Machine Learning: Mini Project Report

Karam Rai -17554

Kelam Goutam-17555

AIM :- To build a machine learning model for Captcha Cracking

PLATFORM USED :- Tensorflow with Anaconda Python (3.6.3)

MODEL USED:- Softmax and Convolution Neural Network

It is a generalization of logistic regression to the case where we want to handle multiple classes.

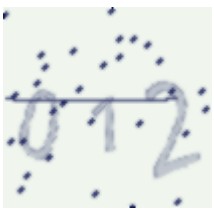
In the Softmax setting, we are interested in multi-class classification, and so the label 'y' can take on K different values, rather than only two. Thus, in our training set $\{ (x^1, y^1), \dots, (x^m, y^m)\}$, we now have that $y^i \in \{1,2, \dots, K\}$. For example, in the MNIST digit recognition task, we would have K = 10 different classes.

We have used Softmax Cross Entropy with Logits as our Loss Function and we have used Gradient Descent Optimizer.

EXPERIMENTS:-

Used Softmax classifier on raw captchas that were generated using 'Captcha' library. The generated captchas consist of digits. We experimented with different sizes of train set and test set.

The generated captchas look like:



Results:

We have considered maximum step-size of 10000 and batch-size of 1000

# of digits	*Training Size*	*Test Size*	*Accuracy*
1	7200	1440	42.99%
2	20000	4000	66.99%
3	20000	4000	64.10%
4	20000	4000	72.03%

Observations on Softmax Model:-

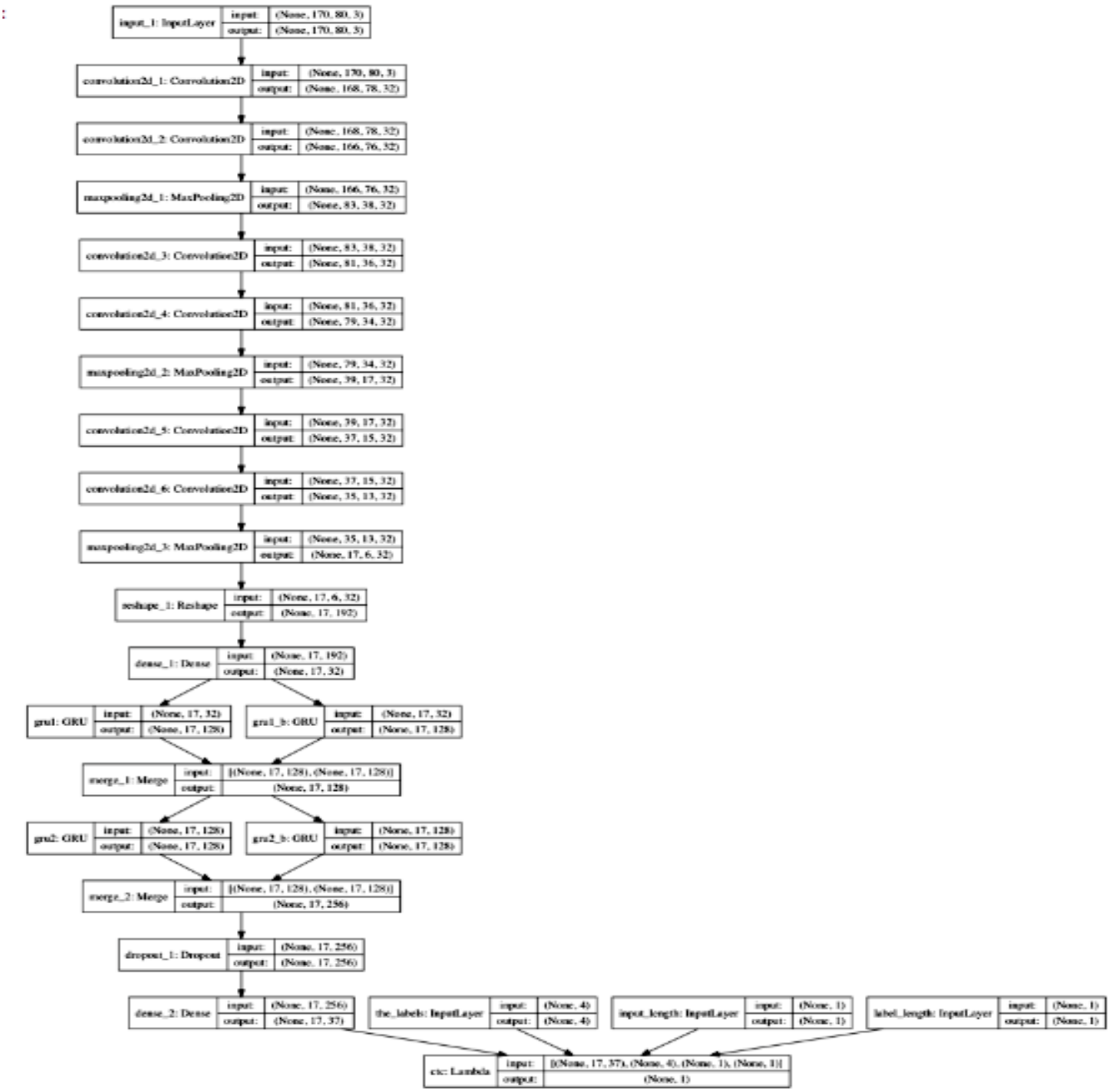
1. As we increase the number of digits in captcha, time to generate the captcha dataset increases.
2. We got good accuracy for captchas with more no of digits compare to captchas with less digits.

The accuracy obtained using Softmax was not satisfactory. The reason we feel for low accuracy of Softmax model is that the Softmax model was trained on MNIST datasets which does not contain noisy data and does not take into consideration of the rotation in the digits. But, our generated captchas include noise and a bit of translation and rotation of digits.

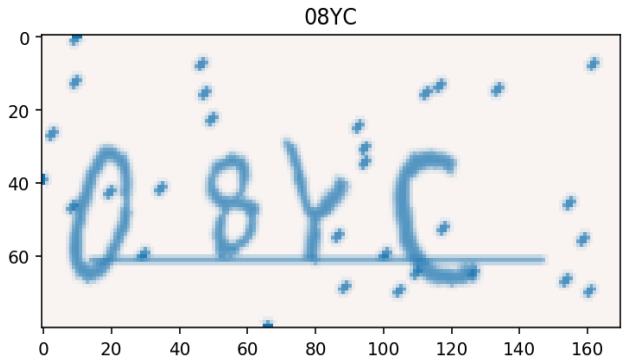
To handle this problem, we tried cracking captchas using the Convolution Neural Network since we know that CNN just looks for the presence of the object and identifies the object even if it is rotated or translated.

In our CNN model, we have an input layer, six convolution layers which use ‘relu’ as activation and three maxpooling layers. We have used CTC Loss function and ADADELTA optimizer.

The architecture is shown below:

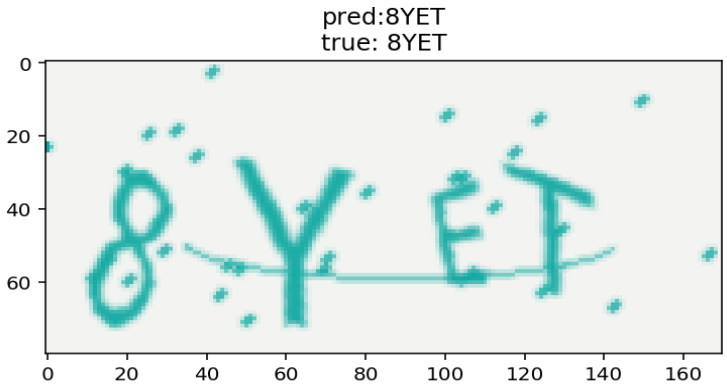


We generated the captcha dataset. The generated captchas look like:



We trained our CNN model using data size of 5000 captchas and we tested our trained model on data size of 1000 captchas. We ran for 3 epochs and got the accuracy close to 98%.

Result:-



Observations on CNN Model:-

This shows that a CNN model is far better compared to the Softmax model for cracking captchas which contain noise as well as translated alpha-numeric characters.

Future Work:-

We have worked with the digit captchas in case of Softmax Model. So, in future we can work with captchas containing both digits as well as alphabets (both uppercase and lowercase).

In case of CNN model, we can try different optimizers, activation functions and loss functions. Due to time constraint, we could not analyze the other optimizers, activation functions and loss functions.

CONCLUSION :-

In this mini-project we tried cracking captchas using two models: Softmax and CNN.

Using Softmax model we got a maximum accuracy of 72% for four digit captchas. Then, we generated captchas containing four alpha-numeric characters and using CNN model we got accuracy close to 98%.