COGS 260: Assignment 2

Instructions

1. Due on **May 4th, 11:59 PM**. To be uploaded on TritonEd.
2. You are required to use MNIST dataset for all the questions in this assignment [3].
3. Reference materials and some useful codes can be found at the end of the document.
4. Write a report including: a) abstract, b) method, c) experiment, d) discussion, and e) references. You can follow leading conferences like CVPR (`http://cvpr2016.thecvf.com/submission/main_conference/author_guidelines`), NIPS (`https://papers.nips.cc/`), or ICML (`http://icml.cc/2016/?page_id=151`).
5. Copy and paste your code as appendix section to your report.
6. You are supposed to provide brief quantitative details and analysis of experiments whenever asked to report the details of the experiment. Please DO NOT write long paragraphs. You are encouraged to use plots and images to explain your results.

In this assignment you are going to compare the performance of different methods of image recognition on MNIST dataset. To begin with divide the dataset into training and test set and use the same training and test set for all parts of the assignment. You may (cleverly) select a subset of the dataset to overcome computational challenges (if any).

1 1-Nearest Neighbor

Using the raw pixel values (28 × 28) as feature vector, use 1-Nearest Neighbor to train a model and report its performance on the test set and the confusion matrix. Also report other details of the experiments like the distance functions used, prototype selection (if you have used), etc.

2 Support Vector Machines

Again using the same feature vector, use SVM to train a model and report its performance on the test set and the confusion matrix. Report other relevant details of the experiments.

3 Spatial Pyramid Matching

Use traditional visual recognition pipeline (handcrafted SIFT local descriptor, bag-of-words codebook building and spatial pyramid matching [1], and classifier training) and report the performance on the test set and the confusion matrix. Report the details of the experiments. A link to the matlab code can be found under references [2]. SPM pipeline could be very inefficient compared to convolutional neural networks. Note that the implementation (Example.m) has several hyper-parameters *dictionarySize, gridSpacing, patchSize, numTextonImages, and pyramidLevels*, profiling and optimizing those hyper-parameters so you can finish the experiments in reasonable amount of time. Report your configuration details.

4 Convolutional Neural Network

For this question, choose a deep learning library (Tensorflow or Torch, etc) [5] that you can prototype your network structures easily and efficiently. MNIST is a standard tutorial dataset for almost all deep learning libraries [6][7], try to take a look and use the off-the-shelf data preprocessing pipelines instead of dealing with that yourself.

Build a convolutional neural network, train it and plot the training and test loss (not error) with the number of iterations. Also report the network architecture used and the classification accuracy on the test set for the experiment. You may play with different network architectures (LeNet, AlexNet, VGGNet, ResNet etc.), different hyper-parameters, and different optimizers like Stochastic Gradient Descent, Adaptive Gradient, RMSprop and/or Nesterov's Accelerated Gradient etc.

5 Extra Credit: Deep Belief Nets

This section is optional but will fetch you extra credits if you do it. Unlike the above methods which are discriminative models, DBNs (Deep Belief Nets) are generative models. The code for training a DBN is available in here (`http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html`) but you are free to use codes from other sources or write your own code. Train a DBN and project the activations of the last layer in two-dimensional space using t-SNE or Multidimensonal Scaling (use matlab/python library for t-SNE and MDS) and clearly label different digits in different colors. You should observe data points with the same class label clustered together. Report your results and other details of experiments like network architecture, training procedure, etc.

References

[1] Spatial Pyramid Matching, `http://slazebni.cs.illinois.edu/slides/ima_poster.pdf`.

[2] Spatial Pyramid Matching, Matlab Code `http://slazebni.cs.illinois.edu/research/SpatialPyramid.zip`.

[3] The MNIST Database of handwritten digits, `http://yann.lecun.com/exdb/mnist/`

[4] Deep Belief Networks Code, `http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html`.

[5] Deep Learning Software Links, `http://deeplearning.net/software_links/`

[6] Tensorflow MNIST Tutorial, `https://www.tensorflow.org/get_started/mnist/pros`

[7] Torch MNIST Demo, `https://github.com/torch/demos/tree/master/train-a-digit-classifier`