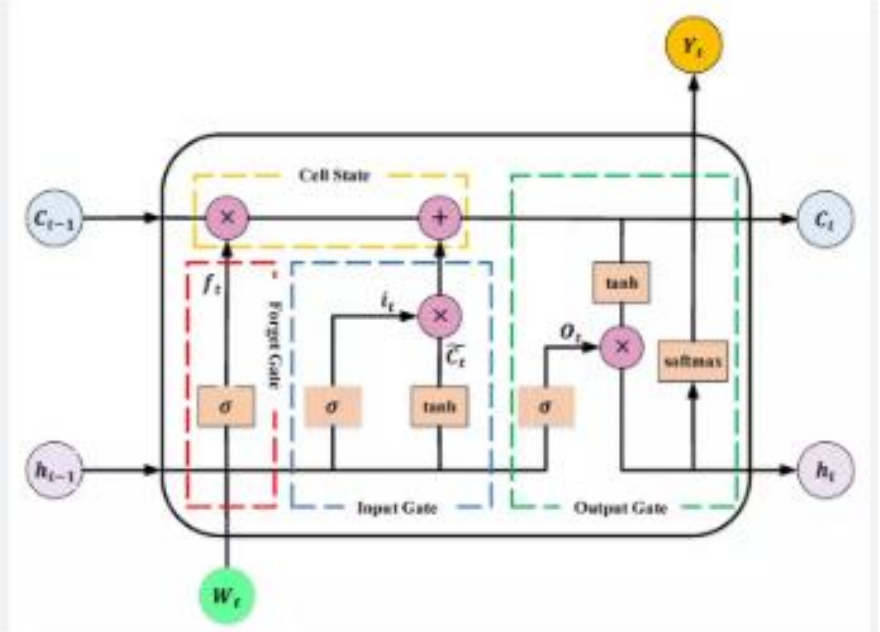
LAB No. 8

Implementation of Long Short-Term Memory (LSTM) Network

In this lab, students will learn and implement the **Long Short-Term Memory (LSTM)** network, an advanced type of **Recurrent Neural Network (RNN)** designed to overcome the limitations of basic RNNs. LSTM networks are capable of learning **long-term dependencies** in sequential data through a special memory cell and gating mechanism. Students will apply LSTM models to sequence classification, time-series prediction, and text-based tasks using Python and Keras, and evaluate model performance using appropriate metrics.

Introduction & Theory (Brief)

Long Short-Term Memory (LSTM) is a special kind of RNN that effectively handles the **vanishing gradient problem**. It introduces a **memory cell** that can store information for long periods and three gates that regulate information flow:



Key Components of LSTM:

- **Forget Gate** – decides what information to discard
- **Input Gate** – decides what new information to store
- **Output Gate** – controls what information to output
- **Cell State** – long-term memory

Advantages:

- Learns long-term dependencies
- Suitable for time-series, speech, and text data
- More stable training than basic RNNs

Applications:

- Time-series forecasting
- Sentiment analysis
- Speech recognition
- Language translation

Solved Examples:

Example 1: LSTM for Binary Sequence Classification

Build an LSTM model to predict whether a student **passes or fails** based on performance over multiple time steps.

Solution:

```
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

# Sequential dataset (samples, timesteps, features)
X = np.array([
   [[1], [1], [0]],
   [[1], [1], [1]],
   [[0], [0], [1]],
   [[0], [0], [0]],
   [[1], [0], [1]]
])

y = np.array([1, 1, 0, 0, 1])

# Build LSTM model
model = Sequential()
model.add(LSTM(8, input_shape=(3, 1)))
model.add(Dense(1, activation='sigmoid'))

# Compile and train
model.compile(optimizer='adam',                    loss='binary_crossentropy',
metrics=['accuracy'])
model.fit(X, y, epochs=100, verbose=0)



# Prediction
prediction = model.predict([[[1], [1], [1]]])
print("Predicted Result (Pass=1, Fail=0):", int(prediction[0][0] > 0.5))
```

The LSTM retains relevant information across time steps using its memory cell and gating mechanism.

Example 2: LSTM for Time-Series Prediction

Use an LSTM network to predict the next value in a numerical time-series sequence.

Solution:

```
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

# Time-series input sequences
X = np.array([
   [[1], [2], [3]],
   [[2], [3], [4]],
   [[3], [4], [5]],
   [[4], [5], [6]]
])

y = np.array([4, 5, 6, 7])

# Build LSTM model
model = Sequential()
model.add(LSTM(10, input_shape=(3, 1)))
model.add(Dense(1))

# Compile and train
model.compile(optimizer='adam', loss='mse')
model.fit(X, y, epochs=200, verbose=0)

# Predict next value
prediction = model.predict([[[5], [6], [7]]])
print("Predicted Next Value:", prediction[0][0])
```

LSTM captures temporal dependencies more effectively than basic RNNs for time-series prediction.

Example 3: LSTM for Text Classification (Sentiment Analysis)

Build an LSTM model to classify text as **positive** or **negative** sentiment.

Solution:

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

# Text samples
texts = [
   "I love this subject",
   "This lab is excellent",
   "I hate this course",
   "This topic is boring",
   "Very good explanation"
]

labels = [1, 1, 0, 0, 1]

# Tokenization
tokenizer = Tokenizer(num_words=100)
tokenizer.fit_on_texts(texts)

sequences = tokenizer.texts_to_sequences(texts)
padded_sequences = pad_sequences(sequences, maxlen=5)


# Build LSTM model
model = Sequential()
model.add(LSTM(16, input_shape=(5,)))
model.add(Dense(1, activation='sigmoid'))

# Compile and train
model.compile(optimizer='adam',                    loss='binary_crossentropy',
metrics=['accuracy'])
model.fit(padded_sequences, labels, epochs=100, verbose=0)

# Predict sentiment
predictions = model.predict(padded_sequences)
print("Predicted Sentiments:", predictions.round())
```

LSTM processes text sequences while preserving context, leading to improved sentiment classification.

Comparison: RNN vs LSTM

Feature	RNN	LSTM
Long-Term Memory	No	Yes
Vanishing Gradient	Yes	No
Training Stability	Low	High

LAB Assignment No.8

LSTM code for text predictor

Question:

Using the given LSTM next-word prediction model and tokenizer, write the code to input a sentence, convert it into sequences, pad it, and predict the next word using the trained model.

Solution:

```
faqs = """About the Program
What is the course fee for  Data Science Mentorship Program (DSMP 2023)
The course follows a monthly subscription model where you have to make monthly payments of Rs 799/month.
What is the total duration of the course?
The total duration of the course is 7 months. So the total course fee becomes 799*7 = Rs 5600(approx.)
What is the syllabus of the mentorship program?
We will be covering the following modules:
Python Fundamentals
Python libraries for Data Science
Data Analysis
SQL for Data Science
Maths for Machine Learning
ML Algorithms
Practical ML
MLOPs
Case studies

Will Deep Learning and NLP be a part of this program?
No, NLP and Deep Learning both are not a part of this program's curriculum.
What if I miss a live session? Will I get a recording of the session?
Yes all our sessions are recorded, so even if you miss a session you can go back and watch the recording.
Where can I find the class schedule?
Checkout this google sheet to see month by month time table of the course -
What is the time duration of all the live sessions?
Roughly, all the sessions last 2 hours.
What is the language spoken by the instructor during the sessions?
Hinglish
How will I be informed about the upcoming class?
You will get a mail from our side before every paid session once you become a paid user.
Can I do this course if I am from a non-tech background?
Yes, absolutely.
I am late, can I join the program in the middle?
Absolutely, you can join the program anytime.
If I join/pay in the middle, will I be able to see all the past lectures?
Yes, once you make the payment you will be able to see all the past content in your dashboard.
Where do I have to submit the task?
You don't have to submit the task. We will provide you with the solutions, you have to self evaluate the task yourself.
Will we do case studies in the program?
Yes.
Where can we contact you?
You can mail us at muhammad.hamedoon@tech.uol.edu.pk
Payment/Registration related questions
Where do we have to make our payments? Your YouTube channel or website?
You have to make all your monthly payments on our website.
Unfortunately no, the program follows a monthly subscription model.
What is the validity of monthly subscription? Suppose if I pay on 15th Jan, then do I have to pay again on 1st Feb or 15th Feb
```

15th Feb. The validity period is 30 days from the day you make the payment. So essentially you can join anytime you don't have to wait for a month to end.
What if I don't like the course after making the payment. What is the refund policy?
You get a 7 days refund period from the day you have made the payment.
I am living outside India and I am not able to make the payment on the website, what should I do?
You have to contact us by sending a mail at muhammad.hamedoon@tech.uol.edu.pk
Post registration queries
Till when can I view the paid videos on the website?
This one is tricky, so read carefully. You can watch the videos till your subscription is valid. Suppose you have purchased subscription on 21st Jan, you will be able to watch all the past paid sessions in the period of 21st Jan to 20th Feb. But after 21st Feb you will have to purchase the subscription again.
But once the course is over and you have paid us Rs 5600(or 7 installments of Rs 799) you will be able to watch the paid sessions till Aug 2024.
Why lifetime validity is not provided?
Because of the low course fee.
Where can I reach out in case of a doubt after the session?
You will have to fill a google form provided in your dashboard and our team will contact you for a 1 on 1 doubt clearance session
If I join the program late, can I still ask past week doubts?
Yes, just select past week doubt in the doubt clearance google form.
I am living outside India and I am not able to make the payment on the website, what should I do?
You have to contact us by sending a mail at muhammad.hamedoon@tech.uol.edu.pk
Certificate and Placement Assistance related queries
What is the criteria to get the certificate?
There are 2 criterias:
You have to pay the entire fee of Rs 5600
You have to attempt all the course assessments.
I am joining late. How can I pay payment of the earlier months?
You will get a link to pay fee of earlier months in your dashboard once you pay for the current month.
I have read that Placement assistance is a part of this program. What comes under Placement assistance?
This is to clarify that Placement assistance does not mean Placement guarantee. So we dont guarantee you any jobs or for that matter even interview calls. So if you are planning to join this course just for placements, I am afraid you will be disappointed. Here is what comes under placement assistance
Portfolio Building sessions
Soft skill sessions
Sessions with industry mentors
Discussion on Job hunting strategies
"""

```
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer

# Tokenizer
tokenizer = Tokenizer()
tokenizer.fit_on_texts([faqs])

input_sequences = []
for sentence in faqs.split('\n'):
    tokenized_sentence = tokenizer.texts_to_sequences([sentence])[0]
```

```
  for i in range(1, len(tokenized_sentence)):
    input_sequences.append(tokenized_sentence[:i+1])

max_len = max([len(x) for x in input_sequences])

from tensorflow.keras.preprocessing.sequence import pad_sequences
padded_input_sequences = pad_sequences(input_sequences, maxlen=max_len,
padding='pre')

X = padded_input_sequences[:, :-1]
y = padded_input_sequences[:, -1]

from tensorflow.keras.utils import to_categorical
y = to_categorical(y, num_classes=len(tokenizer.word_index)+1)

# Model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense

vocab_size = len(tokenizer.word_index) + 1  # must be dynamic

model = Sequential()
model.add(Embedding(vocab_size, 100, input_length=max_len-1))
model.add(LSTM(150, return_sequences=True))
model.add(LSTM(150))
model.add(Dense(vocab_size, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# IMPORTANT FIX → Build model first
model.build(input_shape=(None, max_len-1))

# Corrected Summary
model.summary()

# Print actual trainable parameters (works in ALL TensorFlow versions)
print("\nTrainable parameters per layer:\n")
for layer in model.layers:
  print(layer.name, ":", layer.count_params())

print("\nTotal Trainable Parameters =", model.count_params())
```

```
model.fit(X,y,epochs=100)
```

```
import numpy as np
import time
from tensorflow.keras.preprocessing.sequence import pad_sequences

text = "what is the fee"
```

```
for i in range(10):
  # tokenize
  token_text = tokenizer.texts_to_sequences([text])[0]

  # padding
  padded_token_text = pad_sequences([token_text], maxlen=56, padding='pre')

  # predict
  pos = np.argmax(model.predict(padded_token_text, verbose=0))

  for word, index in tokenizer.word_index.items():
    if index == pos:
      text = text + " " + word
      print(text)
      time.sleep(2)
```

Output :

```
what is the fee of
what is the fee of monthly
what is the fee of monthly subscription
what is the fee of monthly subscription suppose
what is the fee of monthly subscription suppose if
what is the fee of monthly subscription suppose if i
what is the fee of monthly subscription suppose if i pay
what is the fee of monthly subscription suppose if i pay on
what is the fee of monthly subscription suppose if i pay on 15th
what is the fee of monthly subscription suppose if i pay on 15th jan
```

Question No. 2

Train an LSTM model on your chosen text dataset and write the code to predict the next word for a user-given input sentence.

Code:

```
import numpy as np

import tensorflow as tf

from tensorflow.keras.preprocessing.text import Tokenizer

from tensorflow.keras.preprocessing.sequence import pad_sequences

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Embedding, LSTM, Dense
```

```
faqs = """About the Program

What is the course fee for  Data Science Mentorship Program (DSMP 2023)

The course follows a monthly subscription model where you have to make monthly payments of Rs 799/month.What is the total duration of the course?... (use full text from your description) ...Discussion on Job hunting strategies"""

tokenizer = Tokenizer()
```

```
history = model.fit(X, y, epochs=100, verbose=1)
def predict_next_word(model, tokenizer, text_seq, max_len=max_len):
   tokenized = tokenizer.texts_to_sequences([text_seq])[0]
   padded = pad_sequences([tokenized], maxlen=max_len-1, padding='pre')
   predicted_index = np.argmax(model.predict(padded, verbose=0))
   for word, index in tokenizer.word_index.items():
      if index == predicted_index: return word
seed_text = "what is the fee"
next_word = predict_next_word(model, tokenizer, seed_text)
print(f"Next word prediction: '{next_word}'")
```

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 17, 100)	4,200
lstm_2 (LSTM)	(None, 17, 150)	150,600
lstm_3 (LSTM)	(None, 150)	180,600
dense_1 (Dense)	(None, 42)	6,342

```
Input: 'hello how' → Next Word Prediction: 'are'
```

Question No. 3

Modify the next-word prediction code so that the model generates 5 new words sequentially instead of just one.

```
seed_text = "what is the fee"
next_word = predict_next_word(model, tokenizer, seed_text)
print(f"Next word prediction: '{next_word}'")
def generate_words(model, tokenizer, seed_text, n_words=5,
max_len=max_len):
  text = seed_text
  for _ in range(n_words):
    word = predict_next_word(model, tokenizer, text, max_len)
    text += " " + word return text
generated_text = generate_words(model, tokenizer, seed_text, n_words=5)
print(f"\nGenerated sequence (5 words): {generated_text}")
```

```
Generated Text: hello how are you doing today about
```

LAB Assessment

Student Name		**LAB Rubrics**	CLO3 , P5, PLO5
		Total Marks	10
Registration No		**Obtained Marks**	
		Teacher Name	Dr. Syed M Hamedoon
Date		**Signature**	

Laboratory Work Assessment Rubrics

Sr. No.	Performance Indicator	Excellent (5)	Good (4)	Average (3)	Fair (2)	Poor (1)
1	**Theoretical knowledge 10%**	Student knows all the related concepts about the theoretical background of the experiment and rephrase those concepts in written and oral assessments	Student knows most of the related concepts about the theoretical background of the experiment and partially rephrase those concepts in written and oral assessments	Student knows few of the related concepts about the theoretical background of the experiment and partially rephrase those concepts in written and oral assessments	Student knows very little about the related concepts about the theoretical background of the experiment and poorly rephrase those concepts in written and oral assessments	Student has poor understanding of the related concepts about the theoretical background of the experiment and unable to rephrase those concepts in written and oral assessments
2	**Application Functionality 10%**	Application runs smoothly and operation of the application runs efficiently	Application compiles with no warnings. Robust operation of the application, with good recovery.	Application compiles with few or no warnings. Consideration given to unusual conditions with reasonable	Application compiles and runs without crashing. Some attempt at detecting and correcting errors.	Application does not compile or compiles but crashes. Confusing. Little or no error detection or correction.
3	**Specifications 10%**	The program works very efficiently and meets all of the required specifications.	The program works and meets some of the specifications.	The program works and produces the correct results and displays them correctly. It also meets most of the other specifications.	The program produces correct results but does not display them correctly.	The program is producing incorrect results.
4	**Level of understanding of the learned skill 10%**	Provide complete and logical answers based upon accurate technical content to the questions asked by examiner	Provide complete and logical answers based upon accurate technical content to the questions asked by examiner with few errors	Provide partially correct and logical answers based upon minimum technical content to the questions asked by examiner	Provide very few and illogical answers to the questions asked by examiner.	Provide no answer to the questions asked by examiner.
5	**Readability and Reusability 10%**	The code is exceptionally well organized and very easy to follow and reused	The code is fairly easy to read. The code could be reused as a whole or each class could be reused.	Most of the code could be reused in other programs.	Some parts of the code require change before they could be reused in other programs.	The code is poorly organized and very difficult to read and not organized for reusability.

6	**AI System Design 10%**	Well-designed AI models. Code is highly maintainable	Good designed AI models and Little code duplications	Some attempt to make AI models. Code can be maintained with significant effort	Little attempt to design AI models and less understanding of code	Very poor attempt to design AI models and its code
7	**Responsiveness to Questions/ Accuracy 10%**	1. Responds well, quick and very accurate all the time. 2. Effectively uses eye contact, speaks clearly, effectively and confidently using suitable volume	1. Generally Responsive and accurate most of the times. 2. Maintains eye contact, speaks clearly with suitable volume and pace.	1. Generally Responsive and accurate few times. 2. Some eye contact, speaks clearly and unclearly in different portions.	1. Not much Responsive and accurate most of the times. 2. Uses eye contact ineffectively and fails to speak clearly and audibly	. 1. Non Responsive and inaccurate all the times. 2. No eye contact and unable to speak 3. Dresses inappropriately
8	**Efficiency** **10%**	The code is extremely efficient without sacrificing readability and understanding	The code is fairly efficient without sacrificing readability and understanding	Some part of the code is efficient and other part of the code is not understandable and work properly	The code is brute force and unnecessarily long	The code is huge and appears to be patched together
9	**Delivery** **10%**	The program was delivered in time during lab.	The program was delivered in Lab before the end time.	The program was delivered within the due date.	The code was delivered within a day after the due date.	The code was delivered more than 2 days overdue.
10	**Awareness of Safety Guidelines 10%**	Student has sufficient knowledge of the laboratory safety SOPs and protocol and is fully compliant to the guidelines	Student has sufficient knowledge of the laboratory safety SOPs and protocol and is Partially compliant to the guidelines	Student has little knowledge of the laboratory safety SOPs and protocol and is Partially compliant to the guidelines	Student has little knowledge of the laboratory safety SOPs and protocol and is non-compliant to the guidelines	Student has no knowledge of the laboratory safety SOPs and protocol and is non-compliant to the guidelines