CSE 610/676 : Project 1.1
Handwriting matching using Deep Learning

Alizishaan Khatri (alizisha@buffalo.edu)
Fayaz Ahmed Vellore Farooque(fayazahm@buffalo.edu)

Abstract

This project is primarily concerned with the task of determining whether a pair of handwritten samples belong to the same writer or not. The dataset consists of a scanned images of handwritten samples of the word “and”. Another motivation behind this project is to make use of deep learning model to solve a real world problem. In this project we tried 2 different deep learning models and evaluated the model performance for various values of the hyper parameters in a bid to achieve maximum accuracy.

1.Introduction

The task of handwriting matching has its uses in the field of forensics where given a handwritten sample it can be matched against a database of numerous such samples and identify the most likely person who might be the author of the obtained sample. In this project we use deep learning models to solve the above problem. This is a significant advancement over the current state-of-the-art as it requires negligible human involvement ot work. The underlying objective of this project is to achieve results similar to [1] using minimal domain-specific engineering.

2. Dataset

The dataset for this problem consists of about 15 handwritten samples of the word “and” from each writer. We used such sets from over 1500 writers. The images were extracted from a longer handwritten document corresponding to the word “and”.

3. Data Processing

We standardized the size of the images provided for uniformity. Contrast adjustments and random rotations were also performed on the input images. The data was bifurcated into Training and Test Sets in the ratio 9:1.

4. System Architecture

4.1. Approach 1 : Autoencoders and Neural networks

This approach consists of the following components : a) Auto-encoder b) Neural network. Each image was fed into an auto-encoder. We condensed the original image into a vector by passing it through a two hidden layer auto-encoder using sigmoid activations. The weights of the lower layer have been used for identifying the image. We then stacked vectors from the same writer and different writers on top of each other along the long dimension. We also generated samples from these by interchanging the order of stacking of the images. Class labels indicating same / different writers were assigned to each image-pair. This data was then flattened into a single image and fed to a neural network working as binary-classifier. Softmax activations were used at the output layer to produce one-hot encoded data. We also experimented with both fixed and decaying learning rates and found the latter to produce better results.

Learning the Auto-encoder:
Training the auto-encoder network involved learning the auto encoded representation for each of the images in terms of 128 features. The first hidden layer consisted of 256 nodes and the second hidden layer with 128. These 2 layers constituted the encode layers. These 2 layers were followed by decode layers of 256 nodes followed by the layer representing the input itself consisting of 1200 nodes in our case. The auto encoder network was trained using backprop with RMS optimizer. About 20 training epochs were used to learn the auto encoder weights.

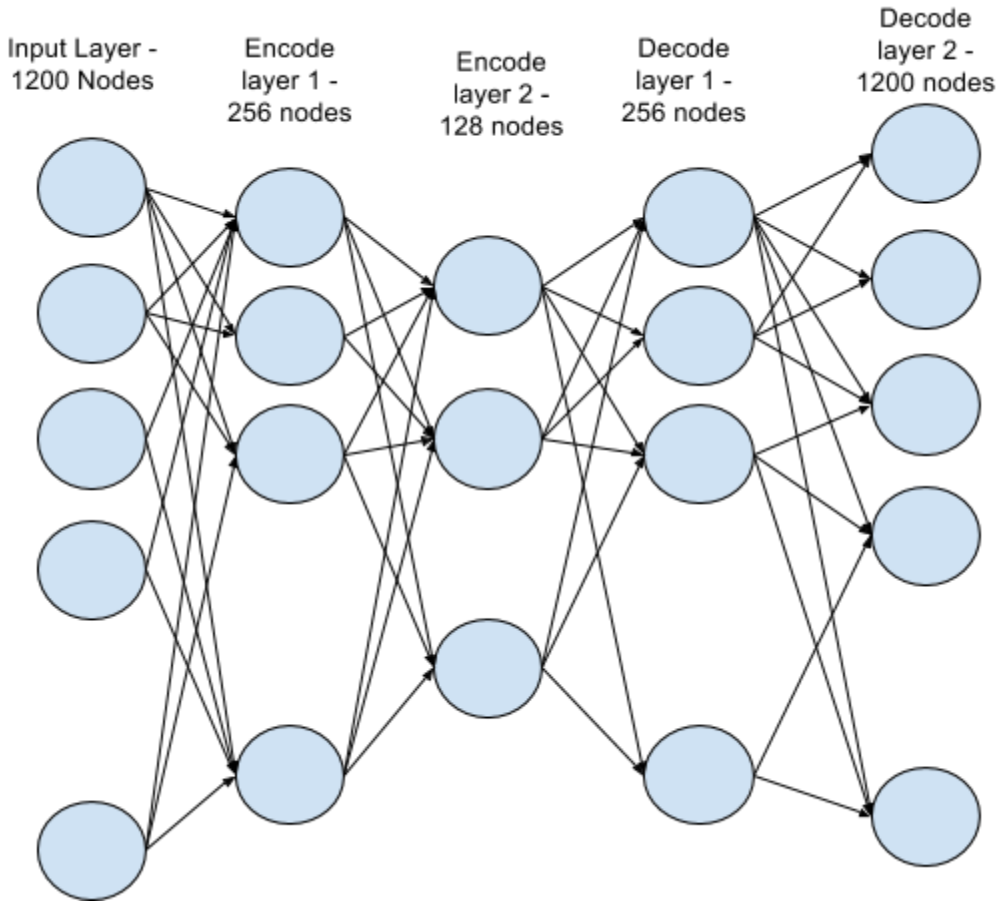

Fig 4.1.1 : Auto-encoder network architecture

Auto Encode network:

Once the auto encoder network was learnt, the auto encoded output at the 2nd layer consisting of 128 nodes representing 128 features were used instead of the actual images. The auto encoder network in its hidden layers learns the representations of the image using which the original image can be approximately reconstructed. Thus using the auto encoded output helps in dealing with lesser number of features as well as learns features from the data. The auto encoded output from 2 images is then fed to a single hidden layer, fully connected neural network and trained as a 2 class classifier to classify whether the images belong the 2 same writer or not. The training data was generated such that we had all the pairs of images from the same writer and an equal number of training samples consisting of one image from that writer and another randomly chosen handwritten sample from another writer to have a balanced dataset from both the classes representing match and nonmatch respectively.

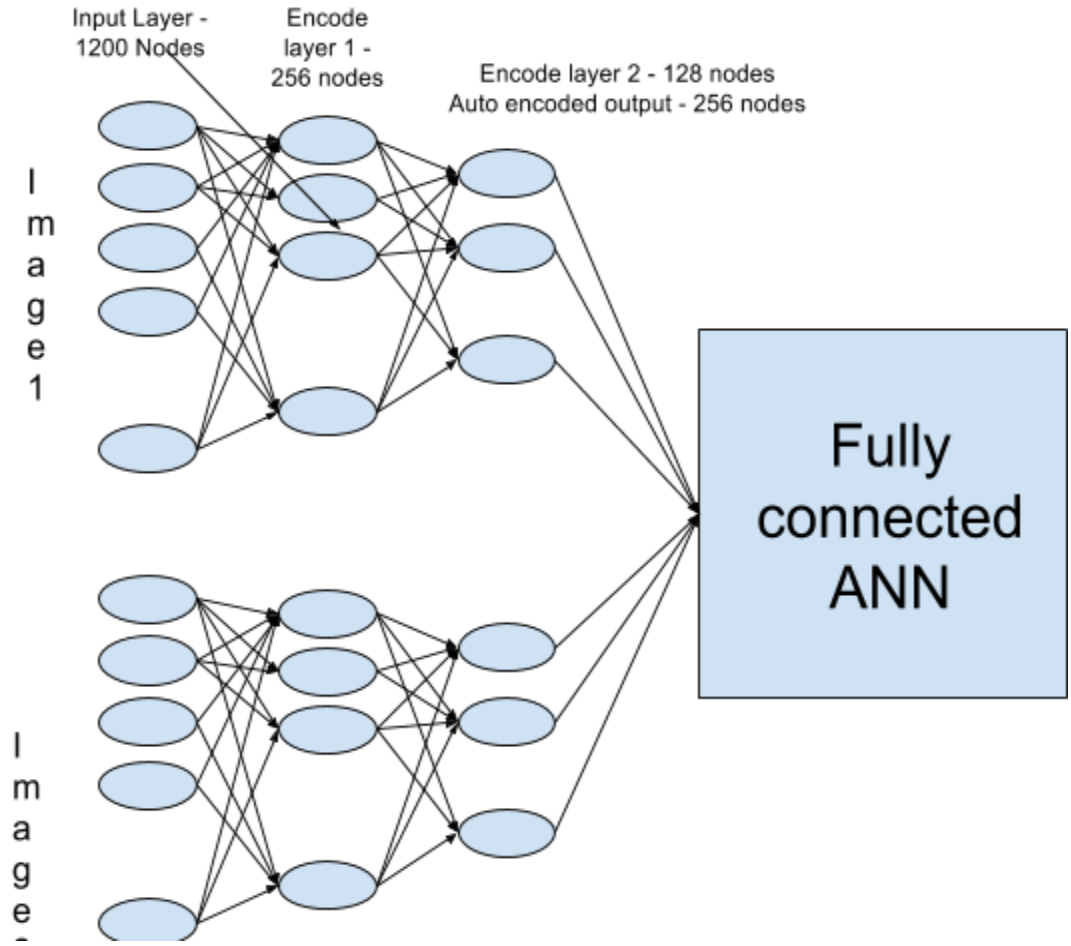

Fig 4.1.2 Connecting the auto-encoder output with different ANNs

4.2. Approach 2 : Convolutional Neural Networks

The second approach we used was that of Convolutional Neural Network(CNN). Our CNN consisted of 2 sets of convolutional and max-pooling layer followed by a fully-connected Neural network. The CNN used a 5x5 variable filter with 16 features in the first convolutional layer and 32 in the second. The max pooling layer used was 2x2. The fully connected ANN part used a 768 hidden node layer. We experimented with different filter sizes, increasing to 8x8 worsened the accuracy. The CNN was trained by varying the number of training epochs with 6 training epochs resulting in about 85% accuracy.

The training data was generated by creating a composite image by vertically stacking 2 images obtained from all possible combinations of sample images from the same writer and an equal number of samples for the samples from that writer composited with samples from other randomly chosen writers to balance out the dataset. The network uses relu activation at all convolutional and hidden layers and a softmax activation at the output layer. The training is done by using Adam optimizer with cross entropy as the cost function.

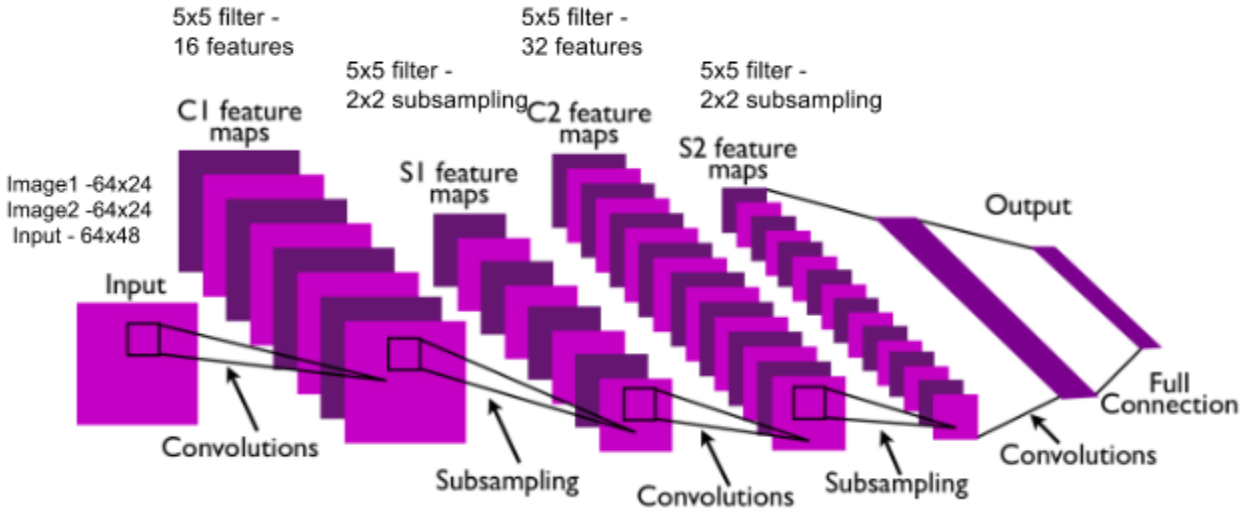

Fig 4.2.1 : The convolutional network used

Results

Approach 1 : Autoencoders and Neural networks

a) For 128 bits / image

Number of hidden layers	Activation functions for hidden layers	Nodes in hidden layers Accuracy	Accuracy
1	relu	32	73.12%
2	relu-tanh	64-32-32	71.1%
3	relu-tanh-linear	64-32-32	70.03%

b) For 256 bits/image

Number of Hidden Layers	Activation functions for hidden layers	Nodes in Hidden Layers	Accuracy
1	relu	128	61.3 %
2	relu-sigmoid	128-32	60.9%
3	relu -sigmoid-tanh	128-32-8	60.1%

Approach 2 : Convolutional Neural networks

Number of training epochs	Filter size	Activation	Nodes in Hidden Layers	Accuracy(%)
4	5x5	relu	16-32-768	82.9
6	5x5	relu	16-32-768	85
4	8x8	relu	16-32-768	69

Conclusion

- We have observed accuracies in the range of 60 - 85 % using various system architectures.
- The convolutional neural network with a 5 x 5 filter outperformed the other architectures with an accuracy of 85%. Increasing th the number of training epochs seemed to improve the system performance.

Future Scope

- Try to improve accuracy by experimenting with smaller filter sizes and different network configurations.
- Adapt model to be script-independent.
- Improve data cleaning to extract only text part from the image samples.

References

[1] Arora, H., Lee, S., Srihari, S.N. and Cha, S.H., 2002. Individuality of handwriting. *Journal of forensic science*, *47*(4), pp.1-17.