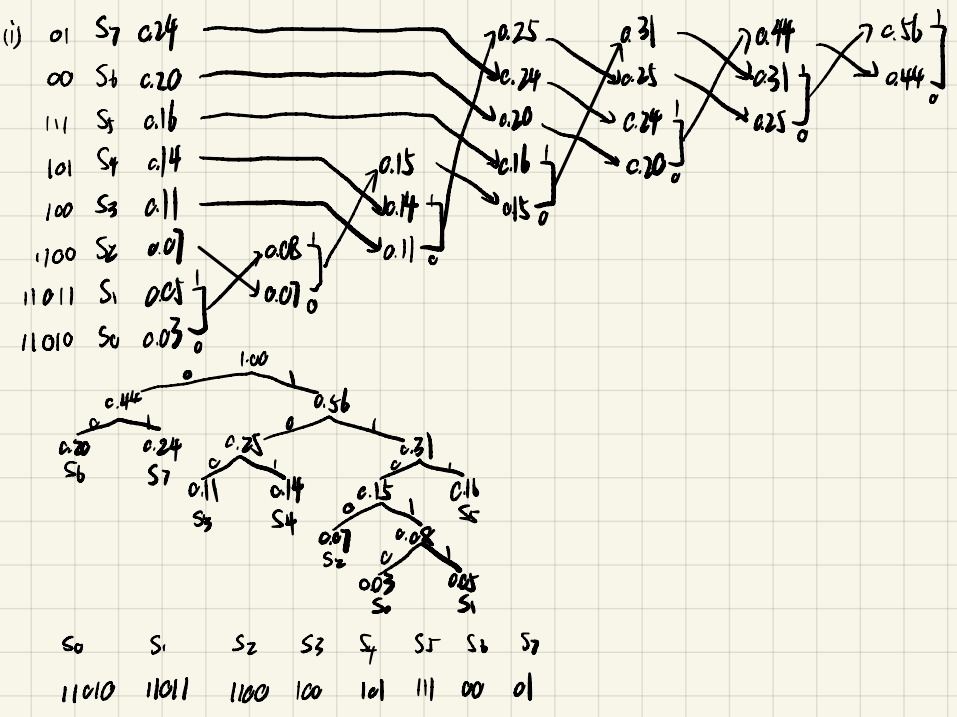
EE6427 24S2

1. (a) In a compression scheme, a data source consists of eight symbols, with the probability distribution is given in Table 1.

Table 1

Symbol	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
Probability of occurrence	0.03	0.05	0.07	0.11	0.14	0.16	0.20	0.24

(i) Design a suitable set of Huffman codewords for the eight symbols. Clearly show all the key steps and calculations.



S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
11010	11011	1100	100	101	111	00	01

(ii) Consider a data stream X consisting of 4000 symbols with probability distribution given in Table 1. The data stream is to be compressed using the Huffman codewords designed in part (i). Find the expected (average) storage requirement of data stream X in bytes.

(ii) average number of bits/symbol
$= (0.03+0.05)\times 5 + 0.07\times 4 + (0.11+0.14+0.16)\times 3 + (0.2+0.24)\times 2$
$= 2.79$ bits/symbol
expected storage requirement in bits
$= 2.79 \times 4000$
$= 11160$ bits
in bytes
$= 11160 \div 8 = 1395$ bytes

(iii) A student would like to compress a particular data sequence consisting of 10 symbols. Each symbol in the sequence can be any of the eight symbols given in Table 1. The student uses two different schemes to represent the sequence.

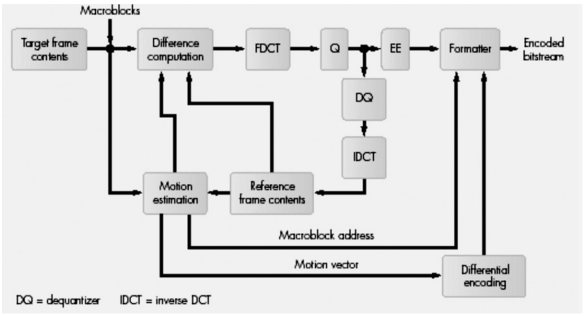
Uncompressed scheme: 4 bits is used to represent each symbol.
Compressed scheme: Huffman codewords designed in part (i) are used to compress the symbols.

Find the lowest possible compression ratio that can be obtained for the sequence.

(iii) $4 \div 2.79 \approx 1.43$?
entropy
$= -\sum_i p_i \log_2 p_i$
$= -0.03\log_2 0.03 - 0.05\log_2 0.05 - 0.07\log_2 0.07 - 0.11\log_2 0.11 - 0.14\log_2 0.14$
$-0.16\log_2 0.16 - 0.2\log_2 0.2 - 0.24\log_2 0.24$
≈ 2.7654
lowest compression ratio
$= 4 \div 2.7654$
≈ 1.446

(b) Briefly explain how Huffman coding is used in the P-frame encoding of the MPEG-1 standard.

MPEG-1: P-Frame Encoding Flowchart



2. (a) In a motion estimation and compensation scheme for video compression, a 4×4 pixel block is shown in Figure 1 and its co-located block in the reference frame is shown by the bounding box in Figure 2. The search window is within ±1 pixels. The metric used for motion estimation is sum of absolute errors.

10	10	10	10
20	20	20	20
30	30	30	30
40	40	40	40

Figure 1

0	0	0	0	0	0
10	10	10	10	10	10
10	10	10	10	10	10
18	20	20	20	20	15
34	30	30	30	32	26
36	40	40	38	40	42

ref

Figure 2

(i) Find the motion vector (M_u, M_v) within the search window, where M_u is the row displacement (positive direction is pointing downwards) and M_v is the column displacement (positive direction is pointing to the right).

2.(a).(i) $(M_u, M_v) = (1, 0)$

(ii) Find the best matched block in the reference frame and write down the prediction error $E(i, j)$.

2.(a).(ii) best matched block $= \begin{bmatrix} 10 & 10 & 10 & 10 \\ 20 & 20 & 20 & 20 \\ 30 & 30 & 30 & 32 \\ 40 & 40 & 38 & 40 \end{bmatrix}$

$E(i,j) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & 2 & 0 \end{bmatrix}$

(iii) The prediction error $E(i, j)$ undergoes the Integer Transform with transform matrix given by:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}.$$

Find the Integer Transform of the prediction error $E(i, j)$.

(17 Marks)

2.(a).(iii) $F = H \cdot E \cdot H^T$

$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & 2 & 0 \end{bmatrix} \cdot H^T$

$= \begin{bmatrix} 0 & 0 & 2 & -2 \\ 0 & 0 & -4 & 2 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & -2 & -4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & 2 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & -4 & 6 \\ -2 & 0 & 6 & -10 \\ 4 & -6 & 0 & 2 \\ -6 & 10 & -2 & 0 \end{bmatrix}$

(b) Briefly explain the effect(s) of increasing the Quantization Parameter (QP) on the video bitrate and video quality in the H.264 video compression.

(8 Marks)

In H.264, increasing the QP increases the quantization step size, so more transform coefficients are heavily quantized or become zero. This reduces the number of bits needed and lowers the video bitrate but it also increases distortion and compression artifacts, so the video quality decreases.

(iii) The prediction error $E(i, j)$ undergoes the Integer Transform with transform matrix given by:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}.$$

Find the Integer Transform of the prediction error $E(i, j)$.

(17 Marks)

(b) Briefly explain the effect(s) of increasing the Quantization Parameter (QP) on the video bitrate and video quality in the H.264 video compression.

(8 Marks)

3. (a) A Long Short-Term Memory (LSTM) network has the following settings:

Initial hidden state, $\mathbf{h}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, Initial cell state, $\mathbf{c}_0 = \begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix}$,

Input gate weight matrix, $\mathbf{W}_i = \begin{bmatrix} \mathbf{W}_{hi} & \mathbf{W}_{xi} \end{bmatrix} = \begin{bmatrix} 0.4 & 0.3 & 0.5 & 0.4 \\ 0.2 & 0.1 & 0.3 & 0.2 \end{bmatrix}$,

Gate gate (also known as candidate gate) at timestep $t = 1$, $\mathbf{g}_1 = \begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix}$,

Forget gate at timestep $t = 1$, $\mathbf{f}_1 = \begin{bmatrix} 0.8 \\ 0.9 \end{bmatrix}$,

Output gate at timestep $t = 1$, $\mathbf{o}_1 = \begin{bmatrix} 0.7 \\ 0.8 \end{bmatrix}$,

A 2-timestep input is given by $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix}$ where $\mathbf{x}_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ and $\mathbf{x}_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

Assume no bias is used in the computation of the LSTM. The sigmoid and tanh functions are given as follows.

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

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Note: Question No. 3 continues on page 4.

3. (a) A Long Short-Term Memory (LSTM) network has the following settings:

Initial hidden state, $\mathbf{h}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, Initial cell state, $\mathbf{c}_0 = \begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix}$,

Input gate weight matrix, $\mathbf{W}_i = [\mathbf{W}_{hi} \quad \mathbf{W}_{xi}] = \begin{bmatrix} 0.4 & 0.3 & 0.5 & 0.4 \\ 0.2 & 0.1 & 0.3 & 0.2 \end{bmatrix}$,

Gate gate (also known as candidate gate) at timestep $t=1$, $\mathbf{g}_1 = \begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix}$,

Forget gate at timestep $t=1$, $\mathbf{f}_1 = \begin{bmatrix} 0.8 \\ 0.9 \end{bmatrix}$,

Output gate at timestep $t=1$, $\mathbf{o}_1 = \begin{bmatrix} 0.7 \\ 0.8 \end{bmatrix}$,

A 2-timestep input is given by $\mathbf{x} = [\mathbf{x}_1 \quad \mathbf{x}_2]$ where $\mathbf{x}_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ and $\mathbf{x}_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

Assume no bias is used in the computation of the LSTM. The sigmoid and tanh functions are given as follows.

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

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

(i) Find the Input gate $\mathbf{i}_1$ at timestep $t=1$.

(ii) Find the cell state $\mathbf{c}_1$ at timestep $t=1$.

(iii) Find the Input gate $\mathbf{i}_2$ at timestep $t=2$. Round your result to 3 decimal places.

3.(a).(i) $i_t = \sigma(W_i \cdot \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}) = \sigma(W_{hi} \cdot h_{t-1} + W_{xi} \cdot x_t)$

$\therefore i_1 = \sigma(W_{hi} \cdot h_0 + W_{xi} \cdot x_1)$

$= \sigma\left(\begin{bmatrix} 0.4 & 0.3 \\ 0.2 & 0.1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.5 & 0.4 \\ 0.3 & 0.2 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix}\right)$

$= \sigma\left(\begin{bmatrix} 1.9 \\ 1.1 \end{bmatrix}\right) \approx \begin{bmatrix} 0.8699 \\ 0.7503 \end{bmatrix}$

3.(a).(ii) $C_t = f \odot C_{t-1} + i \odot g$

$\therefore C_1 = f_1 \odot C_0 + i_1 \odot g_1$

$= \begin{bmatrix} 0.8 \\ 0.9 \end{bmatrix} \odot \begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix} + \begin{bmatrix} 0.8699 \\ 0.7503 \end{bmatrix} \odot \begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix}$

$= \begin{bmatrix} 0.33398 \\ 0.58503 \end{bmatrix}$

3.(a).(iii) $h_t = o \odot \tanh(C_t)$

$\therefore h_1 = o_1 \odot \tanh(C_1)$

$= \begin{bmatrix} 0.7 \\ 0.8 \end{bmatrix} \odot \tanh\left(\begin{bmatrix} 0.33398 \\ 0.58503 \end{bmatrix}\right)$

$\approx \begin{bmatrix} 0.7 \\ 0.8 \end{bmatrix} \odot \begin{bmatrix} 0.3221 \\ 0.5264 \end{bmatrix}$

$= \begin{bmatrix} 0.22547 \\ 0.42112 \end{bmatrix}$

$i_2 = \sigma(W_{hi} \cdot h_1 + W_{xi} \cdot x_2)$

$= \sigma\left(\begin{bmatrix} 0.4 & 0.3 \\ 0.2 & 0.1 \end{bmatrix} \begin{bmatrix} 0.22547 \\ 0.42112 \end{bmatrix} + \begin{bmatrix} 0.5 & 0.4 \\ 0.3 & 0.2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$

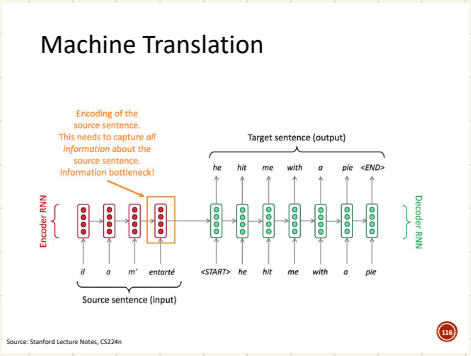
$= \sigma\left(\begin{bmatrix} 1.51624 \\ 0.78726 \end{bmatrix}\right)$

$\approx \begin{bmatrix} 0.820 \\ 0.687 \end{bmatrix}$

(b) A user would like to develop a video clip content summarization application that takes in a video clip and generates a text sentence to describe the video content. The user is planning to use LSTM(s) to achieve the goal. The user has extracted the feature embedding for each frame of the video clips. Briefly list two main steps of how to use LSTM(s) to address this application.

3.(b) Step 1 – Encoder LSTM:
Feed the frame feature sequence into an encoder LSTM, which processes the input sequentially and compresses it into a content vector (the final hidden state).

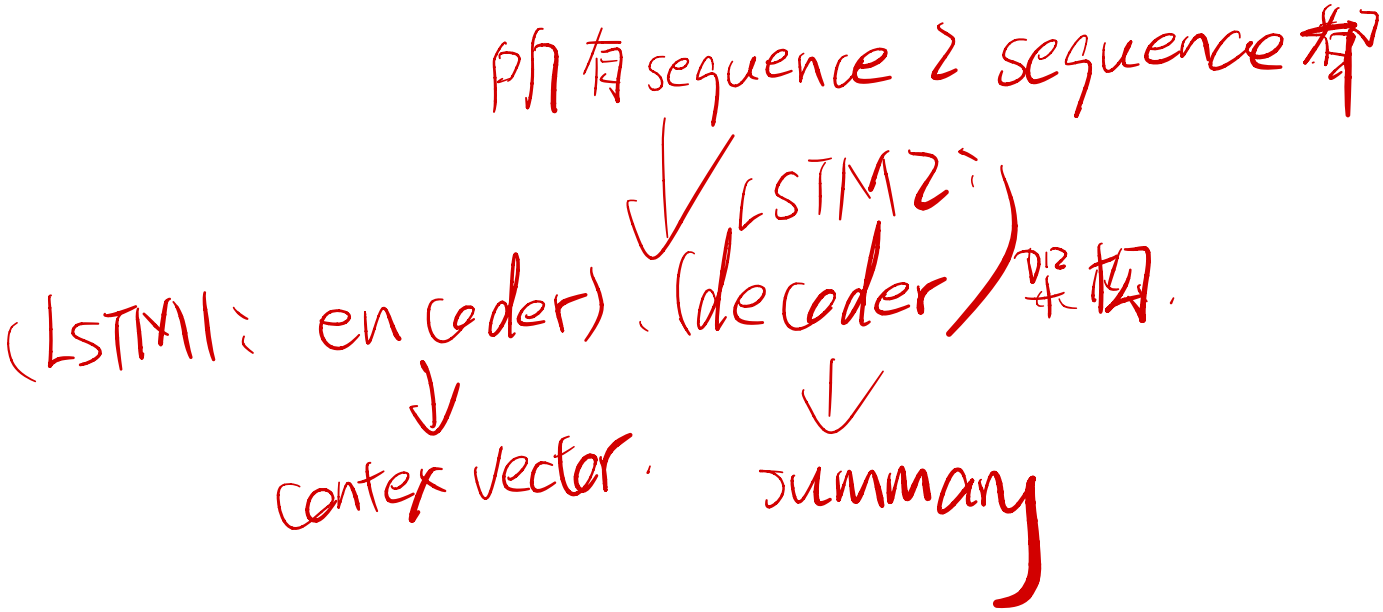
Step 2 – Decoder LSTM:
Use a decoder LSTM that receives the context vector and generates the output sequentially, predicting one word at each timestep until the end token.



Problem type: This is a **sequence-to-sequence (seq2seq)** task: the input is a temporal sequence of frame feature vectors and the output is a sequence of words (a text sentence) describing the clip.

Step 1 — Encoder (video → latent sequence / context):
Use an LSTM encoder to ingest the per-frame feature embeddings in chronological order. The encoder processes the sequence $x_1, \ldots, x_T$ (each x_t is the feature vector for frame t) and produces either (a) a final hidden state vector (a fixed-length context vector) or (b) a sequence of hidden states $\{h_t\}$ that summarize each time step. This encoded representation captures the temporal information and visual content of the clip and will be passed to the decoder.

Step 2 — Decoder (latent → text):
Use a second LSTM as a decoder to generate the output sentence one token at a time. At each decoding step the decoder takes the previous word (during training use teacher forcing) and the encoder context (the final encoder state or an attention-weighted combination of encoder hidden states) and outputs a probability distribution over the vocabulary; the model is trained with cross-entropy loss to predict the ground-truth next word. At inference, generate tokens sequentially (e.g. greedy or beam search) until an end-of-sentence token.


的有sequence 2 sequence架
(LSTM2:
(LSTM1: encoder).
decoder)架构.
context vector.
summary