5: Continuous Bag of Words (CBOW) Model Implementation for text recognition # ---------- # a) Data preparation # b) Generate training data # c) Train model # d) Output # ----------

In [1]:
```
# a) Data Preparation

import numpy as np
import tensorflow as tf
from tensorflow.keras import layers, models, optimizers
print("Imports complete!\n")

# Demo text
raw_text = """Machine learning enables computers to discover patterns from data.
Neural networks are powerful models that learn representations automatically.
Natural language processing allows machines to understand human communication.
Deep learning methods are transforming artificial intelligence research and appl

vocab = sorted(set(raw_text))
vocab_size = len(vocab)
word_to_ix = {word: ix for ix, word in enumerate(vocab)}
ix_to_word = {ix: word for word, ix in word_to_ix.items()}

print("Vocabulary size:", vocab_size)
print("First 5 word-index pairs:", list(word_to_ix.items())[:5])


# b) Generate Training Data

CONTEXT_SIZE = 2         # Two words before, two after
EMBEDDING_DIM = 50       # Size of word embedding vectors

def make_context_vector(context, word_to_ix):
    return [word_to_ix[w] for w in context]

data = []
for i in range(CONTEXT_SIZE, len(raw_text) - CONTEXT_SIZE):
    context = [
        raw_text[i - 2], raw_text[i - 1],
        raw_text[i + 1], raw_text[i + 2]
    ]
    target = raw_text[i]
    data.append((context, target))

X = np.array([make_context_vector(ctx, word_to_ix) for ctx, _ in data])
y = np.array([word_to_ix[target] for _, target in data])

print("Shape of X (contexts):", X.shape)
print("Shape of y (targets):", y.shape)
print("Sample context/target:\n  Context:", [vocab[ix] for ix in X[0]], "\n  Tar

# c) Define CBOW model in Keras

inputs = layers.Input(shape=(4,), dtype="int32")  # 4 context words (indices)
embed = layers.Embedding(input_dim=vocab_size, output_dim=EMBEDDING_DIM)(inputs)
avg_embed = layers.Lambda(lambda x: tf.reduce_mean(x, axis=1))(embed)
dense1 = layers.Dense(128, activation="relu")(avg_embed)
outputs = layers.Dense(vocab_size, activation="softmax")(dense1)

cbow_model = models.Model(inputs=inputs, outputs=outputs)
```

```python
cbow_model.compile(
    optimizer=optimizers.Adam(learning_rate=0.01),
    loss="sparse_categorical_crossentropy",
    metrics=["accuracy"]
)
print("CBOW model summary:")
cbow_model.summary()

# d) Train the model

history = cbow_model.fit(
    X, y,
    epochs=100,
    batch_size=8,
    verbose=1
)
print("Training done!\n")

# e) Output: Test context for prediction

test_context = ['Neural', 'networks', 'that', 'learn']
test_input = np.array([make_context_vector(test_context, word_to_ix)])
pred_probs = cbow_model.predict(test_input)
pred_idx = np.argmax(pred_probs[0])
pred_word = ix_to_word[pred_idx]

print("\n-----------------------------------------")
print("Test Context Words:", test_context)
print("Predicted Target Word:", pred_word)
print("-----------------------------------------")
```

```
Imports complete!

Vocabulary size: 34
First 5 word-index pairs: [('Deep', 0), ('Machine', 1), ('Natural', 2), ('Neura
l', 3), ('allows', 4)]
Shape of X (contexts): (33, 4)
Shape of y (targets): (33,)
Sample context/target:
  Context: ['Machine', 'learning', 'computers', 'to']
  Target: enables
WARNING:tensorflow:From C:\Users\kusha\AppData\Local\Packages\PythonSoftwareFound
ation.Python.3.12_qbz5n2kfra8p0\LocalCache\local-packages\Python312\site-packages
\keras\src\backend\tensorflow\core.py:219: The name tf.placeholder is deprecated.
Please use tf.compat.v1.placeholder instead.

CBOW model summary:
```

Model: "functional"

Layer (type)	Output Shape	
input_layer (InputLayer)	(None, 4)	
embedding (Embedding)	(None, 4, 50)	
lambda (Lambda)	(None, 50)	
dense (Dense)	(None, 128)	
dense_1 (Dense)	(None, 34)	

Total params: 12,614 (49.27 KB)

Trainable params: 12,614 (49.27 KB)

Non-trainable params: 0 (0.00 B)

```
Epoch 1/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 5s 77ms/step - accuracy: 0.0000e+00 - loss: 3.5383
Epoch 2/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 67ms/step - accuracy: 0.2023 - loss: 3.4618
Epoch 3/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.4400 - loss: 3.3280
Epoch 4/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 42ms/step - accuracy: 0.3646 - loss: 3.0546
Epoch 5/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 68ms/step - accuracy: 0.3267 - loss: 2.6271
Epoch 6/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 81ms/step - accuracy: 0.4501 - loss: 2.0882
Epoch 7/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - accuracy: 0.6621 - loss: 1.5267
Epoch 8/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - accuracy: 0.6973 - loss: 1.1062
Epoch 9/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - accuracy: 0.9572 - loss: 0.6727
Epoch 10/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 86ms/step - accuracy: 1.0000 - loss: 0.3636
Epoch 11/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 81ms/step - accuracy: 0.9228 - loss: 0.3123
Epoch 12/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - accuracy: 0.9465 - loss: 0.2005
Epoch 13/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 43ms/step - accuracy: 0.9777 - loss: 0.0986
Epoch 14/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 51ms/step - accuracy: 1.0000 - loss: 0.0598
Epoch 15/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 98ms/step - accuracy: 1.0000 - loss: 0.0319
Epoch 16/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 100ms/step - accuracy: 1.0000 - loss: 0.0230
Epoch 17/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step - accuracy: 1.0000 - loss: 0.0195
Epoch 18/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 1.0000 - loss: 0.0155
Epoch 19/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 51ms/step - accuracy: 1.0000 - loss: 0.0136
Epoch 20/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step - accuracy: 1.0000 - loss: 0.0090
Epoch 21/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - accuracy: 1.0000 - loss: 0.0074
Epoch 22/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 1.0000 - loss: 0.0068
Epoch 23/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 38ms/step - accuracy: 1.0000 - loss: 0.0059
Epoch 24/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 46ms/step - accuracy: 1.0000 - loss: 0.0052
Epoch 25/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 85ms/step - accuracy: 1.0000 - loss: 0.0047
Epoch 26/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 94ms/step - accuracy: 1.0000 - loss: 0.0045
Epoch 27/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0039
Epoch 28/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.0034
Epoch 29/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 53ms/step - accuracy: 1.0000 - loss: 0.0033
Epoch 30/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 70ms/step - accuracy: 1.0000 - loss: 0.0032
```

```
Epoch 31/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 1.0000 - loss: 0.0029
Epoch 32/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step - accuracy: 1.0000 - loss: 0.0026
Epoch 33/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 1.0000 - loss: 0.0027
Epoch 34/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 52ms/step - accuracy: 1.0000 - loss: 0.0025
Epoch 35/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 75ms/step - accuracy: 1.0000 - loss: 0.0021
Epoch 36/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step - accuracy: 1.0000 - loss: 0.0022
Epoch 37/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - accuracy: 1.0000 - loss: 0.0020
Epoch 38/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 40ms/step - accuracy: 1.0000 - loss: 0.0020
Epoch 39/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 64ms/step - accuracy: 1.0000 - loss: 0.0019
Epoch 40/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step - accuracy: 1.0000 - loss: 0.0018
Epoch 41/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 1.0000 - loss: 0.0016
Epoch 42/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 1.0000 - loss: 0.0015
Epoch 43/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - accuracy: 1.0000 - loss: 0.0015
Epoch 44/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 59ms/step - accuracy: 1.0000 - loss: 0.0015
Epoch 45/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 84ms/step - accuracy: 1.0000 - loss: 0.0014
Epoch 46/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 106ms/step - accuracy: 1.0000 - loss: 0.0014
Epoch 47/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 86ms/step - accuracy: 1.0000 - loss: 0.0012
Epoch 48/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0012
Epoch 49/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - accuracy: 1.0000 - loss: 0.0011
Epoch 50/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - accuracy: 1.0000 - loss: 0.0011
Epoch 51/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step - accuracy: 1.0000 - loss: 0.0011
Epoch 52/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step - accuracy: 1.0000 - loss: 0.0010
Epoch 53/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - accuracy: 1.0000 - loss: 9.6663e-04
Epoch 54/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 8.9487e-04
Epoch 55/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 34ms/step - accuracy: 1.0000 - loss: 9.1118e-04
Epoch 56/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 84ms/step - accuracy: 1.0000 - loss: 8.7428e-04
Epoch 57/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 7.8403e-04
Epoch 58/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 1.0000 - loss: 7.7686e-04
Epoch 59/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 66ms/step - accuracy: 1.0000 - loss: 7.6784e-04
Epoch 60/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 1.0000 - loss: 7.6234e-04
```

```
Epoch 61/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - accuracy: 1.0000 - loss: 6.9030e-04
Epoch 62/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - accuracy: 1.0000 - loss: 6.8366e-04
Epoch 63/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - accuracy: 1.0000 - loss: 6.7625e-04
Epoch 64/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 98ms/step - accuracy: 1.0000 - loss: 6.4066e-04
Epoch 65/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step - accuracy: 1.0000 - loss: 6.3340e-04
Epoch 66/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 6.3913e-04
Epoch 67/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 45ms/step - accuracy: 1.0000 - loss: 5.8769e-04
Epoch 68/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 73ms/step - accuracy: 1.0000 - loss: 5.5614e-04
Epoch 69/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 87ms/step - accuracy: 1.0000 - loss: 5.7267e-04
Epoch 70/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - accuracy: 1.0000 - loss: 5.1177e-04
Epoch 71/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - accuracy: 1.0000 - loss: 5.1575e-04
Epoch 72/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 60ms/step - accuracy: 1.0000 - loss: 5.3091e-04
Epoch 73/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 87ms/step - accuracy: 1.0000 - loss: 4.9107e-04
Epoch 74/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 4.8441e-04
Epoch 75/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 106ms/step - accuracy: 1.0000 - loss: 4.7350e-04
Epoch 76/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 1.0000 - loss: 4.6545e-04
Epoch 77/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - accuracy: 1.0000 - loss: 4.5901e-04
Epoch 78/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 86ms/step - accuracy: 1.0000 - loss: 4.5510e-04
Epoch 79/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 121ms/step - accuracy: 1.0000 - loss: 3.9758e-04
Epoch 80/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 80ms/step - accuracy: 1.0000 - loss: 3.8875e-04
Epoch 81/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 38ms/step - accuracy: 1.0000 - loss: 4.2464e-04
Epoch 82/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - accuracy: 1.0000 - loss: 3.8015e-04
Epoch 83/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - accuracy: 1.0000 - loss: 3.8684e-04
Epoch 84/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 1.0000 - loss: 3.6594e-04
Epoch 85/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 60ms/step - accuracy: 1.0000 - loss: 3.4954e-04
Epoch 86/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 130ms/step - accuracy: 1.0000 - loss: 3.3989e-04
Epoch 87/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 86ms/step - accuracy: 1.0000 - loss: 3.4391e-04
Epoch 88/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - accuracy: 1.0000 - loss: 3.5624e-04
Epoch 89/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 73ms/step - accuracy: 1.0000 - loss: 3.2252e-04
Epoch 90/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 1.0000 - loss: 3.0915e-04
```

```
Epoch 91/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 66ms/step - accuracy: 1.0000 - loss: 3.1307e-04
Epoch 92/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 82ms/step - accuracy: 1.0000 - loss: 3.0957e-04
Epoch 93/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - accuracy: 1.0000 - loss: 2.9193e-04
Epoch 94/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - accuracy: 1.0000 - loss: 2.7637e-04
Epoch 95/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 51ms/step - accuracy: 1.0000 - loss: 2.7593e-04
Epoch 96/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 100ms/step - accuracy: 1.0000 - loss: 2.7663e-04
Epoch 97/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 2.7736e-04
Epoch 98/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 35ms/step - accuracy: 1.0000 - loss: 2.8422e-04
Epoch 99/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 62ms/step - accuracy: 1.0000 - loss: 2.7617e-04
Epoch 100/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step - accuracy: 1.0000 - loss: 2.4493e-04
Training done!

1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 569ms/step

--------------------------------------------
Test Context Words: ['Neural', 'networks', 'that', 'learn']
Predicted Target Word: powerful
--------------------------------------------
```

In [ ]: