Final Project. CAPTCHA Image Recognition using CNN and LSTM

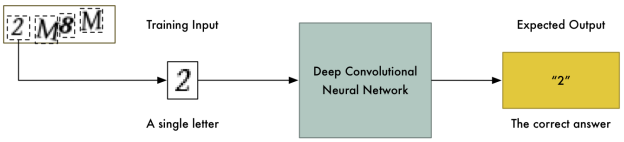
Sangjun Son
Seoul National University
Department of Computer Science and Engineering
lucetre@snu.ac.kr

1. Introduction

Everyone hates CAPTCHAs — those annoying images that contain text you have to type in before you can access a website [1]. A CAPTCHA (an acronym for "Completely Automated Public Turing test to tell Computers and Humans Apart") is a type of challenge–response test used in computing to determine whether or not the user is human [2]. It is to prevent computers from automatically filling out forms by verifying that you are not a robot. But with the rise of deep learning and computer vision, decoding CAPTCHA seems to be possible and some of those systems are even more better than human.

Given an image, how can we recognize characters from the CAPTCHA image? Since each CAPTCHA image has a variable length of characters, considering length will be a significant work while training through deep neural networks. If we can split the image apart so that that each letter is a separate image, remaining work is just to train the neural network to recognize a single letter at a time[1].

We've trained our network with 10,000 training examples whose lengths are in $2 \sim 5$ and tested with 1,000 test images of length $2 \sim 7$. Due to several parameters consisting the ensemble model of CNN and LSTM, we've conduct various experiments. After several experiments, we've found the best set of hyper-parameters that make our model perform more accurately and even faster.

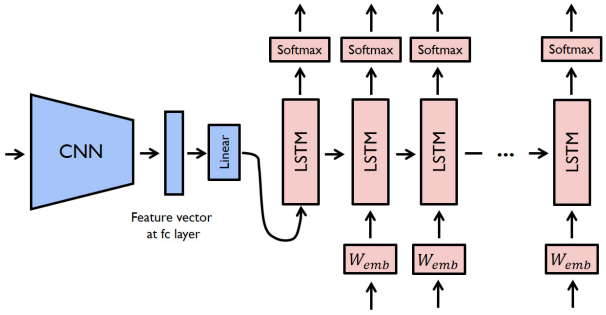


2. Related Works

CATCHA recognition is one part of temporal pattern recognition projects. Simiarly, there are numerous examples like image captioning, speech recognition, part-of-speech tagging. Using ResNet and LSTM modules, which are the representatives of CNN and RNN methods, we can extract useful features from an image and continuously learning them over time.

3. Proposed Method

This section describes overall flows of our proposed method, `BetterNet+LSTM`. Feature extraction and learning the temporal patterns are the key concepts of our model.



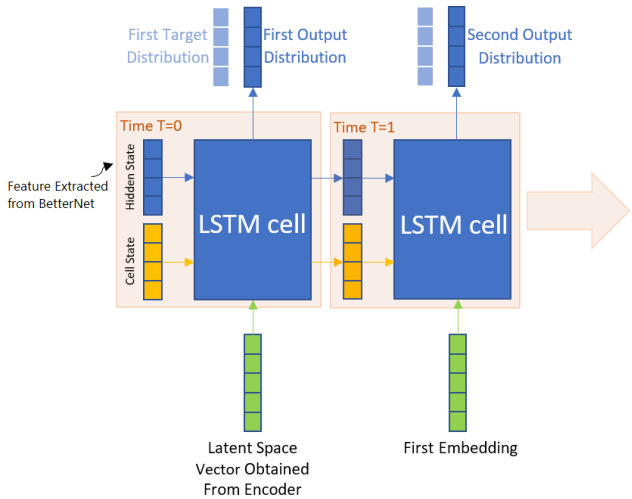
3.1. Feature extraction by BetterNet

Since image vectors contain lots of information, we have to extract meaningful features. There exist lots of models like AlexNet, GoogLeNet and so on. We've implemented our own CNN module, but it showed too much low performance since it has few parameters for feature extraction due to shallow depth. `BetterNet` based on ResNet has been chosen to be our optimal CNN model. Comparison between our old model and `BetterNet` shows remarkable model complexity gap.

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	# of character sets * max length of CAPTCHA	512×259 fully connections

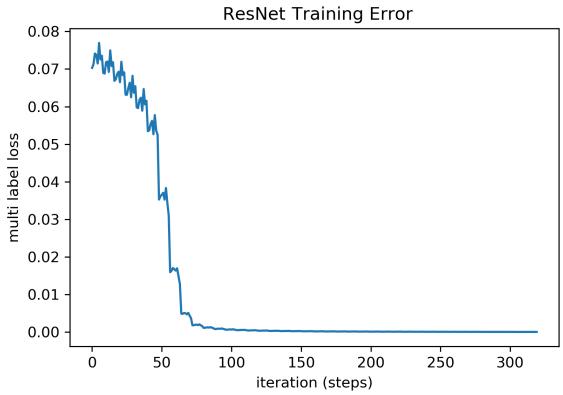
3.2. Learning temporal patterns by LSTM

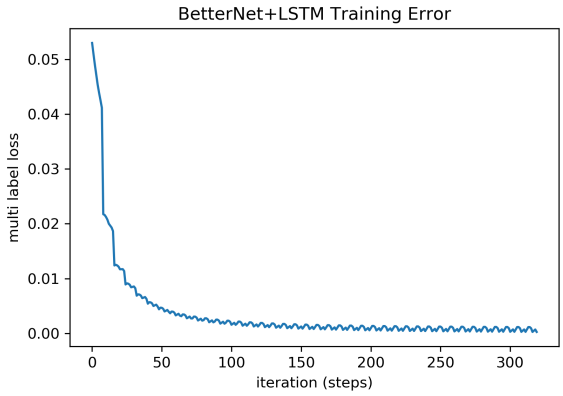
The pretrained CNN extracts the feature vector from a given input image. The feature vector is linearly transformed to have the same dimension as the input dimension of the LSTM network. Using these source and target sequences and the feature vector, the LSTM decoder is trained as a language model conditioned on the feature vector [3]. Putting together, our model `BetterNet+LSTM` was trained by Adam: A Method for Stochastic Optimization[4] with learning rate of 0.01 and maximum iteration of 200. `BetterNet+LSTM` shows fast and accurate performance of 303 sec as training time and 1.99e-4 as training multi-label loss. After testing 1,000 images, both character and word correctness of our model were 100.0%.



4. Empirical Analysis

Here we describes how we came up with `BetterNet+LSTM` model. Comparison analysis of `ResNet` and `BetterNet+LSTM` in aspect of time and accuracy in training is as the figure below. Both were trained by ADAM method with the same tolerance, maximum number of iterations and the learning rate of 1e-4, 200, 1e-2 w.r.t.





4.1. Training Approach

`BetterNet+LSTM` showed instantly decreasing training multi-label loss compared to `ResNet`. LSTM which performs teacher forcer learns temporal patterns from observation vectors again and again. After plotting the graphs with those two models, we concluded that `BetterNet+LSTM` is needed to have more complexity to learn more latent features of images, and higher tolerance to detect convergence of our model quickly.

4.2. Test Approach

We construct `BetterNet+LSTM` and tried to examine test aspect performance by varing many hyperparameters. Since LSTM modules take one-hot encoding captions as inputs, our model can easy find temporal patterns on CAPTCHA labels while training. However in test, foward algorithm in LSTM cell take those ground truth as inputs and `BetterNet+LSTM` trained test data even before comparison between predicted and true labels. By the approach,

we found out the reason why our model performs 100.0% test accuracy.

5. Conclusion

We've selected ResNet and LSTM for CNN and RNN modules respectively. In result, `BetterNet+LSTM` shows fast and accurate performance while training, and predicted every label of test images correctly. However, we found out that we were testing our model while training and have to reconstruct the entire LSTM module. In future works, we'll rebuild our model which has forward algorithm not to take captions as inputs and train with numerous instances and diverse CAPTCHA examples. We'd like to extract more relevant features from the CNN module even when complexity reduction.

References

[1] Adam Geitgey, "How to break a CAPTCHA system in 15 minutes with Machine Learning", Dec. 2017

[2] Adam Geitgey, "The reCAPTCHA Project – Carnegie Mellon University CyLab". *www.cylab.cmu.edu. Archived*, Jan. 2017

[3] yunjey, "Image Captioning", *GitHub pytorch-tutorial*, Nov. 2018

[4] Diederik P. Kingma, Jimmy Ba, "Adam: A Method for Stochastic Optimization", *3rd International Conference for Learning Representations*, San Diego, 2015