

DS 5220 Project Proposal

Chaitya Shah, Tyler Brown, Karan Gulur Muralidhar, Omair Shafi Ahmed; MSDS 19'

Description of the problem

Our team is basing our project idea on the “Quick, Draw!” Kaggle competition which classifies user drawings into one of 340 label categories. Drawings may be incomplete or not match the label. The challenge is to effectively build a recognizer that can work with this noisy data. The challenge is unique from an algorithmic perspective because the data includes both time series and pattern recognition components; areas that have traditionally used two different types of Neural Network Architectures. Our team uses a subset of the data to provide a solution to the Kaggle problem while taking a more in depth look at potential pitfalls and modeling strategy without having to use extensive computational resources.

Summary of the data

Google curated the Quick Draw Dataset as a collection of 50 million drawings mapped to 345 categories. The drawings were submitted by players of the game “Quick, Draw!” and captured as timestamped vectors, tagged with metadata such as the requested drawing category and country code. The format of each drawing array contains 3-dimensional vectors where components are the x,y coordinates and time in milliseconds since the first point. We use a preprocessed version of the dataset which is simplified and takes up about a tenth of the memory. The simplified dataset contains a CSV file for each of the 345 categories. Google was able to preprocess the data by performing the following steps: (1) align each drawing to the top-left corner, (2) uniformly scale each drawing, (3) resample all strokes with a 1 pixel spacing, and (4) simplify all strokes using the Ramer-Douglas-Peucker algorithm with an epsilon value of 2.0.

Methods

We plan to use a Neural Network for this problem. The problem is unique in that it has a time series component related to drawing stroke sequence, and a image recognition component related the cumulative strokes. Recurrent Neural Networks (RNN) are generally associated with time series. Convolutional Neural Networks (CNN) are often associated with image recognition. The inclusion of time series and pattern recognition components requires us to consider tradeoffs related to model accuracy, complexity, and algorithmic scalability. For example, RNNs have been shown to be less scalable than CNNs due to computational resource requirements. We have noted that researchers have attempted to capture benefits of RNNs in CNNs through the use of techniques such as dilation. We are still working through the best Neural Network Architecture for this problem.

Feature engineering will help us capture the temporal component of our data. We recognize that choosing appropriate features can reduce the amount of data required to train a Neural Network appropriately, as well as reduce the amount of computational resources required to fit our model.

Preliminary Results

Given that Neural Networks can be computationally intensive, one of our initial assessments was where and how to run these models. We tried replicating a CNN which had already worked for a team on Kaggle using the Google Colab resources. We were unable to replicate their results. We then reached out to the Instructor to explore alternatives such as better computers or using a subset of our data. We chose to use a subset of the data to focusing on modeling rather than computational infrastructure.

When visualizing our data, we were able to confirm that the data appears to be very noisy. We found that “Quick, Draw!” players interpreted categories such as “cat” correctly but used different perspectives. One drawing may be the face of a cat, another included a side-view with the face, body, and tail. Our model will need to account for these additional variations.

References

Image Based CNN (<https://www.kaggle.com/jpmiller/image-based-cnn>)