

Основы технологий искусственного интеллекта 2.0 (AIF)

ID OT-AIF Цена 120 000,– руб. Длительность 5 дней

Кому следует посетить

- Начинающие специалисты в области ИИ, которые хотят получить базовые знания и навыки.
- Программисты и разработчики, желающие перейти в область ИИ и машинного обучения.
- Менеджеры и руководители проектов, заинтересованные в использовании ИИ для улучшения бизнес-процессов.
- Студенты и исследователи, стремящиеся углубить свои знания в области ИИ.

Предварительные требования

- Знания в объеме курса [Программирование на языке Python. Базовый уровень \(PYTHON_BASIC\)](#)
- Понимание основных математических и статистических концепций.

Цели курса

- Ознакомить участников с основами искусственного интеллекта, его ключевыми концепциями и областями применения.
- Предоставить практические навыки в области машинного обучения и глубокого обучения.
- Научить использовать современные инструменты и библиотеки для разработки и внедрения ИИ-моделей.
- Показать успешные примеры применения ИИ в различных отраслях.
- Развить навыки оптимизации и настройки параметров ИИ-моделей для повышения их эффективности.

Содержание курса

В ходе данного курса участники пройдут от основ искусственного интеллекта к конкретному применению технологий машинного обучения и глубокого обучения в практических сценариях. Внимание уделяется не только теоретическим аспектам, но и развитию практических навыков в создании моделей. Слушатели приобретут опыт работы с инструментами и освоят методы эффективной оптимизации и настройки параметров моделей с целью достижения высокой эффективности в реальных проектах в

области искусственного интеллекта.

Программа курса

День 1: Введение в искусственный интеллект

Основы искусственного интеллекта

- Определение ИИ и его происхождение
- История развития ИИ
- Важные вехи и прорывы в области ИИ
- Основные ученые и их вклад в ИИ
- Основные понятия: машинное обучение, глубокое обучение, нейронные сети
- Различия между ИИ, машинным обучением и глубоким обучением
- Что такое нейронные сети и как они работают
- Примеры нейронных сетей: полносвязные, свёрточные, рекуррентные

Применения искусственного интеллекта

- Обзор областей применения ИИ
- Промышленность: автоматизация производства, предиктивное обслуживание
- Медицина: диагностика заболеваний, персонализированное лечение
- Финансовый сектор: управление рисками, торговые стратегии
- Развлечения: рекомендательные системы, создание контента
- Транспорт: автономные транспортные средства, оптимизация маршрутов
- Успешные примеры внедрения ИИ
- Case study 1
- Case study 2
- Case study 3

Установка и использование инструментов

- Установка и настройка необходимых инструментов
- Обзор популярных инструментов и библиотек для ИИ (Python, TensorFlow, PyTorch, Scikit-learn)
- Установка Python и Jupyter Notebook
- Установка библиотек через pip

Практическое задание

- Настройка рабочего окружения
- Написание первой программы на Python, использующей библиотеку для машинного обучения

День 2: Основы машинного обучения

Введение в машинное обучение

- Что такое машинное обучение и как оно отличается от традиционного программирования
- Основные методы машинного обучения: supervised learning, unsupervised learning, reinforcement learning

Роль данных в машинном обучении

- Важность качества данных
- Методы сбора и подготовки данных
- Понятие "обучающая выборка" и "тестовая выборка"

Обучение с учителем и без учителя

- Различия между обучением с учителем и без учителя
- Основные концепции и примеры supervised learning и unsupervised learning
- Преимущества и недостатки каждого подхода

Примеры задач: классификация, регрессия, кластеризация

- Классификация: распознавание изображений, спам-фильтры
- Регрессия: предсказание цен на жилье, прогнозирование продаж
- Кластеризация: сегментация клиентов, обнаружение аномалий

Практическое задание: создание простой модели машинного обучения на основе данных

- Сбор и подготовка данных
- Загрузка датасета (например, из библиотеки Scikit-learn или Kaggle)
- Очистка и предобработка данных
- Разработка модели
- Выбор алгоритма (например, линейная регрессия, k-ближайших соседей)
- Обучение модели на тренировочных данных
- Оценка модели
- Разделение данных на тренировочную и тестовую выборки
- Оценка точности модели на тестовых данных
- Визуализация и интерпретация результатов
- Построение графиков и визуализация результатов
- Интерпретация результатов и выводы

День 3: Глубокое обучение и нейронные сети

Основы глубокого обучения

- Архитектура нейронных сетей
- Что такое нейронная сеть: основные компоненты (нейроны, слои, связи)
- Типы слоев: входной, скрытые, выходной
- Популярные архитектуры: feedforward, свёрточные нейронные сети (CNN), рекуррентные нейронные сети (RNN)

Обучение глубоких нейронных сетей

- Процесс обучения: функции потерь, градиентный спуск
- Переобучение и способы его предотвращения (регуляризация, dropout)
- Валидация и тестирование моделей

Применение глубокого обучения

- Обзор успешных проектов с использованием глубокого обучения
- GPT-4 и другие модели обработки естественного языка
- Проекты в области компьютерного зрения (например, распознавание лиц, сегментация объектов)

Тенденции в развитии глубокого обучения

- Новые архитектуры и методы (Transformers, Attention Mechanisms)
- Автоматизация машинного обучения (AutoML)

День 4: Техническое применение ИИ и большие языковые модели (LLM)

Инструменты для разработки ИИ-приложений

- Популярные библиотеки: TensorFlow, PyTorch, Keras
- Особенности и различия между ними
- Дополнительные инструменты: NumPy, Pandas, Scikit-learn

Инструменты для разработки моделей и их внедрения

- Jupyter Notebook для разработки и тестирования
- Облачные сервисы для обучения и развертывания моделей (DataSphere, Google Colab, Azure ML)
- Непрерывная разработка и интеграция ИИ-моделей
- Практика MLOps: CI/CD для ИИ-моделей

Большие языковые модели (LLM)

- Что такое LLM и как они работают
- Архитектура Transformers и Attention
- Области применения: чат-боты, генерация текста, резюмирование
- Примеры: GPT, BERT, Claude, LLaMA

Практическое задание: создание простой модели с использованием TensorFlow или PyTorch

- Выбор библиотеки (TensorFlow или PyTorch)
- Построение и обучение простой модели (например, MNIST классификация)
- Визуализация результатов и оценка точности модели

День 5: Оптимизация моделей и генерация с поддержкой извлечения (RAG)

Техники оптимизации и настройки параметров моделей

- Настройка гиперпараметров: выбор оптимальных параметров модели
- Методы оптимизации: Adam, RMSProp, SGD
- Использование кросс-валидации для оценки моделей

Retrieval-Augmented Generation (RAG)

- Что такое RAG и зачем он нужен
- Принцип работы: соединение генерации (LLM) с поиском в документах
- Компоненты RAG-системы
- Векторные базы: FAISS, Chroma, Pinecone
- Интеграция с LangChain, LlamaIndex
- Примеры применения: интеллектуальные ассистенты, чат с базой знаний, документы компании

Практическое задание: оптимизация модели и создание простого RAG-приложения

- Применение техник настройки гиперпараметров
- Повторное обучение модели с новыми параметрами
- Сравнение производительности оригинальной и оптимизированной моделей
- Создание простого RAG-примера
- Индексация текста
- Генерация ответа на основе извлечённой информации
- Запуск в Jupyter Notebook или через Streamlit