Handwriting Transformers

Peter Xingyu He*
Department of Computer Science
Columbia University
New York, NY 10020
xh2513@columbia.edu

Abstract

We are proposing a new way of generating hand-writing strokes offline with transformers. Current work in handwriting synthesis can largely be divided into 2 sets of methods. Sequential methods aim to predict the stroke coordinates given the target text. Image-based methods aim to generate an image containing the target text. While transformers have been compared with CNNs with respect to image-based methods in previous works, they have not been investigated with respect to sequential methods. To better adopt to the continuous data domain, we propose a novel convolution attention based attention model. [2]

1 Introduction

In the modern age of internet and computation, handwriting represents an loss art familiar to only to students. Personal handwritings are not only a intimate means of communication, they are also shown to have an impact on long-term memory. They are uniquely human comparing to the ASCII representation of symbols. Hence, many teaching machines how to write like a human can improve greatly human-computer interaction in the world where typed text dominates.

Some of the current challenges in handwriting generation is to decompose various features consisting the styles of human writers, such as style variation of each characters and the global style of a writer. The solutions to these fundamental problems are important to enabling few-shots style transfer, generating new styles, and writing unseen characters.

Most of the current methods heavily rely on using recurrent neural networks (RNN) Graves [2013] Kotani et al. [2020] Bertugli et al. [2021]. Since handwriting stroke sequences are naturally long, these models take advantage of various attention mechanisms to improve training. Since then, transformers Vaswani et al. [2017] has become the de-facto attention oriented mechanism thanks to its robustness to vanishing/exploding gradient problems. Its ability to be trained in a parallel fashion with respect to sequence length takes adventage of the modern hardwares acceleration for linear algebra operations.

In this paper, we explore novels ways to use transformers Vaswani et al. [2017] to generate realistic sequences of writing strikes. We first investigate the vanilla transformer model from Vaswani et al. [2017] with hyperparameters adopted to our specific applications. Then we develop additional attention mechanism to better address the continuous nature of the feature space. Our models only learn the stroke sequence of several specific characters and fail generalize to long sequences of ASCII characters.

*https://github.com/XingyuHe/

[2]https://github.com/XingyuHe/handwriting_transformer

Preprint. Under review.

2 Literature Review

The handwriting generation problem is approached in 2 ways. They are stroke-based online methods and image-based offline methods. For online stroke based methods, they require sequential data that from human handwriting usually with a digital pencil. This is usually represented by a time series of 2-D cartesian coordinate. Image-based methods use generative models to generate images that contain handwritten texts. Both approach face the same challenges of separating various features spaces that make a real writer's handwritings.

Graves [2013] first proposes a recurrent neural network model (RNN) with Long-Term Memory (LSTM). This model only has a single decoder and does not have encoders for input characters. It attempts to address the natural variation of handwriting of a single writer through a probabilistic approach. Gaussian Mixtures Models (GMM) are used to generate the output stroke sequences as well as to align attention over the input characters. The parameters of the Gaussian mixture models are projected from the hidden output of the LSTM cells at different depths. Aksan et al. [2018] note that a writer's style and content are strongly correlate. To disentangle style and content, their approach treat style and content as two separate latent variables. It uses Variational Recurrent Neural Networks (VRNN) to assume writing style, which results in fast and efficent handwriting generation at inference time. By interpolating style latent vectors, this approach also allows for potential generation of new handwriting styles.

Other works have investigated the use of image generation for handwriting generation. One advantage of image based style learning and generation is that images of handwritings are far more accessible than their stroke sequences. Additionally, writing on a digital tablets is often different from paper due to physical friction. Haines et al. [2016] proposes a dynamic programming approach to form handwritings by selecting existing handwritten characters from segmented source images. Chang et al. [2018] use CycleGAN Zhu et al. [2017] to learn mapping between source styles and target style in the Chinese language using unpaired writing images. Alonso et al. [2019] uses an additional character recognizer to learn handwriting generation from ASCII characters. Davis et al. [2020] uses both GAN and and autoencoder to incorporate the model the natural variation of handwritings in images. Current progress in vision transformers Dosovitskiy et al. [2020] has led to works in transformers and handwriting generation. Bhunia et al. [2021] transformer encoder-decoder architecture to learn mapping between ASCII characters and input writing styles. It is trained in an unsupervised fashion guided by a character recognizer and it is able to achieve few shots style transfers from unseen styles for unseen character sequences.

We investigate the effectiveness of transformers encoder-decoder architecture for stroke-sequence generation. Since our decoder outputs are in continuous space, this task can be viewed as a time series regression problem. Several works have been done in apply transformers to time series prediction. Zerveas et al. [2021] proposes an encoder only transformer as a framework for regression, classification, and time series problems. Wu et al. [2020] proposes and encoder-decoder architecture for influenza forecasting.

3 Model

3.1 Problem Formulation

We aim to train a model that takes a sequence of characters C of length L_c in ASCII representation. Our alphabet is of sizes 64 and includes punctuation and alphanumeric. Our output $X = (x_1, x_2, \ldots, x_t, \ldots)$ where

$$x_t \in \mathbf{R} \times \mathbf{R} \times \{0, 1\}.$$

The first 2 dimensions of x_t represents the Cartesian coordinates for the writing strokes. The last dimension of x_t represents a pen up event where 1 indicates that the stylus is lift up and 0 otherwise.

3.2 Vanilla Transformer Model

The encoder in the vanilla transformer model consists of an embedding layer, self-attention, and a position-wise feedforward layer. The decoder in the vanilla transformer model consists of an linear layer to project stroke sequences into the d_{model}, a self-attention layer, a multi-headed attention layer with the encoding from character sequences, and a position-wise feedforward layer. The self-attention

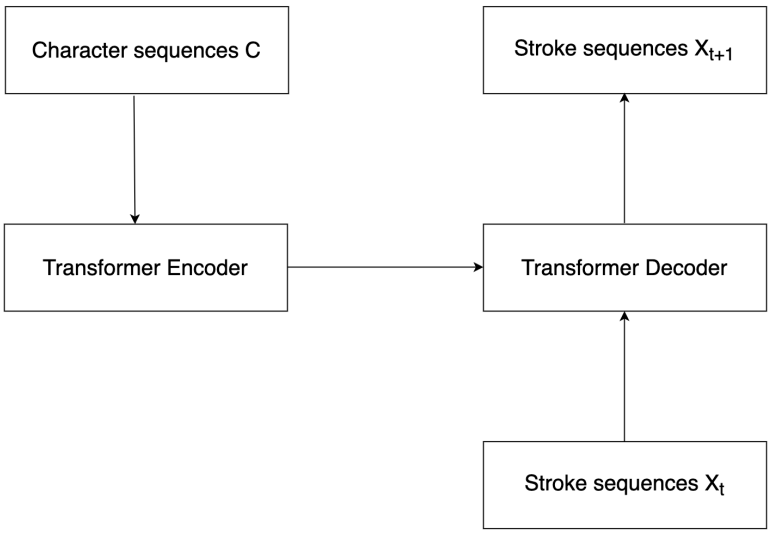

Figure 1: Vanilla Transformer Model

layer in the decoder helps predicting the new stroke points conditioning on the previous stroke sequences. The multi-headed attention layer helps with predicting new stroke sequences conditioning on the character sequence. In NLP application, the embedding layer of the transformer decoder is a table lookup mapping from word tokens to d_{model} vectors. Our data space requires mapping from continuous input to d_{model} vectors. Previous works in time series Wu et al. [2020] suggests to use a linear layer, so we are following their design decision. The elements in 3^{rd} dimension of the output x_{t+1} are then feed into a sigmoid layer to model the probabilistic nature of the pen-up event. Figure 1 demonstrates our vanilla transformer model architecture and Figure 2 depicts the decoder architecture.

3.3 Convolution Attention Transformer Model

The result of our vanilla transformer model isn't too successful, so we look for alternative model architectures. Transformers are successful in NLP where the data is discrete. We believe that the continuous nature of our data could cause the direct adoption of transformers to fail. Specifically, we question whether the self-attention layer in the vanilla transformer decoder is the right mechanism for attention mapping. One reason is that each token in NLP carries a lot of information on its own but a single data point x_t that contains the coordinate and the pen-up is not informational without a region of data points $x_{t-10} \dots x_{t-2}, x_{t-1}$. Paying attention to a single token provides useful guidance to what to predict next but paying attention to a single data point containing the coordinate and the pen-up event is that useful. To incorporate attention over a temporal region of previous stroke sequence, we first perform 1D convolution and linear layer separately on stroke sequence. The result of 1D convolution is projected into d_{model}. The self-attention layer is replaced with a multi-headed attention layer. The query of it remains the same as the self-attention layer, the key and value are the projected output from 1D convolution. We use 1D convolution to generate key and value because convolution layers preserver features invariant to positions. The remaining steps of the convolution attention transformer decoder remain the same. Figure 3 depicts the structure of the convolution attention transformer decoder.

3.4 Loss Function

For both of our models, we use 2 types of loss functions. First we calculate the mean squared error (MSE) the output sequence and the ground truth for coordinate values. Let $X_{t+1}^{(0:2)} \in \mathbf{T} \times \mathbf{R} \times \mathbf{R}$ be the predicted output and $Y^{(0:2)} \in \mathbf{T} \times \mathbf{R} \times \mathbf{R}$ be the ground truth for coordinate values. The we use

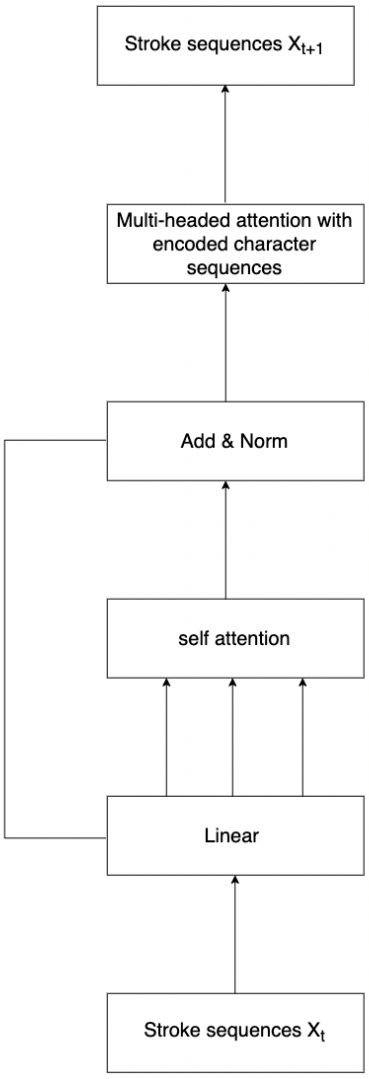

Figure 2: Vanilla Transformer Decoder

the binary cross-entropy (BCE) loss for pen-up events. Let $X_{t+1}^{(2)} \in \mathbf{R} \times \mathbf{R}$ and $Y^{(2)}$ be the predicted probability of the pen-up event and the ground truth pen-up event.

$$\mathcal{L}_{MSE} = MSE(X_{t+1}^{(0:2)}, Y^{(0:2)}),$$

$$\mathcal{L}_{BCE} = \sum_{n=1}^{T} -\left[y_n \cdot \log x_n + (1 - y_n) \cdot \log(1 - x_n)\right],$$

$$\mathcal{L}_{total} = \mathcal{L}_{BCE} + \mathcal{L}_{MSE}.$$

4 Result

We encounter difficulties training both models. Here we present our results from models configured in the following hyper parameters

BATCH SIZE = 64
LEARNING RATE = 0.0005
EPOCHS = 10000
OPTIMIZER = Adam.

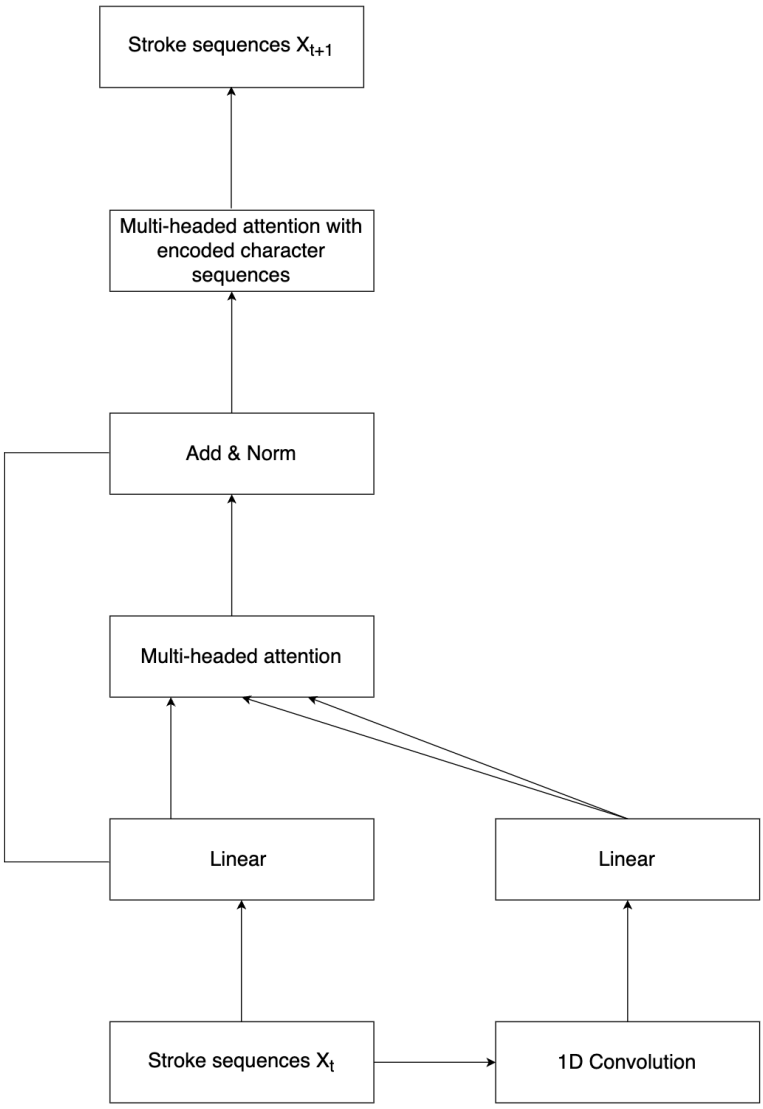

Figure 3: Convolution Transformer Decoder

Transformer parameters:

D MODEL = 64
N HEADS = 4
DIM FEEDFORWARD = 64
ENCODER LAYERS = 1
DECODER LAYERS = 3

Convolution parameters:

KERNEL SIZE = 10
STRIDE = 5
OUT CHANNELS=8

Figure 4 shows the results of vanilla transformer writing unseen characters "hello" after training for $1000, 5000, 10000$ epochs. Although the decoder captures the high slant of the "h" and "l", the writings are not very natural and not recognizable. Figure 5 shows the results of convolution

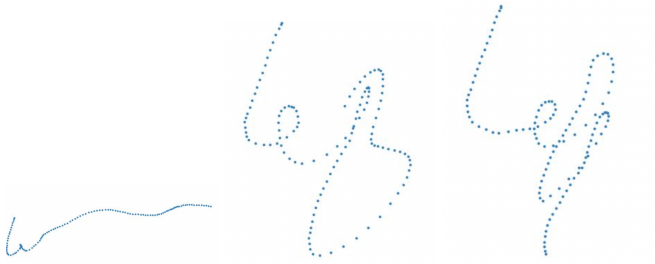

Figure 4: Vanilla Transformer Writing "hello"

Figure 5: Convolution Transformer Writing "hello"

transformer writing "hello" after training for $1000, 5000, 10000$ epochs. Figure 6 shows the result of the convolution transformer writing "b", "d", "f" after training 10000 epochs.

5 Discussion

Although the convolution operation improves the results from the vanilla transformers visually, the improvement only exist for certain characters despite completing 10000 epochs. The results are not as natural and humanistic as the results generated in Graves [2013] and other RNN-based approaches. The transformers are defintely learning specifics about individual characters but do not capture the writing of all characters. There are several points forward to futher investigate and debug the transformer system.

- Even though the overall loss on both training and validation set keeps decreasing, the results didn't improve significantly. This sugguests that the models are learning very little from the guidance of the loss function. Probabilistic based loss functions such as the GMM from Graves [2013] could be used for further investigation.
- Since the dimension of the input stroke sequence is much lower than d_model, the model could be over parameterized and and future investigation could use effective regularization methods. In this work, we apply a dropout rate of 0.1 to 0.2 and do not see any significant effect.
- Since transformers are trained in parallel, a transformer model could copy the last stroke point when predicting the future stroke point to achieve decent performance. Attention

Figure 6: Convolution Transformer Writing "b", "d", "f"

masking with several time steps behind could be used to force the model to learn the handwriting distribution.

6 Conclusion

In this work, we investigate the effectiveness of transformers at modeling humanistic handwriting. We find that direct adoption of the transformer model from Vaswani et al. [2017] yield results worse than the previous RNN methods. We propose a new convolution based transformers to address the continuous nature of stroke sequences and find improved but limited effects comparing to the vanilla transformer model. We provides several suggestions for future investigation based on our observation during model traning and inferencing.

References

E. Aksan, F. Pece, and O. Hilliges. Deepwriting: Making digital ink editable via deep generative modeling. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2018.

E. Alonso, B. Moysset, and R. Messina. Adversarial generation of handwritten text images conditioned on sequences. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 481–486. IEEE, 2019.

A. Bertugli, S. Calderara, P. Coscia, L. Ballan, and R. Cucchiara. Ac-vrnn: Attentive conditional-vrnn for multi-future trajectory prediction. *Computer Vision and Image Understanding*, 210:103245, 2021.

A. K. Bhunia, S. Khan, H. Cholakkal, R. M. Anwer, F. S. Khan, and M. Shah. Handwriting transformers. pages 1086–1094, 2021.

B. Chang, Q. Zhang, S. Pan, and L. Meng. Generating handwritten chinese characters using cyclegan. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 199–207. IEEE, 2018.

B. Davis, C. Tensmeyer, B. Price, C. Wigington, B. Morse, and R. Jain. Text and style conditioned gan for generation of offline handwriting lines. *arXiv preprint arXiv:2009.00678*, 2020.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

A. Graves. Generating sequences with recurrent neural networks, 2013. URL `https://arxiv.org/abs/1308.0850`.

T. S. Haines, O. Mac Aodha, and G. J. Brostow. My text in your handwriting. *ACM Transactions on Graphics (TOG)*, 35(3):1–18, 2016.

A. Kotani, S. Tellex, and J. Tompkin. Generating handwriting via decoupled style descriptors. In *European Conference on Computer Vision*, pages 764–780. Springer, 2020.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

N. Wu, B. Green, X. Ben, and S. O'Banion. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317*, 2020.

G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2114–2124, 2021.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.