AI-LEARN-07a Azure OpenAI (Learn)

**Tip**: As you follow the instructions in this pane, whenever you see a icon, you can use it to copy text from the instruction pane into the virtual machine interface. This is particularly useful to copy code; but bear in mind you may need to modify the pasted code to fix indent levels or formatting before running it!

You will need an Azure subscription that has been approved for access to the Azure OpenAI service. To request subscription access to the Azure OpenAI service, visit <https://aka.ms/oaiapply>. Access is restricted to approved Microsoft enterprise customers.

1. Sign into Windows as **Student** account with the password Pa55w.rd.
2. Follow the instructions below to complete the exercise.

Get started with Azure OpenAI Service

Azure OpenAI Service brings the generative AI models developed by OpenAI to the Azure platform, enabling you to develop powerful AI solutions that benefit from the security, scalability, and integration of services provided by the Azure cloud platform. In this exercise, you'll learn how to get started with Azure OpenAI by provisioning the service as an Azure resource and using Azure OpenAI Studio to deploy and explore OpenAI models.

This exercise takes approximately **30** minutes.

Provision an Azure OpenAI resource

If you don't already have one, provision an Azure OpenAI resource in your Azure subscription.

1. Sign into the **Azure portal** at https://portal.azure.com.
2. Create an **Azure OpenAI** resource with the following settings:
   * **Subscription**: *Select an Azure subscription that has been approved for access to the Azure OpenAI service*
   * **Resource group**: *Choose or create a resource group*
   * **Region**: *Make a****random****choice from any of the available regions*\*
   * **Name**: *A unique name of your choice*
   * **Pricing tier**: Standard S0

\* Azure OpenAI resources are constrained by regional quotas. Randomly choosing a region reduces the risk of a single region reaching its quota limit in scenarios where you are sharing a subscription with other users. In the event of a quota limit being reached later in the exercise, there's a possibility you may need to create another resource in a different region.

1. Wait for deployment to complete. Then go to the deployed Azure OpenAI resource in the Azure portal.

Deploy a model

Azure OpenAI provides a web-based portal named **Azure OpenAI Studio**, that you can use to deploy, manage, and explore models. You'll start your exploration of Azure OpenAI by using Azure OpenAI Studio to deploy a model.

1. On the **Overview** page for your Azure OpenAI resource, use the **Go to Azure OpenAI Studio** button to open Azure OpenAI Studio in a new browser tab.
2. In Azure OpenAI Studio, on the **Deployments** page, view your existing model deployments. If you don't already have one, create a new deployment of the **gpt-35-turbo-16k** model with the following settings:
   * **Model**: gpt-35-turbo-16k *(if the 16k model isn't available, choose gpt-35-turbo)*
   * **Model version**: Auto-update to default
   * **Deployment name**: *A unique name of your choice*
   * **Advanced options**
     + **Content filter**: Default
     + **Tokens per minute rate limit**: 5K\*
     + **Enable dynamic quota**: Enabled

\* A rate limit of 5,000 tokens per minute is more than adequate to complete this exercise while leaving capacity for other people using the same subscription.

Use the Chat playground

The *Chat* playground provides a chatbot interface for GPT 3.5 and higher models.

**Note:** The *Chat* playground uses the *ChatCompletions* API rather than the older *Completions* API that is used by the *Completions* playground. The Completions playground is provided for compatibility with older models.

1. In the **Playground** section, select the **Chat** page. The **Chat** playground page consists of three main sections:
   * **Assistant setup** - used to set the context for the model's responses.
   * **Chat session** - used to submit chat messages and view responses.
   * **Configuration** - used to configure settings for the model deployment.
2. In the **Configuration** section, ensure that your model deployment is selected.
3. In the **Assistant setup** section, in the **System message** box, replace the current text with the following statement: The system is an AI teacher that helps people learn about AI.
4. Below the **System message** box, select **Add an example**, and type the following message and response in the designated boxes:
   * **User**: What are different types of artificial intelligence?
   * **Assistant**: There are three main types of artificial intelligence: Narrow or Weak AI (such as virtual assistants like Siri or Alexa, image recognition software, and spam filters), General or Strong AI (AI designed to be as intelligent as a human being. This type of AI does not currently exist and is purely theoretical), and Artificial Superintelligence (AI that is more intelligent than any human being and can perform tasks that are beyond human comprehension. This type of AI is also purely theoretical and has not yet been developed).

**Note**: Few-shot examples are used to provide the model with examples of the types of responses that are expected. The model will attempt to reflect the tone and style of the examples in its own responses.

1. Save the changes to start a new session and set the behavioral context of the chat system.
2. In the **Chat session** section, enter the user query What is artificial intelligence?

**Note**: You may receive a response that the API deployment is not yet ready. If so, wait for a few minutes and try again.

1. Review the response and then submit the following message to continue the conversation: How is it related to machine learning?
2. Review the response, noting that context from the previous interaction is retained (so the model understands that "it" refers to artificial intelligence).
3. Use the **View Code** button to view the code for the interaction. The prompt consists of the *system* message, the few-shot examples of *user* and *assistant* messages, and the sequence of *user* and *assistant* messages in the chat session so far.

Explore prompts and parameters

You can use the prompt and parameters to maximize the likelihood of generating the response you need.

1. In the **Parameters** pane, set the following parameter values:
   * **Temperature**: 0
   * **Max length (tokens)**: 500
2. Submit the following message
3. Write three multiple choice questions based on the following text, indcating the correct answers.
4. Most computer vision solutions are based on machine learning models that can be applied to visual input from cameras, videos, or images.\*
5. - Image classification involves training a machine learning model to classify images based on their contents. For example, in a traffic monitoring solution you might use an image classification model to classify images based on the type of vehicle they contain, such as taxis, buses, cyclists, and so on.\*
6. - Object detection machine learning models are trained to classify individual objects within an image, and identify their location with a bounding box. For example, a traffic monitoring solution might use object detection to identify the location of different classes of vehicle.\*

- Semantic segmentation is an advanced machine learning technique in which individual pixels in the image are classified according to the object to which they belong. For example, a traffic monitoring solution might overlay traffic images with "mask" layers to highlight different vehicles using specific colors.

1. Review the results, which should consist of multiple-choice questions that a teacher could use to test students on the computer vision topics in the prompt. The total response should be smaller than the maximum length you specified as a parameter.

Observe the following about the prompt and parameters you used:

* + The prompt specifically states that the desired output should be three multiple choice questions.
  + The parameters include *Temperature*, which controls the degree to which response generation includes an element of randomness. The value of **0** used in your submission minimizes randomness, resulting in stable, predictable responses.

Explore code-generation

In addition to generating natural language responses, you can use GPT models to generate code.

1. In the **Assistant setup** pane, select the **Empty Example** template to reset the system message.
2. Enter the system message: You are a Python developer. and save the changes.
3. In the **Chat session** pane, select **Clear chat** to clear the chat history and start a new session.
4. Submit the following user message:

Write a Python function named Multiply that multiplies two numeric parameters.

1. Review the response, which should include sample Python code that meets the requirement in the prompt.

Clean up

When you're done with your Azure OpenAI resource, remember to delete the deployment or the entire resource in the **Azure portal** at https://portal.azure.com.