Pixel-Level BPE Encoding for Auto-Regressive Image Generation

Anonymous Authors[1]

Abstract

The topic of image generation is very hot today. The most impressive results are obtained by GANs and diffustion-based models, or codebook-level autoregression (VQVAE+GPT). Also it is possible to generate images autoregressively on pixel level (image-GPT), however it is too memory-consuming approach due to the resulting total length of modeling pixel-sequence and hence this method is not very popular.

In our research we propose to adopt Byte-Pair-Encoding (BPE) for pixel-level encoding to drastically reduce the length of modeled sequence. Our experiments demonstrate that auto-regressive image generation might be understudied due to the lack of optimal sequence encoding techniques for images.

1. Introduction

There are plenty approaches to image generation: GANs, auto-encoders, auto-regression over discrete features, diffusion models, energy models and optimal transport. One of understudied approaches is auto-regressive pixel-level generating. The main challenge in this kind of models is the length of the modeling sequences. For example for 128x128 RGB image the pixel sequence length is equal to

$$128 \cdot 128 \cdot 3 = 49152$$

which is infeasible to model with standard auto-regressive architectures (RNN, GPT). But in contrast to extremely large sequence length the size of vocabulary is fairly small (255 values) compared to text-based models. We suggest to exploit this fact and to improve the trade-off between the sequence length and the vocabulary size.

We conducted several experiments with different types of sequence tokenization: Byte-Pair-Encoding (BPE), byte-level BPE, color-adapted BPE and spacial-adapted BPE. The proposed tokenization methods efficiently squeeze the length of the pixel sequences and give a green-light for further study of autoregressive pixel-level image generation. In addition we provide experimental results on CelebA and CIFAR datasets.

Figure 1. One of the worst examples :)

Main contributions of this paper:

- Image-GPT implementation
- Suggested several BPE-based tokenization techniques for pixel sequence length squeezing.

2. Related Work

Transformer-based (Vaswani et al., 2017) models are extremely successful in natural language generation and understanding fields. GPT (Alec Radford, 2018) demonstrated human-level performance text generating and zero-shot tasks via prompt engineering. There were attempts in using GPT architecture for image generation, which can be divided into two groups: discrete feature based regression DALLE (Ramesh et al., 2022) or pixel-level regression iGPT (Chen et al., 2020). The second type of models is not fairy popular, as it is too memory-expensive due to the length of the pixel context. However, iGPT model demonstrated decent results in low-resolution image generation and downstream tasks over contextualized features. From the other side, in NLP there are plenty of methods for sequence length compression — different tokenization techniques, which expoit the precomputer merge dictionaries for optimal encoding of words or byte groups. One of the most efficient methods is Byte-Pair-Encoding (Shibata et al., 1999). In GPT models it is used special modification of this algorithm which works at byte-level (blB) which is a one more step for optimal sequence squeezing.

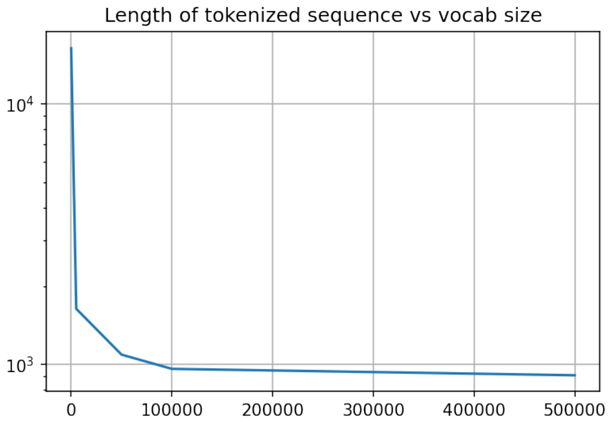

Figure 2. Dependence of image length vs BPE vocabulary size.

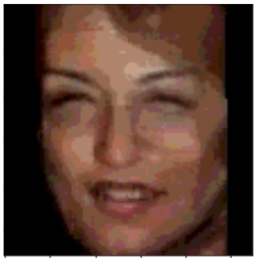

Figure 3. RGB-scale generated face.

3. Experiments

In our experiments we provide results for gray-scale and rgb images for the CELEBA dataset. First of all we convert images to text format by replacing each pixel value with a corresponding char symbol and separate lines of pixels by \n symbol. Then we train byte-level BPE tokenizer on this text file for further training of GPT.

3.1. Encoding Efficiency

To evaluate the sequence squeezing effect of BPE encoding for images we present mean sequence length for images for different vocabulary sizes. In the plot **??** you can see how BPE squeezes the image size.

3.2. Generated results

We tried to generate gray-scale and rgb images with different resolution, the results you can see below.

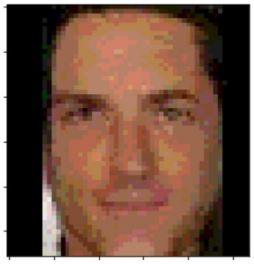

Figure 4. RGB generated face 128x128.

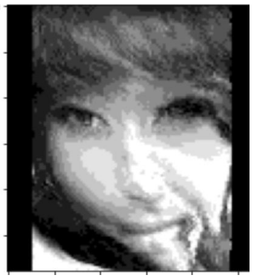

Figure 5. Gray-scale generated face 64x64.

References

Alec Radford, Jeffrey Wu, R. C. D. L. D. A. I. S. Language models are unsupervised multitask learners. 06 2018.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1691–1703. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20s.html.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents, 2022. URL https://arxiv.org/abs/2204.06125.

Shibata, Y., Kida, T., Fukamachi, S., Takeda, M., Shinohara, A., and Shinohara, T. Byte pair encoding: A text compression scheme that accelerates pattern matching. 09 1999.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., and Polosukhin, I. Attention is all you need. 06 2017.