Week 6 Assignment

Generative AI

Saarland University – Winter Semester 2024/25

Martínez
7057573
cama00005@stud.uni-saarland.de

1 Exercise Assignment: E1

When it comes to the similarities of the Image Transformer [1] and DALL-E 1 [2] models, we can highlight the following:

- They are both transformer-based generative models and they both employ autoregressive mechanisms to generate image data sequentially.
- The Image Transformer models images as sequences of image patches, similar to how DALL-E 1 models images as sequences of image tokens.
- They both use discrete representations with respect to how they tokenize image data. On one hand, the Image Transformer treats image patches as discrete tokens in a latent space for representing and generating images, which take values from a fixed visual codebook[1] learnt with a discrete Variational Autoencoder (dVAE). On the other hand, DALL-E 1 employs the same principle but with a dVAE, only this time for learning a discrete latent space for both images and text.

Whereas the differences between the two models are as follows:

- The Image Transformer operates exclusively on image patch sequences, unlike DALL-E 1 which encodes text as tokens and concatenates them with image tokens.
- The Image Transformer focuses purely on reconstructing image sequences.Whereas DALL-E 1 combines text and image sequences, using a single stream to conditionally generate images from textual prompts .
- As a consequence of their different objectives, the Image Transformer and DALL-E 1 have distinct training datasets: the former with image datasets to enable image generation/reconstruction, and the latter with paired text-image datasets to enable text-to-image generation.

2 Exercise Assignment: E2

In the diffusion model introduced in [3], the forward process starts with a noise-free image represented as $\mathbf{x}_0 \sim q(\mathbf{x}_0) :=$ data distribution, and adds Gaussian noise to an image $\mathbf{x}_0$ over T steps, generating a noised version $\mathbf{x}_t$, via a Markov chain:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}),$$

[1] The vocabulary for the latent visual tokens, e.g., patterns.

where β_t are fixed hyperparameters (that is, not learnt), and represent the noise level at step t. Then, we can express the complete forward process as:

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}).$$

On the other hand, starting from a noised image $\mathbf{x}_T \sim p(\mathbf{x}_T) := \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$, the reverse process involves a Markov chain which gradually removes noise to recover $\mathbf{x}_0$ with learnable parameters θ:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \mathbf{\Sigma}_\theta(\mathbf{x}_t, t)),$$

where μ_θ is a learned mean function and $\mathbf{\Sigma}_\theta$ is the variance (which Jo. et al. [3] proposed to fix as $\mathbf{\Sigma}_\theta = \beta_t \mathbf{I}$). Analogous to the forward process, the full reverse process is:

$$p_\theta(\mathbf{x}_0 \mid \mathbf{x}_{1:T}) = \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t).$$

By reparameterization, $\mathbf{x}_t$ can be sampled directly from $\mathbf{x}_0$ at any step t as:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}),$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.

3 Exercise Assignment: E3

First of all, CLIP [4] is a multimodal model that learns to align text and image representations in a shared latent space. It uses a Transformer-based Text encoder and a ViT[2] (or ResNet) for the Image encoder. It was trained with over 400 million text-images pairs and can be readily applied for zero-shot image classification to any domain, with its performance rivaling State-of-Art models trained specifically for that domain.

In LLaVA [5] (i.e., Large Language and Vision Assistant), CLIP [4] serves as the backbone for aligning visual and textual modalities, by connecting CLIP's image encoder with a Transformer-based Language Model. The CLIP's image encoder processes visual inputs into latent representations, which are then concatenated with text embeddings from the Language Model, allowing for further fine-tuning to align with any multimodal tasks (e.g., text generation, image captioning, etc.).

4 Exercise Assignment: E4

In DALL-E 2 [6], since CLIP's text encoder maps textual inputs into a latent space shared with the image encoder (thus learning a prior that converts text embeddings into image embeddings), we leverage this shared space to generate images from text prompts. This process involves three main steps:

1. Encoding the text prompt into the shared latent space via CLIP.
2. Sample an image from the latent space, which as a byproduct of CLIP's architecture and training, ensures that the generated images are semantically consistent with the text prompt.
3. Decoding the image back into the image space.

5 Exercise Assignment: E5

As for similarities:

- Both aim to generate high-quality images conditioned on text prompts, and thus employ paired text-image datasets for training.
- Both models are transformer-based.

[2] Vision Transformer.

Regarding differences:

- DALL-E 1 operates on discrete tokens derived via dVAE, while DALL-E 2 operates on a continuous latent space aligned by CLIP embeddings.
- For generation, DALL-E 1 uses an autoregressive model, while DALL-E 2 uses a diffusion model.
- DALL-E 1 produces lower-resolution outputs, namely 256×256 pixels, while DALL-E 2 generates higher-resolution and semantically consistent images, more recently up to 1024×1024 pixels.

6 Individual Assignment: I1

The generated keywords match the input image. The keywords: `Robot`, `Maze`, `Colorful`, and `Puzzle` appear in all three files. On the other hand, the other keywords include words which qualitatively match the given input image, such as `Geometric` and `Toy`.

7 Individual Assignment: I2

The input text prompt was: `robot, maze, walls, abstract, educational, colorful, 2D`. I would say all three of the generated images qualitatively match this description. Nevertheless, the image `I2_c.png` did not contain `walls`, whereas `I2_a.png` has some abstract, not full-sized walls, and `I2_b.png` had the best wall representation and is also the only one that looks like a `2D` painting (as required by the prompt). The other two emphasize a certain 3D aspect.

8 Individual Assignment: I5

For the `ASCII` representation format, the three outputs show that the model extracted the correct grid elements (avatar, goal, walls) with 100% accuracy, namely that the `AVATAR` is in `0:0:east`, the `GOAL` is in `2:2`, and the `WALLS` are in `[0:3, 1:1, 1:3, 2:3, 3:0, 3:1, 3:2, 3:3]`[3].

On the other hand, for the image representation format,

⇒ `AVATAR`: Position $(3/3)$, Orientation $(0/3)$.

⇒ `GOAL`: $2/3$.

⇒ `WALLS`: $0/3$.

In conclusion, the `ASCII` representation format appears to be easier for the model to process accurately, as it achieved a 100% accuracy in extracting the grid elements. In contrast, the image representation format proved to be challenging for the model. This intuitively makes sense, since GPT-4o, being a Transformer-based Large Language Model, is designed to process text data very well due to their mostly text-based training data, and providing `ASCII` representation format, we are giving the model that in which it is best at.

9 Individual Assignment: I8

For the `ASCII`-based representation of the grid, the model in all three calls successfully identifies that the correct sequence of moves is: `move_forward, move_forward, turn_right, move_forward, move_forward`.

That same sequence of moves was found only once by the image-based input (`I7_c.txt`). On another call, it found another correct sequence of moves though slightly longer (`I7_a.txt`):

```
1. move_forward (move to position (0,1))
2. turn_right (now facing South)
3. move_forward (move to position (1,1))
```

[3] In various markdown formats, but equal final result.

```
    4. move_forward (move to position (2,1))
    5. turn_left (now facing East)
    6. move_forward (move to position (2,2))
```

Finally, the other call (`I7_b.txt`) did not find the correct sequence of moves. It incorrectly proposed: `move_forward, move_forward, turn_right, move_forward, turn_left, move_forward, move_forward`.

This again proves that providing a text-based representation of the grid, such as the `ASCII` format, is more reliable for the model to process and generate the correct sequence of moves.

10 Individual Assignment: I11

For the `ASCII` representation format, the GPT-4o model in all three calls generated a grid which correctly captures the input elements. On the other hand, DALL-E generated in all three calls images which definitely did not match the expected grid with its input elements. It simply generated random colorful and game-like images with some resemblance to the input elements randomly placed, and not in a structured way.

This again proves the superiority of text-based input and output representations for the GPT-4o model. In contrast, DALL-E is a generative model that generates images from text prompts, and via these experiments, proved not to be suitable for outputting structured grid-like images from text prompts with correctly placed elements.

Acknowledgements

This week's slides and listed references.

References

[1] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer, 2018.

[2] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[5] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[6] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.