Neural Net CAPTCHA Cracker

by

Geetika Garg

SJSU SAN JOSÉ STATE UNIVERSITY

Advisor: Dr. Chris Pollett

Committee Members : Dr. Thomas Austin and Mr. James Casaletto

SJSU SAN JOSÉ STATE UNIVERSITY

Agenda

- Introduction to Project
- Preliminary Work Summary
- Our Approach
- Neural Networks
- Experiments
- Results
- Conclusion
- Demo

SJSU SAN JOSÉ STATE UNIVERSITY

Introduction to Project

- **Tried decoding CAPTCHAs using deep neural networks.**
- CAPTCHA: Completely automated Public Turing test to tell Computers and Humans Apart.
- Helps to distinguish between humans and computers.
- First mentioned in a paper by Moni Naor [1] in 1996

Characteristics of a CAPTCHA:

- Easy for a human to decode
- Difficult for a Computer to recognize.

SJSU SAN JOSÉ STATE UNIVERSITY

Why Decode CAPTCHAs??

- AI Problem
- Security Breach

CAPTCHAs are critical for security on internet, if they are no more secure, our system won't be secure and we would have to think of alternatives.

SJSU SAN JOSÉ STATE UNIVERSITY

Preliminary Work Summary

Preliminary Work Summary

- Breaking CAPTCHAs is not a new concept.
- Mori and Malik [2] have broken EZ-Gimpy (92% success) and Gimpy (33% success) CAPTCHAs with sophisticated object recognition algorithms.
- People [3] also have used the following approach for CAPTCHA recognition:
  - Preprocessing.
  - Segmentation.
  - Training the model for individual character recognition.
  - Generating sequence with highest probability.

Problems in Segmentation

- Segmentation is difficult as some digits could be overlapping with some other digits.
- Deformity of digits is also a major concern. For example, digit “2” can have a larger loop or just a cusp.
- Character orientation. Characters could be rotated at arbitrary angles making recognition difficult.
- Unknown scale of characters. We do not know how big a character would be. So, it is not known how big the segmentation boxes should be.

SJSU SAN JOSÉ STATE UNIVERSITY

Idea behind our approach

- Yann Le Cun used neural networks for handwritten digits recognition in 1990[6].
- Google has published a paper[4] in which they used convolutional neural networks to detect home addresses from street view home plate images.
- One more paper from Google[5] in which they used recurrent neural network to generate caption for an image.

We tried combining these ideas !

SJSU SAN JOSÉ STATE UNIVERSITY

Our Approach

SJSU SAN JOSÉ STATE UNIVERSITY

Our Approach

- **End to End model**. Systems with multiple modules following conventional approach tend to behave poorly, because each module is optimized independently and the errors between modules compound. We learned an end to end model that predicts directly from pixels.
- **Convolutional neural network** for Image features and,
- **Recurrent neural networks** for generating output sequence.

SJSU SAN JOSÉ STATE UNIVERSITY

Neural Networks

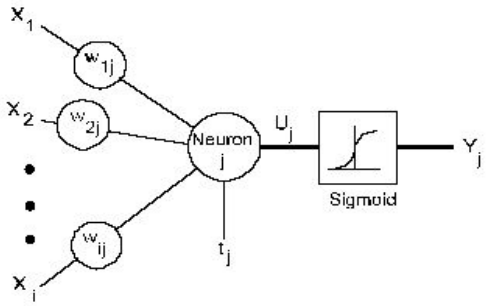
SJSU SAN JOSÉ STATE UNIVERSITY

Brief History of Neural Network

- Started way back in 1940's.
- Became unpopular in 1960's
- Regained popularity in 1980's
- Recently have become one of the hottest areas in the field of machine learning.
- Applications involve face recognition used by Facebook, Image captioning used by Google etc.

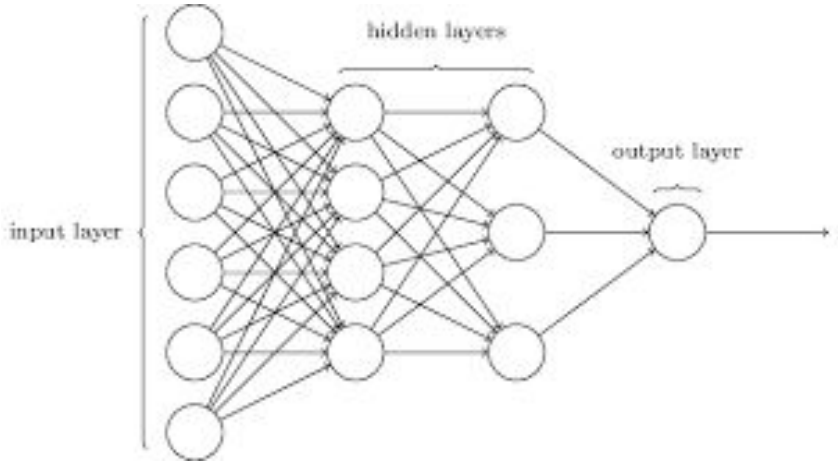
What are Neural Networks

- Inspired from human brain



A simple neural network model

- Many input units and one output unit.
- The inputs are scaled with weights on which an activation function is applied to get the output.



Multilayer neural network

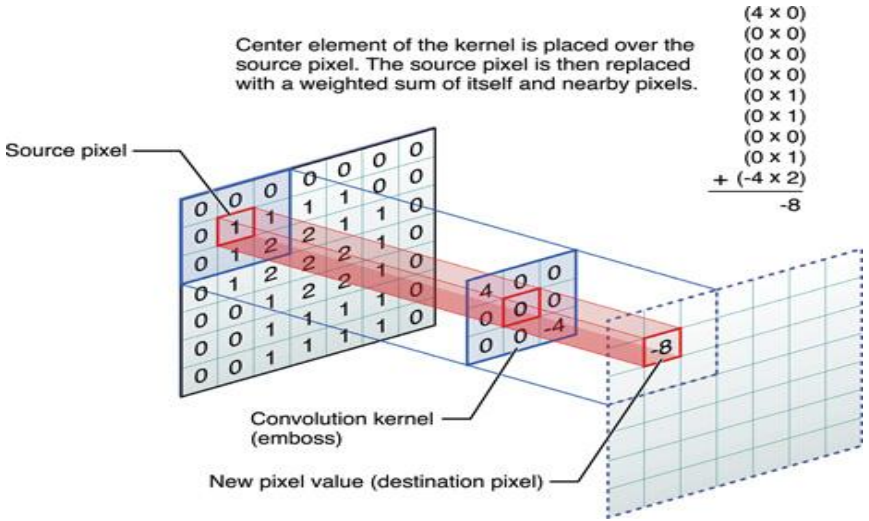
Training Neural Networks

- Backpropagation algorithm.
- The problem is set up as minimization of a loss (objective) function.
- Weights are adjusted using gradients and learning rate.
- Gradients are computed using simple derivative chain rule.

Convolutional Neural Network

- Convolutional Neural Networks are a special kind of multi-layer neural networks.
- In 1995, Yann LeCun Et al. introduced the concept of convolutional neural networks [7].
- Convolutional Neural Networks are designed to recognize visual patterns directly from pixel images with no preprocessing.

Convolutional layer



Feature Extraction

- Shared weights: The same filter weights are applied to all the pixels.
- It helps in detecting same feature at different locations of an image.
- This reduces the number of paramters to be learned.

Feature Extraction

For Instance, if image size is 200*50,

filter size is 5*5

and if there are 32 filters,

we have only 32*(5*5 +1) (1 for bias) = **832** weights to learn.

- Otherwise it would have been number_of_pixels*number_of_pixels*filters, which would be 200*50*200*50*32 = **3.2 million**.

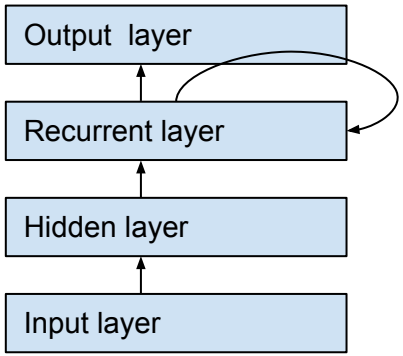
This is several orders of magnitude larger than what we have in CNNs.

Maxpool Layer

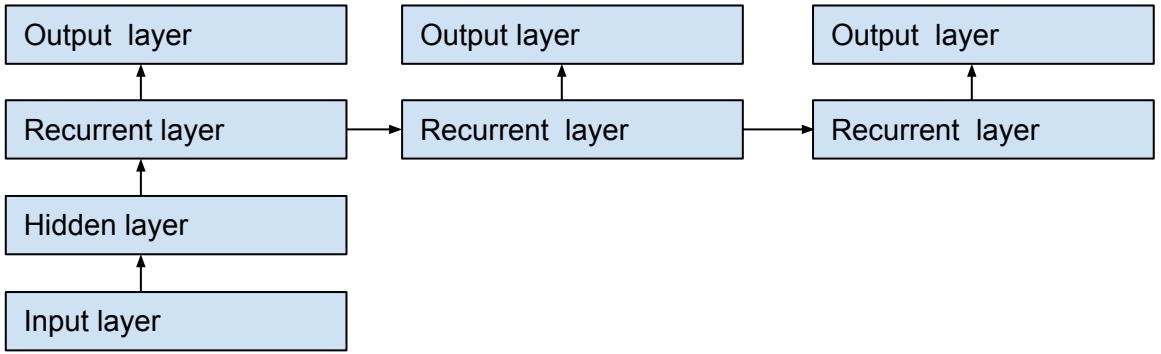
- Typically used after a CNN layer.
- Takes maximum of neighbouring pixels.
- Helps in rotational and translational invariance.

Recurrent Neural Networks

- Feedforward networks accept only fixed sized input and give output of fixed length, whereas RNNs can work with variable length inputs and outputs.
- In RNNs, connections between units have a directed cycle.
- Various applications of RNNs include handwriting recognition and speech recognition.



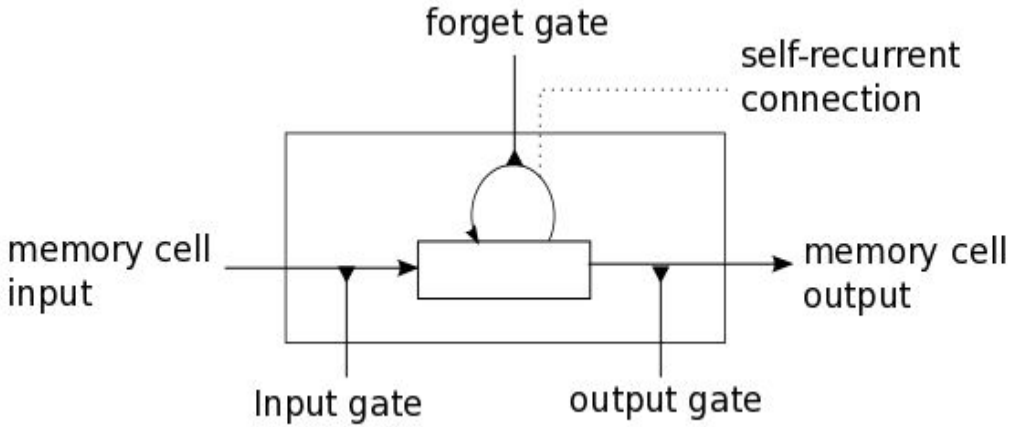
RNN



RNN unrolled

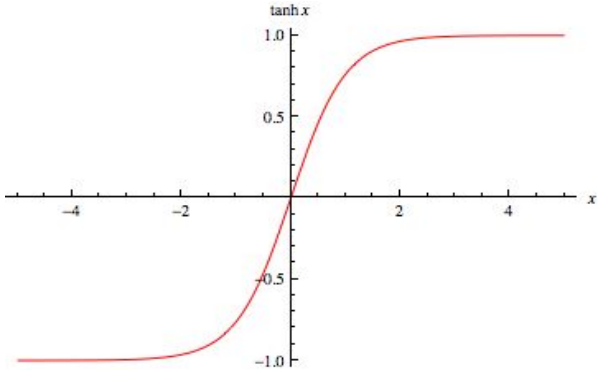
LSTMs

- "Long Short Term Memory" networks are special kinds of RNNs.
- **Vanishing** and **exploding gradient** makes RNNs difficult to train.
- But in LSTMs, the error gets trapped in memory.



LSTMs

- Uses tanh ((e^{x} - e^{-x})/(e^{x} + e^{-x})) activation function.
- Forward and backward LSTMs



tanh function

Softmax Layer

- Used for generating a probability distribution.
- Used typically in classification problems
- In our model, we use it to estimate probability of every character.

SJSU SAN JOSÉ STATE UNIVERSITY

Framework and Dataset

Frameworks Used

- We have used Theano.
- It is a publicly available, flexible library which optimizes, and evaluates mathematical expressions efficiently.
- It is available in Python so was easy to integrate with our project.
- It makes use of GPUs if present making tasks faster.

Frameworks Used

- **Lasagne** is a Python package to train neural networks. It uses Theano internally**.**
- It implements LSTM. Theano by itself does not have implementation of LSTMs.
- It implements the framework to keep track of all the neural network parameters like weights and biases. It makes it easy to save the parameters and initializes the model with pre-trained weights.

Dataset Used

- Training requires lots of images.
- A standard dataset for CAPTCHAs is not available publicly.
- We generated dataset synthetically.
- Java module to generate CAPTCHAs with randomization of noise, characters and backgrounds.
- We generated fixed as well as variable length CAPTCHA dataset.
- 1 million simple images, 2 million complex images of fixed length(5), and 13 million images of variable length.

Dataset

Complex image

Simple Image

- An image contains 4-7 characters, if it is variable length dataset or 5 characters if it is a fixed length captcha.
- A character could be A-Z, a-z or 0-9.
- All the images generated are of same size i.e. (200*50).

SJSU SAN JOSÉ STATE UNIVERSITY

Experiments

Convolution layers

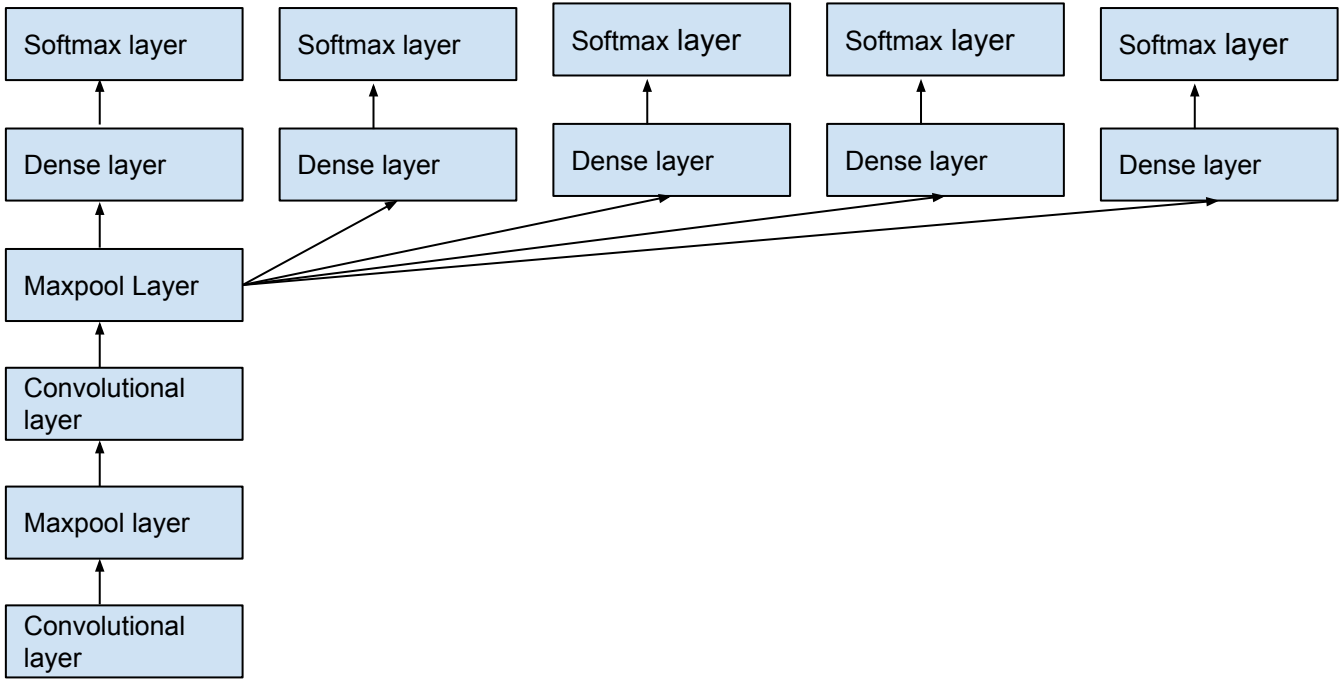
- Input image was converted to single channel using “L” conversion.
- Number of filters used - 32
- Size of a filter - 5*5
- Zero padding
- Maxpooling done over 2*2

Fixed length Model

2 CNN layers to learn image features.

A dense layer and a softmax layer to predict every character.

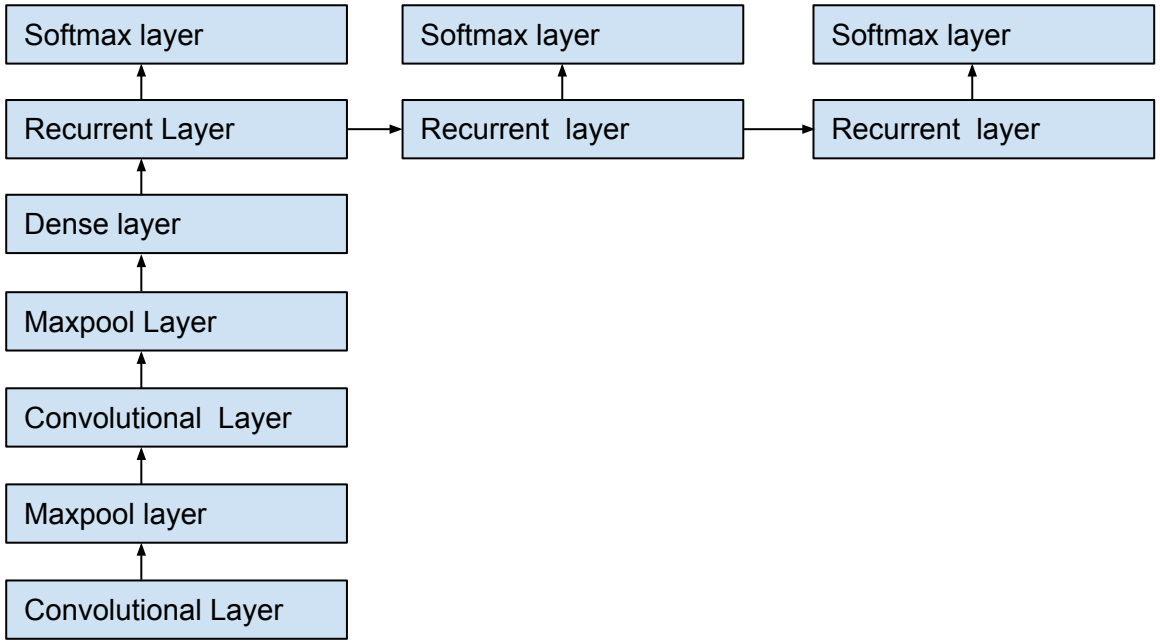
Softmax lfgayers at the top share weights



Variable length model

2 CNNs to learn image features.
RNN to generate output sequence.
The same dense layer vector was inputted for every step of RNN.

RNNs



Training

- Batch Size 1024 (images).
- Gradient clipping: We think that exploding gradients were pulling the model too far in different directions. Gradient clipping gave better results which might have helped in dampening the effect of exploding gradients.
- Special id 0 for ‘unk’ was used to signal termination.
- Images of different captcha length were batched together for randomization.
- High learning rates caused instability. So had to gradually decrease it, as the models would get stuck with high learning rates.

SJSU SAN JOSÉ STATE UNIVERSITY

Results

SJSU SAN JOSÉ STATE UNIVERSITY

Results

Type of model	Individual Character Accuracy
LSTM fixed length (simple dataset)	99.9%
LSTM fixed length (complex dataset)	98.48%
Multiple Softmax fixed length (simple dataset)	99.8%
Multiple Softmax fixed length (complex dataset)	98.96%
LSTM variable length with fixed length data	99.5%
LSTM variable length with variable length data	97.31%

SJSU SAN JOSÉ STATE UNIVERSITY

Results

Type of model	Sequence Accuracy
LSTM fixed length (simple dataset)	99.8%
LSTM fixed length (complex dataset)	91%
Multiple Softmax fixed length(simple dataset)	99%
Multiple Softmax fixed length (complex dataset)	96%
LSTM variable length with fixed length data	98%
LSTM variable length with variable length data	81%

SJSU SAN JOSÉ STATE UNIVERSITY

Human Vs Computer

Human(Me): 3/ 10 were wrong

Computer: 1/10 was wrong

Human 2: 2/10 were wrong

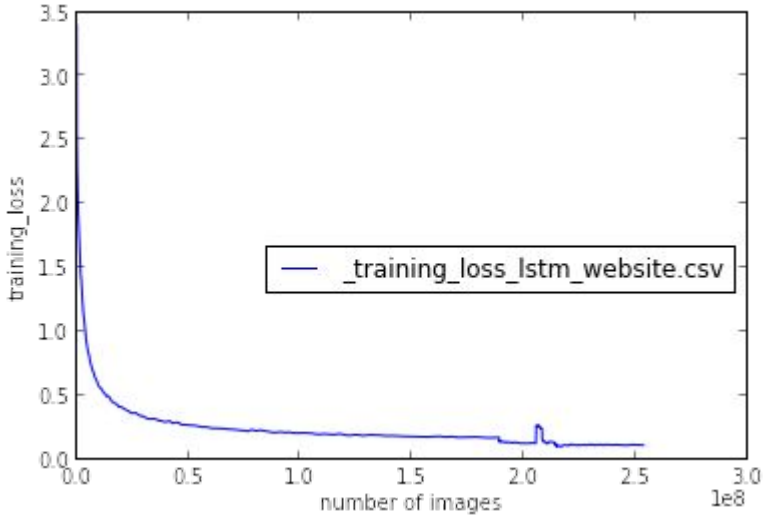
Computer: 0/10 were wrong

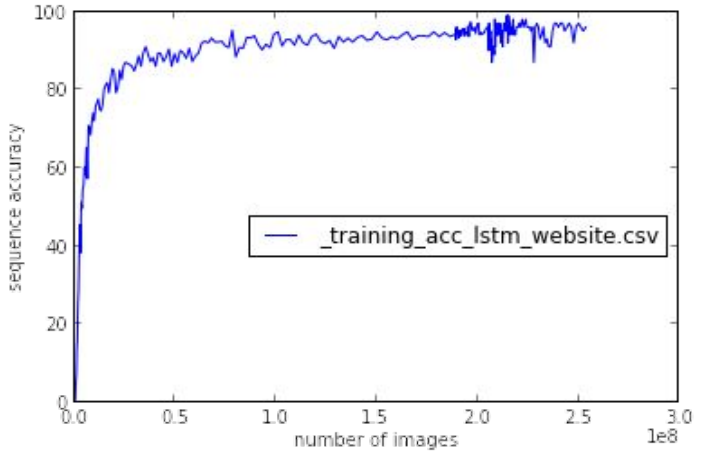
Human 3: 4/15 were wrong

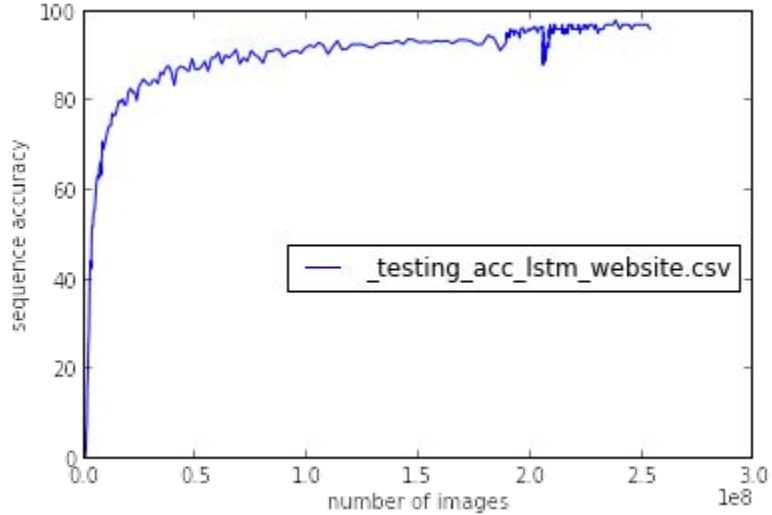
Computer: 2/15 were wrong

SJSU SAN JOSÉ STATE UNIVERSITY

Graphs Generated

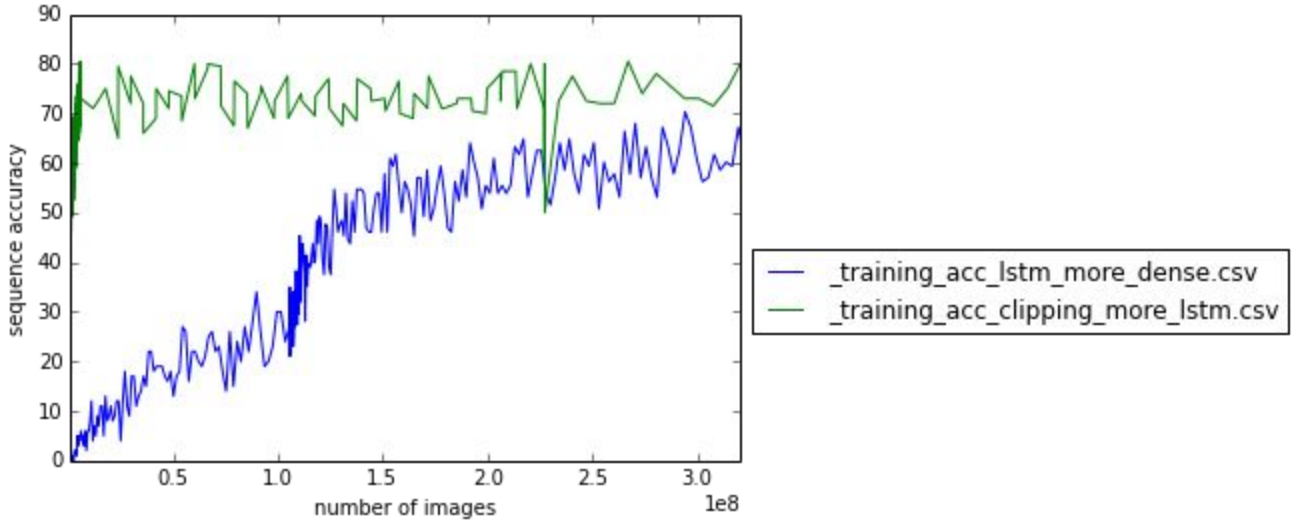





100
80
60
40
20
0
sequence accuracy
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e8
number of images
_testing_acc_lstm_website.csv

SJSU SAN JOSÉ STATE UNIVERSITY

Variable length model variants



SJSU SAN JOSÉ STATE UNIVERSITY

DEMO

SJSU SAN JOSÉ STATE UNIVERSITY

Conclusion

Conclusion

- Deep neural networks showed a really good performance in decoding CAPTCHAs with 80% and 99.8% accuracy for variable and fixed length CAPTCHAs respectively.

- CAPTCHAs are not more secure as computers can do better than humans.

SJSU SAN JOSÉ STATE UNIVERSITY

Future Work

Future Work

- Will try to work on accuracy using more convolutional layers.
- Will make the system robust by increasing the variety in training data.

Project Links

Demo website:

http://cp-training.appspot.com/

GITHUB:

https://github.com/bgeetika/Captcha-Decoder/

SJSU SAN JOSÉ STATE UNIVERSITY

References

[1] Moni Naor. Verification of a human in the loop or Identification via the Turing Test. Unpublished Manuscript, 1997.

[2] Greg Mori and Jitendra Malik. Recognising Objects in Adversarial Clutter: Breaking a Visual CAPTCHA, IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03), Vol 1, June 2003, pp.134-141.

[3] Kumar Chellapilla, Patrice Y. Simard Using Machine Learning to Break VisualHuman Interaction Proofs (HIPs) Microsoft Research, one microsoft way, WA 98052 -2005

[4] Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, Vinay Shet. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks,14 Apr 2014.

[5] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan, Show and Tell: A Neural Image Caption Generator, 20 Apr 2015

[6] Y.Le Cun Et. al. Handwriting Character Recognition using Neural Network Architecture.1990

[7] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time-series. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995.

SJSU SAN JOSÉ STATE UNIVERSITY

References

[8] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In S. C. Kremer and J. F. Kolen, editors, A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press, 2001.

[9] H. Jaeger. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. Science, 2004.

[10] W. Maass, T. Natschläger, and H. Markram. A fresh look at real-time computation in generic recurrent neural circuits. Technical report, Institute for Theoretical Computer Science, TU Graz, 2002.

[11] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber. A Novel Connectionist System for Improved Unconstrained Handwriting Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, 2009.

[12] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, Y. Bengio - Theano: a CPU and GPU Math Expression Compiler.

SJSU SAN JOSÉ STATE UNIVERSITY

QUESTIONS?

SJSU SAN JOSÉ STATE UNIVERSITY

THANK YOU !!