Project 2: Explainable AI
CSE 674: Advanced Machine Learning

Ved Harish Valsangkar
Person Number: 50290388
University at Buffalo
vedharis@buffalo.edu

Abstract

In this project, a task of differentiating between writers of two images was used to explore the performance of probabilistic graphical models (Task 2) and deep learning techniques (Task 3). A preliminary task (Task 1) of data annotation was completed before attempting this project. Finally, a hybrid model was developed which combined the benefits of both paradigms to improve upon their accuracy (Task 4).

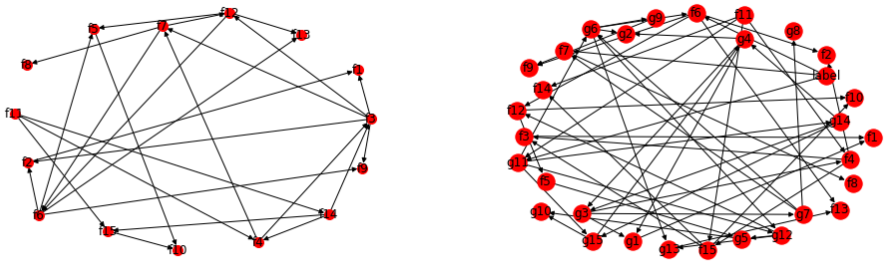
Task 2: Bayesian Modelling

Model Estimation

In order to estimate the model based on the available data, we can simply apply Hill Climb Search algorithm and obtain the relationship between the features. Then, treating the entire model as a node, another node of the same model can be attached with a common node for label.

However, a different approach was attempted. The 15 features for one of the images in the given pairing were concatenated with the features of the other image. A new column called "label" was added, whose value was 1 if the images were of the same writer and 0 if the images were from different writers. We can now apply Hill Climb Search algorithm and obtain the relationship between all 31 features.

The resultant model acquired 58 edges on successful completion of Hill Climb Search algorithm. The



(a) Model for 15 Features (b) Model for 31 Features

Figure 1: Models generated using Hill Climb Search algorithm.

Model Training and Prediction

The model so obtained is now trained by passing a fraction of the data available to the `BayesianModel.fit()` method. For testing our model, we now take the testing data and create a copy of the data without the label column. We shall pass this copy to the predictor and get the inferred outputs. These outputs shall be compared against the ground truth that we had. The total time required for model fitting was $\approx$ 15-20 minutes.

Observations

From the output we see that the accuracy of our method was around 55% - 60%. This is to be expected as the probabilistic models, while being very easy to understand, do not offer high accuracy as compared to Deep Learning methods, especially when inferring from data.

As an alternative, using logical reasoning and environmental observations, one can formulate an approximate model with assumed probability distributions. Such a model, while perhaps receiving a lower score in the scoring rubrics like the K2 scoring algorithm, serve better than models inferred from data.

Task 3: Deep Learning Approach

Siamese Network

The network selected for this task is the Siamese network. This network trains on 2 images simultaneously. Such a flow is desirable if the

The Siamese network works like an ordinary CNN except the weights are not updated at every image input. The pairs of images to be tested are passed to the network one after another. On feeding of each pair, the features extracted are subtracted or concatenated and passed through a classifier layer. The gradients accumulated from both images are now allowed to back-propagate and weights are updated.

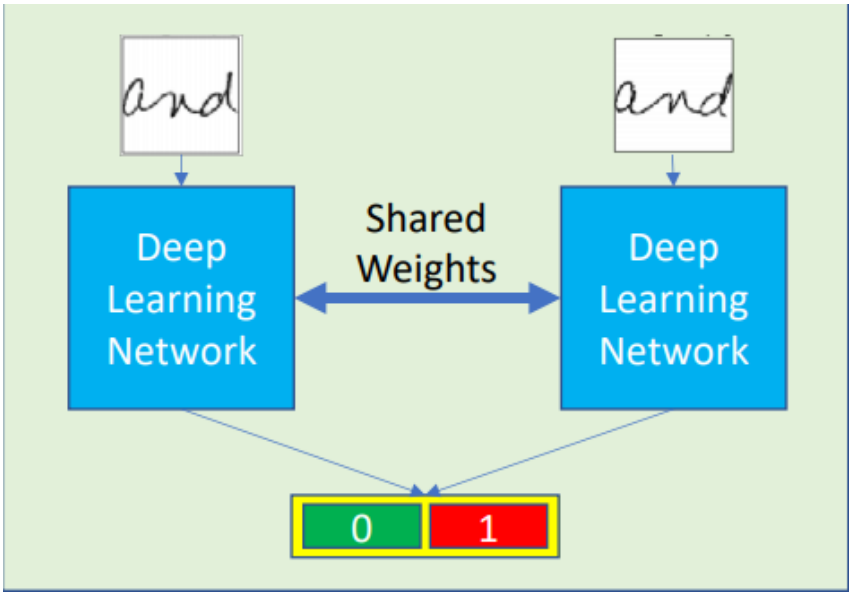


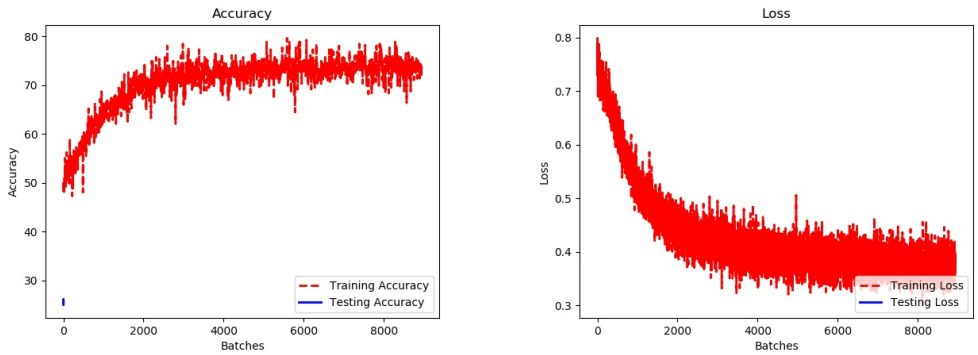
Figure 2: Block Diagram of Siamese Network (as given by the TAs).

The loss function selected for this network is the Binary Cross Entropy function, since there are only 2 nodes in the output stage corresponding to the one-hot vector of a binary decision.

Observations

Convolutional Neural Networks serve as a robust and reliable technique for identifying patterns in complex systems, with some implementations exceeding human capabilities. It is no surprise then that the accuracy of this method is higher that the Bayesian inferences.

Following observations on accuracy and loss of training data for the seen dataset were made.



(a) Training Accuracy for Seen Dataset

(b) Loss while Training for Seen Dataset

Figure 3: Accuracy and Loss for Siamese network on the Seen Dataset.

Task 4: Explainable AI

As an attempt to combine explainability of human modeled features and the accuracy of deep learning framework, we create a hybrid model consisting of an Auto-encoder which functions as a feature extractor on completion of training and 15 separate Multi-Layer Perceptron networks acting as classifiers.

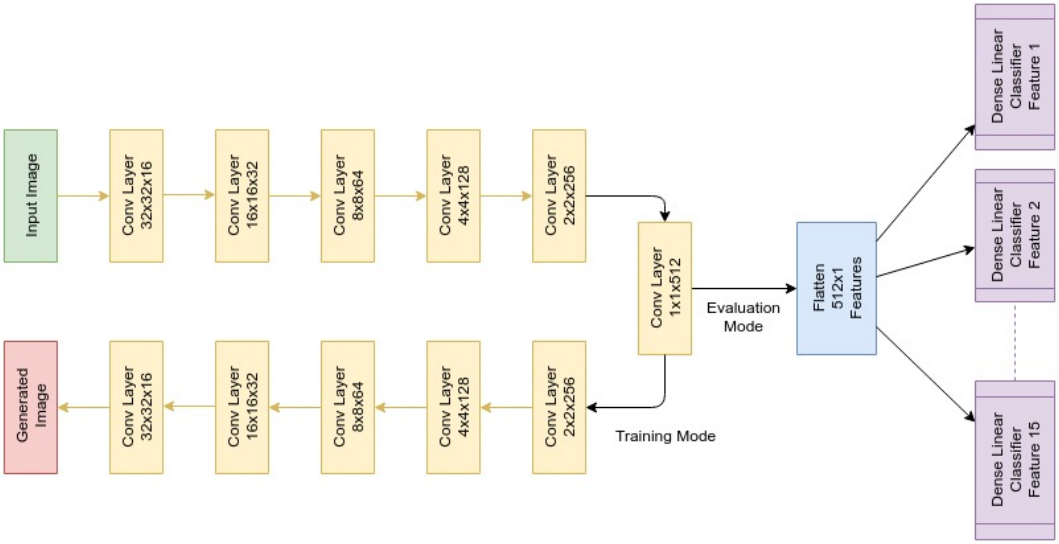


Figure 4: Block Diagram of Auto-encoder and Feature Classifier.

Auto-encoder

The auto-encoder is trained on images available in the training set. We can see the output of the final generation of images below. On successful training, we are able to reproduce the and images back using the feature layer ($1\times1\times512$). This layer is now ready to be used as a feature extractor.

(a) Seen Dataset (b) Unseen Dataset (c) Shuffled Dataset

Figure 5: Generated "AND" image outputs from Auto-encoder during training.

Feature Classifier

On successfully training a feature extractor, we now train 15 MLP networks, one for each feature. We keep these 15 separate so as to accurately map the relationship between the extracted latent features and the human recognized features without worrying about lower accuracy due to generalization as in case of a single shared classifier for all features.

The MLP consists of 3 layers, input layer of 512 nodes, a hidden layer of 256 nodes and the output layer with nodes corresponding to the number of states each feature can take.

Cosine Similarity

Cosine similarity is a score indicating how similar 2 vectors are. It essentially measures the angle between the vectors in the N-dimensional hyper-plane (512 dimension in this case). The score is bound between the range [0, 1]. The cosine similarity between 2 vectors A and B is given by the formula:

$$cosine\_similarity(A, B) = \frac{A \cdot B}{||A|| \times ||B||} = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Evaluation Strategies

Two evaluation strategies can be employed in order to arrive at a conclusion.

1. Finding the cosine similarity between the 512 feature vectors of the given images and thresholding it. Output is false if score the features dips below threshold.
2. Finding the cosine similarity between each of the Softmax output of the 15 features and thresholding them. Output is false if score of even one of the features dips below threshold.

Overall Cosine Similarity

This is the traditional deep learning method where the output vectors are compared and the answer is inferred from the similarity score. If the score is below threshold, the probability of the input pair originating from different writers is high and vice versa.

Feature-wise Cosine Similarity

In this method we compare the cosine similarity of Softmax layer output of each feature and see if any one of these scores is below threshold. If so, the feature is dissimilar, increasing the probability of the input pair originating from different writers.

Observations

On comparing the two strategies following observations were made. The accuracy values for Seen dataset were similar for both strategies. However, the traditional model fared comparatively poorer than the hybrid model.

	Overall Cosine Similarity	Feature-wise Cosine Similarity
Seen Data Set	77.23 %	76.85 %
Unseen Data Set	69.74 %	73.27 %
Shuffled Data Set	71.44 %	74.52 %

Table 1: Task 4 Accuracy Measurements.

Receiver Operating Characteristic Graphs for 3 Datasets

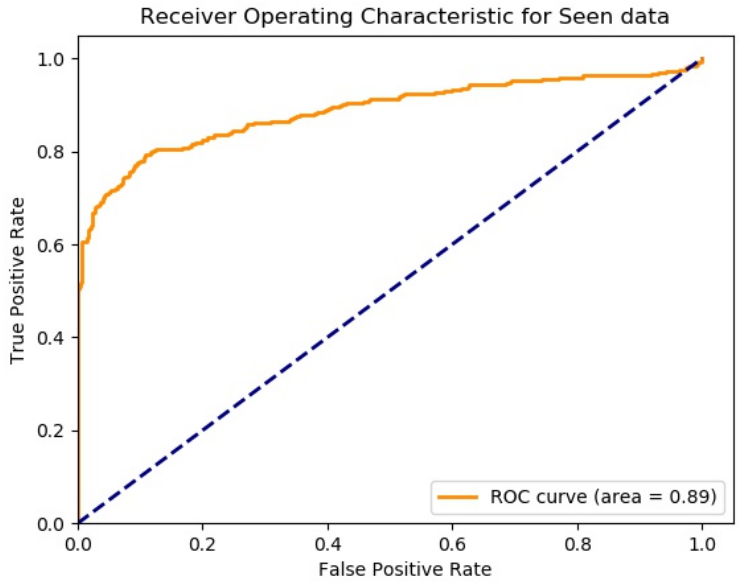


Figure 6: Seen output

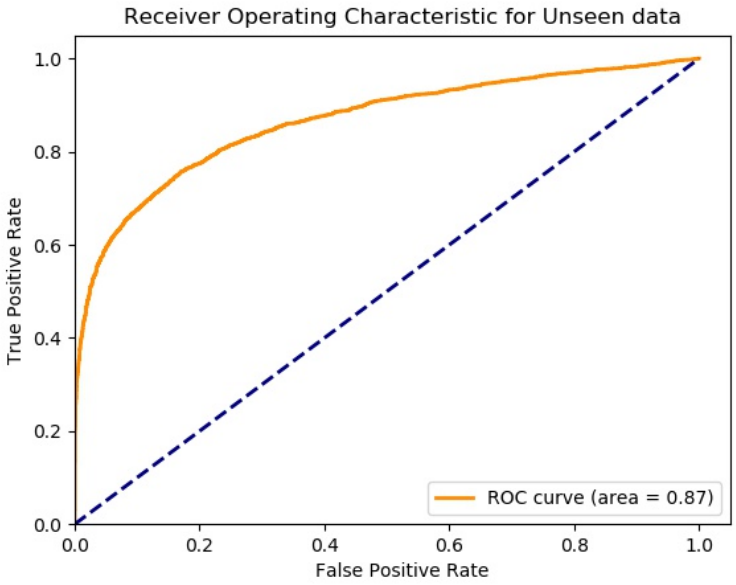


Figure 7: Unseen output

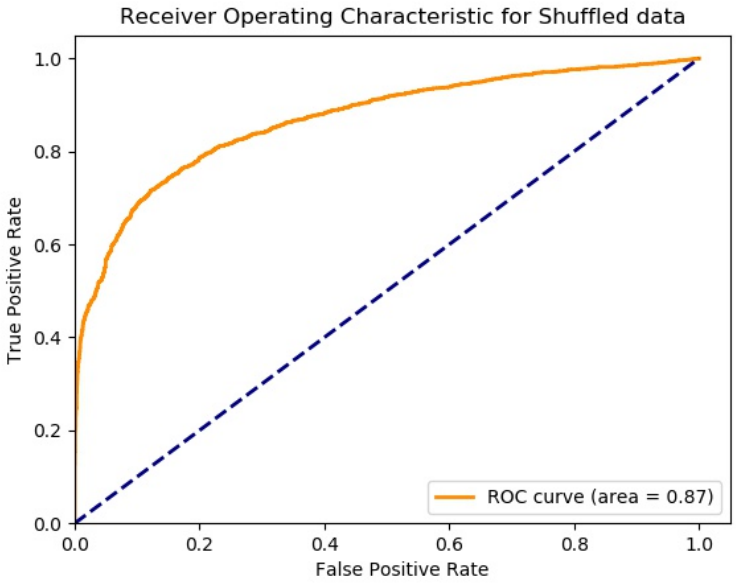


Figure 8: Shuffled output

Conclusion

Task 2

Bayesian network offer high reliability of features and mappings for the given data. It is easy to understand the reasons for any decisions made by the network whether right or wrong. Because we have all the connections and their probabilities with us.

Designing a model manually is best suited for this task. The CPDs involved in this are approximated by humans who have domain knowledge. These human approximated probabilities serve better in practice, when compared to features derived from data. However, due to sensitivity to noise in derived features and human bias induced in manual models, accuracy is considerably low for such models among all machine learning techniques.

Task 3

Deep learning models have proven to be accurate among the machine learning techniques. But the features acquired from the deep learning algorithms cannot be deciphered/understood by a human. These features consist of seemingly random float values.

While we know nothing about these features, we can infer the final answer and be fairly confident about it because we have trained the system to behave that way. Despite the higher accuracy, the inexplicable nature of these features raises ethical questions for sensitive cases like medical field. The question arises, how can we trust an answer if we do not know how the machine arrived at it? What justification can we give should the answer prove to be fatally wrong?

Task 4

Combining the approaches of human understandable features and deep learning techniques, we can create a hybrid system which extracts machine understandable features and maps them to human understandable parameters. Thus, we can have the advantages of both paradigms to some extent. Comparing the Softmax outputs of the classifier networks as a vector rather than a single value using cosine similarity, similar or slightly better accuracy can be achieved as compared to deep learning techniques while still deriving features from data which can be deciphered by humans. Thus we have successfully achieved fusion of two different techniques to improve accuracy.

References

The Cosine Similarity algorithm:
`https://neo4j.com/docs/graph-algorithms/current/algorithms/similarity-cosine/`

ROC example:
`https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html`