ЦЕНТРАЛЬНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«07» марта 2024 г.
Протокол №1

Рабочая программа дисциплины (модуля)
«Эффективное машинное обучение»

Направление подготовки: *02.03.01 Математика и компьютерные науки*

Направленность (профиль) подготовки: *Программа двух дипломов НИУ ВШЭ и ЦУ «Прикладная математика и информатика»*

Квалификация (степень) выпускника: *бакалавр*

Форма обучения: *очная*

Срок освоения программы: *4 года*

Год набора: *2024*

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 0215B0C000D1B3FEB44A37EE631E0C27D4
Владелец: Ивашкевич Евгений Васильевич
Действителен: с 13.01.2026 до 13.04.2027

Электронный документ

Москва
2024

Содержание

1. **Краткая характеристика дисциплины (модуля)** 3
2. **Перечень планируемых результатов обучения** 5
3. **Тематический план** 7
4. **Содержание дисциплины (модуля)** 7
5. **Учебно-методическое обеспечение** 9
6. **Материально-техническое обеспечение** 9
7. **Методические и оценочные материалы** 11

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Эффективное машинное обучение» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению 02.03.01 «Математика и компьютерные науки», профиль «Программа двух дипломов НИУ ВШЭ и ЦУ «Прикладная математика и информатика», утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Эффективное машинное обучение» позволяет преодолеть ключевое ограничение современного ИИ — высокие вычислительные затраты, обеспечивая переход от прототипирования к промышленному развёртыванию. Полученные знания и навыки критически важны для создания масштабируемых, энергоэффективных и производительных систем машинного обучения в условиях ограниченных ресурсов.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 «Математика и компьютерные науки», профиль «Программа двух дипломов НИУ ВШЭ и ЦУ «Прикладная математика и информатика» и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений, как дисциплина по выбору.

Дисциплина (модуль) изучается на 3 курсе в 6 семестре. Доступна к изучению после успешного освоения дисциплин (модулей): «Обучение с подкреплением», «Теория управления».

Цель изучения дисциплины (модуля): сформировать глубокое понимание архитектуры современных вычислительных устройств и развить практические навыки разработки, оптимизации и развёртывания высокопроизводительных моделей машинного обучения на GPU.

Задачи изучения дисциплины (модуля):

— изучить архитектурные особенности gpu, включая иерархию памяти, модель simt и выполнение варпов, для эффективного программирования на уровне ядра;

— освоить программную модель cuda и triton, включая индексацию, типы памяти и методы организации слитного доступа и использования разделяемой памяти;

— научиться разрабатывать и оптимизировать пользовательские ядра и операции, интегрируемые в pytorch, с поддержкой автоматического дифференцирования;

— изучить методы повышения эффективности обучения и инференса: смешанную точность, контрольное сохранение, квантизацию и распределённые стратегии (ddp, fsdp);

— освоить инструменты профилирования и анализа производительности для выявления узких мест и проведения сквозной оптимизации моделей от разработки до развёртывания.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

— архитектурные особенности современных GPU (иерархия памяти, модель SIMT, исполнение варпов);

— программную модель CUDA: индексацию, типы памяти, правила слитного доступа (coalesced access) и использования разделяемой памяти (shared memory);

— принципы написания высокопроизводительных ядер на CUDA и Triton;

— механизм автоматического дифференцирования в PyTorch и способы создания пользовательских операций с поддержкой обратного распространения ошибки (backward);

— методы смешанной точности, градиентного контрольного сохранения (checkpointing) и распределённого обучения;

— инструменты профилирования (Nsight Systems, PyTorch Profiler) и стратегии оптимизации (слияние операций (fusion), квантизация).

уметь:

— писать, отлаживать и профилировать CUDA-ядра на C++, интегрируя их в Python-проекты через расширения PyTorch;

— разрабатывать оптимизированные ядра на Triton с эффективным разбиением на тайлы (tiling) и управлением памятью;

— использовать torch.compile и torch.fx для автоматического ускорения графа вычислений;

— применять автоматическую смешанную точность, распределённые стратегии обучения (DDP, FSDP) и техники экономии памяти;

— анализировать узкие места производительности моделей и проводить целенаправленную оптимизацию;

— квантизировать модели и экспортировать их для высокоэффективного вывода (инференса).

владеть:

— навыками работы с CUDA Toolkit, Nsight Compute/Systems и PyTorch Profiler для поиска и устранения неэффективностей;

— практикой создания пользовательских расширений PyTorch (C++/CUDA) с обратным распространением ошибки;

— инструментарием языка Triton для быстрого прототипирования производительных GPU-программ;

— техниками оптимизации использования памяти (слитный доступ (coalescing), разбиение на тайлы (tiling), разделяемая память, перевычисление (recomputation));

— методикой настройки эффективного многокарточного обучения с минимизацией коммуникационных издержек;

— умением выполнять сквозной цикл оптимизации от профилирования до практического развёртывания ускоренной модели.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
УК-1.	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1.	Знает методы поиска и анализа информации в области искусственного интеллекта, основные принципы критической оценки источников информации и их релевантности.
		УК-1.2.	Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем
		УК-1.3.	Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации
УК-2.	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1.	Знает действующие правовые нормы, регулирующие деятельность в области решения задач, основные методы и подходы к определению круга задач
		УК-2.2.	Умеет определять круг задач в рамках поставленной цели, выбирать оптимальные способы решения задач, учитывая имеющиеся ресурсы и ограничения
		УК-2.3.	Имеет практический опыт применения знаний о правовых нормах и ресурсах в реальных ситуациях, разработки и реализации решений в соответствии с установленными ограничениями
ОПК-1.	Способен находить, формулировать и решать актуальные и значимые проблемы прикладной и компьютерной математики	ОПК-1.1.	Знает основные методы и подходы к решению задач прикладной и компьютерной математики, включая алгоритмы, математическое моделирование и теорию оптимизации, а также современные инструменты и технологии, используемые в этой области

		ОПК-1.2.	Умеет анализировать и формулировать математические задачи, применять соответствующие методы и алгоритмы для их решения, а также интерпретировать и представлять результаты в понятной и доступной форме
		ОПК-1.3.	Имеет практический опыт работы над проектами или исследованиями в области прикладной и компьютерной математики, включая участие в конкурсах, олимпиадах или научных публикациях, где были решены актуальные и значимые задачи
ПК-1.	Способен формулировать задачи с математической точностью, обосновывать утверждения строго и анализировать полученные результаты в области математики и компьютерных наук	ПК-1.1.	Знает методы и подходы к формулированию задач, а также основные принципы математического доказательства и анализа результатов.
		ПК-1.2.	Умеет корректно ставить и формулировать математические задачи, применять строгие методы доказательства и анализировать полученные результаты.
		ПК-1.3.	Имеет опыт работы с задачами в области математики и компьютерных наук, включая применение математических методов для решения практических задач
ПК-2.	Способен решать типовые задачи профессиональной деятельности в области искусственного интеллекта, опираясь на информационную и библиографическую культуру, используя информационно-коммуникационные технологии и учитывая основные требования информационной безопасности	ПК-2.1.	Знает основы информационной и библиографической культуры, а также принципы информационной безопасности и применения информационно-коммуникационных технологий в профессиональной деятельности
		ПК-2.2.	Умеет эффективно использовать информационно-коммуникационные технологии для решения стандартных задач профессиональной деятельности, учитывая требования информационной безопасности
		ПК-2.3.	Имеет опыт работы с информационными ресурсами и технологиями в области искусственного интеллекта, включая соблюдение норм информационной безопасности

3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятельная работа	
		Лекции	Практические занятия			
1	Архитектура GPU и основы программирования CUDA	6	6		26	Домашнее задание Контест
2	Интеграция с PyTorch и продвинутое программирование CUDA	10	10		34	Домашнее задание Контест
3	Профилирование и системная оптимизация PyTorch-моделей	6	6		26	Домашнее задание Контест
4	Специализированные ядра и практический проект	6	6		42	Домашнее задание Контест
	Зачёт с оценкой			4		Проект
	Итого:	***26***	***26***	***4***	***134***	
Объем дисциплины (модуля) (в ак. ч.)		***190***				
Объем дисциплины (модуля) (в зач. ед.)		***5***				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Архитектура GPU и основы программирования CUDA	Гетерогенные вычисления и архитектура GPU: потоковые мультипроцессоры, варпы, иерархия памяти. Модель выполнения CUDA. Обзор инструментов (nvcc, Nsight). Основы CUDA-ядер: индексация потоков, работа с глобальной памятью. Поэлементные операции и редукция. Запуск ядер из Python (cupy, PyCUDA) как мост к PyTorch. Оптимизация доступа к памяти: слитный доступ (coalesced access), разделяемая память и конфликты банков памяти. Профилирование простых ядер с помощью nvprof и Nsight Compute
2	Интеграция с PyTorch и продвинутое программирование CUDA	Продвинутые техники оптимизации ядер: разворачивание циклов, примитивы уровня варпов (перемешивание), атомарные операции. Оптимизация пропускной способности и задержки. Пользовательские расширения PyTorch: создание модуля на C++/CUDA с pybind11 и setuptools. Реализация прямого и обратного прохода для автоматического дифференцирования.

		Введение в Triton: написание пользовательских ядер на языке Triton, разбиение на тайлы (tiling), автоматическая оптимизация под архитектуру GPU. Сравнение с «ручными» CUDA-ядрами. Смешанная точность и оптимизация памяти: torch.cuda.amp, FP16/BF16, управление памятью PyTorch (распределитель, контрольное сохранение). Слияние ядер через nvFuser. Многопроцессорное обучение: DistributedDataParallel, стратегии шардирования (FSDP), коммуникации NCCL, перекрытие вычислений и передачи данных
3	Профилирование и системная оптимизация PyTorch-моделей	Инструменты профилирования: PyTorch Profiler, трассировка, пламя-графики (flame graphs). Анализ узких мест и использование Nsight Systems для сквозной визуализации работы GPU. Компиляторные оптимизации: torch.compile, бэкенды (Inductor, TensorRT), слияние операторов. Работа с torch.fx для преобразования графа вычислений. Квантизация и развёртывание (деплоймент): посттренировочная квантизация (PTQ), обучение с учётом квантизации (QAT), экспорт в TorchScript и ONNX, ускорение вывода с помощью TensorRT
4	Специализированные ядра и практический проект	Эффективные по памяти ядра для ключевых операций: FlashAttention, оптимизированный softmax/нормализация по слоям (layer normalization), слитые ядра. Анализ соотношения вычислительно-ограниченных и ограниченных по памяти задач. Современные тренды в эффективном машинном обучении: разреженность, условные вычисления, проектирование нейросетевых архитектур с учётом аппаратных особенностей

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Вишневский, В. М. Теория очередей и машинное обучение : монография / В.М. Вишневский, Д.В. Ефросинин. — Москва : ИНФРА-М, 2025. — 370 с. : ил. — (Научная мысль). - ISBN 978-5-16-020572-4. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2204580.

2. Родзин, С. И. Машинное обучение: метаэвристики дифференциально-векторного движения : учебное пособие / С. И. Родзин, О. Н. Родзина. - Чебоксары : Среда, 2024. - 140 с. - ISBN 978-5-907830-17-2. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2171753.

Дополнительная литература:

1. Степашкина, А. С. Численные методы и машинное обучение в метрологии : учебное пособие / А. С. Степашкина. - Москва ; Вологда : Инфра-Инженерия, 2024. - 144 с. - ISBN 978-5-9729-1954-3. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2170894.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование"	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое

Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Эффективное машинное обучение» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, домашние задания, контесты, проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удается, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Семинар – это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Контест – интерактивная платформа с заданиями разного уровня сложности и автоматической проверкой результатов.

Контест позволяет оперативно оценивать усвоение материала и выявлять пробелы в знаниях через тесты и практические задачи. Такой формат способствует регулярной самопроверке и повышает мотивацию к изучению дисциплины.

Проект – исследовательская работа по дисциплине (модулю) и презентация

результатов.

Для успешной подготовки к проекту рекомендуется: четко определить цели и задачи проекта; составить план работы, разбив проект на этапы с указанием сроков выполнения каждого из них; использовать разнообразные источники информации и инструменты для исследования темы; регулярно проверять прогресс и вносить коррективы в план, если это необходимо.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Эффективное машинное обучение»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме ***зачёта с оценкой***, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом, если иное не указано в силлабусе дисциплины:

Десятибалльная оценка	**Пятибалльная оценка**	**Общая характеристика результата обучения по дисциплине (модулю)**
10	Отлично	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	
8	Отлично	
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать
6	Хорошо	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
5	Удовлетворительно	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	
1	Не сдан	

Дисциплина (модуль) «Эффективное машинное обучение» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашние задания	40%	12	Набор задач по темам недели
Контест	40%	12	Интерактивные задания разного уровня сложности с автоматической проверкой результатов
Зачёт с оценкой	20%	1	Проект - Исследовательская работа по дисциплине (модулю) и презентация результатов

Формула расчёта итоговой оценки по дисциплине (модулю) «Эффективное машинное обучение»: «0,4 × среднее за домашние задания + 0,4 × среднее за контест + 0,2 × зачёт с оценкой».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1.

1. Объясните, как устроена иерархия памяти в современных GPU и в чём разница между глобальной, разделяемой и регистровой памятью;
2. Опишите, что такое варп и как модель SIMT влияет на производительность выполнения CUDA-ядер;
3. Реализуйте CUDA-ядро для поэлементного сложения двух массивов и запустите его из Python с использованием PyCUDA или CuPy;
4. Напишите ядро редукции суммы с использованием разделяемой памяти и объясните, как избежать конфликтов банков памяти;
5. Проанализируйте производительность простого ядра с помощью Nsight Compute и определите, является ли доступ к глобальной памяти слитным (coalesced).

Домашнее задание 2.

1. Создайте пользовательский PyTorch-модуль на C++/CUDA с помощью pybind11 и реализуйте прямой и обратный проход для операции, например, leaky relu;
2. Напишите ядро на Triton для умножения матриц, используя разбиение на тайлы (tiling), и сравните его производительность с torch.matmul;
3. Объясните, как примитивы уровня варпа (warp shuffle) позволяют ускорить выполнение редукций и атомарных операций;
4. Примените автоматическую смешанную точность (AMP) к обучению простой модели и оцените выигрыш по памяти и скорости;
5. Настройте обучение с использованием DistributedDataParallel (DDP) на двух GPU и измерьте ускорение, а также проанализируйте загрузку коммуникаций через NCCL.

Домашнее задание 3.

1. Используйте PyTorch Profiler для анализа производительности обучения небольшой нейросети и выявите узкие места: вычисления, ожидание памяти или коммуникации;
2. Примените torch.compile к модели и сравните время выполнения с и без компиляции; объясните, какие оптимизации были применены автоматически;
3. Преобразуйте граф вычислений с помощью torch.fx, вставив пользовательскую операцию, и продемонстрируйте возможность модификации графа;
4. Выполните посттренировочную квантизацию (PTQ) модели и оцените изменение качества и скорости инференса;
5. Экспортируйте модель в ONNX и запустите её с помощью TensorRT, измерив ускорение вывода по сравнению с исходным PyTorch.

Примерные задания для контеста

Контест 1.

1. Определите количество активных варпов в блоке из 128 потоков на GPU с размером варпа 32 и введите число;
2. Реализуйте CUDA-ядро на PyCUDA для поэлементного умножения двух массивов и запустите его из Python; введите результат умножения первого элемента;
3. Напишите ядро редукции суммы массива с использованием разделяемой памяти и избежанием конфликтов банков; введите результат суммирования массива из 1024 элементов;
4. Оцените, является ли доступ к глобальной памяти слитным (coalesced), если каждый поток в блоке из 32 читает элемент с шагом 1; введите "да" или "нет";
5. Проанализируйте профиль выполнения простого ядра в Nsight Compute и введите значение пропускной способности памяти (в GB/s) с точностью до одного знака.

Контест 2.

1. Напишите ядро на Triton для умножения матриц 64x64 с разбиением на тайлы 16x16; введите результат умножения элемента [0][0] с точностью до трёх знаков;
2. Создайте пользовательскую функцию PyTorch с кастомным backward на C++/CUDA через pybind11; реализуйте градиент для операции x^2 и введите значение градиента при x=3;
3. Включите автоматическую смешанную точность (AMP) при обучении простой модели и измерьте пиковое потребление памяти; введите значение в мегабайтах;
4. Настройте DistributedDataParallel (DDP) для обучения на двух GPU и введите общее время одной эпохи с точностью до двух знаков;
5. Используйте примитив warp shuffle в CUDA-ядре для агрегации максимума в пределах варпа; введите результат для заданного массива из 32 элементов.

Контест 3.

1. Запустите PyTorch Profiler для обучения модели и определите долю времени, затраченного на коммуникации (в %); введите число с точностью до целого;

2. Примените torch.compile к модели и измерьте ускорение времени инференса; введите отношение времени (без/с компиляцией) с точностью до двух знаков;

3. Преобразуйте граф модели с помощью torch.fx, вставив логирование перед слоем Linear; введите количество узлов в модифицированном графе;

4. Выполните посттренировочную квантизацию (PTQ) модели и измерьте время инференса; введите значение в миллисекундах с точностью до одного знака;

5. Экспортируйте модель в ONNX и запустите через TensorRT; введите скорость вывода (в изображениях/секунду) с точностью до целого.

Примерное описание задания и критерии оценивания к проекту

Цель проекта: разработать, интегрировать и оптимизировать пользовательское вычислительное ядро на CUDA или Triton, обеспечивающее ускорение ключевой операции в PyTorch, и провести сквозной анализ производительности с профилированием, сравнением и оценкой применимости в реальных сценариях.

Задачи проекта:

— выбрать вычислительный узел (например, пользовательская активация, слой нормализации, операция редукции или умножение матриц) и обосновать его значимость для производительности моделей глубокого обучения;

— реализовать эффективное ядро на CUDA с оптимизацией доступа к памяти (слитный доступ, разделяемая память) или на Triton с использованием разбиения на тайлы (tiling) и автоматической векторизацией;

— интегрировать ядро в PyTorch: создать пользовательский модуль с поддержкой прямого и обратного прохода через расширение на C++/CUDA или через torch.compile;

— провести тестирование производительности: сравнить время выполнения с нативными операциями PyTorch (например, torch.matmul, torch.softmax) на разных размерах тензоров и архитектурах GPU;

— выполнить профилирование с помощью Nsight Systems и PyTorch Profiler, выявить узкие места и предложить пути дальнейшей оптимизации;

— подготовить отчёт и презентацию, включающие описание архитектуры ядра, результаты замеров, графики ускорения, анализ эффективности и обсуждение возможности использования в промышленных моделях.

Критерии оценивания проекта:

— корректность реализации ядра и его численная точность по сравнению с эталонной реализацией;

— качество оптимизации: использование слитного доступа, разделяемой памяти, разбиения на тайлы и других техник повышения эффективности;

— успешность интеграции в PyTorch с поддержкой автоматического дифференцирования и совместимостью с графом вычислений;

— глубина анализа производительности: корректное использование профилировщиков, интерпретация метрик и обоснование результатов;

— полнота отчёта, визуализация данных и реалистичная оценка применимости решения в реальных условиях, включая потенциал для развёртывания.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1	Назовите основные уровни иерархии памяти в gpu и их относительную скорость и объём	глобальная, разделяемая, кэш, регистры	УК-1

2	Объясните, что такое варп и как он связан с моделью выполнения simt	группа из 32 потоков, выполняемых одновременно на sm	УК-1
3	Укажите, зачем используется индексация потоков blockIdx, threadIdx и blockDim в cuda	для однозначного определения позиции потока в блоке и сетке	УК-1
4	Назовите тип доступа к глобальной памяти, при котором память читается наиболее эффективно	слитный доступ / coalesced access	УК-1
5	Объясните, как разделяемая память может ускорить выполнение ядра и при каких условиях возникают конфликты банков памяти	позволяет многократно использовать данные; конфликты при одновременном доступе к одному банку	УК-2
6	Назовите инструмент для детального профилирования производительности cuda-ядер на уровне инструкций	nvidia nsight compute	УК-2
7	Укажите, как примитивы уровня варпа (warp shuffle) позволяют избежать использования разделяемой памяти при обмене данными между потоками	передача данных напрямую между потоками в пределах варпа	УК-2
8	Объясните, зачем используется атомарная операция при одновременной записи в одну ячейку памяти несколькими потоками	для предотвращения гонки данных / race condition	УК-2
9	Назовите способ интеграции пользовательского cuda-ядра в pytorch с поддержкой обратного прохода	создание custom function с forward и backward через c++/cuda и pybind11	ОПК-1
10	Объясните, как работает автоматическое дифференцирование в pytorch при использовании кастомных операций	через сохранение контекста и реализацию функции backward	ОПК-1
11	Назовите язык, позволяющий писать эффективные gpu-ядра без глубоких знаний cuda, с автоматической оптимизацией под архитектуру	triton	ОПК-1
12	Объясните, как разбиение на тайлы (tiling) в triton улучшает использование кэша и пропускной способности памяти	уменьшает количество обращений к глобальной памяти за счёт локальной обработки блоков	ОПК-1
13	Назовите стратегию, позволяющую уменьшить потребление памяти при обучении за счёт перевычисления активаций	контрольное сохранение / gradient checkpointing	ПК-1
14	Укажите, какая техника позволяет комбинировать fp16 и fp32 для ускорения обучения без потери стабильности	автоматическая смешанная точность / amp	ПК-1

15	Назовите метод распределённого обучения, при котором параметры модели реплицируются на нескольких gpu с синхронизацией градиентов	distributeddataparallel / ddp	ПК-1
16	Назовите инструмент pytorch для визуализации узких мест в производительности модели	pytorch profiler	ПК-2
17	Объясните, как torch.compile ускоряет выполнение модели	за счёт слияния операторов, устранения накладных расходов и использования бэкендов (inductor)	ПК-2
18	Назовите метод ускорения инференса, при котором веса модели преобразуются в более низкую точность (например, int8)	квантизация	ПК-2
19	Укажите, какая архитектура ядра используется в flashattention для уменьшения обращений к памяти	tile-based вычисления с кэшированием в разделяемой памяти	ПК-2
20	Назовите современный тренд, при котором часть вычислений пропускается динамически для ускорения вывода	условные вычисления / conditional computation	ПК-2