COGS 260: Assignment 3

Instructions

1. Due on **May 18, 11:59 PM**. To be uploaded on TritonEd.
2. Reference materials and some useful codes can be found at the end of the document.
3. 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`).
4. Copy and paste your code as appendix section to your report.
5. 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 work on three generations of neural network: Perceptron learning, Feed-forward Networks and Convolutional Networks.

1 Perceptron Learning

The data is in hw3_iris.zip In this problem, we will attempt to make a flower classifier. We shall use a modified version of the popular UCI Iris dataset provided in the file within the iris folder. The first 70 rows of the data are to be used as training data and the next 30 are to be used as test data. The file describes a data set with the following characteristics:

- Number of Attributes: 4 numeric attributes and the class label.
- Attribute Information: sepal length in cm, sepal width in cm, petal length in cm, petal width in cm
- The last column is the class label (Iris-setosa, Iris-versicolor)

Your tasks are:

1.(a)

Using the MATLAB/Python function scatter(X,Y), plot each of the six 2 dimensional feature spaces (i.e. sepal length vs. sepal width, sepal width vs. petal width, etc.) using a different symbol for each class. Put these graphs in your report. Are the classes linearly separable in each of the feature spaces?

1.(b)

Train a perceptron to classify the data using all the features. Report the learning rate used and no. of iterations for convergence. Test the data on the test set and report the classification accuracy.

1.(c)

Z-score each attribute (feature) of data. Make sure that you are not using test set in any way for computing the parameters for Z-scoring, i.e. use only the training set parameters to normalize both training and test set. Now repeat part 1.(b) on the transformed data. What did you observe?

2 Feed Forward Neural Network

In this problem, you are going to train a feed forward neural network to classify MNIST dataset [2]. Represent each image as a 784-dimensional feature vector. Randomly split the dataset into training and test sets with sizes: 50000 and 10000 respectively. Your tasks are:

2.(a)

Train a feed forward network with 1 hidden layer and write the weight update rules in vector notation (matrix notation). Report the training and test error with number of iterations. Also report the network architecture and the hyperparameters (learning rate) used.

2.(b)

Repeat the above part for feed forward network with 2 hidden layers. Comment on the performance.

2.(c)

Retrain the network in part 2.(a) with regularization and momentum and report the results. Did you observe any change in the performance?

3 Convolutional Neural Network

For this question you should use tools like Caffe, Theano, Tensorflow, Torch, etc [6].

In Assignment 2, you have played with basic convnets (and probably some techniques to accelerate the learning). In this problem, you are going to start from where you left in the previous assignment. You have to train a Convolutional Neural Network on CIFAR-10 dataset [1] with at least 2 convolutional, 2 pooling and one fully connected layer. You have to compare the following techniques to speed-up the learning. For each of the methods, comment on the iterations for convergence, accuracy on the test set, plot of training and test loss with respect to the iterations. Also, report the network architecture. Please refer to [5] for more details about the following methods.

1. Stochastic Gradient Descent [4]
2. Batch Normalization
3. Replace the fully connected layer by global average pooling layer
4. Adaptive Gradient (Extra credit) [3]
5. Nesterovs Accelerated Gradient (Extra Credit) [4]
6. RMSprop (Extra Credit) [3]

References

[1] The CIFAR-10 Dataset, `https://www.cs.toronto.edu/~kriz/cifar.html`

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

[3] AdaGrad and RMSprop, `http://cs231n.github.io/neural-networks-3/#ada`

[4] Stochastic Gradient Descent and Momentum, `http://cs231n.github.io/neural-networks-3/#sgd`

[5] Neural Networks, Andrej Karpathy, http://cs231n.github.io/neural-networks-3/

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