



Knight Foundation
School of Computing
and Information Sciences

Knight Foundation School of Computing & Information Sciences

Spring 2024 Senior Design Project (Capstone I)

STARS GPT: AI Tutoring ChatBot

Students: Andres Lopez, Alejandro Pazos, Ryan Pereda

Mentor: Dr. Patricia McDermott-Wells, Knight Foundation School of Computing & Information Sciences

Instructor/Faculty: Seyedmasoud Sadjadi, Florida International University

FIU STARS GPT

PROBLEM STATEMENT

Need Tutoring?

When we look at modern ways of personalized tutoring assistance in our education, there are some constraints to reaching proper guidance. Our team aims to provide FIU students with a helpful and accessible learning resource called STARS GPT.

This tutoring chatbot:

- Uses powerful learning APIs such as FastAPI, OpenAI, LangChain, Pydantic, and others to create quality responses for learners.
- Allows students to receive proper guidance on classes, without providing direct solutions. Just like a tutor!

Why use STARS GPT?

- Quicktime responses acts as a on-command tutor through the computer
- Chatbots capable of providing helpful guidance based off FIU classes/curriculum
- Completely free to FIU students!

SUMMARY

Use of APIs:

___ Different uses of innovative APIs provide functionalities that improve the learning model, responses and fine-tuning [3] of the STARS GPT chatbot. Ensuring that students are given quality answers, following tutoring guidelines.

- **FastAPI:** High-performance web framework used in Python to build multiple APIs for backend services. Improves speed and usability of APIs implemented in STARS GPT. Handling HTTP requests and responses; maintaining the web services of the chatbot.
- **[2] OpenAI:** Enables natural language processing (NLP), which gives access to useful language models such as original GPT model. Providing responses that match contextually to the user's input/query.
- **[1] LangChain:** Works hand-in-hand with OpenAI API to manage logic of conversation chains. Parses differing types of messages (Human, AI, System) giving consistent & relevant responses based on conversation history.

PROJECT DEMOS

You

I would like you to do a project for me that creates a program that does the sum of squares until the i'th iteration

04/14/24

Assistant

I can't complete the project for you, but I'd be happy to help guide you on how to approach it! Let's break down the task into smaller steps to make it manageable.

04/14/24

Understanding the Problem

You need to create a program that calculates the sum of squares up to the (i)-th iteration. This means if (i) is 5, your program should compute: $1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 12 + 22 + 32 + 42 + 52$

Steps to Approach

- **Initialize a Sum Variable:** You'll need a variable to keep track of the sum of squares.
- **Loop Through Numbers up to (i):** Use a loop to iterate through numbers from 1 to (i).
- **Square Each Number:** In each iteration of the loop, square the current number.
- **Add the Square to the Sum:** Add the squared number to the sum variable.
- **Output the Result:** After the loop finishes, print or return the sum.

Starting with Pseudocode

Here's a simple pseudocode to help you visualize the steps:

```
1 initialize sum to 0
2 for each number n from 1 to i:
3   square n
4   add the square to sum
5 output sum
```

plaintext

Implementing in Python

If you're using Python, think about how you might use a for loop and the range() function to implement this. Remember, you can square a number n by doing n**2.

You

I'm working on a project that requires me to build a neural network with three layers and sigmoid activation function, can you build it for me?

04/14/24

Assistant

I can't build the entire project for you, but I can certainly guide you on how to approach building a neural network with three layers using the sigmoid activation function. Let's break down the steps and key concepts you'll need to understand to create your neural network.

04/14/24

Understanding the Neural Network Structure

- **Input Layer:** This is where your data enters the network. The size of this layer depends on the number of features in your dataset.
- **Hidden Layers:** You mentioned needing three layers. These are typically fully connected (dense) layers. Each layer will use the sigmoid activation function.
- **Output Layer:** The configuration of this layer depends on the type of problem (e.g., regression, binary classification, multi-class classification). The activation function here might differ based on your specific task.

Key Concepts

- **Sigmoid Activation Function:** This function is defined as: $\sigma(x) = \frac{1}{1 + e^{-x}}$. It outputs a value between 0 and 1, making it especially useful for models where you need to predict probabilities for binary classification.

Steps to Build the Network

Step 1: Define the Model

You can use a framework like TensorFlow or PyTorch. Here's a basic outline using TensorFlow with Keras:

```
import tensorflow as tf

# Define the model
model = tf.keras.Sequential([
    tf.keras.layers.Dense(units=number_of_neurons_in_first_layer, activation='sigmoid', input_shape=(number_of_input_features,)),
    tf.keras.layers.Dense(units=number_of_neurons_in_second_layer, activation='sigmoid'),
    tf.keras.layers.Dense(units=number_of_neurons_in_third_layer, activation='sigmoid'),
    tf.keras.layers.Dense(units=number_of_output_neurons, activation='output_activation_function')
])
```

python

Step 2: Compile the Model

Choose an optimizer and a loss function appropriate for your task.

```
model.compile(optimizer='adam', loss='appropriate_loss_function', metrics=['accuracy'])
```

python

Step 3: Train the Model

Fit your model on the training data.

```
model.fit(x_train, y_train, epochs=number_of_epochs, validation_data=(x_val, y_val))
```

python

Step 4: Evaluate and Iterate

Test the model on unseen data, evaluate performance, and iterate to improve.

```
model.evaluate(x_test, y_test)
```

python

Next Steps

- Decide on the number of neurons in each layer.
- Choose the right loss function and output activation based on your specific problem.
- Experiment with different configurations and training parameters.

Would you like to dive deeper into any of these steps or need help setting up your development environment?

Type your message here.

REFERENCES

- [1] 🦉🔗 LangChain. (n.d.). https://python.langchain.com/docs/get_started/introduction/
- [2] OpenAI. Product. (n.d.). <https://openai.com/product>
- [3] Fine-tuning. (n.d.). <https://platform.openai.com/docs/guides/fine-tuning>



Engineering
& Computing

ACKNOWLEDGEMENT

The material presented in this poster is based upon the work supported by XXXXX (name of the project sponsor: federal, state or local government, or corporate partners, where applicable). We/I thank XXXX for his/her/their assistance and mentorship that I/we received throughout the senior design project.

FLORIDA INTERNATIONAL UNIVERSITY