GESTURE GENERATION USING TRANSFORMER DECODER ARCHITECTURE

Motive : Given (Having predicted) t clips each containing T frames along with the word embeddings, Φ_t from a pre-trained model, predicting $(t+1)^{th}$ clip, with the help of self-attention and attention w.r.t word embeddings.

Attention Block

The attention block consists of -

1. Masked Separable self-attention block
2. Frame level attention block with text embeddings

Masked Separable self-attention block

Let the real video be represented as v, with dimensions H x W x N x C, where H is the height and W is the width and C is the number of channels, and N is the total number of frames in the video clip. The whole video is divided into K different clips of T frames each, where $v \in R^{H \times W \times N \times C}$, and $v^t \in R^{H \times W \times T \times C}$

Separable attention operation is performed across Time, Height and Width

Across Time

Video clip v^j is reshaped to v^j_n such that $v^j_n \in R^{(H*W) \times T \times C}$, and with the current clip, v^t_n as query, and the other video clips being keys and values, it is attended with the other masked (K-t+1) blocks and unmasked t blocks of the video frames.
For each $j \in \{0, 1, \dots K-1\}$ and for each $i \in \{0, 1, \dots T-1\}$,

$$Q_1^{\,i} = v^t_n(i).q_1 \qquad \dots (1)$$

$$K_1^{\,i} = v^j_n(i).k_1 \qquad \dots (2)$$

$$V_1^{\,i} = v^j_n(i).v_1 \qquad \dots (3)$$

Where, $v^j_n(i)$ is the i^{th} 2D matrix of v^j_n across time dimension, and $\{q_1, k_1, v_1\} \in R^{C \times C'}$ are shared weights for all the T frames and $\{Q_1^{\,i}, K_1^{\,i}, V_1^{\,i}\} \in R^{H*W \times C'}$

$$A_1^i = \mathrm{softmax}\,(Q_1^i \,.\, (K_1^i)^T)\,.V_1^i \qquad \dots (4)$$
$$A_1^i = A_1^i \,.\, (W_1^0)^T \qquad \dots (5)$$

Where, $W_1^0 \in R^{C \times C'}$
Finally,
the attention output is given as $A_1 = [A_1^0, A_1^1, \dots, A_1^{T-1}]$, such that $A_1 \in R^{H*W \times T \times C}$

The Video frames are hence updated as $v^t_n = A_1$. This is then reshaped and used for attention across height, followed by width.

Across Height

Video clip v^j_n is reshaped to v^j_h such that $v^j_h \in R^{(W*T) \times H \times C}$, and with the current clip, v^t_h as query, and the other video clips being keys and values, it is attended with the other masked (K-t+1) blocks and unmasked t blocks of the video frames.
For each $j \in \{0, 1, \dots K-1\}$ and for each $i \in \{0, 1, \dots H-1\}$,

$$Q_2^i = v^t_h(i).q_2 \qquad \dots (6)$$
$$K_2^i = v^j_h(i).k_2 \qquad \dots (7)$$
$$V_2^i = v^j_h(i).v_2 \qquad \dots (8)$$

Where, $v^j_h(i)$ is the i^{th} 2D matrix of v^j_h across height dimension, and $\{q_2, k_2, v_2\} \in R^{C \times C'}$ are shared weights for all the T frames and $\{Q_2^i, K_2^i, V_2^i\} \in R^{W*T \times C'}$

$$A_2^i = \mathrm{softmax}\,(Q_2^i \,.\, (K_2^i)^T)\,.V_2^i \qquad \dots (9)$$
$$A_2^i = A_2^i \,.\, (W_2^0)^T \qquad \dots (10)$$

Where, $W_2^0 \in R^{C \times C'}$

Finally,
the attention output is given as $A_2 = [A_2^0, A_2^1, \dots, A_2^{H-1}]$, such that $A_2 \in R^{W*T \times H \times C}$

The Video frames are hence updated as $v^t_h = A_2$. This is then reshaped and used for attention across width.

Across Width

Video clip v^j_w is reshaped to v^j_w such that $v^j_w \in R^{(H*T) \times W \times C}$, and with the current clip, v^t_w as query, and the other video clips being keys and values, it is attended with the other masked (K-t+1) blocks and unmasked t blocks of the video frames.

For each $j \in \{0, 1, \ldots K\text{-}1\}$ and for each $i \in \{0, 1, \ldots W\text{-}1\}$,

$$Q_3^i = v^t_w(i).q_3 \qquad \ldots (11)$$
$$K_3^i = v^j_w(i).k_3 \qquad \ldots (12)$$
$$V_3^i = v^j_w(i).v_3 \qquad \ldots (13)$$

Where, $v^j_w(i)$ is the i^{th} 2D matrix of v^j_w across width dimension, and $\{q_3, k_3, v_3\} \in R^{C \times C'}$ are shared weights for all the T frames and $\{Q_3^i, K_3^i, V_3^i\} \in R^{H*T \times C'}$

$$A_3^i = \mathrm{softmax}\,(Q_3^i \,.\, (K_3^i)^T)\,.V_3^i \qquad \ldots (14)$$
$$A_3^i = A_3^i \,.\, (W_3^0)^T \qquad \ldots (15)$$

Where, $W_3^0 \in R^{C \times C'}$

Finally,
the attention output is given as $A_3 = [A_3^0, A_3^1, \ldots, A_3^{W-1}]$, such that $A_3 \in R^{H*T \times W \times C}$

The Video frames are hence updated as $v^t_w = A_3$. This is then reshaped and used for attention across width.

After all the above steps v^t_w is reshaped to $v^{temp} \in R^{H \times W \times T \times C}$, followed by addition with v^t and Layer Normalisation as follows,

$$v_t = (v_{temp} + v_t)$$
$$\mu_t = \frac{1}{T}\sum_{i=1}^{T} v_t(i) \qquad , \qquad \sigma_t^2 = \frac{1}{T}\sum_{i=1}^{T} (v_t(i) - \mu_t)^2$$
$$v_t(i) = (v_t(i) - \mu_t) \,/\, \sqrt{(\sigma_t^2)}$$

Frame level attention block with text embeddings

Masked self attention is followed by this block, where each frame of v^t is attended w.r.t the word embeddings, where every frame acts a query, and word embeddings act as keys and values. Inspired from AttnGan paper.
Let word embeddings be obtained from a pretrained model and be represented as Φ_t, such that $\Phi_t \in R^{D \times L}$. Φ_t is brought to the same semantic space as v^t using a perceptron layer such that,

$$e' = U.\Phi_t \text{ , where } U \in R^{H*W \times D}$$

L is the number of words and D is the dimensionality of the word feature vector, and $e' \in R^{H*W \times L}$

The video frames are divided into T matrices $v^t_i \in R^{H*W \times C}$ for $i \in \{0, 1, ….. T-1\}$
Let $v^t_i(j)$ denote the j^{th} feature of v^t_i , such that $v^t_i(j) \in R^{H*W \times 1}$ and $j \in \{0, 1, ….. C-1\}$
Also, let e'(k) denote the k^{th} column of e', such that $e'(k) \in R^{H*W \times 1}$ and $k \in \{0, 1, …..$ $D-1\}$

$$a_i(j, k) = \sum_k \text{softmax}((v^t_i(j))^T . e'(k)) . e'(k), \text{ such that } a_i(j, k) \in R^{H*W \times 1}$$

When all j features of frame with all the k word features, we get $a_i \in R^{H*W \times C}$ and after all the attentions across all the frames are calculated we get,

$$a = [a^0, a^1, \dots , a^{T-1}], \text{ such that } a \in R^{H*W \times C \times T}$$

Attention output, a, is then reshaped to H x W x C x T, and then added to the input v^t and then normalized, just like the previous layer

This completes the attention block.

Generator

Each attention block will be followed by a series/(single) of convolution layer(s) to downscale the image, to bring it down to $v^t_m \in R^{h \times w \times t \times c}$. Randomness (z) is introduced (experimental) such that, $z \sim \mathcal{N}(0, I)$, derived from Gaussian Distribution, such that $z \in R^{h \times w \times t \times c'}$ and concatenated with v^t_m along the channel dimension.

The generator, G is then defined as,

$$G : \{R^{h \times w \times t \times c}, R^{h \times w \times t \times c'}\} \rightarrow R^{H \times W \times T \times C}, \text{ i.e,}$$

$$G(z \,|\, v^t_m) = v^{pred}, \text{ such that } v^G = \{f_1^G, f_2^G, \dots f_T^G\}$$

$$\text{Where, } f_t^G \in R^{H \times W \times C}$$

Discriminator

Inspired by the paper, "To create what you tell", there are 3 different discriminator networks namely,

1. Video Discriminator (D_0)
2. Frame Discriminator (D_1)
3. Motion Discriminator (D_2)

Let s be the sentence embedding given by the pre-trained text encoder model

$$D_0(v, s) : \{R^{dv}, R^{ds}\} \rightarrow \{0, 1\}$$

$$D_1(f_i, s) : \{R^{df}, R^{ds}\} \rightarrow \{0, 1\}$$
$$D_2(f_i, f_{i-1}) : \{R^{df}, R^{ds}\} \rightarrow R^{c0 \times h0 \times w0}$$

Where, dv is the dimensionality of input video (either generated or ground truth), ds is the dimensionality of the sentence embedding, f_i is the i^{th} frame of the video clip and c0, h0, w0 are the dimensions of the downsampled frame m_f^i

Losses

Video-level Matching aware loss

Let v^+ be the real video with the correct sentence, v^- be the real video with mismatched sentence and v^{pred} be the generated video from G. This loss is calculated across the whole video produced, i.e $v \in R^{H \times W \times T \times C}$

$$L_v = -\frac{1}{3}[\log(D_0(v^+, s)) + \log(1 - D_0(v^-, s)) + \log(1 - D_0(v^{pred}, s))]$$

Frame-level Matching aware loss

Let $f^+(i)$ be the i^{th} frame from the real video with the correct sentence, $f^-(i)$ be the i^{th} frame from the real video with mismatched sentence and $f^{pred}(i)$ be the i^{th} frame from the generated video from G. This loss is calculated from a single frame produced, i.e $f(i) \in R^{H \times W \times C}$

$$L_f = -\frac{1}{3N}[\sum_{i=1}^{N} \log(D_1(f^+(i), s)) + \sum_{i=1}^{N} \log(1 - D_1(f^-(i), s)) + \sum_{i=1}^{N} \log(1 - D_1(f^{pred}(i), s))]$$

Temporal Coherence loss

Let $m^f(i)$ be the downscaled i^{th} frame from the generated video from G. This loss is calculated from 2 consecutive frames produced, i.e $m^f(i) \in R^{h0 \times w0 \times c0}$. Simply calculates the Euclidean Distance between 2 consecutive frames. This loss function corresponds only to the generator.

$$L_t = \frac{1}{N-1} \sum_{i=2}^{N} \|m^f(i) - m^f(i-1)\|_2^2$$

Discriminator Loss : $L_D = \frac{1}{2}(L_v + L_f)$

Generator Loss : $L_G = -\frac{1}{3}[\log(D_0(v^{pred}, s)) + \frac{1}{N}\sum_{i=1}^{N} \log(D_1(f^{pred}(i), s)) - L_t]$