CSE 674: Advanced Machine Learning, Spring 2019
Project 2: Explainable AI

Sargur N. Srihari
University at Buffalo, The State University of New York
Buffalo, New York 14260
Contact: 716-645-6162 (O), srihari@buffalo.edu

December 28, 2018

1 Project Goals

This project is to develop a machine learning system that learns explainable features for a task domain while it is learning to answer a variety of queries in that domain. It involves combining deep learning and probabilistic graphical models.

We are given M samples of an input, e.g., a photograph of a person, scanned images of a handwritten word written by a known writer. Our goal is to work with three different sets of features: human determined features, deep learning features, explainable deep learning features. We use a discrimination task to compare the performance of each method.

1. **Human determined features**
   We assume that we have human-described variables (and the values that they can take) for an input image.

   Each description of an input consists of a set of D discrete random variables (called features): $\mathbf{x} = [x_1, x_2, .., x_D]$, where values taken by the variables are $x_i \in \{x_i^0, x_i^1, .., x_i^{d_i-1}\}$, $i = 1, ..D$ and $x_i^j = j, j = 0, 1, .., d_i - 1$.

2. **Features learnt using deep learning**
   Here the raw input (scanned images) are processed by a network to learn a representation, e.g., by means of a several convolutional network and pooling layers. The training could be performed by either by unsupervised learning (e.g., an autoencoder) or by supervised learning (e.g., learning whether a pair of samples is by the same writer).

3. **Explainable features learnt using deep learning**
   These are the representation learnt using deep learning is as similar as possible to the human explainable features described in the first part. Ideally the representation learnt by the deep learner is the same as the human explainable features. This can be learnt using supervised learning where the desired outputs are the explainable features.

1.1 Probabilistic graphical models

Construct probabilistic graphical models for (i) the human explainable features and (ii) the explainable representation that has been learnt. You can use a standard PGM construction algorithm. For (ii) you can use an RBM.

The goal of the PGMs is to be able to answer queries about the model. One example of a query is whether two samples came from the same person (face or writer). The answer provided should be explainable in terms of the dominant features that led to the conclusion. You need to come up with an evaluation metric as

1.2 Task

A typical task is to determine whether the two images originated from the same source (denoted as hypothesis h^0) or originated from different sources (hypothesis h^1). The decision is made by computing two probabilities: $p(\mathbf{X}_1, \mathbf{X}_2)/h^0$ and $p(\mathbf{X}_1, \mathbf{X}_2)/h^1$ and determining as to which probability is larger.

Construct a probability distribution when two samples come from the same writer. Similarly construct a distribution when they do not come from the same writer. The two distributions are used to make the decision whether or not two inputs come from the same person or not– by computing the likelihood ratio.

2 Project Details

This project should be completed and presented in mid-semester. The project is performed by a team of students consisting of either one, two or three students. The team should divide the work into identifiable components and then provide a combined report.

Implementations of the projects are to be done using Python. You can use libraries such as Tensorflow, Caffe, Keras, etc, for the implementation.

2.1 Project Proposal

You need to present a project proposal about four weeks into the semester. Prepare your proposal in the form of a presentation with four parts: (i) Title (with authors), Problem Domain description, and Data Sources (ii) Variables together with their types, and proposed distributions, (iii) evaluation methods.

2.2 Final Project Report

There are two deliverables: (i) project code and (ii) project report.

The project report should describe the problem domain, data set, algorithms used and performance (time complexity and accuracy). Use a format such as a conference paper for submission to NIPS or ICML. Include appropriate graphs and charts.

The due date for the deliverables will be in mid-semester.