DoodelNet: accurate doodle sketch recognition

Stanford University

Jiajing Wang, Yixin Shi

Project Description

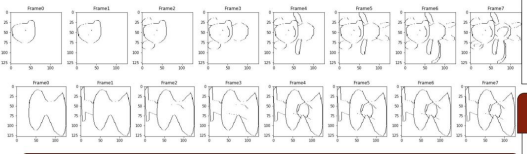
By leveraging deep learning technology such as ConvNet and RNN, we aim to build a better doodle sketch recognition classifier for 50 million sketch drawing from Quick Draw!

Dataset

Original 340 classes and 50 million sample
30 classes with 300K training, 3K evaluation data.

Preprocessing

- Normalize strokes and convert coords to delta
- Normalize strokes, Convert strokes to 2D 128X128X1 images with 8 frames



Error Analysis

Index	Image	Prediction	Ground truth
0		bear	alarm clock
1		asparagus	baseball
2		aircraft carrier	anvil
3		bandage	bat
4		bathtub	bed
5		ant	bat
6		asparagus	angel
7		bear	alarm clock

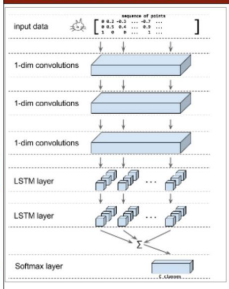
Human would predict correctly 18 %

Human would not predict correctly 60 %

Ambiguous for human 22 %

Given 83% accuracy, the best accuracy would be 86.4% - 89.8%

Base Model - Conv-1D + LSTM

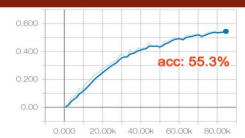


Conv-1d + LSTM + Softmax

Table 1: Tuning the Recurrent QuickDraw model

Dataset size	Learning rate	Batch size	Other parameters	Accuracy
3k drawings per class, 30 classes in total	0.001	8	Steps=100k	83%
	0.01	8	Steps=100k	3% (-80%)
	0.0001	8	Steps=100k	85% (+2%)
	0.0005	8	Steps=270k	85% (+2%)
	0.00003	8	Steps=400k, CUDA-based LSTM	84% (+1%)
	0.00002	8	Steps=400k, 4 LSTM layers	83% (+0%)
10k drawings per class, 30 classes in total	0.0001	8	Steps=800k	89% (+6%)
	0.0003	32	Steps=400k	89% (+6%)
	0.00006	2	Steps=1.2M	85% (+2%)
100k drawings per class, 30 classes in total	0.0001	8	Steps=1.6M	93% (+10%)
100k drawings per class, 340 classes	0.0001	8	Steps=6M	77% (-6%)

Performance evaluation through HP tuning



Slow and subpar with 340 class

Baseline by limiting to 30 classes

Exploration I - Conv-1D + LSTM with Attention

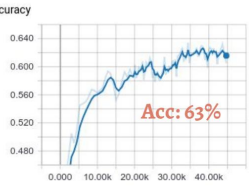
- Modified LSTM layers with fixed length attention mechanism => accuracy of ~79% after very slow training
- Attention doesn't help: useful information is not locally clustered
- At the cost of significantly slowing down training speed (2 -> 0.9 step/s)

Exploration II - Conv-2D (DenseNet-121) + RNN

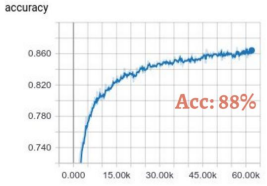
Modified DenseNet-121

Layers	Output size	Architecture / details
Input	8 x 128 x 128 x 1	8 accumulative frames of 128x128 drawings in black & white
Convolution	8 x 64 x 64 x 64	7 x 7 conv, stride 2
Pooling & ReLu	8 x 32 x 32 x 64	3 x 3 max pool, stride 2
Dense Block 1	8 x 32 x 32 x 256	{ 1 x 1 conv, 3 x 3 conv } × 6
Transition Layer 1	8 x 16 x 16 x 128	1 x 1 conv & 2 x 2 avg. pool, stride 2
Dense Block 2	8 x 16 x 16 x 320	{ 1 x 1 conv, 3 x 3 conv } × 12
Transition Layer 2	8 x 8 x 8 x 160	1 x 1 conv & 2 x 2 avg. pool, stride 2
Dense Block 3	8 x 8 x 8 x 352	{ 1 x 1 conv, 3 x 3 conv } × 24
Transition Layer 3	8 x 4 x 4 x 176	1 x 1 conv & 2 x 2 avg. pool, stride 2
Dense Block 4	8 x 4 x 4 x 368	{ 1 x 1 conv, 3 x 3 conv } × 16
Pooling	8 x 1 x 1 x 368	4 x 4 avg. pool
Fully connected 1	8 x 128	8 frames of vectors of 128 nodes
LSTM 1 and 2	1 x 256	LSTM layers of 8 stages
Fully connected 2	1 x 30	Softmax output of 30 classes

Simple Conv2d layers + RNN



DenseNet Conv2d layers + RNN



- A shallow Conv2D + LSTM is not sufficient for large dataset training
- 5-block DenseNet + LSTM deep w/o gradient exploding or vanishing achieves much better results, and has improving potentials if we reduce preprocessing information losses