How to add Memory to an LLMChain

This notebook goes over how to use the Memory class with an LLMChain. For the purposes of this walkthrough, we will add the `ConversationBufferMemory` class, although this can be any memory class.

```
from langchain.memory import ConversationBufferMemory
from langchain import OpenAI, LLMChain, PromptTemplate
```

The most important step is setting up the prompt correctly. In the below prompt, we have two input keys: one for the actual input, another for the input from the Memory class. Importantly, we make sure the keys in the PromptTemplate and the ConversationBufferMemory match up (`chat_history`).

```
template = """You are a chatbot having a conversation with a human.

{chat_history}
Human: {human_input}
Chatbot:"""

prompt = PromptTemplate(
    input_variables=["chat_history", "human_input"],
    template=template
)
memory = ConversationBufferMemory(memory_key="chat_history")
```

```
llm_chain = LLMChain(
    llm=OpenAI(),
    prompt=prompt,
    verbose=True,
    memory=memory,
)
```

```
llm_chain.predict(human_input="Hi there my friend")
```

Skip to main content

```
> Entering new LLMChain chain...
Prompt after formatting:
You are a chatbot having a conversation with a human.


Human: Hi there my friend
Chatbot:

> Finished LLMChain chain.
```

```
' Hi there, how are you doing today?'
```

```
llm_chain.predict(human_input="Not to bad - how are you?")
```

```
> Entering new LLMChain chain...
Prompt after formatting:
You are a chatbot having a conversation with a human.


Human: Hi there my friend
AI:  Hi there, how are you doing today?
Human: Not to bad - how are you?
Chatbot:

> Finished LLMChain chain.
```

```
" I'm doing great, thank you for asking!"
```