

Deep Lake

Contents

- Deep Lake datasets on cloud or local

This notebook showcases basic functionality related to Deep Lake. While Deep Lake can store embeddings, it is capable of storing any type of data. It is a fully fledged serverless data lake with version control, query engine and streaming dataloader to deep learning frameworks.

For more information, please see the Deep Lake documentation or api reference

```
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.text_splitter import CharacterTextSplitter
from langchain.vectorstores import DeepLake
from langchain.document_loaders import TextLoader
```

```
from langchain.document_loaders import TextLoader
loader = TextLoader('../state_of_the_union.txt')
documents = loader.load()
text_splitter = CharacterTextSplitter(chunk_size=1000, chunk_overlap=0)
docs = text_splitter.split_documents(documents)

embeddings = OpenAIEmbeddings()
```

```
db = DeepLake.from_documents(docs, embeddings)

query = "What did the president say about Ketanji Brown Jackson"
docs = db.similarity_search(query)
```

```
Evaluating ingest: 100%|██████████| 41/41 [00:00<00:00
```

```
print(docs[0].page_content)
```

[Skip to main content](#)

In state after state, new laws have been passed, not only to suppress the vote, but to subvert entire elections.

We cannot let this happen.

Tonight. I call on the Senate to: Pass the Freedom to Vote Act. Pass the John Lewis Voting Rights Act. And while you're at it, pass the Disclose Act so Americans can know who is funding our elections.

Tonight, I'd like to honor someone who has dedicated his life to serve this country: Justice Stephen Breyer—an Army veteran, Constitutional scholar, and retiring Justice of the United States Supreme Court. Justice Breyer, thank you for your service.

One of the most serious constitutional responsibilities a President has is nominating someone to serve on the United States Supreme Court.

And I did that 4 days ago, when I nominated Circuit Court of Appeals Judge Ketanji Brown Jackson. One of our nation's top legal minds, who will continue Justice Breyer's legacy of excellence.

Deep Lake datasets on cloud or local

By default deep lake datasets are stored in memory, in case you want to persist locally or to any object storage you can simply provide path to the dataset. You can retrieve token from app.activeloop.ai

```
!activeloop login -t <token>
```

```
/bin/bash: -c: line 0: syntax error near unexpected token `newline'
/bin/bash: -c: line 0: `activeloop login -t <token>'
```

```
# Embed and store the texts
dataset_path = "hub://{username}/{dataset_name}" # could be also ./local/path
(much faster locally), s3://bucket/path/to/dataset, gcs://, etc.
```

```
embedding = OpenAIEmbeddings()
vectordb = DeepLake.from_documents(documents=docs, embedding=embedding,
dataset_path=dataset_path)
```

Evaluating ingest: 100%|██████████| 4/4 [00:00<00:00

[Skip to main content](#)

```
query = "What did the president say about Ketanji Brown Jackson"
docs = db.similarity_search(query)
print(docs[0].page_content)
```

In state after state, new laws have been passed, not only to suppress the vote, but to subvert entire elections.

We cannot let this happen.

Tonight. I call on the Senate to: Pass the Freedom to Vote Act. Pass the John Lewis Voting Rights Act. And while you're at it, pass the Disclose Act so Americans can know who is funding our elections.

Tonight, I'd like to honor someone who has dedicated his life to serve this country: Justice Stephen Breyer—an Army veteran, Constitutional scholar, and retiring Justice of the United States Supreme Court. Justice Breyer, thank you for your service.

One of the most serious constitutional responsibilities a President has is nominating someone to serve on the United States Supreme Court.

And I did that 4 days ago, when I nominated Circuit Court of Appeals Judge Ketanji Brown Jackson. One of our nation's top legal minds, who will continue Justice Breyer's legacy of excellence.

```
vectordb.ds.summary()
```

```
Dataset(path='./local/path', tensors=['embedding', 'ids', 'metadata', 'text'])
```

tensor	htype	shape	dtype	compression
embedding	generic	(4, 1536)	None	None
ids	text	(4, 1)	str	None
metadata	json	(4, 1)	str	None
text	text	(4, 1)	str	None

```
embeddings = vectordb.ds.embedding.numpy()
```