Материалы III Всероссийской научно-практической конференции с междун. участием «ИИ в образовании. Современные достижения и перспективы применения: в генерации знаний, управлении, обучении, оценке результатов обучения и формировании компетенций обучающихся», 2024, №6 (93).

УДК 372.800.2

С. В. Журавлев

S. V. Zhuravlev

Журавлёв Сергей Владимирович, преподаватель, КГПИ ФГБОУ ВО «КемГУ», г. Новокузнецк, Россия.

Zhuravlev Sergei Vladimirovich, lecturer, Kuzbass Humanitarian and Pedagogical Institute of Kemerovo State University, Novokuznetsk, Russia.

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ПРИ ОБУЧЕНИИ ШКОЛЬНИКОВ ПРОГРАММИРОВАНИЮ

THE USE OF NEURAL NETWORKS IN TEACHING PROGRAMMING TO PUPILS

Аннотация. *В статье кратко описываются особенности внедрения искусственного интеллекта в сферу образования. Рассматриваются особенности применения нейронных сетей при обучении школьников программированию на примере Replit AI, которая умеет генерировать коды программ.*

Annotation. *The article briefly describes the features of the introduction of artificial intelligence in the field of education. The features of the use of neural networks in teaching programming to pupils are considered using the example of Replit AI, which can generate program codes.*

С. В. Журавлев 2024-07-08

Ключевые слова: *искусственный интеллект, нейронная сеть, обучение школьников программированию, генерация кода программы.*

Keywords: *artificial intelligence, neural network, teaching programming to pupils, program code generation.*

В настоящее время внедрение искусственного интеллекта в сферу образования приобретает повсеместный характер. Основной целью использования искусственного интеллекта в образовательных процессах является повышение качества обучения.

По мере внедрения искусственного интеллекта место традиционных занятий занимает разноформатное проектное обучение. Ведущей деятельностью учителя становится наставническая, он мотивирует учеников к самостоятельной работе, обучает приёмам работы с информацией, поиску новых решений. Это связано с тем, что искусственный интеллект в образовании основан на использовании разнообразных приложений, включая интеллектуальных наставников, функциях персональной и оперативной обратной связи, контроля прогресса в обучении [1].

Одна из форм искусственного интеллекта – нейронные сети. Нейронные сети используют имитирующие работу мозга математические модели для анализа данных и принятия решений. Главное свойство нейронной сети – способность обучаться на основе имеющихся данных. Благодаря своей высокой адаптивности и универсальности нейронные сети умеют решать задачи, которые требуют сложного анализа и обработки больших объёмов данных [2].

С развитием технологий искусственного интеллекта нейронные сети начали учиться создавать код на разных языках программирования. Это стало возможным благодаря глубокому обучению и моделям, способным анализировать большие объёмы кода и извлекать из него закономерности. Процесс генерации кода с использованием нейронных сетей обычно начинается с подачи на вход модели определённой задачи или описания того, что нужно сделать (промпта). Нейронная сеть затем анализирует этот запрос и использует свои знания о языках программирования, структурах данных и алгоритмах, чтобы сгенерировать соответствующий код.

Нейронные сети могут быть полезными инструментами для обучения новичков программированию, в том числе школьников. Они могут генерировать примеры кода и объяснять, как они работают [4].

Рассмотрим работу нейронных сетей по генерации кода на примере Replit AI, входящей в состав платформы Replit. Эта платформа удобна тем, что её не нужно скачивать, с ней можно работать на любом устройстве с подключением к сети Интернет после регистрации. Уровень бесплатного пользования имеет ограничения на ресурсы, но для школьников, только приступивших к изучению программирования, этого будет достаточно.

В октябре 2023 платформа Replit открыла бесплатный доступ к своему интеллектуальному помощнику, который отлично выполняет все основные функции универсального ассистента разработчика:

1) генерирует код по промптам;

2) дополняет и комментирует код, находит баги и опечатки;

3) тестирует приложения;

4) отвечает на вопросы, помогает найти решение нетривиальной проблемы [3].

Чтобы запустить чат для подачи промпта, нужно выбрать в левой части экрана в разделе Tools пункт AI, а затем в правой части экрана выбрать пункт Generate. Запросы можно делать и на русском языке.

В качестве примера мы, выступая в роли учителя, сформулировали для нейронной сети следующий промпт: «Создай приложение с графическим интерфейсом в Lazarus для решения задачи: дана строка s, состоящая из слов (последовательностей символов, не содержащих пробелов внутри себя), разделенных между собой одним или несколькими пробелами; определить, сколько слов состоит из одинаковых символов».

И вот как ответила нейронная сеть: «To solve this task, we will create a Lazarus application with a graphical user interface where the user can input a string containing words separated by spaces. The application will then determine how many words consist of the same characters. Here is the code snippet to achieve this in Lazarus:».

И далее следует листинг программы:

```
unit MainForm;

{$mode objfpc}{$H+}

interface

uses

  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls;

type

  { TMainForm }

  TMainForm = class(TForm)

    ButtonCheckWords: TButton;

    EditInputString: TEdit;

    LabelInput: TLabel;
```

```
    LabelResult: TLabel;
    procedure ButtonCheckWordsClick(Sender: TObject);
  private
  public
  end;

var
  MainForm: TMainForm;
implementation
{$R *.lfm}
{ TMainForm }
procedure TMainForm.ButtonCheckWordsClick(Sender: TObject);
var
  inputString: string;
  wordsList: TStringList;
  word: string;
  sameCharWordsCount: integer;
  i, j: integer;
  isSame: boolean;
begin
  inputString:=EditInputString.Text;
  wordsList:=TStringList.Create;
  wordsList.Delimiter:=' ';
  wordsList.DelimitedText:=inputString;
```

```
    sameCharWordsCount:=0;
    for i:=0 to wordsList.Count-1 do
    begin
      word:=wordsList[i];
      isSame:=true;
      for j:=2 to Length(word) do
      begin
        if word[j]<>word[1] then
        begin
          isSame:=false;
          Break;
        end;
      end;
      if isSame then
        Inc(sameCharWordsCount);
    end;
     LabelResult.Caption:='Number of words with the same characters: '+IntToStr(sameCharWordsCount);
    wordsList.Free;
  end;
  end.
```

Далее нейронная сеть даёт пояснения по коду: «This code defines a form with an input field for the string, a button to check the words, and a label to display the result. When the button is clicked, the program will count and display how many words consist of the same characters».

При анализе предложенного кода сразу бросается в глаза, что нейронная сеть разместила на форме компонент LabelInput, который ни разу не используется в программе. Сообщаем об этом нейронной сети «Спасибо, только я не понял, для чего нужен компонент LabelInput».

Нейронная сеть ответила следующим образом: «The LabelInput component in the Lazarus form was intended to provide a label indicating the purpose of the input field EditInputString. However, if it's not necessary for your requirements, you can remove or modify it as needed. Here is the revised code without the LabelInput component:» и выдала тот же код, что и в прошлый раз, только без компонента LabelInput.

Анализируя программу глубже, отмечаем, что, хотя она и удовлетворяет условию задачи, но в ней для разделения строки на слова используется класс TStringList, который не изучается школьниками на начальном этапе обучения программированию. Они выделяют слова, используя классический алгоритм. Мы попросили нейронную сеть предложить другой вариант кода, содержащий данный алгоритм: «Твоё решение хорошее, но школьники могут быть незнакомы с классом TStringList. Можно решить задачу, используя такой алгоритм разбиения строки на слова?»

```
s:=Edit1.Text+' ';

s1:='';

for i:=1 to length(s) do

if s[i]<>' ' then

s1:=s1+s[i] else

if s[i-1]<>' ' then

begin

Label1.Caption:=s1;

s1:='';
```

```
end;
```

Вот ответ нейронной сети: «Sure, I can provide a Pascal code snippet that solves the task by splitting the string into words using the provided algorithm. Here is the modified code:».

И новый листинг программы, уже сокращённый, содержащий только процедуру решения задачи:

```
procedure TMainForm.ButtonCheckWordsClick(Sender: TObject);
var
  inputString, word: string;
  currentChar: char;
  sameCharWordsCount: integer;
  i: integer;
  isSame: boolean;
begin
  inputString:=EditInputString.Text+' ';
  word:='';
  sameCharWordsCount:=0;
  for i:=1 to Length(inputString) do
  begin
    currentChar:=inputString[i];
    if currentChar<>' ' then
      word:=word+currentChar
    else
    begin
```

```
        if (i>1) and (inputString[i-1]<>' ') then
        begin
          isSame:=true;
          for i:=2 to Length(word) do
          begin
            if word[i]<>word[1] then
            begin
              isSame:=false;
              Break;
            end;
          end;
          if isSame then
            Inc(sameCharWordsCount);
          word:='';
        end;
      end;
    end;
    LabelResult.Caption:='Number of words with the same characters: '+IntToStr(sameCharWordsCount);
  end;
```

После листинга следует комментарий нейронной сети: «This code snippet uses the provided algorithm to split the input string into words and count how many words consist of the same characters».

Анализируя данный код, замечаем, что нейронная сеть допустила ошибку, используя одну и ту же переменную i в качестве счётчика и внешнего, и вложенного цикла. Указываем ей на эту ошибку: «Во вложенном цикле, наверное, нужно использовать другую переменную для счётчика, т. к. i уже занята».

Для нейронной сети не составило труда исправить ошибку: «Here is the updated Pascal code snippet where a different variable is used as the counter in the nested loop:».

Листинг программы такой же, как и в прошлый раз, только во вложенном цикле в качестве счётчика вместо переменной i используется j.

Так, совместными усилиями нейронной сети и учителя, удалось получить правильно работающий код, подходящий для школьников. Обучающиеся наблюдают за этим процессом и учатся использовать в дальнейшем нейронную сеть самостоятельно.

При обучении школьников программированию использование нейронных сетей вполне допустимо. Эта методика не только позволяет обучающимся своевременно получать помощь в случае затруднений, но и может способствовать развитию у них аналитического мышления и внимательности, поскольку нейронные сети не понимают код в том смысле, как это делает человек. Они обучаются на основе огромных объёмов данных и могут предсказать, какой код больше всего соответствует запросу. Это означает, что нейронные сети способны создавать рабочий код, но не всегда гарантируют его оптимальность или эффективность, так что иногда требуется вмешательство человека для его улучшения [4], т. е. школьники должны будут внимательно анализировать каждый сгенерированный код, находить неточности и предлагать свои варианты для их исправления.

Список литературы

1. Бевза, Д. Как искусственный интеллект меняет обучение в школе и университете / Д. Бевза. – Текст :

электронный. – URL : https://rg.ru/2024/01/31/kak-iskusstvennyj-intellekt-meniaet-obuchenie-v-shkole-i-universitete.html (дата обращения : 20.05.2024).
2. Искусственный интеллект для учебы. – Текст : электронный. – URL : https://developers.sber.ru/help/gigachat/ai-for-study (дата обращения : 20.05.2024).
3. Кайда, Н. 25 бесплатных AI-инструментов для разработчиков / Н. Кайда. – Текст : электронный. – URL : https://proglib.io/p/25-besplatnyh-ai-instrumentov-dlya-razrabotchikov-2023-10-18?ysclid=lwyzis93p1573571313 (дата обращения : 20.05.2024).
4. Нейросети в мире программирования: от текстов до кода // vc.ru : [сайт]. – URL : https://vc.ru/services/858329-neiroseti-v-mire-programmirovaniya-ot-tekstov-do-koda (дата обращения : 20.05.2024). – Текст : электронный.