

КОМПИЛЯТОРЫ ДЛЯ ГЛУБОКИХ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ

КРАТКОЕ ОПИСАНИЕ

В курсе изучаются вопросы построения и использования тензорных компиляторов, предназначенных для ускорения вывода глубоких нейронных сетей. Курс является практико-ориентированным и нацелен на освоение процедуры оптимизации глубоких моделей, анализа и сравнения производительности вывода.

Рассматриваются следующие вопросы.

1. История возникновения и развития глубокого обучения. Примеры эффективно решаемых задач. Общая схема решения задач с использованием глубокого обучения.
2. Основные виды нейросетевых моделей.
3. Программные инструменты для обучения и вывода глубоких моделей.
4. Тензорные компиляторы: понятие, общая схема построения.
5. Существующие тензорные компиляторы, ориентированные на frontend-разработчика, сравнение их возможностей.
6. Функционал и основные возможности Apache TVM (TVM).
7. Автоматическая оптимизация нейросетевых моделей средствами TVM. Анализ производительности вывода глубоких нейронных сетей.

Курс включает в себя 18 часов лекций (7 лекций продолжительностью от 1 до 4 академических часов в зависимости от сложности темы) и 18 часов практических занятий. Лекции проводятся в классической форме с элементами мастер-класса и сопровождаются примерами решения практических заданий. Для демонстрации выполнения практических заданий используются глубокие модели, обеспечивающие решение задачи классификации изображений – классической задачи компьютерного зрения. В процессе выполнения практических работ осуществляется последовательный запуск и пошаговая оптимизация широко известных моделей с целью достижения наилучших показателей производительности вывода. Итоговый контроль знаний предполагает представление результатов выполнения практических заданий.

Курс ориентирован на студентов и аспирантов технических и естественно-научных специальностей, а также инженеров, преподавателей вузов и исследователей.

ПРЕДВАРИТЕЛЬНЫЕ ТРЕБОВАНИЯ

Курс ориентирован на слушателей, имеющих базовые навыки разработки программ на языках C/C++ и Python. Наряду с этим, изучение курса предполагает наличие теоретических знаний и практических навыков в следующих областях: линейная алгебра, алгоритмы и структуры данных, архитектуры, операционные системы. Наличие базовых знаний в области оптимизации программ, параллельном программировании, глубоком обучении, компьютерном зрении ускорит освоение материала, но не является обязательным.

СТРУКТУРА КУРСА

Курс имеет следующую структуру.

Тема	Содержание	Часы	
		Лекции	Практика
Лекция 1. Введение в глубокое обучение.	1. История развития глубокого обучения. 2. Примеры практических задач, успешно решаемых с использованием глубоких нейронных сетей.	1	–

	3. Общая схема решения практических задач средствами глубокого обучения (подготовка данных и инфраструктуры, обучение, тестирование, внедрение).		
Лекция 2. Основные виды нейросетевых моделей.	1. Детерминистская модель нейрона. 2. Полносвязные нейронные сети (полносвязный слой, функции активации). 3. Сверточные нейронные сети (свертка, активация, пуллинг). 4. Обучение нейронных сетей. Постановка задачи, общая схема метода обратного распространения ошибки. 5. Примеры архитектур полносвязных и сверточных нейронных сетей. Анализ результатов качества решения задачи классификации рукописных цифр с использованием сети, соответствующей логистической регрессии, и простой сверточной сети, описание процедуры оценки производительности вывода (определение оптимальных параметров запуска) на примере PyTorch (без демонстрации кода для обучения и тестирования моделей).	2	–
Лекция 3 (в формате мастер-класса). Обзор инструментов глубокого обучения.	1. Инструменты для обучения и тестирования глубоких моделей. – Обзор существующих инструментов. – Библиотека PyTorch: особенности, представление модели нейронной сети, установка и программный интерфейс, пример обучения и тестирования. 2. Инструменты для вывода глубоких нейронных сетей. – Обзор инструментов. – Инструментарий OpenVINO: особенности, установка и программный интерфейс, пример реализации вывода. – Библиотека TensorFlow Lite (сейчас Lite RT): особенности, установка и программный интерфейс, пример реализации вывода.	3	–
Практика 1. Программная реализация вывода глубоких нейронных сетей для классификации изображений средствами нескольких фреймворков.	1. Постановка задачи классификации изображений. 2. Выбор открытой модели для решения поставленной задачи. Изучение ее архитектуры. 3. Изучение программного интерфейса двух-трех фреймворков для вывода открытых нейросетевых моделей (PyTorch, TensorFlow, TensorFlow Lite, OpenVINO toolkit, другие). Выбор фреймворков зависит от формата модели, наличия необходимых конвертеров и целевого аппаратного обеспечения, на котором далее планируется выполнять запуск вывода. 4. Программная реализация вывода выбранной модели с использованием изученных фреймворков. 5. Анализ качества решения задачи. 6. Анализ производительности вывода и определение оптимальных параметров запуска вывода на выбранной аппаратуре	–	4

	(размер пачки входных данных, количество потоков, другие).		
Лекция 4. Общая схема построения тензорных компиляторов.	1. Обзор общих подходов к оптимизации. 2. Понятие тензорного компилятора. 3. Общая архитектура тензорных компиляторов для глубокого обучения. 4. Представление нейросетевой модели в виде графа вычислений. 5. Типовые подходы к оптимизации графа вычислений. 6. Типовые подходы к оптимизации кода операторов (аппаратно-зависимые оптимизации).	3	–
Лекция 5. Обзор тензорных компиляторов.	1. Назначение тензорных компиляторов. 2. Обзор тензорных компиляторов, ориентированных на frontend-разработчика. 3. Glow. 4. XLA. 5. NNFusion. 6. OpenVINO. 7. Apache TVM. 8. Сравнение возможностей тензорных компиляторов.	2	–
Лекция 6. Обзор Apache TVM.	1. Общая информация об инструменте. 2. Последовательность оптимизации нейронных сетей. 3. Конвертация модели в граф вычислений. 4. Оптимизация графа вычислений нейронной сети. 5. Переход к тензорному внутреннему представлению. 6. microTVM. 7. Сборка и установка Apache TVM. 8. Программный интерфейс Apache TVM для вывода нейронных сетей.	3	–
Практика 2. Вывод глубокой модели средствами Apache TVM. Анализ и сравнение производительности вывода.	1. Подготовка тестовых моделей для решения задачи классификации изображений. Изучение архитектуры выбранных моделей. 2. Изучение программного интерфейса Graph Execution (GE API) для вывода глубоких моделей фреймворка Apache TVM. 3. Программная реализация вывода выбранных моделей с использованием GE API фреймворка Apache TVM. Валидация корректности разработанной программной реализации. 4. Программная реализация вывода выбранных моделей с использованием других фреймворков, которые могут быть запущены на RISC-V-устройствах. В данной работе в качестве примера такого фреймворка используется TensorFlow Lite, поскольку на момент подготовки практической работы он оптимизирован под RISC-V-архитектуры. 5. Анализ качества решения задачи классификации с помощью выбранного набора моделей. При этом вывод запускается с использованием TVM и TensorFlow Lite. 6. Анализ производительности вывода и определение оптимальных параметров запуска вывода на устройстве с процессором	–	4

	RISC-V (размер пачки входных данных, количество потоков, другие при наличии).		
Практика 3. Реализация и оптимизация тензорного выражения на примере матричного умножения в рамках Apache TVM.	1. Сборка и установка необходимых инструментов и библиотек: LLVM, OpenBLAS для линковки и вызова высокопроизводительного матричного умножения. Сборка Apache TVM с LLVM и OpenBLAS. 2. Реализация «наивного» умножения матриц по определению с помощью Apache TVM (тензорное выражение с пустым планом вычислений). 3. Рассмотрение нескольких этапов оптимизации матричного умножения с помощью реализации новых планов вычислений. – Перестановка циклов для устранения непоследовательного доступа к памяти. – Реализация блочного алгоритма для улучшения повторного использования данных. 4. Замена тензорного выражения на вызов обертки матричного умножения из высокопроизводительной библиотеки OpenBLAS. 5. Сравнение производительности разработанных реализаций.	–	4
Лекция 7. Автоматическая оптимизация вывода нейросетевых моделей на уровне слоев в Apache TVM.	1. Автоматическая оптимизация вывода нейросетевых моделей на уровне слоев. 2. AutoTVM: краткая информация, схема и алгоритмы работы. 3. Auto-scheduler: краткая информация, схема и алгоритмы работы. 4. MetaSchedule: краткая информация, схема и алгоритмы работы.	4	–
Практика 4. Автоматическая оптимизация нейронной сети с помощью Apache TVM.	1. Предварительная подготовка моделей для последующей оптимизации – обучение модели, соответствующей логистической регрессии, и многослойной полносвязной сети. 2. Оптимизация модели логистической регрессии (загрузка, запуск и проверка корректности, оптимизация с использованием AutoTVM, Auto-scheduler и MetaScheduler, анализ и сравнение результатов оптимизации). 3. Оптимизация многослойной полносвязной нейронной сети (загрузка, запуск и проверка корректности, оптимизация с использованием AutoTVM, Auto-scheduler и MetaScheduler, анализ и сравнение результатов оптимизации).	–	6
Форма отчетности: представление проекта, экзамен/зачет			
Итого:		18	18

МАТЕРИАЛЫ КУРСА

Материалы курса:

1. *Программа курса.*
2. *Лекции: 7 лекций.* Каждая лекция сопровождается презентацией.

3. *Практика*: 4 практических работы. Каждая практика сопровождается тьюториалом и учебным примером в исходных кодах, который демонстрирует реализацию практических заданий (в виде IPython notebooks); учебный пример сопровождается пошаговым описанием выполнения заданий. Также прилагается перечень возможных практических заданий (набор классификационных моделей, для которых можно выполнить задания отдельных практических работ).

ЯЗЫК

Русский

ЛИЦЕНЗИЯ

Свободный доступ. Материалы планируется выложить на сайте ННГУ под лицензией Apache 2.0 [<http://www.apache.org/licenses/LICENSE-2.0.txt>].

АВТОРЫ КУРСА

Мееров Иосиф Борисович, к.т.н., доцент, зав. каф. ВВиСП
Института ИТММ ННГУ им. Н.И. Лобачевского

Кустикова Валентина Дмитриевна, к.т.н., доцент каф. ВВиСП
Института ИТММ ННГУ им. Н.И. Лобачевского

Родимков Юрий Александрович, младший научный сотрудник каф. ВВиСП
Института ИТММ ННГУ им. Н.И. Лобачевского

Сысоев Александр Владимирович, к.т.н., доцент каф. ВВиСП
Института ИТММ ННГУ им. Н.И. Лобачевского