DALL ·E

This is a REALLY short summary of the DALL ·E paper as it is a dense paper in terms of complex computational details. For more detailed information check the paper. The training process is explained with detail in the original paper.

ABSTRACT:

Simple approach for the text-to.image task based on a transformer that autoregressively models the text and image tokens as a single stream of data.

INTRODUCTION:

The authors demonstrate that training a 12-billion parameter autoregressive transformer on 250 million image-text pairs results in a flexible, high fidelity generative model of images controllable through natural language.

The resulting system achieves high quality image generation on MS-COCO dataset *zero-shot*, without using any of the training labels.

METHOD:

The goal is to train a transformer to autoregressively model the text and image tokens as single stream of data. However, using pixel directly as image tokens would require an inordinate amount of memory for high-resolution images. Likelihood objectives tend to prioritize modeling short-range dependencies between pixels, so much of the modeling capacity would be spent capturing high-frequency details instead of the low-frequency structure that makes objects visually recognizable to us.

Two stage training procedure:

- **Stage 1:** Train a discrete variational autoencoder (dVAE) to compress each 256x256 RGB image into a 32x32 grid of image tokens. This reduces the context size of the transformer by a factor of 192 without a large degradation in visual quality.
- **Stage 2:** Concatenate up to 256 BPE-encoded text tokens with the 32x32=1024 image tokens, and train an autoregressive

transformer to model de joint distribution over the text and image tokens.

The overall procedure can be viewed as maximizing the evidence lower bound (ELB) on the joint likelihood of the model distribution over images x, captions y, and the tokens z, for the encoded RGB image.

- **Stage One: Learning the Visual Codebook**

In the first stage of training the ELB is maximized with respect to ϕ and θ, which corresponds to training a dVAE on the images alone.
The relaxed ELB is maximized using Adam with exponentially weighted iterate averaging.
Important considerations for a stable training:

- Specific annealing schedules for the relaxation temperature and step size. We found that annealing τ to 1/16 was sufficient to close the gap between the relaxed validation ELB and the true validation ELB with qϕ instead of qϕ^τ.
- The use of 1x1 convolutions at the end of the encoder and the beginning of the decoder. We found that reducing the receptive field size for the convolutions around the relaxation led to it generalizing better to the true ELB.
- Multiplication of the outgoing activations from the encoder and decoder resblocks by a small constant, to ensure stable training at initialization.

Also, increasing the KL weight to β=6.6 promotes better codebook usage and ultimately leads to a *smaller* reconstruction error at the end of training.

- **Stage Two: Learning the Prior**

In the second stage, ϕ and θ are fixed, and the prior distribution over the text and image tokens is learned by maximizing the ELB with respect to ψ.

The transformer is a decoder-only model in which each image token can attend to all text tokens in any one of its 64 self-attention layers.

The length of a text caption is limited to 256 tokens, though it is not totally clear what to do for the *"padding"* positions in between the last text token and the start-of-image token.

Cross-entropy losses are normalized for the text and image tokens by the total number of each kind in a batch of data.

- **Data Collection:**

To scale up to 12-billion parameters, a dataset of a similar scale to JFT-300M is created by collecting 250 million text-images pairs from the internet.

- **Mixed Precision Training:**

To save GPU memory and increase throughput, most parameters, Adam moments, and activations are stored in 16-bit precision. We also use activation checkpointing and recompute the activations within the resblocks during the backward pass.

- **Distributed Optimization:**

Our 12-billion parameter model consumes about 24 GB of memory when stored in 16-bit precision, which exceeds the memory of a 16 GB NVIDIA V100 GPU.
On the cluster used to train the model, the bandwidth between machines is much lower than the bandwidth among GPUs on the same machine. This makes the cost of the operation used to average the gradient among the machines (all-reduce) the main bottleneck during training. It was possible to drastically reduce this cost by compressing the gradients using PowerSGD.
PowerSGD replaces the large communication operation for an uncompressed parameter gradient with two, much smaller communication operations for its low-rank factors.
Details to get PowerSGD to perform well at scale:

- Saving memory by accumulating the gradient into the error buffers during backpropagation, rather than allocating separate buffers.

- Minimizing instances in which we zero out the error buffers.
- Improving numerical stability by using Householder orthogonalization instead of Gram-Schmidt, together with the addition of a small multiple of the identity matrix to the input.
- Avoiding underflow by using a custom 16-bit floating point format for the error buffers, their low-rank factors, and the all-reduce communication operations involving them.

- **Sample Generation:**

Re-rank the samples drawn from the transformer using a pretrained contrastive model. Given a caption and a candidate image, the contrastive model assigns a score based on how well the image matches the caption.

EXPERIMENTS:

- **Quantitative Results:**

Training the transformer on the tokens from the dVAE encoder allows us to allocate its modeling capacity to the low-frequency information that makes images visually recognizable to us. However, it also disadvantages the model, since the heavy compression renders it unable to produce high-frequency details.

- **Data Overlap Analysis:**

For each validation image, we find the closest image in the training data using a contrastive model specifically trained for this task. We then sort the images in descending order by closeness to their nearest matches in the training data. After inspecting the results by hand, we determine the images to remove by manually selecting a conservative threshold designed to minimize the false negative rate.

- **Qualitative Findings:**

The model is able to generalize in ways that were not anticipated. This suggests that it has developed a rudimentary ability to compose unusual concepts at high levels of abstraction.
Appears to be capable of combinational generalization, such as when rendering text. However, the model performs inconsistently on the task.
To a limited degree of reliability, it is found that the model is capable of zero-shot image-to-image translation controllable by natural language.

This works with several other kinds of transformations including image operations and style transfer. Some transformations such as those that involve only changing the color of the animal, suggest that the model is capable of performing a rudimentary kind of object segmentation.

CONCLUSION:

Scale can lead to improved generalization, both in terms of zero-shot performance relative to previous domain-specific approaches, and in terms of the range of capabilities that emerge from a single generative model.

EXTRA INFORMATION:

The model was trained using 1024, 16GB NVIDIA V100 GPUz and a total batch size of 1024, for a total of 430.000 updates.