Auto Stroke-Based Sketch

Xinrui Ying Jingzhi Zhang Ye Zeng Siqi Du Yuheng Zhang Mingyang Xia

1 Goal

In this project, our team want to develop a sketch model and train the machine to generate a complete figure based on current sketch drawing. This model will mainly utilize the structure of recurrent neural network(RNN), which will take one or a sequences of strokes as input, and predict what the object might be and complete the rest of the sketch.

2 Motivation

Drawing can help people to expand their imagination and stimulate our creativity, more importantly, it is a crucial way for us to observe and document the world since childhood. There are already many work related to paints, such as training the machine to mimic famous artists style or transforming a picture to a specific painting style. Based on the previous achievement, we want to develop a more interactive system that can incorporate both human and machine to accomplish the same work. This model might enable many creative application, one potential application is that it can help children to draw a simple sketch, or it can even inspire the artists based on what the machine drawn in an unprecedented and novel way.

3 Problem Formulation

This section includes three subsections

- Dataset and Environment
- Difference With Existing Work
- Primary Contributions and Novelty

3.1 Dataset and Environment

We obtained the dataset from $Quick, Draw!$, which contains vector drawings where people are asked to draw a particular object in less than 20 seconds. Overall, it has about 75K samples (70K Training, 2.5K Validation, 2.5K Test), images are selected randomly from each category. We will also be providing all 50 million samples from 345 categories if we think it's necessary to use them all. The tutorials and links for getting the dataset is following: `https://github.com/googlecreativelab/quickdraw-dataset#get-the-data`

We are going to use Python 3 programming language for most of our developing, and using GPU and TensorFlow.

3.2 Existing works

There are lots of existing works that imitate painting of human.

1. Reinforcement Learning Approach to Automatic Stroke Generation Artist Agent, Automatic Stroke Generation in Oriental Ink Painting, which inputs a digital picture and produce a series of paint brush strokes.

2. Neural Network Approach to Image Generation There are many approach to transform between digital images and sketches: `https://twitter.com/bgondouin/status/818571935529377792` and Image-to-Image Translation with Conditional Adversarial Networks. However, there are few work on generating vector images due to the lack of public datasets. Related previous work including Recurrent Net Dreams Up Fake Chinese Characters in Vector Format with TensorFlow, which modeling Chinese characters as series of pen stroke actions. And ShadowDraw,predicting the finished drawing from unfinished brush strokes, which used a dataset of raster images and extracted vectorized features. Here we use a larger dataset of vector sketches.
3. Sketch Recognition Approaches in recongizing free hand sketches using local features : Sketch-Based Image Retrieval: Benchmark and Bag-of-Features Descriptors, which compares different local features Shape Context, Spark feature, Histogram of Oriented Gradients (HOG) and sketched HOG, HOG based features outperform others. Because HOG is a highly optimized feature descriptor for encoding edge and gradient properties of images, and human sketches' main property is the edge and gradient.

3.3 Primary Contributions and Novelty

1. Few previous works on generating vector images. Here we use a larger dataset of vector sketches to generate vector images.
2. We plan to build a RNN model and tune the free hand sketch recognizing problem into a more interacting problem: player draw the first several strokes and the model will finish the remaining steps based on it's prediction of player's intention, which feels like you complete the sketch painting together with our agent.

4 High-level steps

- Step1: We define a model of sketch-RNN which contains two parts - encoder and decoder. For encoder, we are using a bidirectional RNN that takes in a sketch as an input and outputs a feature vector to represent it. For decoder, we will try to use autoregressive RNN to output vector image stroke by stroke.
- Step2: We train the above model to the dataset of sketch images we found. Each of them contains a sequence of strokes: where to start, which direction to move our pen and when to stop.
- Step3: By doing so, we created a model which may have many potential applications like showing children how to draw sketch images, or drawing some creative images automatically. In our project, we will let one person to draw the first stroke of a given object(cat, butterfly, house) and let our model to complete it. Then we let others guess what kind of object it draws to show how accuracy our model is.

5 Evaluation Experiments

To evaluate our results, we have two main concerns. First of all, the images we created should be similar to the original datasets. Hence, we calculate the transform vector with the original dataset's image and then compare their difference. Second, it should be a kind of stroke image. The images are consisting of not too many strokes. Hence measure each strokes length and the number of it is very important. Thence L1=min(t(originalImage(i)-t(newdata)), L2=g(number of lines, lines' length). Eval=L1 + w*L2 (w is weight).

6 Expected Results

First, we expect to train an RNN model with the encoder and decoder parts. Second, we expect using the decoder part of the RNN model to build a sketch generator which given some strokes of a specific kind of object can complete a decent sketch of that object based on the strokes. Finally, we will have a demo application in which the user can draw one stroke on the its canvas then a sketch will be finished based on the given stroke automatically.

7 Milestones (Timeline)

Oct.6-13 Designing the specific algorithms and experiment steps
Oct.15 Project spotlight
Oct.13-27 Coding
Oct.27-Nov.3 Training the model
Nov.10 Project mid report
Nov.3-24 Developing the sketch game demo
Nov.24 Project the representation
Dec.1 Project final report