

How to serialize LLM classes

Contents

- Loading
- Saving

This notebook walks through how to write and read an LLM Configuration to and from disk. This is useful if you want to save the configuration for a given LLM (e.g., the provider, the temperature, etc).

```
from langchain.llms import OpenAI
from langchain.llms.loading import load_llm
```

Loading

First, lets go over loading an LLM from disk. LLMs can be saved on disk in two formats: json or yaml. No matter the extension, they are loaded in the same way.

```
!cat llm.json
```

```
{
  "model_name": "text-davinci-003",
  "temperature": 0.7,
  "max_tokens": 256,
  "top_p": 1.0,
  "frequency_penalty": 0.0,
  "presence_penalty": 0.0,
  "n": 1,
  "best_of": 1,
  "request_timeout": null,
  "_type": "openai"
}
```

[Skip to main content](#)

```
llm = load_llm("llm.json")
```

```
!cat llm.yaml
```

```
_type: openai
best_of: 1
frequency_penalty: 0.0
max_tokens: 256
model_name: text-davinci-003
n: 1
presence_penalty: 0.0
request_timeout: null
temperature: 0.7
top_p: 1.0
```

```
llm = load_llm("llm.yaml")
```

Saving

If you want to go from an LLM in memory to a serialized version of it, you can do so easily by calling the `.save` method. Again, this supports both json and yaml.

```
llm.save("llm.json")
```

```
llm.save("llm.yaml")
```