CS 552 Data Science with Python Project 4

Hikmet Bahadir Sahin, 011374

1 Introduction

In this project, we are asked to create a machine learning model to break Captcha images with varying length of digits from 2 to 5.

I propose a not-so-deep convolutional neural network-based solution which is developed by using Tensorflow and trained by using GPU. I train my model by using 12000 training data which have 5-digit labels. The validation and test sets are contains 1500 data for each. In the following section, the network architecture is visualized and each layer will be explained with more details.

2 Detailed Network Architecture Description

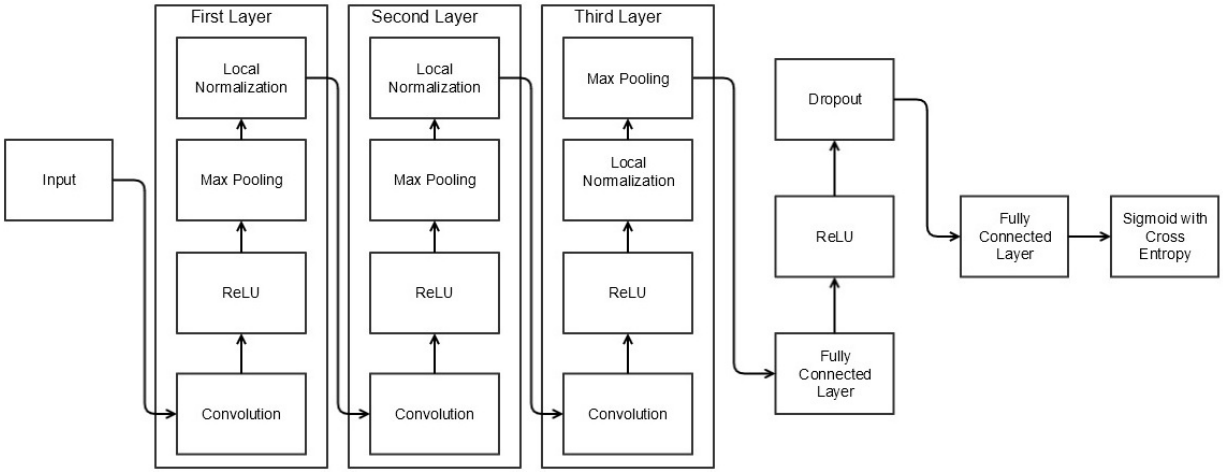


Figure 1: Overall network architecture that is used in this Project.

In Figure 1, the network architecture is illustrated. It contains three convolutional layers with local response normalization (LRN) [1] layers embedded. The aim of the LRN is to normalize the learnt features such that their mean is 0 and variance is 1. By applying such normalization on the input features, one can make stochastic optimizers (gradient descent, rmsprop, adam, etc.) job easier. Then, the learnt features from the convolutional layers are classified in the following fully-connected layers. Since the number of labels is varying depending on the length of the digit, instead of using a softmax loss function, I decide to use sigmoid cross entropy as my loss function.

First convolutional layer has 48 filters with 5x5 window size. Second convolutional layer has 64 filters with 5x5 window size. Last convolutional layer has 128 filters with 5x5 window size. All convolutional layers has 2x2 stride property. All max-pooling layers has 2x2 window size with 2x2

[1] http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

stride property. All LRN layers has same parameters such that depth radius is 4, bias is 1.0, alpha is $\frac{0.001}{9.0}$, and beta is 0.75. Fully connected layer has been initialized by 2048 neurons. Dropout is applied with 0.7 probability in training and not used in (probability = 1.0) validation/test process.

3 Experiments

3.1 Training Setup

The training, validation and test sets are created by using the given code part from the lectures with slight modifications. Since the model aim is to identify CAPTCHA's with 2, 3, 4, and 5 digits, I decide to train whole model by creating a dataset with 5-digit captcha. I created 15000 images with initial size of 60x160 and down-sample them to 48x128. Initial size is determined in a way that all 5 digits are ensured to be visible in the generated image. %80 of images are used as training set and rest of the images are divided into validation and test sets equally.

The epoch count for the training is set to 50 and batch size for each is initialized as 50. Adam is selected as the stochastic optimizer with initial learning rate 0.001.

Note that GPU-based training is conducted in this project and whole training took less than 5 minutes (generating data excluded). In a CPU-based training training times can take up-to 30 minutes.

3.2 Training Loss

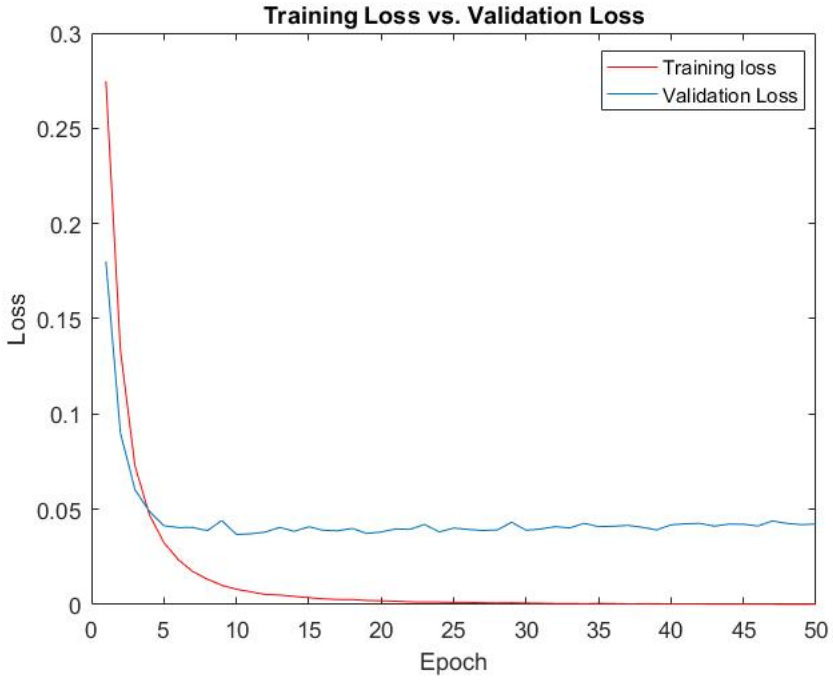
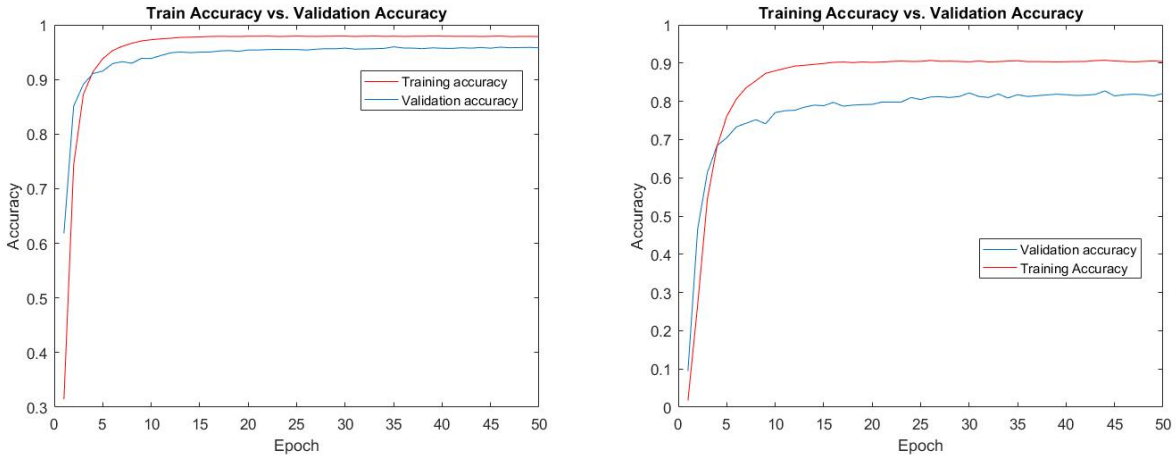


Figure 2: Loss change during training

3.3 Training-Validation Accuracy



(a) (b)

Figure 3: Training and validation accuracy change during (a) digit-level (b) full number

3.4 Test Results

	2-digit	3-digit	4-digit	5-digit
Test Accuracy (w.r.t. digits)	0.9356	0.9462	0.9495	0.9534
Test Accuracy (w.r.t. full number)	0.866	0.8713	0.8493	0.8126

Table 1: Test accuracies for randomly generated, different-sized CAPTCHA's

As it can be seen from the Table 1, the learnt model provides accuracies above %93 if the accuracy is calculated digit-by-digit. For instance, if the ground truth is "12" and the prediction of the model is "11", the accuracy is determined as %50 for this case since the first digits are equal while the second digits are not.

The performance of the model drops by approximately %10, when we calculate accuracy with respect to full number equivalence (prediction is completely equal or not to the ground truth). However, the learnt model can predict 5-digit length numbers with %81 accuracy and the performance increases while digit length decreases.