Supplementary material for BézierSketch: A generative model for scalable vector sketches

Ayan Das[1,2], Yongxin Yang[1,2], Timothy Hospedales[1,3], Tao Xiang[1,2], and Yi-Zhe Song[1,2]

[1] SketchX, CVSSP, University of Surrey, United Kingdom
{a.das,yongxin.yang,t.xiang,y.song}@surrey.ac.uk
[2] iFlyTek-Surrey Joint Research Centre on Artificial Intelligence
[3] University of Edinburgh, United Kingdom
t.hospedales@ed.ac.uk

1 Appendix A

Property 1. Given a $(\mathbf{T}, \mathcal{P})$ pair where $\mathbf{T} = \mathbf{d}(\mathcal{P})$ for an arbitrary set of t, and $\widehat{\mathcal{P}} \sim \mathcal{N}(\mathcal{P}, \Sigma)$, then the decoded $\widehat{\mathbf{T}} = \mathbf{d}(\widehat{\mathcal{P}})$ with the same set of t, is distributed as $\mathcal{N}(\mathbf{T}, \Sigma')$, where Σ and Σ' are diagonal covariance matrices.

Proof. As Σ is diagonal, we can separate each dimension of $\mathcal{N}(\mathcal{P}, \Sigma)$ into individual Gaussians and then group $x - y$ components of each control point with its own Gaussian with diagonal covariance $\Sigma_i \triangleq \begin{bmatrix} \sigma_{x_i}, 0 \\ 0, \sigma_{y_i} \end{bmatrix}$

$$\mathcal{N}(\mathcal{P}, \Sigma) = \prod_{i=0}^{n} \mathcal{N}\left(\mathbf{P}_i, \Sigma_i\right)$$

By drawing samples from the gaussians of individual control points, we get $\widehat{\mathcal{P}} \triangleq \left[\widehat{\mathbf{P}}_i\right]_{i=0}^{n}$ where $\widehat{\mathbf{P}}_i \sim \mathcal{N}(\mathbf{P}_i, \Sigma_i)$. Decoding $\widehat{\mathcal{P}}$ by $\mathbf{d}(\cdot)$ gives

$$\widehat{\mathbf{T}} = \mathbf{d}(\widehat{\mathcal{P}}) = \sum_{i=0}^{n} \mathcal{B}_{i,n}(t) \cdot \widehat{\mathbf{P}}_i \tag{1}$$

Given any value of $t = t$, the random variable $\widehat{\mathbf{T}}$ is a weighted sum of n independent gaussian random variables with weights $[\mathcal{B}_{i,n}(t)]_{i=0}^{n}$. Hence, $\widehat{\mathbf{T}}$ is distributed as

$$\widehat{\mathbf{T}} \sim \mathcal{N}\left(\sum_{i=0}^{n} \mathcal{B}_{i,n}(t) \cdot \mathbf{P}_i, \sum_{i=0}^{n} \mathcal{B}_{i,n}^2(t) \cdot \Sigma_i\right) \tag{2}$$

Now we know that $\sum_{i=0}^{n} \mathcal{B}_{i,n}(t) \cdot \mathbf{P}_i \triangleq \mathbf{T}$ and we denote $\sum_{i=0}^{n} \mathcal{B}_{i,n}^2(t) \cdot \Sigma_i \triangleq \Sigma'$. So,

$$\widehat{\mathbf{T}} \sim \mathcal{N}(\mathbf{T}, \Sigma')$$

2 Appendix B

Sketch-RNN [2] is considered the state-of-the-art generative model for free-hand vector sketches. Sketch-RNN models the consecutive differences of 2D way-points of a sketch along with three bits denoting "touching", "stroke-end" and "sketch-end" state of the pen. In control point mode of BézierSketch, we adopted the same architecture and data representation as Sketch-RNN but with control points instead of waypoints. Hence, a sketch $\mathcal{S}_{cp}$ is transformed to a list (of length N) of 5-tuples $s_i \triangleq (\Delta P_x, \Delta P_y, q_1, q_2, q_3)_i$ where $[\Delta P_x, \Delta P_y]^T \triangleq \Delta \mathbf{P}$ is the successive difference of control points and $(q_1, q_2, q_3) \triangleq q$ are the three flag bits described above. As a normalization step, all sketches have been assumed to start from the origin (i.e., $[0, 0]^T$).

The core model of Sketch-RNN is a Sequence-to-Sequence Variational Autoencoder (Seq2Seq-VAE) [4] with a standard sequence encoder and an autoregressive decoder. The whole sketch sequence is fed into a Bidirectional encoder LSTM with hidden state given as

$$\mathbf{h}_i \triangleq \left[\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i\right] = \text{Bi-LSTM}(s_i, \mathbf{h}_{i-1}) \tag{3}$$

and the last state $\mathbf{h}_N$ is used as a compact representation of the sketch. $\mathbf{h}_N$ is then used to generate the parameters of a gaussian distribution following the VAE framework [3]. A sample is then drawn from the distribution as

$$\mathbf{z} \sim \mathcal{N}(\mu, \sigma), \text{ where } [\mu, \sigma] = f(\mathbf{h}_N) \in \mathbb{R}^Z$$

and decoded by an autoregressive decoder. An unidirectional LSTM is employed to initialize from $\mathbf{z}$ and produce a reconstruction of the sketch sequence similar to [1]. At each time-step j of the decoder, the hidden state is given as

$$\mathbf{g}_j = \text{LSTM}([\mathbf{z}; s_j], \mathbf{g}_{j-1}), \text{ with } \mathbf{g}_0 = \tanh(\mathbf{z})$$

The decoder, at every time-step, outputs the parameters of a GMM (with M mixtures) on $[\Delta P_x, \Delta P_y]^T$ and also a categorical distribution on three flag bits discussed above. Samples from these distributions are fed back as input s_{j+1} at next time step

$$\begin{gathered} s'_j = \left(\Delta \mathbf{P}'_j, q'_j\right), \text{ where} \\ \Delta \mathbf{P}'_j \sim \text{GMM}(\Delta \mathbf{P}; \mathbf{g}_j) \text{ and } q'_j \sim \text{Cat}(q; \mathbf{g}_j) \end{gathered} \tag{4}$$

The network is trained with the following loss that comprises of log-likelihood of the GMM, categorical cross-entropy of the flag bits and a variational KL divergance loss

$$\begin{aligned} L = -\frac{1}{N_{max}} &\left[\sum_{j=1}^{N} \log \text{GMM}(\Delta \mathbf{P}'_j) + \sum_{j=1}^{N_{max}} q_j \log q'_j\right] \\ &- \frac{1}{2Z}(1 + \sigma - \mu^2 - exp(\sigma)) \end{aligned} \tag{5}$$

3 Appendix C

We provide visualizations (Refer to Fig. 1) of the optimization dynamics over time. We also annotate a discrete point of the stroke and its corresponding point on the Bézier curve by joining them by a connector.

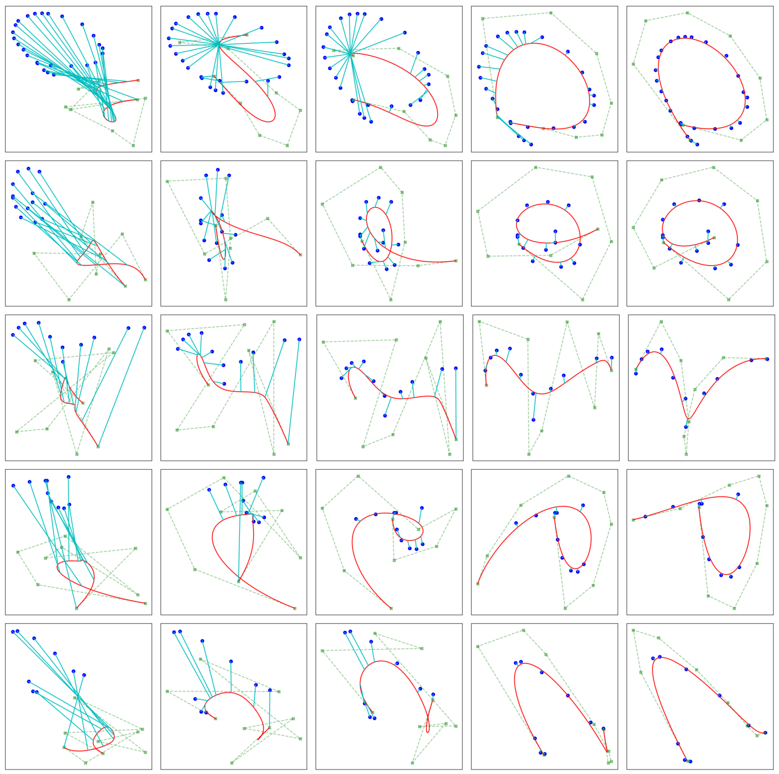

Fig. 1. Visualization of intermediate stages of the fitting for BézierEncoder network. Each row corresponds to one sample and columns denote increasing iterations of training.

References

1. Graves, A.: Generating sequences with recurrent neural networks. CoRR **abs/1308.0850** (2013)
2. Ha, D., Eck, D.: A neural representation of sketch drawings. In: ICLR (2018)
3. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. ICLR (2014)
4. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: ICML (2015)