STA 141C: Homework 4

- Homework due in Canvas: 06/10/2019 at 11:59PM. Please follow the instructions provided in Canvas about homeworks, carefully. **Please be mindful of the different deadlines you might have at the end of the quarter and schedule to work on the homework accordingly.**

Q1: Image Classification Using Deep Features (15 Points)

In the housing price prediction problem in a previous assignment , we tried some manual feature engineering and higher-order features constructed manually. In this problem, we will play with the most powerful *automatic* feature engineering tool in the machine learning field – deep neural networks. As we saw in the class, deep networks learn *layer-wise representation of the input* and is particularly useful for computer vision problems such as image classification. For example, the following deep neural network model can achieve greater than 95% accuracy on MNIST classification.

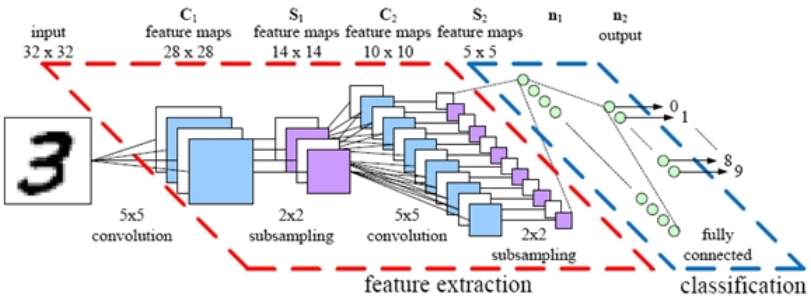

Figure 1: Deep neural network is mostly about feature engineering.

Deep neural nets are useful but painfully slow to train. We have trained a deep neural network (CNN; similar to the above architecture) on a large set of 32×32 RGB images for you. With a trained neural network, we can extract features from any new image of the same dimensions. From a different set of 32×32 RGB images, we have extracted deep features using the trained deep neural network and stored them in `DeepFeature.RData`. Note that this data is in RData format and hence use `pyreadr` to import this dataset in python. Load this RData file and you will see two matrices `deep.features`, `image.array` and a vector `label`. Every row of the matrices represent an image sample. The vector `label` has only two categories "cat" or "dog" with the i-th component representing the true label of the i-th image.

Load and split the data randomly into 70% training data and 30% test data. Use the `image.array` and `deep.features` as inputs respectively to classify the images into "cat" and "dog" using the following different approaches in PyTorch.

1. Use logistic regression to classify the raw data (`image.array`) and CNN representation (`deep.features`) and report the accuracy on test set.
2. Use a 2-layer neural network (with ReLU activation function) to classify the raw data (`image.array`) and CNN representation (`deep.features`) and report the accuracy on test set.

Q2: Generative Adversarial Networks (10 Points)

In this question, you will be required to read-up certain sections from the following textbook available online at `http://www.deeplearningbook.org` and `https://arxiv.org/pdf/1701.00160.pdf` and answer the following questions **in your own words**.

During the initial part of the course, we introduced the maximum likelihood estimation (MLE) technique. To recall, MLE is a way to estimate the parameters of any *parametric density*. MLE is sometime referred to as *explicit generative modeling* technique as we have to state the specific parametric density (for example, Gaussian) that we are using to model the given data. Often times in practice, it becomes hard to pick the right density that fits/models the data at hand.

A recent proposal to overcome this issue is the so-called *Generative Adversarial Networks* (GAN). Here a generative model is developed for the dataset at hand, implicitly. That is, no specific parametric density is assumed. This technique is called as *implicit generative modeling* in the literature. In this homework, you are required to write a 1-page report on GANs. Specifically, answer the following:

1. Read section 3.13 from `http://www.deeplearningbook.org/contents/prob.html`. Then define and interpret Kullback-Leibler (KL) divergence in your own words.
2. Read section 5.5 from `http://www.deeplearningbook.org/contents/ml.html`. What is the relationship between MLE and KL divergence ? Write in your own words.
3. Collectively summarize the issues with explicit density models from section 2.5 in `https://arxiv.org/pdf/1701.00160.pdf`).
4. Read section 3.1 and 3.2 on GAN. Explain what is the main idea behind GANs in your own words. Figure 12 is helpful for this.