Identifying Drawings with the Quick Draw Challenge

Group 3

We're trying to put together a CNN based on the "Quick, Draw!" Kaggle competition (Google 2018).

Introduction

Provide a short background of the project (e.g., what kind of question is to be answered with this work). Provide a short non-technical summary of your analytical strategy.

Our team has based our project idea on the "Quick, Draw!" Kaggle competition which classifies user drawings into one of 345 label categories. Many drawings were incomplete or failed to match the label. The challenge was to effectively build a recognizer that could work with this noisy data. The challenge was unique from an algorithmic perspective because the data included both temporal and spatial components; areas that have traditionally used two different types of Neural Network Architectures. Our team attempted to provide a solution using a subset of the data, to the Kaggle problem, while taking a more in depth look at potential pitfalls and modeling strategy, without having to use extensive computational resources. With this project, we explore image classification in depth. We try out various experiments like fully connected neural network vs CNN, bias vs. variance tradeoff, overfitting data purposefully and trying out techniques like regularization and dropout to fix overfitting and finally, testing neural network with imbalanced data.

Methods

First, let's talk about data collection. Google initially compressed and converted the images to .npy files. These .npy files are basically 28*28 images, containing more than 100,000 rows each.

So here are the experiments we performed.

EDA

We checked the We plotted images from several categories

H_a : Prediction will go up when we use a CNN compared to a fully connected feedforward neural network.

H_a : Prediction will go up when we use regularization.

H_a : Run neural networks without any activations.

H_a : Try different architectures- number of layers, different (or no) activations

H_a : CNN with dilation is preferable to LSTM because it's presumably faster. We can cite some papers here.

Try out: Overfitting, Bias/variance, imbalance, dropout, pushing activation

Results

Here's how the results performed when we were exploring our hypotheses

Discussion

Here's what worked the best, worst, and future areas for research.

References

Appendix

Add computer code, plots, and other relevant technical details that will help me evaluating your work.

Statement of contribution

Google. 2018. "Quick, Draw! Doodle Recognition Challenge | Kaggle." https://www.kaggle.com/c/quickdraw-doodle-recognition.