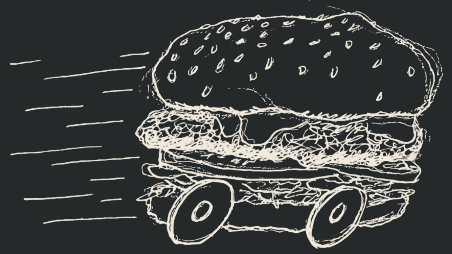
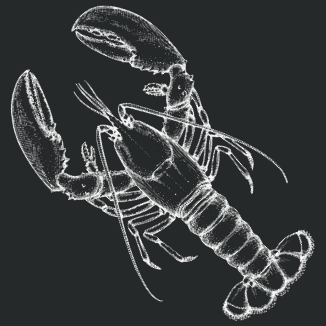
Guess-a-Sketch



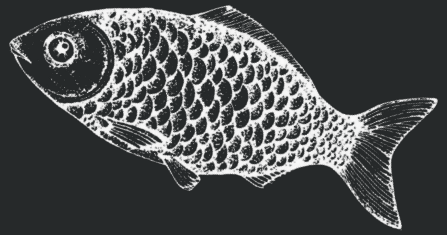


Lucia Zacek, Urvashi Deshpande, John Fernandez

The Setting:


skribbl.io!
Enter your name
English
Play!
Create Private Room

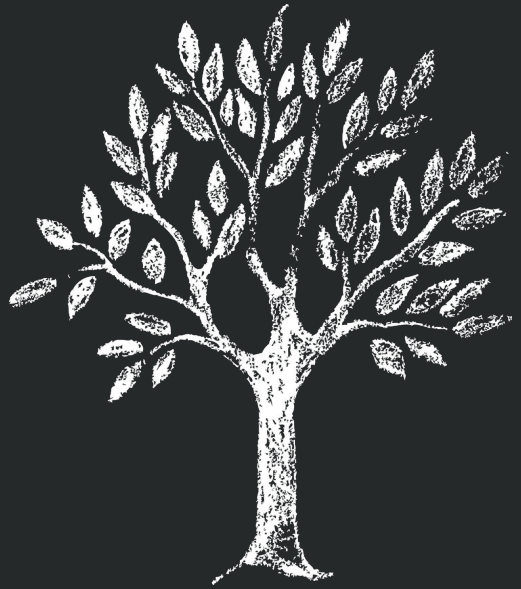

GUESS THIS
5
_ _ _ _ _



Our Goal:

Create a model that given word length:

- Labels black-and-white images
- Uses a word bank of 250 labels
- Guesses with high accuracy



The Setup

- Collecting/Separating Data
- Gaussian filters + downsampling data
- Processing data using DataLoaders

Evaluation:

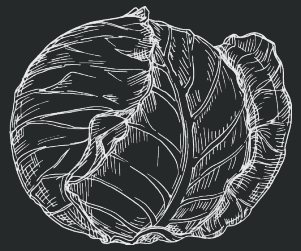
We will measure success by:

- Comparing prediction accuracy to randomly guessing
- Random guess rate overall is 0.4%
- Given word length random guess rate is 6.4%

Evaluation:

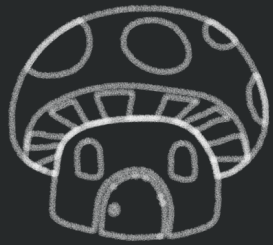
Further evaluation:

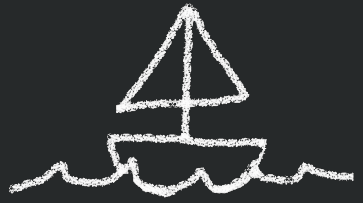
- What Skribbl.io assumptions can we take advantage of?
- Does CNN perform better than Random Forest, as we would expect?
- How well can we optimize the parameters of CNN for learning?



Random Forest: Design 1

- Identify pixels that are influential in distinguishing labels
- Train 10 random forest models total
- Each model distinguishes all 250 labels
- Computationally limited
- Testing accuracy only 2% (random guess 0.4%)





Random Forest: Design 2

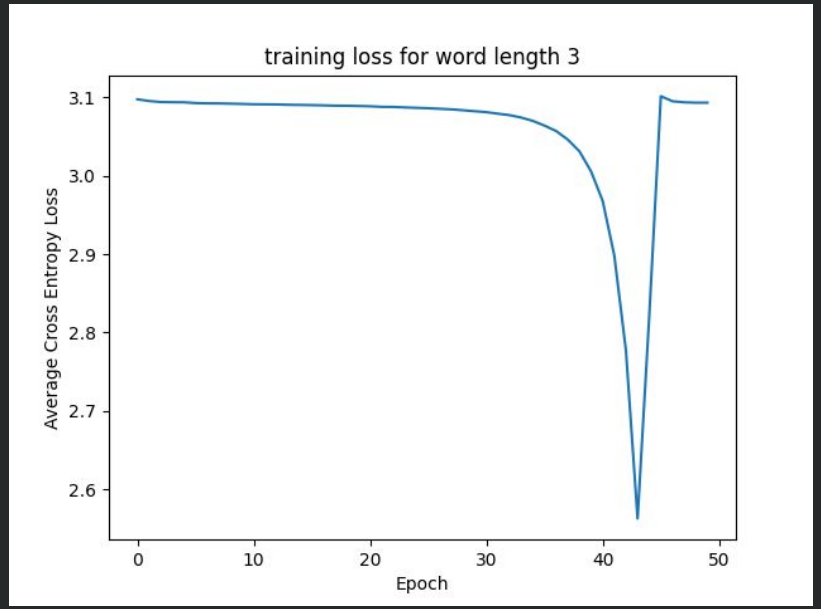
- Use the constraints of the skribbl.io setting
  - Length of label is known, sparse
- Train 16*10 random forest models
  - 10 per label length
- Accuracy: 21.84% (compared to 6.4% random guess)

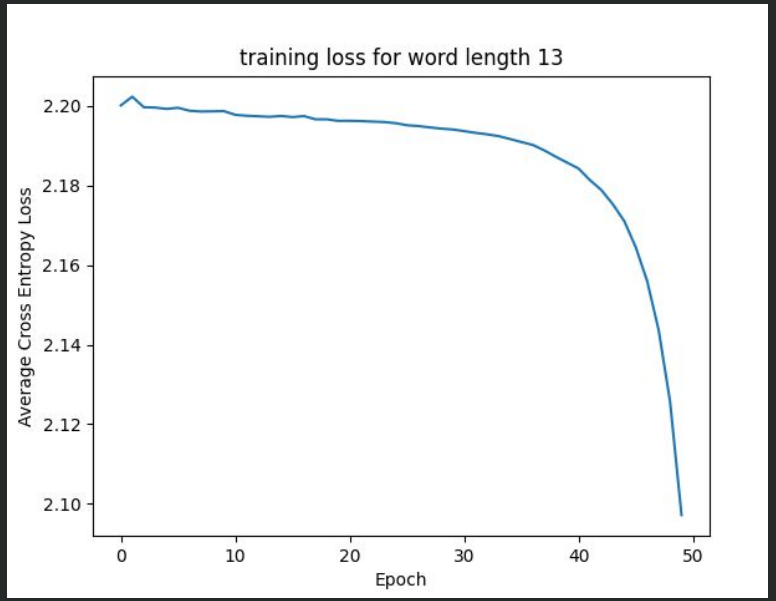
Convolutional Neural Net: Design 1

- 3 CNN layers with small input and output channels
- All images trained together
- 10 epochs, very small learning rate
- 0.4% accuracy because of unsuccessful learning

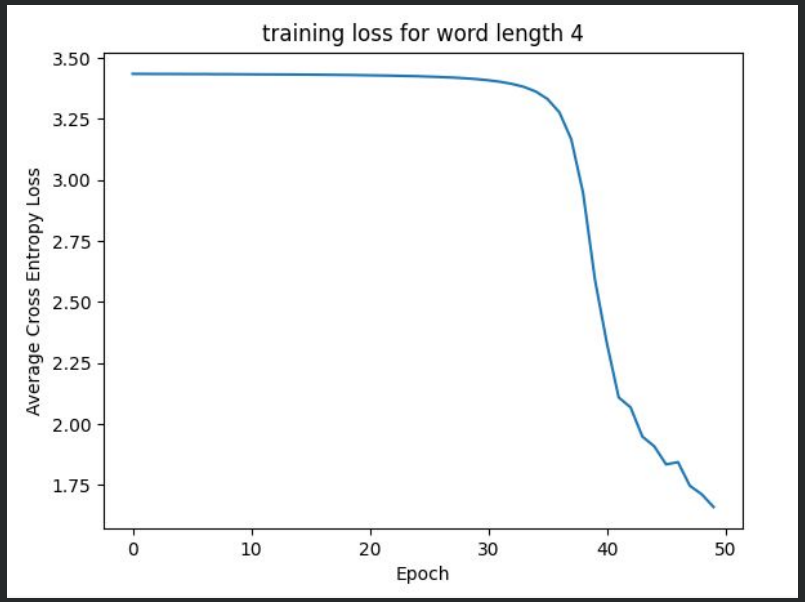
Convolutional Neural Net: Design 2

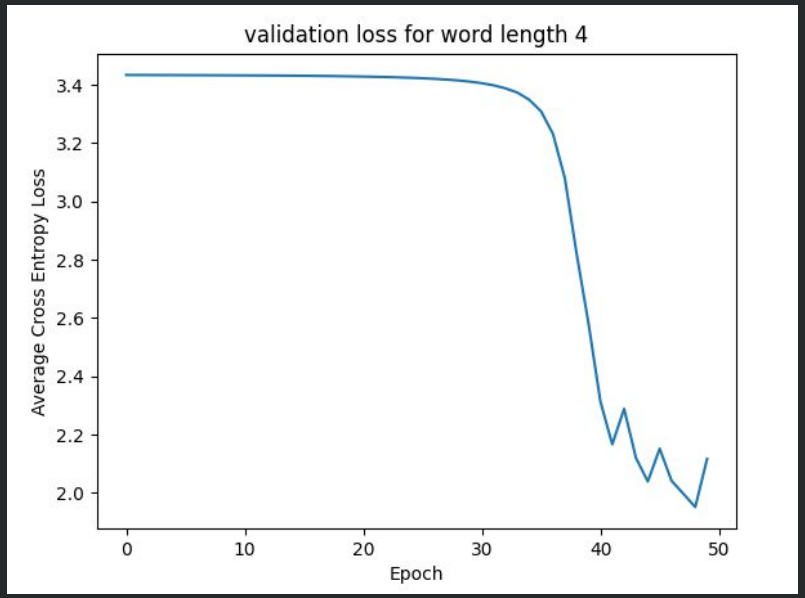
- 5 convolutional layers with larger dimensions + added max pooling layer
- Trained separate CNNs for each class word length
- Similar issue of guessing same label for every word
- Addition of blur and downsampling
- Improvement from random accuracy (~6%) to 34.6%





- added momentum and rate decay
  - Local minima, improve speed, reduce noise





- Final accuracy of 34.6%

Successes:

- Developed a successful model that was 6x better than guessing randomly
- Learned a lot by building RF model and from scratch
- Learned about CNN design features and optimization
- Learned about scripting for use in training/testing splits

Failures:

- Ran out of time to implement SVM model
  - Debugging RF and CNN took significant time
- Did not expand wordbank to full skribbl.io wordbank
- Were not able to crowdsource testing data

THANK YOU