

ПРИМЕНЕНИЕ CHATGPT В РЕШЕНИИ И ГЕНЕРАЦИИ ПРИКЛАДНЫХ ЗАДАЧ МАТЕМАТИЧЕСКОГО АНАЛИЗА

Горбачёв Александр Кириллович

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

Жгутов Евгений Дмитриевич

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

Владимцев Вадим Денисович

научный руководитель, ассистент каф. информатики, Белорусский государственный
университет информатики и радиоэлектроники, Республика Беларусь, г. Минск

Аннотация. Наше исследование посвящено использованию метода чат-бота на базе GPT для решения проблемы генерации и решения прикладных задач математического анализа. Как мы знаем, математика является одной из самых важных наук, и ее применение находит во многих областях, начиная от техники и медицины и заканчивая экономикой и финансами.

Ключевые слова: GPT, ChatGPT, искусственный интеллект, обучение, математические задачи.

Введение. Генерация и решение сложных математических задач часто представляет собой сложность и вызывает трудности у многих учеников и специалистов. Это может привести к недостаточному пониманию концепций и ошибочным решениям задач. Именно здесь наша работа может помочь улучшить этот процесс.

Основная часть. Мы рассмотрим использование метода чат-бота на базе GPT, который может помочь автоматизировать процесс генерации и решения прикладных задач математического анализа. Наш метод основывается на использовании современных алгоритмов и моделей машинного обучения, что позволяет достичь высокой точности и эффективности при решении задач.

Мы проанализировали существующие методы генерации задач математического анализа и исследовали уже проведенные исследования, посвященные использованию чат-ботов в математическом анализе. Мы сравнили наш метод с существующими методами решения задач математического анализа и оценили его эффективность и достоверность. Как же нейросеть генерирует задачи прикладной математики? Она генерирует математические задачи, используя алгоритмы машинного обучения, которые были обучены на большом количестве примеров задач. Эти задачи могут быть разных типов и уровней сложности. Для генерации математических задач нейросеть использует шаблоны или правила, которые определяют условие задачи и способ ее решения. Например, для генерации задач на интеграл, она может использовать правило интегрирования или формулы, которые были заложены в ее модель во время обучения. Кроме того, нейросеть также может использовать

случайные числа или функции для создания разнообразных задач. Например, она может использовать случайный график функции или случайное число для создания задач на поиск площади фигуры. В итоге, нейросеть создает уникальные математические задачи на основе заложенных в ее модель правил и данных, на которых была обучена. Однако, не всегда генерированные задачи будут идеальными. Например, могут возникать случаи, когда формулы, используемые для генерации задач, будут некорректными или неполными. Кроме того, возможно, что сгенерированные задачи будут иметь мало общего с тем, что учат в школе или в университете.

Чтобы уменьшить вероятность появления ошибок в генерируемых задачах, чат-бот использует подходы к контролю качества, такие как проверка сгенерированных задач с использованием известных формул и примеров. При генерации математических задач он также использует базы данных с проверенными задачами, чтобы создавать новые задачи, соответствующие требованиям обучения. Еще одним способом улучшения качества генерации задач является использование обратной связи от людей, которые занимаются математикой. Такие обратные связи позволяют учитывать мнение экспертов и улучшать процесс генерации задач. В итоге, для генерации математических задач чат-бот использует комбинацию правил, данных и проверок, чтобы создать задачи, которые будут правильными, иметь смысл и соответствовать требованиям обучения. Однако, стоит отметить, что не все задачи, которые он генерирует, могут быть полностью правильными или иметь смысл, поэтому их всегда нужно проверять перед использованием.

Почему нейросеть так хорошо генерирует задачи и что такое архитектура трансформера?

Как модель на основе архитектуры трансформера, чат-бот работает в несколько этапов.

Первый этап - это обработка входных данных. Если ему передают текст, то он использует токенизацию, чтобы разделить его на токены (отдельные слова или подстроки), затем преобразует эти токены в численные векторы, которые могут быть обработаны моделью чат-бота. Этот процесс может включать в себя использование специальных токенов для начала и конца последовательности, а также маскировку токенов, которые не должны быть использованы при обработке.

Второй этап - это применение самой архитектуры трансформера. Внутри трансформера данные обрабатываются через серию слоев, каждый из которых обрабатывает данные с разных уровней абстракции. В общем случае, трансформер состоит из множества блоков кодирования (encoder) и блоков декодирования (decoder), но в зависимости от конкретной задачи может использоваться только один или несколько блоков. Каждый блок кодирования трансформера обычно состоит из двух подблоков: многократного внимания (multi-head attention) и полносвязного слоя с пропускными соединениями (feed-forward layer with skip connections). Подблок многократного внимания позволяет модели фокусироваться на разных частях входных данных и учитывать взаимодействие между ними, а полносвязный слой обрабатывает полученную информацию и преобразует ее в более высокоуровневое представление.

В блоке декодирования также присутствует подблок многократного внимания, но с дополнительной маскировкой, чтобы избежать обращения внимания на будущие токены. Это особенно важно для задач генерации последовательностей, таких как машинный перевод.

Третий этап - это обработка выходных данных. Как правило, выходом из трансформера являются последовательности векторов, которые могут быть преобразованы в читаемый текст, перевод или какой-либо другой формат, необходимый для конкретной задачи.

Языковая модель используется в различных задачах искусственного интеллекта, где естественный язык занимает огромное место, как часть инструментария, моделирующего сознание человека. Например, языковые модели используются в задачах обработки естественного языка (Natural Language Processing, NLP), которая является подразделом искусственного интеллекта. Языковые модели также используются в многоцелевых задачах искусственного интеллекта, таких как создание чат-ботов, генерация текстов, распознавание речи и машинный перевод. Кроме того, языковые модели могут стать путем к AGI — сильному

искусственному интеллекту человеческого уровня .

В целом, работа нейросети как модели на основе архитектуры трансформера заключается в обработке входных данных через серию слоев, каждый из которых преобразует данные на более высокий уровень абстракции, и в выдаче предсказания на основе обработанных данных. В ходе работы модели также могут использоваться дополнительные техники, такие как дропаут и нормализация пакета (batch normalization), чтобы повысить ее обобщающую способность и улучшить качество предсказаний.

Одной из ключевых особенностей архитектуры трансформера является ее способность эффективно обрабатывать длинные последовательности. В отличие от более традиционных рекуррентных нейронных сетей, трансформеры не имеют ограничений на длину входных последовательностей и могут обрабатывать их параллельно, что делает их особенно полезными для задач обработки текстов.

Чем больше нейронов в нейронной сети, тем лучше она может выполнять сложные задачи, такие как распознавание образов, управление и моделирование разнообразных систем. Нейронные сети состоят из элементов, связанных между собой, которые выполняют единообразные вычислительные действия и не требуют внешнего управления. Нейронные сети используются в различных областях, таких как искусственный интеллект, обработка естественного языка, распознавание речи, компьютерное зрение и другие. Однако, при увеличении количества нейронов в нейронной сети, возникают проблемы с вычислительной сложностью и временем обучения. Поэтому, при проектировании нейронных сетей, необходимо находить баланс между количеством нейронов и вычислительной сложностью.

Кроме того, архитектура трансформера была успешно применена во многих областях, включая машинный перевод, генерацию текста, распознавание речи, классификацию текстов и многие другие. Благодаря своей мощности и гибкости, трансформеры остаются одной из наиболее популярных и широко используемых архитектур глубокого обучения.

Перейдем к исследованиям:



1. Найдите определенный интеграл функции $f(x)$ на интервале $[a, b]$: $\int [a, b] f(x)dx$, где $f(x) = 3x^2 + 2x + 1$.
2. Вычислите интеграл от функции $g(x)$ по переменной x : $\int g(x)dx$, где $g(x) = e^{(2x)} + \cos(3x)$.
3. Найдите неопределенный интеграл функции $h(x)$ по переменной x : $\int h(x)dx$, где $h(x) = 5/x + 2x^3 + 4x^2$.
4. Вычислите интеграл от функции $p(t)$ по переменной t на интервале $[c, d]$: $\int [c, d] p(t)dt$, где $p(t) = 2t^2 + 3t + 1$.
5. Найдите значение определенного интеграла функции $q(x)$ на интервале $[a, b]$: $\int [a, b] q(x)dx$, где $q(x) = \sin(x) + \cos(x)$.

Рисунок 1. Задания

У нейросети есть несколько способов генерации задач. Один из наиболее распространенных методов генерации задач - это использование случайных чисел. С его помощью можно создать задачи на различные темы математического анализа, такие как дифференцирование, интегрирование, решение дифференциальных уравнений и т.д. Однако, такие задачи могут быть простыми и легко решаемыми, что может не соответствовать потребностям студентов и учащихся, желающих более глубокого понимания математических концепций.

Другой подход - это использование базы данных задач. Такие базы могут содержать задачи различной сложности и на различные темы. Однако, при создании такой базы необходимо вручную создавать каждую задачу, что требует большого количества времени и труда. Кроме того, такие задачи могут быть ограничены в количестве и не всегда соответствовать требованиям пользователей.

Использование машинного обучения и нейронных сетей также может быть эффективным методом генерации задач. Например, можно обучать нейронную сеть на базе данных задач, которая будет генерировать новые задачи, используя свои знания и опыт. Однако, такой подход может требовать большого количества времени на обучение сети и настройку ее параметров.

Таким образом, все существующие методы генерации задач математического анализа имеют свои преимущества и недостатки. Однако, использование чат-бота на базе GPT может предложить новый и эффективный способ генерации задач. Чат-бот может генерировать новые задачи на основе предварительно загруженных данных или базы задач, а также на основе своих моделей и алгоритмов машинного обучения. Например, если у него есть база задач, то он может использовать ее для генерации новых задач, комбинируя элементы из разных задач. Кроме того, если у чат-бота есть модели, обученные на предыдущих данных, он может генерировать новые задачи на основе этих моделей, используя различные алгоритмы генерации, такие как генерация на основе марковских цепей, генерация на основе рекуррентных нейронных сетей и т.д.

А что насчет решения задач? Все так же хорошо, как и с генерацией?

Анализ решения чатом GPT задач по прикладной математике.

Задача №1.

zn найти $\lim (x - \arctg x)/(x^3) x \rightarrow 0$

 Для нахождения предела данного выражения при x , стремящемся к 0, можно воспользоваться правилом Лопиталя, которое позволяет находить пределы отношений функций, когда оба числителя и знаменателя стремятся к 0 или бесконечности. Применяя это правило, произведем дифференцирование числителя и знаменателя по отдельности.

Найдем производную числителя:
 $f(x) = x - \arctan(x)$

Используем производные разности и арктангента:
 $f'(x) = 1 - (1/(1+x^2))$

Теперь найдем производную знаменателя:
 $g(x) = x^3$

Используем производную степенной функции:
 $g'(x) = 3x^2$

Рисунок 2. Алгоритм решения

Теперь можем воспользоваться правилом Лопиталя, подставляя найденные производные:
$$\lim_{x \rightarrow 0} (x - \arctan(x)) / (x^3) = \lim_{x \rightarrow 0} (1 - (1/(1+x^2))) / (3x^2)$$

Подставляем $x = 0$:
$$(1 - (1/(1+0^2))) / (3 * 0^2)$$

Таким образом, предел данного выражения при x , стремящемся к 0, равен $1/3$.

Рисунок 3. Алгоритм решения

Задача №2.

zn

Найти: $\lim_{x \rightarrow 0} (\sin x - x \cos x) / (\sin^3 x)$



Для нахождения предела функции при x , стремящемся к 0, можно использовать разложение в ряд Маклорена и арифметические свойства пределов.

Используя разложение в ряд Маклорена функций $\sin x$ и $\cos x$, получим:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$
$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$$

Подставим эти разложения в исходное выражение:

$$\lim_{x \rightarrow 0} (\sin x - x \cos x) / (\sin^3 x)$$
$$= \lim_{x \rightarrow 0} ((x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots) - x(1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots)) / (\sin x)^3$$
$$= \lim_{x \rightarrow 0} (x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots - x + \frac{x^3}{2!} - \frac{x^5}{4!} + \dots) / (\sin x)^3$$
$$= \lim_{x \rightarrow 0} (2x^3/3! - 2x^5/5! + \dots) / (\sin x)^3$$

Рисунок 4



Для нахождения предела функции при x , стремящемся к 0, можно использовать разложение в ряд Маклорена и арифметические свойства пределов.

Используя разложение в ряд Маклорена функций $\sin x$ и $\cos x$, получим:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$$

Подставим эти разложения в исходное выражение:

$$\lim_{x \rightarrow 0} (\sin x - x \cos x) / (\sin^3 x)$$

$$= \lim_{x \rightarrow 0} ((x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots) - x(1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots)) / (\sin x)^3$$

$$= \lim_{x \rightarrow 0} (x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots - x + \frac{x^3}{2!} - \frac{x^5}{4!} + \dots) / (\sin x)^3$$

$$= \lim_{x \rightarrow 0} (2\frac{x^3}{3!} - 2\frac{x^5}{5!} + \dots) / (\sin x)^3$$

Теперь воспользуемся фактом, что $\sin x$ при x , стремящемся к 0, приближается к x .

Можем заменить $\sin x$ на x в числителе:

$$= \lim_{x \rightarrow 0} (2\frac{x^3}{3!} - 2\frac{x^5}{5!} + \dots) / (x)^3$$

$$= \lim_{x \rightarrow 0} 2/3! - 2x^2/5! + \dots$$

Теперь, когда x стремится к 0, все слагаемые с x^2 и выше будут стремиться к 0, так как x^2 умноженное на какое-либо число будет стремиться к 0 быстрее, чем $1/x^3$. Поэтому можем пренебречь этими слагаемыми:

$$= 2/3!$$

$$= 2/6$$

$$= 1/3$$

Таким образом, итоговый предел функции $(\sin x - x \cos x) / (\sin^3 x)$ при x , стремящемся к 0, равен $1/3$.

Рисунок 5. Алгоритм решения

Задача №3.



Для нахождения приближенного значения функции $y(x) = \sqrt{x}$ в точке $x = 3,98$, можно использовать линейную аппроксимацию, также известную как метод линейного приближения или метод касательной.

Линейная аппроксимация функции $y(x)$ в точке $x = a$ может быть выражена следующей формулой:

$$y(x) \approx y(a) + y'(a) * (x - a),$$

где $y(a)$ - значение функции $y(x)$ в точке $x = a$,
 $y'(a)$ - производная функции $y(x)$ в точке $x = a$,
 x - значение, в котором мы хотим найти приближенное значение функции.

Теперь, найдем значение функции $y(x) = \sqrt{x}$ и ее производной $y'(x)$ в точке $x = 3,98$:

$y(a) = \sqrt{a}$, где $a = 3,98$,
 $y'(a) = 1 / (2 * \sqrt{a})$, так как производная функции $\sqrt{x}$ равна $1 / (2 * \sqrt{x})$.

Теперь, подставим найденные значения в формулу линейной аппроксимации:

$$y(3,98) \approx y(3,98) + y'(3,98) * (x - 3,98) \approx \sqrt{3,98} + 1 / (2 * \sqrt{3,98}) * (3,98 - 3,98).$$

Рисунок 6. Алгоритм решения

Заметим, что $(3,98 - 3,98) = 0$, поэтому последнее слагаемое в формуле обращается в ноль.

Таким образом, приближенное значение функции $y(x) = \sqrt{x}$ в точке $x = 3,98$ можно найти как $\sqrt{3,98}$, что составляет приблизительно 1,995.

Рисунок 7. Алгоритм решения

Задача №4.



Для вычисления дифференциала функции $y = x \cdot \sqrt{64 - x^2} + 64 \arcsin(x/8)$ воспользуемся правилами дифференцирования.



1. Для первого слагаемого $x \cdot \sqrt{64 - x^2}$ применим правило производной произведения функций:

$$(d/dx)(uv) = u'v + uv',$$

где $u = x$, $u' = 1$ (производная x по x) и $v = \sqrt{64 - x^2}$, $v' = d(\sqrt{64 - x^2})/dx$.

Таким образом, производная первого слагаемого будет равна:

$$u'v + uv' = 1 \cdot \sqrt{64 - x^2} + x \cdot d(\sqrt{64 - x^2})/dx.$$

1. Для второго слагаемого $64 \arcsin(x/8)$ применим правило производной арксинуса:

$$(d/dx) \arcsin(u) = du/\sqrt{1 - u^2},$$

где $u = x/8$, $du = dx/8$.

Рисунок 8. Алгоритм решения

Таким образом, производная второго слагаемого будет равна:

$$du/\sqrt{1 - (x/8)^2} = (dx/8)/\sqrt{1 - x^2/64} = dx/(8 \cdot \sqrt{1 - x^2/64}).$$

Таким образом, производная функции y по x будет равна сумме производных её слагаемых:

$$dy/dx = 1 \cdot \sqrt{64 - x^2} + x \cdot d(\sqrt{64 - x^2})/dx + dx/(8 \cdot \sqrt{1 - x^2/64}).$$

Полученное выражение и есть дифференциал функции $y = x \cdot \sqrt{64 - x^2} + 64 \arcsin(x/8)$.

Рисунок 9. Алгоритм решения

Вывод. Чат GPT считается слабой формой ИИ, когда речь идет о решении математических задач или постановке прикладных задач по математическому анализу. Чату GPT не хватает способности понимать контекст и смысл так же, как это делают люди, и он работает на основе шаблонов и статистических вероятностей, что может привести к ошибкам и неточностям в его ответах. В Chat GPT нет калькулятора, и любая операция с некоторой вероятностью может быть ошибочной. Генеральный директор OpenAI прямо упомянул, что GPT-4 не подключен к калькулятору. Поэтому Chat GPT может не подойти для решения сложных математических задач или создания прикладных задач по математическому анализу, так как может не иметь

необходимого понимания задействованных понятий. **Chat GPT смог решить только 25% задач.** Как искусственный интеллект, он не может решать задачи так же, как это делают люди. Он основан на математических алгоритмах и статистических методах, которые позволяют ему обрабатывать большие объемы текстовых данных и генерировать ответы на основе изученного контекста. Однако, он не всегда может точно решить задачу, особенно если не имеет достаточной информации для ее решения. Он смог решить только одну задачу из четырех, возможно, это связано с тем, что информация, необходимая для решения оставшихся трех задач, была недостаточной или некорректной. Также может быть, что Chat GPT не распознал некоторые ключевые слова или понятия в формулировке задачи, что привело к неверным ответам. Chat GPT генерирует ответы на основе анализа контекста и использования своей внутренней базы данных (БД) с ранее изученными текстовыми данными. Если он не имеет достаточной информации для решения задачи, он может попытаться использовать аналогии из своей БД, но это может привести к неверным ответам. Также стоит отметить, что большее количество нейронов не всегда дает лучшую точность. Вместо этого, для достижения высокой точности необходимо обучать Chat GPT на большом и разнообразном наборе данных и постоянно улучшать его алгоритмы. Кроме того, Chat GPT не может решать новые проблемы, которые не были изучены в его БД. Он может только использовать знания, полученные из ранее изученных текстовых данных.

Заключение. Наша работа представляет важный шаг в исследовании возможностей применения искусственного интеллекта для решения и генерации задач математического анализа. Однако, как бы продвинутой не была модель, такая как GPT-3.5, она не может полностью заменить ручное решение задач. Ручное решение задач остается важным навыком для любого математика. Поэтому, комбинированный подход к решению математических задач, используя различные методы, является наилучшим способом достижения наилучшего результата. Стоит также упомянуть о возможностях GPT-4, которая является мультимодальной моделью, способной работать с текстом, звуком и изображениями. Она может распознавать изображения и видео, идентифицировать и объяснять визуальные шутки, определять и даже генерировать более уникальные и творческие ответы. GPT-4 также имеет встроенную функцию управления, которая предоставляет пользователям больший контроль над генерируемыми ответами. Она также более мультязычная и может запоминать больше информации, что делает ее более продвинутой по сравнению с конкурентами. В целом, GPT-4 представляет собой более продвинутую модель, которая может работать с различными типами данных и может стать важным инструментом для генерации прикладных задач математического анализа в комбинации с другими методами.

Список литературы:

1. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. [1]
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008). [2]
3. Л.Д.Кудрявцев, А.Д. Кутасов, В.И. Чехлов, М.И.Шабунин. Сборник задач по математическому анализу. [3]
4. А.П.Рябушко. Сборник индивидуальных заданий по высшей математике. [4]