

CS 5984: Deep Learning

Homework 4

Due Date: December 7th, 2017 (11:59PM)

Total: 100 Points

Note: All implementations are required to be accomplished and submitted using the *Jupyter notebook*. Show your results and necessary comments in the notebook. No handwritten homework is accepted.

Problem 1: Loss function derivation

(10 points)

Show the derivation for computing the derivative of the loss function for an autoencoder for binary inputs when using sigmoid activation function in the output layer. (NOTE: You will need to derive the Equation given in slide 6 of Lecture 8).

Problem 2: Autoencoder

(50 points)

You will need to use the Tensorflow package for this problem. Pick the first 100 samples for each digit (0-9) from the MNIST data and construct the training dataset for your problem. Using the 1000 images in your training data, build a standard autoencoder (with only one hidden layer) containing N hidden units. You are free to choose the activation functions.

- Varying the values of N (2,5,10), plot the reconstruction error.
- Pick one of the images and compute the precision and recall for its nearest 50 images in the latent space. Then, report the average values of precision and recall for all the images corresponding to each digit (vary the values of N=2,5,10).
- Compare these results with the results obtained using PCA method. Use the same N values (latent space dimensions in PCA) for both methods and plot the comparison.

Problem 3: Word2Vec

(40 points)

In this assignment, you will use word2vec embeddings to find the nearest words for a given word. The pretrained word embeddings file (vectors.txt) is already given, so you do not need to run word2vec model. The file contains word2vec embeddings for 400K words, and the dimension of each vector is 50. Each line contains the word and its corresponding vector. The first word in each line is the word, followed by 50 numbers, where each number is a dimension of the vector.

Q-1: Semantics

Use the pre-trained embeddings file to compute the 20 most similar words using cosine similarity for the following words:

- life
- market

- c. stanford
- d. trump
- e. public

Show your work.

Q-2: Visualization

- a. Create a t-sne visualization, displaying all the words in the file.
- b. Use t-sne visualization to display the nearest 20 words for a given word. Create a separate visualization of all the 5 words given in Q-1, where each visualization displays the nearest 20 words for a word.

Submit screenshots, along with the code to generate the visualization.