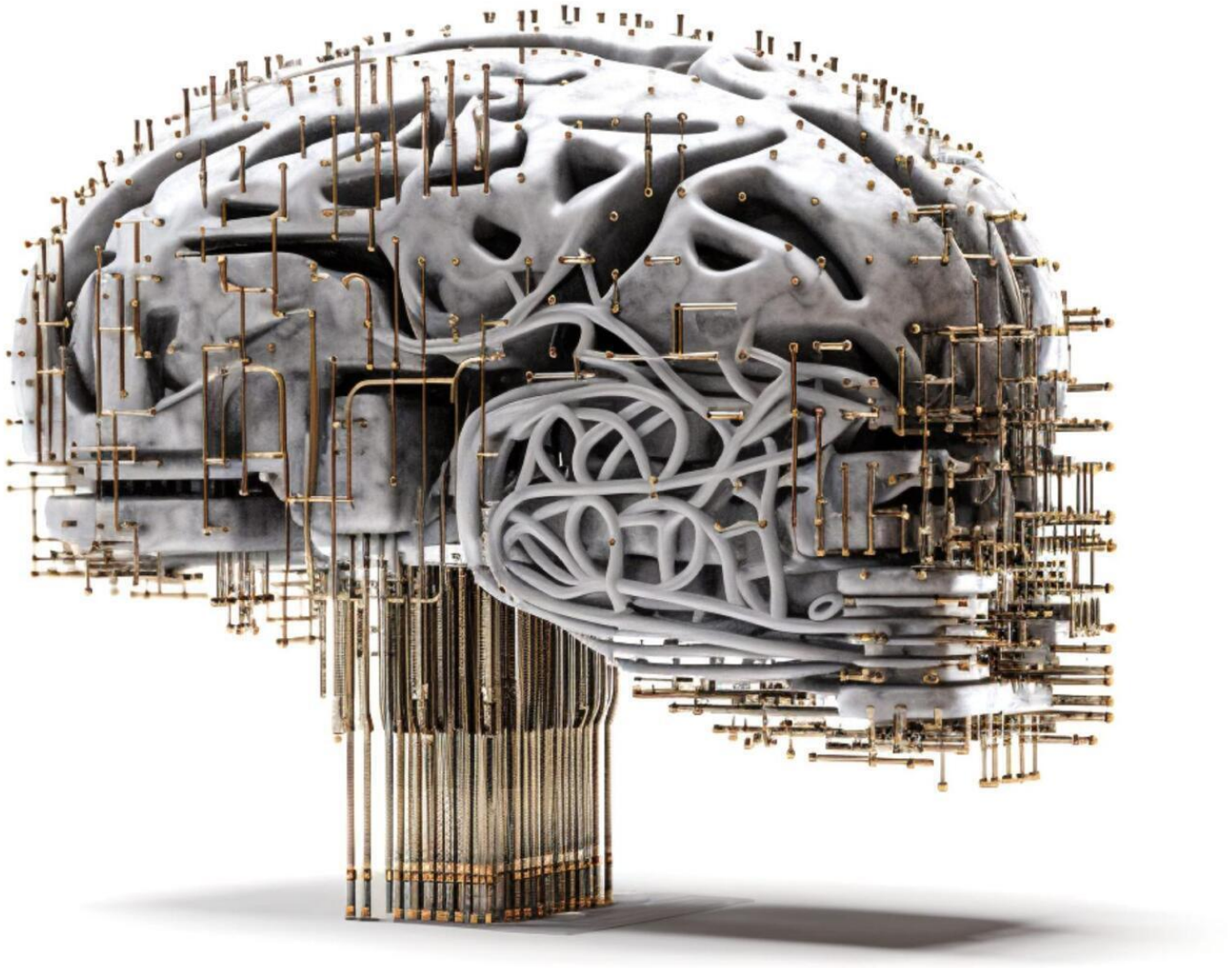
Артем Демиденко / ИИ



Машинное обучение

Погружение в технологию

- Что такое Машинное обучение?
- Погружение в технологию Машинного обучения
- Практическое применение Машинного обучения

Артем Демиденко

Машинное обучение. Погружение в технологию

«Автор»
2023

Демиденко А.

Машинное обучение. Погружение в технологию / А. Демиденко — «Автор», 2023

Практическое руководство, предназначенное для всех, кто хочет войти в мир машинного обучения и освоить его основы. Авторы книги предлагают читателям увлекательное путешествие в эту захватывающую область, начиная с основных концепций и принципов машинного обучения и заканчивая практическими навыками построения и обучения моделей.Внутри книги читатели найдут понятные объяснения ключевых алгоритмов машинного обучения, таких как регрессия, классификация, кластеризация и глубокое обучение. Они узнают, как подготовить данные для обучения моделей, как выбрать и настроить подходящие алгоритмы, а также как оценивать и улучшать производительность моделей.

Содержание

Артем Демиденко
Машинное обучение.
Погружение в технологию

Глава 1: Основы Машинного обучения

1.1 Введение в Машинное обучение

Машинное обучение (Machine Learning) – это область искусственного интеллекта, которая изучает разработку алгоритмов и моделей, позволяющих компьютерам извлекать полезные знания из данных и принимать решения на основе этой информации. Одной из основных идей Машинного обучения является использование данных для построения модели, которая обобщает эти данные и может применяться к новым, ранее не виденным данным.

Процесс обучения модели включает в себя несколько этапов. Сначала необходимо иметь обучающую выборку, которая состоит из пар «входные данные – выходные данные» или «характеристики – целевая переменная». Входные данные представляют собой информацию, на основе которой модель должна сделать предсказание, а выходные данные или целевая переменная представляют собой ожидаемый ответ или результат для данного входа.

Цель обучения модели заключается в подгонке ее параметров на основе обучающей выборки таким образом, чтобы модель могла корректно обрабатывать новые данные и делать предсказания для них. Этот процесс достигается путем минимизации ошибки или разницы между предсказанными значениями и фактическими значениями в обучающей выборке.

Существует различные подходы и алгоритмы в Машинном обучении, включая линейную регрессию, логистическую регрессию, деревья решений, случайные леса, градиентный бустинг, нейронные сети и многое другое. Каждый из этих алгоритмов имеет свои особенности и применяется в зависимости от типа задачи и характеристик данных.

Одним из ключевых аспектов Машинного обучения является обобщение модели на новые данные. Обобщение означает способность модели делать предсказания для данных, которые она ранее не видела. Чем лучше модель обобщает данные, тем более эффективной она является. Обобщение достигается путем обучения на достаточно разнообразных и представительных данных, а также с использованием методов регуляризации, которые помогают контролировать сложность модели и избегать переобучения.

Машинное обучение имеет широкий спектр применений и используется во многих областях, включая компьютерное зрение, обработку естественного языка, рекомендательные системы, финансы, медицину и другие. Прогресс и инновации в области Машинного обучения продолжают улучшать нашу способность анализировать и понимать данные, делать предсказания и принимать более информированные решения.

1.2 История Машинного обучения

История Машинного обучения насчитывает несколько десятилетий развития и прогресса. Одним из первых знаков возникновения Машинного обучения является появление линейной регрессии и метода наименьших квадратов в начале 19-го века. Это был первый шаг к формализации процесса обучения моделей на основе данных.

В середине 20-го века появились первые искусственные нейронные сети, которые были вдохновлены биологическими нейронными сетями и работой мозга. Однако, развитие Машинного обучения замедлилось из-за ограниченных вычислительных ресурсов и сложностей в обучении глубоких нейронных сетей.

В конце 20-го и начале 21-го века произошел резкий прорыв в Машинном обучении. С развитием вычислительной мощности и появлением больших объемов данных появилась возможность обучать сложные модели глубокого обучения. Алгоритмы глубокого обучения, такие как сверточные нейронные сети и рекуррентные нейронные сети, привели к значительным достижениям в областях компьютерного зрения, обработки естественного языка, рекомендательных систем и других областях.

Важным моментом в развитии Машинного обучения стало появление статистического подхода к обучению. В середине 20-го века появились методы статистического обучения, включая линейную и логистическую регрессию, метод наименьших квадратов и метод максимального правдоподобия. Эти методы основывались на статистических принципах и позволяли делать предсказания на основе данных.

Еще одним важным этапом в истории Машинного обучения было развитие метода опорных векторов (Support Vector Machines, SVM) в 1990-х годах. SVM стало мощным алгоритмом для решения задач классификации и регрессии, основанным на идее нахождения гиперплоскости, которая наилучшим образом разделяет данные разных классов.

В последние десятилетия наблюдается интенсивное развитие Машинного обучения и его применение в различных областях. С появлением больших объемов данных и увеличением вычислительной мощности появились новые методы и алгоритмы, такие как глубокое обучение, рекуррентные нейронные сети, сверточные нейронные сети и генетические алгоритмы.

Важным событием в истории Машинного обучения стал конкурс ImageNet Large Scale Visual Recognition Challenge (ILSVRC), который был проведен в 2010 году. Этот конкурс стимулировал развитие глубокого обучения и значительно улучшил результаты в области компьютерного зрения.

Сегодня Машинное обучение играет важную роль во многих сферах, включая медицину, финансы, автомобильную промышленность, рекламу, кибербезопасность и многое другое. Большие компании активно применяют методы Машинного обучения для анализа данных, оптимизации бизнес-процессов и улучшения пользовательского опыта.

С развитием Машинного обучения возникают и новые вызовы и вопросы, такие как этика и безопасность, интерпретируемость моделей и проблемы справедливости и предвзятости. Поэтому важно постоянно развивать и улучшать методы Машинного обучения, чтобы использовать его потенциал в наилучшем интересе человечества.

1.3 Типы задач в Машинном обучении

Машинное обучение решает различные типы задач в зависимости от характера входных данных и желаемого результата. Вот некоторые из основных типов задач в Машинном обучении:

Задачи классификации: в этом типе задачи модель должна отнести объекты к определенным классам или категориям. Например, модель может классифицировать электронные письма на спам и не спам, или определять, является ли изображение кошкой или собакой. В задачах классификации модель обучается прогнозировать класс или категорию, к которой принадлежит объект на основе его характеристик или признаков. Классификация является одним из самых распространенных и важных типов задач в Машинном обучении. Вот некоторые примеры задач классификации:

1. Классификация электронных писем на спам и не спам: Модель обучается на основе различных характеристик электронных писем, таких как слова, фразы, заголовки и т. д., и предсказывает, является ли письмо спамом или не спамом. Это помогает фильтровать нежелательную почту и улучшает опыт пользователей.
2. Классификация изображений: Модель обучается классифицировать изображения на определенные категории. Например, модель может определять, является ли изображение

кошкой или собакой, определять виды растений или классифицировать объекты на дорожных сценах.

3. Классификация текстов: Модель может классифицировать тексты на основе их содержания. Например, модель может определять, относится ли отзыв о продукте к положительному или отрицательному классу, классифицировать новостные статьи по темам или определять тональность текста.

4. Классификация медицинских данных: Модель может использоваться для классификации медицинских данных, таких как изображения рентгена или снимки МРТ, для определения наличия определенных заболеваний или патологий.

5. Классификация финансовых транзакций: Модель может классифицировать финансовые транзакции на основе их характеристик, чтобы обнаружить мошенническую активность или аномалии.

Для решения задач классификации используются различные алгоритмы и методы, включая логистическую регрессию, метод опорных векторов (SVM), решающие деревья, случайные леса, градиентный бустинг и нейронные сети. Выбор конкретного метода зависит от характеристик данных, объема данных и требуемой точности классификации.

Задачи регрессии: в регрессионных задачах модель стремится предсказать непрерывные числовые значения. Например, модель может предсказывать стоимость недвижимости на основе ее характеристик, или прогнозировать спрос на товары на основе исторических данных. Вот несколько примеров задач регрессии:

1. Прогнозирование цен на недвижимость: Модель обучается на основе характеристик недвижимости, таких как размер, расположение, количество комнат и т. д., и предсказывает стоимость недвижимости. Это полезно для покупателей и продавцов недвижимости, агентов по недвижимости и оценщиков.

2. Прогнозирование спроса на товары: Модель может использоваться для прогнозирования спроса на товары или услуги на основе исторических данных о продажах, ценах, маркетинговых активностях и других факторах. Это помогает компаниям оптимизировать производство, планирование запасов и маркетинговые стратегии.

3. Прогнозирование финансовых показателей: Модель может предсказывать финансовые показатели, такие как выручка, прибыль, акции или курс валюты, на основе исторических данных и других факторов, таких как экономические показатели, политические события и т. д. Это полезно для инвесторов, трейдеров и финансовых аналитиков.

4. Прогнозирование временных рядов: Модель может использоваться для прогнозирования временных рядов, таких как погода, трафик, продажи и другие параметры, которые меняются со временем. Это полезно для планирования и управления в различных отраслях, включая транспорт, энергетику и розничную торговлю.

5. Медицинские прогнозы: Модель может предсказывать результаты медицинских тестов, такие как прогнозирование заболеваемости, выживаемости пациентов или оценку эффективности лечения на основе клинических и биологических характеристик пациентов.

В задачах регрессии используются различные алгоритмы, включая линейную регрессию, метод опорных векторов (SVM), решающие деревья, случайные леса, градиентный бустинг и нейронные сети. Выбор конкретного метода зависит от характеристик данных, структуры модели и требуемой точности предсказания.

Задачи кластеризации: в этом типе задачи модель должна группировать объекты на основе их сходства без заранее заданных классов. Кластеризация может помочь выявить скрытые структуры в данных или идентифицировать группы схожих объектов. Вот некоторые примеры задач кластеризации:

1. Сегментация клиентов: Кластеризация может использоваться для разделения клиентов на группы схожих характеристик, таких как покупательские предпочтения, поведение

или демографические данные. Это помогает компаниям в создании более целевых маркетинговых стратегий и персонализации предложений.

2. Анализ социальных сетей: Кластеризация может помочь в выявлении сообществ в социальных сетях на основе взаимодействий между пользователями. Это позволяет понять структуру социальных связей и определить влиятельных пользователей или группы схожих интересов.

3. Анализ текстовых данных: Кластеризация текстовых данных может помочь в группировке документов по схожей тематике или контексту. Например, в новостной отрасли это может использоваться для автоматической категоризации новостей по темам или для выявления семантических групп текстов.

4. Анализ медицинских данных: Кластеризация может быть применена для идентификации групп пациентов с похожими характеристиками или симптомами. Это может помочь в определении подгрупп пациентов с определенными заболеваниями или позволить персонализировать лечение.

5. Обнаружение аномалий: Кластеризация может быть использована для выявления аномальных или необычных групп объектов. Путем сравнения объектов с основным кластером модель может идентифицировать аномалии или выбросы в данных.

Для решения задач кластеризации применяются различные алгоритмы, включая иерархическую кластеризацию, метод k-средних, плотностные методы и алгоритмы DBSCAN. Выбор конкретного метода зависит от структуры данных, размера выборки и требуемого уровня детализации кластеров.

Задачи обнаружения аномалий: такие задачи связаны с выявлением редких или необычных объектов или событий. Например, модель может обнаружить подозрительную кредитную транзакцию или аномалию в работе промышленного оборудования. Вот некоторые примеры задач обнаружения аномалий:

1. Обнаружение мошенничества: В финансовой сфере модель может использоваться для обнаружения подозрительных кредитных транзакций, мошеннических операций или фальшивых документов. Путем анализа и сравнения паттернов поведения модель может выявить аномальные действия.

2. Обнаружение сетевых атак: Модель может применяться для обнаружения аномального сетевого трафика или вторжений в компьютерные системы. Путем анализа характеристик сетевой активности можно выявить аномальные или вредоносные действия.

3. Мониторинг промышленного оборудования: В производственных средах модель может использоваться для обнаружения аномалий в работе оборудования, таких как отклонения в сенсорных данных, вибрации или изменений в параметрах производства. Это позволяет предотвратить сбои и увеличить эффективность обслуживания.

4. Детектирование медицинских аномалий: В медицинской области модель может применяться для обнаружения аномальных паттернов в медицинских изображениях, временных рядах пациентов или результатов анализов. Это помогает выявить ранние признаки заболеваний или необычные медицинские состояния.

5. Мониторинг систем безопасности: Модель может использоваться для обнаружения аномалий в системах безопасности, таких как контроль доступа или видеонаблюдение. Путем анализа поведения людей или объектов модель может выявить подозрительные или незаконные действия.

Для решения задач обнаружения аномалий применяются различные методы, включая статистические методы, методы машинного обучения (например, методы выбросов) и методы глубокого обучения. Алгоритмы такие, как One-class SVM, Isolation Forest и автоэнкодеры, широко используются для обнаружения аномалий в данных. Выбор конкретного метода зависит от типа данных, доступных метрик аномальности и особенностей конкретной задачи.

Задачи понижения размерности: в этом типе задачи модель стремится сократить размерность данных, сохраняя при этом важные информационные характеристики. Это полезно для визуализации данных и удаления шума или лишних признаков. Задачи понижения размерности в Машинном обучении имеют целью снижение размерности данных, то есть уменьшение числа признаков или переменных, представляющих данные, при этом сохраняя важные информационные характеристики. Это полезно для улучшения визуализации данных, ускорения вычислений и удаления шума или избыточности.

Процесс понижения размерности основан на идее о том, что существует некоторая скрытая структура в данных, которую можно извлечь, уменьшив размерность. Вот некоторые методы понижения размерности:

1. Метод главных компонент (Principal Component Analysis, PCA): PCA является одним из наиболее распространенных методов понижения размерности. Он выполняет линейное преобразование данных, чтобы получить новые переменные, называемые главными компонентами, которые представляют наибольшую дисперсию в данных. Таким образом, PCA позволяет уменьшить размерность данных, сохраняя при этом как можно больше информации.
2. Многомерное шкалирование (Multidimensional Scaling, MDS): MDS пытается сохранить относительные расстояния между объектами в исходных данных при проецировании их на пространство меньшей размерности. Это позволяет визуализировать данные в двух или трех измерениях, сохраняя их структуру.
3. Автоэнкодеры (Autoencoders): Автоэнкодеры являются нейронными сетями, которые обучаются реконструировать входные данные на выходе. Они состоят из энкодера, который сжимает данные в скрытое пространство меньшей размерности, и декодера, который восстанавливает данные обратно. Автоэнкодеры могут использоваться для эффективного понижения размерности данных и изучения их скрытых признаков.

Задачи рекомендации в Машинном обучении связаны с предложением наиболее релевантных элементов или ресурсов пользователю на основе его предпочтений, истории взаимодействий или анализа данных. Например, в рекомендательных системах модель может предлагать пользователю фильмы, музыку, товары или новости на основе его предыдущих покупок, оценок или поведения.

Задачи рекомендации: в этом типе задачи модель стремится предложить пользователю наиболее подходящие элементы или рекомендации на основе его предыдущего поведения или предпочтений. Например, модель может рекомендовать фильмы, музыку или товары покупателям. Задачи рекомендации в Машинном обучении направлены на предоставление пользователю наиболее подходящих рекомендаций на основе его предыдущего поведения, предпочтений или характеристик. Целью является улучшение опыта пользователя и увеличение его удовлетворенности. Вот некоторые примеры задач рекомендации:

1. Рекомендация товаров: Это один из самых распространенных видов задач рекомендации. Модель анализирует предпочтения пользователя, историю его покупок или оценки товаров, чтобы предложить ему наиболее подходящие товары или услуги. Например, платформы электронной коммерции могут рекомендовать продукты, основываясь на предыдущих покупках или схожих предпочтениях других пользователей.
2. Рекомендация контента: Модель может рекомендовать пользователю интересный контент, такой как статьи, видео, новости или музыка. Это основано на анализе истории просмотров, оценок или предпочтений пользователя, а также на сходстве с другими пользователями. Например, платформы потокового видео могут рекомендовать фильмы или сериалы на основе предыдущих просмотров и оценок.
3. Рекомендация друзей или социальных связей: Модель может помочь пользователю найти подходящих друзей или социальные связи на основе его интересов, деятельности или

сходства с другими пользователями. Это может быть полезно для социальных сетей, профессиональных платформ или приложений знакомств.

4. Рекомендация маршрутов и путешествий: Модель может предлагать пользователю оптимальные маршруты путешествий, рекомендовать достопримечательности, рестораны или отели на основе его предпочтений, бюджета или предыдущего опыта. Это может быть полезно для туристических агентств, сервисов такси или приложений для путешествий.

Для решения задач рекомендации применяются различные методы, включая коллаборативную фильтрацию, контент-базированные методы, гибридные подходы и методы глубокого обучения. Алгоритмы анализируют большие объемы данных, используют методы паттерн-распознавания и выявления сходств, чтобы предсказывать наиболее релевантные рекомендации для каждого пользователя.

Задачи усиления: в этом типе задачи модель обучается принимать последовательность действий в среде с целью максимизации награды. Такие задачи широко применяются в области управления роботами, автономных агентов и игровой индустрии. Основная идея задач усиления заключается в том, что модель-агент обучается на основе проб и ошибок, пытаясь найти оптимальную стратегию действий для достижения максимальной награды. В процессе обучения модель получает информацию о текущем состоянии среды, выбирает действие, выполняет его, получает награду и переходит в новое состояние. Модель стремится улучшить свою стратегию, максимизируя суммарную награду, которую она получает в ходе взаимодействия со средой.

Задачи усиления широко применяются в различных областях, таких как управление роботами и автономными системами, разработка игр, оптимальное управление процессами и другие. Примеры применения задач усиления включают обучение роботов ходить, игры на компьютере, автономное управление автомобилем, управление финансовыми портфелями и многое другое.

Основные алгоритмы и подходы в усилении включают Q-обучение, SARSA, Deep Q-Networks (DQN), Proximal Policy Optimization (PPO) и многие другие. Эти алгоритмы используются для моделирования взаимодействия агента со средой, оценки ценности действий, определения оптимальной стратегии и обновления параметров модели на основе полученной награды.

Задачи генерации: в этом типе задачи модель обучается генерировать новые данные, такие как изображения, звуки или тексты. Например, модель может генерировать реалистичные фотографии или синтезировать речь. Процесс генерации данных включает в себя обучение модели на большом объеме образцовых данных и последующую способность модели создавать новые примеры, которые соответствуют тем же характеристикам и структуре, что и исходные данные. Задачи генерации находят применение в различных областях, таких как компьютерное зрение, обработка естественного языка, музыкальная композиция и другие.

Примеры задач генерации включают в себя:

1. Генерация изображений: модель обучается создавать новые изображения, которые могут быть реалистичными фотографиями, абстрактными картинами или даже реалистичными лицами.

2. Генерация текста: модель обучается генерировать новые тексты, которые могут быть статьями, романами, поэзией или даже программным кодом.

3. Генерация звука: модель обучается генерировать новые аудиофайлы, которые могут быть речью, музыкой или звуковыми эффектами.

4. Генерация видео: модель обучается создавать новые видеофрагменты, которые могут быть анимациями, синтезированными сценами или даже виртуальной реальностью.

Для решения задач генерации используются различные методы, включая глубокие генеративные модели, такие как генеративные состязательные сети (GAN), вариационные автоэн-

кодеры (VAE) и авторегрессионные модели. Эти методы позволяют модели генерировать новые данные, имитируя статистические свойства исходных данных и создавая новые, качественно подобные примеры.

Задачи обучения с подкреплением: в этом типе задачи модель взаимодействует с динамической средой и учится принимать оптимальные решения для достижения заданной цели. Это типичный подход для обучения агентов в играх и робототехнике. Задачи обучения с подкреплением (reinforcement learning) относятся к типу задач, в которых модель (агент) взаимодействует с динамической средой и учится принимать оптимальные решения для достижения заданной цели. В этом типе задач модель обучается на основе отклика (награды) от среды, которая может изменяться в зависимости от принятых агентом действий. Задачи обучения с подкреплением находят широкое применение в области игровой индустрии, робототехники, автономных агентов и управления системами в реальном времени.

Процесс обучения с подкреплением включает в себя цикл взаимодействия между агентом и средой, где агент принимает решения на основе текущего состояния среды, выполняет действия, а среда возвращает отклик в виде награды или штрафа. Цель агента состоит в том, чтобы максимизировать накопленную награду в долгосрочной перспективе. Для этого агенту необходимо определить оптимальную стратегию действий, которая будет обеспечивать наилучший результат.

В задачах обучения с подкреплением используются понятия состояния (state), действия (action), награды (reward) и стратегии (policy). Состояние представляет собой описание текущего состояния среды, действия определяют выбор агента в данном состоянии, награды предоставляют обратную связь от среды, указывая, насколько хорошо агент выполнил свою задачу, а стратегия определяет, какие действия должен предпринимать агент в каждом состоянии.

Алгоритмы обучения с подкреплением, такие как Q-обучение (Q-learning) и глубокое обучение с подкреплением (deep reinforcement learning), используются для обучения агентов принимать оптимальные решения в динамических средах. Эти алгоритмы исследуют пространство состояний и действий, обновляют значения Q-функции (оценки ценности состояния-действия) и настраивают стратегию агента для достижения максимальной награды.

Задачи обучения с подкреплением широко применяются для обучения агентов играть в компьютерные игры, управлять роботами и автономными транспортными средствами, управлять системами энергетики и многими другими приложениями, где необходимо принимать решения в динамической среде с целью достижения оптимальных результатов.

Задачи обработки естественного языка: в этих задачах модель работает с текстовыми данными, понимая и генерируя естественный язык. Это включает в себя задачи машинного перевода, анализа тональности, генерации текста и другие. Ниже приведены некоторые из задач, которые решаются в области обработки естественного языка:

1. Машинный перевод: Это задача автоматического перевода текста с одного языка на другой. Модели машинного перевода обучаются понимать и генерировать тексты на разных языках, используя различные подходы, такие как статистический машинный перевод, нейронные сети и трансформеры.
2. Анализ тональности: Задача анализа тональности заключается в определении эмоциональной окраски текста, например, положительной, отрицательной или нейтральной. Это может быть полезно в анализе отзывов, комментариев, социальных медиа и других текстовых данных.
3. Классификация текстов: Эта задача заключается в классификации текстовых документов по определенным категориям или темам. Модели могут классифицировать новости, электронные письма, социальные медиа и другие тексты на основе их содержания.
4. Извлечение информации: Задача извлечения информации заключается в автоматическом извлечении структурированных данных из текста, таких как именованные сущности,

ключевые факты, даты и другая релевантная информация. Например, извлечение информации может быть использовано для автоматического заполнения баз данных или составления сводок новостей.

5. Генерация текста: В этой задаче модели обучаются генерировать новые текстовые данные на основе заданного контекста или условия. Примерами являются генерация автоматических ответов на сообщения, синтез статей и создание текстовых описаний.

Это лишь некоторые из задач, с которыми сталкиваются в обработке естественного языка. NLP играет важную роль в различных приложениях, включая автоматический перев

1.4 Принципы обучения с учителем и без учителя

Обучение с учителем и обучение без учителя являются двумя основными подходами в Машинном обучении.

Обучение с учителем: в этом подходе модель обучается на основе обучающей выборки, которая состоит из пар "входные данные – выходные данные" или "характеристики – целевая переменная". Модель учится находить зависимости между входными данными и соответствующими выходными данными, что позволяет ей делать предсказания для новых данных. Примерами алгоритмов обучения с учителем являются линейная регрессия, логистическая регрессия, метод k ближайших соседей и градиентный бустинг. Примеры алгоритмов обучения с учителем, которые мы упомянули:

1. Линейная регрессия: Этот алгоритм используется для решения задач регрессии, где модель стремится предсказывать непрерывные числовые значения. Линейная регрессия моделирует линейную зависимость между входными признаками и целевой переменной.

2. Логистическая регрессия: Этот алгоритм также используется в задачах классификации, но вместо предсказания числовых значений модель предсказывает вероятности принадлежности к определенным классам. Логистическая регрессия обычно применяется для бинарной классификации.

3. Метод k ближайших соседей (k-NN): Это простой алгоритм классификации и регрессии, основанный на принципе ближайших соседей. Модель классифицирует новый пример на основе ближайших к нему соседей из обучающей выборки.

4. Градиентный бустинг: Этот алгоритм используется для задач классификации и регрессии и основан на комбинировании слабых прогнозов (например, деревьев решений) для создания более сильной модели. Градиентный бустинг последовательно добавляет новые модели, корректируя ошибки предыдущих моделей.

Это только несколько примеров алгоритмов обучения с учителем, и в области Машинного обучения существует множество других алгоритмов и методов, которые можно применять в зависимости от конкретной задачи и типа данных.

Обучение без учителя: в этом подходе модель обучается на основе не размеченных данных, то есть данных без явно указанных выходных меток. Цель состоит в том, чтобы найти скрытые закономерности, структуры или группы в данных. Задачи кластеризации и понижения размерности являются примерами обучения без учителя. В этом случае модель сама находит внутренние структуры в данных, не требуя явных ответов. Целью обучения без учителя является нахождение скрытых закономерностей, структур или групп в данных.

Некоторые из примеров задач обучения без учителя:

1. Кластеризация: В задачах кластеризации модель группирует объекты по их сходству без заранее заданных классов или категорий. Это позволяет выявить внутренние структуры в данных и идентифицировать группы схожих объектов. Примером алгоритма для кластеризации является k-средних (k-means).

2. Понижение размерности: Задача понижения размерности состоит в сокращении размерности данных, сохраняя при этом важные информационные характеристики. Это полезно

для визуализации данных, удаления шума или избыточных признаков. Примерами алгоритмов понижения размерности являются метод главных компонент (PCA) и алгоритм t-SNE.

3. Ассоциативное правило: В этой задаче модель ищет статистические связи и ассоциации между различными элементами в наборе данных. Примером является алгоритм Apriori, который используется для нахождения часто встречающихся комбинаций элементов (таких как товары в корзине покупок).

Обучение без учителя полезно для обнаружения структур в данных и получения инсайтов о них, когда отсутствуют явные метки или целевые переменные. Этот подход позволяет модели самой извлекать информацию из данных и обнаруживать их скрытые характеристики.

1.5 Метрики и оценка производительности моделей

Оценка производительности моделей является важной частью процесса Машинного обучения. Для этого используются различные метрики, которые позволяют оценить, насколько хорошо модель справляется с поставленной задачей. Применение соответствующих метрик играет важную роль в измерении и сравнении производительности моделей. Вот более подробное описание некоторых метрик и методов оценки производительности:

1. В задачах классификации:

- Точность (accuracy): Измеряет долю правильно классифицированных объектов относительно общего числа объектов в выборке.
- Полнота (recall): Измеряет способность модели обнаруживать положительные случаи из общего числа положительных объектов.
- Точность (precision): Измеряет способность модели давать правильные положительные предсказания относительно всех положительных предсказаний.
- F-мера (F1 score): Комбинирует точность и полноту в одну метрику, представляющую сбалансированное среднее между ними.

2. В задачах регрессии:

- Средняя абсолютная ошибка (MAE): Измеряет среднее абсолютное отклонение между предсказанными и фактическими значениями.
- Средняя квадратичная ошибка (MSE): Измеряет среднее квадратичное отклонение между предсказанными и фактическими значениями.
- Коэффициент детерминации (R^2): Показывает, насколько хорошо модель объясняет изменчивость целевой переменной относительно базовой модели.

3. В задачах кластеризации:

- Коэффициент силуэта (silhouette coefficient): Измеряет степень разделения кластеров и их компактность на основе расстояний между объектами внутри кластера и между кластерами.
- Индекс Данна (Dunn index): Оценивает компактность и разделение кластеров на основе минимальных и максимальных расстояний между объектами.

4. Методы оценки производительности:

- Кросс-валидация (cross-validation): Позволяет оценить стабильность и обобщающую способность модели путем повторного разделения данных на обучающую и валидационную выборки.
- Разделение выборки на обучающую, валидационную и тестовую: Позволяет проверить производительность модели на новых, ранее не виденных данных, чтобы оценить ее способность к обобщению.

Выбор подходящих метрик и методов оценки производительности зависит от конкретной задачи и характеристик данных. Цель состоит в том, чтобы выбрать метрики, которые наилучшим образом отражают требуемые характеристики модели и задачи, и использовать соответствующие методы оценки для получения надежной оценки производительности модели.

Глава 2: Обучение с учителем

2.1 Линейная регрессия

Линейная регрессия – это один из основных методов Машинного обучения, используемый для предсказания непрерывной зависимой переменной на основе линейной комбинации независимых переменных. Она является простым и интерпретируемым алгоритмом.

В линейной регрессии предполагается, что существует линейная связь между независимыми и зависимой переменными. Модель линейной регрессии определяется уравнением:

y = b0 + b1*x1* + *b2*x2 + … + bn*xn

где y – зависимая переменная, x1, x2, …, xn – независимые переменные, b0, b1, b2, …, bn – коэффициенты модели, которые определяют веса, или важность, каждой независимой переменной.

Для оценки коэффициентов модели используется метод наименьших квадратов (МНК), который минимизирует сумму квадратов разностей между фактическими и предсказанными значениями зависимой переменной.

Линейная регрессия может быть однофакторной (с одной независимой переменной) или многофакторной (с несколькими независимыми переменными). Она может использоваться для прогнозирования значений на основе новых данных или для анализа влияния отдельных переменных на зависимую переменную. Кроме обычной линейной регрессии, существуют различные варианты этого метода, которые могут решать специфические задачи или учитывать особенности данных. Например, существуют регуляризованные модели линейной регрессии, такие как Ridge (гребневая регрессия) и Lasso (лассо-регрессия), которые добавляют штрафы к коэффициентам модели для борьбы с переобучением и улучшения обобщающей способности.

Линейная регрессия также может быть расширена для работы с нелинейными связями между переменными путем добавления полиномиальных или других нелинейных функций признаков. Это называется полиномиальной регрессией или нелинейной регрессией.

Одним из преимуществ линейной регрессии является ее простота и интерпретируемость. Коэффициенты модели позволяют оценить вклад каждой независимой переменной и понять, как они влияют на зависимую переменную. Кроме того, линейная регрессия требует меньше вычислительных ресурсов по сравнению с некоторыми более сложными моделями.

Однако линейная регрессия имеет свои ограничения. Она предполагает линейную связь между переменными, и если это предположение нарушено, модель может быть неправильной. Кроме того, она чувствительна к выбросам и может давать неверные предсказания в случае наличия значительных отклонений в данных.

2.2 Логистическая регрессия

Логистическая регрессия – это алгоритм классификации, используемый для прогнозирования вероятности принадлежности наблюдения к определенному классу. Она часто применяется в задачах бинарной классификации, где требуется разделить данные на два класса.

В логистической регрессии используется логистическая функция (сигмоид), которая преобразует линейную комбинацию независимых переменных в вероятность принадлежности к классу. Функция имеет следующий вид:

p = 1 / (1 + e^(-z))

где p – вероятность принадлежности к классу, z – линейная комбинация независимых переменных.

Модель логистической регрессии оценивает коэффициенты модели с использованием метода максимального правдоподобия. Она стремится максимизировать вероятность соответствия фактическим классам наблюдений.

Логистическая регрессия может быть расширена на многоклассовую классификацию с использованием подходов, таких как one-vs-rest или softmax. Логистическая регрессия является популярным алгоритмом классификации по нескольким причинам. Во-первых, она проста в понимании и реализации. Во-вторых, она обладает хорошей интерпретируемостью, поскольку коэффициенты модели позволяют определить вклад каждой независимой переменной в вероятность классификации. В-третьих, логистическая регрессия может обрабатывать как категориальные, так и числовые признаки, что делает ее гибкой для различных типов данных.

Однако следует отметить, что логистическая регрессия также имеет свои ограничения. Она предполагает линейную разделимость классов, что может быть недостаточным для сложных данных. Кроме того, она чувствительна к выбросам и может давать неверные предсказания, если данные имеют значительные отклонения или нарушают предположения модели.

В применении логистической регрессии важно учитывать также регуляризацию, чтобы справиться с проблемой переобучения и улучшить обобщающую способность модели. Регуляризация может быть выполнена с использованием L1-регуляризации (лассо) или L2-регуляризации (гребневая регрессия).

Логистическая регрессия может быть применена во многих областях, включая медицину, биологию, маркетинг, финансы и многие другие. Она может использоваться для прогнозирования вероятности наступления событий, определения рисков и принятия решений на основе классификации.

2.3 Метод k ближайших соседей

Метод k ближайших соседей (k-NN) – это алгоритм классификации и регрессии, основанный на принципе близости объектов. Он относит новое наблюдение к классу, основываясь на классификации его k ближайших соседей в пространстве признаков.

В алгоритме k-NN выбирается значение k – количество ближайших соседей, которые будут участвовать в принятии решения. Для классификации нового наблюдения происходит подсчет количества соседей в каждом классе, и наблюдение относится к классу с наибольшим числом соседей.

Для классификации с помощью метода k-NN необходимо выбрать значение k – количество ближайших соседей, которые будут участвовать в принятии решения. При поступлении нового наблюдения алгоритм вычисляет расстояние между ним и остальными объектами в обучающем наборе данных. Затем выбираются k объектов с наименьшими расстояниями, и их классы используются для определения класса нового наблюдения. Например, если большинство ближайших соседей относится к классу "A", то новое наблюдение будет отнесено к классу "A".

В задачах регрессии метод k-NN использует среднее или медианное значение целевой переменной y k ближайших соседей в качестве прогноза для нового наблюдения. Таким образом, предсказание для нового наблюдения вычисляется на основе значений его ближайших соседей.

Выбор метрики расстояния является важным аспектом в методе k-NN. Евклидово расстояние является наиболее распространенной метрикой, но также можно использовать и другие метрики, такие как манхэттенское расстояние или расстояние Минковского.

Одним из ограничений метода k-NN является его вычислительная сложность. При большом размере обучающего набора данных поиск ближайших соседей может быть времязатратным. Кроме того, метод k-NN чувствителен к масштабированию данных, поэтому рекомендуется нормализовать или стандартизировать признаки перед применением алгоритма.

Метод k-NN также имеет некоторые проблемы, связанные с выбросами и несбалансированными данными. Выбросы могут искажать результаты, особенно при использовании евклидова расстояния. Кроме того, если классы в обучающем наборе данных несбалансированы (то

есть один класс преобладает над другими), то может возникнуть проблема с предсказанием редкого класса.

В целом, метод k-NN представляет собой простой и гибкий алгоритм, который может быть эффективным во многих задачах классификации и регрессии. Однако для его успешного применения необходимо правильно выбрать значение k, подобрать подходящую метрику расстояния и учитывать особенности данных, такие как выбросы и несбалансированность классов.

2.4 Решающие деревья

Решающие деревья – это графические структуры, которые применяются для принятия решений в задачах классификации и регрессии. Они представляют собой одну из наиболее понятных и интерпретируемых моделей машинного обучения, что делает их популярным выбором во многих областях.

Конец ознакомительного фрагмента.