Exercise sheet 2: Text classification with Logistic Regression and Naive Bayes

A note on exercise sheets In the exercise sheets, you'll encounter three types of exercises:

1. *Mandatory and **G**raded Exercise*: These exercises must be completed and determine your grade for the assignment. You'll find them marked with [MG].
2. *Mandatory and **U**ngraded Exercise*: These exercises are ungraded, but must nevertheless be completed to pass the assignment. These exercises are designed to help your revise the concepts introduced in class. If you solve them correctly, we will consider this for rounding up your final course grade. You'll find them marked with [MU].
3. ***O**ptional Exercise*: These exercises are not mandatory but can help with revision. If you solve them correctly, we will consider this for rounding up your final grade. You'll find them marked with [O].

Individual submission notice: Each student is required to complete and submit their solutions **individually**. While discussing general concepts with peers is OK, the final work must be your own.

Submission format: The solutions to these exercises must be completed in this Colab notebook. Please fill in the required answers as instructed, download the notebook, and submit it to Moodle following the naming convention `STUDENT_ID.ipynb`.

Problem 1: Predicting book categories from titles [MG]

Suppose you are asked to build a text classifier for a bookstore. Given the title of a book, the goal is to predict its category. For simplicity, assume there are only two possible categories: **AI (Artificial Intelligence)** and **Psychology**.

A set of books has been labeled in advance to train your text classifier. The books are as follows:

AI books:

- "*Deep Learning*" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
- "*Artificial Intelligence: A Modern Approach*" by Stuart Russell and Peter Norvig
- "*The Alignment Problem*" by Brian Christian
- "*Reinforcement Learning: An Introduction*" by Richard Sutton and Andrew Barto

Psychology books:

- "*Learning to Breathe: A Mindfulness Curriculum*" by Patricia Broderick
- "*Deep Work: Rules for Focused Success in a Distracted World*" by Cal Newport
- "*Deep Survival: Who Lives, Who Dies, and Why*" by Laurence Gonzales
- "*The Learning Brain: Lessons for Education*" by Sarah-Jayne Blakemore and Uta Frith

DEEP LEARNING
Ian Goodfellow, Yoshua Bengio, and Aaron Courville
GLOBAL EDITION
Stuart Russell
Peter Norvig
Artificial Intelligence
A Modern Approach
Fourth Edition
THE ALIGNMENT PROBLEM
Machine Learning and Human Values
BRIAN CHRISTIAN
Reinforcement Learning
An Introduction
second edition
Richard S. Sutton and Andrew G. Barto
SECOND EDITION
Learning to BREATHE
A Mindfulness Curriculum for Adolescents to Cultivate Emotion Regulation, Attention, and Performance
PATRICIA C. BRODERICK, PHD
DEEP WORK
RULES FOR FOCUSED SUCCESS IN A DISTRACTED WORLD
OVER 1 MILLION COPIES SOLD
Cal Newport
DEEP SURVIVAL
Who Lives, Who Dies, and Why
LAURENCE GONZALES
Sarah-Jayne Blakemore and Uta Frith
the learning brain
lessons for education

To keep things simple, assume that only the main parts of the titles are kept in the training examples. The training set looks as follows:

DOCUMENT	LABEL
deep learning	ai
artificial intelligence	ai
the alignment problem	ai
reinforcement learning	ai
learning to breathe	psychology
deep work	psychology
deep survival	psychology
the learning brain	psychology

You decide to use a **Naive Bayes classifier** for this task. You compute class priors and class conditional probabilities, and get your classifier ready to predict categories for new books. Just as you finish, a new book arrives at the store:

"Understanding Deep Learning" by Simon J.D. Prince.

You think to yourself: "Clearly, this is an AI book". Happy with the opportunity to test your classifier, you decide to see how it performs with the new book.

Complete the following parts (for this problem, use probabilities, not log-probabilities):

1. Determine the predicted class $c_{NB} \in \{\texttt{ai}, \texttt{psychology}\}$ for this new book title.

   The word "`understanding`" is not in the vocabulary of the classifier, so it gets dropped from the book title. Thus, the classification decision will be based only on the words `deep` and `learning`.

   Recall that the Naive Bayes classifier assigns probabilities to classes as follows:

Definition 1: Let $d = (w_1, \dots, w_n)$ be a document and C a set of classes. The Naive Bayes classifier assigns to document d the class c_{NB} with maximum posterior probability, defined as follows:

$$\begin{aligned} c_{NB} &= \arg\max_{c \in C} \underbrace{P(w_1|c)P(w_2|c)\dots P(w_n|c)}_{\text{Likelihood factorizes under NB assumption}} \underbrace{P(c)}_{\text{Prior}} \\ &= \arg\max_{c \in C} \underbrace{P(c)}_{\text{Prior over classes}} \prod_{1 \le i \le n} \underbrace{P(w_i|c)}_{\text{Class-conditional probabilities for words}} \end{aligned}$$

no hace falta utilizar smoothing

   (a) Compute the class priors: $P(\texttt{ai})$ and $P(\texttt{psychology})$.

   (b) Compute the class-conditional probabilities $P(w \mid c)$ for "`deep`" and "`learning`".

   (c) Compute the values $P(\texttt{ai}) \prod_w P(w \mid \texttt{ai})$ and $P(\texttt{psychology}) \prod_w P(w \mid \texttt{psychology})$.

   (d) Determine the predicted class c_{NB} based on $\arg\max$.

   Provide your calculations.

You quickly discover that the classifier has made a mistake with the new book. Unhappy with the results, you decide to ditch the Naive Bayes classifier and replace it with a **logistic regression model** (see Definition 2 in the next exercise for the definition of logistic regression). **Complete the following parts**:

2. **Compute the BoW vectors** for the 8 training examples. Use the following ordering of words for the BoW vectors of the i-th example:

$$\mathbf{x}^{(i)} = \begin{bmatrix} x_1 & \text{deep} \\ x_2 & \text{learning} \\ x_3 & \text{artificial} \\ x_4 & \text{intelligence} \\ x_5 & \text{the} \\ x_6 & \text{alignment} \\ x_7 & \text{problem} \\ x_8 & \text{reinforcement} \\ x_9 & \text{to} \\ x_{10} & \text{breathe} \\ x_{11} & \text{work} \\ x_{12} & \text{survival} \\ x_{13} & \text{brain} \end{bmatrix}$$

3. **Choose a weight vector $\boldsymbol{\theta}$** such that, *for all training examples, the classifier outputs the correct class prediction* (class 1 for `ai` and class 0 for `psychology`). Set the bias term b to 0. **You do not need to provide any calculations here**; just write down suitable values of the weight vector $\boldsymbol{\theta}$.

Exercise 2: Improving a sentiment analysis classifier [MG]

Consider a binary sentiment classification task with the following setup:

- **Vocabulary** $\mathcal{V}$:

  $$\underbrace{\text{the}}_{\text{word 1}},\ \underbrace{\text{storyline}}_{\text{word 2}},\ \underbrace{\text{was}}_{\text{word 3}},\ \underbrace{\text{horribly}}_{\text{word 4}},\ \underbrace{\text{intriguing}}_{\text{word 5}},\ \underbrace{\text{and}}_{\text{word 6}},\ \underbrace{\text{dull}}_{\text{word 7}},\ \underbrace{\text{captivating}}_{\text{word 8}},\ \underbrace{,}_{\text{word 9}},\ \underbrace{\text{not}}_{\text{word 10}}$$

  Notice that word 9 is the comma.

- **Classes**: $\{0, 1\}$, where $0 =$ negative and $1 =$ positive.

- **Input documents and labels**:

  $$d^{(1)} = \text{“the storyline was intriguing and captivating”}, \quad y^{(1)} = 1 \text{ (positive)}$$

  $$d^{(2)} = \text{“dull, horribly dull storyline, not intriguing”}, \quad y^{(2)} = 0 \text{ (negative)}$$

You will use a **logistic regression** classifier to compute class probabilities for these documents. Recall the definition of the logistic regression classifier:

Definition 2: Let $\mathbf{x} \in \mathbb{R}^n$ represent the input document vector and $y \in \{0, 1\}$ a class label.

The **logistic regression** classifier with weights $\boldsymbol{\theta} \in \mathbb{R}^n$ and bias $b \in \mathbb{R}$ computes the probability of $y = 1$ as:

$$P(y = 1 \mid \mathbf{x}) = \sigma(z), \quad z = \boldsymbol{\theta} \cdot \mathbf{x} + b,$$

where

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

is the **sigmoid** function. The **classification decision** is made by choosing the most probable class:

$$\text{decision}(\mathbf{x}) = \begin{cases} 1 & \text{if } P(y = 1|\mathbf{x}) > 0.5, \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ denote the **Bag-of-Words** (BoW) vectors for documents $d^{(1)}$ and $d^{(2)}$, respectively. Your goal is to *choose a weight vector* $\boldsymbol{\theta} \in \mathbb{R}^{10}$ *such that, for these particular documents, the classifier outputs*:

$$P(y^{(1)} = 1 \mid \mathbf{x}^{(1)}) \approx 0.75 \ (\pm 0.01), \quad P(y^{(2)} = 0 \mid \mathbf{x}^{(2)}) \approx 0.75 \ (\pm 0.01).$$

The bias term b must be fixed to 0. In particular, complete the following steps:

1. **Compute the BoW vectors** $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ for the two documents.

2. **Propose a weight vector** $\boldsymbol{\theta}$ such that the logistic regression outputs the desired probabilities $P(y = 1 \mid \mathbf{x}^{(1)}) \approx 0.75$ and $P(y = 0 \mid \mathbf{x}^{(2)}) \approx 0.75$. Briefly justify your approach and show the calculations. *You must use non-zero weights for the following words (and only those)*:

   dull (word 7), horribly (word 4), intriguing (word 5), captivating (word 8).

   Hint: The *inverse* of the sigmoid function $\sigma(z)$ is the **logit** function:

   $$\sigma^{-1}(p) = \ln\!\left(\tfrac{p}{1-p}\right).$$

   Hence, to find weights satisfying $\sigma(\boldsymbol{\theta} \cdot \mathbf{x}^{(i)} + b) = p$ for a certain p, you can use

   $$\boldsymbol{\theta} \cdot \mathbf{x}^{(i)} + b = \sigma^{-1}(p) = \ln\!\left(\tfrac{p}{1-p}\right).$$

3. Let $\hat{y}^{(i)}$ denote the predicted probability for class 1 on document i:

   $$\hat{y}^{(i)} = \sigma(\boldsymbol{\theta} \cdot \mathbf{x}^{(i)}).$$

   Compute the cross-entropy loss for each example separately:

   $$\mathcal{L}_{CE}\big(\hat{y}^{(i)}, y^{(i)}\big) = -\Big[y^{(i)} \log \hat{y}^{(i)} + \big(1 - y^{(i)}\big) \log\big(1 - \hat{y}^{(i)}\big)\Big].$$

4. Perform **one** step of gradient descent with both training examples $\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)})\}$. Use a learning rate $\alpha = 0.1$. Specifically, report:

(a) The gradient $\nabla_{\boldsymbol{\theta}}$.

(b) The new $\boldsymbol{\theta}$ (after the gradient update).

(c) Do the weight changes make sense to you? Which weights changed, and how do these changes help the classifier improve its predictions?

5. Classify the two examples again with the new weights. Compute $P(y = 1 \mid \mathbf{x}^{(1)})$ and $P(y = 0 \mid \mathbf{x}^{(2)})$ again, with the new weights. Did the classification predictions improve? That is, are the predicted probabilities for the true labels now closer to 1, for all examples?