CS6220 Unsupervised Data Mining

HW4 Tensor Flow : Classification, Autoencoders, Word Embedding, Image Features, LSTM

Make sure you check the syllabus for the due date. Please use the notations adopted in class, even if the problem is stated in the book using a different notation.

We are not looking for very long answers (if you find yourself writing more than one or two pages of typed text per problem, you are probably on the wrong track). Try to be concise; also keep in mind that good ideas and explanations matter more than exact details.

Submit all code files Dropbox (create folder HW1 or similar name). Results can be pdf or txt files, including plots/tabels if any.

"Paper" exercises: submit using Dropbox as pdf, either typed or scanned handwritten.

DATATSET **: SpamBase: emails (54-feature vectors) classified as spam/nospam**

DATATSET **: 20 NewsGroups : news articles**

DATATSET **: MNIST : 28x28 digit B/W images**

DATATSET **: FASHION : 28x28 B/W images**

https://en.wikipedia.org/wiki/MNIST_database
http://yann.lecun.com/exdb/mnist/
https://www.kaggle.com/zalando-research/fashionmnist

PROBLEM 1: Setup Tensor Flow, run few demos

The complete instructions for installing tensorflow can be found at: https://www.tensorflow.org/install/
Following is an example of installing CPU version tensorflow on Mac using virtualenv and Python2.7
(Why virtualenv? It's a good habit to isolate Python environment to prevent potential package conflicts. Virtualenv is a very light weighted tool that does the job.)

- Install virtualenv $ sudo easy_install pip
$ pip install --upgrade virtualenv
$ virtualenv --system-site-packages ~/tensorflow
$ source ~/tensorflow/bin/activate

- Install CPU version tensorflow (current version: 1.5.0)
(tensorflow)$ easy_install -U pip
(tensorflow)$ pip install --upgrade tensorflow

- Test if tensorflow is installed properly
(tensorflow)$ python
# Following is testing code
import tensorflow as tf
print(tf.__version__)

```
hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))
```

- Install jupyter notebook
(tensorflow)$ pip install jupyter

Till now, all the development environment is set up properly.

PROBLEM 2 : NNet supervised classification

A) For MNIST dataset, run a TF in supervised mode (train/test) and report results
B) TF classification for 20NG
C) Extra Credit. Run TF classification for MNIST using an Nvidia GPU

PROBLEM 3 : Autoencoders

For each one of the datasets MNIST, 20NG, SPAMBASE, FASHION, run TF as an autoencoder with a desired hidden layer size (try K=5,10, 20, 100, 200- what is the smaleest K that works?). Verify the obtained reencoding of data (the new feature representation) in several ways:

- repeat a classification train/test task , or a clustering taks
- examine the new pairwise distances dist(i,j) agains the old distances obtained with original features (sample 100 pairs of related words)
- examine the top 20 neighbours (by new distance) set overlap with old neighbours, per datapoint
- for images, rebuild the image form output layer and draw to look at it

PROBLEM 4 : Word Vectors

On 20NG, run word-verctors embedding into 300 dimensions using a Tensor Flow setup. Evaluate in two ways:

- given a word (from TA live during the demo), output the most similar 20 words based on embedding distance of your choice like cosine, euclidian, etc. Compare the 20 most similar words with the top 20 words by distance on Google word embeddings (word2vec embeddings)
- use a visulaizer that loads your embedding, projects it in 3 dimmensions and displays the words, for example TF projector

PROBLEM 5 EXTRA CREDIT: Image Feature Extraction

Run a Convolutional Neural Network in Tensor Flow to extract image features. In practice the network usually does both the feature extraction and the supervised task (classification) in one pipeline.

PROBLEM 6 EXTRA CREDIT: LSTM for text

Run a Recurrent Neural Network /LSTM in Tensor Flow to model word dependecies/order in text. Can be use for translation, next-word prediction, event detection etc.
LSTM article