ASSIGNMENT #4 – COMP 3106 ARTIFICIAL INTELLIGENCE

The assignment is an opportunity to demonstrate your knowledge on natural language processing and neural networks and practice applying it to a problem.

The assignment may be completed individually, or it may be completed in small groups of two or three students. The expectations will not depend on group size (i.e. same expectations for all group sizes).

Assignment due date: 2025-12-02

Assignments are to be submitted electronically through Brightspace. It is your responsibility to ensure that your assignment is submitted properly. Copying of assignments is NOT allowed. Discussion of assignment work with others is acceptable but each individual or small group are expected to do the work themselves.

Components

The assignment should contain two components: an implementation and answers to the questions below.

Implementation

Programming language: Python 3

You may use the Python Standard Library (https://docs.python.org/3/library/). You may also use the NumPy, Pandas, and SciPy packages (and any packages they directly depend on). Use of any additional packages requires approval of the instructor.

You must implement your code yourself. Do not copy-and-paste code from other sources, but you may use any pseudo-code we wrote in class as a basis for your implementation. Your implementation must follow the outlined specifications. Implementations which do not follow the specifications may receive a grade of zero. Please make sure your code is readable, as it will also be assessed for correctness. You do not need to prove correctness of your implementation.

You may be provided with a set of examples to test your implementation. Note that the provided examples do not necessarily represent a complete set of test cases. Your implementation may be evaluated on a different set of test cases.

The implementation will be graded both on content and use of good programming practices.

Submit the implementation as a single PY file.

Questions

You must answer all questions posed in the section below. Ensure you answers are clear and concise.

If the assignment was completed in a small group of students, you must also include a statement of contributions. This statement should identify: (1) whether each group member made significant contribution, (2) whether each group member made an approximately equal contribution, and (3) exactly which aspects of the assignment each group member contributed to.

Submit your answers to the questions as a single PDF file.

Implementation

Consider building a simple neural network model to predict the topics of a set of documents: business, entertainment, or politics. Assume we use a term-frequency-inverse document frequency bag of words model to represent documents as vectors and use a simple neural network with three artificial neurons to predict the topic.

We will use a training set of documents to learn the vocabulary and the inverse document frequency vector. You must order your vocabulary in alphabetical order. Use logarithm with base 2 to compute the inverse document frequency vector. The vocabulary and inverse document frequency vector can subsequently be used to compute the representation a never-before-seen document.

We will use a simple neural network architecture to predict topic from a representation of a document. The neural network will have three neurons (one neuron for each topic) that form a single layer. Each artificial neuron has no bias term. The activation function for the layer will be the softmax activation function. That is, each artificial neuron takes the following form, where x is an input vector and w is a vector of weights.

$$\hat{y} = \sigma\left(\sum_j w_j x_j\right)$$

Where the softmax activation function σ takes the following form. Where the i indexes the topics.

$$\hat{z}_i = \frac{e^{\hat{y}_i}}{\sum_j e^{\hat{y}_j}}$$

For this assignment, we will assume that the weights for the neural network will be given. That is, we do not have to train the neural network on a set of training examples.

Your implementation must contain a file named “assignment4.py” with a class named “bag_of_words_model”.

The “bag_of_words_model” class should have three member functions: “__init__”, “tf_idf”, and “predict”.

The function “__init__” is a constructor that should take one input argument (in addition to “self”). The first input argument is a full path to a directory containing documents for training. Each document within the directory will be a TXT file with only lowercase letters and no punctuation.

The function “tf_idf” should take one input argument (in addition to “self”). The first input argument is a full file path to a document. The document will be a TXT file with only lowercase letters and no punctuation. The function should return the term frequency-inverse document frequency vector for the document. Use the vocabulary and inverse document frequency vector learned from the documents passed to the constructor. Ignore any words which do not appear in the vocabulary learned from the documents passed to the constructor.

The function “predict” should take four input arguments (in addition to “self”). The first input argument is a full file path to a document. The document will be a TXT file with only lowercase letters and no punctuation. The second input argument is a list of weights for the artificial neuron for the topic “business”. The third input argument is a list of weights for the artificial neuron for the topic “entertainment”. The fourth input argument is a list of weights for the artificial neuron for the topic “politics”. The function should return two values. The first returned value should be a string indicating the predicted topic class label for the input vector (i.e. “business”, “entertainment”, or “politics”). The second returned value should be a Python list indicating the output of the neural network for each topic: [business score, entertainment score, politics score].

The “tf_idf” and “predict” functions will be called after the constructor is called. They may be called multiple times and in any order. I recommend computing the vocabulary and inverse document frequency vector within the “__init__” function (store them in member variables) and use these in the “tf_idf” and “predict” functions.

Attached are example inputs and corresponding example outputs. Note that your functions should not write anything to file. These examples are provided in separate files for convenience.

Attached is skeleton code indicating the format your implementation should take.

Grading

The implementation will be worth 70 marks.

50 marks will be allocated to correctness on a series of test cases, with consideration to both the term frequency-inverse document frequency vector and the predictions. These test cases will be run automatically by calling your implementation from another Python script. To facilitate this, please ensure your implementation adheres exactly to the specifications.

20 marks will be allocated to human-based review of code.

Questions

Please answer the following questions. Explain why your answers are correct.

1. What type of agent have you implemented (simple reflex agent, model-based reflex agent, goal-based agent, or utility-based agent)? [3 marks]
2. Is the task environment: [7 marks]
    a. Fully or partially observable?
    b. Single or multiple agent?
    c. Deterministic or stochastic?
    d. Episodic or sequential?
    e. Static or dynamic?
    f. Discrete or continuous?
    g. Known or unknown?
3. In the implementation, we assumed that the weights for the neural network were given to us. Describe how we could use a dataset with labelled documents to learn the weights for the neural network (assume the same simple neural network structure described above). Include the following information in your description. [10 marks]
    a. A loss function for your neural network.
    b. The partial derivatives of the loss function with respect to weights in the network.
    c. A gradient-based algorithm to optimize the weights in the network.
    d. A cross-validation protocol.
4. In the implementation, we have used a term frequency-inverse document frequency vector to represent documents as vectors. Describe an alternative representation we could use for documents. [5 marks]
5. Describe an alternative neural network architecture for predicting topic from a term frequency-inverse document frequency vector representation of a document that could provide better performance than the simple architecture described above. Explain why this alternative architecture will achieve better performance. [5 marks]

Grading

The questions will be worth 30 marks, allocated as described above.