

ВЫЗОВЫ МАШИННОГО ОБУЧЕНИЯ И МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

© 2025 г. В.П. Ильин^{a,b,*}

^aИнститут вычислительной математики и математической геофизики СО РАН, Новосибирск, Россия

^bНовосибирский государственный технический университет, Новосибирск, Россия

*E-mail: ilin@sccc.ru

Поступила в редакцию 16.10.2024 г.

После доработки 21.01.2025 г.

Принята к публикации 14.02.2025 г.

В статье рассмотрены вызовы и проблемы машинного обучения, возникающие в суперкомпьютерном математическом моделировании процессов и явлений реального мира. В настоящее время такое моделирование стало основным инструментом получения фундаментальных и прикладных знаний, а также условием заметного повышения производительности труда и внутреннего валового продукта. Описаны принципы современного предсказательного моделирования на основе высокопроизводительных вычислений, искусственного интеллекта и обработки больших объёмов данных. Проанализированы тенденции развития наукоёмкого математического и программного обеспечения в рамках интегрированных вычислительных окружений; последние предполагают гибкое расширение состава изучаемых моделей и применяемых алгоритмов, эффективное использование внешних продуктов, адаптацию к эволюции компьютерных платформ, ориентированных на длительный жизненный цикл. Изложена методология машинного обучения на основе технологического цикла, который включает формирование и модификацию моделей, реализацию вычислительного эксперимента с решением прямых и обратных задач, анализ результатов и принятие решений по оптимизации видов деятельности для достижения поставленных целей.

Ключевые слова: машинное обучение, математическое моделирование, высокопроизводительные вычисления, искусственный интеллект, большие данные, наукоёмкое программное обеспечение, интегрированные вычислительные окружения.

DOI: 10.31857/S0869587325030021, EDN: STOINN

Задачи опережающего научно-технологического развития в приоритетных областях в наступившую эпоху эксафлопсных суперкомпьютеров неизбежно должны опираться на массовую цифровизацию всех сфер жизнедеятельности человеческого общества при условии активного использования искус-

ственного интеллекта, огромных объёмов данных, нейросетей и методологий машинного обучения. Особая роль принадлежит средствам получения новых фундаментальных и прикладных знаний, базирующимся на наукоёмком математическом моделировании сложнейших процессов и явлений. Эти исследования, как правило, включают решение междисциплинарных прямых и обратных задач и обуславливают вовлечение экстремальных объёмов вычислительных ресурсов с масштабируемым распараллеливанием алгоритмов на многопроцессорных вычислительных системах (МВС). Принципиально, что в условиях непрерывно развивающихся наук проблемы моделирования неизбежно связаны с динамичными разработками, сопровождением и внедрением математического и программного обеспечения (МПО), высокая стоимость которого требует экономичного отношения к профессиональной его эксплуатации. С другой



ИЛЬИН Валерий Павлович — доктор физико-математических наук, главный научный сотрудник лаборатории вычислительной физики ИВМиМГ СО РАН.

стороны, пользователи МПО должны владеть суперкомпьютерными технологиями получения знаний и принимать решения на основе их анализа.

Важный элемент такого подхода — построение цифровых двойников, или виртуальных реальностей, изучение и оптимизация характеристик которых позволяют повышать эффективность труда; фактически речь идёт о новой производительной силе, которая обеспечивает рост валового внутреннего продукта. Не менее значимо проникновение интеллектуальных вычислительно-информационных новаций в социальную и гуманитарную сферы, не говоря уже о национальной безопасности. Образно говоря, компьютерные науки и инструментарий играют роль кровеносной системы, обеспечивающей жизнедеятельность человеческого сообщества как единого организма. Такая ситуация требует по-новому взглянуть на взаимосвязь высокопроизводительных вычислений с искусственным интеллектом и преобразованиями больших данных — взаимосвязь трёх китов 4-й индустриальной революции.

По всем этим направлениям за последние десятилетия достигнуты выдающиеся успехи, требующие философско-методологического осмысления, имея в виду гармоничное развитие совокупности наук о получении знаний [1–7]. Надо иметь в виду, что, например, появившиеся возможности достижения положительных практических результатов на основе когнитивного анализа больших данных с помощью их простой статистической обработки иногда имеют своим следствием вывод о возникновении некоей “постнауки”, вроде Data Science, которая якобы не требует традиционных глубоких исследований. Чтобы не ввести в заблуждение подобными сомнительными утверждениями, необходимо показать, что для создания искусственного интеллекта и нейросетей необходимо вложить огромный труд профессионалов с высоким уровнем естественного интеллекта. За последние десятилетия наукоёмкое математическое моделирование сформировалось в многоликий творческий процесс, а бурные успехи в области искусственного интеллекта активно влияют на технологические стадии машинного эксперимента, что сулит значительные синергические эффекты.

Настоящая работа посвящена системному анализу указанной проблематики с точки зрения математика, программиста и пользователя-специалиста в конкретной прикладной области. Будут рассмотрены: основные характеристики и тенденции математического моделирования как научной дисциплины и как технологии решения практических задач; функциональные и системные наполнения программного обеспечения; методологические принципы основанного на вычислительном эксперименте машинного обучения.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ: СУПЕРВЫЧИСЛЕНИЯ, ИНТЕЛЛЕКТУАЛИЗАЦИЯ, БОЛЬШИЕ ДАННЫЕ

Философское изречение “бытие определяет сознание” имеет прямое отношение к нашей теме в том смысле, что достижимый уровень искусственного интеллекта определяется мощностью ресурсов ЭВМ, включая быстродействие и объём памяти. Речь пойдёт только о традиционных архитектурах кластерного типа, включая вычислительные узлы с распределённой памятью и многоядерными процессорами (CPU с несколькими десятками ядер над общей памятью и, возможно, графическими ускорителями GPGPU).

Хотя и в нашей стране, и за рубежом активно ведутся разработки новых поколений компьютеров, в том числе квантовых и реконфигурируемых, в ближайшие 5–10 лет они, по-видимому, ещё не станут массовыми и конкурентоспособными. А значит, в течение указанного времени можно рассматривать конфигурации с суммарным быстродействием около петафлопса и оперативной памятью в несколько десятков или сотен терабайт. История эволюции ЭВМ показывает, что параметры их быстродействия и объёмов памяти растут примерно пропорционально. При этом достаточно “большими задачами” (в терминологии Н.Н. Яненко [8, 9], то есть теми, которые решают достаточно долго — в течение часов или десятков часов, суток или нескольких суток) сейчас можно считать такие, постановка которых предполагает решение междисциплинарных многомерных прямых и обратных задач с числом неизвестных функций около десятка и которые требуют применения неструктурированных адаптивных сеток с числом пространственных узлов порядка 1000^3 , а также выполнения шагов по времени до 10^3 – 10^5 и более. Для таких расчётных параметров вычислительный процесс даже на суперкомпьютере — это крупномасштабный эксперимент с экстремальными объёмами данных и арифметических действий.

Согласно сложившейся методологии математического моделирования [8–13] её технологическая цепочка, при всём многообразии, включает относительно небольшое количество стадий, качественное содержание которых, однако, существенно меняется с развитием суперкомпьютерных поколений, вычислительных алгоритмов и инструментов. Мы будем рассматривать некую универсальную ЭВМ, или сеть машин, с программным обеспечением, интегрированным в том смысле, что оно позволяет решать самый широкий круг задач.

Изучение любого объекта начинается с формирования его модели, которая может быть представлена совокупностью дифференциальных и/или интегральных, а также дискретных уравнений и соотношений, с дополнительными ограни-

чениями и условиями на оптимизацию, с большими объёмами реально измеряемых данных. Последние могут быть приближёнными и даже противоречивыми, никак не подпадающими под строгие математические понятия существования, единственности и корректности решения. Такие сложные системы приходится анализировать гидрометеорологическим службам при прогнозировании погоды, изменений климата или предупреждении природных катастроф, при использовании данных всемирной сети наземных и космических наблюдений. Другая иллюстрация — многофазные процессы в нефтегазовой отрасли с применением химических, электрофизических и других современных технологий добычи. Одна из экстремальных проблем — последствия термоядерного взрыва, где происходит наложение процессов гидрогазодинамики, физики твёрдого тела, плазмы и т.д.

Естественно, что стратегия и тактика моделирования должны обеспечивать максимум эффекта при минимизации вычислительных ресурсов, отнюдь не дешёвых. Все науки быстро развиваются, а с ними растут, количественно и качественно, всевозможные модели, возникают их иерархии — от простых и экономичных до более сложных и точных. Взаимодействие человека с ЭВМ требует комфортных интерфейсов со входными языками, определяющими уровень общения с пользователями самых разных специальностей. Данная стадия предназначена для формулирования задания для ЭВМ той модели (или их последовательностей из возможного набора), которая необходима в соответствии с поставленной задачей. Очевидно, что такой уровень функциональности программной подсистемы и её пользователя может быть достигнут только в результате машинного обучения путём накопления опыта эксплуатации и анализа получаемых результатов. Отметим, что “умные” реализации данной стадии могут включать качественный анализ математических свойств моделей, что, несомненно, повышает ценность такого продукта.

После конкретизации математической задачи на непрерывном уровне требуется её дискретизация, то есть построение сетки. Данный этап весьма важен с точки зрения эффективности моделирования и представляет собой трудоёмкую алгоритмическую проблему в многомерных задачах с реальными данными, включающими сложные геометрические конфигурации расчётных областей с кусочно-гладкими неодносвязанными разномасштабными границами (в том числе движущимися) и контрастными материальными свойствами сред. В подобных случаях зачастую приходится учитывать сингулярности решений и строить адаптивные неструктурированные сетки, вопрос об оптимизации которых пока остаётся открытым.

Необходимо отметить, что наиболее эффективные численные методы решения больших задач

связаны с декомпозицией областей, являющейся основным орудием распараллеливания алгоритмов, а также с применением многосеточных подходов, обеспечивающих асимптотическую оптимальность по порядку решений (при характерных шагах сетки $h \rightarrow 0$ общее число арифметических действий пропорционально числу неизвестных [14]). Это ставит перед “сеточными генераторами” новые интересные задачи, связанные с построением сложных структур данных и с интеллектуальными операциями на графах [15].

Существует достаточно много подходов к построению сеточных аппроксимаций различного порядка точности: методы конечных разностей, конечных объёмов, конечных элементов, разрывных алгоритмов Галёркина и т.д. Создание многофункционального программного обеспечения с этой целью для различных типов сеток и видов операторов — актуальная и востребованная задача (проект соответствующей подсистемы CHEBYSHEV описан в [16]). Отметим, что, хотя проблема автоматизации построения алгоритмов для сеточных аппроксимаций достаточно стара [17], широкого распространения этот подход пока не получил.

Большинство практических задач — нелинейные и нестационарные, но после применения к ним квазилинеаризации и неявных по времени аппроксимаций неизбежно приходится решать системы линейных алгебраических уравнений (СЛАУ), как правило, с большими разреженными матрицами, симметричными и несимметричными. Основной подход здесь — итерационные предобусловленные методы в пространствах Крылова [18] (соответствующая подсистема называется библиотекой KRYLOV).

Ключевую роль в машинном обучении играют оптимизационные методы решения обратных задач, поскольку именно на их основе можно отыскать и исследовать наилучшие сценарии процессов и явлений [19]. Типичная методология состоит в планировании серии вычислительных экспериментов, в которых на основе предыдущего опыта описывается минимизируемый целевой функционал, а после анализа полученных результатов формируется следующий сеанс машинного обучения. Такое взаимодействие человека и ЭВМ представляется безальтернативным во многих многокритериальных практических поисковых задачах, когда детерминированный алгоритм их решения формализовать в принципе невозможно. Отметим важный момент: варьируемые при этом параметры могут относиться не только к изучаемой модели, но и к самому вычислительному процессу, поскольку в патовых ситуациях он может или совсем не сходиться, или длиться неприемлемо долго.

Таким образом, в итерационный цикл вовлекаются все рассмотренные стадии моделирования. К ним следует добавить методы и технологии

постобработки, визуализации и анализа расчётных результатов, на основе которых функционируют средства принятия решений [20].

МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КАК ЭКОСИСТЕМА

Прикладное программное обеспечение, как и системное, уже более полувека бурно развивается одновременно с вычислительной техникой, для которой всем на удивление до сих пор почти выполняется закон Мура (увеличение быстродействия в 1000 раз за 11 лет). Однако надо признать, что за последние десятилетия темпы роста производительности труда программиста стали существенно отставать от темпов производительности компьютеров, то есть в определённом смысле можно говорить о кризисе программирования. Чтобы исправить этот перекос, начинает активно использоваться искусственный интеллект.

К настоящему времени в мире накоплено огромное количество общедоступного (Open Source) и коммерческого “софта” в виде библиотек, специальных инструментов и проблемно-ориентированных пакетов прикладных программ (ППП) [10], которые представляют высокую интеллектуальную ценность. Здесь можно назвать такие высокопрофессиональные разработки, как библиотеки PETSc, HYPRE, PARDISO, пакеты программ FENIX, DEAL II, специализированные системы PARVIEW, MAPLE и многие другие. Отдельный мировой рынок составляют системы автоматизаций проектирования (САПР) [21], причём в последние годы наметилась конвергенция этих систем с классическими ППП. В то же время наметился переход от специализированных к интегрированным программным окружениям. Примерами крупных проектов в данной области являются DUNE, OPEN FOAM, INMOST, а также Базовая система моделирования (БСМ) [12, 22–24]; все они в первую очередь методоориентированные.

Функциональное наполнение БСМ — это набор автономных подсистем, каждая из которых отвечает за соответствующую технологическую стадию моделирования и связана с другими посредством согласованных структур данных. Благодаря подсистеме формирования моделей создаются функциональная и геометрическая структуры данных (ФСД и ГСД) — исходные для этапа генерации сетки. На основе получаемой сеточной структуры данных (ССД) в результате выполнения этапа аппроксимации строятся алгебраические информационные массивы в общепринятых мировых форматах (АСД), обеспечивающих высокопроизводительное решение широкого класса систем линейных алгебраических уравнений.

Такая архитектура позволяет независимым группам профессиональных разработчиков реализовывать и развивать различные подсистемы. При этом

легко достигается гибкая расширяемость состава модулей и алгоритмов для каждого вычислительного этапа, в том числе с эффективным переиспользованием внешних программных продуктов и с адаптацией на новых компьютерных платформах. Итоговое интегрированное вычислительное окружение (ИВО) представляет собой самоподдерживающуюся экосистему с длительным жизненным циклом, ориентированную на успешную эксплуатацию широким кругом пользователей. Предусмотренный переизбыточный набор моделей и алгоритмов призван поддерживать методологии машинного обучения как самой экосистемы, так и её пользователей.

Обеспечение богатой функциональности и эффективного использования интегрированного вычислительного окружения требует создания разнообразного системного наполнения. С математической точки зрения это означает в первую очередь разработку средств автоматизации построения алгоритмов, включая инструменты многоверсионных конфигураций вычислительных модулей. Что касается языков программирования, отметим популярность сочетания стилей объектно-ориентированных подходов C++ и богатых изобразительных возможностей языка интерпретирующего типа Python, который включает такие важные интеллектуальные компоненты, как компьютеризация сложных аналитических выкладок. Это открывает путь к активному использованию аппроксимационных методов повышенной точности, которые перспективны в теоретическом плане в силу значительного сокращения требуемой памяти и энергозатратных коммуникационных расходов, но имеют пока недостаточно широкое распространение из-за трудоёмкого программирования.

В мире также ведутся разработки специализированных естественных языков для математиков-вычислителей (существует даже лозунг “программирование без программирования”). Хотя в публикациях в Интернете упоминаются такие заманчивые проекты по “фабрикам языков”, как SIDL (Scientific Interface Definition Language) и DSL (Domain Specific Languages) [25], где насущная проблема трактуется как переход от “палеоинформатики к неоинформатике”, в ближайшее время прорывные перспективы здесь не просматриваются, технологии формирования алгоритмических библиотек остаются неизменными уже несколько десятилетий. В целом языковой контент является одним из ключевых для уровня машинного обучения, а текущую ситуацию можно оценить как мягкий выход из кризиса программирования.

Несомненно, одно из главных качеств прикладной программы — её производительность, определяемая в первую очередь качеством распараллеливания алгоритмов. Здесь огромные возможности предоставляют ПЛИС (программируемые логические интегральные схемы), которые позволяют

спроектировать и собрать специализированный компьютер максимального быстродействия для заданного алгоритма. Однако у таких компьютеров есть один недостаток — коммерческая неконкурентоспособность в сравнении с типовыми суперкомпьютерами кластерного типа, вычислительные узлы которых соединены шинами и обмениваются между собой данными с помощью очень простой библиотеки MPI, а расчёты на каждом из них осуществляют многоядерные процессоры с иерархической общей памятью (её различные уровни имеют разные объёмы и скорости обменов) под управлением программной системы типа OpenMP. Существуют и быстрые графические ускорители (GPGPU), но их связь с общей памятью медленная, что существенно снижает их эффективность.

Ситуацию с распараллеливанием вычислений можно считать парадоксальной ввиду отсутствия соответствующей массовой, или штатной, системы автоматизации программирования, за исключением нескольких малораспространённых языков или подсистем (например, SHAPL и DVM — Distributed Virtual Machine [26]). Фактически распараллеливание алгоритмов на многопроцессорных вычислительных системах — это сугубо ручная работа с экспериментальными подбираемыми способами ускорения вычислений, которые измеряются двумя простыми параметрами:

$$S_p = T_1 / T_p, E_p = S_p / p$$

— коэффициентами ускорения и эффективности, где T_p есть время решения задачи на p процессорах. В качестве перспективных можно назвать исследования по оптимизации распараллеливания на основе концепции D-детерминанта [27]. С точки зрения машинного обучения можно сформулировать проблему о поиске самим суперкомпьютером наилучшего способа распараллеливания на основе серии расчётов для конкретного класса задач.

Очень важное значение для повышения производительности труда программиста имеет скорость прохождения таких неизбежных рутинных процедур, как отладка, тестирование, верификация и валидация реализуемого кода. Возникающие здесь проблемы неизбежно усугубляются при создании больших программных комплексов, которые как раз и подразумеваются в концепции интегрированного вычислительного окружения; это обусловлено в первую очередь драматическим усложнением информационных связей, а также внутренних межмодульных и пользовательских интерфейсов. В больших профессиональных командах разработчиков операционных систем и компиляторов эти проблемы давно решены с помощью известных компонентных технологий COM/DCOM и CCA (Common Component Architecture) [28], но в прикладном программировании они пока находятся в ожидании перехода на индустриальное мышление.

Один из главных компонентов ИВО — репозиторий (хранилище), обеспечивающий целостность разработки и её связи с внешним миром, поддерживающий свойства многоязыковости и кроссплатформенности, а также взаимодействия с разработчиками и пользователями. Широкое распространение в компьютерном сообществе получила система такого типа GITHUB с большим разнообразием сервисов.

О НЕКОТОРЫХ ПРИНЦИПАХ МАШИННОГО ОБУЧЕНИЯ И ПРИНЯТИЯ РЕШЕНИЙ

Всякое обучение заключается в приобретении знаний и навыков для какой-то сферы деятельности, в том числе для принятия решений на основе анализа содержательной информации. В отношении математического моделирования мы можем выделить три категории участников: неодушевлённый объект — компьютер, состоящий из технического оборудования и программного обеспечения, и два типа субъектов — разработчики инструментов моделирования и конечные пользователи — специалисты в конкретных предметных областях, осуществляющие суперкомпьютерные инновации. Понятно, что персонал может быть представлен людьми разной квалификации, между которыми возникают отношения учитель—ученик.

Знания являются активными, если они приводят к каким-то действиям и результатам [29]. Их накопление и систематизация для определённой предметной области предполагают разработку соответствующей базы активных знаний (БАЗ), содержащей всевозможную информацию об объектах, их спецификациях, взаимосвязях и о возможных действиях над ними. С расширением и углублением добываемой информации появление БАЗ неизбежно, поскольку объём информации становится непосильным для усвоения человеком. Структура БАЗ окончательно ещё не сложилась, пример проекта такой разработки для задач вычислительной алгебры приведён в работе [30]. Прототипом такой разработки можно считать систему ALGOWIKI [31], созданную под руководством Дж. Донгарры и В.В. Воеводина.

Базы знаний должны содержать всю необходимую информацию по соответствующей теме. Если мы говорим о математическом моделировании для определённого класса проблем, то сюда входят описания моделей, вычислительных методов и технологий, примеры задач и результаты их решения (для чего должны создаваться специальные архивы), рекомендации по всевозможным применениям, а также литературные источники и доступные программные материалы с документацией. Другими словами, должна быть разработана онтология, позволяющая проводить разбор текстов, статистический анализ данных и другие интеллектуальные действия. База знаний предполагает систему сбора и усвоения огромных объёмов информации, например косми-

ческой, метеорологической и т.д., которую необходимо интегрировать с оперативными расчётами.

Наиболее содержательными здесь являются проблемы, связанные с оптимизационными методами решения обратных задач, позволяющие достичь наибольшего практического эффекта, например, при идентификации параметров модели по результатам натурных измерений (иллюстрация – нефтегазовое месторождение) или оптимизации эксплуатационных режимов технической установки (самолёт, корабль, заводской цех и т.п.) [19]. При этом постановка проблемы формулируется как поиск минимума некоторого целевого функционала

$$\Phi_0(\vec{u}(\vec{x}, t, \vec{p}_{opt})) = \min_{\vec{p}} \Phi_0(\vec{u}(\vec{x}, t, \vec{p}))$$

для решения какой-то прямой начально-краевой задачи, который подчиняется дополнительным линейным и/или нелинейным ограничениями:

$$\begin{aligned} p_k^{min} &\leq p_k \leq p_k^{max}, k = 1, \dots, m_1, \\ \Phi_l(\vec{u}(\vec{x}, t, \vec{p})) &\leq \delta_l, l = 1, \dots, m_2, \\ \vec{p} &= \{p_k\} \in \mathcal{R}^m, m = m_1 + m_2. \end{aligned}$$

Здесь $\vec{p}$ – m -мерный вектор оптимизируемых параметров, а t , $\vec{x}$ суть временная и пространственная координаты. Исходную прямую задачу, или уравнение состояния оптимизируемой сложной системы, формально можно представить в следующем виде:

$$\begin{aligned} L\vec{u} &= \vec{f}(\vec{x}, t), \vec{x} \in \bar{\Omega} = \Omega \cup \Gamma, 0 < t \leq T < \infty \\ l\vec{u} &= \vec{g}(\vec{x}, t), \vec{x} \in \Gamma, \vec{u}(\vec{x}, 0) = \vec{u}^0(\vec{x}), \\ \bar{\Omega} &= \Omega \cup \bar{\Omega}_j, \Gamma = \Gamma^e \cup \Gamma^i, \\ \Gamma^i &= \cup \Gamma_{j,k}^i = \cup (\bar{\Omega}_j \cap \bar{\Omega}_k), \end{aligned}$$

где L есть в общем случае матричного типа оператор (в дифференциальной и/или интегральной форме), l – оператор граничных условий, а $\bar{\Omega} \times [0, T]$ – расчётная область, которая зачастую состоит из подобластей $\bar{\Omega}_k$ с различными контрастными материальными свойствами, а также с внутренними и внешними границами Γ^e , Γ^i , в том числе имеющими разномасштабные детали и кусочно-гладкие многосвязные поверхностные сегменты. В общем случае речь идёт о междисциплинарных неклассических формулировках, когда даже вопросы существования, единственности и корректности остаются открытыми. Что касается проблемы условной минимизации функционала, то она может быть локального типа или глобального. В последнем случае требуется найти все имеющиеся минимумы.

Приведу следующий характерный пример машинного обучения. Пусть нам требуется в течение длительного времени (год или более) проводить оперативное оптимальное управление некоторой сложной системой, зависящей от 10 параметров, путём решения обратных и прямых задач, причём каждая из них требует длительных расчётов (часы или десятки часов). Отметим, что если по каждому параметру перебирать 10 его возможных значений, то общее количество вариантов составит 10^{10} (проклятие размерности)! В этом случае машинное обучение компьютера можно осуществить следующим образом. Сначала в течение нескольких, скажем, недель с помощью классических методов оптимизации решают сотни задач, результаты которых запоминают и как-то статистически обрабатывают (например, с помощью популярного вида обобщённой линейной регрессии – кригинга), формируя соответствующие аппроксимации в сеточном пространстве параметров. Затем на основе накопленных данных начинается оперативная работа машинно-программного комплекса, который быстро находит приближение к искомому состоянию системы, а далее с участием человека последовательно происходит необходимое уточнение параметров.

В работах по искусственному интеллекту сложилось понятие “базовые модели” (Foundation Models, или LxM – Large x Model), определяемые как средства глубокого обучения путём “тренинга” на огромном количестве тестовых примеров и задач [32] и обширной литературе, главным образом ещё неопубликованной, размещаемой на общедоступных интернет-ресурсах arXiv. Кроме того, в обиход вошёл термин “суррогатная оптимизация”, который подразумевает, что поиск наилучшего решения производится не для реального объекта или процесса, а для его модели, возможно, грубой. В целом решение сложных обратных задач с многообразным поведением минимизируемых функционалов требует применения иерархии моделей, выбор которых требует высокого искусства. Фактически в таких случаях работают метаалгоритмы взаимодействия компьютера с пользователем, имеющим большой опыт решения конкретного класса задач.

Развитие машинного обучения в вычислительной математике связано в первую очередь с выбором оптимального или “хорошего” алгоритма для каждой стадии технологической цепочки моделирования: генерация сетки, аппроксимация и дискретизация исходной задачи, решение получаемых систем линейных и нелинейных уравнений и т.д. В результате поиск наилучшего общего вычислительного процесса сводится к разноуровневым повторяющимся циклам с многократными численными экспериментами, анализом промежуточных результатов и выработкой итоговой стратегии практических расчётов. Зачастую здесь необходимо идти на компромисс типа “лучшее – враг хорошего”, поскольку сама по себе оптимизация может

оказаться дороже, чем уже известный достаточно эффективный подход. Проникновение машинного обучения в вычислительную математику происходит достаточно активно, и здесь можно отметить работы по методам конечных элементов и по итерационным алгоритмам [33, 34] и цитируемую в них литературу. Относительно решения различных прикладных задач сложилась специальная нейрометодология PINN (Physics Informed Neural Networks), ориентированная на решение дифференциальных уравнений, описывающих законы сохранения, в том числе на непрерывном уровне, без перехода к сеточным дискретизациям [35].

Нейросетевые подходы очень быстро эволюционируют. В 2017 г. в работе [36] была предложена архитектура нейросети под названием трансформер, базирующаяся на механизме внимания. Её суть в определении глобальных зависимостей между входными и выходными данными путём предварительного обучения на примерах. Построенные на этих принципах программные реализации уже значительно усилили инструментарий для обработки изображений, анализа текстов, машинного перевода и др. В качестве убедительного примера эффективного использования машинного обучения для решения действительно очень сложной физико-математической проблемы рассмотрим задачу определения межатомных потенциалов, которые аппроксимируют модели квантово-механических взаимодействий, описываемых изначально экстремально ресурсоёмкой теорией Кана–Шема. Достаточно общий подход к решению данной задачи основан на нейросетевых потенциалах, современная версия которых представлена с помощью eMTP — потенциала тензора электронного момента, лежащего в основе класса машинно-обучаемых моделей межатомных взаимодействий с требуемой высокой точностью [37].

Другое направление исследований — это нейрооператоры (NO) [38], являющиеся развитием “физически-информированных” нейронных сетей типа PINN. В отличие от последних, NO ориентированы на аппроксимации обратных операторов, характеризующих связи между функциональными пространствами, что позволяет переходить на решение целых классов задач. Следует заметить, что новые нейросетевые и нейрооператорные технологии — это в определённом смысле хорошо забытые старые подходы классических методов математической физики середины прошлого века, которые основаны на непрерывных базисных функциях, эффективно применяемых для небольших порядков, но резко усложняющихся в случае повышения точности расчётов.

Аналогичные методологии применимы и в других науках (химия, биология и т.д.) или на производствах: модель, компьютерный эксперимент, анализ результатов, новые знания, принятие решений. Однако это не самоцель, далее должны следовать

какие-то решения по оптимизации человеческой деятельности и инновациям — повышению производительности труда, количеству и качеству выпускаемой продукции, достижению социального и/или гуманитарного эффекта и т.д. Фактически речь идёт о кардинальном изменении способов деятельности и о появлении новых массовых профессий. Несомненно, машинное обучение должно значительно изменить педагогические подходы и саму систему образования — от дошкольного до высшего, и эти актуальные вопросы требуют своих исследований.

* * *

В современных условиях машинное обучение стало неотъемлемым атрибутом получения новых знаний. Одна из проблем в области использования искусственного интеллекта — принятие человеком решений исходя из анализа получаемых данных. Оптимизация этого вида деятельности — актуальная, но отнюдь не новая проблема. В качестве иллюстрации можно привести многолетние работы Г.С. Альтшуллера и его последователей [39] по созданию ТРИЗ — теории решения изобретательских задач. Современные подходы в рамках этого направления основаны на построении онтологий тех или иных предметных областей, которые позволяют создавать когнитивный инструментарий для принятия решений [40].

Яркий пример такого рода современного проекта содержит отчёт [41] Центра исследований базовых моделей (CRFM, Стенфордский университет), в котором описана концепция экосистемы, ориентированной на эффективные интеллектуальные инновации в самых широких прикладных сферах: здравоохранении и биомедицине, юриспруденции и образовании, экономике и окружающей среде и т.д. В определённом смысле этот подход коррелирует с методологией интегрированного вычислительного окружения с базовой системой моделирования, которое представляет собой наукоёмкое функциональное наполнение для высокопроизводительного решения междисциплинарных прямых и обратных задач математического моделирования, различные аспекты которых изложены в работах [12, 13, 18–20, 42].

Универсальный характер машинного обучения и оптимизация на его основе планирования человеческой деятельности обуславливают его глобальную экспансию, что в совокупности с роботизацией неизбежно побуждает к философскому осмыслению особенностей и вызовов цифровой трансформации общества. Неслучайно появляются публикации о моральных аспектах внедрения искусственного интеллекта [43]. Очевидно, что инновации непосредственно влияют и на обеспечение научно-технологического суверенитета, и на производственные задачи государственного масштаба, и на обеспечение национальной безопасности, и на устойчивое развитие цивилизации.

ИСТОЧНИК ФИНАНСИРОВАНИЯ

Работа выполнена при финансовой поддержке Минобрнауки России (код проекта FSUN-2024-0003).

ЛИТЕРАТУРА

1. *LeCun Y., Bengio Y., Hinton G.* Deep learning // *Nature*. 2015, vol. 521, pp. 436–444.
2. *Weinan E.* Machine learning and computational mathematics // *Commun. Comput. Phys.* 2020, vol. 28, pp. 1639–1670.
3. *Dongarra J., Grigori L., Higham N.J.* Numerical algorithms for high performance computational science // *Phil. Trans. R. Soc.* 2020, vol. 378, iss. 2166.
4. *Xu Y., Zeng T.* Sparse Deep Neural Network for Nonlinear Partial Differential // *Equations. Numer. Math. Theor. Meth. Appl.* 2022, vol. 16, no. 1, pp. 58–78.
5. *Ильин В.П.* Математическое моделирование: философия науки // Сб. научно-поп. статей “Математика, механика и информатика”. М., 2017. С. 8–16.
Il'in V.P. Mathematical modeling: The philosophy of science // *Collection of Popular Science Articles “Mathematics, Mechanics, and Informatics”*. Moscow, 2017. Pp. 8–16. (In Russ.)
6. *Il'in V.P.* Artificial intelligence problems in mathematical modeling // *Voevodin V., Sobolev S. (eds.) Russian supercomputing days 2019. CCIS – Springer*. 2019, vol. 1129, pp. 505–516.
7. *Forrester A., Sobester A., Keane A.* Engineering Design via Surrogate Modeling: A Practical Guide. Wiley, New York, 2008.
8. *Яненко Н.Н., Коновалов А.Н.* Некоторые вопросы теории модульного анализа и параллельного программирования для задач математической физики и механики сплошной среды // *Современные проблемы математической физики и вычислительной математики*. М.: Наука, 1982. С. 200–217.
Yanenko N.N., Kononov A.N. Some questions of the theory of modular analysis and parallel programming for problems of mathematical physics and continuum mechanics // *Modern Problems of Mathematical Physics and Computational Mathematics*. Moscow: Nauka, 1982. Pp. 200–217. (In Russ.)
9. *Яненко Н.Н., Рычков А.Д.* Актуальные проблемы прикладной математики и математического моделирования. Новосибирск: Наука, 1982.
Yanenko N.N., Rychkov A.D. Topical Problems of Applied Mathematics and Mathematical Modeling. Novosibirsk: Nauka, 1982. (In Russ.)
10. *Ершов А.П., Ильин В.П.* Пакеты программ – технология решения прикладных задач. Новосибирск: ВЦ СО АН СССР. Препринт № 121, 1978.
Ershov A.P., Il'in V.P. Software packages: Technology for solving applied problems. Preprint no. 121. Computer Center of the Siberian Branch of the USSR Acad. Sci. Novosibirsk, 1978.
11. *Самарский А.А., Михайлов А.П.* Математическое моделирование. М.: Физматлит, 2002.
Samarskii A.A., Mikhailov A.P. Mathematical Modeling/ (Moscow: Fizmatgiz, 2002. (In Russ.)
12. *Ильин В.П.* Как реорганизовать вычислительные науки и технологии // *Вестник РАН*. 2019. № 2. С. 232–242.
Il'in V.P. How to reorganize computer science and technology // *Vestn. Ross. Akad. Nauk*. 2019, no. 2, pp. 232–242. (In Russ.)
13. *Ильин В.П.* Математическое моделирование. Ч. 1. Непрерывные и дискретные модели. Новосибирск: Изд-во СО РАН, 2017.
Il'in V.P. Mathematical Modeling. Part 1. Continuous and Discrete Models. Novosibirsk: SB RAS, 2017. (In Russ.)
14. *Il'in V.P.* Iterative Preconditioned Methods in Krylov Spaces: Trends of the XXI Century // *Computational Mathematics and Mathematical Physics*. 2021, vol. 61, no. 11, pp. 1750–1775.
15. *Il'in V.P.* Integrated Computational Environment for Grid Generation Parallel Technologies / L. Sokolinsky, M. Zymbler (eds.). CCIS. 2020, vol. 1263, pp. 58–68.
16. *Бутюгин Д.С., Ильин В.П.* CHEBYSHEV: принципы автоматизации построения алгоритмов в интегрированной среде для сеточных аппроксимаций начально-краевых задач // *Труды Международной конференции ПАВТ'2014*. Челябинск: изд-во ЮУрГУ, 2014. С. 42–50.
Butyugin D.S., Il'in V.P. CHEBYSHEV: Principles of automating the construction of algorithms in an integrated environment for grid approximations of initial-boundary value problems // *Proceedings of the International Conference PAVT'2014*. Chelyabinsk, 2014). Pp. 42–50. (In Russ.)
17. *Валиуллин А.Н., Ганжа В.Г., Ильин В.П., Яненко Н.Н.* Задача автоматического построения и исследования на ЭВМ разностных схем в аналитическом виде // *Доклады АН СССР*. 1984. Т. 275. № 3. С. 528–532.
Valiullin A.N., Ganzha V.G., Il'in V.P., Shapeev V.P., Yanenko N.N. Problem of automatic generation and investigation of difference schemes in symbolic form on a computer // *DAN SSSR*. 1984, vol. 275, no. 3, p. 528–532. (In Russ.)
18. *Il'in V.P.* Parallel intelligent computing in algebraic problems / Sokolinsky. *Parallel Computational Technologies. Communications in Computer and Information Science*. 2021, vol. 1437, pp. 108–117.
19. *Il'in V.P.* The integrated computational environment for optimization of complex systems // *Proceedings*

- of the 15th International Asian School-Seminar “Optimization Problems of Complex Systems” 2019. Pp. 65–67. <https://doi.org/10.1109/opcs.2019.888015>
20. *Il'in V.P., Skopin I.N.* About performance and intellectuality of supercomputer modeling // *Programming and Computer Software*. 2016, vol. 42, iss. 1, pp. 5–16.
 21. *Ушаков Д.М.* Введение в математические основы САПР. Новосибирск: ЗАО “ЛЕДАС”, 2008.
Ushakov D.M. Introduction to mathematical foundations of CAD. Novosibirsk, LEDAS Publ., 2008. (In Russ.)
 22. *Bastian P., Blatt M., Dedner A. et al.* The Dune Framework: Basic Concepts and Recent Developments, Computers and Mathematics with Applications, 2020. DOI.org/10.1016/j.camwa.2020.06.007
 23. OpenFOAM. <https://www.openfoam.com/>
 24. INMOST: A Toolkit for Distributed Mathematical Modeling. <https://www.inmost.org>
 25. *Kleppe A.* Software language engineering: Creating domain-specific language using metamodels. N.Y.: Addison-Wesley, 2008.
 26. DVM Systems. <http://www.keldush.ru/dvm>
 27. *Aleeva V.* Designing Parallel Programs on the Base of the Conception of Determinant // *Supercomputing. RuSCDays 2018 (Communications in Computer and Information Science)*. 2019, vol. 965, pp. 565–577.
 28. *Allan B., Armstrong R., Wolfe A. et al.* The CCA Core specification in a Distributed Memory // *SPMD Framework Concurrent Practice and Expedience*. 2002, vol. 14, pp. 323–345.
 29. *Malyshkin V.E.* Active knowledge, LuNA and literacy for oncoming centuries // *LNCS*. 2015, vol. 9465, pp. 292–303.
 30. *Ильин В.П.* Концепция и архитектура базы знаний систем линейных алгебраических уравнений // *Материалы IX конференции “Знания–Онтологии–Теории”* 2023, 2–6 октября 2023 г. Новосибирск: ИСИ СОРАН, 2023. С. 143–154.
Il'in V.P. The conception and architecture of the knowledge base of the systems of linear algebraic equations // *Proced. IX Conference “Knowledge-Ontology-Theory”*. Novosibirsk: ISI SBRAS, 2023. Pp. 143–154. (In Russ.)
 31. *Antonov A., Dongarra J., Voevodin V.* AlgoWiki Project as an Extension of the Top500 Methodology // *Supercomput. Frontiers and Innovations*. 2018, vol. 5, no. 1, pp. 4–10.
 32. *Yang S., Nachum O., Du Yi. et al.* Foundation Models for Decision Making: Problems, Methods, and Opportunities. ArXiv, abs/2303.04129. [cs. AI] 7 mar 2023.
 33. *Grossmann T.G., Komorowska U.J., Latz J., Schönlieb C.* Can physics informed neural networks beat the finite element method? // *IMA Journal of Applied Mathematics*, 2024, p. hxae011.
 34. *Rudikov A., Fanaskov V., Muravleva E. et al.* Neural operators meet conjugate gradients: The FCG-NO method for efficient PDE solving // *Proceedings of the 41st International Conference on Machine Learning, ICML 2024*, 2024.
 35. *Goswami S., Bora A., Yu Y., Karniadakis G.E.* Physics-informed deep neural operator networks // *Machine Learning in Modeling and Simulation: Methods and Applications*, Springer, 2023, pp. 219–254.
 36. *Vaswani A. et al.* Attention Is All You Need // *Proceed. of 31st Conference on Neural Information Processing Systems. NIPS*. 2017, pp. 1–11.
 37. *Srinivasan P., Demuriya O., Grabowski B., Shapeev A.V.* Electronic moment tensor potentials include both electronic and vibrational degrees of freedom // *Computational Materials*. 2024, vol. 10, iss. 1, id. 41.
 38. *Fanaskov V.S., Oseledets I.V.* Spectral neural operators // *Doklady Mathematics*. 2023, vol. 108, pp. S226–S232.
 39. *Альтшуллер Г.С.* Найти идею. Введение в теорию решения изобретательских задач. Новосибирск: Наука, 1986.
Altshuller G.S. To find idea. The introduction into theory of solving the invent tasks. Novosibirsk: Nauka, 1986. (In Russ.)
 40. *Zagorulko Yu., Zagorulko G.* Architecture of extensible tools for development of intelligent decision support systems // *New Trends in Software Methodologies, Tools and Techniques. Proc. of the 10th SoMeT-11. Hamido Fujita (Eds.)*. Amsterdam: IOS Press, 2011. Pp. 457–466.
 41. *Bommasani R., Hudsa D.A., Adeli E. et al.* On the opportunities and Risks of Foundation Models. CRFM, Stanford, 2021. doi: 10.48550/arXiv.2108.07258.
 42. *Ильин В.П.* Программирование ближайшего будущего: концепция и прагматика // *Вестник РАН*. 2023. № 2. С. 150–161. DOI: 10.31857/S086958732302007X
Ilyin V.P. Programming for the Near Future: Concepts and Pragmatic Considerations // *Her. Russ. Acad. Sci.* 2023, no. 1, pp. 92–102. <https://doi.org/10.1134/S1019331623010112> (In Russ.)
 43. *Luccioni A., Bengio Yo.* On the Morality of Artificial Intelligence. arXiv:1912.11945 [cs.CY] /doi.org/10.48550/arXiv.1912.11945