Improvement suggestion using adversarial examples

by

Gautam Kumar Shom

Report
Submitted by
Gautam Kumar Shom

Supervisor: Dr. Nabeel Mohammed
Associate Supervisor: Dr. Sifat Momen

University of Liberal Arts Bangladesh
January,2017

Contents

List of Figures

Improvement suggestion using adversarial examples

Gautam Kumar Shom
gautam.kumar.cse@ulab.edu.bd
University of Liberal Arts Bangladesh (ULAB)

Supervisor: Dr. Nabeel Mohammed
nabeel.mohammed@ulab.edu.bd
University of Liberal Arts Bangladesh (ULAB)

Associate Supervisor: Dr. Sifat Momen
sifat.momen@ulab.edu.bd
University of Liberal Arts Bangladesh (ULAB)

Abstract

This thesis report represents Bangla handwriting improvement suggestion based on adversarial examples and also mark prediction. There is a project from ICT division where we are working on predict score of Bangla handwriting and also give suggestion how to improve that handwritten. A dataform consist of 84 Bangla character including number, compound character is used to take data. Then train those data on supervised model to predict mark. After predict mark add some noise into input image and train on autoencoder model which give an improved solution based on that input.

Improvement suggestion using adversarial examples

Declaration

I declare that this thesis report present our won work and it is not copied from any other report or paper. Also this report has not been submitted in any other university or institute. In the reference section, we mention some published and unpublished work of others which has been acknowledge. Some information derived from those work.

Gautam Kumar Shom
12 January, 2017

Acknowledgments

First of all thanks to my supervisor Dr. Nabeel Mohammed and Dr. Sifat Momen sir who gave me so much support to do this work and also thanks to all of my respected teachers. Also thanks to my friends Mithun Bishwas, Nadimozzaman Pappo, Rafiqul Islam who helped me a lot to accomplish this project. I am also thankful to my family who have supported and encouraged me. Also thanks to everyone of the R&D lab.

Gautam Kumar Shom
University of Liberal Arts Bangladesh(ULAB)
12 January, 2017

Chapter 1

Introduction

Studies, since the 1920's, consistently shown that there exists strong positive correlation between good handwriting and getting high scores in essay writing[8] . This has important implications for children in particular, as a series of poor grades at a young age can quite severely dent the confidence of a child, who otherwise could have been a high achieving student. Furthermore, recent studies have shown that teaching fluent handwriting actual remediates certain reading-related learning disabilities[16].

While there exists certain standards, which have been used in the English Language to score handwriting[13], creating such a standard for Bangla requires widespread agreement on "what are the components of good Bangla handwriting", and are better suited for education specialists. However, as IT practitioners we strongly believe that there are scopes to contribute to this problem with the aid of technology.

1.1 Goal

The main goal of this report is to give an improvement suggestion of Bangla handwriting. Improvement suggestion is a part of a project from the ICT division, Bangladesh where aim of this project is to create a mobile application which can judge Bangla handwriting quality of the Bangla isolated characters. We are making an artificial models which will predict mark, age, gender also give improvement suggestion of that handwriting and this prediction is based on human teacher evaluations. So there is a part which will give an improvement suggestion. That is the goal of this report.

1.2 Motivation

At the early stage we learn handwriting. There is room for technology to assist in teaching Bangla handwriting. But remote parts of a country couldn't get that advantages, though most of the people use smartphone.

There are many mobile applications exist[1] which allow a user to trace a character on top of a character map, there is no such application which actually gives the user a suitable score on her/his performance. Suitable score means how a teacher would have score the Bangla handwriting. Also there is no such work in Bangladesh where some model can can evaluate Bangla handwriting and also predict score based on those handwritten.

1.3 Contribution

In this report we try to explain improvement suggestion of Bangla handwritten by using adversarial examples. Machine learning models are not good at classifying adversarial examples. Adversarial image pose a difficult problem to all machine learning approaches. Adversarial examples are images that misclassify by our training algorithms. Adversarial examples are one of the important vulnerable point to every machine learning algorithms.

Linear behavior in high-dimensional spaces is the cause to occur adversarial examples[5]. We use fast gradient sign method to generate images that are given high prediction scores by Convolutional Neural Network(CNN).

Chapter 2

Background

2.1 Handwriting Evaluate

Evaluating handwriting is difficult. There are many criteria to evaluate handwriting.[4] Marlow established some criteria for teachers to evaluate handwriting of the students. He also proposed a rubric-based approach to evaluate more precisely[4]. Assessing a person's handwriting is based on the parameters given below:

- the progress of a student her/his previous efforts
- how well a student monitors her/his handwriting
- how much effort is the student putting in
- how much practice a student is doing in forming each individual character

2.2 Convolutional Neural Network(CNN)

Convolutional Neural Networks(CNN) is made of 2D neurons that have weights and biases. Neurons contains learnable weights and biases. CNN contains lots of neurons and each neuron receives some inputs then execute some operations. There is a good reason why we use Convolutional Neural Networks, because it's architecture is a good fit for spatial distribution to images. The convolutional neural network is also known as Shift Invariant or Space Invariant Artificial Neural Network (SIANN).

[3] In the year 2012, Alex Krizhevsky used neural net to win Imagenet Large Scale Visual Recognition Competition(ILSVRC) of that year. His model drop the classification error from 26% to 15%. From that time many companies including Facebook, Google, Instagram, Pinterest, Amazon are using neural nets as the core of their services.

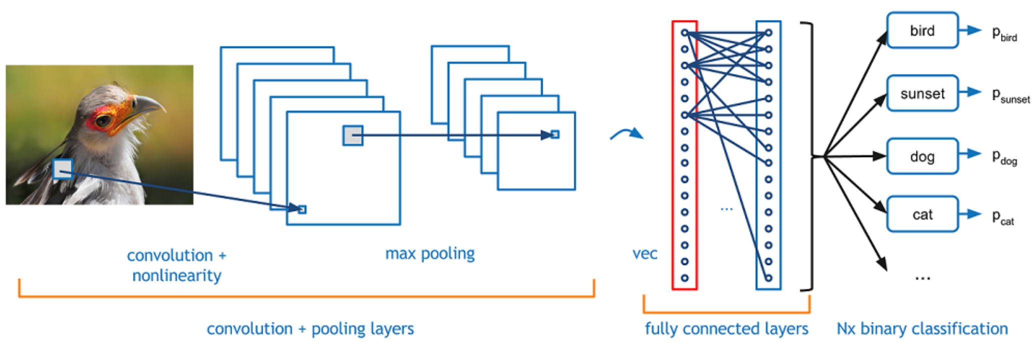

Figure 2.1: How Convolutional Neural Network works

Since birth if we see a thing then we learn features to classify it. We got this ability form nature and nurture.[12] Convolution Neural Network mainly uses three basic concepts.

1. Local receptive fields
2. Shared weights
3. Pooling

2.2.1 Local Receptive Fields

Figure 2.2 shows a 28x28 square grid of neurons.The neurons are similar to pixels whose values are same, then the grid can be thought of as an image.

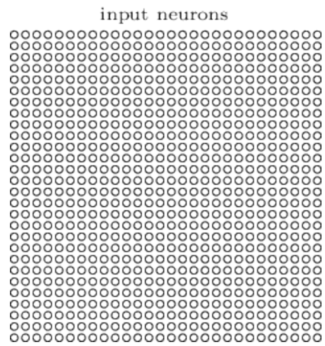

Figure 2.2: Input neurons

Every neuron in a convolutional layer represents the response of a filter applied to the previous layer. The job of this neuron is to pass this response through some non-linearity.The area of the previous layer that this filter is applied to, is called the receptive field of that neuron. As an example, if we have this 28x28 input image neurons and make a 5x5 region or 25 input pixels which will be also connected with that one hidden layer output.

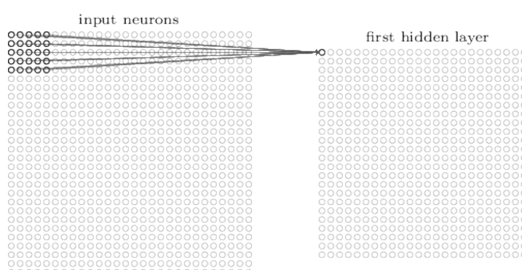

Figure 2.3: Hidden layer in local receptive field

So if we have 28x28 input image neurons and 5x5 local receptive field then there will 24x24 square neurons, which is also called first hidden layers. So that's how local receptive fields work.

2.2.2 Shared weights and bias

In the previous section we see that each hidden neuron layer contains a bias and 5x5 connected to it's local receptive field.This same bias and weights are used for each of 24x24 hidden neurons.The output of the (j,k)th neuron can be calculated by:

$$\sigma\left(b+\sum_{l=0}^{4}\sum_{m=0}^{4} w_{l,m}a_{j+l,k+m}\right). \tag{2.1}$$

Here,
σ = Neural activation function
b = Shared value for the bias
w = Shared weights
a = input activation at position j,k.

Shared weights and bias are also called kernel or filters. A convolutional layer contains many features. In the image in Figure 2.4 there are three features:

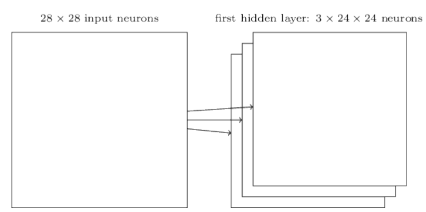

Figure 2.4: Hidden layer with feature map

On the first hidden layers, there are 3 feature maps and each layer contains 5x5 local receptive fields.

2.2.3 Pooling layers

Convolutional Neural Networks contains pooling layers. Usually this pooling layer come after convolutional layers. Pooling layers reduce information of convolutional layer. Pooling layers reduce the number of parameters by applying some kind of process(average pooling, L2-norm pooling etc.) on spatially local area of the feature map. Figure 2.5 is the image of a pooling layer:

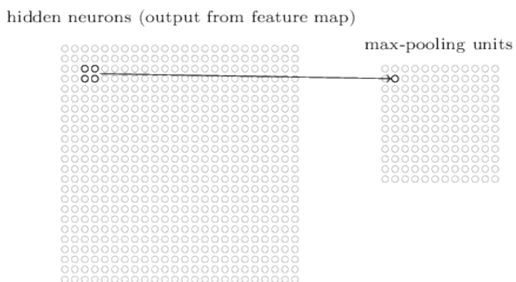

Figure 2.5: Max-pooling layer

On figure 2.5 we can see that a hidden layer which have 24x24 square neurons and after pooling that layer into 2x2, then it became 12x12 max-pooling layer. Pooling layer is also called max-pooling layer. A max-pooling layer consist of simplify version of convolution layer.

2.2.4 In Summary

Summarize of all this Convolutional Neural Network is there will be a local receptive fields of a input image neurons. So there will be a hidden layer of input layer by local receptive field. Then a layer will contains some shared weights and bias. The most important thing is it will contains some feature maps. Last thing is max-pooling layer which will contains simplify version of a convolutional neural network. After that layers will give an output of that image or predict a class. Here is an image of getting all together:

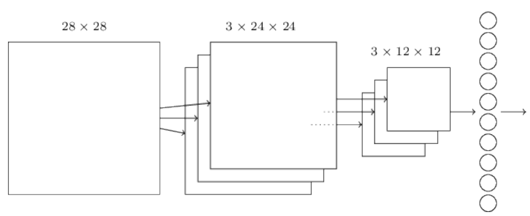

Figure 2.6: All together in CNN

2.3 Dropout

A neural network contains huge number of parameters. Those huge number of parameters are slow to use in a machine. But the main problem with this many parameters is the possibility of overfitting. To prevent this problem there is a technique called dropout[15]. The main idea of dropout is to randomly drop unit from the neuron network during training.

[15] The dropout approach randomly drops some neurons with it's connections. It works well even if dropout drop some neurons from layers.

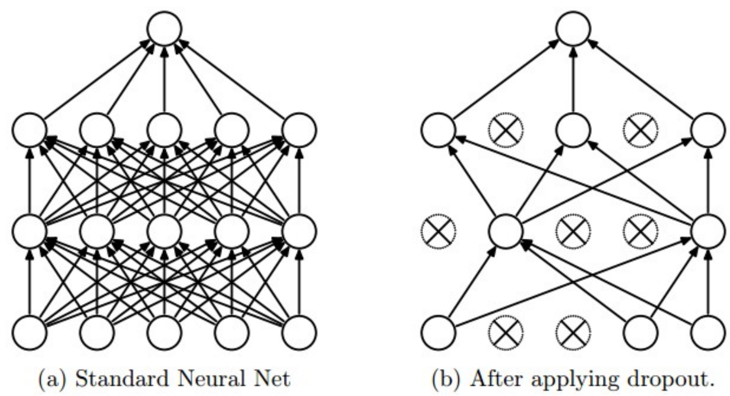

Figure 2.7: (a) A standard neural network and (b) Neural net after apply dropout

2.4 Adversarial examples

It is possible to force these networks to misclassify examples by adding imperceptible changes to images. Today's advanced object recognition algorithms are good as recognize an object as much as a human can.[6] Some explanation shows that because of extreme nonlinearity of deep neural networks can cause of adversarial examples. But it is not true. Adversarial examples can occur because of linear behavior in high-dimensional spaces [6].

Fooling images are examples that the model assigns to some class with high confidence even though they should not belong to any class. [5] Changing real examples slightly in order to make a classifier believe that adversarial examples belong to the wrong class with high confidence.

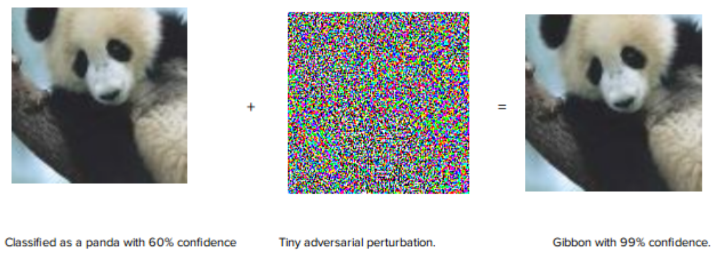

Figure 2.8: Adversarial images

In this example on the left most image we can see that a panda image(human eyes). A good model will classify that image as a panda with 60% confidence. With the original image we add a tiny adversarial perturbation image(middle image). And then the interesting part occur, which is the rightmost image. A human will see the rightmost image as a panda, the same as leftmost image. But the addition of the unseen noise forces the classifier to predict the class 'gibbon' for this image. The confidence of prediction on the rightmost image is about 99%.

So our part for our Bangla handwritten project is to predict mark of an image and also give suggestion of that by using adversarial examples. After mark predict of an image, then by using fast sign gradient method we generate adversarial examples so that it can add noise on that mark predicted image.

2.5 Problems of adversarial examples

This is an interesting topic to study but a serious security problem for any machine learning algorithms. There are many examples of adversarial examples like attacks in spam filtering, computer security, in network packets inject malware code, mislead signature detection, biometric recognition.

2.6 Autoencoder

[2] Autoencoder is a data compression algorithm where three functions occur by compression and decompression. Three functions are 1) data-specific, 2) lossy, and 3) learned automatically from examples. Figure 2.9 shows an example of autoencoder:

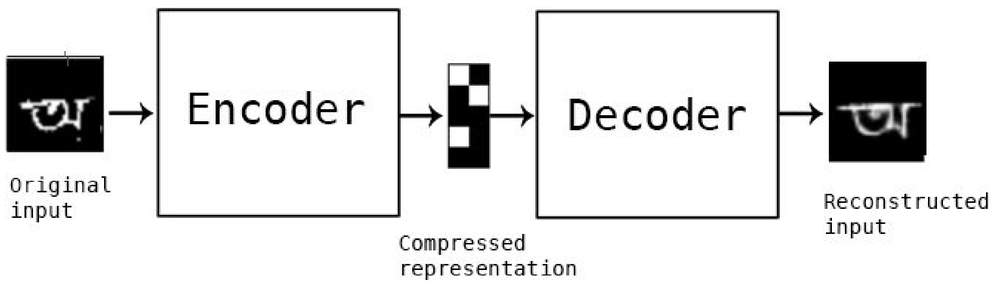

Figure 2.9: Autoencoder

[2] The term autoencoder means compression and decompression of an image in neural networks. After go through encoder phase an image will be at compressed representation level. Then if that compressed image go through decoder phase then we again got the reconstructed input image. The advantage here is, in the reconstructed image or output image is noise free. Here are two practical application of an autoencoder:

1. Data denoising

2. Dimensionality reduction for data visualization

Autoencoder also learns data projection which is most interesting part then others technique like PCA(Principal Component Analysis) or other basic techniques. So that's how a autoencoder work , it takes a image as input image then go through encoder, decoder phase and gives a reconstructed output image which is also noise free.

2.7 Fast sign gradient method

After mark prediction we use fast sign gradient method for generating adversarial image. Fast sign gradient is a method by which an adversarial image can be generated. We use gradient ascent for maximize the cost function in fast sign method. Here is the equation for fast sign gradient method[6]:

$$\eta = \epsilon sign\left(\nabla_x J\left(\Theta, x, y\right)\right) \tag{2.2}$$

Here,

$\epsilon = 0.07$ (We used in our model)
Θ = Output of a model
x = Input of a model
y = Improvement suggestion of an image
$J(\Theta, x, y)$ = Cost function to train gradient ascent

There are many solutions for adversarial examples but not every solution is efficient for classify rubbish class.For this work, the important aspect is that the adversarial image generation process converts one image of a certain class to an image of another class.[5] There are many regularization strategies like averaging across multiple models, averaging across multiple glimpses of an image, training with weight decay or noise, and classifying via inference in a generative model but all of them were unsuccessful to adversarial images. We used autoencoder based solution for our work.

2.8 Adadelta optimizer

Adadelta is learning rate method for gradient descent. Adadelta method dynamically adapts over time using only first order information and has minimal computational overhead beyond vanilla stochastic gradient descent[17].

The method requires no manual tuning of a learning rate and appears robust to noisy gradient information, different model architecture choices, various data modal ities and selection of hyperparameters[17].

Adadelta is[14] an extension of Adagrad that seeks to reduce its aggressive, monotonically decreasing learning rate. Instead of accumulating all past squared gradients, Adadelta restricts the window of accumulated past gradients to some fixed size w.

Chapter 3

Overall Approach

3.1 Final Model

Figure 3.1 shows the full model of image diagram we work on.

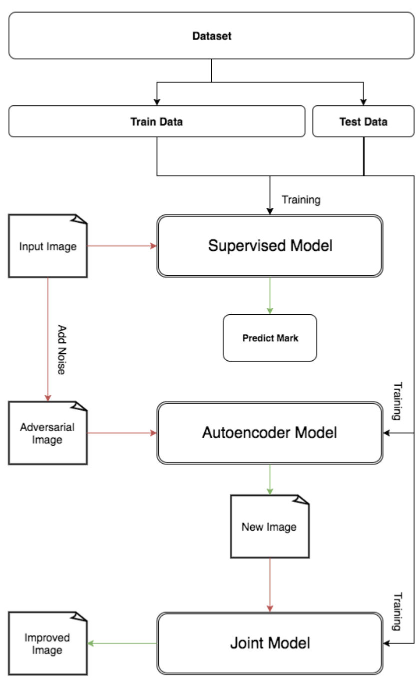

Figure 3.1: Overall Model

In Figure 3.1 diagram, we can see that there are several things occur. We have dataset, which divided into train data and test data. First of all we have an train data, then it go through supervised model. Then it will predict a mark of the test image. After predict mark it will also add some noise through fast gradient sign method. Then that input image will train on autoencoder model. We used autoecoder for denoise of that data. Then there will be another model which is joint model that predict the final image of that new image.

3.2 Details of Models

Figure 3.2 shows the description image of that three model :

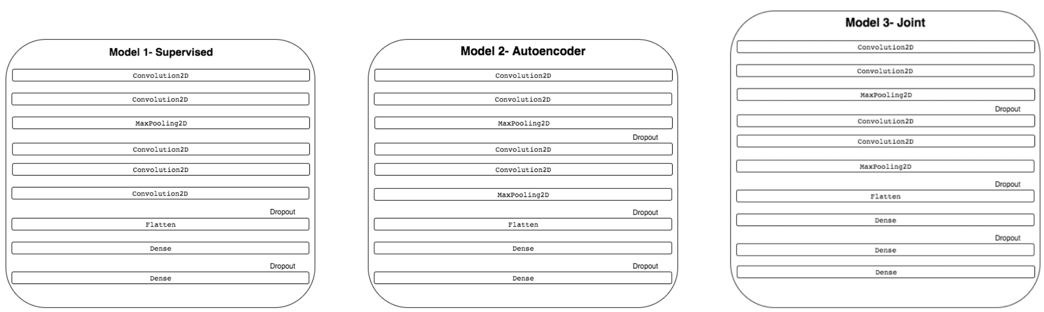

Figure 3.2: Left one is supervised, middle one is autoencoder and Right one is Joint model

3.2.1 Supervised Model

Supervised methods are methods that attempt to discover the relationship between input attributes (sometimes called independent variables) and a target attribute (sometimes referred to as a dependent variable)[10]. Our supervised model consist of 2 Convolution2D, 1 MaxPooling2D, again 3 Convolution2D, 1 dropout,1 flatten,1 dense,1 dropout, 1 dense layer. This model will predict score(Between 1 to 5) of an isolated image. This score will be given based on how human evaluate a handwritten.

3.2.2 Autoencoder Model

A autoencoder model is used to learn generic features, and as such is part of a representation learning system[11]. Our Autoencoder model consist of 2 Convolution2D, 1 MaxPooling2D, 1 dropout, 2 Convolution2D, 1 MaxPooling2D, 1 dropout, 1 flatten, 1 dense, 1 dropout, 1 dense layers.

3.2.3 Joint Model

Our final joint model consist of 2 Convolution2D, 1 MaxPooling2D, 1 dropout, 2 Convolution2D, 1 MaxPooling2D, 1 dropout, 1 flatten, 1 dense, 1 dropout, 2 dense layers.

3.3 Flatten Layer

In the convolutional neural networks, flatten layer is designed for fast feed-forward execution. Excess of parameters, especially weights of the convolutional filters in convolutional neural networks has been extensively stud- ied and different heuristics have been proposed to construct a low rank basis of the filters after training[9].

In this work, we train our flatten network after Convolution2D, MaxPooling2D and dropout layers. The flattened convolution pipelines provide around two times speed-up during feedforward pass compared to the baseline model due to the significant reduction of learning parameters[9]. Furthermore, our models does not require efforts in manual tuning or post processing once the model is trained.

3.4 Dense Layer

Dense layer which connects each layer to every other layer in a feed-forward fashion. Whereas traditional convolutional networks with L layers have L connections - one between each layer and its subsequent layer[7]. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. [7]DenseNets have several compelling advantages: they alleviate the vanishing-gradient

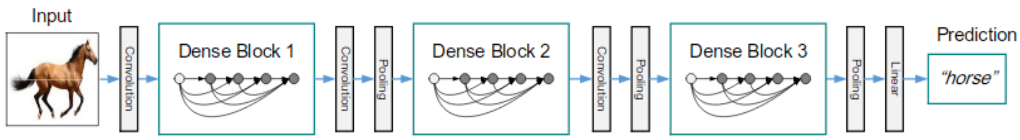

Figure 3.3: A deep DenseNet with three dense blocks

problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters.

Chapter 4

Dataset

4.1 Data form

Figure 4.1 shows an image of that form we take those data. There are eighty four(84) Bangla alphabet on those forms including vowels, numbers(0 to 9) and consonants. We also kept some information like age, district, male or female of the participants. There is reason why we choose those Bangla alphabet and numbers. Because those alphabets and numbers are the most basic and most usage alphabets and numbers anyone written in any place. Those data were taken from different age of people. There are so much variation in those data we collect.

Figure 4.1: Data Form

4.2 Collect data

For this project we need to collect data of Bangla handwritten. We made a form of Bangla handwritten and two thousand(2000) form written by students from different schools and universities. Here is the two image of collecting data:

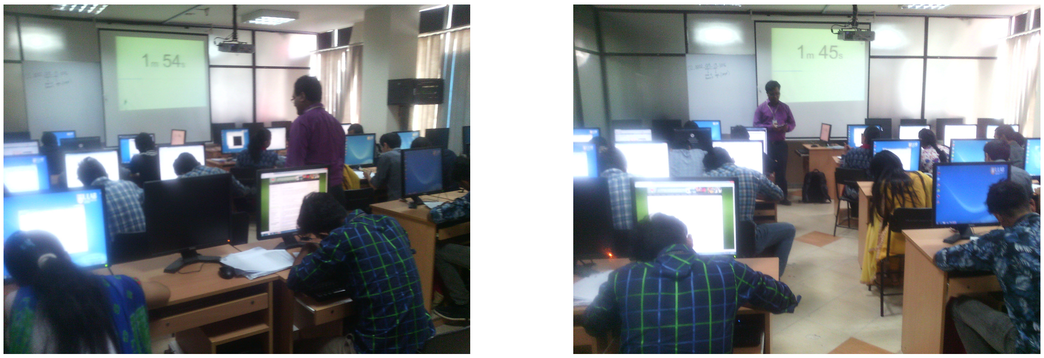

Figure 4.2: Collecting data

4.3 Evaluate data

After taking those data forms, it have to evaluate from a handwritten expert. Then handwritten expert mark all those forms. Mark was given from one(1) to five(5) for each form. This mark was given by Dr. Zareef sir who is a Bangla handwritten expert of 3 fingers academy. He scored handwritten based on some criteria like how clean is the handwritten, consistency, dimension of a character etc. Which is a standard way to score a handwriting.

4.4 Scanning and extracting data

We scanned all of those form into images for digitization. Then we extract of those form images into eighty four(84) different images. After extract we got about 166181 images of all Bangla alphabets.

4.5 Sample size

Sample size of our project was about 2000 forms, each form contains 84 Bangla alphabet. Which is about 2000 x 84 = 1,68,000 (one lakh sixty eight thousand) images. So our sample size is about 1,68,000 images. We made the biggest dataset in Bangladesh.

4.6 Train and test data

Since we are working with improvement suggestions so we had to make a .pkl.gz file to work with. So we made a .pkl.gz file of all 166181 images and load that .pkl.gz file to run my script. We resize those images into 32x32(height and weight) pixels. Then we gave Eighty percent(80%) of the data(132944 images) is for train data and twenty percent(20%) of the data(33237 images) is for test data. After make those train and test data separate then we made .pkl.gz file, to load on any script by those images.

Chapter 5

Result

Figure 5.1 shows the sample result of our model:

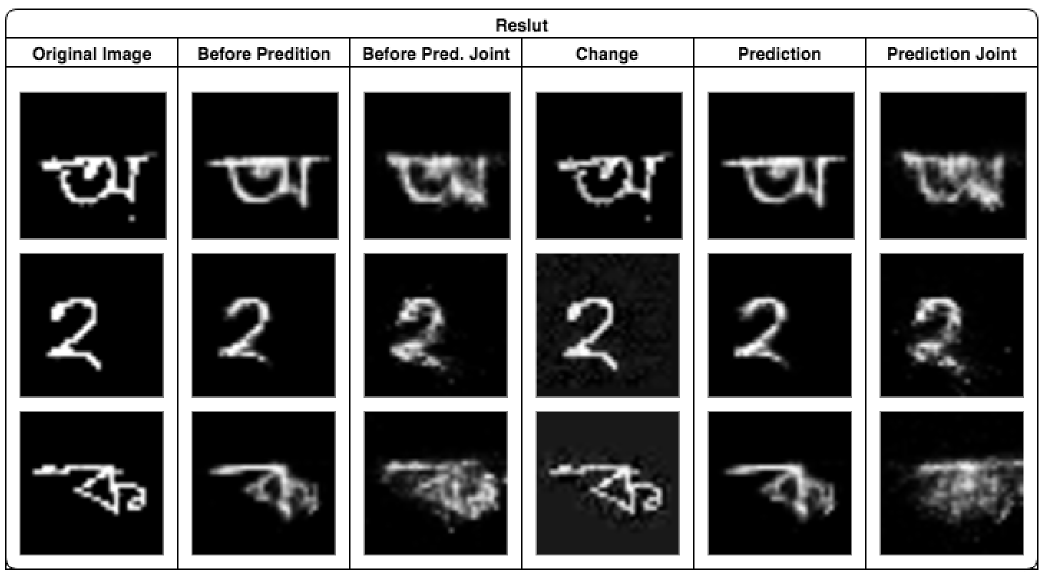

Figure 5.1: Result of our training models

In the first column those three are the original images. After change that image before add noise it will predict autoencoder prediction and joint prediction. Then it will change image. After change that image two model autoencoder and joint model will predict fifth and sixth column result. Result will be better if we train on more train and test data.

Chapter 6

Conclusion

Summary of this report is that how our neural network give suggestion a handwriting through adversarial images. We had to go through many things to do that. Adversarial examples is an important and also interesting fact to work in improvement suggestion.

First of all we had to predict mark of a handwriting based on evaluation of a handwritten expert gave. Then add some noise of that image so that input image can improve. But with improve it also add noise which only seen by machine not by human. So this is a great challenge we had to overcome. Then we use autoencoder for this solution of canceling noise. Autoencoder also did an great work it cancel data noise and also reduce dimension. After autoencoder model a joint model came to give an output.

Reference

[1] Deep learning adversarial examples – clarifying misconceptions.

[2] Francois Chollet. Building autoencoders in keras.

[3] Adit Deshpande. A Beginner's Guide To Understanding Convolutional Neural Networks, 2017.

[4] Marlow Ediger. Assessing handwriting achievement. *Reading Improvement*, 39(3):103, 2002.

[5] Ian Goodfellow. Deep learning adversarial examples – clarifying misconceptions.

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[7] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[8] David C Hughes, Brian Keeling, and Bryan F Tuck. Effects of achievement expectations and handwriting quality on scoring essays. *Journal of Educational Measurement*, 20(1):65–70, 1983.

[9] Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello. Flattened convolutional neural networks for feedforward acceleration. *arXiv preprint arXiv:1412.5474*, 2014.

[10] Oded Maimon and Lior Rokach. Introduction to supervised methods. In *Data Mining and Knowledge Discovery Handbook*, pages 149–164. Springer, 2005.

[11] David Meyer. Introduction to autoencoders. 2015.

[12] Michael Nielsen. Introducing convolutional networks.

[13] Joanne Phelps, Lynn Stempel, and Gail Speck. The children's handwriting scale: A new diagnostic tool. *The Journal of Educational Research*, 79(1):46–50, 1985.

[14] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[15] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[16] Rowe A Young, Robert V Rose, and Rand Nelson. Teaching fluent handwriting remediates many reading-related learning disabilities. *Creative Education*, 6(16):1752, 2015.

[17] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.