The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

イラスト自動生成のための Sketch RNN の解析

Analysis of Sketch RNN for Sketch Auto Generation

藤井 涼佑 *1 Ryosuke Fujii　藤野 紗耶 *2 Saya Fujino　森 直樹 *2 Naoki Mori　松本 啓之亮 *2 Keinosuke Matsumoto

*1大阪府立大学 Osaka Prefecture University　*2大阪府立大学工学研究科 Graduate School of Engineering, Osaka Prefecture University

We present an sketch generation system composed of Sketch RNN, Google released. It is a recurrent neural network (RNN) constructed stroke-based drawings. Our system is different from traditional way, it is based on serial strokes which human draws. The diference makes output image much smoother and more natural. We introduce Sketch RNN into a tool of sketch generation. This paper focuses on two points. The first, we check that sketch production with Sketch RNN makes output image similar to input one. The second, we confirm that all the latent vector given by Sketch RNN make a sketch compiled soft strokes. Through these condideration, we explore new sketch generation system.

1. はじめに

近年，機械学習の発展を背景とした人工知能 (Artificial Intelligence : AI) が注目を浴びている．その中で画像認識の分野は画像検出や顔検出などの応用手法が多く，AI の適用が積極的になされている．AI は単純なパターン認識では人間の能力を凌駕する一方で，人間の情緒や感性に関する分野への AI の適用はいまだ難しく AI 研究の重要な課題とされている．例えば人の嗜好にあったものを提示するといった感性の理解は研究途上のものが多い．したがって，複数の手法を用いて，人の感性に寄り添った作品を計算機に生成させたいと考えている．筆者は人の嗜好を理解し，補完するためにスケッチ画像から完成画像の推定を目的としている．しかしながら，描いている途中の絵の推定は画像推定で優秀な性能をもつ深層畳み込みニューラルネットワーク (Deep Convolutional Neural Network : DCNN) では困難であった．そこで，今回は Google が発表した A Neural Representation of Sketch Drawings [1] に着目し，論文の提案モデルである Sketch RNN を使用した．Sketch RNN はスケッチ画像を時系列ベクトルで構成された画像としてとらえることにより，完成されたスケッチ画像の推定を可能としている．以上の点を背景として，本研究では，Google が公開した再帰型ニューラルネットワーク (Recurrent Neural Network : RNN) である Sketch RNN を解析する．時系列をもつ筆順から作成された潜在変数空間のベクトルを変えることで，生成イラストがどのように変化するかについて示す．また，Google の提供するデータセットではなく，ユーザの入力を中心に考察を進める．

2. Sketch RNN

本章では，スケッチ補完手法において優秀な性能をもつ Sketch RNN を説明する．

2.1 Sketch RNN の概要

Sketch RNN とは，人間がイラストを描くときの一連の描き順を訓練することでユーザのイラストを補完，推測する深層学習を利用したモデルである．入力はスケッチの時系列を含む筆順データであり，潜在ベクトルが中間出力となる．また，最終出力は画像を生成可能とする筆順ベクトルデータとなる．

2.2 Model

Sketch RNN の encoder は bidirectional RNN で構成されている．スケッチの時系列データ S とその逆順データ S_{R} の出力を全結合層 h に入力する．h の出力は全結合層によって，サイズ $N_{\boldsymbol{z}}$ の分布の平均 μ と標準偏差 $\hat{\sigma}$ を得る．μ と $\hat{\sigma}$ からガウス分布に従った $\boldsymbol{z} \in \mathbb{R}^{N_{\boldsymbol{z}}}$ を得る．(1) 式が $\boldsymbol{z}$ となり，$\boldsymbol{z}$ は i.i.d. となる．また，(2) 式は，(1) 式の σ の定義式である．

$$\boldsymbol{z} = \mu + \sigma \odot \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{1}$$

$$\sigma = \exp(\frac{\hat{\sigma}}{2}) \tag{2}$$

この潜在ベクトル $\boldsymbol{z}$ が入力されたスケッチの条件付きランダムベクトルとなる．

Sketch RNN の decoder は自己回帰的 RNN で構成され，出力を $\boldsymbol{z}$ に対して条件付きでサンプリングする．ここで，状態 $S_i(\Delta x, \Delta y, p_1, p_2, p_3)$ は二次元座標情報 Δx，Δy と，ペンが紙に接触している状態 p_1，ペンが紙から離れている状態 p_2，書き終わり p_3 を表す．(3) 式は S_i が (p_1, p_2, p_3) をモデルとして常に $q_1 + q_2 + q_3 = 1$ となる条件下で，Δx，Δy を導出したものである．ここで，M は正規分布による混合ガウス分布の個数を示す．さらに，$\mathcal{N}$ は二変数ガウス分布である．また，係数部分 $\boldsymbol{\Pi}_j$ はいくつかのガウス分布を重ね合わせた混合ガウス分布の係数である．

$$p(\Delta x, \Delta y) = \sum_{j=1}^{M} \boldsymbol{\Pi}_j \mathcal{N}(\Delta x, \Delta y | \mu_{x,j}, \mu_{y,j}, \sigma_{x,j}, \sigma_{y,j}, \rho_{xy,j}) \quad \text{where} \quad \sum_{j=1}^{M} \boldsymbol{\Pi}_j = 1 \tag{3}$$

(4) 式はペンと紙との状態を表す $\hat{q}$ を 0 から 1 までの値をとる確率に変換している．

$$q_k = \frac{\exp(\hat{q}_k)}{\sum_{j=1}^{3} \exp(\hat{q}_j)}, \quad k \in \{1, 2, 3\} \tag{4}$$

連絡先: 藤井 涼佑，大阪府立大学，fujii@ss.cs.osakafu-u.ac.jp

(5) 式はガウス分布の係数部分の $\hat{\mathbf{\Pi}}$ を 0 から 1 までの値をとる確率を示す．

$$\mathbf{\Pi}_k = \frac{\exp(\hat{\mathbf{\Pi}}_k)}{\sum_{j=1}^{M}\exp(\hat{\mathbf{\Pi}}_j)}, \quad k \in \{1, ..., M\} \tag{5}$$

2.3 訓練手法

KullbackLeibler divergence (KL) ロスの和を損失関数として利用する．(6) 式は再構成ロスの (Δx,Δy) の損失を表す．

$$L_{\mathrm{s}} = -\frac{1}{N_{\max}}\sum_{i=1}^{N_{\mathrm{s}}} \log\Big(\sum_{j=1}^{M}\mathbf{\Pi}_{j,i}\,\mathcal{N}(\Delta x_i, \Delta y_i|\mu_{x,j,i}\,, \mu_{y,j,i}\,, \sigma_{x,j,i}\,, \sigma_{y,j,i}\,, \rho_{xy,j,i})\Big) \tag{6}$$

また，(7) 式は再構成ロスの (p_1, p_2, p_3) の損失を表す．

$$L_{\mathrm{p}} = -\frac{1}{N_{\max}}\sum_{i=1}^{N_{\max}}\sum_{k=1}^{3} p_{k,i}\log(q_{k,i}) \tag{7}$$

(8) 式は再構成ロスで，L_s，L_p の和である．

$$L_{\mathrm{R}} = L_{\mathrm{s}} + L_{\mathrm{p}} \tag{8}$$

(9) 式は KL ロスを表す．$\boldsymbol{z}$ がどの程度標準正規分布から離れているかを示す損失関数となる．

$$L_{\mathrm{KL}} = -\frac{1}{2N_{\boldsymbol{z}}}\Big(1 + \hat{\sigma} - \mu^2 - \exp(\hat{\sigma})\Big) \tag{9}$$

したがって，(10) 式が Sketch RNN の損失関数となる．

$$\mathrm{Loss} = L_{\mathrm{R}} + w_{\mathrm{KL}}L_{\mathrm{KL}} \tag{10}$$

ここで，w_{KL} とは 2 項のロスへの重みを決めるスカラーである．L_{R} と L_{KL} との間にはトレードオフの関係があり，KL 項を焼きなましてロスを小さくする．

2.4 温度パラメータ τ について

Sketch RNN は decode する際，乱数部分を調整する変数である温度 τ が存在する．温度 τ は $0 \leq \tau \leq 1$ をとる．(11) 式，(12) 式は温度 τ で補正したものである．

$$q_k = \frac{\exp(\frac{\hat{q}_k}{\tau})}{\sum_{j=1}^{3}\exp(\frac{\hat{q}_j}{\tau})}, \quad k \in \{1, 2, 3\} \tag{11}$$

$$\mathbf{\Pi}_k = \frac{\exp(\frac{\hat{\mathbf{\Pi}}_k}{\tau})}{\sum_{j=1}^{M}\exp(\frac{\hat{\mathbf{\Pi}}_j}{\tau})}, \quad k \in \{1, ..., M\} \tag{12}$$

(13) 式，(14) 式は x，y の分散を温度 τ だけ縮小させている．

$$\sigma_x^2 \to \sigma_x^2\tau \tag{13}$$

$$\sigma_y^2 \to \sigma_y^2\tau \tag{14}$$

温度 τ は (13) 式や (14) 式から分散に関して調整をすることで乱数を制御している．この式から，温度 τ が 0 の場合分散が 0 となり，出力は決定的になることがわかる．

表 1: Sketch RNN のパラメータ

パラメータ名	パラメータ
decorder model	hyper
decorder rnn size	2048
encoder model	hyper
encoder rnn size	512
save every	500
num steps	45000
class	30

3. 従来手法の問題点および改良手法

3.1 従来の手法の問題点

従来の絵本等に画像を生成するシステムはビットマップ形式の画像を使用するものが多かった．画像が一般的な画素集合で構成されている場合は，この手法で正確な画像を生成することができる．しかし，ペン等による線のみで構成されたスケッチ筆順という時系列性を保持しているため難しい．ところが，既存の画像生成手法ではイラスト画像をビットマップ形式に変換していたため，入力画像の連続性に関する情報落ちという問題があった．

3.2 今回の改良点

連続的な筆順を保持するため，画像生成システムの内部構造に Sketch RNN を採用した．Sketch RNN は筆順をすべて 5 次元のベクトル $(\Delta x, \Delta y, p_1, p_2, p_3)$ に変換する．$\Delta x, \Delta y$ はひとつ前の点からペンの移動距離を表す．また，p_1, p_2, p_3 はそれぞれ，ペンが紙に接触している状態，ペンが紙から離れている状態，描き終わり，を表している．最後の 3 つは one-hot ベクトル表現になっている．これらを VAE (Variational AutoEncoder) に渡す．連続性を生かすため，encoder と decoder は RNN となっている．Sketch RNN の入力は時系列をもった 5 次元ベクトル $(\Delta x, \Delta y, p_1, p_2, p_3)$ である．encoder によって中間出力の潜在ベクトル $\boldsymbol{z}$ を得る．さらに decoder によって出力として再び 5 次元ベクトル $(\Delta x, \Delta y, p_1, p_2, p_3)$ を得る．この出力ベクトルを組み合わせることで画像を生成する．従来の DCNN ではピクセルごとに画素値を出力する手法と比較して，Sketch RNN はスケッチの連続性をもち，筆順の前状態に依存して次状態が決定する．したがって，既存の画像生成よりも連続的な筆順をもった自然なイラストを出力として手に入れることができる．また，RNN で構成されているので，副次的に描きかけ途中のイラスト入力から完全な状態のイラストを得ることができる．これにより，イラストの補完も同時に処理することができる．この補完性能は DCNN では実現が困難であったことから，画像生成システムに新しい可能性が生じる．

4. 実験

今回は画像生成システムを形成するにあたって，簡単な GUI を作成した．学習に関するネットワーク構造は Sketch RNN を使用する．表 1 に Sketch RNN のパラメータを示す．

encoder および decoder model における hyper とは，HyperLSTM [2] を示す．他のパラメータは Sketch RNN の奨励値を使用した．また，学習に使用はすべて Google が公開するデータセットを使用した．データセットは 30 クラスに対して train / valid / test をそれぞれ 20000 枚 / 500 枚 / 500 枚

ずつ使用した．今回は，以下の 2 点に焦点を当てた 2 つの実験をした．

- 実験 1. 生成した画像と入力画像の比較
- 実験 2. 複数の入力画像の潜在ベクトル間の出力画像の変化確認

実験 1 では Sketch RNN による生成画像が画像生成システムに適用できるか確認する．こちらはユーザが描いた画像と乖離することがないよう定量的な評価が必要となる．また，実験 2 では Sketch RNN の潜在空間の内部構成を確認する．実験 2 では，任意の潜在空間中のベクトルを decode した場合でも，生成されたベクトルを組合せた画像が筆順として連続性を保持しているか確認することが重要なため，画像の連続性に着目した．

4.1 実験手法

4.1.1 実験 1

Sketch RNN の学習に使用したデータセットを使用して DCNN による識別器を予め作成する．今回は画像のクラス分類に優秀な性能を持つ AlexNet [3] を使用した．また，Sketch RNN には任意の画像 60 枚を入力させる．Sketch RNN は温度パラメータ τ の変動を考慮して τ の値を 10 段階に分けて出力する．1 枚の入力画像から 10 枚の出力画像を得る．この 10 枚を強識別器に渡して最も accuracy の高い値を記録する．これを 60 枚に対して実行し，平均と分散を求める．また，この実験では出力画像が明瞭であることが望ましいため，入力画像はユーザの手で描き終えたスケッチのみとする．

4.1.2 実験 2

2 組の画像を Sketch RNN に入力し，それぞれの潜在ベクトルを取り出す．この潜在ベクトルをそれぞれ $\boldsymbol{z}_{\mathbf{a}}, \boldsymbol{z}_{\mathbf{b}}$ とし，いずれも 128 次元とする．(15) 式に出力対象となる $\boldsymbol{z}_{\mathbf{INT}}$ を導く過程を示す．

$$\begin{aligned} \boldsymbol{z}_{i,\mathbf{INT}} &= (1-t)\boldsymbol{z}_{i,\mathbf{a}} + t\boldsymbol{z}_{i,\mathbf{b}} \\ i &= 0, 1, 2, 3, ..., 127 \\ 0 &\leq t \leq 1 \end{aligned} \tag{15}$$

(15) 式により導かれた潜在ベクトルを decode して生成されたベクトルを組合せて画像を得る．2 つの潜在ベクトルを今回は 10000 等分に内分した様子を見る．10 組のテスト画像から特に変化の大きいものを確認する．

4.2 実験結果

4.2.1 実験 1

AlexNet による 30 クラス識別器のテストデータに対する accuracy は 0.85 であった．30 クラス識別の場合ベースラインは一様選択した場合の確率 0.03 を大きく上回るため，実験 1 の識別器として適当であった．表 2 に Sketch RNN の入力画像と出力画像を AlexNet による識別器にかけた時の softmax に関する度数階級表を示す．ただし，60 枚の画像を入力して，正解ラベルに分類した件数のみをまとめている．そのため，入力画像と出力画像の度数の数は一致していない．

さらに，表 3 に Sketch RNN の入力画像と出力画像を識別器にかけた時の softmax の平均値と分散を示す．こちらは不正解ラベルの分類したものも含めて，正解ラベルの softmax の値のみで算出している．

表 2: 入力画像と出力画像の softmax に関する度数階級表

softmax 階級	入力画像 (枚)	出力画像 (枚)
0.50 未満	0	2
0.50 以上 0.60 未満	0	0
0.60 以上 0.70 未満	0	1
0.70 以上 0.80 未満	0	0
0.80 以上 0.90 未満	1	6
0.90 以上 1.00 以下	59	25

表 3: 入力画像と出力画像の softmax の平均値と分散

	softmax 平均値	softmax 分散
入力画像	0.99	0.00
出力画像	0.56	0.21

表 2 より，入力画像とラベルは強い相関を持っていることが確認される．これにより，入力画像は実験 1 において適切であることがわかる．出力画像からは正解ラベルに分類された場合は高い softmax の値をとるという特徴がみられる．また，表 3 より，出力画像の softmax の平均値がベースラインの 0.03 よりも高い水準である 0.56 ととなっている．分散が大きいことと，softmax の高い階級の度数がが多いことから，生成画像において極端に安定しているクラスや，反対に不安定なクラスが存在すると考えられる．しかし，正解クラスに分類されている出力画像が 60 件中 34 件と過半数を超えていることから，ユーザのスケッチに合わせた出力ができているといえる．

また，図 3 にユーザが入力したスケッチを示す．同様に，図 4 にユーザが入力したスケッチを decode した結果を示す．これらは decode された画像の softmax が高いものから選んだものである．したがって，AlexNet による識別器の softmax が高ければ正解ラベルの画像が明瞭に表れることがわかる．以上の結果から softmax の値によってユーザの描いたスケッチと出力結果の乖離度合を定量的に評価できることがわかった．

4.2.2 実験 2

図 1, 図 2 に 2 つの入力画像の潜在ベクトル間を紙面の都合上 10 等分に内分されたベクトルを使用して得られた画像を示す．

図 3 の入力画像は fork と cloud である．左側が fork で，右側のものになるにつれて cloud に移っていることがわかる．同様にして，図 4 の入力画像は donut と moon である．左側が donut で，右側のものになるにつれて moon に移っていることがわかる．fork と cloud の内分ベクトルによって生成された画像は筆順が滑らかに変化していることが確認できる．特に図 3 の左から 4 番目から 6 番目にかけて fork の持ち手のような部分が徐々に上部へ移動して雲の上部輪郭へと変化している．このような現象から，ピクセルアートのような不自然さは軽減されていると考えられる．また，図 4 より，入力で使用した画像クラス以外の画像を生成されることが確認された．潜在空間が複雑な分布をもっていると推測される．

これらの実験結果より，Sketch RNN による画像生成では連続性が保存され，任意の潜在空間上の潜在ベクトルは滑らかなタッチのスケッチとなることが確認できた．

図 1: fork と cloud 潜在ベクトル内分点出力画像

図 2: donut と moon 潜在ベクトル内分点出力画像

図 3: ユーザ入力画像の一例

図 4: decode された画像の一例

5. まとめ

本稿では，従来型の画素の値から畳み込む DCNN の手法とは異なる筆順を基にした時系列ベクトルである筆順を RNN に与える Sketch RNN によるスケッチ生成の可能性について検討した．Sketch RNN の長所は画素値のような離散値でなく，連続したベクトルデータを扱えることである．とりわけ，写真ではなく，イラストのような画像では連続した線が構成の大半を占めるため，畳み込み型の学習では直前までのペンの状態が情報落ちしていると考えられた．そこで，2 つの実験を通して Sketch RNN のスケッチ画像生成における有効性を示した．1 つ目は入力画像と出力画像がどの程度異なるかについて調べた．2 つ目は中間出力である潜在ベクトルを任意に変更した場合でも出力が断続的でないか調べた．

実験 1 では，表 3 より，自動生成された画像は正解ラベルに対して softmax の値が平均で 0.56 であることが確認された．入力画像の正解ラベル対する softmax の値よりも低いが，ベースライン 0.03 を大きく上回り，生成精度は高いといえる．また，図 4 より，人間の判断も加えながらこの結果の妥当性を補完した．したがって，Sketch RNN に基づく画像生成には入力画像にある程度近い出力画像が得られるという結果になった．

実験 2 では，図 1, 図 2 を通して，潜在空間の可視化をした．Sketch RNN は VAE で構成されているので，任意の潜在空間上のベクトルは decode することで画像を出力する．一方で，出力された画像における線分の連続性は保証しないが，いずれの結果も筆順が保持されていた．また，2 つの入力画像の潜在ベクトル間に沿ったベクトルを decode すると出力画像を構成する線分が動いている様子が観察された．したがって，スケッチのような線で構成されている画像に対して Sketch RNN は非常に有効であると導かれた．

これらの結果からスケッチの自動生成に関して RNN の構造を持つ Sketch RNN は従来の DCNN と比べて有利なものが多い．今後，線描画の多い画像に対して優秀な性能をもつ可能性があると考えられる．

今後の課題としては Generative Adversarial Network (GAN) [4] と Sketch RNN の比較およびスケッチ画像の生成精度の向上が上げられる．

また Sketch RNN を GUI を作ることも重要な目標としている．現在スケッチする際に必要最小限の機能のみとなっている．研究目的から応用へと転換させるため，アプリケーションなどの作成も視野に入れている．なお，本研究は一部，日本学術振興会科学研究補助金基盤研究 (C) (課題番号 26330282) の補助を得て行われたものである．

参考文献

[1] David Ha and Douglas Eck. A neural representation of sketch drawings. *CoRR*, abs/1704.03477, 2017.

[2] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. *CoRR*, abs/1609.09106, 2016.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.