University of Central Missouri

Department of Computer Science & Cybersecurity

CS5760 Natural Language Processing

Spring 2026

Homework 3.

Student name: Nikhilesh Katakam

Submission Requirements:

- No programming Question.
- No need for GitHub this time.
- Any submission after provided deadline is considered as a late submission.

Part I. Writing Calculation:

Q1. Sigmoid Classification with New Values

You are given a feature vector `x = [2, 4]`, weight vector `w = [0.5, -0.3]`, and bias `b = 0.2`.

Tasks:

- (a) Compute the linear combination: z=w·x+b
- (b) Apply the sigmoid function: $\sigma(z) = \frac{1}{1+e^{-z}}$
- (c) Predict the class label using threshold 0.5

Solution:
(x = [2, 4]), (w = [0.5, -0.3]), (b = 0.2)

(a) Linear combination

$$z = w \cdot x + b$$

$$= (0.5)(2) + (-0.3)(4) + 0.2$$

$$= 1 - 1.2 + 0.2$$

$$= 0$$

Answer**:** (z = 0)

(b) Sigmoid function

$\sigma(z) = \frac{1}{1+e^{-z}}$

$= \frac{1}{2}$

$= 0.5$

Answer: Sigmoid output = **0.5**

(c) Class prediction (threshold = 0.5)

Solution: Since output = 0.5

Prediction = Class 1

Q2. One SGD Update Step

A sample has input vector x=[1,2], true label y=0, current weights w=[0.1,−0.2], bias b=0.05, and learning rate η=0.1.

Tasks:

- (a) Compute $z = w \cdot x + b$ and $\hat{y} = \sigma(z)$
- (b) Compute the gradients of the loss w.r.t. weights and bias
- (c) Update weights and bias using gradient descent

  Solution:

I'll solve this as **logistic regression with sigmoid activation and binary cross-entropy loss**, which is the standard assumption for this type of question.

Given:

- (x = [1, 2])
- (y = 0)
- (w = [0.1, -0.2])
- (b = 0.05)
- Learning rate (\eta = 0.1)

(a) Compute the predicted output (\hat{y})

Step 1: Linear combination

Z = (0.1) (1) +(-0.2) (2) +0.05

= 0.1 -0.4+0.05

= -0.25

Step 2: Sigmoid activation

$$\gamma\hat{} = \frac{1}{1+e0.25}$$

$$\approx 0.4378$$

(b) Compute gradients

$$\gamma^{-y} = 0.4378 - 0 = 0.4378$$

Gradients For weights:

dw = 0.4378 × [1,2]

= [0.4378,0.8756]

Gradient For bias:

db = 0.4378

(c) Update weights and bias (Gradient Descent)

= [0.1, -0.2] -0.1[0.4378,0.8756]

= [0.0562, -0.2876]

Bais:

b = 0.55-0.1(0.4378)

= 0.0062

Q3. Cosine Similarity Between Word Vectors

Given word vectors:

- $\vec{a}$=[3,4]
- $\vec{b}$=[4,3]

Tasks:

- (a) Compute the dot product $\vec{a}\cdot\vec{b}$
- (b) Compute magnitudes || a|| and || b||
- (c) Compute cosine similarity:

$$\cos(\theta) = \frac{\vec{a}\cdot\vec{b}}{||a||\cdot||b||}$$

Solution:

a. Dot product

= (3)(4) + (4)(3)
= 12+12
= 24

b. Magnitudes:

$||a|| = \sqrt{3^2 + 4^2}$

$= \sqrt{25}$

$= 5$

$||b|| = 5$

c. Cosine similarity

$= \frac{24}{5 \times 5}$

$= \frac{24}{25}$

$= 0.96$

Q4. TF-IDF Ranking Task

Query: "apple pie"
Documents:

- D1: apple=1/5, pie=1/5
- D2: apple=0.1, pie=0
- D3: apple=0, pie=0.25
  Assume IDF("apple") = 0.6, IDF("pie") = 0.8

Tasks:

- (a) Compute TF-IDF score for each term in each document
- (b) Compute total TF-IDF per document
- (c) Rank D1, D2, D3 by score

Solution:

a. TF-IDF per term

D1: Apple = (1/5) (0.6) = 0.12
Pie = (1/5) (0.8) = 0.16

D2: Apple = (0.1) (0.6) = 0.06
Pie = 0

D3: Apple = 0
Pie = (0.25) (0.8) = 0.20

b. Total per Document:

D1= 0.12+0.16 = 0.28
D2= 0.06
D3 = 0.20

c. Ranking:

1. D1(0.28)
2. D3(0.20)
3. D2(0.06)

Q5. PMI vs PPMI

Given:

- P(word, context)=0.015
- P(word)=0.1
- P(context)=0.2

Tasks:

- (a) Compute PMI
- (b) Compute PPMI by thresholding negatives at 0

Solution:

A. PMI

$$\text{PMI} = \log \frac{0.0015}{0.1 \times 0.2}$$

$$= \log \frac{0.015}{0.02}$$

$$= \log (0.75)$$

$$= -0.125$$

b. PPMI

Since PMI is negative PPMI = 0

Part II: Short Answer

Q1. Understanding Model Behavior and Training in Machine Learning

(a) What is the main difference between generative and discriminative models? Provide one example of each.

Answer:

Generative models

- Learn the combined probability distribution of features and labels (P (x, y))
- Are capable of creating new data samples
- Aim to understand and represent how the data is generated
- Example: Naive Bayes

Discriminative models

- Learn the probability of a label given the input (P (y \mid x))
- Concentrate on distinguishing between different classes
- Commonly used for classification and regression tasks
- Example**:** Logistic Regression

(b) What role does the cross-entropy loss function play during training?

Answer: Cross-entropy loss evaluates how far the model's predicted probabilities are from the actual labels. It assigns a higher penalty to wrong predictions, especially when the model is very confident but incorrect. During training, this loss function supplies useful gradient information that guides parameter updates through optimization techniques such as gradient descent. By minimizing cross-entropy loss, the model improves both its classification accuracy and the quality of its probability predictions.

Q2. Generalization and Representation in NLP

(a) What is the difference between synonymy and word similarity? Provide an example.

Answer:

Synonymy vs. Word Similarity

- Synonymy describes words that share the same or almost the same meanin**g** and can often replace each other in sentences.
  Example: car *and* automobile
- Word similarity refers to how closely connected two words are in meaning, even if they are not interchangeable.
  Example: car *and* road.

(b) How does TF-IDF improve upon raw term frequency when representing document content?

Answer: TF-IDF vs. Raw Term Frequency

- Raw term frequency simply counts how many times a word appears in a document, without considering its importance.
- TF-IDF improves this by down-weighting very common words and highlighting less frequent but more informative terms**.**
- This leads to a better representation of a document’s key topics**.**

(c) Explain why PPMI values are clipped at 0. What does a negative PMI indicate?

Answer: PPMI Clipping and Negative PMI

- PMI measures whether two words appear together more or less often than expected**.**
- A negative PMI shows that two words co-occur less frequently than chance would predict**.**
- In PPMI, negative values are set to zero because they do not contribute useful semantic information**.**
- This keeps only positive associations between words.