Doodle Recognition Challenge

Arsh Buch, Sophia Chen, John Yu
{arshbuch, schen10, johnyu2}@stanford.edu

Stanford | ENGINEERING
Electrical Engineering

Project Overview

- Our project involves categorizing human generated doodles
- Used various techniques such as hyperparameter tuning, learning rate decay and data preprocessing on various CNNs
- Each net was trained from scratch

Dataset

- Kaggle "Quick, Draw! Doodle Recognition" dataset[1]: 100 categories, 200 train/50 dev/50 test for each category
- Converted bitmap representation into a 28x28 grayscale image and upsampled using bilinear interpolation.
- Shown: coffee cup, apple, donut

Future Work

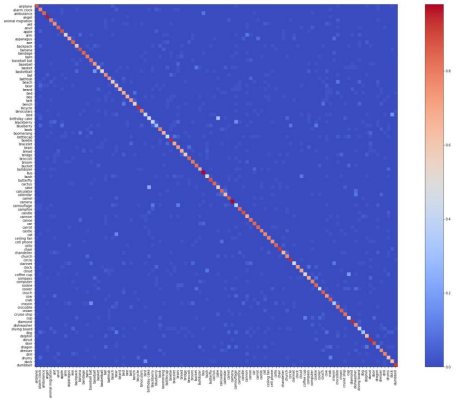
- Incorporate all data points given with images (country code, brush stroke timing)
- Better representations of images through image generation
- Increase compute capabilities to train on all images

Summary of Results

Model	Training	Validation	Test
VGG-16	92.8%	71.4%	71.8%
MobileNet	99.5%	67.4%	67.2%
ResNet-34	90.1%	64.5%	62.3%
ResNet-50v2	85.2%	57.3%	56.2%

Notes for this section:
[1] VGG-16, ResNet-34 and MobileNet were trained for 30 epochs. ResNet-50 was trained for 15 epochs since the validation accuracy never improved past 60%
[2] Displayed confusion matrix is for VGG-16

1. **VGG-16 has the best performance**
2. **However, networks that are too deep overfit the dataset**
3. **Dataset could be better – upsampling becomes blurry**
4. **Bottleneck: amount of classes and data is 100s of GBs**
5. **Preprocessing techniques such as image generation may decrease gap between training and test accuracy**



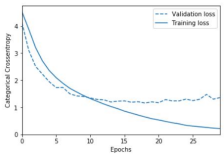
VGG-16

Overview

- ► 16 layer network consisting of 5 convolutional blocks, 2 fully connected layers, and a softmax output layer[2]
- ► 224x224 gray scale images
- ▲ Allows for deeper network with comparable receptive field sizes
- ▲ More expensive to evaluate and uses a lot more memory

Results

- ▲ Less overfit than other networks, validation accuracy within 20% of training accuracy
- ▲ Tuned hyperparameters: SGD optimizer, dropout
- ▲ Highest test accuracy



MobileNet

Overview

- ► Designed for visual inference on mobile and real time applications[3]
- ▲ Light weight, deep neural network with depth wise separable convolutions with batchNorm and ReLU

Results

- ► Utilized learning rate decay and Adam optimizer to train all of the layers on 128x128 grayscale images
- ▼ Simplicity of the model did not let us learn lower level features resulting in high variance even after regularization techniques (changing batch size, adding more training data, drop out, L2 regularization)

References:
[1] AI, Google. "Quick, Draw! Doodle Recognition Challenge." *Kaggle*, 2019, www.kaggle.com/c/quickdraw-doodle-recognition.
[2] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014). arXiv: 1409.1556. url: http://arxiv.org/abs/1409.1556.

Residual Network

ResNet-34

- ▲ 3-convolutional layer v2 residual blocks, 34 layers total, deeper network but not too complex[4]
- ► Dropout layer inserted between convolutional layers in each block, dropout rate 0.3
- ► 224 x 224 grayscale images
- ▲ Tuned optimizer, batch size, learning rate, dropout
- ▲ Some overfitting (~20%)

ResNet-50v2

- ► 3-convolutional layer v2 residual blocks, 50 layers total[4]
- ► 224 x 224 grayscale images
- ► Tuned hyperparameters: learning rate, rate decay, dropout
- ▼ Overfitting due to large network

[3] Andrew G. Howard et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: CoRR abs/1704.04861 (2017). arXiv: 1704.04861. url: http://arxiv.org/abs/1704.04861.
[4] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: CoRR abs/1512.03385 (2015). arXiv: 1512.03385. url: http://arxiv.org/abs/1512.03385.