A simple text to video model via transformer

Gang Chen
Vividity tech
info@vividitytech.com

Abstract

We present a simple text to video model based on Transformer. Since both text and video are sequential data, we encode both text and images into the same hidden space, which are further fed into Transformer to capture the temporal consistency. Considering the image signal may become weak in the long sequence, we introduce the U-Net to reconstruct the original image. Specifically, we use the $down$ module from U-Net to encode images, which are further input to transformer to predict next images. While the up module from U-Net is used to enhance the signal in the long sequence. We test our approach on UCF101 dataset and show it can generate promising videos.

1 Introduction

We have seen significant progress in text to video, unfortunately these methods either required the fixed video length for training or only generate the constrained videos such as the same background scene. In this work, we present an approach to train on (text, video) pair with varied lengths and scenes based on transformer framework. In addition, to handle the weak signal in the long sequence, we introduce U-Net to reconstruct the video data.

2 Model

We present a simple text to video model via transformer. In the following parts, we will introduce the language models and then discuss how to combine transformer and U-Net to generate videos from texts.

2.1 Background

Given a vocabulary $\mathcal{V}$ and an ordered sequence of symbols (or tokens) $(x_1, x_2, ..., x_n)$ with $x_i \in \mathcal{V}$, the language model [2] is defined as the joint probability over sequences of tokens $\mathcal{V}^n$, which is factorized into product of conditional probabilities

$$p(x_1, x_2, ..., x_n; \theta) = \prod_{1 \le i < n} p(x_i | x_1, x_2, ..., x_{i-1}; \theta) \tag{1}$$

where the vocabulary $\mathcal{V}$ is a large but finite set, and θ is the model parameter. $p(x_i|x_1, x_2, ..., x_{i-1})$ is conditional probability to predict next word given the previous sequences.

Many NLP problems can be formulated as $p(Y|X; \theta)$, where $X \in \mathcal{V}^n$ is the input sequence and $Y \in \mathcal{V}^m$ is the output. There have been many models that can compute these conditional probabilities, such as recurrent neural networks LSTM [3] and self-attention Transformer [4]. Especially the transformer architecture have significant improvements in the expressiveness and accuracy of models [5; 6]. To learn the model parameters θ, we can use cross entropy loss:

$$L_i(\theta) = -\log p(y_i) = -\log p(y_i | y_{<i}, X; \theta) \tag{2}$$

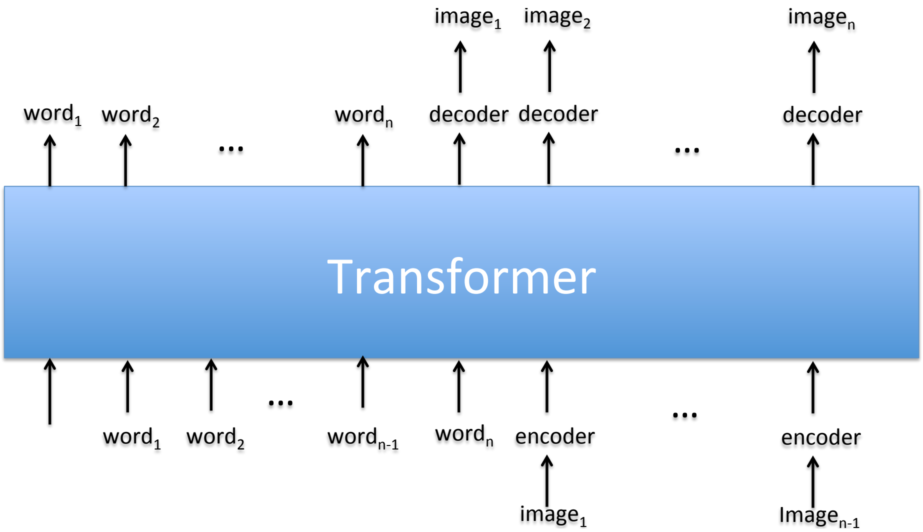

Figure 1: The figure shows the architecture to encode both text and video using transformer.

where only y_i holds and other tokens in $\mathcal{V}\backslash y_i$ are zeros. Then, the cross entropy error over the sequence of size m is:

$$L(\theta) = \sum_{i=1}^{m} L_i(\theta) = -\sum_{i=1}^{m} \log p(y_i) \tag{3}$$

While predicting the next symbol $\hat{y}_i \sim p(y_i|y_{<i}, X; \theta)$, we can either sample it or take a greedy strategy to select $\hat{y}_i$ with maximum probability. Note that the conditional probability $p(y_i) = p(y_i|y_{<i}, X; \theta)$ to predict next token is a discrete space $\mathcal{V}$, which is constrained by the vocabulary size. Compared to text, the image space is significant large and it is a much challenge problem to generate video from text.

2.2 The text to video model

What if $Y = \{y_1, y_2, ..., y_m\}$ is a video, not text? In this part, we will introduce how to extend Transformer to handle both text and videos.

To generate the video $Y = \{y_1, y_2, ..., y_m\}$ as the sequence of frames, we need to take the similar approach as we encode the token in the transformer framework in Fig. 1. In other words, we have an encoder function $e : y \to h$ and a decoder $d : h \to y$. And we also require the generated $\hat{Y}$ matches the ground truth Y. We can minimize the following square error:

$$loss = L(\theta) = \sum_{i=1}^{m} |y_i - \hat{y}_i|^2 \tag{4}$$

where $\hat{y}_i = d(h_{<i})$ and $h_{<i}$ is the last hidden output from the transformer. The square error loss above on images is similar to the cross entropy in language models.

Another assumption that we make is that the image signal may become weak in the long sequential video. To enhance the signal, we take a similar approach from diffusion model. Thus, We use U-Net [7] to construct images from its noised versions. The process is as follows:

1. create the noised data $\bar{y}_i = (1 - \beta_i)y_i + \beta_i\epsilon$, where β_i is the noise level coefficient
2. encode $h_i = e(\bar{y}_i)$ using the $down$ module from U-Net($down, up$), where we use the down module as our encoder
3. predict h_{i+1} using the transformer
4. decode the output $\hat{y}_{i+1} = d(h_{i+1})$ and reconstruct the $\hat{\bar{y}}_i = up(down(\bar{y}_i))$ with the up module from U-Net($down, up$)

5. update the model parameters by minimizing the loss equation 5 below

$$
\begin{aligned}
loss &= L(\theta) + \alpha J(\theta) \\
&= \sum_{i=1}^{m} |y_i - \hat{y}_i|^2 + \lambda \sum_{i=1}^{m} |y_i - \hat{\hat{y}}_i|^2
\end{aligned} \tag{5}
$$

where λ is the weight to balance two items.

3 Experimental results

We used the smallest version of GPT-2 with 124M parameters and U-Net , and tested our approach on UCF101 dataset [1].

4 Conclusion

We present a simple text to video model via transformer. In this work, we combine both Transformer and U-Net to handle sequential and long video datasets.

References

[1] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998 – 6008. Curran Associates, Inc., 2017.

[5] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[8] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, 2022.

[1] https://www.crcv.ucf.edu/data/UCF101.php