

28. НЕЙРОСЕТИ: ЗАМЕНА ПРОГРАММИСТОВ ИЛИ ОЧЕРЕДНОЙ ИНСТРУМЕНТ?

Вихотенко Ф.Е., Гинько В.А.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Шкор О.Н. – старший преподаватель

Аннотация. С появлением нейронных сетей в информационных технологиях возникли дебаты о их роли в современном обществе. Вопрос о том, представляют ли нейросети замену для программистов или они являются лишь еще одним инструментом в арсенале разработчиков, стал объектом широкого обсуждения. Эта статья предназначена для анализа этой проблемы и выявления ключевых аспектов, определяющих роль нейронных сетей в современной информационной среде.

Ключевые слова: нейросети, информационные технологии.

В последние десятилетия нейросети и искусственный интеллект в целом стали все более присутствовать в различных сферах человеческой деятельности, привнося в них инновации и изменения. Одним из обсуждаемых вопросов стало возможное влияние нейросетей на роль программистов и будущее профессии разработчика программного обеспечения. Обсуждение о замене программистов нейросетями привлекает внимание как специалистов в области информационных технологий, так и широкой общественности, вызывая разнообразные мнения и предположения о перспективах и последствиях этого развития.

В мире программирования существует два вида помощи от нейросетей программистам: одна предоставляется генераторами текста, такими как ChatGPT, а другая встраивается непосредственно в редакторы кода, например, GitHub Copilot и Codeium.

Первым рассмотрим ChatGPT.

Сценарии использования ChatGPT:

генерация кода по запросу. может генерировать код на различных языках программирования, помогая разработчикам автоматизировать задачи, создавать прототипы приложений. Благодаря своим способностям к анализу и синтезу информации, ChatGPT может создавать функции, классы, алгоритмы и многое другое, облегчая процесс разработки программного обеспечения.

возможность писать тесты для своих программ с помощью ChatGPT. Он способен генерировать тестовые случаи для программ, обеспечивая покрытие различных сценариев и проверку функциональности. Это помогает разработчикам автоматизировать процесс тестирования и обнаруживать потенциальные ошибки в программном коде.

возможность упрощать код и делать его более эффективным, предоставляя разработчикам советы по оптимизации и рефакторингу. Этот инструмент может предложить альтернативные способы написания кода, устранение избыточных операций и улучшение алгоритмов. Благодаря этим возможностям, разработчики могут создавать более чистый, читаемый и производительный код.

способность искать ошибки в коде программы. ChatGPT обладает способностью анализировать код на различных языках программирования и выявлять потенциальные ошибки или недочеты, помогая программистам повысить качество своих проектов. С его помощью можно быстро обнаруживать синтаксические ошибки, опечатки, проблемы в логике кода и другие потенциально проблемные моменты, что способствует эффективной отладке и улучшению стабильности программного обеспечения.

Хотя ChatGPT обладает впечатляющими способностями в обработке любого языка программирования и может помочь в некоторых аспектах программирования, у него также есть некоторые недостатки.

Иногда он сталкивается с трудностями из-за отсутствия контекста программы, что может затруднить его способность эффективно найти ошибку или сгенерировать подходящий код. Без полного понимания задачи и контекста работы программы, его рекомендации могут быть неполными или неоптимальными. Также он сильно зависит от данных, на которых обучался, что может приводить к ошибкам в сгенерированном коде. Если модель не была обучена на достаточном разнообразии программистских стилей или конкретных языках программирования, её предложения могут быть несоответствующими или неправильными для конкретного контекста.

ChatGPT, как и другие модели генерации текста, иногда может случайно создавать небезопасный код с уязвимостями, такими как SQL-инъекции, XSS-атаки или утечки данных. Это может происходить из-за недостаточного понимания моделью контекста безопасности программирования или из-за включения в обучающие данные фрагментов кода с уязвимостями.

Некоторые из этих проблем были решены во встроенных в редактор кода нейросетях, таких как GitHub Copilot.

Разработки, такие как GitHub Copilot и Codeium, поднимают вопросы не только о будущем программирования, но и о будущей роли программистов в целом. Действительно ли нейросети могут заменить программистов, или же они лишь еще один инструмент в их арсенале?

GitHub Copilot, разработанный GitHub в сотрудничестве с OpenAI, использует мощные алгоритмы искусственного интеллекта для предложения кода в реальном времени, основываясь на комментариях и коде, который вы пишете. Этот инструмент обучался на миллиардах строк кода, доступных в открытом доступе, что позволяет ему предложить решения для широкого спектра задач программирования, от простых функций до более сложных алгоритмов. Copilot может значительно ускорить процесс разработки, предлагая код, который программисты могут использовать как есть или адаптировать под свои нужды. Он также может служить обучающим инструментом, показывая менее опытным разработчикам, как можно решать те или иные задачи. Примеры использования:

Автозаполнение кода: например, когда программист начинает писать функцию на JavaScript для фильтрации массива, Copilot может предложить полное тело функции, основываясь на нескольких введенных словах.

Генерация тестов: пользователи могут комментировать, что они хотят достичь с помощью теста, и Copilot генерирует соответствующие тестовые случаи.

Обучение через программирование: новички могут изучать практики и методологии программирования через предложения Copilot, которые охватывают широкий спектр технологий и языков программирования.

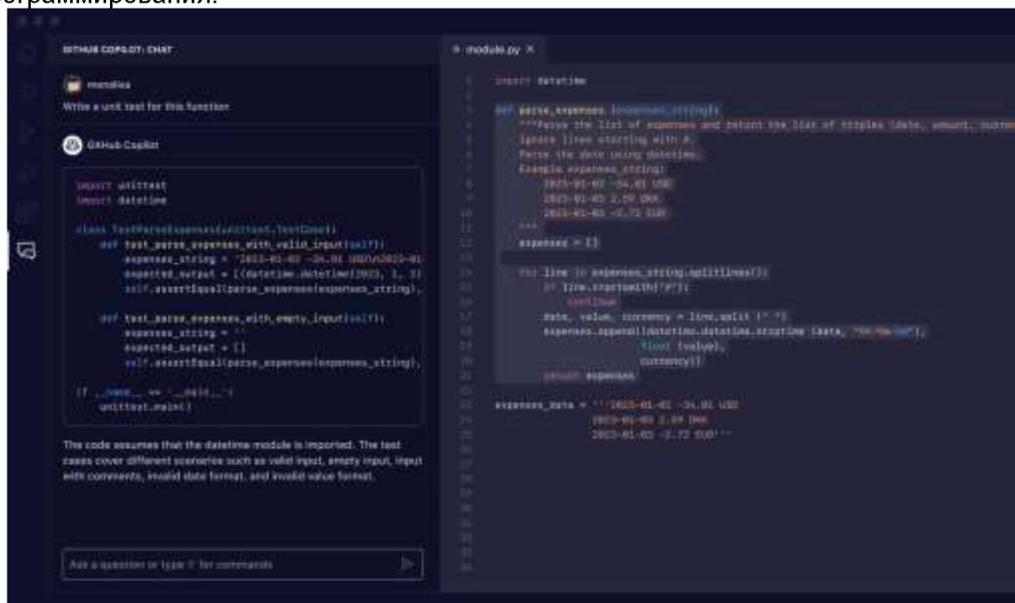


Рисунок 1 - Использование GitHub Copilot в редакторе кода VS Code

Codeium — еще один пример применения ИИ в программировании, который направлен на упрощение процесса разработки за счет автоматизации написания кода. Этот инструмент стремится минимизировать рутинную и повторяющуюся работу, позволяя разработчикам сосредоточиться на более сложных и творческих аспектах проектов.

В отличие от Copilot, который предлагает код на основе уже написанного программистом, Codeium может генерировать рекомендации на основе более абстрактных запросов, делая его более гибким в некоторых сценариях использования. Это открывает новые возможности для ускорения разработки и повышения ее качества.

Разработчики могут описать функциональность, которую они хотят реализовать, и Codeium предложит одно или несколько возможных решений. Например, если вам нужно создать функцию для парсинга даты из строки в формате ISO в объект DateTime в Python, Codeium может сгенерировать соответствующий код.

Представьте, что у вас есть кусок кода, который работает, но вы хотите его оптимизировать или привести к соответствию с новыми стандартами кодирования. Codeium может предложить альтернативные варианты реализации. Также благодаря обширной базе данных и обучающим материалам, Codeium способен предлагать решения для широкого диапазона технологий, от вебразработки до машинного обучения, включая поддержку множества языков программирования.

```
test.py
1  # Nth Fibonacci number
2  def fibonacci(n):
    if n < 0:
        print("Incorrect input")
    # First Fibonacci number is 0
    elif n == 0:
        return 0
    # Second Fibonacci number is 1
    elif n == 1:
        return 1
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)
```

Рисунок 2 - Использование Codeium

Оба инструмента значительно упрощают процесс разработки, позволяя программистам сосредоточиться на более творческих и сложных аспектах своих проектов, минимизируя рутину и повторения. Они предоставляют мощную поддержку в создании новых проектов, изучении новых языков и технологий, а также в оптимизации и рефакторинге существующего кода.

Таким образом, на данный момент, нейросети не могут каким-либо образом заменить программистов. Они выступают лишь в качестве инструмента для обучения, написания простых прикладных задач

Список использованных источников:

1. [Электронный ресурс]. – Режим доступа: <https://www.knowledgehut.com/blog/data-science/will-chatgpt-replaceprogrammers>. – Дата доступа: 31.03.2024.
2. [Электронный ресурс]. – Режим доступа: <https://code.visualstudio.com/docs/copilot/overview>. – Дата доступа: 31.03.2024.
3. [Электронный ресурс]. – Режим доступа: https://codeium.com/vscode_tutorial. – Дата доступа: 31.03.2024.