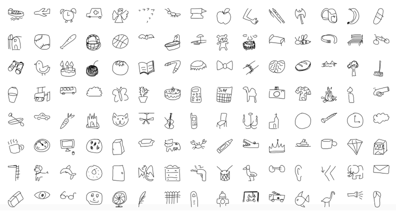
Guess Your Doodling

Liu Jiang, Xu Song, Jingwen Sun
DSC305 Final Project



1

Introduction

In recent years, convolutional neural network(CNN) has literally dominated the realm of image classification. Due to the explosion of computational power, human beings finally unleashed the blade of artificial neural network(ANN). As the descendent of ANN, CNN takes into account the fact that pixels close to each other are more related than pixels further apart and outperforms the previously used methods like logistic regression and support vector machine.

In this project, we explored the power of CNN using *Quick, Draw!* dataset provided by Google. *Quick, Draw!* is a web application. It prompts user with a word, the user draws an image according to that word, and then the computer guesses the word while the user is drawing the image. The *Quick, Draw!* dataset provides not only the final picture of users' hand drawings with the corresponding class labels, but also the sequence of strokes that the image is drawn by. Analyzing sequence of strokes would require recurrent neural network(RNN) which is complicated and hard to implement. Therefore, our project mainly focused on using CNN with pixels as inputs. The project consists of 2 parts:

1. Build a CNN model that is able to classify hand-drawn images from 10 classes using Keras, a high-level API for TensorFlow.

2. Deploy our model into a web application that mimic Quickdraw.

Model

Here is the architecture of our CNN model (see Figure 1 for visualization of layers. Notice that channels are different):

```
Conv2D(64, (3, 3), activation='relu', input_shape=(28, 28, 1))
MaxPooling2D((2, 2))
Conv2D(256, (3, 3), activation='relu’)
MaxPooling2D((2, 2))
Dropout(0.2)
Flatten()
Dense(7 * 7 * 128, activation='relu')
Dropout(0.5)
Dense(10, activation='softmax')
```

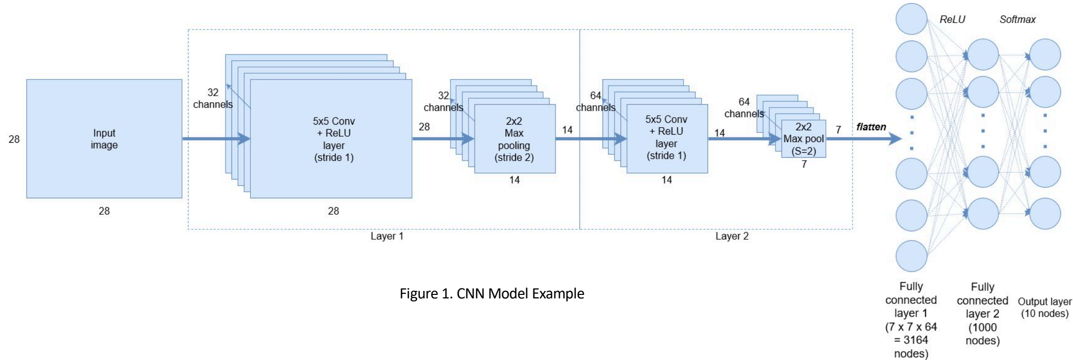

28
Input image
28
32 channels
5x5 Conv + ReLU layer (stride 1)
28
28
32 channels
2x2 Max pooling (stride 2)
14
14
64 channels
5x5 Conv + ReLU layer (stride 1)
14
14
64 channels
2x2 Max pool (S=2)
7
7
flatten
Layer 1
Layer 2
ReLU
Softmax
Fully connected layer 1 (7 x 7 x 64 = 3164 nodes)
Fully connected layer 2 (1000 nodes)
Output layer (10 nodes)

Figure 1. CNN Model Example

2

Model Training:

Training set:	50000 images(5000 for each class)
Validation set:	10000 images(1000 for each class)
Test set:	20000 images(2000 for each class)

Accuracy: 94%

Deployment

After constructing the model, we deploy it into a web application that can recognize hand drawings from users

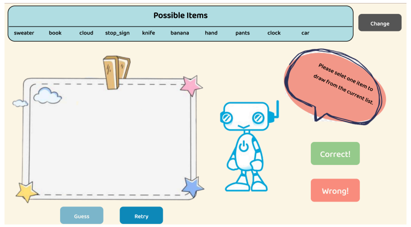


Figure 2. Web Application

Figure 3 shows the underlying logic of our application

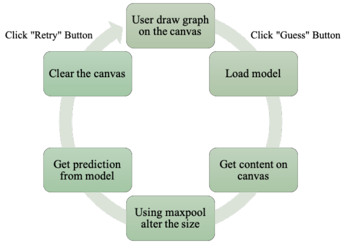


Figure 3. Application Flowchart

Future Work

- ➢ Create larger model to enrich our application
- ➢ Learn from users’ input

Reference

1. https://quickdraw.withgoogle.com/data
2. https://adventuresinmachinelearning.com/convolutional-neural-networks-tutorial-in-pytorch/