МАШИННОЕ ОБУЧЕНИЕ: ОПЫТ ПРЕПОДАВАНИЯ И ВНЕДРЕНИЯ*

М. М. Липкович
lipkovich.mikhail@gmail.com

27 апреля 2023 г.

Доклад посвящен описанию опыта преподавания и внедрения машинного обучения на математико-механическом факультете СПбГУ. Он состоит из двух частей: в первой части представлена программа курса по машинному обучению, который читается четверокурсникам отделений «Прикладная математика и информатика» и «Математика и компьютерные науки», во второй части рассматриваются практические проекты, выполненные совместно со студентами и исследователями из биологического факультета СПбГУ и Института Мозга Человека им. Н.П. Бехтеревой РАН.

1°. Курс по машинному обучению. Курс по машинному обучению был введен в 2019 году для студентов математико-механического факультета СПбГУ и с тех пор претерпел ряд изменений. Он позиционируется как курс, по завершении которого студент, добросовестно выполняющий все задания, будет готов устроиться на работу в коммерческую компанию на должность, связанную с применением методов машинного обучения. В связи с этим курс требует регулярной корректировки с учетом изменений в области. При этом акцент сделан не на последних достижениях в академической среде, а на подходах, которые активно используются в коммерческих компаниях. С одной стороны, это позволяет не так часто обновлять курс, так как не все академические разработки находят широкое применение в промышленности, но, с другой стороны, это усложняет возможность разработки курсовых и дипломных работ для студентов, поскольку им приходится осваивать новые темы, с которыми они еще не сталкивались.

Таким образом, за период с начала чтения курса произошли значительные изменения, связанные с усилением акцента на глубокое обучение по сравнению

*Семинар по оптимизации, машинному обучению и искусственному интеллекту «O&ML» `http://oml.cmlaboratory.com/`

с классическим машинным обучением, переходом от рекуррентных нейронных сетей к трансформерам, от сверточных сетей к визуальным трансформерам, а также больший фокус на интерпретируемости моделей. При этом более ранние подходы не исключены из курса, а остаются для лучшего понимания идей и их эволюции, а также для решения задач, где они по-прежнему могут быть более эффективными.

Кроме того, за это время значительно улучшились доступные вычислительные ресурсы для обучения моделей, в том числе моделей с большим числом параметров. Такие сервисы как Google Colab[1] и Kaggle Kernels[2] предоставляют возможность бесплатно использовать вычислительные ресурсы, включая графические процессоры (GPU). В последние годы увеличились объемы предоставляемых ресурсов, улучшилась стабильность сервисов и снизилось время ожидания. Это позволило вводить в курс задачи, связанные с обучением более глубоких нейронных сетей.

2°. В настоящее время курс состоит из следующих разделов:

1) Введение в машинное обучение. В рамках этого раздела рассматриваются типы задач машинного обучения, вводятся основные понятия, а также на примере модели линейной регрессии рассматриваются вариации алгоритма градиентного спуска с доказательством сходимости для случая выпуклых непрерывно дифференцируемых функций и виды метрик для задачи регрессии.

   Далее обсуждается проблема компромисса между смещением и дисперсией (bias-variance tradeoff), с выводом разложения ошибки обобщения на смещение и дисперсию, а также методы борьбы с переобучением с помощью регуляризации.

   Заканчивается раздел обсуждением различных стратегий обучения: разбиение на тренировочную и тестовую выборки, кросс-валидация, вложенная кросс-валидация, а также подбор гиперпараметров моделей.

2) Работа с признаками. Поскольку значительная часть работы в области машинного обучения связана с подготовкой данных, на работу с признаками выделен отдельный раздел. Здесь обсуждаются такие проблемы, как пропуски в данных, нормализация, преобразование категориальных признаков, а также базовые методы кодирования текстов.

3) Модели классификации. По мнению автора курса, модели классификации применяются на практике чаще, чем модели регрессии, поэтому они

[1] https://colab.google/

[2] https://www.kaggle.com/code/dansbecker/running-kaggle-kernels-with-a-gpu

вынесены в отдельный раздел, в то время как регрессионные модели упоминаются преимущественно в контексте моделей, которые могут решать как классификационные, так и регрессионные задачи. Рассматриваются такие модели классификации, как логистическая регрессия с различными видами регуляризации, метод ближайших соседей (kNN) с обсуждением подбора параметра k, а также метод опорных векторов (SVM), включая обсуждение ядерного трюка и мягкого зазора (soft margin).

Обсуждаются метрики полноты, точности, F-меры, ROC-кривые и площадь под кривой (ROC AUC, PR AUC).

4) Деревяные модели. Этот раздел начинается с обсуждения классических деревьев принятия решений и переходит к построению ансамблей деревьев: случайный лес и градиентные бустинги. Эти модели являются победителями множества соревнований и, как правило, являются первыми моделями, с которых начинают решение задач на практике, прежде чем переходить к более сложным или специализированным моделям.

5) Обучение без учителя. В рамках обучения без учителя рассматривается задача понижения размерности на примере метода главных компонент (PCA) и модели кластеризации: метод k-средних (kMeans), DBScan, иерархическая кластеризация.

   Задача детектирования аномалий отдельно не рассматривается, поскольку она узкоспециализирована, но упоминается при обсуждении DBScan.

6) Обработка естественного языка (NLP). Этот раздел посвящен изучению классических методов, применяемых в обработке естественного языка. Раздел начинает устаревать и, вероятно, вскоре будет заменен. Здесь рассматриваются предобработка текста, базовые токенизаторы и методы векторизации текстов. Далее изучаются методы моделирования тематик: LSA/LSI, LDA и базовые эмбеддинги: word2vec, glove, fasttext. Несмотря на то что студенты на момент изучения этого раздела еще не знакомы с нейронными сетями, они без особых проблем понимают устройство этих эмбеддингов, хотя и не до конца осознают, как именно происходит их настройка.

7) Рекомендательные системы. Этот блок посвящен не столько ознакомлению с рекомендательными системами, сколько изучению подходов и идей, применяемых в этой области, поскольку они могут быть полезны и в других областях машинного обучения. Основное внимание уделяется рекомендательным системам на основе коллаборативной фильтрации (user-based, item-to-item-based), с отдельным блоком по рекомендациям

на основе матричной факторизации. Также немного времени отведено на рекомендации на основе контента и гибридные рекомендательные системы.

8) Теория Вапника – Червоненкиса (VC-теория). Это единственный полностью теоретический раздел. Он был введен относительно недавно с целью ознакомления студентов с теоретическими основами машинного обучения, поскольку без этого машинное обучение воспринимается как подраздел теории оптимизации, в то время как эта область имеет свою специфику в необходимости минимизировать реальный (неизвестный) риск через эмпирический риск, полученный на основе выборки.

   Теория Вапника – Червоненкиса вводит понятие размерности класса моделей и объясняет, как реальный риск может отличаться от эмпирического в зависимости от размерности рассматриваемого класса моделей. Этот раздел завершает первый семестр курса.

9) Введение в нейронные сети. Вводится перцептрон Розенблатта и рассматривается его обобщение на глубокие нейронные сети. Вводится алгоритм обратного распространения ошибки, на основе которого обучаются современные сети. На примере небольшой нейронной сети с несколькими слоями проводится полный цикл вычислений прямого и обратного распространения. Здесь также обсуждаются модификации градиентного спуска, такие как RMSprop, Adam, проблемы затухающих и разрывающихся градиентов и пакетная нормализация (batch normalization).

10) Сверточные нейронные сети. Вводятся понятие сверточного слоя, виды пуллинга и их параметры. Объясняется физический смысл свертки. Далее рассматриваются архитектуры классических сверточных сетей для задач классификации, семантической сегментации и детектирования объектов.

    Новые архитектуры для решения этих задач появляются постоянно и было решено не пытаться гнаться за все новыми архитектурами, а дать понимание ключевых идей, таких как остаточные соединения (residual connections) и факторизация сверток. Как показывает опыт преподавания, наибольшую сложность для студентов вызывают алгоритмы детектирования объектов, поэтому их изучение начинается с простейших, хоть и устаревших, алгоритмов типа R-CNN.

    Заканчивается раздел обсуждением вопросов дообучения, файнтюнинга (fine-tuning) и переноса знаний (knowledge transfer).

11) Рекуррентные нейронные сети. Этот раздел также является кандидатом на удаление или, по крайней мере, сокращение в будущих редакциях курса, ввиду все более редкого использования рекуррентных сетей на практике. Здесь рассматриваются различные топологии рекуррентных сетей, архитектуры LSTM, GRU и механизм внимания (attention mechanism).

12) Трансформеры. Трансформеры полностью преобразовали подходы к работе с текстами. С каждым годом этот раздел становится все более обширным. Здесь обсуждается механизм самовнимания (self-attention), общая схема трансформеров и подробно рассматриваются модели BERT и GPT.

13) Автокодировщики. Рассматриваются неполные, разреженные и шумоподавляющие автокодировщики. Далее изучаются вариационные автокодировщики (VAE) и общие идеи, заимствованные из генеративно-состязательных сетей (GAN). В будущем, вероятно, будет добавлено рассмотрение диффузионных нейронных сетей.

14) Обучение с подкреплением. Этот раздел необходим, в частности, для добавления раздела по большим языковым моделям (LLM), но важен и сам по себе. По теме обучения с подкреплением можно было бы вести отдельный курс, однако здесь ставится задача дать слушателям базовые представления о решаемых задачах. Рассматриваются задачи многорукого бандита, Q-Learning, Deep Q-Learning, а также подходы на основе Policy Optimization.

    Практических задач по этому блоку нет, только базовая теория.

15) Интерпретация моделей. Обсуждаются моделезависимые методы интерпретации линейных и деревяных моделей, моделезависимые методы SHAP и LIME, а также различные методы интерпретации нейронных сетей (Saliency map, Grad-CAM, Integrated Gradients).

Для получения зачета или допуска к экзамену по большинству разделов студенту необходимо выполнить практическую задачу. В типовой задаче студенту предлагается взять определенный набор данных и применить к нему модели из соответствующего раздела, достигнув необходимого минимального уровня показателя целевых метрик. Задания выполняются на языке Python с использованием наиболее популярных библиотек для машинного обучения и анализа данных: numpy[3], pandas[4], scikit-learn[5]. На ранних порах

[3]https://numpy.org/
[4]https://pandas.pydata.org/
[5]https://scikit-learn.org/stable/

курса практика по нейронным сетям проводилась с использованием фреймворка tensorflow[6], но ввиду того, что pytorch[7] значительно потеснил его по популярности, в настоящее время вся работа с нейронными сетями ведется с использованием библиотек pytorch и transformers[8]. Студенты сдают задачи в так называемых ноутбуках (jupyter notebooks[9]) — интерактивной среде для работы с кодом, которая позволяет не только исполнять код, но и визуализировать данные и оставлять текстовые комментарии. Таким образом, в этой среде студент не только предоставляет решение, но и сразу оформляет отчет, в котором описывает выводы, сделанные при работе с данными, а также дает свою оценку качества работы моделей.

3°. Практические проекты. Совместно со студентами, прошедшими курс выполняются практические проекты, ориентированные на решение реальных задач в области нейробиологии и нейротехнологий. Все проекты были инициированы двумя организациями: биологическим факультетом СПбГУ и Институтом Мозга Человека им. Н.П. Бехтеревой РАН. Результаты работы отражены в совместных публикациях ([1-4]).

Участие студентов в проектах организовано в формате курсовых и дипломных работ, что позволяет интегрировать образовательный процесс с решением прикладных задач. Первоначально предпринимались попытки объединять студентов в команды для совместной работы над проектами. Однако практика показала, что такой формат оказывается трудно реализуемым. Основными проблемами стали различия в степени вовлеченности и индивидуальных графиках студентов. Это создавало зависимость одних участников от других, что могло приводить к задержкам в выполнении задач. К тому же, для студентов важно, чтобы проект заканчивался работой, в которой он сможет четко отразить свой вклад. При совместной работе над прооектами такое разделение осуществить сложнее.

С учетом этих обстоятельств было решено разбивать работу над проектом на независимые подходы. Каждый студент занимается своим подходом, не блокируя остальных. При этом элементы совместной работы сохраняются: продуктивные идеи и наработки одного участника могут быть использованы другими. Например, новый метод предобработки данных, предложенный одним студентом и показавший хорошие результаты, может быть протестирован и адаптирован для других подходов. Такой формат работы обеспечивает обмен знаниями и опытом, не создавая взаимозависимости.

Ниже приведено краткое описание некоторых проектов, реализованных вместе со студентами.

[6]https://www.tensorflow.org/
[7]https://pytorch.org/
[8]https://huggingface.co/docs/transformers/en/index
[9]https://jupyter.org/

4°. Определение намерения совершить самоинициированное движение. Данный проект был инициирован биологическим факультетом СПбГУ и направлен на решение задачи детектирования намерений испытуемых совершить самоинициированное движение. Эксперимент состоял в том, что испытуемым, сидящим в удобном кресле, необходимо было нажать на клавишу в произвольные моменты времени по своему желанию, что фиксировалось с помощью ЭЭГ. На основе этих записей требовалось с использованием методов машинного обучения определить момент, предшествующий началу движения, то есть момент формирования намерения это сделать.

Процесс решения задачи включал несколько ключевых этапов:

- **Предварительная обработка данных.** На данном этапе из ЭЭГ сигналов удалялись артефакты движения глаз, а также участки сигналов, амплитуды которых выходят за пределы заранее установленного диапазона. Это позволяло устранить шум и подготовить данные для дальнейшего анализа.

- **Нарезка сигналов на эпохи.** Каждый ЭЭГ-сигнал был разделен на эпохи — временные сегменты, начинающиеся за 2500 мс до предполагаемого начала движения и заканчивающиеся через 500 мс после его начала. Эпохи, содержащие артефакты или нарушения, были исключены из дальнейшего анализа, чтобы не искажать результаты классификации.

- **Выделение признаков.** Для успешной классификации необходимо было извлечь наиболее информативные признаки из ЭЭГ сигналов. Признаки были выделены как в частотной, так и во временной областях, что позволило улавливать различные характеристики сигналов, связанные с намерением совершить движение.

- **Применение моделей машинного обучения.** Задача классификации заключалась в бинарном определении: было ли в эпохе совершено движение или нет. Для этого использовались различные алгоритмы классификации, такие как метод опорных векторов (SVM), случайный лес (Random Forest) и другие. При выборе модели учитывался баланс между точностью классификации и возможностью интерпретации результатов, чтобы можно было выделить ключевые характеристики сигналов, которые отвечают за факт совершения движения.

В результате реализации проекта удалось достичь средней точности классификации в 72% для нажатий правой рукой и 77% для нажатий левой рукой. Эти результаты демонстрируют успешное применение методов машинного обучения для анализа ЭЭГ и могут быть использованы для дальнейших исследований в области нейроуправления и нейроинтерфейсов.

5°. Классификация психиатрических заболеваний. Вторая задача проекта была поставлена представителями Института Мозга Человека им. Н.П. Бехтеревой РАН. Задача заключалась в классификации ЭЭГ сигналов здоровых людей и людей с диагнозами шизофрения и обсессивно-компульсивное расстройство (ОКР). Для решения этой задачи была использована база данных ЭЭГ, подготовленная ИМЧ РАН.

В базе данных содержатся записи ЭЭГ, полученные во время выполнения испытуемыми Go-NoGo теста, разработанного коллективом ИМЧ РАН. Тест состоял в предъявлении визуальных изображений животных, растений и человека, на которые испытуемые должны были реагировать либо действием, либо игнорированием в зависимости от инструкции. Среди испытуемых были как здоровые люди, так и пациенты с диагнозами шизофрения и ОКР. Помимо ЭЭГ записей, база данных включает поведенческие данные, такие как количество ошибок, время реакции и другие параметры, характеризующие точность и скорость выполнения теста.

Процесс решения задачи включал следующие этапы:

- **Предобработка данных.** Для извлечения информативных признаков из ЭЭГ сигналов были рассчитаны компоненты вызванных потенциалов (ВП) для двух условий теста: Go и NoGo. Для каждого условия использовался метод анализа независимых компонент (ICA), что позволило выделить 11 компонентов для каждого состояния. Для шизофрении было выбрано 18 компонентов, а для ОКР — 14.

- **Выбор признаков.** Для каждого компонента ВП и для всего набора сигналов были извлечены признаки, такие как минимальные, максимальные и средние значения в пересекающихся временных окнах. Также вычислялись глобальные статистики, включая среднее, максимум и минимум для каждого сигнала. Параметры окон и сдвигов, а также гиперпараметры модели выбирались независимо для каждого условия и алгоритма классификации.

- **Отбор признаков.** Для борьбы с переобучением и ускорения процесса обучения были использованы методы отбора признаков, такие как последовательный отбор и усеченный SVD.

- **Классификация.** Для классификации сигналов использовались различные алгоритмы, включая метод опорных векторов (SVM), логистическую регрессию, случайный лес и метод k-ближайших соседей (kNN). Все параметры классификации были оптимизированы с помощью процедуры поиска по сетке с перекрестной проверкой.

Результаты эксперимента показали, что наилучшую точность классификации для шизофрении дал метод SVM при комбинировании признаков на основе ICA и поведенческих данных. Сбалансированная точность классификации составила 97%. Для ОКР также был выбран метод SVM, который показал точность 92%.

Кроме того, была проведена статистическая проверка значимости различий в точности классификации для различных наборов признаков, а также построены доверительные интервалы для оценок точности моделей. Полученные результаты демонстрируют высокую эффективность машинного обучения для диагностики психических заболеваний на основе ЭЭГ.

ЛИТЕРАТУРА

1. N. Shanarova, M. Pronina, M. Lipkovich, J. Kropotov. Machine learning based diagnostics of schizophrenia patients. *2022 6th Scientific School Dynamics of Complex Networks and their Applications (DCNA)*, 252–255, 2022.

2. N. Shanarova, M. Pronina, M. Lipkovich, V. Ponomarev, A. Müller, J. Kropotov. Application of Machine Learning to Diagnostics of Schizophrenia Patients Based on Event-Related Potentials. *Diagnostics*, 13(3), 2023.

3. М.М. Липкович, А.Р. Сагатдинов. Применение алгоритма “Полоска” для онлайнового декодирования ЭЭГ-паттернов. *Мехатроника, автоматизация, управление*, 24(6):300–306, 2023.

4. M. Lipkovich, V. Knyazeva, A. Aleksandrov, N. Shanarova, A. Sagatdinov, A. Fradkov. Evoked Potentials Detection During Self-Initiated Movements Using Machine Learning Approach. *2023 Fifth International Conference Neurotechnologies and Neurointerfaces (CNN)*, 47–50, 2023.