CS 4650 Fall 2021: Homework 2

September 8, 2021

Instructions

1. This homework has two parts: Q1–2 are theory questions, and Q3 is a programming assignment with some parts requiring a written answer.

   We will be using Gradescope to collect your assignments. Please read the following instructions for submitting to Gradescope carefully!

   (a) Each subproblem must be submitted on a separate page. When submitting to Gradescope (under HW2 Writing), make sure to mark which page(s) correspond to each problem or subproblem. For instance, Q1 has 2 subproblems, so the solution to each must start on a new page.

   (b) For the coding problem (Q3), please upload the following files on Gradescope:

   i. 'hw2\src\dataset.py'
   ii. 'hw2\src\models.py'
   iii. 'hw2\src\eval.py'
   iv. 'hw2\hw2.ipynb'

   Write your solutions for Q3 (b) **at the end of hw2.ipynb**, and attach pdf exports of 'hw2.ipynb', including outputs, to your writeup. You will be submitting your hw2.ipynb file to Gradescope, **along with** attached to the writeup.

   (c) Note: This is a large class and Gradescope's assignment segmentation features are essential. Failure to follow these instructions may result in parts of your assignment not being graded. We will not entertain regrading requests for failure to follow instructions.

2. LaTeX solutions are strongly encouraged (a solution template is available on the class website), but scanned handwritten copies are also acceptable. Hard copies are not accepted.

3. We generally encourage collaboration with other students. You may discuss the questions and potential directions for solving them with another student. However, you need to write your own solutions and code separately, and not as a group activity. Please list the students you collaborated with on the submission site.

1. Logistic Regression is used to model the probability of a data point belonging to a class or an event occurring. For this, we apply the sigmoid function to a linear combination of independent variables. Logistic Regression is a versatile model that can be used for many different types of models.

   Mathematically, logistic regression is defined as:

$$\begin{aligned} \hat{y} &= \sigma(z) \\ \sigma(z) &= \frac{1}{1+e^{-z}} \\ z &= (wx+b) \end{aligned}$$

   The result is the probability that tells us how the model would classify the given data-point x based on the learnable weights w and bias b. Let's assume we have a labelled dataset for detecting hate speech. We can then use Gradient Descent to optimize the weights of the model. However, we need to give Gradient Descent a good loss function to optimize. Let's use the cross entropy loss function for this. Since the cross entropy loss is convex for logistic regression, it has only one minima.

$$\mathcal{L}_{\mathcal{CE}}(\hat{y}, y) = -[ylog(\hat{y}) + (1-y)log(1-\hat{y})]$$

   where $\hat{y}$ is the predicted probability and y is the ground-truth label.

   With gradient descent,

$$w^{t+1} = w^t - \eta \frac{\partial \mathcal{L}}{\partial w}$$

   where η is the learning rate.

   (a) Calculate the partial derivative of the cross entropy loss with respect to the weights w and bias b. [show the calculation] (4 points)

   (b) What is the gradient descent update step for logistic regression with the cross entropy loss? (1 point)

2. You find yourself eating popcorn and watching The Office after your NLP course. You notice that Michael, Jim and Dwight use very distinctive language from one another and get to wondering whether you could apply your text classification knowledge to build a multi-class classifier. You collect the dataset of all transcripts from The Office and utilize regex to isolate dialogues by Michael, Jim and Dwight. You build and train a simple multinomial logistic regression using bag-of-words input representation (not from scratch, you utilize Sci-Kit Learn because you'd still like to go back to streaming).

   Here is the confusion matrix obtained after running the classifier on your held-out test set:

		True Labels		
		Michael	Jim	Dwight
Predicted Labels	Michael	8200	300	100
	Jim	800	4200	200
	Dwight	500	500	3500

Table 1: The Office Confusion Matrix

(a) Calculate the **precision** and **recall** for all three classes (2 points).

(b) Calculate the **macroF1** score (2 points).

(c) Calculate the **microF1** score. **Explain** why it looks different from macroF1. (4 points)

(d) You want to improve the classifier using a perceptron classifier. You decide to go with one hidden layer. However, you're unsure about what activation functions to use. The default option is ReLU but you are unsure. What are the **advantages** and **disadvantages** of using Sigmoid or Tanh activations? *Which* activation function should you use and **why**? (3 points)

(e) **What** is Leaky ReLU? **Would** you use it over the default option (ReLU)? **Why** or why not? (2 points)

3. In this assignment, you will implement some introductory DataLoaders and Models using PyTorch, a popular deep-learning library. [35 points]

   Specifically, you'll be writing code to classify if headlines come from The Onion (a fake satirical newspaper) or are actual headlines (**binary classification**). We'll be using this dataset, included in the assignment.zip file.

   Sometimes, telling the differences can be tricky. Here are two examples:

   (a) From The Onion:
   Entire Facebook Staff Laughs As Man Tightens Privacy Settings

   (b) Real Headline:
   Cyclist's Bike Stolen at Police Station while Reporting iPhone Theft

   You can download the assignment zip here:
   `https://www.cc.gatech.edu/classes/AY2022/cs4650_fall/programming/h2_torch.zip`

   A jupyter notebook: 'hw2\hw2.ipynb' will guide you through the overall workflow for the homework, from loading and reshaping data to implementing a model and executing a training/testing loop. We strongly recommend using Google Colab to import the hw2.ipynb notebook, and following the instructions from there!

   Here is a shared, edit-only colab that you may find useful. You can clone this notebook into your local drive.

   `https://colab.research.google.com/drive/1DuP7gFA2w0NJZO6jLdCSvOAel2kR6MMl`

   You need to submit your the following files to assignment HW2 Code in Gradescope. For this HW, only a portion of your assignment will be autograded by Gradescope.

   (a) 'hw2\src\dataset.py'
   (b) 'hw2\src\models.py'
   (c) 'hw2\src\eval.py'
   (d) 'hw2\hw2.ipynb'

   Finally, you should respond to the written questions in the analysis section of the notebook, and you should attach a PDF of your notebook output to your HW2 Writing submission.

   (a) Run the jupyter notebook 'hw2/hw2.ipynb' and follow-along, implementing the necessary functions in neighboring files as the script calls for them. To run the notebook locally (*only if you don't want to use Colab*), navigate to the 'hw2' directory in your terminal and run 'jupyter notebook'.
   (b) Complete the analysis questions at the end of the notebook.
      i. What happens to the vocab size as you limit words by their frequency? Can you explain this in the context of Zipf's Law?
      ii. Can you qualitatively describe (1 paragraph) what cases the model is getting wrong in the witheld test-set (see notebook for details)?