DALL-E: Text to Image Generation

TL;DR

- Goal: given the description of an image, create an image matching the description
- The method will use the Language Modeling objective ("predict the next token")
    - Same as we use for Text completion
- We have already learned how to represent an image as a sequence of "image" tokens
- We train a "predict the next token" model on examples of the form
    - sequence of text tokens describing the image
    - a separator token
    - a **prefix** of a sequence of image tokens for an image matching the description
- At inference time: pass in only the text description tokens and the separator
    - The model should create a sequence of image tokens matching the description !

```
 Training:  <text token> <text token> ... <text token> [SEP] <image toke
n> <image token> ...
 Inference: <text token> <text token> ... <text token> [SEP]
```

References

[paper (https://arxiv.org/pdf/2102.12092.pdf)](https://arxiv.org/pdf/2102.12092.pdf)

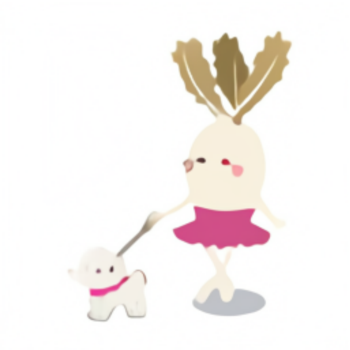
[OpenAI DALL-E 2 announcement (https://openai.com/dall-e-2/)](https://openai.com/dall-e-2/)

DALL-E is a generative model for creating Images, conditioned on Text input.

That is:

- Given some Text that describes the output
- DALL-E will create an image based on the Text input

DALL-E: Text to Image
Text input: "An illustration of a baby daikon radish in a tutu walking a dog"
Image output:



Before we began our journey into Generative ML, imagining how to achieve such a result would have been puzzling.

But, in concept, it follows a familiar path

- A Generative Language Model (e.g., GPT) is able to solve a "predict the next token" task
    - Giving a sequence of tokens
    - Generate a token likely to follow
- "Text to Text" as a universal API
    - turn *your* task into a type of translation of
        - an input sequence of Source tokens
        - to to an output sequence of tokens

The key prerequisite to DALL-E (in addition to a Language Model)

- representation of a (flattened) Image as a sequence of (discrete) tokens

We have *already* encountered such a representation

- A Discrete Variational Autoencoder (dVAE) such as the VQ-VAE represents an Image
- as a higher dimensional Tensor (e.g, 2D grid)
- of discrete values
    - indexes into a finite "codebook" of latent vectors
    - the codebook implements an embedding of Image "elements" into a finite set of vector encodings

Hopefully: the idea underlying DALL-E has come into focus

- We train a Language Model on a sequences of generalized tokens
    - First part of sequence are Text tokens
    - Separated by an "end of Text" token
    - Second part of sequence are Image tokens
- The Language Model learns
    - not only what "words" follow the prefix of a sequence of word tokens
    - but what Image tokens follow the "end of Text"

Voila !

Text to Image is just another use of the "Text to Text" API

- akin to translating from one language to another

Details

A dVAE is trained

- Encoder reduces a 3D (RGB) image to smaller (32×32) spatial grid of latent vectors
- the dVAE learns a codebook
    - of 8192 elements (the Image Vocabulary size)
- Using the dVAE Encoder, an image is translated into a (32×32) spatial grid

A Text Encoder is used

- BPE encoding of tokens
- Text Vocabulary size is $16,384$

The Language Model is trained

- Examples are (Text, Image) pairs
    - Obtained from images with captions
- Given training example $(\text{text}^{\mathbf{(i)}}, \text{image}^{\mathbf{(i)}})$
- Text $\text{text}^{\mathbf{(i)}}$ is converted to a sequence using the Text Encoder
    - maximum text sequence length limited to 256 Text tokens
- Image $\text{image}^{\mathbf{(i)}}$ is converted to a (32×32) spatial grid
    - which is flattend to a sequence of $1024 = 32 * 32$ Image tokens

The training example is thus a sequence of

- 256 Text tokens
- 1024 Image tokens

The Language Model learns to Autoregressively model the training sequences

- Using Decoder style Transformer

Here is what a training example that mixes text and an image looks like:

Training example: Representing mixed Text + Image as a sequence

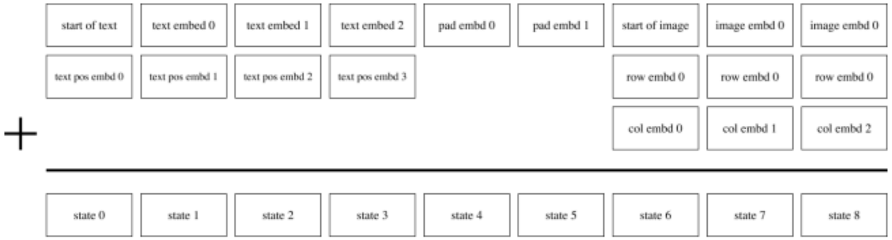


Figure 10. Illustration of the embedding scheme for a hypothetical version of our transformer with a maximum text length of 6 tokens. Each box denotes a vector of size $d_{\text{model}} = 3968$. In this illustration, the caption has a length of 4 tokens, so 2 padding tokens are used (as described in Section 2.2). Each image vocabulary embedding is summed with a row and column embedding.

Notice the positional embeddings

- position in text
- row/column position for image

A Language Model is not deterministic: running the same input multiple times can generate different outputs

- LLM creates a probability distribution over token vocabulary
    - we *sample* a token from this distribution as the LLM "next token" output

DALL-E uses CLIP to rank the outputs

- CLIP finds the images that are the least distance from the Text

Interesting results and observations

[OpenAI announcement of DALL-E 2 (https://openai.com/dall-e-2/)](https://openai.com/dall-e-2/)

DALL-E seems to have learned to generalize

- the "baby daikon radish wearing a tutu walking a dog"
    - even though radishes don't wear clothing, DALL-E positions the tutu around the "waist" of the radish
    - the radish is in a ballerina (dancer who wears a tutu) pose
    - the dog is on a leash (as are dogs being walked)
    - even though a radish does not have arms, DALL-E creates an arm to hold the leash
- Different styles of Image are created according to the text "illustration", "sketch", "photo", "painting in the style of Picasso"

Image editing via Text

- Given the text sequence "the exact same cat on the top as a sketch at the bottom"
- Followed by Image tokens (half the length of the total Image sequence length) encoding a cat
- Will generate the remaining available Image tokens of the cat in the style of a sketch

Doesn't always understand

- "tree bark" (the "skin" of a tree) is **not** an animal barking at a tree

Social concerns

The DALL-E 2 preview included a model card (https://github.com/openai/dalle-2-preview/blob/main/system-card_04062022.md#bias-and-representation) describing possible Risks and Limitations.

Among them are

- Inappropriate training examples (https://github.com/openai/dalle-2-preview/blob/main/system-card_04062022.md#model-training-data)
- "Signing" to indicate DALL-E generated image (https://github.com/openai/dalle-2-preview/blob/main/system-card_04062022.md#signature-and-image-provenance)
- Biases present in training data (https://github.com/openai/dalle-2-preview/blob/main/system-card_04062022.md#bias-and-representation)

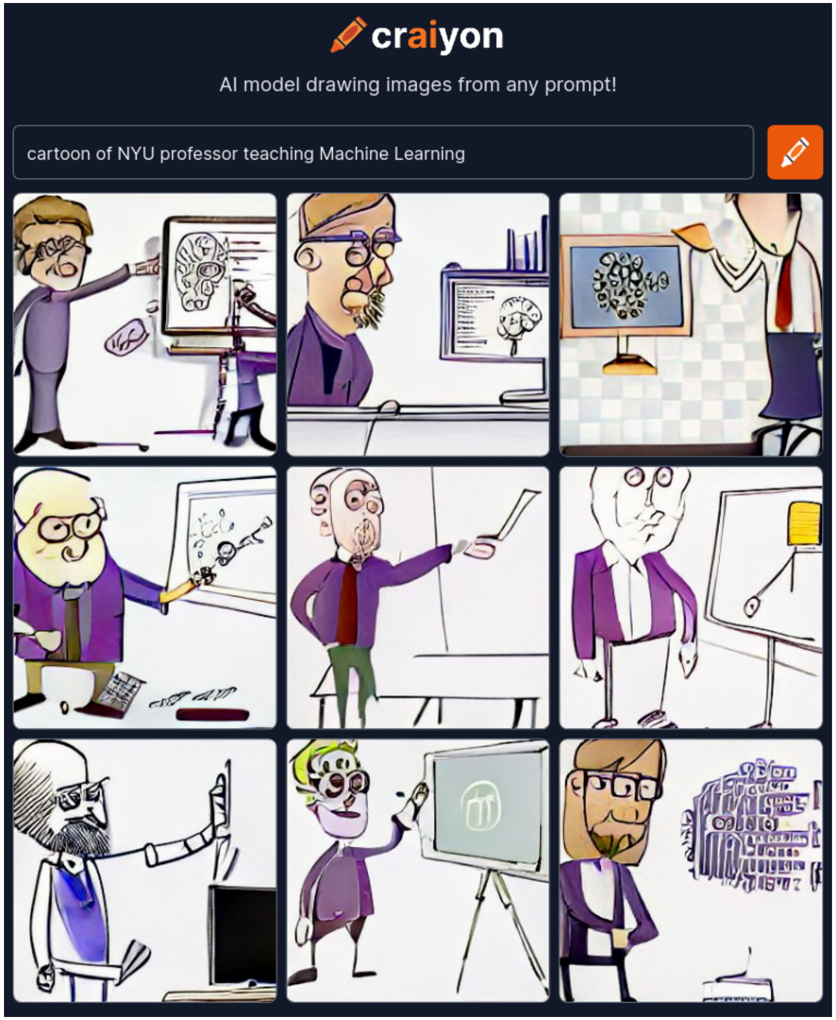
Playing with DALL-E

There is a waiting list (https://labs.openai.com/waitlist) to get access to the official version of DALL-E.

However there are some Open Source replicas (trained on much less data)

- Craiyon (https://craiyon.com)
    - Advertisement supported :-(
- Huggingface.co (https://huggingface.co/spaces/dalle-mini/dalle-mini)
    - Moving to Craiyon

Craiyon
Text input: "Cartoon of NYU professor teaching Machine Learning"
Image output:



AI model drawing images from any prompt!

cartoon of NYU professor teaching Machine Learning

- It gets the NYU colors right !
    - Princeton professors wear orange

In [2]:

```
print("Done")
```

```
Done
```