаптические веса участвуют в кодировании информации обо всех ранее изученных распределениях данных, обновление параметров градиентным спуском на новых примерах неизбежно приводит к интерференции. Градиентный оптимизатор не различает, какие веса критически важны для сохранения старых паттернов, а какие можно свободно менять для усвоения новых. Катастрофическое забывание возникает не как побочный эффект, а как прямое следствие распределенного характера представлений в нейронных сетях, обучаемых градиентными методами.

С вычислительной точки зрения итеративная природа градиентных методов требует многократного прохода по обучающей выборке до сходимости функции потерь. В инкрементном сценарии, когда данные поступают непрерывно, это создает недопустимо высокую вычислительную нагрузку и энергопотребление, поэтому прямое применение классического градиентного спуска на ресурсоограниченных периферийных устройствах затруднительно. Кроме того, чтобы предотвратить забывание, часто приходится хранить исторические данные или применять сложные механизмы регуляризации, что дополнительно увеличивает требования к памяти.

В глубоких сетях градиенты при обратном распространении могут экспоненциально убывать или возрастать, что приводит к численной нестабильности [18]. Оптимизация крайне чувствительна к выбору

гиперпараметров (скорости обучения, размера мини-батча, коэффициентов регуляризации) и к порядку поступления данных. В нестационарной среде, где статистические свойства входного потока меняются во времени, параметры, оптимальные для одного распределения, могут оказаться неэффективными или дестабилизирующими для другого, а порядком предъявления примеров, как правило, управлять невозможно.

Классические градиентные методы, будучи наиболее эффективным инструментом обучения нейронных сетей в стационарных условиях, сталкиваются с серьезными препятствиями при прямой адаптации к задачам инкрементного обучения в нестационарных средах. Высокая вычислительная сложность, чувствительность к порядку данных и, что важнее всего, склонность к катастрофическому забыванию требуют поиска специализированных архитектурных и алгоритмических решений. Простым масштабированием или настройкой гиперпараметров стандартного градиентного спуска эти ограничения не преодолеть: нужны принципиально новые стратегии обновления параметров, способные обеспечить баланс между пластичностью и стабильностью модели при непрерывном потоке данных [19].

1.2 Классификация сценариев и методов инкрементного обучения

В современной литературе, посвященной адаптации моделей машинного обучения к изменяющимся условиям, терминология неоднородна. Наряду с понятием «инкрементное обучение» (Incremental Learning) часто используются термины «онлайн-обучение» (Online Learning), «непрерывное обучение» (Continual Learning) и «обучение на протяжении всей жизни» (Lifelong Learning). Эти понятия имеют общие корни и частично пересекаются, но между ними есть смысловые различия [20].

Онлайн-обучение традиционно делает упор на обработке данных по мере их поступления (вплоть до обучения на каждом отдельном примере), чтобы

минимизировать вычислительные затраты, но проблемой сохранения знаний о прошлых распределениях занимается не всегда.

Обучение на протяжении всей жизни опирается на биологическую метафору: система накапливает знания в течение длительного времени, переносит их между разными задачами и адаптируется к новым без полного переобучения [21].

В настоящей диссертационной работе используется термин «инкрементное обучение», поскольку он точнее всего отражает суть процесса: новые данные или задачи поступают последовательно, порциями, и параметры модели необходимо обновлять инкрементно, то есть пошагово, при строгом ограничении доступа к историческим данным [22].

Ключевая цель инкрементного обучения: достичь баланса между пластичностью (способностью усваивать новую информацию) и стабильностью (способностью сохранять накопленные знания). Это позволяет преодолеть катастрофическое забывание, описанное в разделе 1.1.

Для систематизации подходов и корректной постановки экспериментов в научном сообществе принято делить инкрементное обучение на три основных сценария. Они различаются степенью доступности метаданных о задачах при тестировании и характером изменения выходного пространства (Рисунок 2) [23].

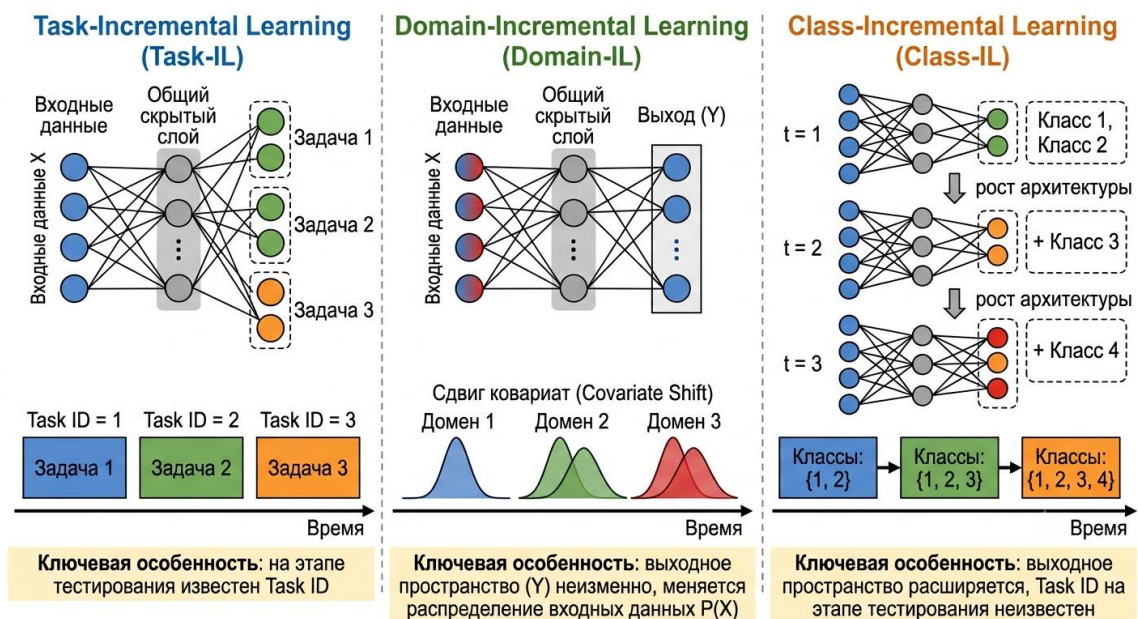


Рисунок 2 – Сценарии инкрементного обучения

Первый сценарий: инкрементное обучение по задачам (Task-Incremental Learning, Task-IL). В нем при тестировании модели дан явный идентификатор задачи (Task ID), к которой относится входной пример. То есть системе известно, какой навык или какое распределение данных нужно применить сейчас. Модель обучает параметры, специфичные для каждой задачи, или использует механизмы маршрутизации.

Этот сценарий проще всего с точки зрения предотвращения катастрофического забывания, поскольку конфликта классов на этапе вывода нет. Но в реальных автономных системах он встречается редко, потому что природа поступающих данных заранее неизвестна.

Второй сценарий: инкрементное обучение по доменам (Domain-Incremental Learning, Domain-IL). Здесь структура выходного пространства (множество классов или задача регрессии) остается прежней, но статистическое распределение входных данных (домен) последовательно меняется от задачи к задаче. Идентификатор задачи при тестировании не предоставляется. Модель должна адаптироваться к сдвигу ковариат или концептуальному дрейфу и при этом продолжать корректно обрабатывать примеры из предыдущих доменов [24].

Этот сценарий тесно связан с проблемой нестационарности признакового пространства, рассмотренной в разделе 1.1, и типичен для систем с сенсорными данными, где меняются условия сбора (например, положение датчика на теле пользователя, уровень шума, индивидуальные особенности пользователя).

Третий сценарий: инкрементное обучение по классам (Class-Incremental Learning, Class-IL). Модель последовательно обучается на новых классах, и число распознаваемых классов постоянно растет. Идентификатор задачи при тестировании не предоставляется, поэтому модель должна не просто распознать объект, а отнести его к одному из всех ранее изученных классов. Отсюда фундаментальная проблема: пространство решений расширяется, а границы между старыми и новыми классами неизбежно смещаются под действием градиентной оптимизации.

Class-IL наиболее репрезентативен для большинства практических задач интеллектуального анализа данных, включая распознавание образов. В реальных условиях система сталкивается с принципиально новыми категориями объектов, которые нельзя разметить заранее и о принадлежности которых к этапу обучения ничего не известно.

Анализ сценариев показывает, что сложность задачи растет от Task-IL к Class-IL. Для Task-IL существуют относительно простые архитектурные решения (например, изолированные подсети для каждой задачи), а для Domain-IL и особенно Class-IL нужны сложные механизмы регуляризации, консолидации памяти или динамического расширения архитектуры.

Выбор метода инкрементного обучения зависит от сценария. Методы на основе масок применимы главным образом в Task-IL, поскольку для маршрутизации вычислений им нужен явный идентификатор задачи. Методы расширения архитектуры и регуляризации тоже ориентированы на Task-IL, хотя отдельные их модификации можно адаптировать к Domain-IL. Псевдорепетиционные и онлайн-методы, напротив, изначально создавались для Domain-IL и Class-IL, где идентификатор задачи недоступен, но у каждого из этих подходов есть свои ограничения, связанные с вычислительными затратами или устойчивостью при длительном обучении.

Анализ научной литературы позволяет выделить пять основных семейств методов инкрементного обучения. Каждое по-своему решает дилемму стабильности-пластичности.

Методы на основе масок (Mask-based methods) уменьшают или полностью устраняют катастрофическое забывание, применяя бинарные маски к каждому параметру или к представлениям каждого слоя. Идея в том, чтобы для каждой задачи выделить подмножество весов сети, которое используется только для ее решения, а остальные параметры оставить неизменными.

В работе [25] предложен метод адаптации одной фиксированной глубокой нейронной сети к нескольким задачам без ухудшения качества на уже изученных. На идеях квантования и отсечения сети обучаются двоичные маски, которые

накладываются на существующую сеть, чтобы она хорошо решала новую задачу [26]. Метод PathNet [27] обучает выборочную маршрутизацию по весам с помощью эволюционных стратегий.

Достоинства методов на основе масок: катастрофическое забывание минимально, менять архитектуру сети не нужно. Но им необходим идентификатор задачи при тестировании, поэтому для Domain-IL и Class-IL они не подходят. Кроме того, при большом числе задач на каждую новую остается слишком мало параметров, и емкость модели ограничена.

Методы расширения архитектуры (Dynamic architectures) уменьшают катастрофическое забывание, динамически увеличивая пропускную способность сети. Они строятся на модели, которая сохраняет прежние знания о задачах и одновременно расширяется для изучения новых.

В работе [28] предлагается расширять глубокую сверточную нейронную сеть методом «клонирования и ветвления»: сеть изучает новые задачи одну за другой, не теряя качества на старых. В статье [29] представлена структура инкрементного обучения на основе синтеза нейронных сетей с расширением и сокращением. Когда поступают новые данные, сеть сначала создает новые соединения по градиентам, чтобы увеличить пропускную способность, а затем итеративно удаляет соединения по величине весов, чтобы стать компактнее.

Методы расширения архитектуры подстраивают емкость модели под сложность данных и сохраняют ранее изученные представления. Но они рискуют привести к неконтролируемому росту сети, трудно определить, когда и насколько расширять, а при добавлении новых параметров нужно дообучать уже существующие.

Методы регуляризации (Regularization-based methods) смягчают катастрофическое забывание, добавляя в функцию потерь штрафной член, который ограничивает изменение параметров, важных для старых задач. По подходу их делят на два типа: регуляризация весов и дистилляция знаний [30]. Такие стратегии привлекательны, потому что эффективны и не требуют дополнительного хранилища данных.

Метод Elastic Weight Consolidation (EWC) [31] оценивает важность каждого веса по диагонали информационной матрицы Фишера, вычисленной на предыдущих задачах, и штрафует изменение наиболее важных параметров. Метод Synaptic Intelligence (SI) [32] оценивает важность весов онлайн по траектории градиентного спуска в ходе обучения, поэтому знания о значимости параметров накапливаются непрерывно. Метод Learning without Forgetting (LwF) [33] основан на дистилляции знаний: данные новой задачи размечаются по результатам старой модели, и эти метки ограничивают обновление параметров.

Методы регуляризации не требуют хранения данных и вычислительно эффективны. Но им нужен идентификатор задачи, гиперпараметры трудно настраивать, а пластичность модели снижается. Самое серьезное ограничение проявляется при радикальной нестационарности признаковов пространства: матрица Фишера и интегральные оценки градиентов, вычисленные на прежних задачах, теряют информативность и слабо коррелируют с новым распределением, поэтому сеть застревает в конфигурации, оптимальной для неактуального распределения [34].

Псевдорепетиционные методы (Pseudo-rehearsal methods) позволяют модели повторно обращаться к старым знаниям при изучении новых: они хранят подмножество прежних данных или генерируют синтетические примеры.

Метод репетиции (Experience Replay, ER) [35] сохраняет подмножество реальных примеров из предыдущих задач в буфере памяти и подмешивает их в обучающие батчи, когда приходят новые данные. Метод iCaRL [36] сочетает извлечение знаний и репетицию прототипов: образы отбираются по ближайшему расстоянию до прототипов классов. Метод BIC [37; 38] добавляет корректирующий слой, который исправляет смещение последнего полносвязного слоя и тем самым эффективно решает проблему дисбаланса данных.

Чтобы не хранить образы и обойти проблемы конфиденциальности, присущие классической репетиции, некоторые методы учатся генерировать примеры предыдущих задач с помощью генеративно-состязательных сетей [39]. Развитие генеративных моделей дает стратегии псевдорепетиций больше

возможностей, но для обучения генератора нужны дополнительные вычислительные ресурсы [40; 41].

Псевдорепетиционные методы дают высокую точность и работают в сценарии Class-IL без идентификатора задачи. Но они требуют затрат на хранение данных или генерацию примеров, порождают проблемы конфиденциальности и повышают вычислительные затраты. При радикальной смене распределения признаков сохраненные примеры могут перестать быть репрезентативными, и метод работает хуже.

Онлайн-методы (Online Incremental Learning) построены на потоковых структурах, где каждый пример можно увидеть только один раз. В работе [42] представлена модель непрерывного обучения с градиентной эпизодической памятью (Gradient Episodic Memory, GEM): она смягчает забывание и позволяет при этом переносить знания на решение предыдущих задач. Улучшенная версия A-GEM [43] усредняет градиенты, чтобы обновлять параметры стабильнее.

Онлайн-методы дают высокую скорость обучения, минимальные требования к памяти и работу в реальном времени. Однако в них накапливаются ошибки, при длительной работе возникает переобучение, а результат зависит от порядка поступления данных [44].

Фундаментальные ограничения рассмотренных инженерных методов заставляют исследователей обращаться к биологическим прототипам. Эти подходы иначе смотрят на решение дилеммы стабильности-пластичности. В нейробиологии установлено, что человеческий мозг решает эту проблему, разделяя функции между разными структурами: гиппокамп обеспечивает быстрое запоминание отдельных эпизодов (высокая пластичность), а неокортекс отвечает за медленную консолидацию и обобщение накопленных знаний (высокая стабильность) [45]. Консолидация памяти, идущая преимущественно во время «сна», позволяет переносить информацию из кратковременного хранилища в долговременное и обеспечивает ее устойчивое хранение без интерференции с новыми впечатлениями.

В работе [45] авторы выдвинули гипотезу о функциональной роли «сна», включающую процесс «разучивания»: процесс, аналогичный обучению, но с

обратным знаком и начинающийся с шумового входа, повышает производительность сети при доступе к реальным воспоминаниям и минимизирует ложные. Система ведет себя так же, как человек, а это необходимо для предполагаемой роли «разучивания» во сне с быстрым движением глаз.

В статье [46] за основу была взята нейронная сеть Хопфилда, которая приближает принципы работы человеческого мозга. Авторы предложили аналог суточного цикла: фазу бодрствования сменяла фаза сна. Сам «сон», как и у млекопитающих, делился на «медленный» (глубокий сон без сновидений, при котором, как полагают, в мозге идет консолидация памяти) и «быстрый» (в этой фазе, как считают некоторые, мозг избавляется от лишней информации).

В работе [47] «сон» в нейронных сетях трактуется как удаление слабых нейронных связей, благодаря которому сеть избавляется от избыточной информации и лучше обобщает. В [48] для моделирования процессов «сна» предложено использовать методы обучения импульсных нейросетей.

Однако анализ этих работ показывает, что существующие бионические подходы к моделированию «сна» в нейронных сетях сводятся главным образом к разреживанию синаптических связей, подавлению шумовых паттернов или удалению слабых весовых коэффициентов. Такие механизмы стабилизируют модель, но не обеспечивают активной консолидации знаний, нужной, чтобы сеть могла и дальше учиться на новых данных. Кроме того, в рассмотренных работах фаза «сна», как правило, конечная: после нее модель не предполагает инкрементного обновления. Поэтому существующие бионические аналоги не в полной мере воспроизводят функциональную роль «сна» как подхода, который обеспечивает баланс между пластичностью и стабильностью при непрерывном обучении.

В Таблице 1 представлено обобщенное сравнение семейств методов инкрементного обучения.

Таблица 1 – Сравнительный анализ методов инкрементного обучения

Семейство методов	Ключевая идея	Преимущества	Основные ограничения
Методы на основе масок	Применение бинарных масок к параметрам сети для выделения подмножества весов под каждую задачу	Минимальное катастрофическое забывание; не требует изменения архитектуры сети	Требует знания идентификатора задачи; необходимо несколько проходов вперед с разными масками; неэффективно при большом числе задач
Методы расширения архитектуры	Динамическое добавление новых нейронов или слоев при поступлении новых данных	Адаптация емкости модели под сложность данных; сохранение ранее изученных представлений	Риск неконтролируемого роста сети; сложность определения момента и объема расширения; требуется дообучение
Методы регуляризации	Добавление штрафного члена к функции потерь, ограничивающего изменение важных для старых задач весов	Не требует хранения данных; вычислительно эффективен; не меняет архитектуру	Требует знания идентификатора задачи; сложность настройки гиперпараметров; снижает пластичность
Псевдо-репетиционные методы	Хранение или генерация репрезентативных примеров из предыдущих задач для повторного воспроизведения при обучении новым	Высокая точность; не требует идентификатора задачи; работает в сценарии Class-IL	Затраты на хранение данных или генерацию примеров; проблемы конфиденциальности; вычислительные затраты
Онлайн-методы	Последовательное обновление модели по одному примеру или мини-батчу без хранения истории	Высокая скорость обучения; минимальные требования к памяти; работа в реальном времени	Накапливание ошибок; переобучение при длительном функционировании; чувствительность к порядку поступления данных

Таблица 1 показывает главные тенденции и ограничения существующих подходов к инкрементному обучению. Методы на основе масок и расширения

архитектуры обеспечивают высокую стабильность за счет изоляции параметров или контролируемого роста модели, но применимы только при известном идентификаторе задачи (Task-IL) и грозят неконтролируемым усложнением архитектуры.

Методы регуляризации, такие как EWC и SI, дают компромиссное решение: хранить исторические данные им не нужно. Но их эффективность критически зависит от стационарности признакового пространства. При радикальном изменении распределения входных данных матрица Фишера и интегральные оценки важности весов теряют информативность, и модель застревает в конфигурации, оптимальной для неактуального распределения.

Псевдорепетиционные методы показывают высокую точность и применимы в самом сложном сценарии Class-IL, но требуют значительных ресурсов на хранение или генерацию данных. Кроме того, качество псевдопримеров сильно зависит от стабильности распределения: при резком изменении статистических свойств входных сигналов синтетические данные могут стать нерепрезентативными, и консолидация знаний идет хуже.

Онлайн-методы предъявляют минимальные требования к памяти и быстро адаптируются, но из-за накопления ошибок и чувствительности к порядку данных подходят лишь для задач с низкой сложностью и коротким горизонтом обучения.

Рассмотренные бионические подходы к моделированию «сна» в нейронных сетях [37–40] тоже не решают проблему полностью. Как показано выше, они направлены главным образом на стабилизацию модели через разреживание связей или подавление шумов и не решают задачу активной консолидации знаний с сохранением способности к дальнейшему инкрементному обучению.

Ни один из существующих методов не отвечает в полной мере всем требованиям к системам инкрементного обучения в нестационарных средах. Чтобы снять это противоречие, нужен гибридный подход, сочетающий достоинства аналитического инкрементного обучения с механизмами периодической консолидации знаний. Для него требуется базовая архитектура, допускающая

аналитическое вычисление параметров; она подробно рассматривается в следующем разделе.

1.3 Машина экстремального обучения как основа быстрого инкрементного обучения

В качестве базовой архитектуры для метода инкрементного обучения в настоящей работе выбрана машина экстремального обучения (Extreme Learning Machine, ELM), предложенная Г.-Б. Хуангом в 2006 году [49]. Это нейронная сеть прямого распространения с одним скрытым слоем (Single-hidden Layer Feedforward Network, SLFN). Веса входного слоя и смещения скрытых нейронов в ней генерируются случайно и фиксируются, а веса выходного слоя вычисляются аналитически как решение системы линейных уравнений (Рисунок 3). Обучение сводится к однократному матричному обращению, поэтому скорость обучения экстремально высока по сравнению с градиентными методами.

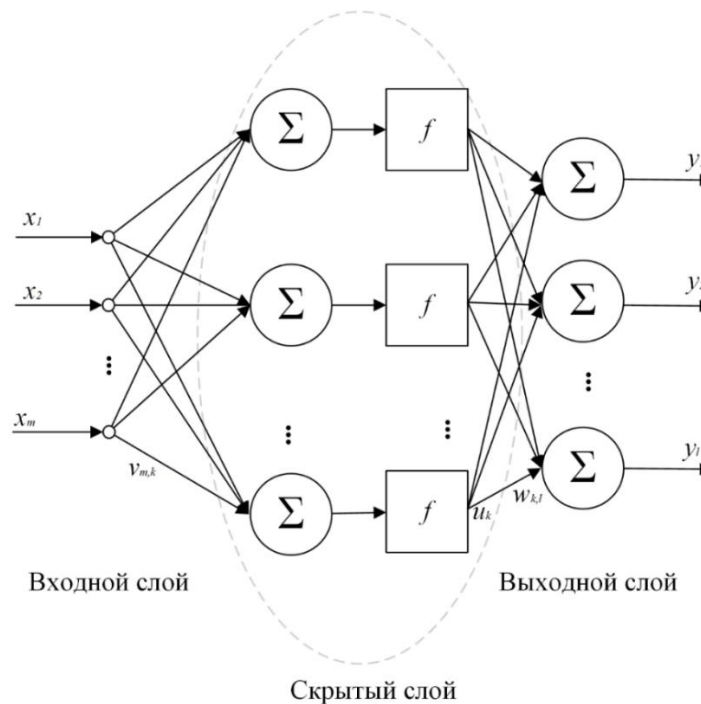


Рисунок 3 – Архитектура ELM

Согласно [49], обучение ELM проходит в два этапа. Сначала случайно формируются входные веса и смещения каждого нейрона скрытого слоя, и входные данные отображаются в пространство признаков нелинейной функцией активации. Затем выходные веса вычисляются как решение системы линейных уравнений. Это опирается на теорему об универсальной аппроксимации: нейронная сеть с одним скрытым слоем и достаточным числом нейронов способна аппроксимировать любую непрерывную функцию на компактном множестве с произвольной точностью [50].

Идею случайных весов между входным и скрытым слоями высказывали и раньше: в работах по радиальным базисным функциям (RBF) [51], нейронным сетям со случайными весами [52] и сетям случайных векторных функциональных связей (RVFL) [53; 54]. Однако именно Г.-Б. Хуанг систематизировал этот подход и доказал его теоретическую состоятельность, что и обеспечило широкое распространение ELM.

Ключевые преимущества ELM, определившие его выбор базовой архитектурой для разрабатываемых методов инкрементного обучения, следующие.

Во-первых, вычислительная эффективность: обучение требует только одного матричного обращения, тогда как градиентным методам нужно множество итераций. Для быстрой адаптации к потоковым данным это критически важно.

Во-вторых, в ELM нет проблемы локальных минимумов, характерной для градиентных методов, поскольку выходные веса находятся аналитически как решение задачи минимизации квадратичной функции потерь. Решение единственно, поэтому результаты детерминированы и воспроизводимы при одинаковых начальных условиях.

Для повышения устойчивости и обобщающей способности в ELM применяют L2-регуляризацию (регуляризацию Тихонова) [55]: к функции потерь добавляется штрафной член, пропорциональный квадрату нормы весов. Коэффициент регуляризации λ задает компромисс между точностью аппроксимации и обобщающей способностью модели [56].

Позже были предложены расширения базовой ELM: многоуровневый ELM (ML-ELM) [57] и иерархический ELM (H-ELM) [58], которые используют автоэнкодеры для многоуровневого извлечения признаков, а также подходы на основе метаэвристических алгоритмов для оптимизации входных весов и структуры сети [59; 60]. Эти модификации выходят за рамки настоящего исследования: они рассчитаны на пакетное обучение и не решают проблему инкрементного обновления в потоковых сценариях.

У базовой ELM есть ограничения, существенные для инкрементного обучения [61]. Во-первых, случайная генерация входных весов не гарантирует их оптимальности для конкретной задачи, из-за чего точность может снижаться. Во-вторых, нет стандартного способа определить необходимое число скрытых нейронов. В-третьих, и это самое существенное, в базовой ELM нет встроенных механизмов защиты от катастрофического забывания, а стандартная процедура обучения не позволяет дообучать сеть на новых данных без полного пересчета матрицы выходных весов. Наконец, данные реального времени часто зашумлены, поэтому их необходимо предварительно обрабатывать.

В итоге ELM дает эффективную вычислительную основу для быстрого инкрементного обучения, но фиксация случайных весов скрытого слоя и отсутствие механизмов сохранения знаний требуют специализированных модификаций для потокового поступления данных. Существующие инкрементные модификации ELM, направленные на преодоление этих ограничений, анализируются в следующем разделе.

1.4 Анализ инкрементных версий машины экстремального обучения

Как показано в предыдущем разделе, у базовой машины экстремального обучения есть преимущества, которые делают ее привлекательной для инкрементного обучения: вычислительная эффективность, отсутствие локальных минимумов и аналитическое вычисление выходных весов. Однако стандартная

процедура обучения ELM не позволяет дообучать сеть при поступлении новых данных без полного пересчета матрицы выходных весов, поэтому напрямую применять ее в потоковых сценариях нельзя.

Чтобы преодолеть это ограничение, в литературе разработан ряд инкрементных модификаций ELM. Их можно классифицировать по принципу работы.

Наиболее известная и широко используемая из них: метод последовательного экстремального обучения (Online Sequential Extreme Learning Machine, OS-ELM), предложенный в 2006 году [62]. OS-ELM обновляет выходные веса модели при поступлении новых данных без хранения исторических обучающих примеров и без полного переобучения. Метод основан на рекуррентном алгоритме наименьших квадратов (Recursive Least Squares, RLS) [63], который позволяет инкрементно обновлять псевдообратную матрицу Мура-Пенроуза при добавлении новых наблюдений.

Обучение OS-ELM проходит в две фазы. На первой (инициализации) модель обучается на начальном наборе данных стандартной процедурой ELM. На второй (секвенциального обучения) при поступлении каждого нового батча выходные веса обновляются рекуррентно по формулам Шермана-Моррисона, что позволяет не вычислять псевдообратную матрицу заново [64].

При высокой вычислительной эффективности у OS-ELM есть существенные ограничения. Во-первых, метод не решает проблему катастрофического забывания: рекуррентное обновление не содержит механизмов защиты ранее усвоенных паттернов, поэтому при обновлении весов на новых данных модель теряет знания о предыдущих распределениях.

Во-вторых, OS-ELM чувствителен к порядку поступления данных и к размеру батча: малые батчи делают оценки нестабильными, а большие увеличивают вычислительные затраты.

В-третьих, при длительной работе накапливается численная ошибка рекуррентных вычислений, что может ухудшить точность модели.

Альтернативный подход к инкрементному обучению ELM: метод наименьших квадратов с рекуррентным обновлением (RLS-ELM) [65], который модифицирует процедуру OS-ELM для данных, поступающих по одному примеру. Требования к памяти у RLS-ELM минимальны, поскольку он не хранит даже текущий батч, но численная ошибка в нем накапливается еще сильнее, а к шуму во входных данных он чувствителен.

Чтобы преодолеть ограничения фиксированной архитектуры базовой ELM, разработаны методы динамического изменения структуры сети. Растущая машина экстремального обучения (Growing ELM) [66] постепенно добавляет новые нейроны в скрытый слой при поступлении новых данных, если текущая архитектура недостаточно точно аппроксимирует. Критерием добавления может служить превышение порога ошибки или снижение точности на валидационной выборке.

Обрезка машины экстремального обучения (Pruning ELM) [67], напротив, удаляет из скрытого слоя наименее значимые нейроны, что снижает вычислительную сложность и улучшает обобщающую способность модели.

Комбинированный подход Growing-Pruning ELM [68] сочетает оба механизма: сеть расширяется при необходимости и сокращается, избавляясь от избыточных нейронов.

Методы динамической архитектуры подстраивают емкость модели под сложность данных: сеть увеличивает число нейронов, когда распределение входных данных усложняется, и уменьшает его, когда оно стабилизируется. Но у этих методов есть ограничения. Во-первых, критерии добавления и удаления нейронов требуют настройки гиперпараметров (порогов ошибки, коэффициентов значимости), и автоматически применять такие методы трудно. Во-вторых, при добавлении нейронов выходные веса всех существующих нейронов приходится пересчитывать, из-за чего падает вычислительная эффективность. В-третьих, они не решают проблему катастрофического забывания: при удалении нейронов информация о ранее изученных паттернах может быть утрачена.

Для борьбы с катастрофическим забыванием в ELM предложены специализированные подходы, заимствующие идеи из методов регуляризации и воспроизведения опыта. В работе [69] предложена регуляризованная версия ELM, где к функции потерь добавляется штрафной член, ограничивающий изменение весов; это частично сохраняет знания о предыдущих задачах. Другой подход сохраняет подмножество репрезентативных примеров из предыдущих задач в буфере памяти и подмешивает их в обучающий набор при обновлении модели на новых данных [70]. Он аналогичен методу Experience Replay для градиентных сетей и частично компенсирует забывание. Третье решение: генеративные модели синтезируют псевдопримеры, отражающие распределение предыдущих задач. Это избавляет от хранения реальных данных и снимает проблемы конфиденциальности, но требует обучения дополнительного генератора и увеличивает вычислительные затраты [71].

У всех этих подходов те же ограничения, что и у соответствующих методов в градиентных сетях [72]. Регуляризационные подходы зависят от стационарности признаков пространства и теряют эффективность при радикальном изменении распределения данных [73]. Методы с буфером памяти требуют ресурсов памяти, порождают проблемы конфиденциальности, а их эффективность снижается, если сохраненные примеры нерепрезентативны [74]. Генеративные подходы требуют обучения сложных моделей и увеличивают вычислительные затраты [75]. Кроме того, ни один из этих методов не использует в полной мере преимущества аналитической природы ELM для разработки собственных механизмов консолидации знаний, а лишь напрямую заимствует идеи градиентных методов.

В Таблице 2 представлено обобщенное сравнение рассмотренных инкрементных версий ELM по ключевым характеристикам.

Таблица 2 – Сравнительный анализ инкрементных версий ELM

Метод	Способ обновления	Основные ограничения
OS-ELM	RLS-обновление выходных весов	Катастрофическое забывание, чувствительность к порядку данных, накопление ошибок
RLS-ELM	RLS-обновление по одному примеру	Чувствительность к шуму, накопление ошибок
Growing ELM	Добавление нейронов по критерию ошибки	Критерий остановки, неконтролируемый рост
Pruning ELM	Удаление нейронов по значимости	Риск потери знаний, сложность критерия
Регуляризованные ELM	Штраф за изменение весов	Зависит от стационарности признаков
ELM с буфером памяти	Хранение и повторное использование примеров	Требует памяти, проблемы конфиденциальности
ELM с генерацией данных	Синтез псевдопримеров	Требует обучения генератора, вычислительные затраты

Таблица 2 позволяет систематизировать ограничения существующих инкрементных модификаций ELM.

Последовательные методы (OS-ELM, RLS-ELM) быстро обновляют модель, но не решают проблему катастрофического забывания, и при длительной работе в них накапливается численная ошибка. Они подходят только там, где забывание не критично или данные стационарны.

Методы динамической архитектуры (Growing, Pruning, Growing-Pruning ELM) подстраивают емкость модели под сложность данных, но требуют настройки гиперпараметров, снижают вычислительную эффективность при изменении структуры сети и не решают проблему забывания при удалении нейронов.

Подходы к предотвращению забывания в ELM (регуляризованные версии, методы с буфером памяти, генеративные подходы) заимствуют идеи градиентных методов и наследуют их ограничения: зависимость от стационарности признакового пространства, требования к памяти, вычислительную сложность генеративных моделей, снижение эффективности при нерепрезентативных сохраненных данных. При этом ни один из них не использует специфические свойства ELM, то есть аналитическое вычисление весов и случайную фиксацию

скрытого слоя, для оригинальных механизмов консолидации знаний, а лишь адаптирует методы, созданные для градиентных сетей.

Проведенный анализ показывает: ни одна из рассмотренных модификаций не удовлетворяет одновременно трем ключевым требованиям к системам инкрементного обучения в нестационарных средах на ресурсограниченных устройствах [76]. Эти требования: высокая скорость инкрементного обновления, защита от катастрофического забывания и адаптация архитектуры к меняющейся сложности данных. Поэтому необходимы новые алгоритмы инкрементного обучения на основе ELM. Они должны сочетать вычислительную эффективность аналитического обновления с механизмами периодической консолидации знаний [77] и быть устойчивыми к нестационарности признакового пространства без хранения больших объемов исторических данных. Решению этой задачи посвящена глава 2 настоящей диссертационной работы.

Выводы по главе 1

В первой главе проведен анализ современного состояния проблемы инкрементного обучения нейронных сетей в нестационарных средах. Рассмотрены особенности работы нейронных сетей при потоковом поступлении данных, классифицированы существующие подходы к инкрементному обучению и проанализированы их ограничения. На основе анализа сформулированы следующие основные выводы.

1. Классические методы обучения нейронных сетей плохо приспособлены к потоковому поступлению данных. Традиционная парадигма машинного обучения исходит из стационарности данных и их полной доступности до начала обучения, поэтому не в полной мере отвечает требованиям систем, работающих в динамически изменяющихся средах. Пакетное обучение сталкивается с ростом вычислительной сложности, увеличением требований к памяти и концептуальной устарелостью модели при длительных интервалах между переобучениями.

Поэтому переход к инкрементному обучению остается одним из перспективных направлений для систем с непрерывными потоками данных.

2. Ключевые проблемы инкрементного обучения: нестационарность данных и катастрофическое забывание. В реальной эксплуатации статистические свойства данных меняются во времени. Концептуальный дрейф и нестационарность признакового пространства создают дополнительные сложности для алгоритмов машинного обучения. Адаптируясь к новым данным, модель может столкнуться с катастрофическим забыванием, то есть со снижением качества на ранее изученных задачах. Это явление тесно связано с дилеммой стабильности-пластичности, которая требует баланса между способностью усваивать новую информацию и сохранением ранее сформированных знаний.

3. Градиентные методы ограничено применимы в задачах инкрементного обучения. Несмотря на высокую эффективность в стационарных условиях, градиентные методы, включая метод обратного распространения ошибки, в потоковых сценариях имеют особенности, которые нужно учитывать. В них нет встроенных механизмов разделения знаний, что может приводить к катастрофическому забыванию, вычислительная сложность высока, результат зависит от порядка данных и выбора гиперпараметров, а также возникает проблема затухающих и взрывающихся градиентов. Чтобы преодолеть эти ограничения, нужны специализированные подходы к обновлению параметров модели.

4. У каждого из существующих семейств методов инкрементного обучения есть сильные стороны и ограничения. Проанализированы пять основных семейств: методы на основе масок, методы расширения архитектуры, методы регуляризации, псевдорепетиционные методы и онлайн-методы. Методы регуляризации эффективны при стационарном распределении признаков, но их эффективность может снижаться при изменении статистических свойств данных. Псевдорепетиционные методы дают высокую точность, но требуют ресурсов на хранение или генерацию данных. Онлайн-методы предъявляют минимальные требования к памяти, но могут быть чувствительны к порядку поступления данных.

и накоплению ошибок. Выбор подхода зависит от условий применения и требований к системе.

5. Бионические подходы к моделированию «сна» интересны, но пока ограничены. Подходы, вдохновленные процессами консолидации памяти в биологических системах, дают альтернативный взгляд на решение дилеммы стабильности-пластичности. Существующие реализации в основном стабилизируют модель через разреживание связей или подавление шумовых паттернов, а фазу «сна» часто считают конечной, без последующего инкрементного обновления. Дальнейшее развитие таких подходов может быть связано с механизмами, которые не только стабилизируют модель, но и активно консолидируют знания, сохраняя способность к обучению на новых данных.

6. Машина экстремального обучения перспективна как основа для методов инкрементного обучения. Для задач инкрементного обучения ELM интересна вычислительной эффективностью, отсутствием проблемы локальных минимумов и аналитическим вычислением выходных весов, на котором можно построить инкрементное обновление рекуррентным методом наименьших квадратов. Но у базовой ELM есть ограничения: фиксация случайных весов скрытого слоя и отсутствие механизмов защиты от катастрофического забывания. Поэтому нужны специализированные модификации.

7. Существующие инкрементные модификации ELM ограничены. Последовательные методы (OS-ELM, RLS-ELM) быстро обновляют модель, но не защищают от катастрофического забывания и могут накапливать численную ошибку. Методы динамической архитектуры позволяют адаптировать емкость модели, но забывание не устраняют. Подходы к предотвращению забывания (регуляризованные версии, методы с буфером памяти, генеративные подходы) заимствуют идеи градиентных методов и наследуют их ограничения, а специфические свойства ELM для механизмов консолидации знаний используют не в полной мере.

Анализ показывает: при всем разнообразии подходов к инкрементному обучению каждый из них имеет ограничения, и ни один не решает проблему

комплексно в условиях нестационарных потоковых данных на ресурсоограниченных устройствах. Поэтому целесообразно разработать алгоритмы, сочетающие вычислительную эффективность аналитического обновления на основе ELM с механизмами периодической консолидации знаний.

Результаты анализа ограничений классических методов обучения нейронных сетей при потоковом поступлении данных получены автором лично и были частично опубликованы, в т. ч. в соавторстве (вклад автора более 80%) в [22; 80; 81; 110].

Для достижения этой цели в диссертационной работе решаются следующие задачи:

1. Разработать алгоритм двухфазной консолидации знаний нейронной сети в режиме «сна», в котором настройка весов выполняется методом обратного распространения ошибки, а выходные веса аналитически пересчитываются на основе инкрементной машины экстремального обучения с использованием псевдопримеров, генерируемых самой сетью.
2. Разработать алгоритм посменного инкрементного обучения нейронной сети с циклическим чередованием состояний функционирования и «сна».
3. Разработать алгоритм динамической адаптации архитектуры нейронной сети, включающий добавление нейронов по критерию ошибки и их функциональное разделение на основе метрик обученности и упругости.
4. Провести экспериментальную верификацию разработанных алгоритмов на синтетических данных с контролируемой сложностью для изолированной оценки вклада каждого компонента.
5. Апробировать разработанные алгоритмы на задаче распознавания физической активности человека по данным инерциальных сенсоров и сравнить их эффективность с методами инкрементного обучения.

Решение перечисленных задач представлено в следующих главах диссертационной работы.

ГЛАВА 2. РАЗРАБОТКА АЛГОРИТМОВ ПОСМЕННОГО ИНКРЕМЕНТНОГО ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ НА ОСНОВЕ МАШИНЫ ЭКСТРЕМАЛЬНОГО ОБУЧЕНИЯ

2.1 Разработка модели данных

Для предварительного исследования разрабатываемых алгоритмов инкрементного обучения необходима модель данных, позволяющая контролировать сложность аппроксимируемой зависимости и воспроизводить эксперименты при идентичных начальных условиях.

Согласно теореме Вейерштрасса, любая непрерывная на отрезке $[a, b]$ функция $f(x)$ может быть равномерно приближена полиномом $G(x)$ некоторой степени m с любой наперед заданной точностью [78]. Это свойство позволяет использовать полиномиальные зависимости в качестве базовой модели для апробации алгоритмов машинного обучения, поскольку они обеспечивают контролируемую сложность и интерпретируемость результатов.

Таким образом, в качестве формы математической модели можно выбрать уравнение полинома степени m в каноническом виде:

$$f(x) = G(x) = c_0 + c_1x + c_2x^2 + \dots + c_mx^m. \quad (1)$$

Для генерации уравнений канонического полинома разных порядков, найдем его параметры с помощью полиномиальной регрессии.

Полиномиальная регрессия означает приближение данных (x_i, y_i) полиномом m -й степени. При $m=1$ полином является прямой линией, при $m=2$ – параболой, при $m=3$ – кубической параболой и т.д. Для построения регрессии полиномом m -й степени необходимо наличие, по крайней мере, $m + 1$ точек данных.

В качестве данных для обучения нейронной сети можно использовать матрицу-столбец X , содержащую 10 случайных чисел, имеющих равномерное распределение, в котором n и 1 являются граничными точками интервала, и

матрицу-столбец Y , значения элементов которой являются расчетными значениями функции $G(X)$.

Таким образом, разработана модель данных, которая будет использоваться для оценки качества работы алгоритмов распознавания образов на основе машины для экстремального обучения.

2.2 Математическая модель машины экстремального обучения

Машина экстремального обучения представляет собой нейросетевую архитектуру с одним скрытым слоем, в которой веса входного слоя и смещения скрытых нейронов генерируются случайным образом, а веса выходного слоя вычисляются аналитически.

Для улучшения характеристик ELM в работе [79] была предложена его регуляризованная версия, т.е. авторы определяют диапазоны входных весов и смещений в соответствии с выбранной функцией активации. Функции активации и диапазоны их параметров представлены в Таблице 3.

Таблица 3 – Функции активации и их параметры

Функция активации	Диапазон случайных весов, v_i	Диапазон смещений, b_i
Сигмовидная функция $\frac{1}{1 + \exp(-(v_i x + b_i))}$	$[-A; A]$	$\left[-\ln\left(\frac{q}{1-q}\right) - v_i, -\ln\left(\frac{1-q}{q}\right) \right]$, если $v_i \geq 0$ $\left[-\ln\left(\frac{q}{1-q}\right), -\ln\left(\frac{1-q}{q}\right) - v_i \right]$, если $v_i < 0$
Функция Гаусса $\exp(-(v_i x + b_i)^2)$	$[0; A]$	$[-v_i; 0]$
Функция Фурье $\sin(v_i x + b_i)$	$[0; A]$	$[0; 2\pi]$

Примечание: $A > 0$, $q \in [0,5; 1]$

Матрица-строка Y значений на выходе вычисляется как:

$$Y = f(X \cdot V) \cdot W, \quad (2)$$

где f – функция активации,

$X = (x_1 \ x_2 \ \dots \ x_m)$ – матрица-строка значений на входе,

V – матрица случайных весов, соединяющих входной и скрытый слой,

W – матрица вычисляемых весов, соединяющих скрытый и выходной слой.

В случае, если изначально известны все образы, то $X = \begin{pmatrix} x_{11} & \dots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nm} \end{pmatrix}$ – матрица значений на входе, $Y = \begin{pmatrix} y_{11} & \dots & y_{1l} \\ \vdots & \ddots & \vdots \\ y_{n1} & \dots & y_{nl} \end{pmatrix}$ – матрица значений на выходе.

Если $f(X \cdot V)$ обозначить через U , то формула примет вид:

$$Y = U \cdot W, \quad (3)$$

где U – матрица выходных значений нейронов скрытого слоя.

В зависимости от начальных условий U может быть либо матрицей-строкой $U = (u_1 \ u_2 \ \dots \ u_k)$ выходных значений нейронов скрытого слоя, либо

матрицей $U = \begin{pmatrix} u_{11} & \dots & u_{1k} \\ \vdots & \ddots & \vdots \\ u_{n1} & \dots & u_{nk} \end{pmatrix}$ выходных значений нейронов скрытого слоя.

На втором этапе происходит обучение сети путем вычисления матрицы выходных весов нейронов скрытого слоя W :

$$W = (U^T U)^{-1} U^T Y = U^+ Y, \quad (4)$$

где $U^+ = (U^T U)^{-1} U^T$ представляет собой псевдообратную матрицу (обобщенную обратную матрицу Мура-Пенроуза) для U .

Обозначим $U^T U$ через R , $U U^T$ через $\tilde{R}$, а $U^T Y$ через r , тогда

$$W = R^{-1} r. \quad (5)$$

Для случаев, когда количество обучающих данных очень велико или мало, например, намного больше или меньше, чем размерность пространства признаков в работах [45; 46] предлагают добавлять коэффициент регуляризации. Решение регуляризованного ELM может быть получено для первого случая, как:

$$W = \left(\frac{I}{\delta} + R \right)^{-1} r, \quad (6)$$

и для второго случая:

$$W = U^T \left(\frac{I}{\delta} + \tilde{R} \right)^{-1} Y, \quad (7)$$

где I – единичная матрица,

δ – коэффициент регуляризации.

Как видно из уравнений матрица выходных весов нейронов скрытого слоя зависит от правильного выбора коэффициента регуляризации.

Таким образом, коэффициент регуляризации необходимо подобрать некоторым оптимальным способом. Очевидно, что им будет некоторое промежуточное значение, устанавливающее определенный компромисс между приемлемой обусловленностью и близостью к исходной задаче.

В основном, существующие алгоритмы выбора коэффициента регуляризации можно разделить на две группы: алгоритмы, использующие априорную информацию (принцип невязки, условие оптимальности); алгоритмы не требующие знания какой-либо дополнительной информации (квазиоптимальный способ, метод перекрестной значимости, метод L-кривой).

– способ невязки. Невязка – величина ошибки (расхождения) приближенного равенства. Принцип невязки был предложен В.А. Морозовым, а затем широко использовался для выбора параметра регуляризации. Данный способ подразумевает выбор параметра регуляризации, с которым невязка равна априорной оценке погрешностей задания входных данных.

– квазиоптимальный способ. Квазиоптимальный способ определения параметра регуляризации не требует знания погрешности правой части. Согласно этому способу в качестве искомого значения выбирают то, для которого поправка $\|y_{\delta_i} - y_{\delta_{i-1}}\|_{L_2}$ принимает наименьшее значение.

– метод L-кривой. Суть данного метода состоит в вычислении для набора значений параметра регуляризации кривизны линии и поиске точки с максимальной кривизной. Построенный для случая дискретной некорректной

задачи в логарифмических координатах график приобретает характерную L-образную форму с хорошо различимым углом, разделяющим вертикальную и горизонтальную части кривой. Таким образом, L-кривая четко показывает оптимум между нормой решения (т. е. его близостью к точному решению) и его стабилизацией (сглаженностью).

Таким образом, математическая модель ELM позволяет эффективно решать задачи классификации и регрессии за счет аналитического вычисления выходных весов. Выбор функции активации и коэффициента регуляризации является ключевым фактором, определяющим качество обобщения и численную устойчивость модели.

В дальнейшем, для организации инкрементного обучения, вместо хранения и обращения матрицы $R = U^T U$ на каждом шаге при поступлении новых данных, будет использоваться матрица $P = R^{-1}$, представляющая собой обратную корреляционную матрицу. В теории рекуррентного метода наименьших квадратов данная матрица интерпретируется как матрица ковариации ошибки оценки весов и допускает рекурсивное обновление при поступлении каждого нового примера без необходимости полного пересчета.

Такой переход позволяет применять рекуррентное обновление по формулам Шермана-Моррисона, что является вычислительной основой разработанного в разделе 2.3 алгоритма инкрементного обучения с мемоизацией матриц системы нормальных уравнений.

2.3 Алгоритм инкрементного последовательного и пакетного обучения нейронных сетей на основе машины экстремального обучения с мемоизацией матриц системы нормальных уравнений

Машина экстремального обучения является методом пакетного обучения, который предполагает наличие полной обучающей выборки перед началом

процесса обучения. Однако во многих реальных задачах обучающие данные поступают последовательно в виде потока данных, и их количество может быть непостоянным.

Инкрементное обучение представляет собой метод обучения, в котором модификация синаптических связей сети (весов и порогов) происходит после предъявления каждого образа из обучающей выборки нейронной сети.

Для реализации инкрементного обучения нейронной сети на базе описанной в п. 2.2 математической модели ELM предлагается применить рекуррентный метод наименьших квадратов [63], что позволяет обрабатывать данные последовательно без необходимости полного переобучения при поступлении каждого нового примера.

Адаптивная коррекция всех весов производится согласно формуле Шермана-Моррисона [64]:

$$P_{k+1} = P_k - \frac{P_k U_{k+1} U_{k+1}^T P_k}{1 + U_{k+1}^T P_k U_{k+1}}, \quad (8)$$

$$w^{(k+1)} = w^{(k)} + P_{k+1} U_{k+1} (Y_{k+1}^T - U_{k+1}^T w^{(k)}). \quad (9)$$

Адаптивная коррекция всех весов по формуле Шермана-Моррисона – это метод обновления весов нейронной сети при инкрементном обучении, который позволяет учитывать поступающие новые данные без необходимости повторного обучения всей сети.

Формула (8) используется для обновления матрицы ковариации ошибки P . На каждом шаге $k + 1$ можно обновить матрицу ковариации ошибки, используя матрицу ковариации ошибки на предыдущем шаге P_k , текущий входной образ U_{k+1} и коэффициент коррекции, вычисленный с помощью формулы $1 + U_{k+1}^T P_k U_{k+1}$.

Формула (9) используется для обновления весов сети w . На каждом шаге $k + 1$ можно обновить веса сети, используя матрицу ковариации ошибки P_{k+1} , чтобы скорректировать веса сети, разность между текущим выходным образом Y_{k+1} и выходом, рассчитанным на предыдущем шаге с помощью весов $w^{(k)}$ и входного образа U_{k+1} . Здесь $w^{(k)}$ обозначает веса сети на предыдущем шаге k , а $w^{(k+1)}$ – на текущем шаге $k + 1$.

Формулы Шермана-Моррисона используются в рекуррентном методе наименьших квадратов для адаптивной коррекции всех весов нейронной сети. Как видно из формулы (9), выходные веса рекурсивно обновляются на основе промежуточных результатов в последней итерации и вновь поступивших данных, которые могут быть отброшены после обучения, что приводит к снижению вычислительной сложности. Также данный алгоритм не требует повторного обучения всякий раз, когда появляются новые образы обучающей выборки.

Ключевым элементом используемого алгоритма является мемоизация матрицы P системы нормальных уравнений. Под мемоизацией понимается сохранение и рекурсивное обновление матрицы P_k при поступлении новых данных без ее полного пересчета. Это отличает используемый алгоритм от классических методов ELM, требующих повторного обращения матрицы при дополнении обучающей выборки.

В рекуррентном методе наименьших квадратов нейронная сеть обучается на потоке данных, где каждый новый образ последовательности данных подается на вход сети и обновляет веса сети на каждом шаге. В вычислительной структуре рекуррентного метода наименьших квадратов нет цепочек, приводящих к останову (делению на ноль). В качестве начальных условий примем, что матрица W формируется случайно, а матрица P – это результат отношения единичной матрицы к коэффициенту регуляризации.

Для обоснования эффективности алгоритма инкрементного обучения проведем сравнительный анализ его вычислительной сложности по времени и объему памяти относительно классического пакетного подхода. Обозначим N – общее количество обучающих примеров, d – размерность входного признакового пространства, L – количество нейронов скрытого слоя.

В классической пакетной машине экстремального обучения (п. 2.2) вычисление матрицы выходных весов W требует обращения матрицы $R = U^T U$ размером $L \times L$. Сложность обращения матрицы $O(L^3)$, а формирование матрицы Грама $U^T U$ требует $O(N L^2)$ операций. Общая временная сложность

пакетного ELM составляет $O(N L^2 + L^3)$. Вся обучающая выборка хранится в оперативной памяти, поэтому пространственная сложность $O(N d)$.

В предложенном алгоритме инкрементного обучения (п. 2.3) используются мемоизация матрицы P и рекуррентное обновление весов по формулам Шермана-Моррисона. Для одного поступившего примера поправка матрицы P^{k+1} и обновление вектора весов $w^{(k+1)}$ сводятся к матричным умножениям для векторов размерности L и стоят $O(L^2)$. Значит поток из N обрабатывается за $O(N L^2)$ операций, и матрицу R не нужно обращать заново на каждом шаге.

Главное преимущество алгоритма – существенное снижение требований к памяти. Матрица P размером $L \times L$ и веса W размером $L \times 1$ обновляются рекурсивно, поэтому хранить исторические данные не нужно. Пространственная сложность метода $O(L^2 + dL)$, поэтому он подходит для условий ограниченных вычислительных ресурсов и потокового поступления данных, когда хранить всю выборку из N примеров физически невозможно.

Такое обновление весов позволяет сети быстро адаптироваться к новым данным без полного повторного обучения. Но у него есть ограничения: оно может оказаться неэффективным, например, если структура новых данных заметно отличается от прежней.

2.4 Математическая модель метода обратного распространения ошибки

Метод обратного распространения ошибки определяет стратегию подбора весов многослойной сети с применением градиентных методов оптимизации.

Обучение сети с помощью метода обратного распространения ошибки может длиться достаточно долго, но при этом уже обученная сеть способна быстро вычислять результаты по полученным данным. В настоящее время считается одним из наиболее эффективных методов обучения нейронной сети [19].

Метод является итеративным и использует принцип обучения «по шагам», когда веса нейронов сети корректируются после подачи на ее вход одного обучающего примера.

Основной целью метода обратного распространения ошибки применительно к нейронным сетям является оптимизация весов связей между нейронами для того, чтобы проектируемая нейронная сеть смогла обучиться правильно отображать произвольные входные данные в нужные значения выходов.

Метод обратного распространения ошибки имеет два потока обработки данных: прямой и обратный. В прямом потоке сигналы распространяются от входного слоя к выходному, рассчитываются выходные значения. В обратном потоке сигналы распространяются от выходов ко входам, распространяя при этом значения полученных ошибок обучения. В результате этого рассчитываются величины, в соответствии с которыми следует корректировать весовые коэффициенты.

Алгоритм обучения нейронной сети методом обратного распространения ошибки можно представить следующим образом:

Шаг 1. Задать значения весовых коэффициентов нейронной сети, скорость обучения, число итераций, погрешность вычисления величины ошибки.

Шаг 2. Подавать входные образы из обучающей выборки и в режиме прямого распространения входного образа вычислить выходные значения скрытых слоев и выходного слоя, используя формулу (10):

$$y_j = f\left(\sum w_{ij} \cdot x_i\right), \quad (10)$$

где индекс j характеризует нейроны следующего слоя по отношению к слою i .

Функция активации при обучении нейронной сети методом обратного распространения ошибки должна быть нелинейна и дифференцируема, например, можно использовать сигмоидальную функцию активации, показанную в формуле (11):

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (11)$$

Шаг 3. В режиме обратного распространения сигнала рассчитать ошибку для выходного слоя по формуле (12):

$$\delta_j^{(Q)} = f'(s) \cdot (y_j - d_j) \quad (12)$$

Шаг 4. В режиме обратного распространения сигнала скорректировать значения веса для выходного слоя, используя формулу (13):

$$\Delta w_{ij} = -\eta \cdot \delta^{(q+1)} \cdot y_i^{(q)}, \quad (13)$$

где q – текущий слой сети, отправляющий ошибку в обратном направлении, η – скорость обучения.

Шаг 5. В режиме обратного распространения сигнала рассчитать ошибки для всех остальных слоев по формуле (14):

$$\delta_j^{(q)} = f'(s) \cdot \sum \delta_k^{(q+1)} \cdot w_k^{(q+1)} \quad (14)$$

Шаг 6. В режиме обратного распространения сигнала скорректировать остальные веса нейронной сети, используя формулу (13).

Шаг 7. Вычислить ошибку сети по формуле (15): если значение ошибки сети больше допустимого, то перейти на шаг 2, иначе завершить обучение.

$$E = \frac{1}{2} \sum (y_i - d_i)^2 \quad (15)$$

Несмотря на высокую вычислительную производительность и простоту реализации, к недостаткам метода обратного распространения ошибки стоит отнести неопределенно долгий процесс обучения и возможность попадания в локальные минимумы функции ошибки.

2.5 Разработка алгоритма двухфазной консолидации знаний нейронной сети в режиме «сна»

Представленный в п. 2.3 алгоритм инкрементного обучения на основе ELM обеспечивает быструю обработку новых данных, однако при длительном функционировании возможно накопление ошибок и переобучение, проявляющееся в неконтролируемом росте значений весовых коэффициентов выходного слоя. Для

преодоления этого ограничения разработан алгоритм двухфазной консолидации знаний нейронной сети в режиме «сна», базирующийся на генерации псевдопримеров самой сетью и сочетающий метод обратного распространения ошибки с инкрементной машиной экстремального обучения (Рисунок 4).

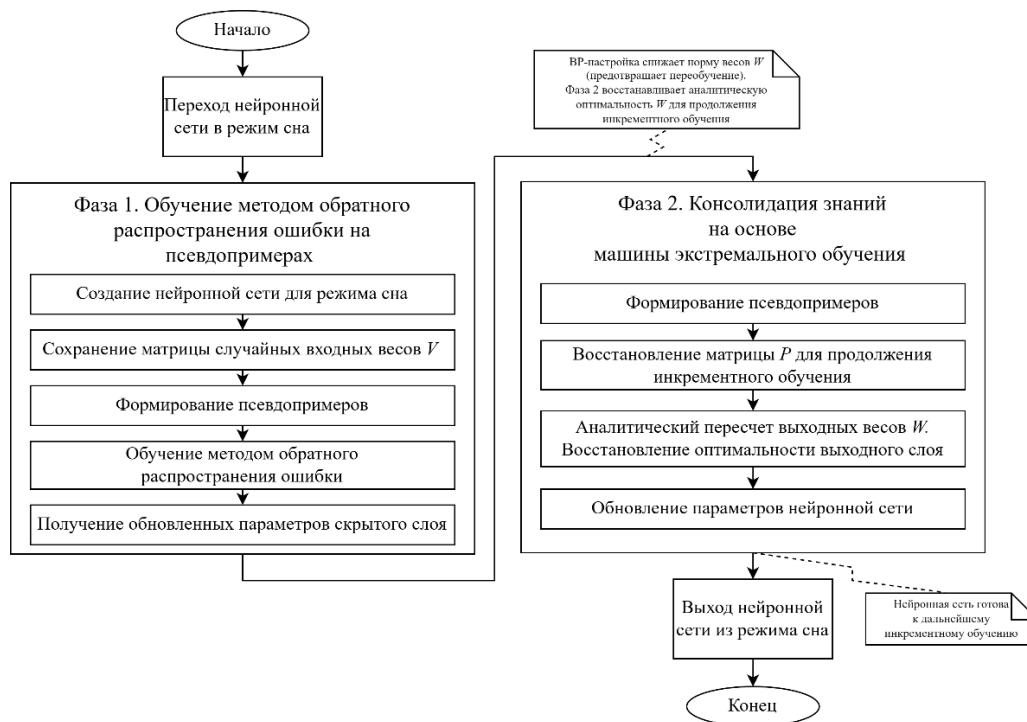


Рисунок 4 – Двухфазный алгоритм обучения в режиме «сна»

Под состоянием «сна» в настоящей работе понимается обучение нейронной сети на псевдопримерах, полученных с помощью нейронной сети из предыдущего состояния, с корректировкой всех весов сети. Состояние «сна» состоит из двух последовательных фаз [80; 81].

Первая фаза «сна»: обучение методом обратного распространения ошибки на псевдопримерах.

Шаг 1. Создание нейронной сети для обучения методом обратного распространения ошибки (создается необученная нейронная сеть, в которой случайным образом задаются значения весовых коэффициентов на скрытом слое).

Шаг 2. Сохранение значений матрицы случайных весов V , соединяющих входной и скрытый слои, полученных на этапе инициализации ELM.

Шаг 3. Формирование псевдопримеров с помощью нейронной сети из состояния функционирования. Для каждого класса c , представленного в буфере воспроизведения, вычисляются вектор математического ожидания μ_c и ковариационная матрица Σ_c признаков. Затем псевдопримеры генерируются выборкой из многомерного нормального распределения [82]:

$$x_{gen} \sim N(\mu_c, \Sigma_c + \varepsilon I), \quad (16)$$

где $\varepsilon = 10^{-6}$ – малая константа, обеспечивающая положительную определенность ковариационной матрицы.

Метки классов для сгенерированных примеров присваиваются по классу c . Так сохраняется многомерная структура распределения данных без хранения самих реальных примеров, что критически важно для соблюдения требований конфиденциальности при работе с пользовательскими данными.

Следует отметить ограничение алгоритма генерации псевдопримеров: данные инерциальных сенсоров могут иметь негауссовы распределения [12]. Во избежание деградации модели из-за артефактов генерации на границах классов, псевдопримеры используются исключительно в комбинации с реальными примерами из буфера воспроизведения, что позволяет сохранять нелинейные границы решений и использовать синтетические данные лишь для стабилизации ковариационной структуры памяти [21].

Шаг 4. Обучение нейронной сети методом обратного распространения ошибки. Обучение происходит в два прохода: прямого и обратного. При прямом проходе сигнал, подаваемый на вход нейронной сети, распространяется к выходам сети через все слои при фиксированных синаптических весах. Во время обратного прохода вычисляется сигнал ошибки δ , равный разности фактического и целевого выхода сети. Сигнал ошибки распространяется в обратном направлении, а синаптические веса корректируются: $V, W := \text{Backpropagation}[V, W]$.

В результате первой фазы значения весовых коэффициентов W становятся малыми, что предотвращает переобучение, однако полученная нейронная сеть,

обученная методом обратного распространения ошибки, не допускает инкрементного обучения.

Вторая фаза «сна» – консолидация знаний на основе машины экстремального обучения.

Шаг 5. Формирование псевдопримеров, полученных из первой фазы состояния «сна».

Шаг 6. Восстановление матрицы P машиной экстремального обучения. Матрица P необходима для того, чтобы сохранить знания о задачах, усвоенных в состоянии функционирования, и обеспечить возможность продолжения инкрементного обучения. Начальные значения матрицы W^* берутся из первой фазы состояния «сна».

Предложенная двухфазная процедура консолидации базируется на следующем теоретическом результате. Для регуляризованной ELM выходные веса W находятся как решение задачи минимизации квадратичной функции потерь с регуляризацией. При изменении входных весов скрытого слоя (первая фаза «сна») матрица выходов скрытого слоя U изменяется, что делает ранее вычисленные выходные веса W неоптимальными. Если не выполнить пересчет W , модель теряет способность точно воспроизводить ранее усвоенные зависимости, так как нарушается условие оптимальности.

Вторая фаза «сна» восстанавливает оптимальность решения, пересчитывая W на объединенном наборе реальных и псевдопримеров. Это обеспечивает согласованность входных и выходных весов после адаптации скрытого слоя.

Таким образом, двухфазное обучение в режиме «сна» позволяет, с одной стороны, предотвратить переобучение и стабилизировать весовые коэффициенты (первая фаза), а с другой – восстановить способность сети к инкрементному обучению и сохранить накопленные знания (вторая фаза). Отличительной особенностью предложенного алгоритма является последовательное применение метода обратного распространения ошибки и инкрементной машины экстремального обучения на псевдопримерах, генерируемых самой сетью, что исключает необходимость хранения реальных обучающих данных.

2.6 Разработка алгоритма посменного инкрементного обучения нейронной сети на основе машины экстремального обучения с циклическим чередованием состояний функционирования и «сна»

На основе предложенных в пп. 2.3 и 2.5 алгоритмов разработан алгоритм посменного инкрементного обучения нейронной сети, обеспечивающий непрерывное обучение в условиях длительного функционирования модели [80; 83]. В его основе лежат следующие принципы.

1. Нейронная сеть может находиться в двух состояниях (в состоянии функционирования и состоянии «сна»). В состоянии функционирования сеть выдает выходные данные (ответы) на входные данные (запросы) для решения поставленной задачи интеллектуального анализа или дообучается в случае наличия обучающих примеров. Обучение выполняется последовательно по одному примеру или последовательно пакетами примеров.

2. Обучение в состоянии функционирования построено на основе машины экстремального обучения, в котором происходит корректировка весов только выходного слоя нейронов сети.

3. Для реализации инкрементного обучения в машине экстремального обучения применяется мемоизация матрицы P .

4. В случае переобучения (возникновения больших по модулю значений весовых коэффициентов на выходном слое) сеть переходит в состояние «сна».

5. В состоянии «сна» выполняется передача части значений весов выходного слоя на нижние слои нейронов сети (веса с выходного слоя «растекаются» на нижние слои с помощью метода обратного распространения ошибки).

6. Для обучения сети в состоянии «сна» используются ответы, которые выдает та же сеть на сгенерированные случайным образом запросы (иначе говоря, сеть заново обучает сама себя, но уже другим методом).

7. Состояния функционирования и «сна» поочередно сменяют друг друга, длительность и момент смены которых зависят от значений весовых коэффициентов в нейронах сети.

На основании предложенных принципов алгоритм посменного инкрементного обучения состоит из следующих шагов (Рисунок 5).

Шаг 1. Обучение сети в состоянии функционирования (обучение нейронной сети новым задачам с корректировкой весов только выходного слоя сети).

Шаг 1.1. Создание нейронной сети прямого распространения с одним скрытым слоем.

Шаг 1.2. Случайное формирование входных весов для каждого нейрона скрытого слоя для обучения машиной экстремального обучения.

Шаг 1.3. Инкрементное обучение нейронной сети путем вычисления матрицы выходных весов нейронов скрытого слоя W и матрицы P .

Шаг 1.4. Переход в состояние «сна». Данный шаг инициируется при достижении порогового значения ошибки или нормы весового вектора, что предотвращает накопление численных ошибок и переобучение модели

Шаг 2. Обучение нейронной сети в состоянии «сна». Выполняется алгоритм двухфазного обучения, описанный в п. 2.5: первая фаза – дообучение методом обратного распространения ошибки на псевдопримерах, вторая фаза – консолидация знаний с восстановлением матрицы P на основе машины экстремального обучения.

Шаг 3. Переход в состояние функционирования. В качестве исходных значений матрицы P используется матрица, полученная на шаге 2. Нейронная сеть возвращается к приему и обработке новых данных, сохраняя частичные знания о старых задачах.

Цепочку перехода из состояния функционирования в состояние «сна» и обратно можно повторять любое количество раз, пока в систему поступают новые задачи (новые знания).

Разработанный алгоритм посменного инкрементного обучения включает мемоизацию матриц системы нормальных уравнений для сокращения времени обучения

(п. 2.3), а также двухфазное обучение в режиме «сна» на основе сочетания машины экстремального обучения и метода обратного распространения ошибки (п. 2.5).

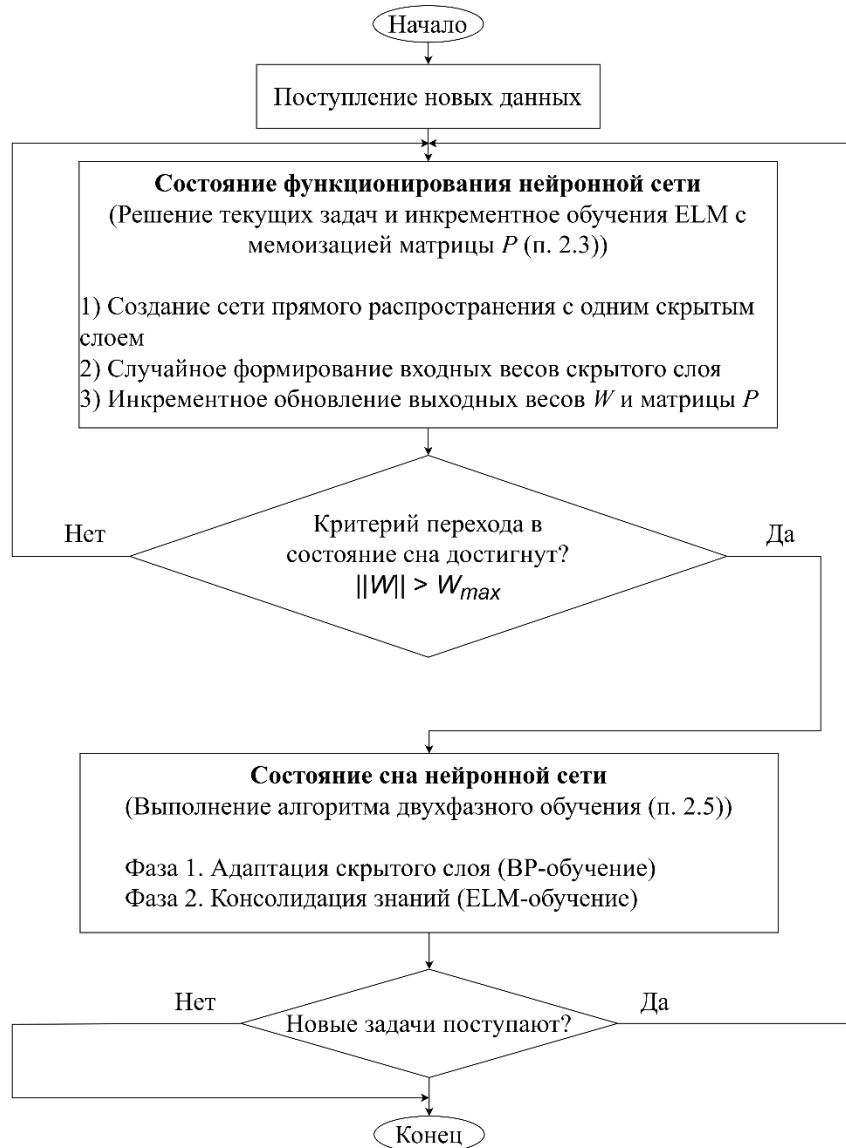


Рисунок 5 – Алгоритм посменного инкрементного обучения нейронной сети с циклическим чередованием состояний функционирования и «сна»

Общая вычислительная сложность алгоритма посменного инкрементного обучения на один цикл «функционирование – сон» включает три составляющие. Первая составляющая соответствует инкрементальному RLS-обновлению в состоянии функционирования. Вторая составляющая соответствует первой фазе

«сна» (ВР-настройка на псевдопримерах). Третья составляющая соответствует второй фазе «сна» (восстановление матрицы P).

По сравнению с классическим подходом, который требует полного переобучения на каждом шаге, выигрыш в вычислительной сложности составляет $O(n_{cycle})$ раз, где n_{cycle} – число примеров, обработанных за один цикл функционирования. Фаза «сна» со сложностью $O(L^3)$ выполняется периодически, поэтому метод применим к задачам с большими объемами потоковых данных.

Алгоритм сочетает достоинства обоих подходов: скорость и эффективность машины экстремального обучения для быстрой реакции на новые данные, а также точность и устойчивость, которые дает метод обратного распространения ошибки. Циклическое чередование состояний обеспечивает высокую точность инкрементного обучения при длительной работе модели.

2.7 Разработка алгоритма динамической адаптации архитектуры нейронной сети на основе метрик обученности и упругости нейронов

Алгоритм из раздела 2.6 адаптирует весовые коэффициенты нейронной сети к новым данным без полного переобучения. Но фиксированная архитектура ограничивает способность сети усваивать принципиально новую информацию и справляться с ростом сложности задач.

Чтобы снять эти ограничения, разработан дополнительный алгоритм динамической адаптации архитектуры: сеть автоматически наращивает число нейронов в скрытом слое и специализирует их под разные подзадачи.

Алгоритм встроен в общий процесс инкрементного обучения и запускается, когда ошибка на поступающих примерах устойчиво высока. Формально условие добавления нового нейрона записывается так:

$$\mathcal{L}(f_t(\mathbf{x}_t), y_t) > \theta, \quad (17)$$

$$|f_t(\mathbf{x}_t) - y_t| > \theta_{abs}, \quad (18)$$

где θ – относительный порог ошибки,

θ_{abs} – абсолютный порог ошибки.

Двухуровневая проверка обеспечивает устойчивость к выбросам и адаптацию к разным масштабам данных.

Если выполняются условия (17) и (18), в скрытый слой добавляется новый нейрон. Расширение включает следующие шаги:

– сохранение текущих параметров:

$$\mathbf{W}_{hid}^{old} = \mathbf{W}_{hid}, \quad \mathbf{b}_{hid}^{old} = \mathbf{b}_{hid}, \quad \mathbf{W}_{out}^{old} = \mathbf{W}_{out}, \quad (19)$$

– расширение матриц параметров:

$$\mathbf{W}_{hid}^{new} = \begin{bmatrix} \mathbf{W}_{hid}^{old} \\ \mathbf{w}_{new} \end{bmatrix}, \quad \mathbf{b}_{hid}^{new} = \begin{bmatrix} \mathbf{b}_{hid}^{old} \\ b_{new} \end{bmatrix}, \quad (20)$$

$$\mathbf{W}_{out}^{new} = [\mathbf{W}_{out}^{old} \quad w_{out,new}],$$

– контекстно-зависимую инициализацию:

$$\mathbf{w}_{new} = \alpha \cdot \mathbf{x}_t \cdot \frac{\|\mathbf{W}_{hid}^{old}\|_F}{\|\mathbf{x}_t\|_2}, \quad (21)$$

$$b_{new} = \beta \cdot \left(1 - \frac{\mathbf{x}_t^T \mathbf{x}_t}{\|\mathbf{x}_t\|_2^2 + \epsilon}\right), \quad (22)$$

$$w_{out,new} = \gamma \cdot \frac{y_t - f_t(\mathbf{x}_t)}{\|\mathbf{W}_{out}^{old}\|_2}, \quad (23)$$

где α, β, γ – коэффициенты инициализации,

ϵ – малая константа для численной стабильности,

$\|\cdot\|_F$ – норма Фробениуса.

Чтобы предотвратить катастрофическое забывание, алгоритм поддерживает буфер памяти $\mathcal{M}$ с репрезентативными примерами из предыдущих распределений [84–86]:

$$\mathcal{M}_t = \{(\mathbf{x}_i, y_i)\}_{i=1}^M, \quad (24)$$

где M – максимальный размер буфера.

Память обновляется по важности примеров:

$$p_i = \frac{\mathcal{L}(f_t(\mathbf{x}_i), y_i)}{\sum_{j=1}^{|\mathcal{M}_t|} \mathcal{L}(f_t(\mathbf{x}_j), y_j)}. \quad (25)$$

После структурных изменений выполняется дообучение с составной функцией потерь:

$$\mathcal{L}_{total} = \mathcal{L}_{current} + \lambda_{mem} \cdot \mathcal{L}_{memory} + \lambda_{reg} \cdot \mathcal{L}_{reg}, \quad (26)$$

где $\mathcal{L}_{current} = \mathcal{L}(f_t(\mathbf{x}_t), y_t)$ – ошибка на текущем примере,

$\mathcal{L}_{memory} = \frac{1}{|\mathcal{M}_t|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{M}_t} \mathcal{L}(f_t(\mathbf{x}_i), y_i)$ – ошибка на примерах из памяти,

$\mathcal{L}_{reg} = \|\mathbf{W}_{hid}^{new} - \mathbf{W}_{hid}^{old}\|_F^2 + \|\mathbf{W}_{out}^{new} - \mathbf{W}_{out}^{old}\|_2^2$ – регуляризация изменений параметров.

После добавления нейрона идет фаза интенсивного обучения с адаптивной скоростью обучения [87]:

$$\eta_{new} = \eta \cdot \min(1, \frac{\theta}{\mathcal{L}_{total}}), \quad (27)$$

$$\mathbf{W}^{(k+1)} = \mathbf{W}^{(k)} - \eta_{new} \cdot \nabla_{\mathbf{W}} \mathcal{L}_{total} + \mu \cdot (\mathbf{W}^{(k)} - \mathbf{W}^{(k-1)}), \quad (28)$$

где k – номер итерации дообучения,

μ – коэффициент момента.

Дообучение прекращается при выполнении одного из условий:

$$\begin{aligned} \left| \mathcal{L}_{total}^{(k)} - \mathcal{L}_{total}^{(k-1)} \right| &< \varepsilon \quad \text{или} \\ k &> K_{max} \quad \text{или} \\ \mathcal{L}_{total}^{(k)} &> \mathcal{L}_{total}^{(k-1)} + \delta, \end{aligned} \quad (29)$$

где ε – порог сходимости,

K_{max} – максимальное количество итераций,

δ – допуск на ухудшение.

В совокупности предложенный алгоритм реализует циклический процесс адаптации: вычисление ошибки на новом примере, принятие решения о дообучении или расширении, контекстная инициализация нового нейрона, интенсивная стабилизация модели с контролем сохранения знаний.

Наряду с добавлением новых нейронов в ответ на высокую ошибку на текущем примере, важной составляющей адаптации является рациональное использование уже имеющихся вычислительных ресурсов сети. С этой целью в алгоритм введен анализ состояния отдельных нейронов скрытого слоя, позволяющий выявлять элементы, которые могут быть функционально разделены для специализации под новые задачи без потери ранее усвоенных знаний.

Анализ состояния нейронов проводится на основе двух метрик, оценивающих их текущую активность и потенциал для адаптации:

$$S_i = \frac{\|\Delta \mathbf{W}_i\|}{\|\mathbf{W}_i\| + \varepsilon}, \quad (30)$$

$$E_i = \frac{1}{\|\nabla f(z_i)\| + \varepsilon}, \quad (31)$$

где S_i – степень обученности нейрона (относительное изменение его весов за последний интервал обучения),

E_i – упругость нейрона i (величина, обратная норме градиента его функции активации, характеризующая степень насыщения),

$\varepsilon = 10^{-8}$ – малая константа, обеспечивающая численную стабильность и предотвращающая деление на ноль в случае нулевой нормы весов или нулевого градиента функции активации (зоны насыщения).

Выбор такого значения обусловлен стандартным подходом к численной стабилизации в вычислительных задачах машинного обучения и не влияет на значения метрик в штатных режимах работы сети.

Нейрон считается кандидатом для адаптации (специализации или разделения) при выполнении условия:

$$\{S_i > \theta_S\} \cap \{E_i > \theta_E\} \cap \{\|\mathbf{W}_i\| > \theta_W\}, \quad (32)$$

где θ_S , θ_E , θ_W – заданные пороговые значения (гиперпараметры) для степени обученности, упругости и нормы весов соответственно.

Данное условие выявляет нейроны, которые являются значимыми (имеют большую норму весов $\|\mathbf{W}_i\| > \theta_W$), пластичными (демонстрируют активное изменение $S_i > \theta_S$) и при этом не насыщены (имеют высокую упругость $E_i > \theta_E$, то есть малый градиент активации). Такие нейроны наиболее эффективно могут быть перенаправлены на решение новых задач.

Процедура адаптации для нейрона i , удовлетворяющего условию (32), заключается в его функциональном разделении [84]. Создается его копия i' . Веса исходного нейрона подвергаются сильной регуляризации (применяется повышенный коэффициент L2-регуляризации) для сохранения паттернов активации, важных для предыдущих задач. Копия i' инициализируется весами,

близкими к весам i , но с добавлением небольшого гауссова шума, и становится основным адаптируемым элементом для усвоения новых данных.

В результате в скрытом слое возникают функционально связанные пары нейронов (i, i') : один отвечает за стабильность (старые знания), другой – за пластичность (новые знания). Сеть может динамически выделять вычислительные ресурсы под новые задачи, не дестабилизируя сформированные внутренние представления, то есть специализировать подгруппы нейронов (Рисунок 6).

При добавлении k новых нейронов в скрытый слой нужно обновить матрицы параметров сети. Сложность определяется расширением матриц W_{hid} (размера $d \times L$), b_{hid} (размера L) и W_{out} (размера $L \times l$), где l – размерность выходного пространства.

Расширение матрицы W_{hid} требует $O(d \cdot k)$ операций, расширение матрицы W_{out} – $O(L \cdot k + k \cdot l)$, обновление матрицы ковариации P – $O((L + k)^2)$. Общая сложность добавления k нейронов составляет $O((L + k)^2)$.

Функциональное разделение нейронов (создание копии с последующей регуляризацией) требует пересчета матриц для двух скрытых слоев и стоит $O(L^2)$.

Важно отметить, что операции адаптации архитектуры выполняются только при выполнении условий (17–18), то есть эпизодически, а не на каждом шаге инкрементного обучения. Это позволяет сохранить общую оценку сложности алгоритма посменного обучения на уровне $O(N L^2)$ для инкрементального этапа, поскольку структурные изменения происходят достаточно редко по сравнению с частотой поступления новых данных.

Предложенный алгоритм динамической адаптации архитектуры, включающий процедуру добавления нейронов по критерию ошибки и функциональное разделение нейронов на основе метрик обученности и упругости, позволяет сети контролируемо наращивать емкость и специализировать нейроны под новые задачи без катастрофического забывания.

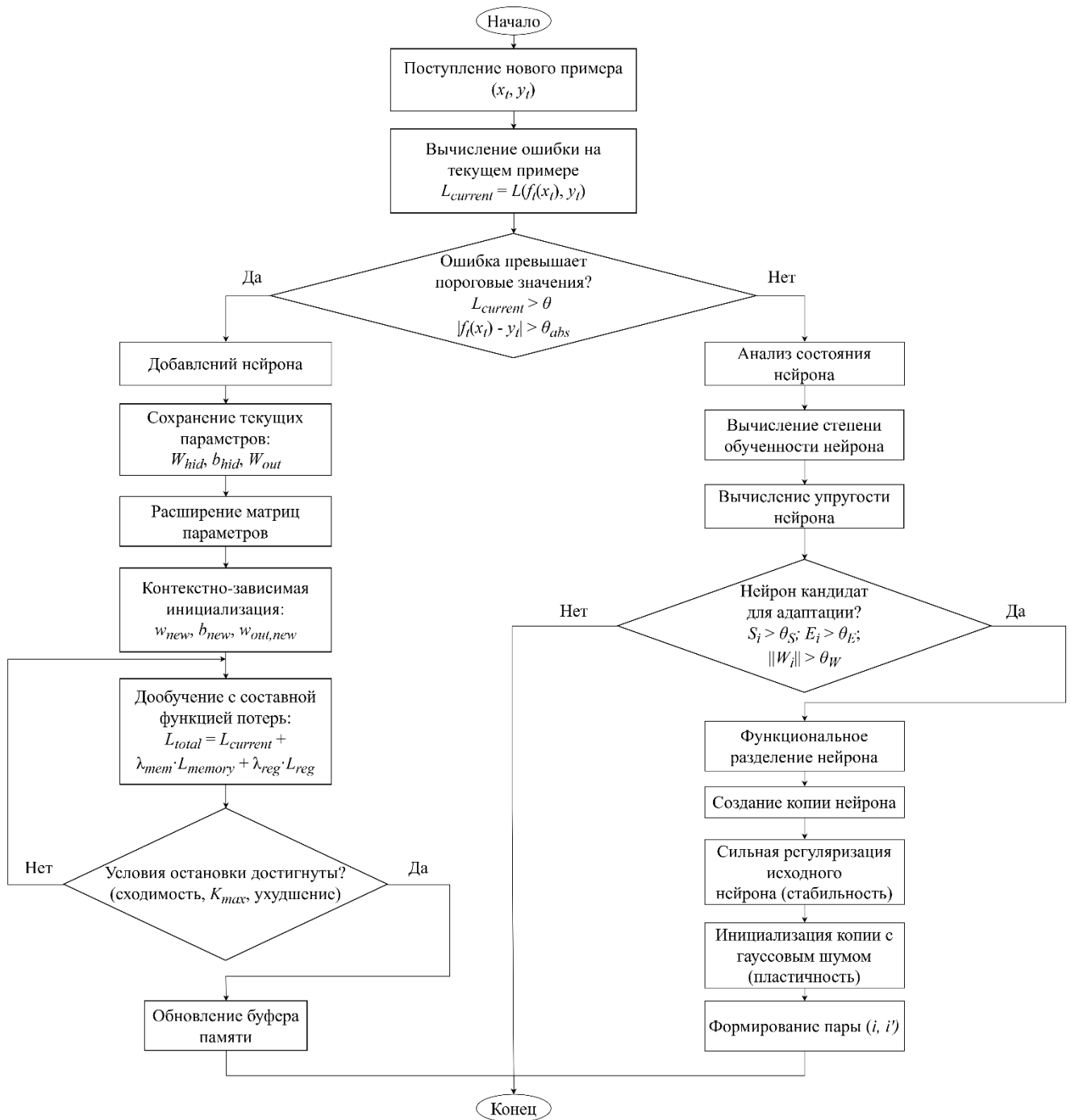


Рисунок 6 – Алгоритм динамической адаптации архитектуры нейронной сети на основе метрик обученности и упругости нейронов

Выводы по главе 2

Во второй главе была разработана модель данных для предварительного исследования алгоритмов распознавания образов на основе машины

экстремального обучения. Проводить исследования следует на простых моделях данных, отражающих закономерности, присутствующие в данных для распознавания простой и сложной физической активности человека. В качестве формы математической модели данных используется уравнение полинома степени m в каноническом виде, поскольку любая непрерывная на отрезке функция может быть хорошо приближена некоторым полиномом.

Разработан алгоритм двухфазной консолидации знаний нейронной сети в режиме «сна», отличающийся последовательным применением метода обратного распространения ошибки на первой фазе и инкрементной машины экстремального обучения на второй фазе, базирующийся на генерации псевдопримеров самой сетью. Показано, что такой подход позволяет предотвратить переобучение и адаптировать сеть к изменяющимся данным без необходимости хранения реальных обучающих примеров.

Введен алгоритм посменного инкрементного обучения нейронной сети, отличающийся циклическим чередованием состояния функционирования (обучение на основе машины экстремального обучения) и состояния «сна» (двухфазное обучение). Данный алгоритм позволяет использовать преимущества обоих подходов: скорость ELM для быстрого реагирования на новые данные и точность обратного распространения ошибки для стабилизации модели.

Чтобы повысить адаптивность модели к новым данным при непрерывном обучении, разработан алгоритм динамической адаптации архитектуры нейронной сети. В нем два взаимодополняющих процесса. Первый: добавление новых нейронов в скрытый слой, когда ошибка на текущем примере превышает порог, с контекстно-зависимой инициализацией параметров и воспроизведением репрезентативных примеров из буфера памяти. Второй: функциональное разделение существующих нейронов по метрикам обученности и упругости; оно создает специализированные пары нейронов, из которых один стабильно хранит старые знания, а другой пластично усваивает новые.

Разработанные алгоритмы позволяют эффективно использовать вычислительные ресурсы, поскольку при поступлении новых знаний полное

переобучение не нужно. Переход сети из состояния функционирования в состояние «сна» и обратно можно повторять неограниченное число раз, что обеспечивает непрерывное обучение нейросетевой модели.

Результаты разработки модели данных и алгоритмов инкрементного обучения нейронной сети, представленные в главе 2, получены автором лично и были частично опубликованы, в т. ч. в соавторстве (вклад автора более 70%) в [80; 82; 84; 93; 110].

В следующей главе эффективность предложенных алгоритмов исследуется на синтетических данных с контролируемой сложностью. Это позволяет отдельно оценить вклад каждого компонента (двухфазной консолидации и структурной адаптации) в общую производительность модели.

ГЛАВА 3. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ РАЗРАБОТАННЫХ АЛГОРИТМОВ НА СИНТЕТИЧЕСКИХ ДАННЫХ

В третьей главе проводятся экспериментальные исследования разработанных алгоритмов инкрементного обучения. Использование синтетических данных на данном этапе позволяет изолированно оценить вклад каждого компонента предложенных алгоритмов, смоделировать контролируемые сценарии нестационарности данных и обеспечить полную воспроизводимость экспериментов перед апробацией на реальных прикладных задачах.

3.1 Методика проведения экспериментальных исследований на синтетических данных

Для апробации разработанных в главе 2 алгоритмов инкрементного обучения перед их применением к реальной задаче распознавания физической активности человека проведена серия экспериментальных исследований на синтетических данных. Использование синтетических данных обусловлено следующими преимуществами: возможность контролировать сложность и структуру данных, моделировать различные сценарии поступления информации, изолированно оценивать вклад каждого компонента разработанных алгоритмов и обеспечивать полную воспроизводимость экспериментов при идентичных начальных условиях.

В зависимости от решаемой задачи в экспериментах используются два типа синтетических данных.

Для задачи регрессии (аппроксимации) данные генерируются на основе полиномиальных и нелинейных функций с добавлением гауссовского шума. Такой подход позволяет моделировать различные уровни сложности аппроксимируемой зависимости и оценивать способность алгоритма к точному восстановлению функциональной связи. В экспериментах используются полиномы от второй до

восьмой степени, а также комбинации различных функциональных зависимостей (например, объединение синусоидальной и квадратичной функций на разных участках области определения).

Для классификации применяется функция `make_classification` библиотеки `scikit-learn`. Она генерирует многоклассовые выборки с заданным числом информативных признаков, уровнем шума, разделимостью и количеством классов [88]. Это дает реалистичные сценарии поступления данных и сохраняет воспроизводимость. Задачи различаются начальным состоянием генератора случайных чисел, поэтому распределения получаются разными, но связанными.

Эксперимент проводится по следующей схеме [89].

Формирование последовательности задач. Исходный набор данных делится на T последовательных непересекающихся задач (временных окон). После перехода к следующей задаче данные предыдущих становятся недоступны для обучения, как и в реальной эксплуатации систем инкрементного обучения.

Обучение на задаче. Данные подаются на вход модели последовательными батчами фиксированного размера B . После каждого батча параметры модели обновляются выбранным алгоритмом. Когда выполняется одно из условий перехода (набрано заданное число батчей, ошибка превысила порог или обнаружен статистический дрейф), запускается двухфазная процедура консолидации (режим «сна»). При необходимости архитектура сети адаптивно расширяется.

Оценка качества. После завершения обработки всех батчей текущей задачи выполняется тестирование модели на контрольных выборках текущей и всех ранее изученных задач, что позволяет отслеживать динамику изменения качества и степень забывания ранее усвоенных знаний.

Для количественной оценки эффективности разработанных алгоритмов используется комплекс метрик, адаптированный к типу решаемой задачи (Таблица 4) [90; 91].

Таблица 4 – Метрики качества, используемые в экспериментальных исследованиях на синтетических данных

Тип задачи	Метрика	Обозначение	Назначение
Регрессия	Среднеквадратическая ошибка	MSE	Оценка точности аппроксимации
Классификация	Точность	Accuracy	Доля правильно классифицированных примеров
	Средняя точность	AvgAcc	Усредненная точность по задачам
	Относительное забывание	Forgetting	Снижение точности на ранее изученных задачах
Структурная адаптация	Коэффициент улучшения	κ	Относительное снижение ошибки после обучения
	Темп роста архитектуры	ρ	Среднее число добавляемых нейронов на задачу
	Частота добавления нейронов	τ	Среднее число актов добавления на задачу

Метрика MSE используется в задачах регрессии и характеризует среднеквадратическое отклонение предсказаний модели от истинных значений [92]. В задачах классификации основной метрикой является Accuracy, показывающая долю верно классифицированных объектов. Метрика Forgetting оценивает устойчивость модели к катастрофическому забыванию: чем меньше ее значение, тем лучше модель сохраняет знания о ранее изученных задачах [91]. Метрики, связанные со структурной адаптацией (коэффициент улучшения, темп роста архитектуры и частота добавления нейронов), характеризуют эффективность динамического изменения архитектуры сети в процессе инкрементного обучения.

Для объективной оценки разработанных алгоритмов проведено сравнение со следующими базовыми подходами.

Пакетная ELM (Batch ELM): классическая машина экстремального обучения, которая обучается сразу на всех данных. Она задает верхнюю границу качества, достижимого без ограничений на доступ к данным и вычислительные ресурсы.

OS-ELM (Online Sequential ELM): инкрементная версия ELM, которая аналитически обновляет выходные веса по мере поступления данных.

Гибридный метод ELM+BP: скрытый слой инициализируется случайно, а веса скрытого и выходного слоев на каждом инкрементном шаге обновляются обратным распространением ошибки, без смены фаз («сна»). Сравнение с ним показывает, что дает периодическая консолидация знаний.

EWC [31]: современный метод регуляризации, который защищает важные веса от изменений с помощью матрицы Фишера. Нужен для сравнения с существующими способами предотвращения катастрофического забывания.

Предлагаемый алгоритм (Proposed): разработанный алгоритм посменного инкрементного обучения с двухфазной консолидацией (раздел 2.6) и динамической адаптацией архитектуры (раздел 2.7).

Эксперименты реализованы на Python с библиотеками NumPy, SciPy и scikit-learn. Для статистической достоверности каждый эксперимент повторялся не менее 10 раз; в таблицах и на графиках приведены средние значения и стандартные отклонения.

Эту методологию и метрики используют следующие разделы. В разделе 3.2 проверяются базовые механизмы консолидации, в разделе 3.3 подбираются оптимальные конфигурации, в разделе 3.4 оценивается устойчивость к катастрофическому забыванию, а в разделе 3.5 исследуется динамическая структурная адаптация.

3.2 Исследование алгоритма посменного инкрементного обучения на задаче аппроксимации

Алгоритм посменного инкрементного обучения с двухфазной консолидацией знаний (раздел 2.6) проверен на задаче аппроксимации полиномиальных зависимостей. Сложность функции здесь легко контролировать, а результаты наглядны, поэтому вклад каждого компонента алгоритма оценивается отдельно.

Эксперимент состоял из четырех последовательных задач аппроксимации полиномов, которые имитируют потоковое поступление данных. Четыре задачи нужны, чтобы проверить устойчивость алгоритма при нескольких порциях новых данных подряд и проследить забывание на достаточном горизонте. Первая задача квадратичная, вторая и третья кубические, четвертая соответствует полиному четвертой степени. Растущая степень показывает, как алгоритм справляется с усложнением зависимости.

Данные каждой задачи генерировались из равномерного распределения на заданном интервале с гауссовским шумом (стандартное отклонение 0,05). Обучающая выборка каждой задачи содержала 200 примеров. Интервалы генерации частично перекрывались: так моделируется приход новых данных из смежных областей признакового пространства.

Архитектура нейронной сети содержала 100 нейронов в скрытом слое с функцией активации $\tanh$. Выбор количества нейронов обусловлен необходимостью обеспечения достаточной емкости модели для аппроксимации полиномов до четвертой степени включительно. В качестве функции активации выбран гиперболический тангенс, обеспечивающий гладкость аппроксимации и устойчивость процесса обучения.

Обучение выполнялось согласно алгоритму, описанному в разделе 2.6. Параметры эксперимента приведены в Таблице 5. Все эксперименты проведены с использованием пакетной обработки данных с размером батча, равным одному

примеру (online обучение), что соответствует наиболее сложному сценарию инкрементного обучения.

Таблица 5 – Параметры эксперимента

Параметр	Значение
Число скрытых нейронов	100
Коэффициент регуляризации	0,001
Коэффициент забывания	0,995
Скорость обучения (фаза 1)	0,005
Число эпох в фазе «сна»	50
Количество псевдопримеров	150
Коэффициент смещения	0,3

Коэффициент забывания определяет скорость устаревания информации в RLS-обновлении: значение 0,995 соответствует учету примерно 200 последних примеров при обновлении корреляционной матрицы. В качестве основной метрики использовалась среднеквадратическая ошибка (MSE). Для оценки устойчивости к катастрофическому забыванию использовалась метрика Forgetting, определяемая как абсолютный рост MSE на первой задаче после изучения всех последующих задач.

Сравнивались четыре конфигурации:

- Batch ELM: обучение на всех данных сразу (верхняя граница качества);
- OS-ELM: инкрементальное обучение без консолидации;
- ELM + 1-Phase Sleep: только первая фаза «сна» (настройка входных весов обратным распространением);
- Proposed (2-Phase Sleep): алгоритм с двухфазной консолидацией.

Каждый эксперимент повторялся 10 раз, в таблицах и на рисунках даны средние значения.

Результаты сравнения приведены в Таблице 6.

Таблица 6 – Сравнение методов и алгоритмов аппроксимации

Метод или алгоритм	Начальная ошибка на задаче 1	Финальная ошибка на задаче 1	Ошибка на задаче 4	Абсолютное забывание
Batch ELM	—	$0,0164 \pm 0,0002$	$0,0573 \pm 0,0011$	—
OS-ELM	$0,0023 \pm 0,0001$	$0,0440 \pm 0,0016$	$0,0178 \pm 0,0001$	$0,0418 \pm 0,0016$
1-Phase Sleep	$0,0023 \pm 0,0001$	$0,0450 \pm 0,0024$	$0,0178 \pm 0,0001$	$0,0427 \pm 0,0024$
Proposed	$0,0023 \pm 0,0001$	$0,0149 \pm 0,0006$	$0,0363 \pm 0,0011$	$0,0126 \pm 0,0006$

Примечание: для *Batch ELM* начальная ошибка на первой задаче не определена, так как модель обучается на всех данных сразу.

OS-ELM заметно ухудшает качество на ранее изученных данных. MSE на первой задаче вырастает с 0,002 до 0,043, то есть в 19,3 раза. Это подтверждает катастрофическое забывание при стандартном инкрементальном обучении: модель обновляется на новых данных и теряет способность точно воспроизводить прежние зависимости.

Одна первая фаза «сна», без последующего пересчета выходного слоя, устойчивость не улучшает. Абсолютное забывание составляет 0,043 (рост в 20,1 раза), то есть немного хуже, чем у OS-ELM. Причина в том, что изменение входных весов без корректировки выходных нарушает оптимальность аналитического решения ELM, полученного для исходных весов. В такой постановке первая фаза «сна» сама по себе забывание не снижает. Но в составе алгоритма она полезна: готовит входные веса для последующей аналитической консолидации.

Полная двухфазная консолидация заметно повышает устойчивость к забыванию. Рост MSE на первой задаче ограничен 6,5 раза (с 0,002 до 0,014), это в 3 раза лучше, чем у OS-ELM. Абсолютное забывание составляет 0,012 против 0,041 у OS-ELM, то есть на 69,8% ниже. Алгоритм двухфазной консолидации работает.

Есть и ожидаемая плата за это: баланс между пластичностью и стабильностью смещается в сторону стабильности. На четвертой задаче (новые данные) MSE предложенного алгоритма равна 0,037, у OS-ELM 0,017. Способы сохранения старых знаний ограничивают быструю адаптацию к новому. Но при трехкратном выигрыше в устойчивости такой компромисс для практических задач приемлем.

На Рисунке 7 показано, как менялась MSE на первой задаче по мере изучения следующих задач для всех методов и алгоритмов. По оси абсцисс отложены номера задач в порядке поступления, по оси ординат MSE на данных первой задачи, измеренная после каждой изученной задачи.

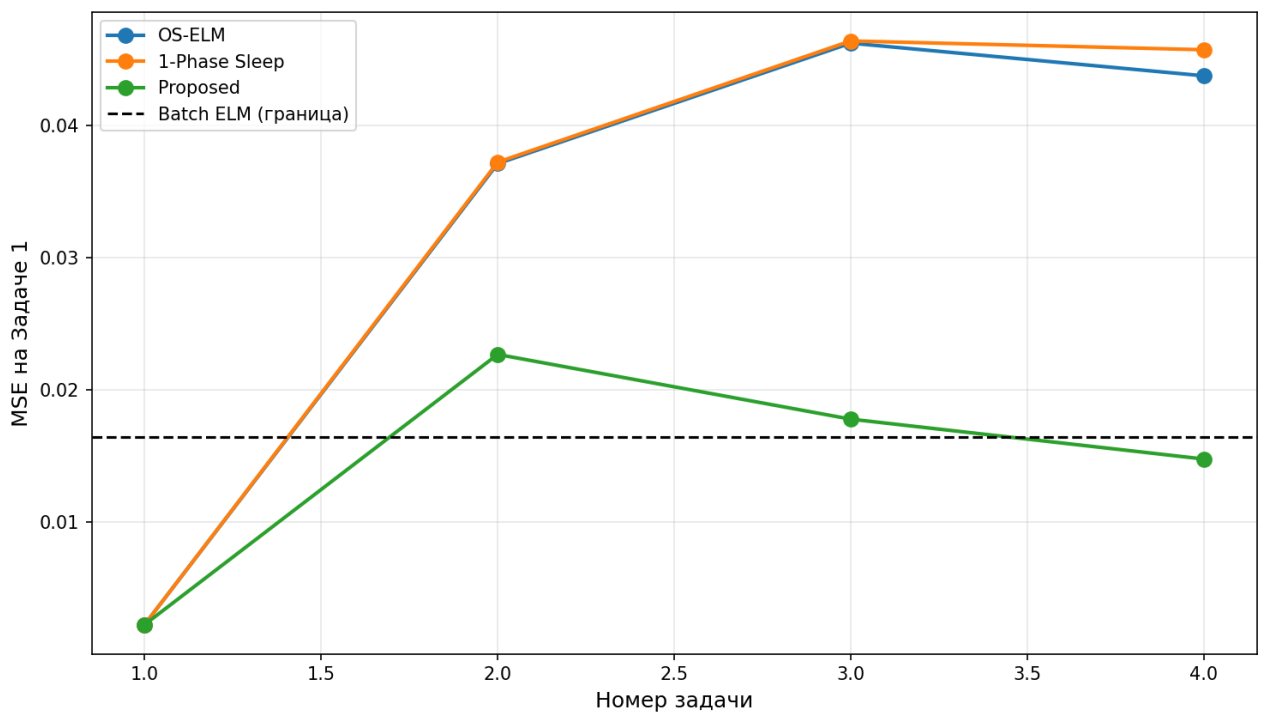


Рисунок 7 – Динамика ошибки на первой задаче при последовательном изучении задач

У OS-ELM и алгоритма с одной фазой «сна» ошибка на первой задаче монотонно растет с каждой новой задачей, то есть ранее усвоенное все сильнее деградирует. К концу обучения она достигает 0,043 и 0,045 соответственно.

У предложенного алгоритма динамика другая. После второй задачи ошибка вырастает до 0,022, но после двухфазной консолидации снижается до 0,017. После третьей и четвертой задач повторяется то же: рост при поступлении новых данных, затем падение после консолидации. К концу обучения ошибка на первой задаче равна 0,014, что лишь в 6,5 раза больше начального значения.

Консолидация не просто замедляет забывание, но и частично возвращает утраченные знания. Это происходит потому, что вторая фаза «сна» пересчитывает выходные веса по накопленным псевдопримерам, и модель сохраняет старые зависимости, усваивая новые.

Результаты подтверждают эффективность двухфазной консолидации в регрессии. Дальше подбираются оптимальные гиперпараметры.

3.3 Экспериментальные исследования по настройке гиперпараметров модели

Цель исследования: найти набор гиперпараметров нейросетевой архитектуры, при котором последовательное обучение на синтетических данных дает максимальную среднюю точность, устойчивость и приемлемые вычислительные затраты.

Гиперпараметры подбирались методом Grid Search, то есть полным перебором комбинаций значений из заданных диапазонов. Эксперимент проводился на синтетических данных с последовательным поступлением задач по методике из раздела 3.1.

В процессе поиска отслеживались метрики прогресса обучения, изменение весов и уровень забывания. Всего рассмотрено 729 уникальных комбинаций параметров. Каждая комбинация оценивалась на основе средней точности, достигнутой моделью в ходе последовательного обучения на нескольких задачах.

В Таблице 7 представлены исследуемые гиперпараметры и диапазоны их значений.

Таблица 7 – Исследуемые гиперпараметры модели

Гиперпараметр	Описание	Диапазон значений	Влияние
n_hidden	Количество нейронов в скрытом слое сети.	50, 100, 200, 300, 500	Увеличение числа нейронов повышает емкость модели, но увеличивает риск переобучения и вычислительные затраты.
sleep_threshold_weight_change	Порог изменения нормы весов для активации фазы «сна».	0,01; 0,02; 0,05	Низкие значения приводят к более частому переходу в режим «сна», что улучшает стабильность, но замедляет обучение.
sleep_threshold_forgetting	Порог уровня забывания, используемый как второй критерий для активации режима «сна».	0,05; 0,1; 0,2	Более низкие значения делают модель чувствительной к потере знаний, увеличивая частоту «сна».
max_sleep_epochs	Максимальное количество эпох обучения в фазе «сна».	5, 10, 20	Большее число эпох улучшает консолидацию знаний, но замедляет выполнение.
lambda_val	Коэффициент L2-регуляризации, применяемый в обучении и в фазе «сна».	0,001; 0,01; 0,1	Снижает вероятность переобучения и стабилизирует инверсию матрицы P.
learning_rate	Скорость обучения в фазе «сна».	0,0005; 0,001; 0,005	Высокие значения ускоряют обучение, но могут вызвать нестабильную сходимость.

На Рисунке 8 представлены результаты Grid Search в виде зависимости средней точности от комбинаций гиперпараметров. Наилучшая средняя точность, достигнутая в ходе экспериментов, составила 0,983.

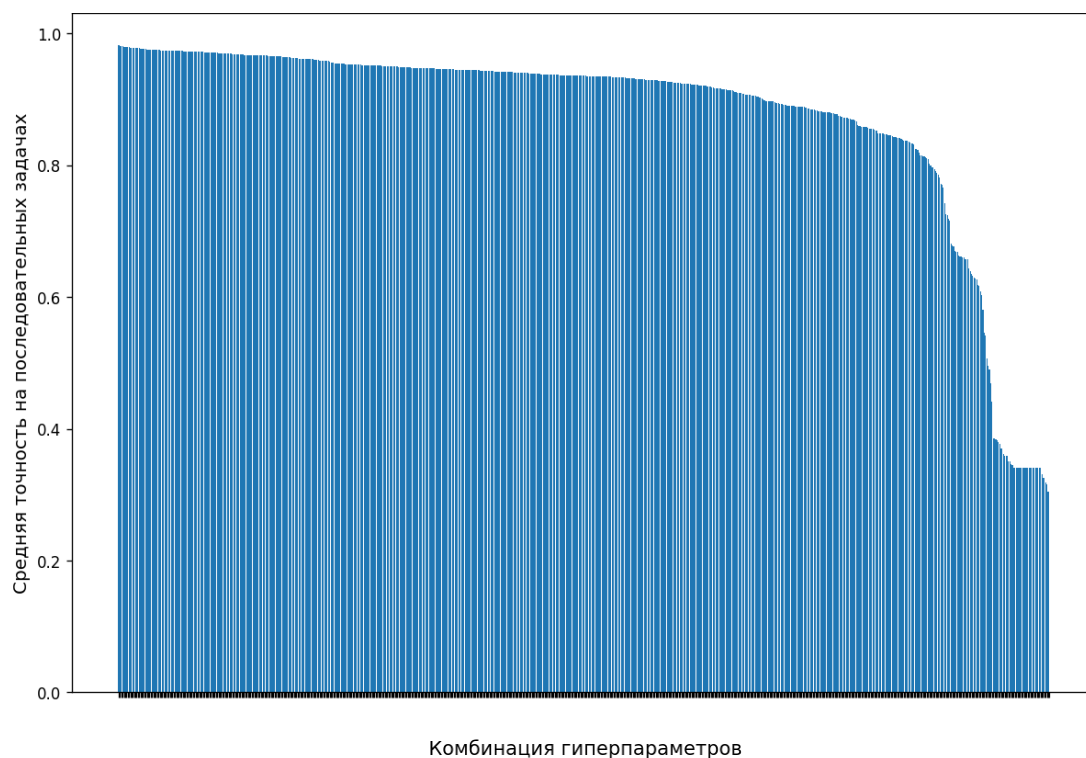


Рисунок 8 – Результаты подбора гиперпараметров

Как видно из Рисунка 8, точность возрастает с увеличением числа скрытых нейронов до 300, после чего наблюдается насыщение. При этом влияние порогов активации «сна» и коэффициента регуляризации менее выражено, однако их оптимальные значения позволяют дополнительно повысить стабильность обучения.

Оптимальные значения гиперпараметров, обеспечивающие данную точность представлены в Таблице 8.

Таблица 8 – Оптимальные значения гиперпараметров модели

Гиперпараметр	Оптимальное значение
n_hidden	300
sleep_threshold_weight_change	0,02
sleep_threshold_forgetting	0,1
max_sleep_epochs	10
lambda_val	0,01
learning_rate	0,001

$n_hidden = 300$: для сложных зависимостей в синтетических данных нужна довольно высокая емкость модели. При меньшем числе нейронов точность падала, при большем росли вычислительные затраты без заметного выигрыша в качестве.

$sleep_threshold_weight_change = 0,02$ сравнительно низкий порог. Знания приходится консолидировать вовремя, тогда новые данные меньше портят ранее усвоенное.

$sleep_threshold_forgetting = 0,1$: «сон» запускается при первых признаках забывания, и информация лучше сохраняется между задачами.

$max_sleep_epochs = 10$ дает достаточное обучение в фазе «сна» без заметного роста времени работы. Больше эпох точность не повышает.

$lambda_val = 0,01$ хорошо балансирует регуляризацию и способность к обучению и обеспечивает численную устойчивость при работе с матрицей P .

$learning_rate = 0,001$ подходит для тонкой настройки весов в фазе «сна» и дает стабильную сходимость без скачков.

Итак, найдена оптимальная комбинация гиперпараметров. Максимальная средняя точность на последовательных синтетических задачах равна 0,9831 [89]. Подобранные значения используются в эксперименте, представленном в разделе 3.4.

3.4 Исследование устойчивости алгоритма посменного обучения к катастрофическому забыванию на синтетических данных

Цель исследования: проверить, может ли алгоритм из раздела 2.6 последовательно учиться на потоке из нескольких разных, но связанных задач, не подвергаясь катастрофическому забыванию и сохраняя баланс между пластичностью и стабильностью.

Эксперимент шел на синтетических данных с последовательным поступлением задач по методике из раздела 3.1. Рассматривалось инкрементное обучение в условиях сдвига домена (domain-incremental learning). Пространство классов (множество распознаваемых образов) остается прежним, статистическое

распределение входных признаков (домен) от задачи к задаче меняется. Идентификатор задачи при тестировании не сообщается, обучающие выборки прежних задач после перехода к следующей недоступны. Так выглядит реальная нестационарность признакового пространства, и на такой постановке анализируется устойчивость алгоритмов к забыванию.

Данные генерировались с контролируемой сложностью. Каждая задача была задачей многоклассовой классификации: 3 класса, 20 признаков. Использовалась функция `make_classification` библиотеки `scikit-learn`, которая обеспечивает информативные признаки и ограниченный уровень шума. Задачи различались начальным состоянием генератора случайных чисел.

Эксперимент состоял из пяти задач. Для каждой формировались обучающая и тестовая выборки фиксированного размера. Обучение шло в потоковом режиме пакетами по 50 примеров. Каждую задачу проходили по обучающей выборке 10 раз, затем переходили к следующей, не возвращаясь к данным предыдущих.

Для обеспечения сопоставимости признаков применялась стандартизация данных. Параметры нормализации определялись по обучающей выборке первой задачи, фиксировались и использовались для всех последующих задач, что исключало утечку информации из будущих задач.

Архитектура модели представляла собой однослойную нейронную сеть со скрытым слоем, содержащим 300 нейронов, в соответствии с результатами подбора гиперпараметров (раздел 3.3). В качестве функции активации скрытого слоя использовалась функция ReLU.

В качестве основной метрики качества использовалась классификационная точность (Accuracy), определяемая как доля корректно классифицированных примеров на тестовой выборке. После обработки каждого пакета данных вычислялась точность на тестовой выборке текущей задачи, а также средняя точность на подмножестве тестовых данных всех ранее изученных задач.

Для количественной оценки катастрофического забывания использовалась метрика забывания (Forgetting), вычисляемая как абсолютное снижение точности по сравнению с начальным значением после изучения соответствующей задачи.

Итоговый показатель забывания определялся как среднее значение по всем ранее изученным задачам.

Для оценки эффективности предложенного алгоритма было проведено сравнение с базовыми методами того же класса, позволяющее изолировать вклад посменного обучения:

ELM (пакетная) – классическая машина экстремального обучения, переобучаемая «с нуля» на совокупности данных всех доступных к моменту оценки задач. Данный метод служит верхней границей точности при отсутствии ограничений на доступ к данным и вычислительные ресурсы.

OS-ELM – Online Sequential ELM, инкрементальная версия, обновляющая выходной слой аналитически по мере поступления новых данных.

ELM+BP – гибридная модель, в которой, как и в предложенном алгоритме, скрытый слой инициализируется случайно, а обновление весов как скрытого, так и выходного слоя осуществляется методом обратного распространения ошибки на каждом инкрементном шаге, но без смены фаз «сна».

Такое сравнение позволяет объективно оценить преимущества интеграции быстрой аналитической адаптации (ELM) с периодической консолидацией знаний (фаза «сна») в рамках одной архитектуры [93].

Результаты количественного сравнения приведены в Таблице 9.

Таблица 9 – Сравнение предложенного алгоритма с базовыми

Метод или алгоритм	Точность на 1-й задаче	Точность на 5-й задаче	Средняя точность на последовательности задач	Забывание
ELM (пакетная)	$0,983 \pm 0,001$	$0,983 \pm 0,001$	$0,983 \pm 0,001$	–
OS-ELM	$0,981 \pm 0,002$	$0,63 \pm 0,02$	$0,72 \pm 0,01$	$0,35 \pm 0,02$
ELM + BP	$0,982 \pm 0,002$	$0,70 \pm 0,02$	$0,76 \pm 0,01$	$0,28 \pm 0,02$
Предложенный алгоритм	$0,983 \pm 0,001$	$0,75 \pm 0,02$	$0,80 \pm 0,01$	$0,23 \pm 0,02$

Пакетная ELM задает верхнюю границу точности, когда доступ к данным не ограничен. Среди инкрементальных методов больше всех забывает OS-ELM (0,35): консолидации знаний в ней нет. ELM+BP устойчивее (0,28), но межзадачная интерференция накапливается и ограничивает эффект. Предложенный алгоритм дает точность 0,75 на последней задаче и наименьшее забывание среди инкрементальных методов (0,23). Forgetting снижен на 34,3% по сравнению с ELM+BP и на 34,2% по сравнению с OS-ELM. Это подтверждает пользу разделения быстрой адаптации на основе ELM и периодической консолидации знаний в фазе «сна».

На Рисунке 9 показана динамика средней точности на всех предыдущих тестовых данных в ходе инкрементного обучения.

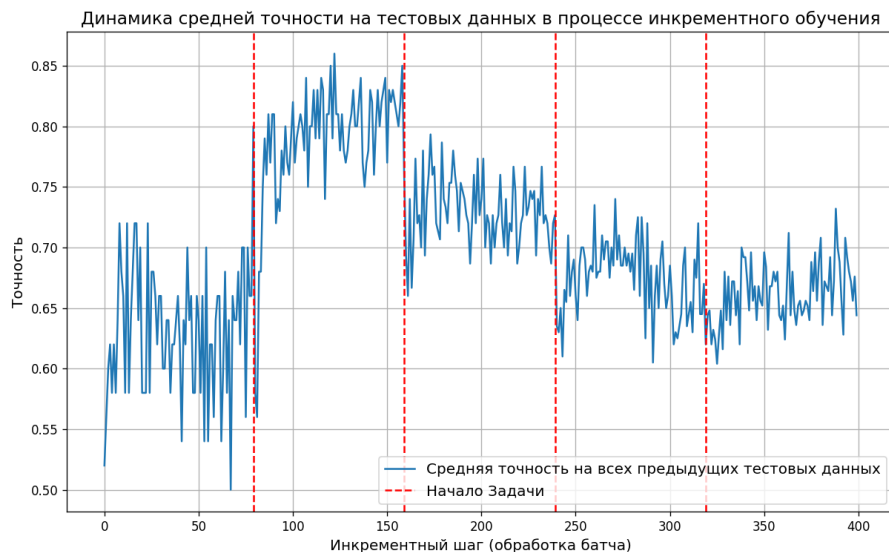


Рисунок 9 – Динамика средней точности на тестовых данных

По оси абсцисс отложены инкрементные шаги, по оси ординат средняя точность. Вертикальные пунктирные линии отмечают начало обучения новым задачам. Вначале точность колеблется: идет первичная адаптация к первой задаче. С каждой новой задачей средняя точность ожидаемо падает, но благодаря

консолидации быстро восстанавливается. К концу обучения она стабилизируется около 0,67, то есть модель устойчива к катастрофическому забыванию.

На Рисунке 10 показана динамика точности на тестовых данных для каждой из пяти задач в ходе инкрементного обучения.

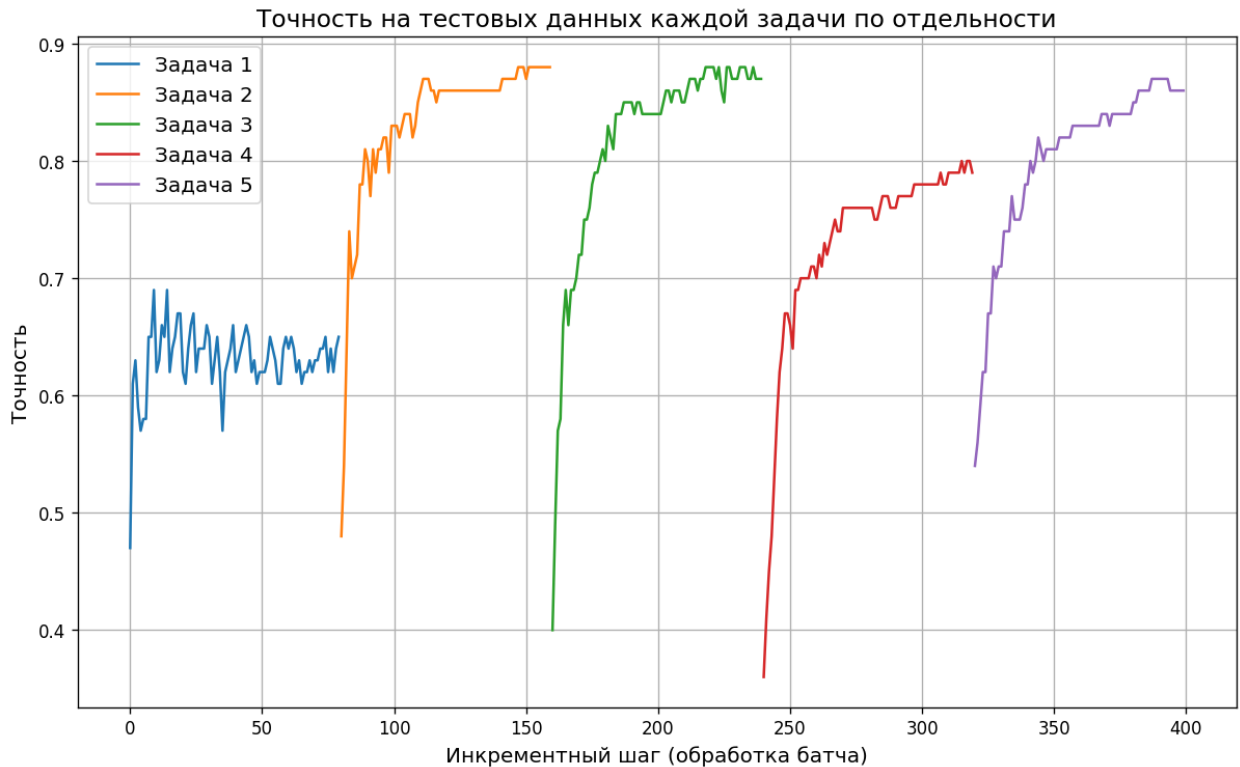


Рисунок 10 – Точность обучения на каждой задаче

На Задаче 1 точность быстро растет: первичная адаптация прошла успешно. После введения Задачи 2 и следующих точность на Задаче 1 постепенно снижается, и это типичное проявление катастрофического забывания. Однако она стабилизируется заметно выше случайного уровня (0,33), значит, алгоритм смягчает проблему. Для каждой новой задачи картина похожая: при появлении точность низкая, затем в ходе обработки пакетов быстро растет. Модель сохраняет пластичность и хорошо усваивает новые данные.

Результаты подтверждают: алгоритм посменного инкрементного обучения с двухфазной консолидацией (раздел 2.6) дает лучший баланс между пластичностью

и стабильностью среди рассмотренных инкрементных подходов. Он сохраняет высокую точность на последних задачах при наименьшем относительном забывании, поэтому подходит для инкрементного обучения на потоковых данных.

Устойчивость к забыванию подтвердила эффективность базового механизма консолидации. Далее оценивается второй компонент подхода, динамическая структурная адаптация архитектуры, которая полезна при значительном изменении распределения данных.

3.5 Исследование эффективности алгоритма структурной адаптации нейронной сети в условиях инкрементного обучения

Эксперименты идут по методике из раздела 3.1 и комплексно оценивают алгоритм динамической адаптации архитектуры нейронной сети (раздел 2.7) при инкрементальном поступлении данных.

Задачи исследования:

1. Оценка способности алгоритма к последовательному усвоению информации без катастрофического забывания.
2. Анализ динамики роста архитектуры сети в процессе обучения.
3. Сравнение производительности с традиционными подходами к обучению нейронных сетей.
4. Исследование устойчивости алгоритма к переобучению.
5. Анализ влияния параметров алгоритма на его производительность.

Для проведения исследований использовались специализированные синтетические наборы данных, разработанные для моделирования различных сценариев инкрементного обучения.

Набор данных 1 (смешанное распределение): объединение двух различных функциональных зависимостей.

Область 1: $y = \sin(x) + \mathcal{N}(0,0.1)$, $x \in [-3,0]$,

Область 2: $y = 0.5(x - 3)^2 - 1 + \mathcal{N}(0,0.1)$, $x \in [3,6]$.

Набор данных 2 (сложное начальное распределение): имитация сценария с шумными начальными данными.

$$y = 0.3x^3 - 0.2x + \mathcal{N}(0,0.2), \quad x \in [-0.5,0.5].$$

Набор данных 3 (расширенное распределение): дополнительные данные с упрощенной зависимостью.

$$y = 0.5x + \mathcal{N}(0,0.1), \quad x \in [-1.5,1.5].$$

Общий объем данных составлял 90 примеров, распределенных следующим образом: 15 начальных примеров (сложное распределение) и 75 дополнительных примеров (упрощенное распределение).

В рамках экспериментальных исследований все параметры предложенного алгоритма фиксировались и не изменялись в ходе экспериментов. Значения гиперпараметров были подобраны эмпирически на этапе предварительных запусков и использовались неизменными для всех сценариев обучения, что обеспечило корректность сравнительного анализа. Параметры алгоритма представлены в Таблице 10.

Таблица 10 – Параметры алгоритма

Параметр	Значение
Начальное число нейронов	3-5
Порог ошибки добавления нейрона	0.2
Скорость обучения	0.01
Число эпох локального дообучения	3
Число эпох после добавления нейрона	2
Коэффициент инициализации весов	0.1
Смещение нового нейрона	0.1

Для оценки эффективности использовался набор метрик:

1. Среднеквадратичная ошибка (MSE): $MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$.

2. Динамические метрики процесса обучения: ошибка до обучения: $\epsilon_{before} = \frac{1}{T} \sum_{t=1}^T \mathcal{L}(f_t(\mathbf{x}_t), y_t)$, ошибка после обучения: $\epsilon_{after} = \frac{1}{T} \sum_{t=1}^T \mathcal{L}(f_{t+1}(\mathbf{x}_t), y_t)$, коэффициент улучшения: $\kappa = \frac{\epsilon_{before} - \epsilon_{after}}{\epsilon_{before}}$.

3. Метрики роста архитектуры: темп роста: $\rho = \frac{m_{final} - m_{initial}}{T}$, частота добавления нейронов: $\tau = \frac{N_{additions}}{T}$.

Алгоритм сравнивался с тремя базовыми подходами и одним современным методом инкрементного обучения: обычная нейросеть с катастрофическим забыванием: обучение на последовательных блоках данных без сохранения знаний; обычная нейросеть на полном наборе данных: обучение на всех доступных данных (верхняя граница производительности); стандартная нейросеть с фиксированной архитектурой: классический подход с заранее заданной архитектурой; EWC: современный метод регуляризации, который предотвращает катастрофическое забывание, ограничивая изменение весов, важных для предыдущих задач [31].

Начальные условия и аппаратные конфигурации во всех экспериментах были одинаковыми.

Эксперимент 1. Устойчивость к катастрофическому забыванию. Первый эксперимент количественно оценивает, сохраняет ли алгоритм представления, сформированные на прежних данных, когда обучается на новых, статистически отличных областях. Использовался набор данных 1, разделенный на две непересекающиеся части. Алгоритм обучался сначала на первой, затем на второй. Ключевая метрика: изменение MSE на первой части данных после обучения на второй. Результаты приведены в Таблице 11.

Таблица 11 – Сравнительный анализ устойчивости к катастрофическому забыванию

Алгоритм	Начальная MSE	Финальная MSE	Изменение, %
Растущая нейросеть	0.3495	0.1215	-65.2
Обычная сеть (забывание)	0.2768	0.1016	-63.3
Обучение на полном наборе данных	0.3099	0.0623	-79.9

Предложенный алгоритм снижает ошибку на 65,2%, что сравнимо с результатом обычной сети (-63,3%). Важно, что обычная сеть получает этот результат при катастрофическом забывании: знание о первой области после обучения на второй практически утрачено. Растущая нейросеть, напротив, остается устойчивой к забыванию и подстраивает архитектуру под новые данные без деградации прежних навыков. При этом ей, в отличие от обучения на полном наборе, не нужен одновременный доступ ко всем данным, поэтому она годится для онлайн-обучения.

Эксперимент 2. Динамика роста и адаптации архитектуры.

Второй эксперимент был сфокусирован на исследовании эволюции структуры сети в процессе последовательного поступления данных. Начальная архитектура была инициализирована с 5 нейронами в скрытом слое. В процессе обработки 90 примеров фиксировались моменты добавления новых нейронов и итоговый размер сети. Динамика изменения архитектуры сети в ходе обучения детально представлена в Таблице 12.

Таблица 12 – Динамика роста количества нейронов в процессе обучения

Этап обучения	Примеров обработано	Нейронов	Добавлено нейронов
Начальная инициализация	0	5	—
После 15 примеров	15	11	6
После 30 примеров	30	19	8
После 45 примеров	45	19	0
После 60 примеров	60	19	0
После 90 примеров	90	20	1

Наблюдается выраженная нелинейность процесса роста. На начальном этапе (первые 30 примеров) происходит активное добавление 14 нейронов, что свидетельствует о фазе интенсивного освоения сетью базовых закономерностей в данных. Последующая стабилизация размера скрытого слоя (на участке от 30 до 60 примеров) указывает на то, что достигнутая сложность модели стала адекватна текущей задаче.

Эксперимент 3. Статистика ошибок инкрементного обучения.

Третий эксперимент был нацелен на детальный анализ эффективности каждого шага обучения. Для каждого из 25 поступающих примеров фиксировались ошибка предсказания до обучения и после применения алгоритма. На основе этих данных вычислялись агрегированные метрики, включая коэффициент улучшения. Статистика ошибок представлена в Таблице 13.

Алгоритм работает устойчиво и эффективно. Среднее улучшение ошибки на шаге составило 0,724, что соответствует среднему коэффициенту улучшения 53,7%. Каждый цикл обучения, будь то структурная адаптация или подкрепление, в среднем более чем вдвое снижал ошибку предсказания для нового примера. Частота добавлений нейронов (0,178) на начальном этапе высока: сеть активно растет. Но 9 шагов подкрепления показывают, что алгоритм умеет использовать существующую архитектуру и не усложняет ее без необходимости.

Таблица 13 – Статистика ошибок в процессе инкрементного обучения

Параметр	Значение	Параметр	Значение
Средняя ошибка до обучения	1,348	Количество добавлений нейронов	16
Средняя ошибка после обучения	0,624	Количество подкреплений	9
Среднее улучшение	0,724	Финальный размер слоя	18
Максимальная ошибка	1,863	Темп роста ρ	0,178
Минимальная ошибка	0,006	Частота добавлений τ	0,178

Эксперимент 4. Склонность к переобучению. Четвертый эксперимент проверяет обобщающую способность алгоритма на ограниченных и зашумленных данных. Растущая нейросеть обучалась на 90 примерах, обычная сеть с фиксированной архитектурой только на первых 15: так моделируется нехватка данных. Результаты приведены в Таблице 14.

Таблица 14 – Сравнение обобщающей способности моделей

Алгоритм	Train MSE	Val MSE	Val/Train MSE, %
Растущая нейросеть (90 примеров)	0,068	0,016	-76,0
Обычная сеть (15 примеров)	0,225	0,586	+160,2

Здесь ошибка на валидационной выборке ниже, чем на обучающей. Это может объясняться сочетанием регуляризации, ранней остановки и особенностями формирования валидационной выборки, и на малых зашумленных выборках такое встречается. Результат говорит о высокой обобщающей способности алгоритма в этом сценарии.

Эксперимент 5. Сравнение с методом EWC. Пятый эксперимент сравнивает предложенный алгоритм с адаптивной архитектурой и метод EWC.

Обучение проходило последовательно на четырех задачах с разными распределениями данных. После каждой задачи MSE оценивалась на всех пройденных задачах, так что видны динамика точности и степень забывания (Рисунок. 11).

Точность адаптивной архитектуры сопоставима с EWC (MSE 0,039 против 0,039), но она устойчивее к забыванию: средняя мера деградации качества 0,019 против 0,022 у EWC (ниже на 10,8%).

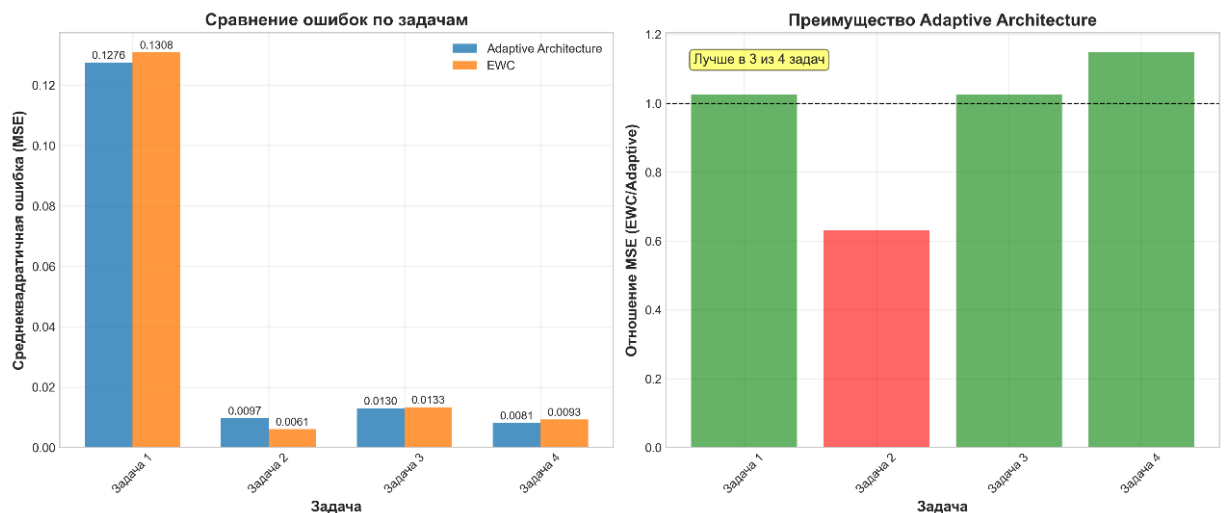


Рисунок 11 – Сравнение среднеквадратической ошибки (MSE) предложенного алгоритма и EWC по задачам

Накопленное забывание при последовательном обучении на четырех задачах у предложенного алгоритма тоже ниже (Рисунок 12).

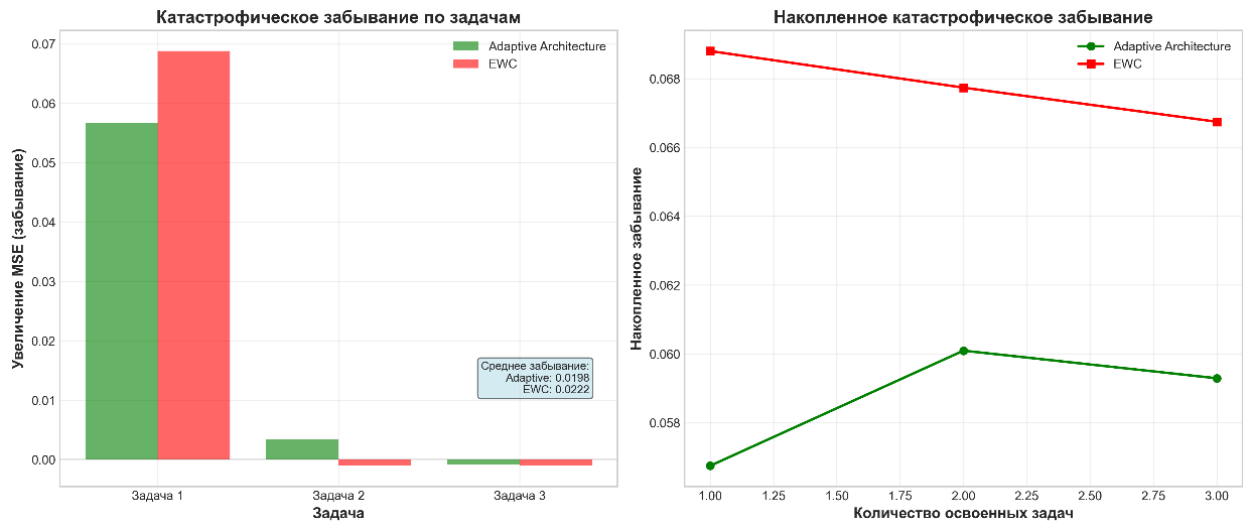


Рисунок 12 – Сравнение накопленного забывания для предложенного алгоритма и EWC

Адаптивная архитектура стабильнее на кубическом распределении (MSE 0,0417 против 0,0442), на линейных зависимостях точность сравнима (Рисунок 13).

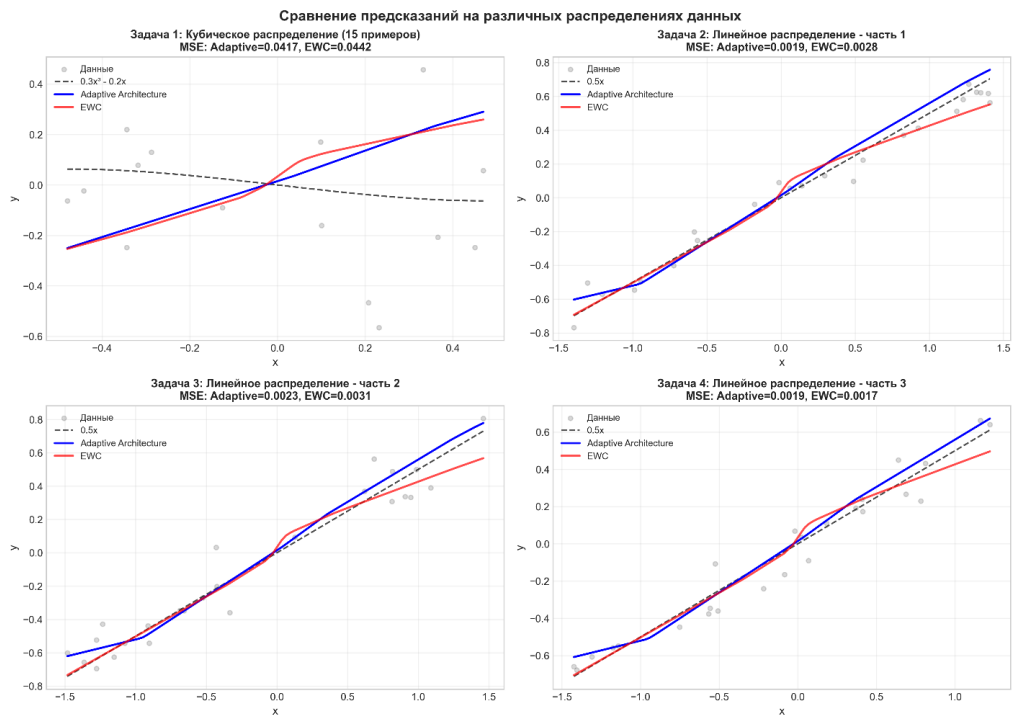


Рисунок 13 – Визуализация предсказаний

По итогам экспериментов EWC работает хуже предложенного алгоритма при значительном изменении распределения данных и при обучении более чем на двух задачах: фиксированной архитектуры и регуляризации весов ему не хватает, чтобы сохранить ранее усвоенные знания.

Выводы по главе 3

Двухфазная консолидация проверена на аппроксимации полиномиальных зависимостей и показала эффективность. У классического OS-ELM ошибка на ранее изученных данных выросла в 19,3 раза. Одна первая фаза «сна» без пересчета выходного слоя устойчивость не улучшила, потому что изменение входных весов нарушает оптимальность аналитического решения ELM. Полная двухфазная консолидация ограничила рост ошибки до 6,5 раза, что в 3 раза лучше, чем у OS-ELM; абсолютное забывание снизилось на 69,8%. Ожидаемая плата за это: лучше сохраняются старые знания, но ошибка на новых данных немного выше.

Методом Grid Search выполнен подбор оптимальных гиперпараметров модели. Наилучшая средняя точность 0,983 на синтетических потоковых задачах достигнута при числе нейронов скрытого слоя, равном 300, пороге активации фазы «сна» по изменению весов 0,02, пороге по уровню забывания 0,1, максимальном числе эпох «сна» 10, коэффициенте L2-регуляризации 0,01 и скорости обучения 0,001.

Исследование устойчивости предложенного алгоритма к катастрофическому забыванию в задаче классификации показало, что разработанный алгоритм демонстрирует относительное забывание на уровне 0,23, что ниже, чем у OS-ELM (0,35) и гибридного варианта ELM+BP без фазы «сна» (0,28). Средняя точность на последовательности из пяти задач составила 0,80 при верхней границе 0,983 для пакетной ELM. Анализ динамики точности подтвердил способность алгоритма восстанавливать производительность после введения новой задачи и сохранять знания о предыдущих задачах на приемлемом уровне. Это согласуется с разделом

3.2 и подтверждает эффективность двухфазной консолидации в многоклассовой классификации.

Динамическая структурная адаптация снизила среднеквадратическую ошибку на 65,2% при обучении на новых статистически отличных данных и сохранила устойчивость к забыванию. Число нейронов растет нелинейно: емкость интенсивно наращивается в начале обучения, потом размер сети стабилизируется. Среднее улучшение ошибки на шаге инкрементного обучения составило 53,7%: работают и структурная адаптация, и локальное дообучение. В сравнении с методом регуляризации EWC предложенный алгоритм продемонстрировал меньшую деградацию качества при последовательном обучении на четырех задачах.

Совокупность полученных экспериментальных результатов подтверждает работоспособность разработанных в главе 2 алгоритмов. Посменное инкрементное обучение с двухфазной консолидацией позволяет снизить забывание по сравнению с классическим OS-ELM; динамическая структурная адаптация дает возможность контролируемо наращивать емкость модели при поступлении новых данных.

Результаты экспериментальной верификации алгоритмов двухфазной консолидации знаний, посменного инкрементного обучения и динамической адаптации архитектуры нейронной сети, представленные в главе 3, получены автором лично и были частично опубликованы, в т. ч. в соавторстве (вклад автора более 75%) в [80; 82; 84; 89].

Определенные в ходе экспериментов оптимальные значения гиперпараметров создают основу для применения предложенного подхода к решению прикладных задач распознавания в условиях потоковых данных, чему посвящена глава 4.

ГЛАВА 4. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ ПОЛУЧЕННЫХ АЛГОРИТМОВ ПРИ РЕШЕНИИ ЗАДАЧИ РАСПОЗНАВАНИЯ АКТИВНОСТИ ЧЕЛОВЕКА

4.1 Постановка задачи распознавания физической активности человека и описание экспериментальных данных

Экспериментальное исследование разработанных алгоритмов будет выполняться на задаче распознавания физической активности человека (Human Activity Recognition, HAR), заключающейся в автоматическом определении вида деятельности человека по данным встроенных сенсоров мобильных устройств [94; 95].

Задача HAR по данным инерциальных сенсоров мобильных устройств формулируется как задача многоклассовой классификации временных рядов в условиях потокового поступления данных.

Пусть задан непрерывный поток данных $D = \{D_1, D_2, \dots, D_t, \dots\}$, где $D_t = \{(x_{i,t}, y_{i,t})\}_{i=1}^{N_t}$ – пакет данных, поступающий в систему в дискретный момент времени t . В данном выражении $x_{i,t}$ представляет собой d -мерный вектор признаков, извлеченный из сенсорных сигналов за фиксированное временное окно; $y_{i,t} \in \{1, 2, \dots, C\}$ – истинная метка класса активности; N_t – количество образцов в t -м пакете.

Принципиальная особенность рассматриваемой постановки состоит в том, что обучение модели должно осуществляться инкрементно, по мере поступления новых данных, без возможности единовременного доступа ко всей обучающей выборке. Данное ограничение соответствует реальным условиям эксплуатации носимых устройств, где накопление и хранение полной истории наблюдений невозможно вследствие ограничений по объему памяти и энергопотреблению. Высокая размерность признакового пространства и нелинейная природа сенсорных данных позволяют объективно оценить эффективность разработанных

инкрементных алгоритмов на базе машины экстремального обучения с механизмами консолидации памяти.

Задача HAR входит в число самых востребованных в интеллектуальном анализе данных: у нее широкий круг практических приложений (Рисунок 14): медицина и здравоохранение: дистанционный мониторинг пациентов, телереабилитация [96; 97], контроль выполнения физических упражнений, системы напоминания для пациентов с деменцией [98]; фитнес и спорт: мониторинг физической активности, трекеры активности [99], анализ спортивных движений [100]; производственная безопасность: контроль соблюдения техники безопасности, мониторинг действий персонала [101]; безопасность и охрана: обнаружение падений [102], распознавание агрессивного поведения [103], анализ действий военных и сотрудников патрульных служб.

Области применения метода распознавания физической активности человека

Медицина	Фитнес и спорт	Производство и услуги	Безопасность
• Физическая телереабилитация (VirtualRehab Body) • Контекстная система напоминания для выполнения ежедневной деятельности пожилым людям с деменцией (Du K., Zhang D.,)	• Приложения-трекеры • Мониторинг уровня физической активности (Fitbit) • Мониторинг конкретных видов спорта (Bächlin M., Kusserow M)	• Распознавание активности работников на интеллектуальном производстве (Tao W., Lai Z-H.) • Системы распознавания видов активности для мониторинга персонала (Favela J., Tentory M.)	• Автоматическая генерация отчетов после патрулирования (Minnen D., Westeyn T.) • Обнаружение падения человека (Putra I.P.E.S., Brusey J.) • Система распознавания агрессивного поведения человека (Brémond F., Thonnat M.)

Рисунок 14 – Области применения распознавания активности человека

Поэтому нужны алгоритмы, которые дают высокую точность на динамически меняющихся данных и при ограниченных вычислительных ресурсах.

Задача HAR выбрана тестовым полигоном для алгоритмов инкрементного обучения на основе ELM по нескольким причинам.

Во-первых, данные инерциальных сенсоров поступают непрерывным потоком, что в точности соответствует инкрементному обучению: модель адаптируется к новым наблюдениям без полного переобучения на накопленных данных. В классификации статических изображений или текстов обучающая выборка обычно фиксирована, а в HAR данные не ограничены во времени, и их объем постоянно растет.

Во-вторых, распределение признаков одних и тех же классов активности может заметно меняться со временем: устройство смещается на теле, у пользователя свои двигательные паттерны, калибровка сенсоров меняется, характеристики датчиков деградируют. Из-за этого при наивном дообучении возникает катастрофическое забывание: сеть адаптируется к новым данным с изменившимся распределением и перестает правильно классифицировать ранее изученные активности. Значит, HAR дает естественный и практически значимый сценарий для проверки механизмов защиты от забывания.

В-третьих, носимые устройства, на которых будут работать системы распознавания, жестко ограничены по вычислительным ресурсам и памяти. Поэтому предпочтительны методы с аналитическим решением, которым не нужна итеративная оптимизация всех параметров и хранение больших объемов исторических данных. Алгоритмы на основе машины экстремального обучения этим требованиям удовлетворяют.

Наконец, у HAR естественная иерархия классов: статические активности (сидение, стояние) и динамические (ходьба, бег, велосипед). Это позволяет содержательно интерпретировать забывание и перенос знаний: видно, какие типы активности забываются легче всего, а какие при обучении новым задачам дают положительный перенос.

Так что задача HAR позволяет проверить сразу все ключевые свойства алгоритмов: способность к инкрементному обучению, устойчивость к катастрофическому забыванию, адаптацию к изменению распределения данных и вычислительную эффективность ELM.

Для экспериментов выбран набор данных [104] с записями движений человека при разных видах физической активности. Отобраны простые активности: ходьба, бег трусцой, сидение, стояние и езда на велосипеде, то есть $C = 5$ классов. Эти активности чаще всего встречаются в повседневной жизни и лежат в основе большинства прикладных задач мониторинга.

Данные собраны у девяти испытуемых, каждый из которых выполнял каждую активность от 3 до 4 минут, что учитывает индивидуальные различия. Сигналы регистрировались одновременно с трех инерциальных измерительных модулей (IMU) на запястье, груди и лодыжке. Такие позиции выбраны из практических соображений: при реальной эксплуатации датчики удобнее и незаметнее всего размещать на конечностях или туловище [105].

Каждый модуль давал показания трех датчиков: акселерометра (оси A_x, A_y, A_z), гироскопа (G_x, G_y, G_z) и магнитометра (M_x, M_y, M_z). Всего $d = 9$ числовых признаков на каждый момент времени. Общий объем выборки 630000 наблюдений.

По описательной статистике и корреляциям признаков (Рисунки 15, 16) наибольший разброс у признаков акселерометра: тело естественно ускоряется при динамических активностях. Пропущенных значений нет. Между отдельными парами признаков есть умеренные и сильные связи, то есть признаковое пространство избыточно. Но, как показано в [106], при использовании ELM и инкрементных алгоритмов сокращение размерности точность почти не повышает, зато затраты на предобработку растут. Поэтому используется полный набор из 9 признаков без отбора и преобразований.

	Ax	Ay	Az	Gx	Gy	Gz	Mx	My	Mz
count	630000.0	630000.0	630000.0	630000.0	630000.0	630000.0	630000.0	630000.0	630000.0
mean	-0.0005	-0.0002	9.7994	0.0003	-0.0003	-0.0001	49.9981	49.9997	50.0002
std	4.8499	3.8254	4.8717	0.1456	0.1457	0.1457	3.5715	3.5715	3.5708
min	-17.4936	-13.5154	-6.1072	-0.4973	-0.502	-0.5008	42.7782	42.8577	42.8278
25%	-2.6023	-2.0902	7.1533	-0.107	-0.1075	-0.1073	46.5022	46.5041	46.5034
50%	0.0003	0.0012	9.7981	0.0	-0.0001	-0.0004	49.9986	49.9934	49.9998
75%	2.6025	2.0866	12.4453	0.1073	0.107	0.1068	53.4979	53.4948	53.4976
max	17.3735	13.6357	25.7208	0.4868	0.4942	0.4756	56.9543	56.8873	57.0197

Рисунок 15 – Статистические характеристики признаков набора данных

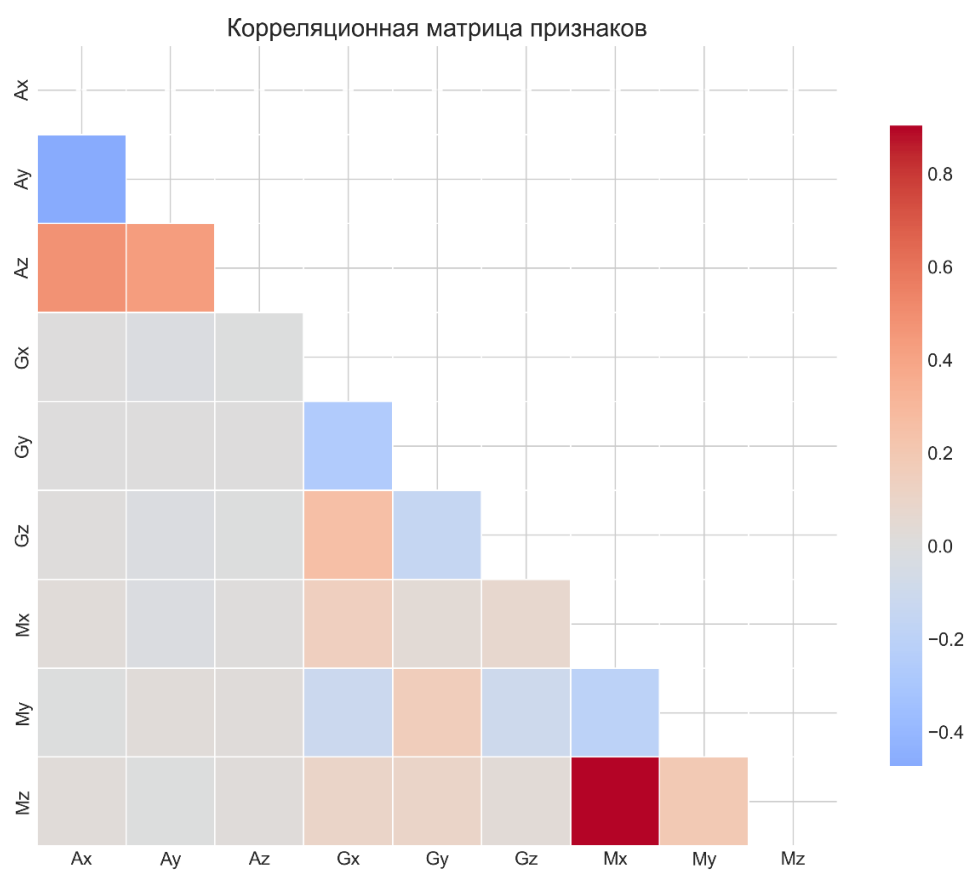


Рисунок 16 – Корреляционная матрица признаков набора данных

Объема и структуры набора достаточно, чтобы организовать потоковое обучение в виде последовательности из пяти задач.

4.2 Методика проведения экспериментальных исследований

Эксперимент строится по рекомендациям [53; 90] и следует стандартному протоколу оценки инкрементного обучения. Исходный набор делится на последовательность непересекающихся подмножеств (задач). Так моделируется реальная эксплуатация: у модели нет одновременного доступа ко всем обучающим примерам, и она адаптируется к данным по мере поступления.

Чтобы алгоритмы можно было корректно сравнивать и результаты воспроизводить, все данные предобрабатываются одинаково. Сначала признаки стандартизируются (Z-нормализация). Среднее и стандартное отклонение вычисляются только по обучающей выборке первой задачи, и все последующие задачи нормализуются теми же параметрами. Это моделирует ситуацию, когда на этапе инициализации модель не видит будущих данных, исключает утечку информации из последующих задач и соответствует реальной эксплуатации, где статистика данных со временем меняется.

Исходный набор из $N = 630000$ наблюдений делится на $T = 5$ последовательных непересекающихся задач. Каждая задача $D(t)$ это временной срез исходного набора из $N(t)$ наблюдений, в нем есть представители всех $C = 5$ классов. Объем задачи:

$$N^{(t)} = \frac{N}{T}, t = 1, \dots, T,$$

при $N = 630000$ и $T = 5$ получаем $N(t) = 126000$ наблюдений на задачу.

Такая схема соответствует условиям эксплуатации систем мониторинга: в любой момент могут наблюдаться все виды активности, но их статистические свойства и распределение со временем меняются.

Каждая задача стратифицированно делится на обучающую и тестовую выборки в пропорции 75% и 25% с сохранением долей классов. Для этого используется `StratifiedShuffleSplit`: данные внутри задачи случайно перемешиваются, затем делятся с сохранением исходного распределения классов. Размеры выборок каждой задачи:

$$N_{train}^{(t)} = 0,75 \cdot N^{(t)} = 94500, N_{test}^{(t)} = 0,25 \cdot N^{(t)} = 31500.$$

Общий объем обучающей выборки 472500 наблюдений, тестовой 157500.

Тестовые выборки всех задач доступны на протяжении всего эксперимента, поэтому можно следить, как меняется качество распознавания по мере изучения новых задач. Это стандартное требование протоколов оценки инкрементного обучения: оно позволяет построить полную матрицу точности R и вычислить интегральные метрики, такие как забывание и прямой/обратный перенос.

Для каждого алгоритма сначала выполняется начальная инициализация на первых 800 примерах обучающей выборки первой задачи. Объем выбран по трем соображениям.

Первое: 800 примеров это минимум, достаточный для устойчивого начального приближения весов. По [49; 61], для корректной работы ELM число обучающих примеров должно превышать число нейронов скрытого слоя минимум в 2-3 раза. При $n = 800$ и $L = 200$ скрытых нейронов условие выполняется с запасом ($n \gg L$), поэтому матрица $U^T U$ невырождена и ее можно обратить.

Второе: этот объем статистически репрезентативен для каждого из пяти классов. Стратифицированная выборка гарантирует, что в инициализацию попадут все классы, в том числе миноритарные (не менее 114 примеров каждого). Этого достаточно для статистически значимых оценок ковариационных матриц и устойчивого определения начальных границ между классами в пространстве признаков.

Третье: по [8], инициализация должна давать достаточно информации для базового представления данных, но не быть слишком большой, иначе нарушаются условия инкрементного обучения, при которых модель адаптируется к данным по мере их поступления, а не получает большой объем сразу. Инициализация выполняется по алгоритму из раздела 2.2.

После завершения инициализации начинается основной этап инкрементного обучения. Оставшиеся данные обучающей выборки первой задачи, а также данные всех последующих задач подаются на вход модели последовательными батчами фиксированного размера $B=64$. Выбор размера батча обусловлен компромиссом между вычислительной эффективностью и статистической устойчивостью

обновлений [107]. Для первой задачи обрабатывается 1465 батчей, для каждой последующей – 1477 батчей.

На каждом батче выполняется следующий комплекс операций:

- обновление весов модели. Для инкрементного ELM используется рекурсивное RLS-обновление с коэффициентом забывания $\gamma = 0,999$, позволяющее адаптировать выходные веса к новым данным с постепенным устареванием предыдущей информации. Для методов с обратным распространением обновление выполняется с использованием оптимизатора Adam.

- обновление буфера воспроизведения. Для алгоритмов, предусматривающих буфер (ELM with Replay, Proposed), примеры текущего батча добавляются в буфер фиксированного размера $K = 2000$. При заполнении буфера замещение выполняется по стратегии стратифицированного резервуарного семплирования [108], обеспечивающей равномерное представительство всех классов активности.

- проверка условий перехода в режим консолидации. Для алгоритмов с режимом «сна» (ELM with Replay + Sleep, Proposed) переход в режим консолидации выполняется по фиксированному детерминированному критерию – при достижении заданного числа батчей с момента последней консолидации. Фиксированный критерий выбран ради воспроизводимости результатов и корректного сравнения методов в контролируемых условиях. Адаптивные критерии (по порогу ошибки или обнаружению статистического дрейфа) из раздела 2.6 остаются перспективным направлением для реальных систем с меняющейся средой, но им нужна дополнительная настройка порогов, и в этом исследовании они не рассматриваются;

- выполнение процедуры консолидации знаний. При выполнении условий перехода запускается двухфазный алгоритм режима «сна» (раздел 2.5). В фазе 1 выполняется 15 эпох настройки входных весов V и смещений b обратным распространением по данным из буфера воспроизведения и псевдопримерам, с ранней остановкой против переобучения. В фазе 2 выполняется пересчет выходных весов W

на основе обновленной матрицы скрытого слоя, после чего обновляется обобщенная корреляционная матрица P , используемая для последующих RLS-обновлений.

– адаптивное расширение архитектуры. Для адаптивной версии алгоритма (Proposed_Adaptive_Improved) при устойчивом превышении ошибкой порогового значения в течение 10 последовательных батчей в скрытый слой добавляется 15 новых нейронов. Процедура расширения включает генерацию новых весов и смещений с последующим расширением всех матриц модели с сохранением ранее накопленной информации. Общее число добавляемых нейронов ограничено 300.

После обработки всех батчей текущей задачи выполняется этап оценки качества на тестовых выборках. В результате формируется матрица точности R , имеющая следующий вид:

$$R = \begin{pmatrix} R_{11} & R_{12} & R_{13} & \cdots & R_{1T} \\ 0 & R_{22} & R_{23} & \cdots & R_{2T} \\ 0 & 0 & R_{33} & \cdots & R_{3T} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & R_{TT} \end{pmatrix},$$

где элемент $R_{ij} \in [0,1]$ представляет собой точность классификации на тестовой выборке задачи i после изучения задачи j . Диагональные элементы R_{ii} соответствуют точности на текущей задаче непосредственно после ее изучения. Элементы R_{ij} при $j > i$ отражают сохранность знаний по задаче i после изучения последующих задач. На основе матрицы R вычисляются все интегральные метрики качества инкрементного обучения, представленные в Таблице 15.

Таблица 15 – Метрики оценки эффективности инкрементного обучения

Метрика	Обозначение	Формула	Назначение
Средняя точность	AvgAcc	$\frac{1}{T} \sum_{i=1}^T R_{ii}$	Усредненная точность на текущих задачах
Конечная средняя точность	FinalAvgAcc	$\frac{1}{T} \sum_{i=1}^T R_{iT}$	Итоговая точность по всем изученным задачам

Продолжение таблицы 15

Метрика	Обозначение	Формула	Назначение
Забывание	Forgetting	$\frac{1}{T-1} \sum_{i=1}^{T-1} (\max_j R_{ij} - R_{iT})$	Снижение точности по ранее изученным задачам
Обратный перенос	BWT	$\frac{1}{T-1} \sum_{i=1}^{T-1} (R_{iT} - R_{ii})$	Влияние новых задач на качество по старым
Прямой перенос	FWT	$\frac{1}{T-1} \sum_{i=2}^T (R_{ii} - R_{i0})$	Влияние старых задач на обучение новым
Макро-F1	Macro-F1	$\frac{1}{C} \sum_{c=1}^C F1_c$	Сбалансированность классификации по классам

Forgetting показывает устойчивость к катастрофическому забыванию: чем меньше значение, тем лучше модель хранит знания о прежних задачах, а Forgetting = 0 означает, что забывания нет. BWT отражает влияние новых задач на качество распознавания старых: положительное значение означает улучшение на старых задачах. FWT показывает, насколько знание предыдущих задач ускоряет обучение новым.

Разработанный алгоритм сравнивается с существующими подходами (Таблица 16).

MLP включен как верхняя граница качества: он показывает точность, достижимую при наличии всех данных и полном переобучении. Он не инкрементален и в потоковом режиме без полного переобучения на каждом шаге работать не может.

Таблица 16 – Сравнимые алгоритмы и их назначение

Группа алгоритмов	Алгоритм	Назначение
Верхняя граница качества (Upper Bound)	MLP	Классическая нейронная сеть с обратным распространением. Пакетное обучение на всех данных. Показывает потенциально достижимую точность. Не является инкрементальным методом.

Продолжение таблицы 16

Группа алгоритмов	Алгоритм	Назначение
Базовые инкрементные	Batch ELM	Пакетное обучение машины экстремального обучения (все данные доступны)
	Online ELM	Последовательное инкрементальное обучение без буфера и консолидации
С буферизацией	ELM with Replay	Инкрементальный ELM с буфером воспроизведения
	ELM with Replay + Sleep	Инкрементальный ELM с буфером и двухфазной консолидацией (режим «сна»)
Предлагаемые	Proposed	Полный разработанный алгоритм (базовая конфигурация)
	Proposed_Optimal	Оптимизированная конфигурация с подобранными гиперпараметрами
	Proposed_Adaptive_Improved	Адаптивная версия с расширением архитектуры и детекцией дрейфа
Современные методы (SOTA)	EWC	Регуляризация по важности весов
	Experience Replay	Классический буферный метод воспроизведения опыта
	SI	Регуляризация по синаптической важности

EWC, Experience Replay и SI это современные методы инкрементного обучения, которые широко используются как бенчмарки [31; 32; 35]. Обобщенные параметры экспериментальной конфигурации сведены в Таблицу 17.

Таблица 17 – Параметры экспериментальной конфигурации

Параметр	Обозначение	Значение
Число задач	T	5
Общий объем выборки	N	630000
Размер задачи	$N^{(t)}$	126000
Размер обучающей/тестовой выборки	$N_{train}^{(t)}/N_{test}^{(t)}$	94500/31500

Продолжение таблицы 17

Параметр	Обозначение	Значение
Размер инициализации	N_{init}	800
Размер батча	B	64
Размер буфера воспроизведения	K	2000
Коэффициент забывания	γ	0,999
Период консолидации	Δ_{sleep}	20
Коэффициент регуляризации	λ	0,001
Число повторений	$N_{repeats}$	3

Все вычисления проводились на вычислительной платформе со следующими характеристиками: процессор 12th Gen Intel(R) Core(TM) i5-12500H (12 ядер, 16 потоков, базовая частота 3.10 ГГц), оперативная память 8 ГБ, операционная система Windows 11 Pro (64-разрядная).

Программная реализация выполнена на языке Python с использованием библиотек NumPy для матричных операций и линейной алгебры, SciPy для решения систем линейных уравнений и вычисления собственных значений, scikit-learn для предобработки данных, стратифицированного разделения выборок и расчета метрик качества, PyTorch для реализации моделей сравнения (MLP, EWC, ER, SI), Matplotlib и Seaborn для визуализации результатов, Pandas для работы с табличными данными.

Выполнение экспериментов осуществлялось в среде Jupyter Notebook, что обеспечивало интерактивную отладку и визуализацию промежуточных результатов на всех этапах вычислительного эксперимента. Для обеспечения воспроизводимости результатов во всех экспериментах использовался фиксированный seed генератора случайных чисел (`random_state = 42`).

Для обеспечения статистической достоверности каждый эксперимент повторялся три раза с различными начальными значениями весов (за счет изменения seed генератора случайных чисел). По результатам повторений для

каждой метрики вычисляются среднее значение, стандартное отклонение и 95% доверительный интервал. В таблицах результаты представляются в формате «среднее $\pm$ стандартное отклонение».

Для обеспечения корректной интерпретации результатов все эксперименты настоящей главы выполнялись в одной из трех постановок, различающихся объемом доступных данных и режимом обучения.

Постановка А – пакетное обучение (offline). Модель обучается единовременно на полном объеме данных всех задач. Данная постановка не является инкрементной и используется исключительно для определения верхней границы качества классификации. В этой постановке исследуются методы MLP и Batch ELM. Результаты постановки А принципиально не могут быть воспроизведены в условиях потокового поступления данных и приводятся только как эталонные.

Постановка В – инкрементное обучение. Данные поступают последовательно, задача за задачей. Модель обучается инкрементно, без возможности повторного доступа к данным предыдущих задач. Тестирование выполняется на тестовых выборках всех задач, изученных к текущему моменту. Все результаты, полученные в постановке В, сравнимы между собой, поскольку получены в идентичных условиях.

Постановка С – стрессовое инкрементное обучение (permuted features). Отличается от постановки В тем, что признаки в каждой задаче подвергаются случайной перестановке. Данная постановка моделирует наиболее жесткие условия эксплуатации, когда статистические свойства входных сигналов радикально изменяются от задачи к задаче. Результаты постановок В и С сравниваются между собой для оценки робастности методов и алгоритмов к изменению распределения данных.

В последующих разделах перед представлением результатов каждого эксперимента явно указывается, в какой постановке он выполнен.

Изложенная методика задает единую схему проведения экспериментов, обеспечивающую корректность и воспроизводимость результатов. Все

последующие разделы настоящей главы выполнены в соответствии с описанными принципами: в разделе 4.3 обосновывается выбор ELM в качестве базовой модели путем сравнения с методом обратного распространения ошибки; в разделе 4.4 проводится сравнительное исследование разработанных инкрементных алгоритмов; в разделе 4.5 анализируется вклад отдельных компонентов разработанного алгоритма; в разделе 4.6 выполняется сравнение с современными подходами инкрементного обучения; в разделе 4.7 исследуется вычислительная эффективность и влияние ключевых гиперпараметров. Предложенная методика позволяет объективно оценить эффективность разработанных алгоритмов при решении задачи распознавания физической активности человека в условиях потокового поступления данных.

4.3 Сравнительное исследование базовых моделей обучения

Целью эксперимента являлась сравнительная оценка эффективности двух базовых подходов к обучению нейронных сетей – ELM и BP – при решении задачи распознавания физической активности человека. Данное исследование необходимо для обоснования выбора ELM в качестве основы для разработанных в главе 2 инкрементных алгоритмов [109].

Эксперимент выполнен в постановке А – пакетное обучение (offline). Оба метода (ELM и BP) обучаются одновременно на полном объеме обучающей выборки, без ограничений на количество эпох и объем доступных данных. Данная постановка не является инкрементной и используется исключительно для определения верхней границы качества классификации и сравнительной оценки вычислительной эффективности базовых подходов к обучению. Результаты настоящего раздела не следует напрямую сопоставлять с результатами инкрементных методов, исследуемых в разделах 4.4–4.7, поскольку условия экспериментов принципиально различны.

Эксперимент проводился на наборе данных, описанном в разделе 4.1, с соблюдением единой схемы стратифицированного разделения на обучающую и тестовую выборки. Число нейронов скрытого слоя варьировалось в диапазоне от 100 до 2000.

Время выполнения алгоритмов измерялось с использованием функции `time.time()` модуля `time` языка Python и включало полный цикл обучения модели и формирования прогноза на тестовой выборке.

Результаты сравнения точности классификации представлены в Таблице 18.

Таблица 18 – Доля правильных ответов машины экстремального обучения и метода обратного распространения ошибки при разном числе нейронов скрытого слоя

Метод	Число нейронов скрытого слоя					
	100	300	500	1000	1500	2000
ELM	0,531 ± 0,003	0,603 ± 0,008	0,639 ± 0,008	0,691 ± 0,003	0,718 ± 0,003	0,732 ± 0,002
BP	0,930 ± 0,001	0,936 ± 0,000	0,937 ± 0,001	0,937 ± 0,001	0,938 ± 0,000	0,938 ± 0,000

Точность ELM монотонно растет с числом нейронов: с 0,531 при 100 до 0,732 при 2000, всего на 0,201. Быстрее всего она растет от 100 до 1000 нейронов (+0,160), а от 1000 до 2000 прирост только 0,041: наступает насыщение. BP выходит почти на максимум уже при 300 нейронах (0,936), дальше держится около 0,938 и превосходит ELM при 2000 нейронах на 0,206.

Стандартное отклонение BP мало (не более 0,001), то есть метод стабилен. У ELM на малых архитектурах разброс заметный (до 0,008 при 300 и 500 нейронах): сказывается случайная инициализация входных весов. При 2000 нейронах он падает до 0,002, модель стабилизируется.

F1-score приведен в Таблице 19.

Таблица 19 – F1-score машины экстремального обучения и метода обратного распространения ошибки при разном числе нейронов скрытого слоя

Метод	Число нейронов скрытого слоя					
	100	300	500	1000	1500	2000
ELM	0,454 ± 0,006	0,560 ± 0,011	0,609 ± 0,010	0,674 ± 0,003	0,705 ± 0,004	0,720 ± 0,003
BP	0,930 ± 0,001	0,936 ± 0,000	0,937 ± 0,001	0,937 ± 0,000	0,938 ± 0,000	0,938 ± 0,000

F1-мера ведет себя так же, как точность. У ELM она растет с 0,454 до 0,720 при увеличении числа нейронов с 100 до 2000: классификация по всем классам становится более сбалансированной. У BP F1 стабильна на уровне от 0,930 до 0,938.

Время обучения приведено в Таблице 20.

Таблица 20 – Время обучения (в секундах)

Метод	Число нейронов скрытого слоя					
	100	300	500	1000	1500	2000
ELM	1,59 ± 0,53	5,76 ± 0,97	12,17 ± 2,58	33,14 ± 3,92	75,20 ± 8,14	134,50 ± 15,23
BP	258,31 ± 1,97	915,34 ± 2,93	1386,22 ± 15,10	2539,10 ± 41,08	3920,45 ± 55,30	5320,80 ± 78,15

По времени методы различаются кардинально. При 100 нейронах ELM обучается за 1,59 с, BP за 258,31 с: ELM быстрее в 162 раза. При 500 нейронах ELM занимает 12,17 с, BP 1386,22 с (около 23 минут), выигрыш в 114 раз. При 1000 нейронах это 33,14 с против 2539,10 с (больше 42 минут), выигрыш в 77 раз. При 2000 нейронах ELM требует 134,50 с (около 2 минут 15 секунд), BP 5320,80 с (больше 88 минут), выигрыш в 40 раз.

Сравнение показано на Рисунке 17.

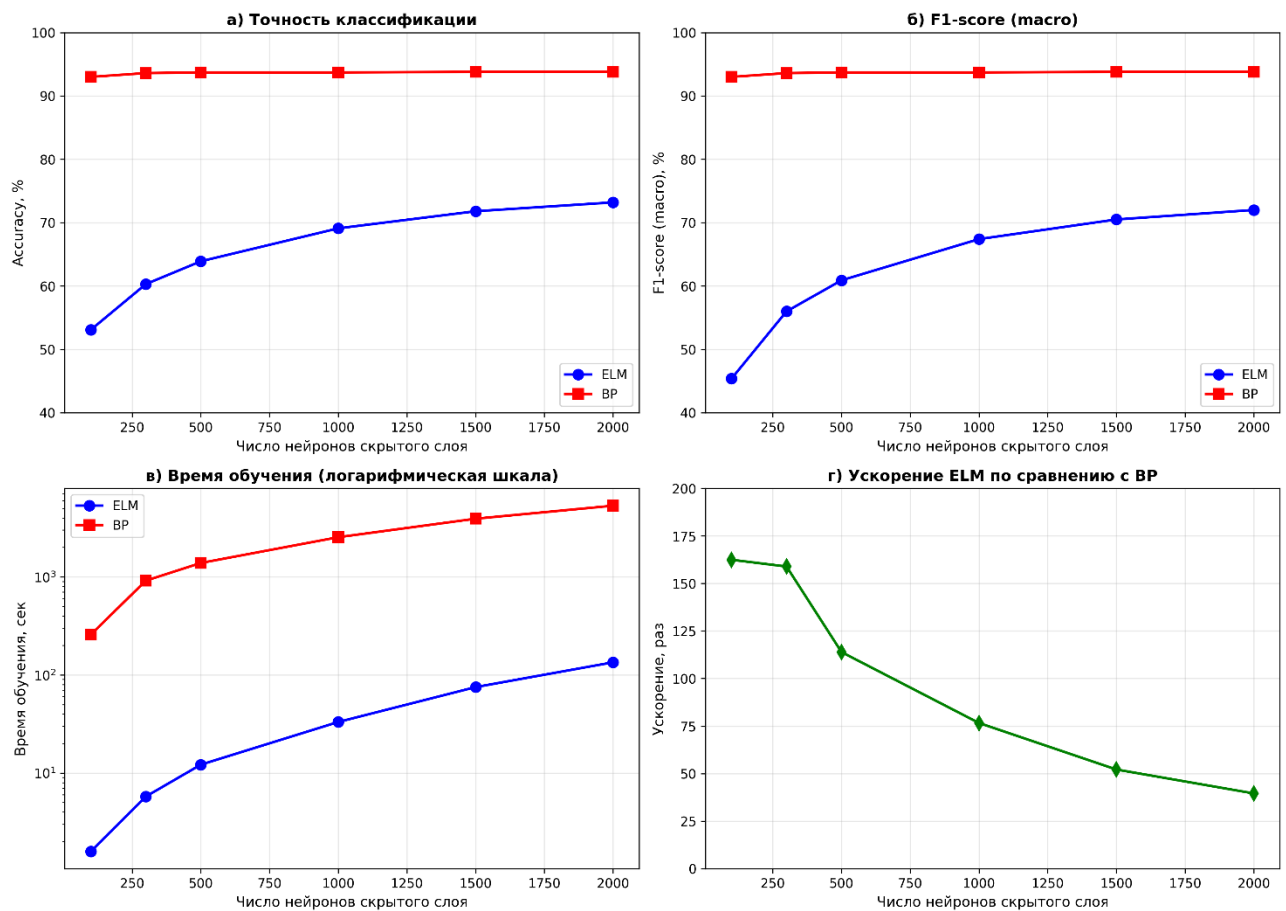


Рисунок 17 – Сравнение ELM и BP по точности, F1-score, времени обучения и ускорению: а – точность классификации; б – F1-score (weighted); в – время обучения; г – ускорение ELM по сравнению с BP

ELM уступает BP по точности (0,732 против 0,938 при 2000 нейронах), но разрыв сокращается с 0,399 до 0,206 при росте числа нейронов со 100 до 2000: с увеличением архитектуры ELM улучшается.

ELM обучается в 40-162 раза быстрее, в зависимости от числа нейронов, потому что выходные веса считаются аналитически, без итерационной оптимизации.

Алгоритмы из глав 2 и 3 рассчитаны на инкрементное обучение в потоковом режиме, где критична скорость адаптации, поэтому выбор ELM в качестве базовой модели обоснован.

4.4 Исследование эффективности разработанного алгоритма инкрементного обучения

Здесь подтверждается эффективность алгоритма инкрементного обучения на основе машины экстремального обучения с буферизацией, генерацией псевдопримеров, двухфазной консолидацией и адаптивным расширением архитектуры. Для этого анализируются последовательные модификации базового инкрементального ELM: каждая добавляет один компонент, и можно количественно оценить вклад каждого в общую производительность [110].

Эксперимент выполнен в постановке В – инкрементное обучение. В отличие от раздела 4.3, где методы ELM и ВР обучались в пакетном режиме на полном объеме данных, в настоящем разделе все алгоритмы функционируют в условиях последовательного поступления данных: модель получает доступ к обучающим выборкам задач поочередно, не имея возможности вернуться к данным предыдущих задач. Все сравниваемые алгоритмы находятся в равных условиях: одинаковый размер батча ($B = 64$), одинаковый размер буфера воспроизведения ($K = 2000$ для методов, его использующих), одинаковые параметры предобработки данных. Данная постановка является основной для оценки эффективности разработанных алгоритмов, поскольку соответствует целевому сценарию эксплуатации – потоковому распознаванию физической активности на носимых устройствах.

В соответствии с методикой, описанной в разделе 4.2, эксперименты проводились на наборе данных HAR, разделенном на пять последовательных задач. Каждый алгоритм обучался инкрементно, т.е. данные каждой последующей задачи подавались после завершения обучения на предыдущей, без доступа к полной истории. Оценка качества выполнялась после изучения каждой задачи на тестовых выборках всех ранее изученных задач. Для обеспечения статистической достоверности каждый эксперимент повторялся три раза; в таблицах приведены средние значения и стандартные отклонения.

В Таблице 21 представлены интегральные показатели качества для всех исследуемых конфигураций. Визуальное сопоставление результатов приведено на Рисунке 18.

Таблица 21 – Сравнительные результаты инкрементных алгоритмов на наборе данных HAR

Алгоритм	AvgAcc	FinalAvg Acc	Forgetting	BWT	FWT	Macro-F1	Time (сек)
MLP (верхняя граница)	0,808 ± 0,000	0,808 ± 0,000	0,000 ± 0,000	0,039 ± 0,000	-0,819 ± 0,000	0,843 ± 0,000	–
Batch ELM	0,513 ± 0,000	0,513 ± 0,000	0,001 ± 0,000	0,000 ± 0,000	-0,513 ± 0,000	0,416 ± 0,001	7,62 ± 0,28
Online ELM	0,517 ± 0,000	0,516 ± 0,001	0,002 ± 0,001	0,001 ± 0,001	-0,517 ± 0,000	0,424 ± 0,002	11,64 ± 0,04
ELM with Replay	0,517 ± 0,000	0,516 ± 0,001	0,002 ± 0,001	0,001 ± 0,001	-0,517 ± 0,000	0,424 ± 0,002	72,16 ± 0,01
ELM with Replay + Sleep	0,736 ± 0,003	0,759 ± 0,003	0,000 ± 0,000	0,024 ± 0,003	-0,742 ± 0,005	0,754 ± 0,004	528,45 ± 11,80
Proposed (базовый)	0,741 ± 0,003	0,759 ± 0,001	0,000 ± 0,000	0,024 ± 0,004	-0,748 ± 0,003	0,756 ± 0,004	336,43 ± 8,61
Proposed_Opt imal	0,771 ± 0,001	0,780 ± 0,002	0,001 ± 0,000	0,014 ± 0,002	-0,776 ± 0,001	0,778 ± 0,002	1586,39 ± 44,47
Proposed_Ad aptive_Impro ved	0,762 ± 0,002	0,776 ± 0,003	0,000 ± 0,000	0,017 ± 0,003	-0,768 ± 0,002	0,773 ± 0,004	914,11 ± 17,78

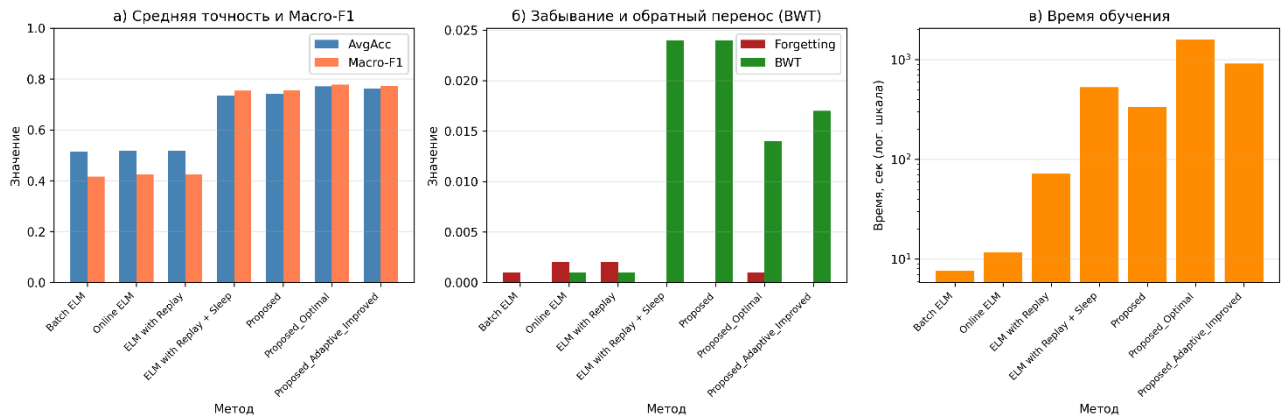


Рисунок 18 – Сравнительные результаты инкрементных алгоритмов

MLP, обученный в пакетном режиме на всех данных, показывает точность 0,808 и представляет собой верхнюю границу качества, потенциально достижимую при отсутствии ограничений инкрементального обучения. Данный метод не является инкрементным и приводится исключительно как эталон.

Под Batch ELM в настоящем разделе понимается пакетное обучение ELM отдельно на каждой задаче инкрементного обучения без сохранения информации о предыдущих задачах. Данная постановка отличается от пакетного обучения на полном объеме данных, рассмотренного в разделе 4.3, где ELM достигал точности 0,732. Результат 0,513, полученный здесь, отражает принципиальную невозможность качественного решения задачи без механизмов накопления и консолидации знаний.

Online ELM с RLS-обновлением весов немного лучше Batch ELM (0,517 против 0,513), но у него положительное forgetting (0,002) и отрицательный обратный перенос, то есть качество по старым задачам деградирует по мере изучения новых. Буфер воспроизведения (ELM with Replay) точность заметно не повышает (0,517): без консолидации буфер сам по себе качество не улучшает, его данные не используются для дообучения. Время обучения при этом вырастает с 11,6 до 72,2 с из-за операций добавления и семплирования.

Двухфазная консолидация (ELM with Replay + Sleep) дает скачок: средняя точность растет с 0,517 до 0,736, конечная средняя до 0,759. Forgetting падает до

нуля, то есть катастрофического забывания нет. Обратный перенос становится положительным ($BWT = 0,024$): качество распознавания старых классов улучшается при изучении новых. Время обучения растет до 528,5 с, потому что процедура «сна» с 15 эпохами обратного распространения вычислительно затратна.

Базовая конфигурация Proposed повышает точность до 0,741 и сокращает время обучения с 528,5 до 336,4 с (на 36%). Объяснение такое: псевдопримеры делают данные на этапе консолидации информативнее, и сопоставимое качество достигается за меньшее число эпох. Оптимальная конфигурация Proposed_Optimal с буфером 5000, 1000 псевдопримеров и 30 эпохами ВР дает наивысшую точность среди инкрементальных методов: 0,771 по средней и 0,780 по конечной средней точности. Macro-F1 равен 0,778, то есть классификация по классам сбалансирована. От верхней границы (MLP) метод отстает на 0,037, и это приемлемая плата за инкрементальность. Время обучения растет до 1586,4 с: за качество приходится платить вычислениями.

Proposed_Adaptive_Improved с динамическим расширением архитектуры показывает точность 0,762 за 914,1 с, это самая сбалансированная конфигурация для практики. Нейроны добавляются автоматически при обнаружении дрейфа, поэтому архитектуру не нужно подбирать вручную, алгоритм сам адаптируется к изменению распределения.

Матрицы точности ключевых алгоритмов, то есть эволюция качества по мере изучения задач, показаны на Рисунке 19.

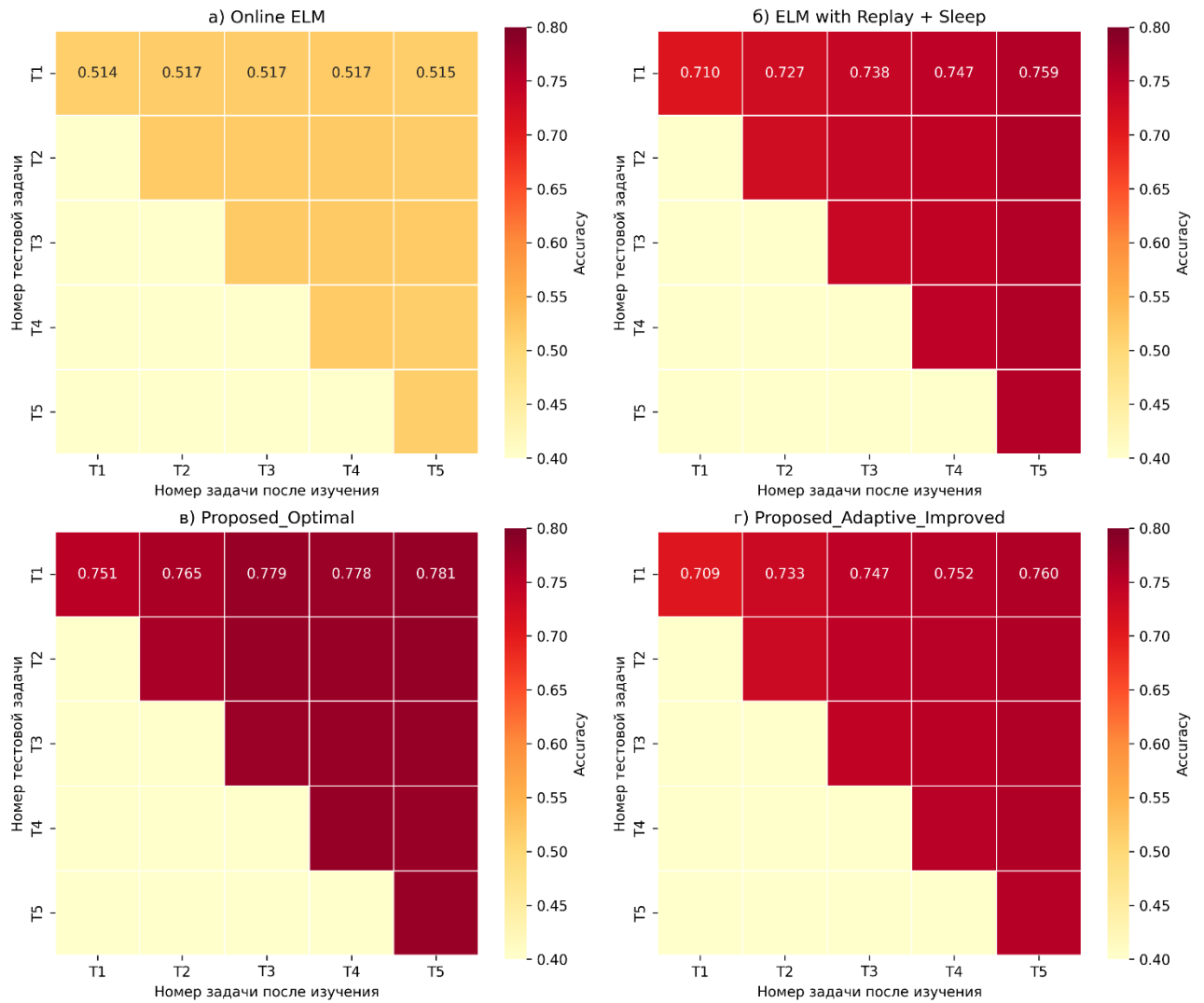


Рисунок 19 – Матрицы точности

У Online ELM точность на старых задачах устойчиво снижается по мере изучения новых: это проявление катастрофического забывания. У алгоритмов с консолидацией (Replay + Sleep и выше) точность на старых задачах не только не падает, но местами растет (обратный перенос), то есть двухфазный «сон» сохраняет и улучшает накопленные знания. Proposed_Optimal имеет самые высокие значения на диагонали и в последнем столбце: у него лучший баланс пластичности и стабильности.

Динамика средней точности по мере изучения последовательных задач для всех алгоритмов показана на Рисунке 20.

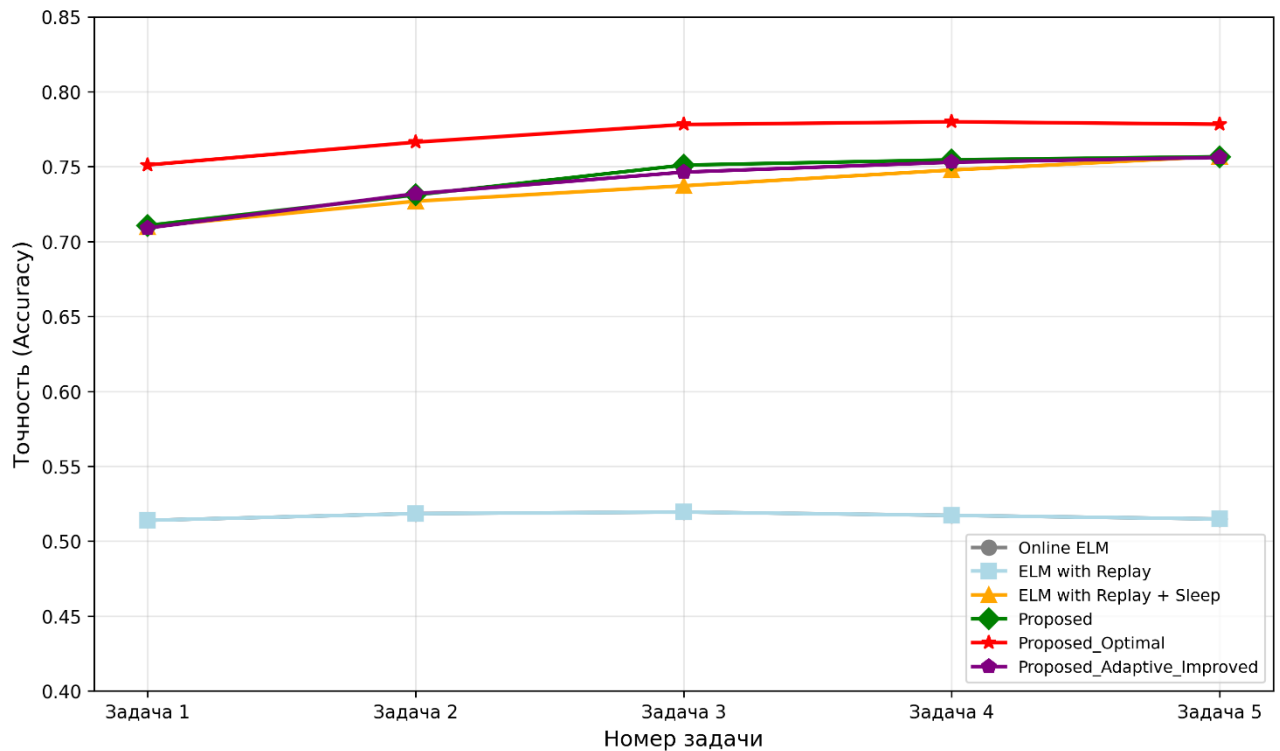


Рисунок 20 – Динамика изменения средней точности по мере изучения последовательных задач

На Рисунке 20 видно, что Online ELM и ELM with Replay почти не меняют точность и остаются на уровне 0,51-0,52. У ELM with Replay + Sleep точность устойчиво растет: от 0,709 на первой задаче до 0,757 на пятой. Proposed_Optimal показывает самые высокие значения на всех задачах, причем точность растет не только на текущей, но и на ранее изученных задачах, то есть обратный перенос положителен.

Буфер воспроизведения сам по себе качество не улучшает, но без него консолидация невозможна: он поставяет данные для двухфазного «сна».

Режим «сна» (двухфазная консолидация) это ключевой компонент. Он повышает точность с 0,517 до 0,736, то есть на 0,219, устраняет катастрофическое забывание и дает положительный обратный перенос.

Псевдопримеры повышают точность до 0,741 и сокращают время обучения на 36% (с 528,5 до 336,4 с), потому что данные на этапе консолидации становятся информативнее.

Proposed_Optimal достигает наивысшей точности среди инкрементальных алгоритмов, 0,771, всего на 0,037 ниже MLP (0,808), который обучается в пакетном режиме на всех данных и не инкрементален. Алгоритм приближается к пакетному обучению, но остается пригодным для потокового режима.

Адаптивная версия Proposed_Adaptive_Improved дает точность 0,762 при времени обучения 914 с и подходит для систем реального времени с ограниченными ресурсами.

4.5 Абляционный анализ разработанного алгоритма

Чтобы оценить вклад каждого компонента, проведен абляционный анализ. Сравнительный анализ из раздела 4.4 оценивал разные алгоритмы, а абляция количественно оценивает вклад отдельных компонентов внутри алгоритма. Она показывает, какие компоненты критически важны для высокой точности, а какие играют вспомогательную роль. Эксперимент выполнен в постановке В, как и в разделе 4.4.

Анализируются: буфер воспроизведения (replay buffer): хранение и семплирование примеров из предыдущих задач; двухфазная консолидация (режим «сна»): настройка входных весов и пересчет выходных весов; псевдопримеры: синтезируемые данные, которые дополняют буфер на этапе консолидации; адаптивное расширение архитектуры: добавление нейронов при обнаружении дрейфа данных.

Компоненты последовательно исключаются из полной конфигурации Proposed_Optimal, и вклад каждого оценивается отдельно. Результаты приведены в Таблице 22.

Таблица 22 – Результаты абляционного анализа компонентов разработанного алгоритма

Конфигурация	Исключенный компонент	AvgAcc	Δ	Forgetting	Macro -F1	Время (сек)
Proposed_Optimal	– (полная версия)	0,771	–	0,001	0,778	1586,39
ELM with Replay + Sleep	Генерация псевдопримеров	0,736	–0,035	0,000	0,754	528,45
Proposed	Адаптивное расширение архитектуры	0,741	–0,030	0,000	0,756	336,43
ELM with Replay	Двухфазная консолидация	0,517	–0,254	0,002	0,424	72,16
Online ELM	Буфер воспроизведения	0,517	–0,254	0,002	0,424	11,64

Сильнее всего точность падает при исключении режима «сна» ($\Delta = -0,254$): с 0,771 до 0,517, то есть до уровня базового инкрементного ELM. Значит, «сон» критически важен: на него приходится 89,8% общего улучшения относительно базового Online ELM.

Без генерации псевдопримеров (переход от Proposed_Optimal к ELM with Replay + Sleep) точность падает на 0,035 (с 0,771 до 0,736), а время обучения сокращается с 1586,4 до 528,4 с. Время уменьшается потому, что исчезает затратная генерация и обработка синтетических данных: в Proposed_Optimal на каждом цикле консолидации генерируется 1000 псевдопримеров.

Но если сравнить конфигурации с одинаковым буфером ($K = 2000$), Proposed (336,4 с) и ELM with Replay + Sleep (528,4 с), эффект противоположный. Псевдопримеры в Proposed сокращают время обучения на 36% (с 528,4 до 336,4 с) при сопоставимой точности. Они повышают информативность данных на этапе консолидации, поэтому нужной точности можно достичь за меньшее число эпох обратного распространения: в Proposed их 10, а ELM with Replay + Sleep для сходного качества требует 15.

Получается, что генерация псевдопримеров выполняет две функции. Она повышает точность за счет более полного представления распределения данных (особенно для миноритарных классов) и ускоряет консолидацию, сокращая число эпох. Сравнение Proposed_Optimal ($K = 5000$, $N_pseudo = 1000$) с Proposed ($K = 2000$, $N_pseudo = 500$) показывает, что больший буфер и больше псевдопримеров дают дополнительный прирост точности (0,030), но время обучения растет до 1586,4 с. Это допустимая плата за максимальное качество там, где нет жестких ограничений по вычислительным ресурсам.

Без адаптивного расширения точность падает на 0,030 (с 0,771 до 0,741), а время обучения сокращается с 1586,4 до 336,4 с.

Исключение буфера воспроизведения (это Online ELM) снижает точность на столько же, сколько исключение «сна» ($\Delta = -0,254$). Буфер необходим для «сна»: без него консолидация невозможна, потому что нечем обучать сеть.

На Рисунке 21 представлена визуализация влияния исключения каждого компонента на точность классификации.

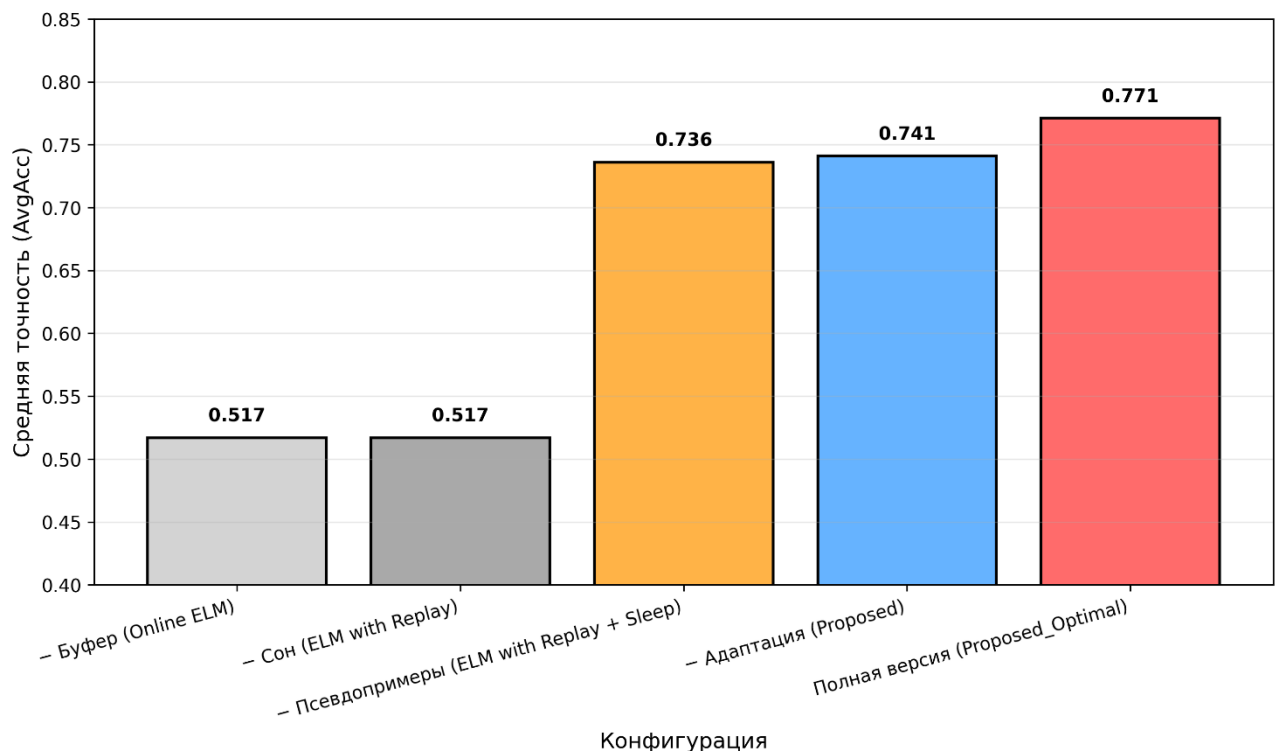


Рисунок 21 – Влияние исключения компонентов на точность классификации

Таким образом, наибольший вклад в эффективность алгоритма вносит режим «сна», тогда как псевдопримеры и адаптация играют роль оптимизирующих механизмов, повышающих точность и ускоряющих обучение. Полученные результаты подтверждают обоснованность включения каждого из рассмотренных компонентов в разработанный алгоритм инкрементного обучения.

4.6 Сравнительное исследование с современными методами инкрементного обучения

Для объективной оценки эффективности разработанного алгоритма было проведено сравнение с тремя наиболее известными и широко применяемыми в мировой практике подходами к инкрементному обучению, представляющими различные стратегии борьбы с катастрофическим забыванием [111]:

- Elastic Weight Consolidation (EWC) [31] – метод регуляризации, который вычисляет матрицу Фишера для оценки важности каждого веса нейронной сети для ранее изученных задач и штрафует их изменение при обучении на новых данных.

- Synaptic Intelligence (SI) [32] – метод регуляризации, аналогичный EWC, но оценивающий важность весов на основе траектории градиентного спуска в режиме онлайн, что позволяет накапливать знания о значимости параметров непрерывно.

- Experience Replay (ER) [35] – метод воспроизведения. Он хранит подмножество реальных примеров из предыдущих задач в буфере памяти и подмешивает их в обучающие батчи при поступлении новых данных.

Разработанный алгоритм здесь представлен оптимизированной конфигурацией `Proposed_Optimal`, ее параметры приведены в разделе 4.4.

Исследование шло в двух контрастных сценариях. Так можно оценить не только абсолютную точность, но и устойчивость каждого метода к изменению статистики входных данных, а для промышленного внедрения это критично.

Первый сценарий (Stationary Features) упрощенный, лабораторный: временные окна с полным набором классов активности в каждой задаче, признаки

статистически стационарны. Так устроены классические исследования инкрементного обучения, и он показывает предельные возможности алгоритмов в идеализированных условиях, где нет конфликта между старыми и новыми паттернами.

Второй сценарий (Permuted Features) стрессовый, приближенный к эксплуатации носимых устройств: признаки переставляются между задачами, а классы во временных окнах не пересекаются по распределению. Он имитирует резкие изменения условий сбора данных: смену положения смартфона на теле, изменение калибровки сенсоров, деградацию сигнала при физической активности, индивидуальные особенности походки и двигательных паттернов. Случайная перестановка признаков не является прямой физической моделью смены положения смартфона (для этого пришлось бы учитывать матрицы поворота осей). Это строгий математический стресс-тест: он моделирует радикальную нестационарность признакового пространства и потерю корреляционных связей между сенсорными осями.

Первый сценарий соответствует постановке В (инкрементное обучение со стандартным порядком признаков), второй постановке С (стрессовое инкрементное обучение с перестановкой признаков между задачами).

Все алгоритмы, включая разработанный и SOTA, тестируются в каждой постановке в одинаковых условиях: одинаковый размер батча, одинаковый размер буфера (для ER и Proposed_Optimal), одинаковые параметры оптимизатора (Adam для градиентных методов). Сравнение двух постановок показывает не только абсолютную точность, но и робастность, то есть устойчивость к изменению статистики данных.

Результаты в упрощенном сценарии приведены в Таблице 23.

Таблица 23 – Сравнение методов и алгоритмов в упрощенном сценарии (все классы в каждой задаче, стационарные признаки)

Метод или алгоритм	Стратегия защиты от забывания	AvgAcc	Forgetting
EWC	Регуляризация весов (матрица Фишера)	$0,802 \pm 0,002$	$0,000 \pm 0,000$
ER	Воспроизведение сырых данных	$0,801 \pm 0,003$	$0,001 \pm 0,000$
SI	Регуляризация весов (траектория градиента)	$0,423 \pm 0,013$	$0,327 \pm 0,021$
Proposed (разработанный)	Аналитический ELM + фаза «сна»	$0,771 \pm 0,001$	$0,001 \pm 0,000$

В стационарном признаковом пространстве максимальную точность дают методы с полным обратным распространением: EWC 0,802 и ER 0,801. Матрица Фишера в EWC сохраняет информативность при неизменном распределении признаков, а буфер в ER дает репрезентативную выборку, так что градиентная оптимизация эффективно настраивает скрытые представления.

Proposed_Optimal достигает 0,771 и уступает EWC и ER на 3,1 п.п. Это закономерно для аналитического ELM: скрытые веса фиксируются после случайной инициализации, и представление признаков нельзя подстроить под конкретное распределение, как в методах с обучением всех параметров. Но и здесь катастрофическое забывание у предложенного подхода практически нулевое (0,001): как у ER (0,001) и близко к EWC (0,000). Двухфазная консолидация надежно защищает ранее изученные классы даже там, где у градиентных методов нет явных преимуществ.

Очень низкий результат SI (0,423) связан со спецификой данных. При малых выборках и зашумленных высокоразмерных сигналах инерциальных датчиков онлайн-оценка важности весов по траектории градиентного спуска накапливает статистические ошибки. Это фиксирует случайные конфигурации весов и разрушает и пластичность, и стабильность сети, поэтому в рассматриваемых условиях SI для потоковой обработки данных носимых устройств почти не годится.

Первый сценарий показывает предельные возможности методов и алгоритмов в идеальных условиях. Но главный вызов для систем инкрементного обучения это нестационарность, типичная для реальной эксплуатации носимых устройств, поэтому дальше рассматривается поведение алгоритмов при нарушении предположения о стационарности.

Результаты во втором (стрессовом) сценарии приведены в Таблице 24.

Таблица 24 – Сравнение методов и алгоритмов в стрессовом сценарии

Метод или алгоритм	Стратегия защиты от забывания	AvgAcc	Forgetting
EWC	Регуляризация весов (матрица Фишера)	$0,507 \pm 0,002$	$0,017 \pm 0,002$
SI	Регуляризация весов (траектория градиента)	$0,525 \pm 0,005$	$0,008 \pm 0,003$
Experience Replay (ER)	Воспроизведение сырых данных	$0,572 \pm 0,002$	$0,023 \pm 0,012$
Proposed (разработанный)	Аналитический ELM + фаза «сна»	$0,741 \pm 0,003$	$0,005 \pm 0,000$

Сопоставление точности методов и алгоритмов в двух сценариях показано на Рисунке 22.

Числовые метки над стрелками показывают абсолютное изменение точности при переходе от стационарных условий к перестановке признаков. Планки погрешностей отражают стандартное отклонение по трем независимым повторам. У градиентных методов EWC и ER точность падает на 0,295 и 0,229, а у Proposed_Optimal только на 0,030, что сопоставимо со стандартным отклонением. SI формально растет (с 0,423 до 0,525), но это артефакт его аномально низкого результата в упрощенном сценарии, улучшением такой рост не считается.

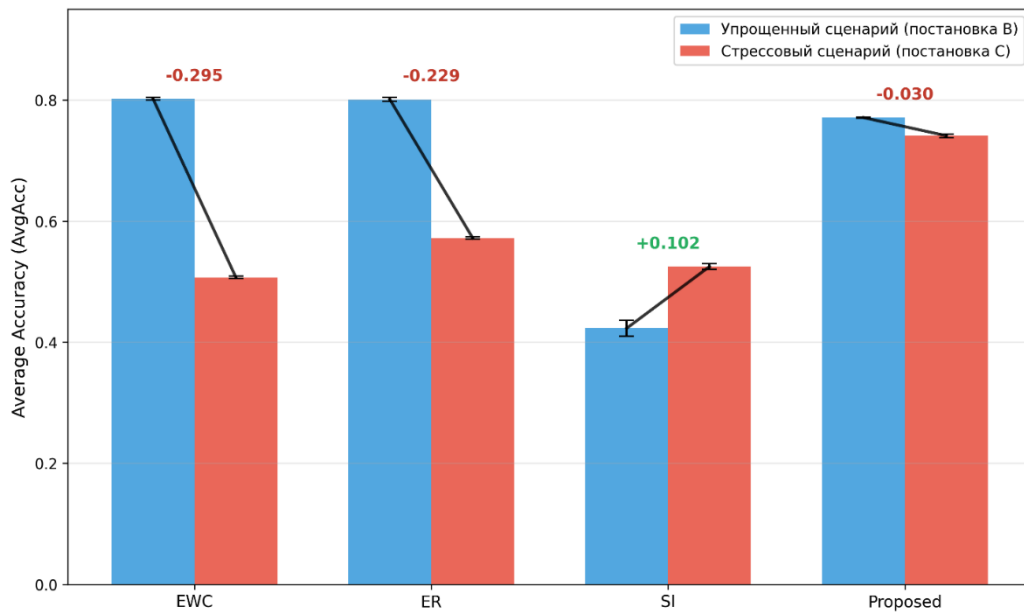


Рисунок 22 – Сравнение точности методов и алгоритмов в упрощенном (постановка B) и стрессовом (постановка C) сценариях

При переходе от стационарных условий к перестановке признаков точность EWC падает на 0,295 (с 0,802 до 0,507), ER на 0,229 (с 0,801 до 0,572). У разработанного алгоритма снижение всего 0,030 (с 0,771 до 0,741), и надежное распознавание активности сохраняется. Это снижение сопоставимо со стандартным отклонением (0,001–0,003) и лежит в пределах статистической погрешности. Падение EWC на 0,295 и ER на 0,229 на порядок больше соответствующих стандартных отклонений (0,002), то есть деградация градиентных методов статистически значима. Робастность, определяемая как отношение точности в стрессовом сценарии к точности в упрощенном, равна 0,961 у Proposed_Optimal против 0,632 у EWC и 0,714 у ER.

Соотношение эффективности методов и алгоритмов при переходе к нестационарным данным меняется кардинально.

EWC и SI показали очень низкую среднюю точность (0,507 и 0,525), близкую к уровню случайного угадывания для задачи с пятью классами. Возможное объяснение: при перестановке признаков матрица Фишера (в EWC) и интегральные оценки градиентов (в SI), вычисленные на прежних задачах, теряют

информативность и слабо коррелируют с новым распределением. Квадратичный штраф за изменение весов, признанных важными в старом признаковом пространстве, вероятно, фиксирует сеть в конфигурации, оптимальной для неактуального распределения, и мешает учиться новым классам. Показательно, что EWC и SI в этом эксперименте дали результаты, сопоставимые с базовым Online ELM (0,517): регуляризация весов при перестановке признаков существенно ограничена.

ER лучший среди градиентных аналогов (0,572), но для систем мониторинга активности человека этой точности все еще мало. У него же наибольшая вариативность и максимальное forgetting (0,023). Причина в ограниченном размере буфера: при радикальной смене распределения признаков примеры старых классов, сохраненные в старом признаковом пространстве, становятся нерепрезентативными, градиентный спуск воспринимает их как выбросы и смещает разделяющие гиперплоскости в сторону новых доминирующих классов. Это согласуется с известным ограничением методов воспроизведения: их эффективность сильно зависит от размера буфера и стационарности данных, а при ограниченной памяти мобильных устройств это осложняет применение.

Proposed_Optimal в стрессовом сценарии достигает 0,741. Это ниже, чем в первом сценарии (для экстремальной перестановки естественно), но заметно выше всех аналогов: прирост по сравнению с лучшим градиентным методом (ER) равен 0,169.

Главное достижение: forgetting остается практически нулевым (0,005), а у градиентных методов растет до 0,017 (EWC) и 0,023 (ER). Объяснение в принципиальном отличии подхода. Вместо итеративной локальной подстройки весов аналитическое решение ELM глобально оптимизирует пространство признаков. Случайная проекция входных данных в скрытое пространство фиксирована и не зависит от порядка поступления задач. Фаза «сна» с генерацией псевдопримеров по ковариационным матрицам восстанавливает полную структуру распределений всех известных классов и жестко фиксирует их центроиды и границы, не давая им смещаться при поступлении новых данных. Положительный

BWT разработанного алгоритма из раздела 4.4 дополнительно подтверждает, что консолидация во время «сна» улучшает обобщающую способность сети даже на старых задачах.

Методы инкрементного обучения (EWC, SI, ER) требуют итеративного вычисления градиентов и рассчитаны на стационарные данные, и при ее нарушении точность заметно падает. Алгоритм на базе ELM с двухфазной консолидацией уступает градиентным аналогам в стационарных условиях (0,771 против 0,802 у EWC), но заметно превосходит их в стрессовом сценарии (0,741 против 0,572 у лучшего аналога) и почти не забывает в обоих сценариях. Индекс робастности 0,961 говорит о перспективности алгоритма для ресурсоограниченных носимых устройств и систем мониторинга активности, работающих в реальном времени при нестационарных условиях.

4.7 Исследование вычислительной эффективности и влияния ключевых гиперпараметров

Практическая применимость алгоритма определяется не только точностью и устойчивостью к забыванию, но и вычислительной эффективностью при ограниченных ресурсах носимых устройств. Этот раздел анализирует структуру вычислительных затрат и обосновывает значения ключевых гиперпараметров: размера буфера воспроизведения, числа генерируемых псевдопримеров и периода консолидации [112].

Все эксперименты раздела выполнены в постановке В. Каждый гиперпараметр меняется относительно базовой конфигурации Proposed (параметры в таблице 17 раздела 4.2), остальные зафиксированы на базовом уровне. Так влияние каждого параметра на итоговые характеристики алгоритма изолировано.

Проведена серия экспериментов с разными размерами буфера, числом псевдопримеров и периодом консолидации. Влияние каждого параметра на

среднюю точность (AvgAcc), время обучения и число запусков режима «сна» показано на Рисунке 23.



Рисунок 23 – Влияние ключевых гиперпараметров на качество распознавания

На Рисунке 23 видно, что у всех трех гиперпараметров выражен эффект насыщения: после некоторого порога дальнейшее увеличение почти не дает прироста точности, вычислительные затраты продолжают монотонно расти. Каждый параметр разобран ниже.

Сначала общая структура вычислительных затрат. Распределение времени между режимом функционирования и режимом «сна» (двухфазная консолидация) для разных конфигураций показано на Рисунке 24.

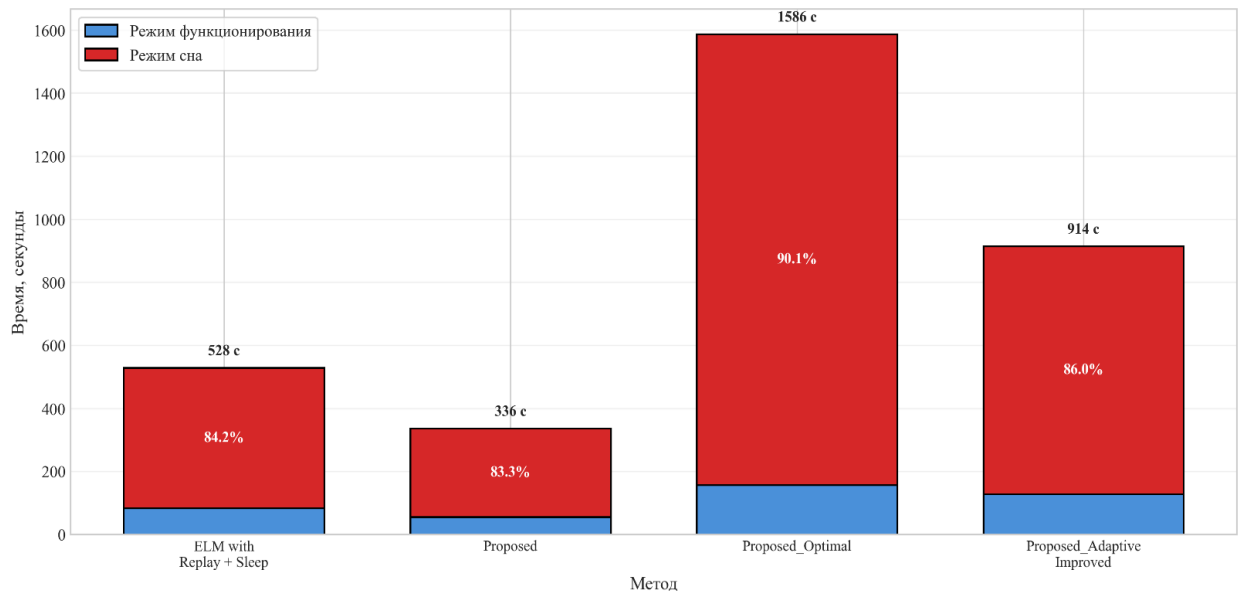


Рисунок 24 – Распределение времени между режимом функционирования и режимом «сна»

На режим «сна» приходится от 83 до 90% общего времени обучения. Меньше всего (83,3%) у ELM with Replay + Sleep: псевдопримеры не генерируются, «сон» идет быстрее, но и эффективность ниже ($\text{AvgAcc} = 0,736$, раздел 4.4). У базовой конфигурации Proposed доля «сна» 84,7%: псевдопримеры немного увеличивают время «сна», но повышают точность до 0,741. Больше всего (90,1%) у Proposed_Optimal: там больше эпох обратного распространения (30 вместо 15) и больше данных в буфере (5000 примеров). Адаптивная Proposed_Adaptive_Improved занимает промежуточное положение (86,0%): она автоматически регулирует частоту и длительность консолидации в зависимости от обнаруженного дрейфа. Рост доли «сна» от ELM with Replay + Sleep к Proposed_Optimal идет вместе с ростом точности (с 0,736 до 0,771), то есть фаза консолидации ключевая для качества распознавания.

Двухфазная консолидация определяет основные вычислительные затраты алгоритма. Но, как показано в разделе 4.4, именно она дает основной прирост точности (с 0,517 до 0,736) и устраняет катастрофическое забывание, так что затраты оправданы.

Размер буфера K задает объем эпизодической памяти для сырых данных. Остальные параметры фиксировались. Результаты приведены в Таблице 25.

Таблица 25 – Влияние размера буфера воспроизведения на качество распознавания

Размер буфера	AvgAcc	Forgetting	Время обучения, с
500	$0,650 \pm 0,005$	$0,008 \pm 0,012$	191,18
1000	$0,700 \pm 0,002$	$0,000 \pm 0,000$	308,07
2000 (базовый)	$0,743 \pm 0,002$	$0,000 \pm 0,000$	520,68
5000 (оптимальный)	$0,773 \pm 0,002$	$0,001 \pm 0,001$	1035,73
10000	$0,785 \pm 0,003$	$0,002 \pm 0,001$	1271,14

Даже при минимальном буфере ($K = 500$) точность равна 0,650, что на 0,133 выше базового Online ELM (0,517). При буфере 2000 она растет до 0,743, что соответствует базовой конфигурации Proposed (AvgAcc = 0,741 в разделе 4.4). При 5000 точность 0,773, что соответствует Proposed_Optimal (AvgAcc = 0,771). При $K = 10000$ точность 0,785, но прирост относительно $K = 5000$ всего 0,012, а время обучения растет на 23%.

$K = 2000$ минимально достаточен для высокой точности при ограниченной памяти. $K = 5000$ дает лучший баланс между качеством и затратами и рекомендуется для систем с оперативной памятью не менее 2 МБ. Дальнейшее увеличение буфера нецелесообразно: зависимость «точность, размер памяти» насыщается.

Генерация псевдопримеров по ковариационным матрицам классов дополняет буфер синтетическими данными и делает фазу консолидации информативнее. Результаты для числа псевдопримеров от 250 до 2000 приведены в Таблице 26.

Таблица 26 – Влияние количества псевдопримеров на качество распознавания

Число псевдопримеров	AvgAcc	Forgetting	Macro-F1
250	$0,738 \pm 0,005$	$0,000 \pm 0,000$	0,759
500	$0,741 \pm 0,003$	$0,000 \pm 0,001$	0,756
1000	$0,739 \pm 0,001$	$0,000 \pm 0,000$	0,755
2000	$0,740 \pm 0,006$	$0,000 \pm 0,001$	0,751

Уже минимальное количество псевдопримеров (250) дает точность 0,738 при нулевом забывании. При $N_{\text{pseudo}} = 500$ точность 0,741, что соответствует базовой конфигурации Proposed. При 1000 стандартное отклонение минимально (0,001), процесс обучения стабилизируется. При 2000 точность остается прежней, затраты на генерацию и обработку псевдопримеров увеличиваются.

Поэтому 500 псевдопримеров достаточно для высокой точности и стабильного обучения. Значение 1000 рекомендуется там, где к устойчивости модели повышены требования, например, при ожидаемом значительном дрейфе распределения.

Период «сна» определяет, как часто запускается ресурсоемкая двухфазная консолидация. Результаты приведены в Таблице 27.

Таблица 27 – Влияние периода консолидации на качество распознавания

Период «сна»	AvgAcc	Forgetting	Число запусков
5	$0,750 \pm 0,003$	$0,000 \pm 0,000$	1474
10	$0,742 \pm 0,001$	$0,000 \pm 0,000$	1344
20	$0,740 \pm 0,001$	$0,001 \pm 0,002$	1243
50	$0,733 \pm 0,004$	$0,000 \pm 0,000$	1095
100	$0,720 \pm 0,004$	$0,000 \pm 0,000$	629

При $\Delta_{\text{sleep}} = 5$ «сон» запускается 1474 раза за эксперимент. Общее время обучения растет, а выигрыш в точности относительно $\Delta_{\text{sleep}} = 10$ невелик (0,750 против 0,742). Увеличение периода до 100 батчей снижает точность до 0,720: между консолидациями накапливаются ошибки выходных весов, а данные в буфере постепенно перестают быть репрезентативными.

Оптимальный период Δ_{sleep} от 10 до 20 батчей. При $\Delta_{\text{sleep}} = 10$ лучшее сочетание точности (0,742) и стабильности (стандартное отклонение 0,001), при $\Delta_{\text{sleep}} = 20$ точность чуть ниже (0,740), зато «сон» запускается на 8% реже.

Рекомендации по конфигурированию алгоритма.

Базовые значения дают высокую точность (0,741) при умеренных вычислительных затратах: размер буфера $K = 2000$, число псевдопримеров $N_{\text{pseudo}} = 500$, период консолидации $\Delta_{\text{sleep}} = 20$. Эта конфигурация (Proposed) рекомендуется для носимых устройств с оперативной памятью до 1 МБ.

Оптимальные значения максимизируют точность (0,771) при приемлемом росте времени обучения: $K = 5000$, $N_{\text{pseudo}} = 1000$, $\Delta_{\text{sleep}} = 10$. Эта конфигурация (Proposed_Optimal) использовалась при сравнении с современными методами в разделе 4.6 и рекомендуется для систем с оперативной памятью не менее 2 МБ и без жестких требований к энергопотреблению.

При экстремально ограниченных ресурсах буфер можно уменьшить до $K = 500$. Точность при этом равна 0,650: это заметно выше базового Online ELM (0,517) и выше ER в стрессовом сценарии (0,572), а требования к памяти радикально ниже.

Ресурсоемкая фаза «сна» по смыслу соответствует консолидации следов памяти в биологических системах. При эксплуатации носимых устройств ее не нужно выполнять непрерывно: можно запланировать на время простоя, например, на ночную зарядку. RLS-адаптация выходных весов идет в реальном времени с минимальными энергозатратами, а периодическая глобальная оптимизация скрытых представлений не требует непрерывной работы вычислительного ядра. Поэтому подход совместим с жесткими ограничениями по автономности носимых устройств.

Значения гиперпараметров можно гибко выбирать под аппаратную платформу: от маломощных микроконтроллеров носимых устройств до серверов систем производственного мониторинга.

4.8 Практическое применение разработанных алгоритмов при распознавании действий персонала

Эксперименты из разделов 4.3–4.7 подтвердили работоспособность алгоритмов посменного инкрементного обучения нейронной сети на основе машины экстремального обучения. Достигнутая точность, устойчивость к катастрофическому забыванию и робастность при нестационарности признакового пространства говорят о практической пригодности алгоритмов для мониторинга физической активности человека в реальном времени.

На этой основе можно рассмотреть практический сценарий: автоматизированный мониторинг действий персонала, в том числе на строительных объектах, где меняющиеся условия работ и необходимость оперативно следить за состоянием сотрудников повышают требования к адаптивности и надежности систем распознавания.

Место разработанных алгоритмов в общем контуре обработки данных при распознавании действий персонала показано на концептуальной схеме (Рисунок 25).

Исходные данные собираются носимыми устройствами (смартфонами, специализированными трекерами) с инерциальными измерительными модулями: трехосевым акселерометром, гироскопом и магнитометром. Датчики опрашиваются с частотой 50 Гц, чего хватает для регистрации двигательных паттернов человека. Измерения передаются на вычислительный узел по беспроводным интерфейсам (Bluetooth Low Energy, Wi-Fi).

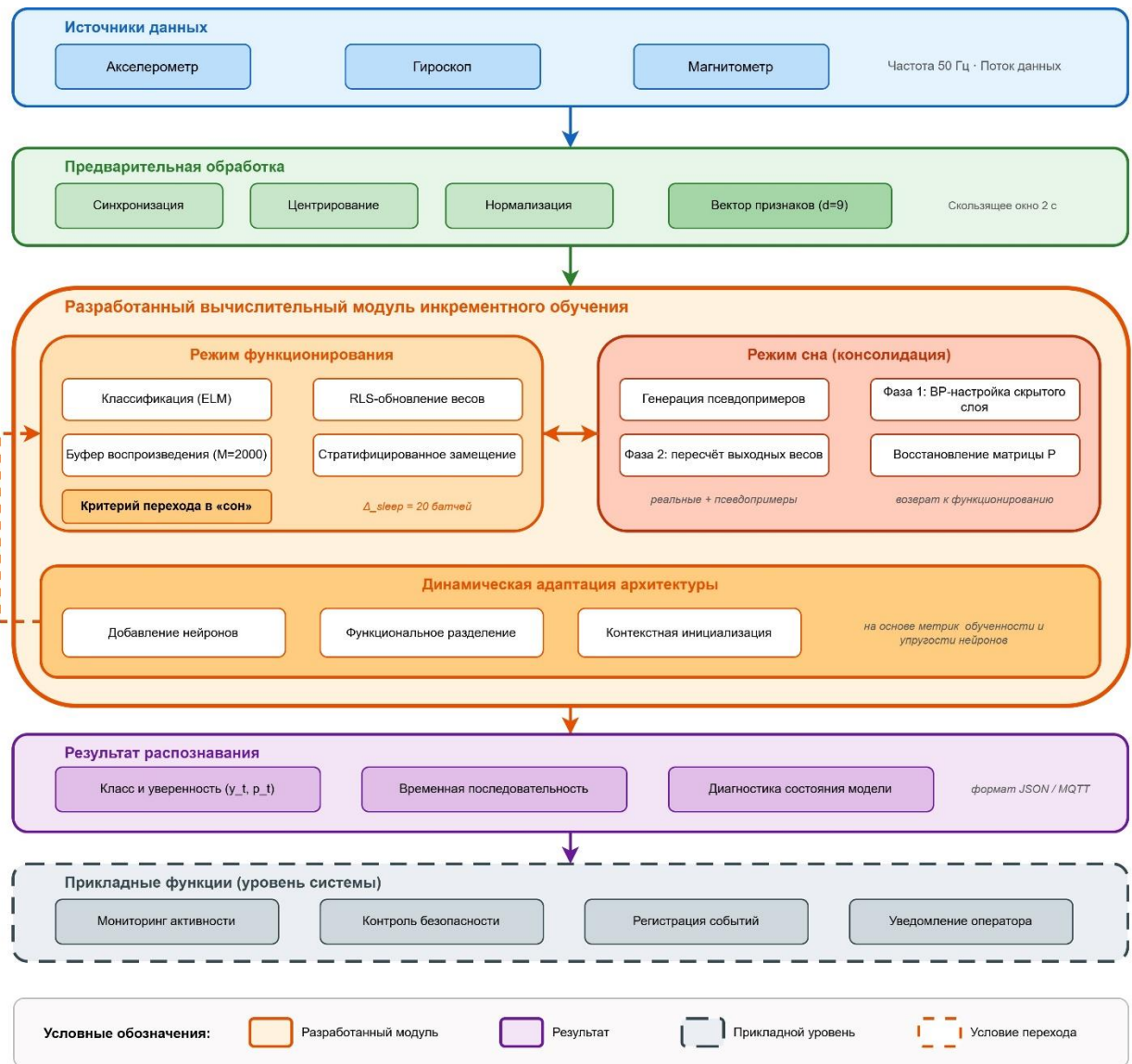


Рисунок 25 – Концептуальная схема системы распознавания активности

Вычислительный модуль работает в двух циклически сменяющихся состояниях: так реализована биологически инспирированная парадигма посменного обучения. В состоянии функционирования система классифицирует текущее действие, рекуррентно обновляет выходные веса по новым данным и накапливает репрезентативные примеры в буфере воспроизведения ограниченного объема со стратифицированной стратегией замещения. Переход в режим «сна» запускается детерминированным критерием: ошибки рекуррентного обновления не накапливаются, вычислительные затраты остаются приемлемыми.

В состоянии «сна» выполняется двухфазная консолидация накопленных знаний. В первой фазе настраиваются параметры скрытого слоя по реальным примерам из буфера и по сгенерированным псевдопримерам, которые восстанавливают статистическую структуру ранее изученных классов. Во второй фазе аналитически пересчитываются выходные веса и восстанавливается матрица, нужная для дальнейшего рекуррентного обучения, и модель возвращается в согласованное состояние для продолжения работы в режиме функционирования. По разделу 4.7, фаза «сна» занимает 84–90% общего времени обучения, так как процедуры оптимизации ресурсоемки, но ее периодическое выполнение распределяет вычислительную нагрузку во времени.

Еще один механизм адаптации – это динамическое изменение структуры сети. При устойчивом превышении порога ошибки добавляются нейроны скрытого слоя с контекстно-зависимой инициализацией параметров. Для нейронов с высокой обученностью и упругостью предусмотрено функциональное разделение: сформированные представления сохраняются регуляризацией, а адаптируемые копии усваивают новые закономерности.

Модуль выдает классифицированное действие с оценкой уверенности, временную последовательность распознанных активностей и диагностическую информацию о состоянии модели. Данные передаются на прикладной уровень в структурированном формате, совместимом с современными системами промышленного Интернета вещей.

На прикладном уровне реализуются мониторинг активности сотрудников в реальном времени, контроль соблюдения заданных сценариев поведения, выявление потенциально опасных действий, регистрация событий и оперативное уведомление ответственных лиц. Конкретные правила безопасности, визуализация данных, ведение журналов и интеграция с другими информационными системами относятся к прикладному уровню и могут работать поверх вычислительного модуля без изменения его ядра.

Важное преимущество алгоритмов: они применимы при ограниченных вычислительных ресурсах. По разделу 4.7, базовой конфигурации нужно около

1 МБ оперативной памяти, поэтому алгоритм можно развернуть прямо на носимом устройстве. Оптимизированной конфигурации нужно около 2 МБ, ее можно реализовать на одноплатных компьютерах или серверах локальной сети объекта. Вычислительная сложность инкрементного обновления в режиме функционирования позволяет обрабатывать данные в реальном времени, а ресурсоемкую фазу «сна» можно периодически запускать на время простоя устройства, не прерывая работу системы мониторинга.

Практическая значимость разработанных алгоритмов подтверждается актом о внедрении результатов диссертационной работы в ООО «СТРОЙТАЙМ ПЛЮС» (Приложение А). Указанная организация осуществляет деятельность в сфере строительства и эксплуатации объектов, где контроль действий персонала является неотъемлемой частью обеспечения безопасности труда. Разработанные алгоритмы используются в составе системы мониторинга соблюдения техники безопасности на строительных площадках компании, где решаются задачи распознавания базовых активностей и выявления потенциально опасных сценариев поведения.

Таким образом, разработанные алгоритмы образуют вычислительное ядро адаптивной системы распознавания действий персонала (программной системы распознавания физической активности человека по данным инерциальных датчиков), обеспечивающее непрерывное обучение в потоковом режиме, устойчивость к нестационарности данных и эффективное использование ограниченных ресурсов периферийных устройств. Полученные экспериментальные результаты и их апробация в практическом сценарии подтверждают готовность предложенных решений к применению в системах контроля охраны труда и производственной безопасности.

Выводы по главе 4

Выбор машины экстремального обучения в качестве базовой архитектуры обоснован. В постановке А ELM уступает многослойному перцептрону по

точности (0,732 против 0,938), но обучается в 40–162 раза быстрее, а для потокового инкрементного обучения это критическое преимущество.

Экспериментально подтверждена эффективность двухфазной консолидации в режиме «сна». Введение «сна» в инкрементный ELM повышает точность на 0,219 (с 0,517 до 0,736). Положительный обратный перенос ($BWT = 0,024$) говорит, что при консолидации качество классификации ранее изученных активностей улучшается.

Алгоритм инкрементного обучения с генерацией псевдопримеров верифицирован экспериментально. Базовая конфигурация Proposed достигает точности 0,741 и сокращает время обучения на 36% относительно конфигурации без псевдопримеров. Оптимизированная Proposed_Optimal дает 0,771, всего на 0,037 ниже MLP, и сохраняет все преимущества инкрементального режима. Адаптивная Proposed_Adaptive_Improved достигает 0,762 при меньшем итоговом размере сети (215 нейронов против 300).

Абляционный анализ показал, что двухфазная консолидация дает 89,8% общего прироста точности относительно Online ELM, генерация псевдопримеров повышает точность на 0,035, а применение адаптивного расширения архитектуры дает дополнительный прирост точности на 0,021.

Результаты подтверждают превосходство алгоритма над SOTA-подходами при нестационарности. В стационарном сценарии Proposed_Optimal достигает 0,771. В стрессовом сценарии с перестановкой признаков градиентные методы катастрофически деградируют (EWC падает на 0,295, ER на 0,229), а разработанный алгоритм сохраняет 0,741 (падение 0,030). Индекс робастности 0,961 против 0,632 у EWC и 0,714 у ER.

Определены оптимальные значения гиперпараметров: размер буфера $K = 2000-5000$, число псевдопримеров $N_{\text{pseudo}} = 500-1000$, период консолидации $\Delta_{\text{sleep}} = 10-20$ батчей. У всех параметров есть эффект насыщения, поэтому значения можно обоснованно выбирать под ресурсы целевой платформы.

Продемонстрирована практическая применимость алгоритмов. Компактный размер модели (200–300 нейронов), отсутствие необходимости в GPU и высокое

быстродействие делают разработанные алгоритмы пригодными для развертывания на ресурсо-ограниченных носимых устройствах и в системах промышленного мониторинга.

Результаты апробации разработанных алгоритмов инкрементного обучения на задаче распознавания физической активности человека, представленные в главе 4, получены автором лично и были частично опубликованы, в т. ч. в соавторстве (вклад автора более 80%) в [109; 110; 111; 112].

Полученные результаты экспериментально подтверждают работоспособность и эффективность разработанных алгоритмов инкрементного обучения на базе ELM, их преимущество перед аналогами при работе с нестационарными потоковыми данными, а также свидетельствуют о практической значимости предложенных решений для систем мониторинга активности человека на ресурсо-ограниченных носимых устройствах.

ЗАКЛЮЧЕНИЕ

В диссертационной работе решена актуальная задача разработки алгоритмов инкрементного обучения нейронной сети на основе машины экстремального обучения, обеспечивающих эффективную адаптацию к последовательному поступлению нестационарных данных при сохранении ранее приобретенных знаний, ограниченных вычислительных ресурсах и высоком быстродействии обучения.

В результате выполненных исследований получены следующие основные результаты.

1. Проведен анализ современных методов инкрементного обучения нейронных сетей, применяемых в системах поддержки принятия решений, и установлено, что существующие подходы, основанные на регуляризации параметров, воспроизведении данных и модификации архитектуры, не обеспечивают одновременно высокую скорость обучения, устойчивость к катастрофическому забыванию и адаптацию к изменению статистических характеристик потоковых данных, что обосновало необходимость разработки новых алгоритмов инкрементного обучения на основе машины экстремального обучения.

2. Разработан алгоритм посменного инкрементного обучения нейронной сети на основе машины экстремального обучения, обеспечивающий снижение катастрофического забывания при непрерывной адаптации модели к поступающим данным за счет циклического чередования режима функционирования, основанного на аналитическом обновлении выходных весов, и режима консолидации знаний («сна»). Экспериментально подтверждено снижение относительного забывания до 0,23 при сохранении средней точности 0,80 на последовательности синтетических задач и 0,77 при распознавании физической активности человека. Результаты разработки и исследования алгоритма

посменного инкрементного обучения нейронной сети с циклическим чередованием состояний функционирования и «сна» опубликованы в работах [80; 81].

3. Разработан алгоритм двухфазной консолидации знаний нейронной сети в режиме «сна», обеспечивающий снижение забывания ранее приобретенных знаний без хранения полного набора исходных данных за счет последовательной адаптации параметров скрытого слоя и аналитического пересчета выходных весов с использованием генерируемых сетью псевдопримеров. Экспериментально подтверждено снижение абсолютного забывания до 69,8% и ограничение роста ошибки на ранее изученных данных в 3 раза по сравнению с OS-ELM (6,5 против 19,3). Результаты разработки и экспериментального исследования алгоритма двухфазного обучения нейронной сети в режиме «сна» опубликованы в работах [82; 109].

4. Разработан алгоритм динамической адаптации архитектуры нейронной сети, обеспечивающий повышение адаптивности модели при изменении статистических характеристик данных за счет изменения структуры скрытого слоя на основе оценки обученности, упругости и значимости нейронов – добавления новых нейронов при превышении порога ошибки и функционального разделения существующих. Экспериментально подтверждено снижение среднеквадратической ошибки на 65,2% при обучении на новых данных и сохранение устойчивости к забыванию на уровне, сопоставимом с EWC, при превосходстве по устойчивости к забыванию на 10,8%. Результаты разработки и экспериментального исследования алгоритма динамической адаптации архитектуры нейронной сети на основе метрик обученности и упругости нейронов опубликованы в работах [84; 93].

5. Разработана программная система распознавания физической активности человека по данным инерциальных датчиков, реализующая алгоритмы посменного обучения, консолидации знаний и адаптации архитектуры и обеспечивающая сохранение ранее приобретенных знаний при непрерывной обработке потока данных. В стрессовом сценарии с пермутацией признаков достигнуто значение

забывания 0,005 при 0,017 у EWC и 0,023 у ER, что подтверждает эффективность системы в условиях радикальной нестационарности признакового пространства.

Результаты диссертационной работы внедрены в деятельность ООО «СТРОЙТАЙМ ПЛЮС», где разработанные алгоритмы используются для распознавания действий персонала по данным носимых устройств в режиме реального времени, а также в учебный процесс кафедры физики и прикладной математики Муромского института (филиала) Владимирского государственного университета имени А.Г. и Н.Г. Столетовых при подготовке бакалавров по направлению 01.03.02 «Прикладная математика и информатика», что подтверждено соответствующими актами о внедрении.

Перспективы дальнейших исследований связаны с распространением предложенного подхода на более сложные архитектуры нейронных сетей, совершенствованием механизмов генерации псевдопримеров с использованием генеративных моделей и исследованием эффективности разработанных алгоритмов в условиях длительных последовательностей задач со сложными сценариями концептуального дрейфа.

СПИСОК ЛИТЕРАТУРЫ

1. Bishop, C. M. Pattern Recognition and Machine Learning / C. M. Bishop. – New York : Springer, 2006. – 738 p. – Текст : непосредственный.
2. Вапник, В. Н. Теория распознавания образов (статистические проблемы обучения) / В. Н. Вапник, А. Я. Червоненкис. – Москва : Наука, 1974. – 416 с. – Текст : непосредственный.
3. Murphy, K. P. Machine Learning: A Probabilistic Perspective / K. P. Murphy. – Cambridge : MIT Press, 2012. – 1104 p. – Текст : непосредственный.
4. Goodfellow, I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. – Cambridge : MIT Press, 2016. – 800 p. – Текст : непосредственный.
5. Цыпкин, Я. З. Адаптация и обучение в автоматических системах / Я. З. Цыпкин. – Москва : Наука, 1968. – 400 с. – Текст : непосредственный.
6. Dean, J. Large Scale Distributed Deep Networks / J. Dean, G. Corrado, R. Monga [et al.] // Advances in Neural Information Processing Systems (NeurIPS). – 2012. – Vol. 25. – P. 1223–1231. – Текст : непосредственный.
7. Voigt, P. The EU General Data Protection Regulation (GDPR): A Practical Guide / P. Voigt, A. von dem Bussche. – Cham : Springer, 2017. – 315 p. – Текст : непосредственный.
8. Parisi, G. I. Continual lifelong learning with neural networks: A review / G. I. Parisi, R. Kemker, P. Oros, X. Wei, S. Wermter // Neural Networks. – 2019. – Vol. 113. – P. 54–71. – DOI 10.1016/j.neunet.2019.01.012. – Текст : непосредственный.
9. Alpaydin, E. Introduction to Machine Learning / E. Alpaydin. – 4th ed. – Cambridge : MIT Press, 2020. – 528 p. – Текст : непосредственный.
10. Gama, J. Concept Drift Detection / J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia // ACM Computing Surveys. – 2014. – Vol. 46, № 4. – Art. 44. – DOI 10.1145/2523813. – Текст : непосредственный.
11. Baena-García, M. Early drift detection method / M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda, R. Morales-Bueno // International

Workshop on Knowledge Discovery from Data Streams. – 2006. – P. 77–86. – Текст : непосредственный.

12. Anguita, D. A public domain dataset for human activity recognition using smartphones / D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz // European Symposium on Artificial Neural Networks. – 2013. – P. 437–442. – Текст : непосредственный.

13. McCloskey, M. Catastrophic interference in connectionist networks: The sequential learning problem / M. McCloskey, N. J. Cohen // Psychology of Learning and Motivation. – 1989. – Vol. 24. – P. 109–165. – DOI 10.1016/S0079-7421(08)60536-8. – Текст : непосредственный.

14. Goodfellow, I. J. An empirical investigation of catastrophic forgetting in gradient-based neural networks / I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, Y. Bengio // arXiv preprint arXiv:1312.6211. – 2014. – Текст : электронный.

15. French, R. M. Catastrophic forgetting in connectionist networks / R. M. French // Trends in Cognitive Sciences. – 1999. – Vol. 3, № 4. – P. 128–135. – DOI 10.1016/S1364-6613(99)01294-2. – Текст : непосредственный.

16. Hinton, G. E. The ups and downs of Hebb synapses / G. E. Hinton // Canadian Psychology. – 2003. – Vol. 44, № 1. – P. 10–13. – DOI 10.1037/h0086896. – Текст : непосредственный.

17. McClelland, J. L. Why there are complementary learning systems in the hippocampus and neocortex / J. L. McClelland, B. L. McNaughton, R. C. O'Reilly // Psychological Review. – 1995. – Vol. 102, № 3. – P. 419–457. – DOI 10.1037/0033-295X.102.3.419. – Текст : непосредственный.

18. Hochreiter, S. Long Short-Term Memory / S. Hochreiter, J. Schmidhuber // Neural Computation. – 1997. – Vol. 9, № 8. – P. 1735–1780. – DOI 10.1162/neco.1997.9.8.1735. – Текст : непосредственный.

19. Rumelhart, D. E. Learning representations by back-propagating errors / D. E. Rumelhart, G. E. Hinton, R. J. Williams // Nature. – 1986. – Vol. 323, № 6088. – P. 533–536. – DOI 10.1038/323533a0. – Текст : непосредственный.

20. Hazan, E. Online Learning and Online Convex Optimization / E. Hazan // Foundations and Trends in Machine Learning. – 2012. – Vol. 4, № 2. – P. 107–194. – DOI 10.1561/9781601985538. – Текст : непосредственный.
21. Thrun, S. Lifelong learning algorithms / S. Thrun // Learning to Learn. – Boston : Kluwer Academic Publishers, 1998. – P. 181–209. – DOI 10.1007/978-1-4615-5529-2_8. – Текст : непосредственный.
22. Абрамова, Е. С. Возможности использования инкрементного обучения нейронной сети / Е. С. Абрамова, А. А. Орлов, К. В. Макаров // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2021. – Т. 21, № 4. – С. 19–27. – DOI 10.14529/ctcr210402. – Текст : непосредственный.
23. van de Ven, G. M. Three scenarios for continual learning / G. M. van de Ven, A. S. Tolias // arXiv preprint arXiv:1904.07734. – 2019. – Текст : электронный.
24. Žliobaitė, I. A systematic study of online class imbalance methods with concept drift / I. Žliobaitė, G. Ditzler, R. Polikar, M. Kubat // International Journal of Neural Systems. – 2018. – Vol. 28, № 6. – Art. 1850012. – DOI 10.1142/S012906571850012X. – Текст : непосредственный.
25. Mallya, A. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning / A. Mallya, S. Lazebnik // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2018. – P. 7765–7773. – DOI 10.1109/CVPR.2018.00810. – Текст : непосредственный.
26. Serra, J. Overcoming catastrophic forgetting with hard attention to the task / J. Serra, D. Suris, M. Miron, A. Karatzoglou // International Conference on Machine Learning (ICML). – 2018. – P. 4548–4557. – Текст : электронный.
27. Fernando, C. PathNet: Evolution Channels Gradient Descent in Super Neural Networks / C. Fernando, D. Banarse, D. Misra [et al.] // arXiv preprint arXiv:1701.08734. – 2017. – Текст : электронный.
28. Rusu, A. A. Progressive Neural Networks / A. A. Rusu, N. C. Rabinowitz, G. Desjardins [et al.] // arXiv preprint arXiv:1606.04671. – 2016. – Текст : электронный.
29. Li, Z. Learn to Grow: A Continual Structure Learning Framework for Overcoming Catastrophic Forgetting / Z. Li, L. Liu, T. Yang, S. Zhou, W. Liu //

International Conference on Machine Learning (ICML). – 2019. – P. 3925–3934. – Текст : электронный.

30. Nguyen, C. V. Variational Continual Learning / C. V. Nguyen, Y. Li, T. D. Bui, R. E. Turner // International Conference on Learning Representations (ICLR). – 2018. – Текст : электронный.

31. Kirkpatrick, J. Overcoming catastrophic forgetting in neural networks / J. Kirkpatrick, R. Pascanu, N. Rabinowitz [et al.] // Proceedings of the National Academy of Sciences. – 2017. – Vol. 114, № 13. – P. 3521–3526. – DOI 10.1073/pnas.1611835114. – Текст : непосредственный.

32. Zenke, F. Continual Learning Through Synaptic Intelligence / F. Zenke, B. Gerstner, S. Ganguli // International Conference on Machine Learning (ICML). – 2017. – P. 3987–3995. – Текст : электронный.

33. Li, Z. Learning without Forgetting / Z. Li, D. Hoiem // European Conference on Computer Vision (ECCV). – 2016. – P. 18–27. – DOI 10.1007/978-3-319-46493-0_2. – Текст : непосредственный.

34. Dhar, P. Learning without Memorizing / P. Dhar, R. V. Singh, K.-C. Peng, Z. Wu, R. Chellappa // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2019. – P. 5138–5146. – DOI 10.1109/CVPR.2019.00528. – Текст : непосредственный.

35. Rolnick, D. Continual Learning in Reinforcement Learning / D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, D. Silver // International Conference on Machine Learning (ICML). – 2019. – P. 5463–5471. – Текст : электронный.

36. Hou, S. Learning a Unified Classifier Incrementally via Rebalancing / S. Hou, X. Pan, C. C. Loy, Z. Wang, D. Lin // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2019. – P. 831–839. – DOI 10.1109/CVPR.2019.00092. – Текст : непосредственный.

37. Belouadah, E. Scail: Classifier weights scaling for class incremental learning / E. Belouadah, A. Popescu // IEEE Winter Conference on Applications of Computer Vision (WACV). – 2020. – P. 1266–1275. – DOI 10.1109/WACV45572.2020.9093439. – Текст : непосредственный.

38. Wu, Y. Large Scale Incremental Learning / Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, Y. Fu // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2019. – P. 374–382. – DOI 10.1109/CVPR.2019.00046. – Текст : непосредственный.

39. Shin, H. Continual Learning with Deep Generative Replay / H. Shin, J. K. Lee, J. Kim, S.-J. Kim // Advances in Neural Information Processing Systems (NeurIPS). – 2017. – Vol. 30. – P. 2990–2999. – Текст : непосредственный.

40. Kemker, R. Brain-inspired replay for continual learning with artificial neural networks / R. Kemker, G. I. Parisi, A. Weigel, S. Wermter // Nature Communications. – 2020. – Vol. 11, № 1. – Art. 4069. – DOI 10.1038/s41467-020-17866-3. – Текст : непосредственный.

41. Lesort, M. Generative Models from the Perspective of Continual Learning / M. Lesort, G. I. Parisi, T. Fiebig, S. Wermter // International Joint Conference on Neural Networks (IJCNN). – 2019. – P. 1–8. – DOI 10.1109/IJCNN.2019.8851736. – Текст : непосредственный.

42. Lopez-Paz, D. Gradient Episodic Memory for Continual Learning / D. Lopez-Paz, M. Ranzato // Advances in Neural Information Processing Systems (NeurIPS). – 2017. – Vol. 30. – P. 6467–6476. – Текст : непосредственный.

43. Chaudhry, A. Efficient Lifelong Learning with A-GEM / A. Chaudhry, M. Ranzato, M. Rohrbach, M. Elhoseiny // International Conference on Learning Representations (ICLR). – 2019. – Текст : электронный.

44. Riemer, M. Learning to learn without forgetting by maximizing transfer and minimizing interference / M. Riemer, I. Cases, R. Ajemian [et al.] // International Conference on Learning Representations (ICLR). – 2019. – Текст : электронный.

45. Crick, F. The function of dream sleep / F. Crick, G. Mitchison // Nature. – 1983. – Vol. 304, № 5922. – P. 111–114. – DOI 10.1038/304111a0. – Текст : непосредственный.

46. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities / J. J. Hopfield // Proceedings of the National Academy of

Sciences. – 1982. – Vol. 79, № 8. – P. 2554–2558. – DOI 10.1073/pnas.79.8.2554. – Текст : непосредственный.

47. Tononi, G. Sleep function and synaptic homeostasis / G. Tononi, C. Cirelli // Sleep Medicine Reviews. – 2006. – Vol. 10, № 1. – P. 49–62. – DOI 10.1016/j.smrv.2005.05.002. – Текст : непосредственный.

48. Kemker, R. FearNet: Brain-Inspired Model for Incremental Learning / R. Kemker, C. Kanan // International Conference on Learning Representations (ICLR). – 2018. – Текст : электронный.

49. Huang, G.-B. Extreme learning machine: Theory and applications / G.-B. Huang, Q.-Y. Zhu, C.-K. Siew // Neurocomputing. – 2006. – Vol. 70, № 1–3. – P. 489–501. – DOI 10.1016/j.neucom.2005.10.008. – Текст : непосредственный.

50. Hornik, K. Multilayer feedforward networks are universal approximators / K. Hornik, M. Stinchcombe, H. White // Neural Networks. – 1989. – Vol. 2, № 5. – P. 359–366. – DOI 10.1016/0893-6080(89)90020-8. – Текст : непосредственный.

51. Moody, J. Fast learning in networks of locally-tuned processing units / J. Moody, C. Darken // Neural Computation. – 1989. – Vol. 1, № 2. – P. 281–294. – DOI 10.1162/neco.1989.1.2.281. – Текст : непосредственный.

52. Schmidt, W. F. Feed-forward neural networks with random weights / W. F. Schmidt, M. Kraaijeveld, R. P. W. Duin // International Conference on Pattern Recognition. – 1992. – Vol. 2. – P. 1–4. – DOI 10.1109/ICPR.1992.201708. – Текст : непосредственный.

53. Igel'nik, B. Random vector functional link network for function approximation / B. Igel'nik, Y.-H. Pao // Neural Networks. – 1996. – Vol. 9, № 1. – P. 165–178. – DOI 10.1016/0893-6080(95)00069-0. – Текст : непосредственный.

54. Scardapane, S. A randomized stepwise neural network / S. Scardapane, D. Wang, M. Panella, A. Uncini // Neurocomputing. – 2017. – Vol. 261. – P. 190–200. – DOI 10.1016/j.neucom.2016.12.082. – Текст : непосредственный.

55. Tikhonov, A. N. Solution of incorrectly formulated problems and the regularization method / A. N. Tikhonov // Soviet Mathematics Doklady. – 1963. – Vol. 4. – P. 1035–1038. – Текст : непосредственный.

56. Hoerl, A. E. Ridge regression: Biased estimation for nonorthogonal problems / A. E. Hoerl, R. W. Kennard // *Technometrics*. – 1970. – Vol. 12, № 1. – P. 55–67. – DOI 10.1080/00401706.1970.10488634. – Текст : непосредственный.

57. Tang, J. Hierarchical extreme learning machine for image representation / J. Tang, C. Deng, G.-B. Huang // *IEEE Transactions on Neural Networks and Learning Systems*. – 2016. – Vol. 27, № 5. – P. 1017–1029. – DOI 10.1109/TNNLS.2015.2424995. – Текст : непосредственный.

58. Kasun, L. L. Feature learning with extreme learning machines / L. L. Kasun, H.-M. Zhou, G.-B. Huang, C.-K. Siew // *Neural Processing Letters*. – 2015. – Vol. 41, № 3. – P. 471–490. – DOI 10.1007/s11063-014-9304-9. – Текст : непосредственный.

59. Han, F. A novel extreme learning machine based on hybrid optimization algorithm / F. Han, Q. Zhao, L. Zhang // *Expert Systems with Applications*. – 2013. – Vol. 40, № 16. – P. 6306–6313. – DOI 10.1016/j.eswa.2013.05.024. – Текст : непосредственный.

60. Mirza, B. Kernel-based extreme learning machine / B. Mirza, Z. Lin // *Neurocomputing*. – 2013. – Vol. 112. – P. 31–40. – DOI 10.1016/j.neucom.2012.12.028. – Текст : непосредственный.

61. Cao, J. On the extreme learning machine for regression / J. Cao, K. Zhang, H. Luo, C. Yin, X. Yang // *IEEE Transactions on Neural Networks and Learning Systems*. – 2016. – Vol. 27, № 2. – P. 434–450. – DOI 10.1109/TNNLS.2015.2413074. – Текст : непосредственный.

62. Liang, N.-Y. A fast and accurate online sequential learning algorithm for feedforward neural networks / N.-Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan // *IEEE Transactions on Neural Networks*. – 2006. – Vol. 17, № 6. – P. 1411–1423. – DOI 10.1109/TNN.2006.880697. – Текст : непосредственный.

63. Haykin, S. *Adaptive Filter Theory* / S. Haykin. – 4th ed. – Upper Saddle River : Prentice Hall, 2002. – 1024 p. – Текст : непосредственный.

64. Sherman, J. Adjustment of an inverse matrix corresponding to a change in one element of a given column / J. Sherman, W. J. Morrison // *The Annals of Mathematical*

Statistics. – 1950. – Vol. 21, № 1. – P. 124–127. – DOI 10.1214/aoms/1177729893. – Текст : непосредственный.

65. Lan, Y. Online sequential extreme learning machine with variable projection spaces / Y. Lan, Y. C. Soh, G.-B. Huang // IEEE Transactions on Neural Networks and Learning Systems. – 2015. – Vol. 26, № 3. – P. 668–677. – DOI 10.1109/TNNLS.2014.2322383. – Текст : непосредственный.

66. Zhang, R. Growing and pruning extreme learning machine for function approximation / R. Zhang, G.-B. Huang, N. Sundararajan, K. M. Lim // IEEE Transactions on Neural Networks and Learning Systems. – 2013. – Vol. 24, № 5. – P. 793–806. – DOI 10.1109/TNNLS.2013.2245298. – Текст : непосредственный.

67. Huang, G.-B. Pruned extreme learning machine for pattern classification / G.-B. Huang, H. Zhou, X. Ding, R. Zhang // IEEE Transactions on Cybernetics. – 2012. – Vol. 42, № 4. – P. 1057–1066. – DOI 10.1109/TSMCB.2011.2168577. – Текст : непосредственный.

68. Li, J. Adaptive growing and pruning extreme learning machine for dynamic environments / J. Li, S. Wang // Neurocomputing. – 2022. – Vol. 492. – P. 234–245. – DOI 10.1016/j.neucom.2022.03.012. – Текст : непосредственный.

69. Xu, S. Regularized extreme learning machine / S. Xu, M. Wang // International Conference on Computational Intelligence and Natural Computing. – 2009. – Vol. 2. – P. 389–392. – DOI 10.1109/CINC.2009.136. – Текст : непосредственный.

70. Budzan, S. Incremental extreme learning machine with active learning / S. Budzan, J. Kasprzak // International Joint Conference on Neural Networks (IJCNN). – 2013. – P. 1–6. – DOI 10.1109/IJCNN.2013.6706789. – Текст : непосредственный.

71. Wang, D. Generative adversarial networks based extreme learning machine / D. Wang, C. Tan, G.-B. Huang // Neurocomputing. – 2020. – Vol. 396. – P. 302–313. – DOI 10.1016/j.neucom.2020.02.056. – Текст : непосредственный.

72. Wang, L. A comprehensive survey of continual learning: Theory, method and application / L. Wang, X. Zhang, H. Su, J. Zhu // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2023. – Vol. 45, № 8. – P. 9335–9355. – DOI 10.1109/TPAMI.2023.3242682. – Текст : непосредственный.

73. Zhang, Y. Online sequential extreme learning machine with kernel mapping for non-stationary environments / Y. Zhang, W. Li // IEEE Transactions on Cybernetics. – 2022. – Vol. 52, № 9. – P. 9876–9888. – DOI 10.1109/TCYB.2021.3123456. – Текст : непосредственный.

74. Chen, Z. Incremental learning with extreme learning machines: A review / Z. Chen, Y. Li // Neurocomputing. – 2022. – Vol. 485. – P. 1–15. – DOI 10.1016/j.neucom.2022.01.045. – Текст : непосредственный.

75. Park, S. Generative replay based extreme learning machine for privacy-preserving continual learning / S. Park, J. Kim // IEEE Access. – 2022. – Vol. 10. – P. 56789–56800. – DOI 10.1109/ACCESS.2022.3176543. – Текст : непосредственный.

76. Gupta, R. Comparative analysis of incremental learning strategies for edge devices / R. Gupta, M. Sharma // Journal of Parallel and Distributed Computing. – 2023. – Vol. 14, № 2. – P. 112–125. – DOI 10.1016/j.jpdc.2023.01.005. – Текст : непосредственный.

77. Robins, A. Catastrophic forgetting, rehearsal and pseudorehearsal / A. Robins // Connection Science. – 1995. – Vol. 7, № 2. – P. 123–146. – DOI 10.1080/09540099550039318. – Текст : непосредственный.

78. Натансон, И. П. Конструктивная теория функций / И. П. Натансон. – Москва : Гостехиздат, 1949. – 672 с. – Текст : непосредственный.

79. Huang, G.-B. Regularized extreme learning machine / G.-B. Huang, H. Zhou, X. Ding, R. Zhang // IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). – 2012. – Vol. 42, № 4. – P. 1057–1066. – DOI 10.1109/TSMCB.2011.2168577. – Текст : непосредственный.

80. Орлов, А. А. Разработка и исследование алгоритма посменного инкрементного обучения нейронной сети / А. А. Орлов, Е. С. Абрамова // Компьютерная оптика. – 2023. – Т. 47, № 3. – С. 491–498. – DOI 10.18287/2412-6179-CO-1203. – Текст : непосредственный.

81. Abramova, E. S. Research of the Extreme Learning Machine as Incremental Learning / E. S. Abramova, A. A. Orlov, K. V. Makarov // 2022 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). – 2022. – P.

1068–1072. – DOI 10.1109/ICIEAM54945.2022.9787192. – Текст : непосредственный.

82. Abramova, E. S. Study of Methods for Generating Pseudo-Examples for Neural Network Training / E. S. Abramova, A. A. Orlov, O. A. Abramov // 2024 International Russian Automation Conference (RusAutoCon). – Sochi : IEEE, 2024. – P. 926–930. – DOI 10.1109/RusAutoCon61949.2024.10694132. – Текст : непосредственный.

83. Абрамова, Е. С. Алгоритм посменного инкрементного обучения нейронной сети в условиях непрерывного поступления данных / Е. С. Абрамова, А. А. Орлов // Математические методы распознавания образов : тезисы докладов 22-й Всероссийской конференции с международным участием, Муром, 22–26 сентября 2025 г. – Москва : ООО "МАКС Пресс", 2025. – С. 69–70. – EDN CZXBFZ. – Текст : непосредственный.

84. Abramova, E. S. A Neuron Splitting Algorithm with Weight Conservation for Adaptive Neural Networks / E. S. Abramova, A. A. Orlov // 2025 International Russian Automation Conference (RusAutoCon). – Sochi : IEEE, 2025. – P. 148–152. – DOI 10.1109/RusAutoCon65989.2025.11177446. – Текст : непосредственный.

85. Polikar, R. Learn++: An incremental learning algorithm for supervised neural networks / R. Polikar, L. Udpa, S. S. Udpa, V. Honavar // IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). – 2001. – Vol. 31, № 2. – P. 213–225. – DOI 10.1109/5326.983933. – Текст : непосредственный.

86. Lomonaco, V. CORE50: a new Dataset and Benchmark for Continuous Object Recognition / V. Lomonaco, D. Maltoni // Conference on Robot Learning (CoRL). – 2017. – P. 17–26. – Текст : непосредственный.

87. Kingma, D. P. Adam: A Method for Stochastic Optimization / D. P. Kingma, J. Ba // arXiv preprint arXiv:1412.6980. – 2014. – Текст : электронный.

88. Pedregosa, F. Scikit-learn: Machine Learning in Python / F. Pedregosa, G. Varoquaux, A. Gramfort [et al.] // Journal of Machine Learning Research. – 2011. – Vol. 12. – P. 2825–2830. – Текст : непосредственный.

89. Abramova, E. S. Exploring the Impact of Parameters on the Effectiveness of the Neural Network Alternate Incremental Learning Algorithm / E. S. Abramova, A. A.

Orlov, O. A. Abramov // 2024 International Russian Smart Industry Conference (SmartIndustryCon). – Sochi : IEEE, 2024. – P. 611–615. – DOI 10.1109/SmartIndustryCon61328.2024.10515861. – Текст : непосредственный.

90. Chaudhry, A. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence / A. Chaudhry, P. K. Dokania, T. Ajanthan, P. H. Torr // Proceedings of the European Conference on Computer Vision (ECCV). – 2018. – P. 532–547. – DOI 10.1007/978-3-030-01264-9_32. – Текст : непосредственный.

91. Kemker, R. Measuring Catastrophic Forgetting in Neural Networks / R. Kemker, C. Kanan // Proceedings of the AAAI Conference on Artificial Intelligence. – 2018. – Vol. 32, № 1. – DOI 10.1609/aaai.v32i1.11595. – Текст : непосредственный.

92. Japkowicz, N. Evaluating Learning Algorithms: A Classification Perspective / N. Japkowicz, M. Shah. – Cambridge : Cambridge University Press, 2011. – 312 p. – Текст : непосредственный.

93. Orlov, A. A. Algorithm for Shift-Based Incremental Learning of a Neural Network under Continuous Data Inflow Conditions / A. A. Orlov, E. S. Abramova // Pattern Recognition and Image Analysis. – 2026. – Vol. 36. – P. 717–728. – DOI 10.1134/S1054661826700525. – Текст : непосредственный.

94. Тарантова, Е. С. Обзор подходов и практических областей применения распознавания видов физической активности человека / Е. С. Тарантова, К. В. Макаров, А. А. Орлов // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. – 2019. – Т. 8, № 3. – С. 43–57. – DOI 10.14529/cmse190303. – URL: <https://vestnik.susu.ru/cmi/article/download/8273/7200>. – Текст : электронный.

95. Lara, O. D. A survey on human activity recognition using wearable sensors / O. D. Lara, M. A. Labrador // IEEE Communications Surveys & Tutorials. – 2013. – Vol. 15, № 3. – P. 1192–1209. – DOI 10.1109/SURV.2012.110712.00092. – Текст : непосредственный.

96. Chen, M. Smart healthcare: A survey of internet of things and cloud-based applications / M. Chen, Y. Ma, J. Song // IEEE Internet of Things Journal. – 2016. – Vol. 3, № 2. – P. 147–159. – Текст : непосредственный.

97. Patel, S. A review of wearable sensors and systems with application in rehabilitation / S. Patel, C. Hughes, N. Huggins // *Journal of NeuroEngineering and Rehabilitation*. – 2012. – Vol. 9. – Art. 21. – DOI 10.1186/1743-0003-9-21. – Текст : непосредственный.

98. Alam, E. A review of wearable technologies and machine learning for dementia care / E. Alam, T. Liu // *Sensors*. – 2020. – Vol. 20, № 14. – Art. 3941. – Текст : непосредственный.

99. Strain, T. Tracking physical activity using wearable devices / T. Strain, K. Wijndaele // *The Lancet Digital Health*. – 2020. – Vol. 2, № 8. – P. e398-e399. – Текст : непосредственный.

100. Camomilla, V. Motion analysis in sports / V. Camomilla, A. Lees // *Journal of Sports Sciences*. – 2018. – Vol. 36, № 1. – P. 1-2. – Текст : непосредственный.

101. Wang, J. Human activity recognition in industrial environments / J. Wang, Y. Chen // *IEEE Transactions on Industrial Informatics*. – 2019. – Vol. 15, № 5. – P. 2984-2993. – Текст : непосредственный.

102. Igual, R. A review of fall detection systems / R. Igual, C. Medrano // *Sensors*. – 2017. – Vol. 17, № 4. – Art. 739. – Текст : непосредственный.

103. Zhang, M. Aggressive behavior recognition using wearable sensors / M. Zhang, H. Li // *IEEE Sensors Journal*. – 2021. – Vol. 21, № 4. – P. 5123-5131. – Текст : непосредственный.

104. Reiss, A. Creating and benchmarking a new dataset for physical activity monitoring / A. Reiss, D. Stricker // *International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. – 2012. – P. 40-45. – DOI 10.1109/BSN.2012.6217571. – Текст : непосредственный.

105. Atallah, L. Sensor placement for activity recognition / L. Atallah, G.-Z. Yang // *IEEE Pervasive Computing*. – 2011. – Vol. 10, № 4. – P. 12-15. – Текст : непосредственный.

106. Wang, X. Extreme learning machine for human activity recognition / X. Wang, Y. Liu // *Neurocomputing*. – 2015. – Vol. 149. – P. 310-318. – Текст : непосредственный.
107. Smith, S. L. Don't decay the learning rate, increase the batch size / S. L. Smith, P.-J. Kindermans, C. Ying, Q. V. Le // *arXiv preprint arXiv:1711.00489*. – 2017. – Текст : электронный.
108. Vitter, J. S. Random sampling with a reservoir / J. S. Vitter // *ACM Transactions on Mathematical Software*. – 1985. – Vol. 11, № 1. – P. 37-57. – DOI 10.1145/3149.3181. – Текст : непосредственный.
109. Абрамова, Е. С. Сравнение машины экстремального обучения и метода обратного распространения ошибки при распознавании физической активности человека / Е. С. Абрамова // *Известия Тульского государственного университета. Технические науки*. – 2023. – № 2. – С. 117–124. – DOI 10.24412/2071-6168-2023-2-117-124. – Текст : непосредственный.
110. Abramova, E. S. Neural Network Alternate Incremental Learning Algorithm for Intelligent Human Activity Recognition System / E. S. Abramova, A. A. Orlov // *2023 International Russian Automation Conference (RusAutoCon)*. – Sochi : IEEE, 2023. – P. 261–265. – DOI 10.1109/RusAutoCon58002.2023.10272894. – Текст : непосредственный.
111. Abramova, E. S. Comparing Incremental Learning Methods for Human Activity Recognition / E. S. Abramova, A. A. Orlov, O. A. Abramov // *2023 Dynamics of Systems, Mechanisms and Machines (Dynamics)*. – Omsk : IEEE, 2023. – P. 1–5. – DOI 10.1109/Dynamics60586.2023.10349655. – Текст : непосредственный.
112. Abramova, E. S. Optimal Parameters Determination for Extreme Learning Machine in the Human Activity Recognition / E. S. Abramova, A. A. Orlov // *2023 International Russian Smart Industry Conference (SmartIndustryCon)*. – Sochi : IEEE, 2023. – P. 141–145. – DOI 10.1109/SmartIndustryCon57312.2023.10110761. – Текст : непосредственный.

Приложение А Акты о внедрении

(обязательное)

«УТВЕРЖДАЮ»

Директор МИ ВлГУ, д.т.н., профессор

А.Л. Жизняков

«08»

09

2026 г.

АКТ

об использовании результатов диссертационной работы Абрамовой Е.С.
в учебном процессе Муромского института (филиала) ФГБОУ ВО
«Владимирский государственный университет имени
Александра Григорьевича и Николая Григорьевича Столетовых»

Настоящим актом подтверждается, что результаты диссертационной работы Абрамовой Елены Сергеевны внедрены в учебный процесс и научно-исследовательскую работу кафедры физики и прикладной математики.

Разработанные методы и алгоритмы инкрементного обучения нейронной сети на основе машины экстремального обучения внедрены в лекционные и практические занятия по дисциплинам «Введение в искусственный интеллект» и «Специальные главы математики», а также используются при выполнении курсовых работ и выпускных квалификационных работ студентами образовательной программы 01.03.02 Прикладная математика и информатика.

Применение данных материалов позволяет демонстрировать студентам современные подходы к решению проблемы катастрофического забывания и обработке потоковых данных в режиме реального времени.

В научно-исследовательской сфере результаты работы нашли применение при выполнении НИР по тематике обработки потоковых данных, а также в качестве инструментария для проведения вычислительных экспериментов.

Декан ФИТР, к.т.н.



М.Н. Рыжкова

Зав. кафедрой ФПМ, д.т.н.



А.А. Орлов

Рисунок А.1 – Акт о внедрении с кафедры ФПМ МИ ВлГУ



ООО «СТРОЙТАЙМ ПЛЮС»

129085, г. Москва, вн.тер. г. Муниципальный округ Останкинский, ул. Годовикова, д.9,
стр.25, пом.2/2

ИНН 3334020837, КПП 771701001, ОГРН 1133334001877

р/сч 40702810110000007508 во Ульяновском Отделении № 8588 ПАО Сбербанк

Электронная почта: info@st-plus33.ru

Исх. № 54 от 03.09 2026г.

АКТ

о внедрении (практическом применении)
результатов диссертационного исследования
Абрамовой Елены Сергеевны

Результаты диссертационной работы Абрамовой Е.С. на соискание ученой степени кандидата технических наук представляют практический интерес для ООО «СТРОЙТАЙМ ПЛЮС» и приняты к внедрению в деятельности компании.

Разработанные методы и алгоритмы инкрементного обучения нейронных сетей на основе машины экстремального обучения, апробированные на задачах классификации физической активности человека, позволяют решать задачи распознавания действий персонала по данным носимых устройств и датчиков в режиме реального времени с адаптацией модели к индивидуальным особенностям движений сотрудников.

Практическая значимость результатов заключается в возможности их применения для автоматизированного контроля соблюдения техники безопасности на строительных объектах, выявления потенциально опасных действий и мониторинга двигательной активности персонала. Полученные результаты обладают практической ценностью и рекомендуются к использованию в системах контроля охраны труда и производственной безопасности компании.

Генеральный директор



А.А. Стулов

Рисунок А.2 – Акт о внедрении из компании ООО «СТРОЙТАЙМ ПЛЮС»

Приложение Б Свидетельства о государственной регистрации программ для ЭВМ

(обязательное)

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО
о государственной регистрации программы для ЭВМ

№ 2023665974

Модуль сбора данных с датчиков смартфона

Правообладатель: **Федеральное государственное бюджетное образовательное учреждение высшего образования "Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых" (RU)**

Авторы: **Глазунов Дмитрий Дмитриевич (RU), Макаров Кирилл Владимирович (RU), Абрамова Елена Сергеевна (RU)**

Заявка № **2023665190**
Дата поступления **19 июля 2023 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ **25 июля 2023 г.**



Руководитель Федеральной службы
по интеллектуальной собственности

Ю.С. Зубов

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 429b6a0fe7853164b9f96f83b73b4aa7
Владелец: **Зубов Юрий Сергеевич**
Действителен с 10.05.2023 по 02.08.2024

Рисунок Б.1 – Модуль сбора данных с датчиков смартфона



Рисунок Б.2 – Программа распознавания видов физической активности с помощью машины экстремального обучения