Introduction to Jina

A cloud-native neural search framework to build SOTA and scalable deep learning search applications

What is Cross Modal and Multi Modal?

Types of Data Modality

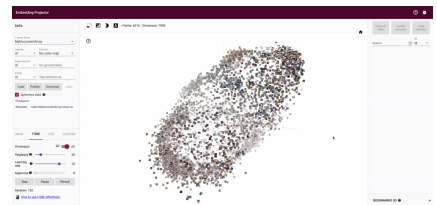
- Single Modality -For example, a tweet is a modal of type “text”; a photo is a modal of type “image”; a video is a modal of type “video”; etc.
- Multi - Modality - For example, a tweet often contains not only text, but also images, videos, and links. A video often contains not only video frames, but also audio and text (e.g., subtitles).
- Cross - Modality - For example, learning from images and text where the images and text are not necessarily about the same thing.

Potential Applications

- a cross-modal machine learning algorithm could be used to automatically generate descriptions of images (e.g., for blind people)
- A search system could use a cross-modal machine learning algorithm to search for images by text queries (e.g., “find me a picture of a dog”)
- A text-to-image generation system could use a cross-modal machine learning algorithm to generate images from text descriptions (e.g., “generate an image of a dog”)
- a cross-modal system could be used to improve the accuracy of facial recognition algorithms by using information from other modalities such as body language or voice
- For example, if an image recognition algorithm is having difficulty identifying an object due to poor lighting conditions, information from another modality such as sound could be used to help identify the object

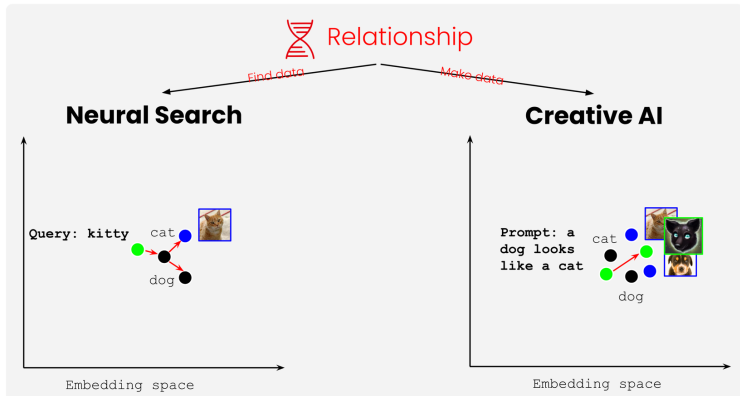
Broad Families of Cross-Modal Applications

- **Neural Search**



- **Creative AI**

Relationship between Neural Search and Creative AI



The relationship between or within the modalities ties Neural Search and Creative AI together

What is Jina?

Jina is the framework for helping you to build cross-modal and multi-modal systems on the cloud. With Jina, developers can easily build high performant cloud native applications, services and systems in production

Why do we need Jina?

Problems of building such a system

- lack of design pattern for such system
- gap between a proof-of-concept and a production system
- long go-to-market time

Jina is a solution to address above problems by providing a consistent design pattern for cross-modal/multi-modal systems with the latest cloud native technologies.

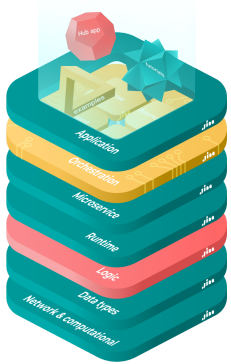
Taste of Jina

```python
from jina import DocumentArray, Executor, Flow, requests

class MyExec(Executor):
    @requests
    async def foo(self, docs: DocumentArray, **kwargs):
        for d in docs:
            d.text += 'hello, world!'

f = Flow().add(uses=MyExec).add(uses=MyExec)

with f:
    r = f.post('/', DocumentArray.empty(2))
    print(r.texts)
```

```
──────────────────────────  Flow is ready to serve! ──────────────────────────
╭──────────────  Endpoint ──────────────╮
│        Protocol                  GRPC  │
│          Local          0.0.0.0:52570  │
│        Private    192.168.1.126:52570  │
│         Public   87.191.159.105:52570  │
╰────────────────────────────────────────╯

['hello, world!hello, world!', 'hello, world!hello, world!']
```

Design Principles

We only need to know three concepts to master Jina. They are Document, Executor and Flow.
A full-fledged cross-modal/multi-modal system is a combination of the following seven layers:



Developer Focus

Jina Ecosystem