MGTA 415 Homework 3

General Instructions and Tips

In this assignment, we will work on phrase mining and image recognition. Our first goal is to get you familiar with **AutoPhrase**, a data-driven phrase extraction framework. You will study the extracted phrases from the DBLP corpus using AutoPhrase and apply Word2Vec on the segmented corpus to capture the semantic meaning of phrases. Our second goal is to experience a simple CNN model for classifying images.

For the simplicity of grading, please prepare a Jupyter-Notebook for the HW answers using a combination of code and markdown cells. Please **generate the PDF/HTML file with necessary intermediate results, comments, explanations/analyses for results, and answers to open-ended questions** from the Jupyter-Notebook, and submit this PDF/HTML file to Canvas, so the TA will have sufficient information to evaluate your work. Please submit your files directly onto Canvas, **without zipping/compressing**, if possible. These can be done in the same notebook with either markdown or code cells. Please submit your solution by the beginning of the Week 10 Lecture, which is 11AM **Pacific Time on March 15th**. Submissions should be made on **Canvas**.

Problem 1: Phrase Mining experiments (20 points)

AutoPhrase is a data-driven phrase extraction framework. You can find the source code at here. You DO NOT need to run the code yourself. We provide an example of extracted phrases using AutoPhrase. The example is run on the **DBLP** corpus, a web text corpus consisting of a computer science bibliography.

In the **data** zip file, there are three text files listing phrases extracted from DBLP.
AutoPhrase.txt: the unified ranked list for both single-word phrases and multi-word phrases.
AutoPhrase_single-word.txt: the sub-ranked list for single-word phrases only.
AutoPhrase_multi-words.txt, the sub-ranked list for multi-word phrases only.

a Did you find any phrases with abnormal scores (e.g., non-phrase with a high score or good phrases with a low score)? Do they show a systematic pattern? What can be the possible reason behind it? **Note**: *This is an open-ended question. Feel free to propose new ideas.* (5 points)

b There is another file **segmentation.txt**, which is partial results from AutoPhrase phrasal segmentation model. In each line, phrases are separated by space; and in each phrase, words are separated by underscores. You need to write your own script to parse the file by splitting each string by space, replacing the in-phrase underscore with white space, and lowercase all words. For example, parsing this document `"Performance_Benchmark Object-Oriented Database_Systems"`, you will get such a list of phrases: {`performance benchmark, object-oriented, database systems`}. Print the result you obtain from first 20 sentences. (5 points)

c Run Word2Vec with gensim library on the above text list. For each phrase listed below, report the ten most similar phrases based on the word2vec model. Phrase list: `computer science, resource management, natural language processing, performance evaluation, data structure, artificial intelligence`. Interpret the results in your words and describe if there are any interesting findings. (10 points)

Problem 2: Image classification with CNN (20 points)

In this question, you will experiment with Convolutional Neural Networks (CNN). We will utilize the **Fashion-MNIST** dataset. You can choose the deep learning framework to use, e.g. Keras, Tensorflow, or Pytorch. These libraries already provide APIs to access the dataset, so you do not have to download the dataset manually. Please conduct a little research to access the dataset using your preferred way. Make sure you have built an environment in which the correct libraries installed. In this assignment, GPUs are not required.

a Construct a CNN model pipeline with one convolution layer with 32 filters of size 3-by-3, stride 1-by-1, followed by one 2-by-2 MaxPooling layer. For the classification, you will apply a linear (fully-connected) layer with the `relu` activation function and 100 output hidden units, and finally a `softmax` layer. Depending on your chosen deep learning framework, you may need to implement a flattening process. Also, apply normalization in your pre-processing as appropriate. Use the Stochastic Gradient Descent (SGD) as the optimizer. You can use a 0.01 learning rate. Report and evaluate the classifier performance. (10 points)

b Play with different learning rates for the optimizer. You may employ a small learning rate $1e-5$ and a large learning rate 1. Report and comment on your observations. (5 points)

c Instead of using one convolution layer, this time we add a convolution layer with 64 filters. Thus we will have two convolution layers in the model. Make sure you set up the activation and pooling layers properly. Present and discuss the results. (5 points)

d (**Extra, Not required, Not graded.**) Till now we have tried various learning rates with SGD. There are various ways to make SGD behave more intelligently, one of which is momentum. Intuitively, when SGD tries to descend down a valley (an analogy for the case where the gradient of one dimension is larger than the gradient of another dimension), SGD might bounce between the walls of the valley instead of descending along the valley. This makes SGD converge slower or even stuck. Momentum works by dampening the oscillations of SGD and encourages it to follow a smoother path. Formally, SGD with momentum update weights by the following way:

$$z^{k+1} = \beta z^k + \frac{\partial L}{\partial w^k}$$
$$w^{k+1} = w^k - \eta * z^{k+1}$$

Here β is the momentum and is between 0 and 1. Please conduct a little research on how to apply a momentum in your optimizer and try it out.

Also, there are a number of other optimizers commonly used in deep learning, including `Adam`, which applies momentum and adaptive learning rates. You are encouraged to try it out. You can also play with different activation functions and gather some interesting observations. You are encouraged to explore as much as you can.