Bangla character recognition using deep neural network

Muhammad Rifayat Samee

Hand written character recognition is a common problem that has been extensively studied for many different languages. However, there has been very little work on Hand written character recognition for Bangla language. In this project, I have tried Convolutional deep neural network to classify Bangla hand written characters. I have used data set from CMATERdb[1] pattern recognition database repository. The model achieved test set F1 score of 93.26%.

Brief introduction to Bangla language

Bangla Language is spoken by around 215 million people and it is native to around 196 million people. There are 50 basic letters in Bangla language. Among them 11 vowels and 39 constants. However, basic letters can be compounded to form many different compound characters. In this project, I will try to classify only the basic 50 letters.

Description of the Dataset

Data set that was used in this project is from CMATERdb[1] pattern recognition database repository. There are total 50 different characters from the 50 basic Bangla letters. For each class of characters, there are 240 training data and 60 test data. In total, the training folder consists of 12,000 images and the test folder consists of 3,000 images. Images are of random dimensions. Most of the images are of white background and image itself being black. Some of the images have grey background.

Data pre-processing

The data set obtained from CMATERdb3.1.2 pattern recognition database has different dimensional images. So, each image has been resized to 50*50 dimension. There are many images with Grey background. Grey background was replaced with white background. To reduce the computational cost, for each image the color was inverted. It means making black pixel white and white pixel black. This is computationally less costly because, doing the inversion will fill the image matrix with mostly zeros (number of zeros in the matrix will be maximized). All the images were normalized to have approximately zero mean and standard deviation ~0.5 before training phase**.**

The model

Dependencies:

Pytorch 0.4

numpy

sklearn

Model Structure:

```
Model( (conv1): Conv2d(1, 16, kernel_size=(3, 3), stride=(1, 1))
  (conv2): Conv2d(16, 16, kernel_size=(5, 5), stride=(1, 1))
  (fc1): Linear(in_features=1600, out_features=400, bias=True)
  (fc2): Linear(in_features=400, out_features=200, bias=True)
  (fc_drop): Dropout(p=0.5)
  (fc3): Linear(in_features=200, out_features=50, bias=True)
  (conv_drop): Dropout(p=0.5)
)
```

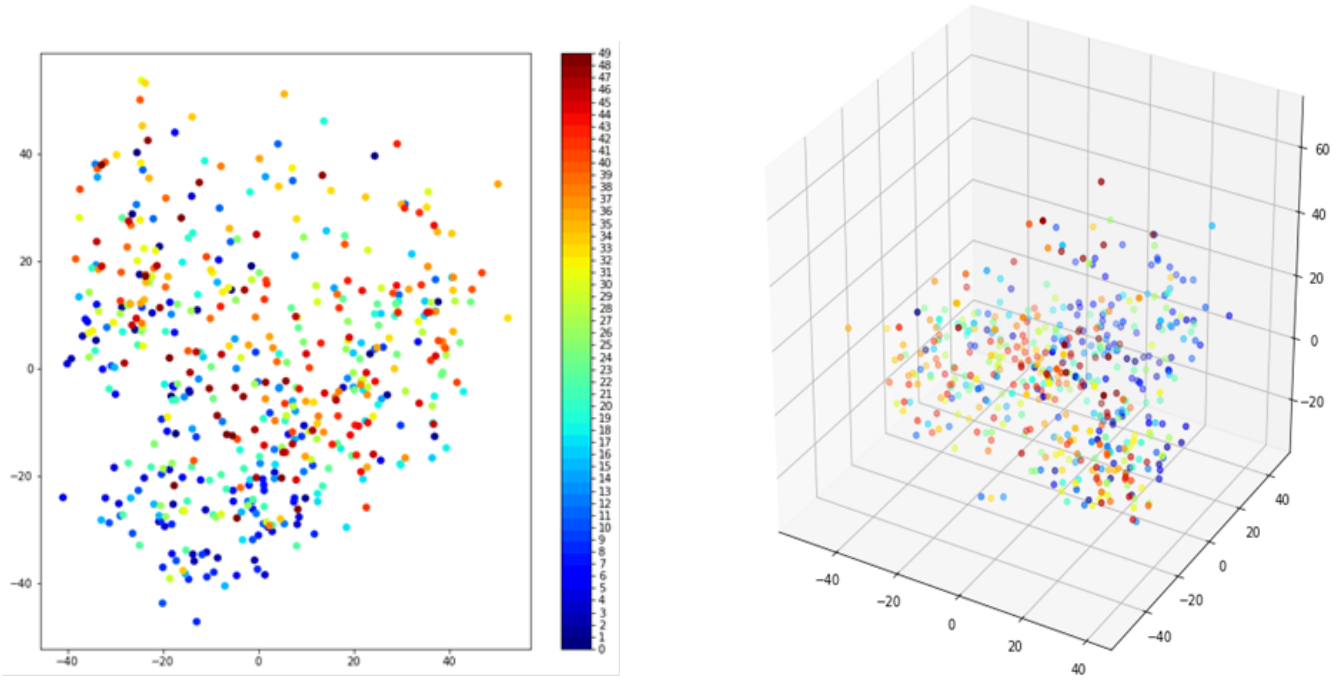
Optimizer used : Adam with initial learning rate 0.001 and weight_decay was 1e-5

Total training time : 2.5hour on CPU

Some Figures

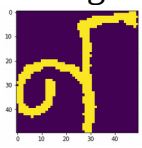
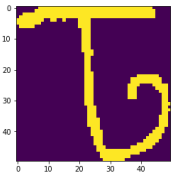
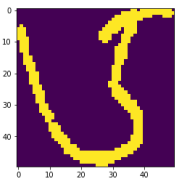
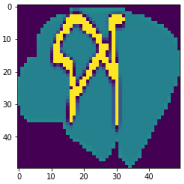
Figure 1: Sample Dataset

Figure 2: After Pre-processing



Projection of sample data in 2D and 3D space using ISOMAP

Some frequent misclassification by the model

Image	Predicted class	True Class
	ন	ণ
	চ	ঢ
	ভ	ড
	শ	প