МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО»

Институт компьютерных наук и кибербезопасности

Кафедра: «Телематика (при ЦНИИ РТК)»

Направление: Математика и компьютерные науки

Отчет по дисциплине:

«Модуль мобильности»

«Основы нейроинформатики и машинного обучения»

Выполнил: Салимли Айзек. гр. 5130201/20102 _____

Руководитель: Курочкин Михаил Александрович _____

«____»_____ 20__ г.

Санкт-Петербург 2024

Содержание

Введение

В рамках модуля мобильности был выбран курс «Основы нейроинформатики и машинного обучения», так как направление данного курса, является одним из самых востребованных на рынке труда.

Автором курса, является ведущий специалист в области нейроинформатики и машинного обучения - Уткин Лев Владимирович.

Курс включает в себя 2 модуля и 4 содержательные темы. Все материалы курса доступны с момента открытия курса: видеолекции, кратко раскрывающие содержание каждой темы, презентации и конспекты, с которыми в дальнейшем можно ознакомиться в любое удобное время. Все темы включают практические занятия и самостоятельные работы. В материалах курса подготовлены методические рекомендации к выполнению заданий и примеры решения типовых заданий.

1 Постановка задачи

В рамках курса «Модуль мобильности», было необходимо пройти выбранный по желанию онлайн курс «Основы нейроинформатики и машинного обучения» на портале «Открытое образование» (https://openedu.ru/).

Онлайн-курс предполагает успешное освоение предлагаемых десяти лекций, написание контрольных заданий по лекциям и итогового теста.

Цель изучения дисциплины «Основы нейроинформатики и машинного обучения» заключается в освоении базовых принципов и методов анализа данных, разработки и применения интеллектуальных систем, основанных на моделях искусственного интеллекта.

2 Аннотация курса и разделов

В настоящее время нейроинформатика и машинное обучение развиваются как ключевые направления современной науки и технологий, объединяя достижения в области искусственного интеллекта, анализа данных и когнитивных систем. Эти дисциплины формируют основу для создания интеллектуальных систем, способных адаптироваться, обучаться и принимать решения на основе анализа больших объемов информации. Технологии машинного обучения находят применение в различных областях науки, техники и промышленности, предоставляя возможности для решения сложных задач, включая автоматизацию процессов, прогнозирование и разработку инновационных решений, что делает их особенно востребованными в современном мире.

3 Теоретическая часть курса

3.1.1 Базовые понятия и инструментов МО

1. Машинное обучение и его цель:
Машинное обучение (МО) — это наука о том, как научить компьютер решать задачи на основе данных. Его главная цель — создать алгоритм, который, обучаясь на примерах, сможет находить закономерности и применять их для прогнозирования или классификации новых данных.

2. Основные задачи МО:

- Обучение с учителем (Supervised Learning): предполагает наличие обучающих данных, включающих объекты и их известные ответы.
    - Классификация: определение категории объекта. Например, предсказание, сдадут ли студенты экзамен.
    - Регрессия: прогнозирование непрерывного значения, например, прибыли ресторана.
- Обучение без учителя (Unsupervised Learning): ответы неизвестны, задачи включают поиск закономерностей или структуры в данных.
    - Кластеризация: группировка объектов на основе их признаков (например, сегментация клиентов).
    - Оценивание плотности: приближение распределения данных, например, для обнаружения аномалий.
    - Понижение размерности: уменьшение количества признаков без значительной потери информации.

3. Признаки и их роль в МО:
Признаки — это характеристики объекта, которые используются для его описания. Их качество и выбор (feature engineering) определяют успех модели. Признаки бывают:

- Бинарные: принимают значения "да" или "нет".
- Вещественные: числа, например, стоимость квадратного метра.
- Категориальные: значения из множества (например, название города).
- Ординальные: значения с упорядочением (например, оценки по шкале).

4. Обучающая выборка:
Обучающая выборка включает пары: объект и его целевая переменная. Объекты описываются через признаки, образуя матрицу «объекты-признаки», где каждая строка соответствует одному объекту, а столбцы — характеристикам.

5. Инструменты и методы:

- Регрессия и классификация: базовые методы для прогнозирования вещественных значений и категорий.
- Глубинное обучение (Deep Learning): работа с большими и сложными данными, например, изображениями.
- Кластеризация и аномалия: обнаружение групп и отклонений в данных.

Машинное обучение предоставляет широкий инструментарий для анализа и автоматизации, делая акцент на обучении модели находить и использовать закономерности для решения задач.

3.1.2 Визуализация данных. Математические модели и методы. Обзор алгоритмов машинного обучения

1. Визуализация данных:

задача изображения многомерных объектов в двумерном или трехмерном пространстве таким образом, что сохранялось как можно больше зависимостей и отношений между ними.

Визуализация данных играет важную роль на начальном этапе анализа. Она позволяет лучше понять структуру данных, выявить зависимости, аномалии и закономерности. Основные методы визуализации включают:

- Гистограммы: для отображения распределения данных.
- Диаграммы рассеяния: для анализа зависимости между двумя или более переменными.
- Тепловые карты: для представления корреляций между признаками.
- Понижение размерности (например, PCA, t-SNE): для визуализации многомерных данных на плоскости или в 3D-пространстве.

Эти методы позволяют предварительно оценить качество данных и выбрать подходящие алгоритмы машинного обучения.

2. Математические модели и методы:
Математический фундамент машинного обучения строится на нескольких ключевых подходах:

- Линейные модели: простейшие модели, такие как линейная регрессия и логистическая регрессия, которые предполагают линейную связь между признаками и целевой переменной.
- Регуляризация: добавление штрафов (например, L1 или L2-регуляризация) для предотвращения переобучения.
- Оптимизация: методы градиентного спуска, стохастического градиентного спуска и их модификации для минимизации функций ошибки.
- Вероятностные модели: использование вероятностных подходов, например, наивного Байеса или скрытых марковских моделей, для задач классификации и предсказания.
- Глубинные нейронные сети: обучение сложных моделей с помощью нейронных сетей для работы с текстами, изображениями и аудиоданными.

3. Обзор алгоритмов машинного обучения: Алгоритмы машинного обучения делятся на несколько категорий в зависимости от типа задачи:

- Обучение с учителем:
    - Линейная регрессия: для прогнозирования вещественных значений.
    - Деревья решений: для классификации и регрессии с интерпретируемыми результатами.
    - Методы ансамблей (Random Forest, Gradient Boosting): для улучшения качества предсказания.

    - SVM (Метод опорных векторов): для задач классификации и регрессии, особенно при небольших объемах данных.
- Обучение без учителя:
    - Кластеризация (K-Means, DBSCAN): для разделения данных на группы.
    - Методы понижения размерности (PCA, t-SNE): для упрощения данных и визуализации.
    - Оценка плотности (Kernel Density Estimation): для поиска аномалий.
- Глубинное обучение:
    - Сверточные нейронные сети (CNN): для обработки изображений.
    - Рекуррентные нейронные сети (RNN): для работы с последовательностями, например, текстами или временными рядами.
    - Трансформеры: для анализа текстов и создания языковых моделей.

4. Актуальность методов:
Выбор алгоритма зависит от задачи и структуры данных. Например, простые методы, такие как линейная регрессия, эффективны для интерпретируемых решений, тогда как глубокие нейронные сети лучше справляются с обработкой сложных данных, таких как изображения и тексты. Современные подходы часто комбинируют разные алгоритмы для достижения наилучших результатов.

3.2.1 Алгоритмы распознания

Алгоритмы распознавания в машинном обучении применяются для анализа данных и классификации объектов на основе их признаков. Основные алгоритмы делятся на несколько категорий, каждая из которых решает специфические задачи:

1. Классификация

- Бинарная классификация ($Y \in \{0, 1\}$): Определение, принадлежит ли объект определённому классу. Примеры:
    - Предсказание, кликнет ли пользователь по рекламному объявлению.
    - Вернёт ли клиент кредит в срок.
    - Сдаст ли студент экзамен.
- Многоклассовая классификация ($Y \in \{1, ..., K\}$): Пример:
    - Определение предметной области научной статьи (например, математика, биология, психология).
- Многоклассовая классификация с пересекающимися классами (multi-label classification):
    - Пример: автоматическое проставление тегов для ресторанов, где объект может принадлежать сразу нескольким классам.

2. Регрессия

- Прогнозирование вещественного значения, например, предсказание прибыли ресторана в течение первого года работы.

3. Частичное обучение (semi-supervised learning):

- Используется, когда для части данных известны ответы, а для другой части — только признаки. Пример:
    - Медицинские задачи, где сбор данных требует дорогостоящих анализов.

4. Обучение без учителя

- Кластеризация: Группировка объектов на основе их сходства. Пример:
    - Кластеризация документов в электронной библиотеке.
    - Сегментация абонентов мобильного оператора.
- Оценивание плотности: Определение распределения данных. Пример:
    - Обнаружение аномалий (например, выявление некорректной работы оборудования).
- Понижение размерности: Генерация нового набора признаков с уменьшением их количества. Пример:
    - Построение тематических моделей и рекомендаций.

5. Разработка признаков (Feature Engineering):

- Один из ключевых этапов распознавания. Правильный выбор признаков (например, демографические данные, средняя стоимость квадратного метра в окрестности) сильно влияет на качество модели.

Алгоритмы для задач распознавания:

- Линейные модели: Простые предсказания на основе линейной комбинации признаков.
- Глубинное обучение (Deep Learning): Используется для сложных данных, таких как изображения или текстовые данные, с применением нейронных сетей.

Все эти методы позволяют решать задачи распознавания объектов, классификации, прогнозирования и анализа данных на основе признаков.

3.2.2 Методы обучения. Обучение с учителем, машинное обучение без учителя

1. Обучение с учителем (Supervised Learning):

Обучение с учителем предполагает наличие размеченного набора данных, где каждому объекту соответствует его целевая переменная (ответ). Этот подход используется для построения модели, способной предсказывать результаты на новых данных.

Ключевые этапы:

- Тренировочный набор (training set): используется для обучения модели. Он включает объекты и соответствующие ответы.
- Тестовый набор (test set): применяется для оценки точности обученной модели на новых данных.

Основные задачи:

- Классификация: отнесение объекта к одному из заданных классов. Примеры:
    - Разделение фотографий на категории: кошки, собаки, лошади.
    - Разметка слов по частям речи в предложении.

- Регрессия: прогнозирование непрерывного значения. Примеры:
    - Предсказание веса человека по его росту.

    - Прогнозирование цен на акции.

Пример: Обучение модели для анализа синтаксического дерева предложений на основе заранее размеченных данных. Предполагается, что законы формирования деревьев едины, иначе модель может работать некорректно (например, при попытке применить модель, обученную на английском языке, к немецкому).

Особые задачи:

- Обучение ранжированию (learning to rank): упорядочение объектов по целевой функции (например, ранжирование документов в поисковой системе).

2. Обучение без учителя (Unsupervised Learning):

Обучение без учителя применяется, когда в данных отсутствуют метки или ответы, и требуется выявить скрытые закономерности, группы или структуры.

Основные задачи:

- Кластеризация (Clustering): разбиение данных на группы (кластеры) по определённой мере похожести. Примеры:
    - Сегментация пользователей сайта.
    - Разделение генов на семейства.
- Снижение размерности (Dimensionality Reduction): уменьшение количества признаков при сохранении максимального объёма информации. Пример:
    - Представление текста меньшим числом параметров для облегчения обработки.
- Оценка плотности (Density Estimation): оценка распределения данных для определения закономерностей и аномалий. Пример:
    - Обнаружение некорректной работы оборудования.

Пример: Задача кластеризации медицинских снимков для автоматического выделения областей с опухолями.

3.3.1 Базовые понятия и определения

1. Биологические основы нейронных сетей

Нейронные сети — это концепция, вдохновленная устройством и функциями биологических нейронов. Каждый нейрон выполняет обработку информации и передачу сигналов другим нейронам через сложные сети синаптических соединений.

Основные элементы биологического нейрона:

- Сома (тело клетки): выполняет функции обработки сигналов.
- Дендриты: воспринимают входные сигналы от других нейронов.
- Аксон: передает обработанные сигналы другим нейронам.
- Синапсы: точки контакта между нейронами, где происходит передача сигналов.

Функционирование нейронов основано на электрической активности мембран, изменение которой происходит за счет ионных градиентов. Биологическая изменчивость и обучение нейронных сетей обеспечиваются пластичностью синаптических контактов, что позволяет адаптироваться к изменениям в окружающей среде.

2. Искусственный нейрон

Искусственный нейрон моделирует поведение биологического нейрона, выполняя преобразование входных сигналов в выходные с учетом весов и порогового значения.

Структура искусственного нейрона:

- Входы: принимают сигналы в виде числовых значений.
- Весовые коэффициенты: определяют значимость каждого входа.
- Сумматор: вычисляет взвешенную сумму входных сигналов:net

3. Архитектура нейронных сетей

Искусственные нейронные сети (ИНС) состоят из множества связанных нейронов, объединенных в слои:

- Входной слой: принимает данные для обработки.
- Скрытые слои: выполняют вычисления для извлечения сложных закономерностей.
- Выходной слой: формирует результат.

Классы нейронных сетей:

- Полносвязные сети: каждый нейрон соединен со всеми нейронами следующего слоя.
- Сверточные сети (CNN): используются для обработки изображений.
- Рекуррентные сети (RNN): применяются для последовательных данных (текст, временные ряды).

4. Обучение нейронных сетей

Обучение заключается в настройке весов нейронов для достижения оптимального соответствия между входными данными и целевыми выходами. Основной метод обучения — градиентный спуск, который минимизирует функцию потерь.

Этапы обучения:

1. Прямой проход: вычисление выходов сети.
2. Обратное распространение ошибки: расчет градиентов и обновление весов с использованием метода оптимизации.

5. Применение нейронных сетей

Нейронные сети используются для решения различных задач:

- Классификация: отнесение объектов к определенным категориям (например, распознавание изображений).
- Регрессия: прогнозирование числовых значений (например, предсказание цен).
- Кластеризация: объединение объектов в группы.
- Снижение размерности: упрощение данных без потери значимой информации.

Нейронные сети позволяют обрабатывать сложные, плохо структурированные данные, такие как изображения, текст и речь, что делает их важным инструментом в современных технологиях.

3.3.2 Базовые архитектуры нейронных сетей

Свёрточные нейронные сети (Convolutional Neural Networks, CNN) — это специализированные архитектуры, которые отличаются от других типов сетей использованием свёрточных слоёв. Они предназначены для обработки двумерных данных, таких как изображения, и в некоторых случаях — аудио.

Основные элементы CNN:

1. Свёрточные слои:
   Эти слои используют фильтры для выделения локальных признаков, таких как границы объектов, текстуры и формы.
2. Пулинговые слои (Pooling):
   Эти слои уменьшают размер данных и извлекают наиболее важные признаки. Примером является максимальный пулинг (max pooling), который выбирает максимальное значение из каждого блока входных данных.
3. Полносвязные слои:
   На заключительных этапах CNN используется полносвязная нейронная сеть (FFNN) для окончательной классификации.

Особенности CNN:

- Свёрточные сети используют небольшой фильтр для всего изображения, что сокращает количество параметров и предотвращает переобучение.
- Они сжимаются с глубиной сети, например: 32, 16, 8, 4, 2, 1.

2. Глубинные свёрточные сети (DCNN)

Глубинные свёрточные сети (Deep Convolutional Neural Networks, DCNN) представляют собой усовершенствованную версию CNN, включающую множество слоёв.

Особенности DCNN:

- Содержат десятки и сотни слоёв для более глубокого анализа изображений.
- Используются в задачах высокой сложности, таких как классификация объектов на больших наборах данных.

Transfer learning (передача обучения):

При недостатке данных для обучения DCNN применяется подход transfer learning:

1. Используется уже обученный граф, например VGG-19.
2. Извлекаются выходы одного из последних слоёв.
3. Эти выходы используются как признаки для стандартных алгоритмов машинного обучения.

3. Развёртывающие нейронные сети (Deconvolutional Networks, DN)

Развёртывающие сети, также известные как обратные графические сети, выполняют обратную задачу свёрточных сетей: генерируют данные.

Особенности DN:

- На вход подаётся бинарный вектор (например, [0,1][0,1] для кошки), а сеть генерирует изображения.
- DN часто комбинируют с полносвязными сетями (FFNN).

4. Использование в задачах компьютерного зрения

- CNN и DCNN идеально подходят для задач классификации изображений (например, «кошка» или «собака»).
- Они анализируют данные с использованием «сканера», который проходит по изображениям блоками (например, 20×20) и вычисляет признаки на каждом шаге.

Пример стандартной архитектуры CNN

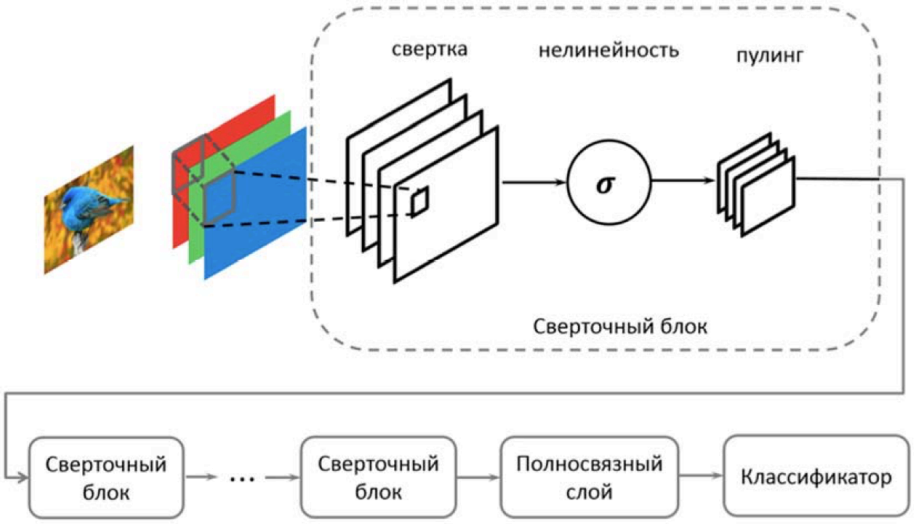

Рис.1 Стандартная схема свёрточной сети

Свёрточные и развёртывающие нейронные сети являются фундаментальными инструментами в задачах компьютерного зрения и обработки изображений, их использование продолжает расширяться благодаря высокой эффективности в задачах классификации, генерации и анализа данных.

3.3.3 Алгоритмы машинного обучения

Классификация алгоритмов

1. Алгоритмы обучения с учителем:
    - Используются для прогнозирования или классификации.
    - Подразумевают наличие размеченных данных: каждый объект данных сопровождается меткой, которая показывает, к какому классу он относится или какое значение нужно предсказать.

2. Алгоритмы обучения без учителя:

    - Применяются, когда данные не размечены.
    - Задача: выявить скрытые структуры, объединить данные в кластеры или снизить их размерность.

Обучение с учителем

- Регрессия:
    - Прогнозирование непрерывных значений. Например, предсказание цен на жильё.
    - Основная цель — минимизировать разницу между реальными и предсказанными значениями.
- Классификация:
    - Задача — разделить данные на конечное число классов. Пример: распознавание текста или изображения.

Обучение без учителя

1. Кластеризация:

    - Группировка данных по их схожести. Пример: сегментация клиентов.
    - Основной метод — нахождение центров кластеров и отнесение объектов к ближайшим.
2. Снижение размерности:

    - Уменьшение числа признаков в данных при сохранении их структуры.
    - Пример: Principal Component Analysis (PCA).
3. Оценка плотности:

    - Выявление распределения данных.
    - Задача: оценить, как вероятностное распределение описывает данные.

Оценка качества модели

1. Матрица ошибок (Confusion Matrix):

    - Используется для анализа качества классификации.
      Пример:
        - True Positive (TP): верно предсказанные положительные объекты.
        - True Negative (TN): верно предсказанные отрицательные объекты.
        - False Positive (FP): ошибочно предсказанные положительные.
        - False Negative (FN): ошибочно предсказанные отрицательные.
2. Метрики качества:

    - Accuracy:
      accuracy = (TP + TN) / (TP + TN + FP + FN).
    - Precision:
      precision = TP / (TP + FP).
    - Recall:
      recall = TP / (TP + FN).

- F-мера:
  F1 = 2 * (precision * recall) / (precision + recall).

3. AUC-ROC:
   - Площадь под кривой ошибок, показывает качество классификатора при разных порогах.

3.4.1 Постановка задачи объяснительного интеллекта

Зачем нужна интерпретация?

Интерпретация результатов моделей машинного обучения необходима по нескольким ключевым причинам:

1. Объяснение результатов:
   - Основная цель интерпретации — понять, почему модель дала конкретный ответ. Это особенно важно в критически важных областях, таких как медицина, финансы, транспорт и юриспруденция.
   - Например, система, выдающая диагноз, должна обосновать, какие симптомы или данные пациента привели к такому выводу.
2. Улучшение качества:
   - Анализ интерпретаций позволяет выявить недостатки модели и улучшить её. Например, оценка важности признаков помогает оптимизировать признаки, использованные для обучения.
   - Некоторые методы интерпретации, такие как оценка влияния признаков, непосредственно используются для создания новых признаков или выбора важных переменных.
3. Понимание данных:
   - Интерпретация модели помогает понять, как данные представлены и обрабатываются в модели. Хотя нужно помнить, что интерпретируется модель, а не сами данные, и модель может содержать погрешности.
4. Проверка:
   - Перед внедрением модели в реальные процессы важно убедиться, что она работает корректно и не содержит скрытых ошибок. Это особенно актуально для сложных и критичных задач.

Типы интерпретации:

1. Визуализация:
   - Примеры:
     - Области внимания модели: В связке CNN+RNN можно визуализировать, на какие части изображения модель обращала внимание. Например, при выводе слова «птица» модель анализирует область, где изображена птица, а при выводе слова «вода» — область с водой.

        - Проекции: Примеры пространств кодировок слов (например, word2vec), где можно увидеть закономерности, такие как различия между столицами и странами.
2. Числовые показатели и таблицы:
    - Выражают важность признаков или другие метрики, например, значения весов модели для конкретных признаков.
3. Аналитические формулы:
        - Коэффициенты модели указывают, как каждый признак влияет на результат.
        - Например, увеличение площади на 1 кв.м. добавляет 50,000 руб. к стоимости квартиры.
4. Объекты и признаки:
    - Пример: визуализация, какая часть объекта (например, изображения) максимально повлияла на вывод модели.
5. Глобальная и локальная интерпретация:
    - Глобальная: объясняет общие закономерности работы модели.
    - Локальная: анализирует причину конкретного вывода модели, объясняя, как она пришла к этому результату.

Требования к интерпретациям:

1. Сравнение (contrastive):
    - Важно объяснять, почему модель выбрала один вариант, а не другой. Например, в случае отказа в кредите клиенту важно знать не только причину отказа, но и что нужно изменить, чтобы получить одобрение.
2. Краткость и конкретика (selectivity):
    - Интерпретация должна быть компактной и фокусироваться на нескольких ключевых факторах (1-3 причины), вместо того чтобы перечислять десятки малозначимых факторов.
3. Контентность:
    - Интерпретация должна быть адаптирована к аудитории. Например, объяснения должны быть понятны человеку без технического образования.
4. Соответствие ожиданиям и правдивость:
    - Объяснение должно быть логичным. Например, странно утверждать, что кредит не выдан из-за слишком высокого дохода, хотя формально модели (например, SVM) могут давать такие результаты из-за особенностей своих разделяющих поверхностей.

3.4.2 Элементы объяснительного интеллекта

1. Простые интерпретируемые модели

- Простые модели могут использоваться для объяснения поведения сложных чёрных ящиков.

- Для обучения таких моделей можно собрать выборку данных, используя ответы чёрного ящика на случайных объектах.
- Проблема: простая модель может плохо моделировать поведение сложной на всём пространстве данных.
- Пример: Для нейросетей учитываются не только предсказанные классы, но и распределения вероятностей (темное знание, Dark Knowledge).

2. Локальные суррогатные модели (Local Surrogate Models)

- Принцип: объяснение конкретного ответа чёрного ящика с помощью линейной модели, настроенной на данных, сгенерированных в окрестности точки.
- Алгоритм:
  1. Получить ответ чёрного ящика в выбранной точке.
  2. Сгенерировать дополнительные данные вокруг этой точки.
  3. Узнать ответы чёрного ящика на этих данных.
  4. Построить линейный классификатор, объясняющий поведение чёрного ящика в данной окрестности.
- Пример: LIME (Local Interpretable Model-agnostic) используется для выделения суперпикселей, ответственных за вероятности классов при классификации изображений.

3. Исследование отдельных блоков модели

- Если сложная модель состоит из модулей или блоков, то можно интерпретировать их по отдельности.
- В нейронных сетях исследуются:
  - Какие входы вызывают максимальную активацию конкретного нейрона, слоя или канала.
  - Генерируются изображения, усиливающие активацию, с использованием регуляризации:
    - Соседние пиксели генерируемого изображения должны быть похожи.
    - Изображение периодически размывается.
    - Проверяется устойчивость изображения к трансформациям.
    - Вводится априорное распределение для генерируемых изображений.

Пример на изображении:

- Результаты анализа суперпикселей:
  - Electric Guitar: $p = 0.32$.
  - Acoustic Guitar: $p = 0.24$.
  - Labrador: $p = 0.21$.
- Это позволяет визуализировать, какие области изображения наиболее значимы для классификации.

Достоинства подходов:

- Локальные модели позволяют объяснить поведение сложных чёрных ящиков в конкретных областях данных.
- Анализ отдельных блоков помогает понять, как сложная модель обрабатывает входные данные.

Недостатки:

- Простая модель может не подходить для глобального объяснения.
- Локальные суррогаты ограничены своей окрестностью и не могут объяснить модель целиком.

3.4.3 Элементы объяснительного интеллекта

1. Анализ частичной зависимости (Partial Dependence):
Чёрный ящик зависит от n признаков. Чтобы исследовать зависимость от конкретного признака, нужно проинтегрировать по остальным. Формула:
PD(X) = sum(f(X1, ..., Xj-1, X, Xj+1, ..., Xn)) / m
Для двух признаков часто используют H-статистику:
H = (PD(X1, X2) - PD(X1) - PD(X2)) / PD(X1, X2)
Эти зависимости визуализируются на Partial Dependence Plot (PDP). H-статистика позволяет оценить взаимодействие признаков в модели, но вычисляется достаточно долго.

2. Индивидуальное условное ожидание (Individual Conditional Expectation, ICE):
Частичная зависимость показывает усреднённые изменения модели, а ICE отображает, как меняется ответ модели на каждом объекте при варьировании конкретного признака. Для одного объекта:
ICE(Xi) = f(X1, ..., Xi-1, x, Xi+1, ..., Xn)
Среднее арифметическое ICE по всем объектам даёт PD. Графики ICE позволяют исследовать вариативность модели на конкретных объектах.

3. Важности признаков (Feature Importance):
Оценка важности признаков показывает, насколько модель зависит от каждого признака. Простейший метод:

   - Перемешать значения признака (рандомная перестановка столбца).
   - Проверить, как изменится качество модели.

4. SHAP (SHapley Additive exPlanations) — согласованный метод оценки важности:
SHAP(i) = sum((f(S ∪ {i}) - f(S)) * (|S|!(n - |S| - 1)!)) / n!
Где S — подмножество признаков, f(S) — результат модели на этих признаках.
Этот метод требует переобучения модели на всевозможных подмножествах признаков, поэтому на практике используют приближения, например, метод Монте-Карло.

4. Глобальные суррогатные модели (Global Surrogate Models):
Строится простая интерпретируемая модель, которая моделирует поведение чёрного ящика. Для её обучения можно использовать ответы чёрного ящика на случайных объектах.
Ограничение: простая модель может не охватить поведение сложной во всём пространстве. Для улучшения используют вероятности, предсказанные сетью (тёмное знание, Dark Knowledge).

5. Локальные суррогатные модели (Local Surrogate Models):
Простая модель строится только в окрестности конкретной точки. Формула для локальной интерпретации:
LIME = sum(f(X) - f(X')) / m
Этот метод применяется, чтобы объяснить конкретный ответ чёрного ящика.

6. Поиск конфликтных примеров (Counterfactual Examples):
Конфликтные примеры — объекты, которые отличаются незначительно, но ответ модели на них отличается существенно. Формула оптимизации:
$\lambda a(x) + y + d(x, x_0) \rightarrow \min$
Здесь $d(x, x_0)$ — расстояние между объектами, а y — целевой результат.

7. Влиятельные объекты (Influential Instances):
Это объекты, от которых сильно зависят параметры модели. Например, для SVM это опорные объекты. Удаление влиятельного объекта может существенно изменить модель.
Простейший метод поиска:

   - Удаление объекта из выборки и проверка изменения модели.

8. Прототипы и критика (Prototypes and Criticisms):

   - Прототипы: типичные объекты выборки, которые хорошо описывают её структуру.
   - Критика: объекты, которые сильно отличаются от прототипов или выявляют проблемы модели.

Эти элементы используются для анализа данных и оценки поведения модели на нетипичных объектах.

4 Результаты аттестации по модулям

Результаты прохождения аттестации по модулям 1 и 2 представлены на Рис. 2-7.

Тема 1. Введение в машинное обучение	Статус прокторинга	Оценка
Практическое занятие 1 Оценки по заданиям: 1/1 0/1	Без прокторинга	1/2
Амнистия. Практическое занятие 1 Оценки по заданиям: 0/1 1/1	Без прокторинга	1/2
Контрольный тест по теме 1 Оценки по заданиям: 0/1 0/1 1/1 0/1 1/1 1/1 1/1 1/1 1/1 1/1	Без прокторинга	7/10
Амнистия. Контрольный тест по теме 1	Без прокторинга	0/10

Рис. 2 Тест по теме №1

Тема 2. Методы машинного обучения	Статус прокторинга	Оценка
Контрольный тест по теме 2 Оценки по заданиям: 0/1 1/1 1/1 1/1 1/1 1/1 1/1 1/1 1/1 1/1	Без прокторинга	9/10
Амнистия. Контрольный тест по теме 2	Без прокторинга	0/10

Рис. 3 Тест по теме №2

Тема 3. Введение в нейронные сети	Статус прокторинга	Оценка
Практическое занятие 2	Без прокторинга	0/2
Амнистия. Практическое занятие 2 Оценки по заданиям: 1/1 0/1	Без прокторинга	1/2
Практическое занятие 3 Оценки по заданиям: 1/1 1/1	Без прокторинга	2/2
Амнистия. Практическое занятие 3 Оценки по заданиям: 0/1 0/1	Без прокторинга	0/2
Контрольный тест по теме 3 Оценки по заданиям: 1/1 1/1 1/1 1/1 1/1 1/1 1/1 1/1 0/1 0/1	Без прокторинга	8/10
Амнистия. Контрольный тест по теме 3	Без прокторинга	0/10

Рис. 4 Тест по теме №3

Тема 4. Модели знаний и элементы объяснительного интеллекта	Статус прокторинга	Оценка
▲ Контрольный тест по теме 4 Оценки по заданиям: 1/1 1/1 1/1 1/1 1/1 0/1 1/1 0/1 1/1 0/1	Без прокторинга	7/10
▼ Амнистия. Контрольный тест по теме 4	Без прокторинга	0/10

Рис. 5 Тест по теме №4

Итоговый тест с ограничением по времени (без сертификата)	Статус прокторинга	Оценка
▲ Итоговое тестирование Оценки по заданиям: 0/1 1/1 1/1 1/1 0/1 1/1 1/1 1/1 1/1 1/1 1/1 0/1 1/1 0/1 1/1 1/1 1/1 0/1 1/1 1/1 1/1 1/1 0/1 1/1 0/1 1/1 1/1 1/1 0/1 1/1 0/1 1/1 0/1 1/1 1/1 1/1 1/1 1/1 1/1 1/1	Без прокторинга	30/40

Рис. 6 Итоговый тест

Прогресс

salimliam salimli.am@edu.spbstu.ru

Оценки

Ваша оценка по сравнению с оценкой, необходимой для прохождения этого курса.

Рис. 7 Итоговый прогресс

Заключение

Прохождение онлайн-курса «Основы нейроинформатики и машинного обучения», автором которого является Уткин Лев Владимирович, позволил ознакомиться с фундаментальными концепциями и принципами современных методов машинного обучения и нейроинформатики. Курс охватывал широкий спектр тем, включая основы работы искусственных нейронных сетей, применение методов машинного обучения для обработки данных, а также их интеграцию в сложные информационно-управляющие системы. Одной из ключевых особенностей курса стала связь изучаемых технологий с актуальными тенденциями развития индустрии.

Курс предоставил глубокое понимание роли интеллектуальных алгоритмов в решении задач с применением машинного обучения и управления в условиях неопределенности.

Подводя итоги, хочется отметить, что использование дистанционных образовательных технологий, на которых базировался данный курс, представляет собой важное дополнение к традиционным методам обучения. Однако, несмотря на очевидные преимущества онлайн-обучения, такие как гибкость и доступность, личное общение с преподавателем и участие в практических занятиях остаются незаменимыми для полноценного усвоения материала. Онлайн-курсы, подобные этому, играют важную роль в расширении образовательных возможностей и развитии самостоятельного обучения. Они отлично дополняют основное образование, предоставляя удобные инструменты для освоения сложных концепций и навыков, востребованных в современных технологиях.

Список источников

[1] Основы нейроинформатики и машинного обучения. Открытое образование: URL: https://apps.openedu.ru/learning/course/course-v1:spbstu+NEUROINF+fall_2024/progress/.- (Дата обращения: 10.12.2024 г.)