Faculty of Science and Technology

CHARLES DARWIN UNIVERSITY AUSTRALIA

Final Examination

HIT391 – Machine Learning: Advancements and Applications	DURATION	
	Reading Time:	10 minutes
	Writing Time:	120 minutes (2:00pm – 4:00pm, 26 May)
	Scanning and uploading:	10 minutes
INSTRUCTIONS TO CANDIDATES		
Notes: 1.1 The examination has 4 sections (10 questions). You must answer all the questions. Please ensure that your name and student number are indicated on your Answer sheets. 1.2 Read ALL questions carefully. 1.3 Scan and upload your answer sheets (.pdf). 1.4 Total marks available on this test are 30.		
EXAM CONDITIONS		
You may begin writing from the commencement of the examination session. The reading time indicated above is provided as a guide only. Please make sure that you submit your work on time.		
This is a OPEN BOOK (slides only) examination		

Student Name: Musrat Jahan Student Number: S380098

.

Total No. of Marks for the Examination: 30

Part I: Mutiple Choice Questions(*2 marks* for each question)

1. **Which of the following is not a measure of data quality? (C)**

   A. Accuracy B. Completeness C. Scalability D. Timeliness
   Answer: C. Scalability

2. **Which of the following best describes the role of the self-attention mechanism in transformer-based language models such as GPT? (B)**

   A. It forces each token to only attend to the next token, preserving left-to-right generation order.
   B. It allows each token to attend to all other tokens in the sequence to learn contextual relationships, enabling long-range dependency modeling.
   C. It compresses the input into a fixed-size vector, which is used for downstream tasks.
   D. It enhances training speed by removing token dependencies.

Answer: B. It allows each token to attend to all other tokens in the sequence to learn contextual relationships, enabling long-range dependency modeling.

Part II: Question and Answering (short explanations required, *2 marks* for each question)

3. **When designing a machine learning model for disease diagnosis, what ethical issues should we consider? Please list at least TWO examples of ethical issues and explain their significance in the context of disease diagnosis.**

   Answer:
   Two Examples of ethical issues in Disease diagnosis:
   1. Privacy: When using a machine learning model for diagnosis, the system needs to keep patients' data safe and secure. Medical data is very personal. It can hurt people and break their trust if it gets leaked or misused.

   2. Transparency: When using a machine learning model for disease diagnosis, doctors and patients need to know about how the machine makes decisions. Also, they need to understand why the AI says someone is sick. They may not trust it if they do not know how it works.

4. **How do you handle missing values in a dataset? Please list at least TWO methods for dealing with missing values.**

   - Regression: Using regression functions to detect missing values.
   - Combined computer and human inspection: The computer detects suspicious or missing values and checks and fixes by human.

5. **Sampling strategies are commonly used in machine learning, please list at least TWO types of sampling methods.**

   - Simple random sampling: Every data has an equal chance of selecting any item.
   - Sampling without replacement: Once a data is selected, it is removed and it cannot be choose again.

6. **In deep neural networks, what is an activation function? What is a loss function? Briefly explain the differences between them.**

   - The activation function selects whether a neuron should be activated by introducing nonlinearity into the network.
   - The loss function defines how far the model's predictions differ from the actual values.

Difference	**Activation Function**	**Loss Function**
Purpose	Adds non-linearity to model	Measures prediction error
Applied To	Neuron output (within the network)	Final model output vs actual value
Helps With	Learning complex patterns	Training the model using optimization

7. **In Support Vector Machine (SVM), what is the hyperplane? Please provide a brief explanation of its role in the classification problem.**

   In Support Vector Machine (SVM), a hyperplane is a decision boundary that divides data into classes. The hyperplane assists in classifying new data points by placing them on one side or the other—ideally with the most significant margin (distance) from the nearest points of each class.

8. **What is Natural Language Processing (NLP) used for? Please provide a brief description of at least TWO applications of NLP.**

   - Chatbots and Virtual Assistants – NLP allows systems like Siri or ChatGPT to understand and respond to user queries in natural language.
   - Spam Detection – Email services use NLP to analyze the content of messages and detect unwanted or harmful spam emails.

Part III: Calculation (7 *marks*)

9. **Calculation for Feedforward Process in a Neural Network**

.

Consider a 2-layer multi-layer perceptron(MLP) shown in *Figure 1* with the following architecture,
Input layer: 2 neurons
Hidden layer: 2 neurons with ReLU activation function
Output layer: 1 neuron with Sigmoid activation function

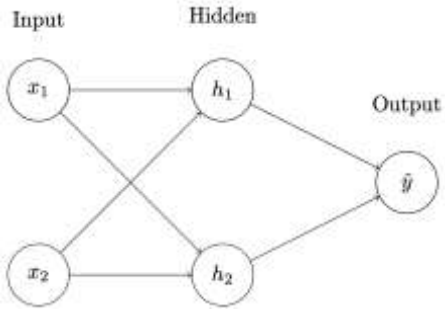


Figure 1: 2-layer MLP

Instructions:

- Apply the ReLU activation function after the hidden layer.

$$\mathrm{ReLU}(z) = \max\{0, z\}$$

If the input z is positive, ReLU returns the input value.

If the input z is zero or negative, ReLU returns 0.

- Use the following Sigmoid function for the output layer

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

You are allowed to use a scientific calculator (which usually has an "exp" button for e^x), where the e refers to the natural exponential base e≈2.71828, not base 10.

Or, you can leave the answer in expression form.

- The weights and biases are as follows:

Input to Hidden Layer:

$$w_1 = \begin{bmatrix} 0.2 & 0.4 \\ -0.5 & 1.0 \end{bmatrix}, \quad b_1 = \begin{bmatrix} 0.1 \\ -0.2 \end{bmatrix}$$

Hidden to Output Layer:

$$w_2 = [0.3, -0.8], \quad b_2 = [0.05]$$

Requirement: Given the input $x = \begin{bmatrix} 1.0 \\ -1.0 \end{bmatrix}$, compute the output of the network.

Answer:

.

Input Layer: x =

$$= \begin{bmatrix} 1.0 \\ -1.0 \end{bmatrix}$$

Input to Hidde n Layer weighs:

$$w_1 = \begin{bmatrix} 0.2 & 0.4 \\ -0.5 & 1.0 \end{bmatrix}$$

Input to Hidden Layer biases:

$$b_1 = \begin{bmatrix} 0.1 \\ -0.2 \end{bmatrix}$$

Hidden to Output Layer weighs:

$$w_2 = [0.3, -0.8]$$

Hidden to Output Layer Biases:

$$b_2 = [0.05]$$

Hidden layer input :

$$z_1 = w_1 \times x + b_1 = \begin{bmatrix} -0.1 \\ -1.7 \end{bmatrix}$$

ReLU activation on hidden layer: $a_1 = \text{ReLU}(z) = \max\{0, z\} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Output layer input:

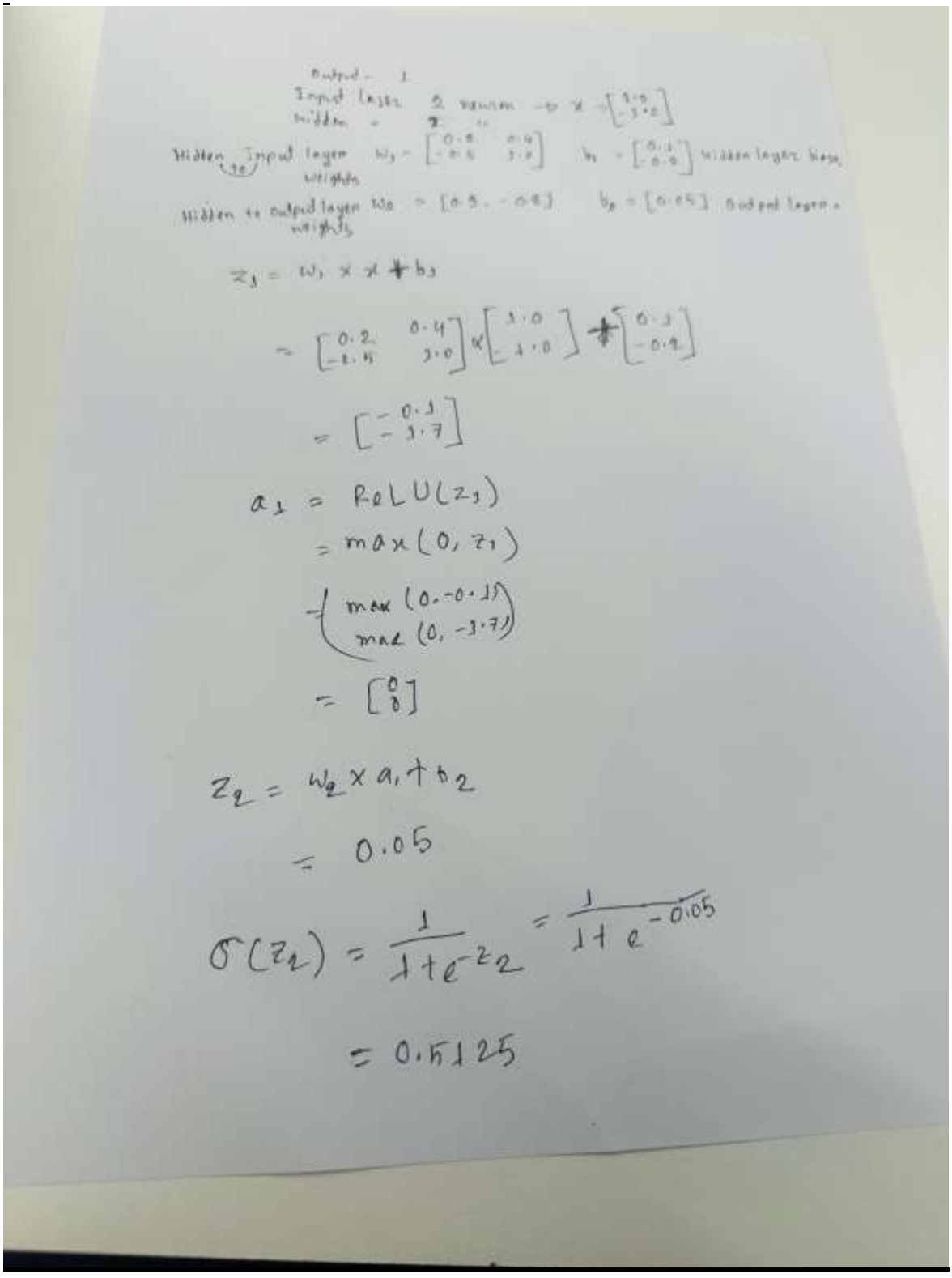
$$z_2 : w_2 \times a_1 + b_2 = 0.05$$

Sigmoid function for the output layer:

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

$$= 0.51$$

.



Part IV: Programming (7 *marks*)

10. Programmng for Sentiment Classification with SVM

Write a Python program (or Pseudo code) using scikit-learn to perform sentiment analysis on the following toy dataset:

.

```
texts = ["I love this movie!",
    "This film was terrible.",
    "What a great experience.",
    "I hated every moment of it.",
    "Absolutely fantastic story.",
    "Worst acting I've seen."]

labels = [1, 0, 1, 0, 1, 0]  # 1: Positive, 0: Negative
```

Requirements:

1.Preprocess the text using TF-IDF vectorization.
2.Train a Support Vector Machine (SVM) classifier to perform sentiment classification on the provided dataset.
3.Print the classification accuracy on the training data (you do not need to output the actual result—just ensure the print statement is present).

Answer:

Pseudo Code for Sentiment Classification with SVM

Start

# Step 1: Define the dataset
texts = ["I love this movie!",
"This film was terrible.",
"What a great experience.",
"I hated every moment of it.",
"Absolutely fantastic story.",
"Worst acting I've seen."]

labels = [1, 0, 1, 0, 1, 0] # 1 = Positive, 0 = Negative

Step 2: Convert text into numeric features using TF-IDF
Initialize a TF-IDF Vectorizer # scikit-learn: TfidfVectorizer()
Transform texts into TF-IDF features # scikit-learn: vectorizer.fit_transform(texts)→ X

Step 3: Train the SVM model
Initialize an SVM classifier with linear kernel # scikit-learn: SVC(kernel='linear')
Train the SVM model on TF-IDF features and labels # scikit-learn: svm.fit(X, labels)

Step 4: Predict and evaluate
Predict sentiment labels using trained model # scikit-learn: svm.predict(X)
Compare predicted labels with actual labels
Calculate accuracy # scikit-learn: accuracy_score(labels, predictions)
Print the accuracy

End