Cheat Sheet: Building Apps with RAG

Estimated Reading Time: 5 minutes

Gradio is an open-source Python package that allows for the quick building of an application interface. No JavaScript, CSS, or web hosting experience is required!

Interface class core arguments

Gradio's Interface class wraps any arbitrary Python function. The interface class has the following three core arguments:

Term	Definition
Fn	Fn is the function to wrap. Each function parameter represents an input component, and the function's output should be either a single value or a tuple. Each element within the tuple corresponds directly to a specific output component.
inputs	The Gradio component(s) to use for the inputs of the function. The number of inputs should match the number of arguments in the function.
outputs	Outputs are the Gradio component(s) to use for the outputs of the function.

Common input types

Term	Definition
`checkbox`	A checkbox that can be set to `True` or `False` only.
`checkboxGroup`	An input type that allows users to select multiple values from a predefined checkbox list.
`dropdown`	An input type that provides a dropdown list where, by default, one value can be selected. If `multiselect` is set to `True`, then one or more values can be selected.
`file`	An input type that allows a user to upload a file.
`image`	An input type that allows the user to select or upload an image.
`radio`	An input type that forces the user to choose one value.
`slider`	An input type that provides a slider where a value must be selected between a minimum and a maximum range. The `value` parameter defines the default value, and `step` provides the increment value. Setting the minimum, maximum, and `step` values to integers will select integer values.
`textbox`	An expandible text box that allows the user to type in text.

Common output types

Term	Definition
`textbox`	An expandible text box that allows the user to type in text.
`label`	A type typically used for classification models. Can output the classes along with their predicted probabilities. You can control the number of output classes using the `num_top_classes` parameter.

Launching the application

Use the `launch()` method to launch the application. This method starts a simple web server that serves the interface. If you set `share=True` in `launch()`, you can share the application with the rest of the world. The system will set up a public URL allowing anyone worldwide to access your application. Users can interact with your app through this URL, while all computations run on your local machine.

Author

Wojciech "Victor" Fulmyk