RV UNIVERSITY Go, change the world an initiative of RV EDUCATIONAL INSTITUTIONS	USN													

RV University, Bengaluru

School of Computer Science and Engineering

B.Tech (Hons.)

CP-2 Question Paper – Set1 with Answer keys

Academic Year 2024-2025

Course: IoT and Edge Computing		**Course Code: CS3100**	**Semester: V**
Date: Oct 2024	**Duration: 90 minutes**	**Max Marks: 30 (weightage 25)**	

Sl. No.	Questions	Marks	L1-L6	CO
1.	For the below given code, write the shape of the array arr1 = np.array([[[1, 2, 3, 5, 8]]]) arr2 = np.array([[1,2]]) print("arr1 shape ", arr1.shape) print("arr2 shape ", arr2.shape)	**2**	**L2**	**CO2**
2.	What will be the contents of row and column variables? a = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]) row = np.array([0,1,2]) column = np.array([1,0,2]) result = a[row, column] print(result)	**2**	**L3**	**CO2**
3.	What will be printed by the code given below? npIntStrDef = np.dtype([('integer', 'i4'), ('string', 'S5')]) npData = np.array([(1, b'hello')], dtype=npIntStrDef) print('Item Size: ', npData.nbytes) print('Value: ', npData)	**2**	**L4**	**CO2**
4.	What is the output of this program and values in the result array? A = np.ones((3, 4, 1)) B = np.zeros((1, 4, 5)) result = A + B print("result shape: ", result.shape) print("result dim: ", result.ndim)	**2**	**L3**	**CO2**

5.	L1 = [1, 4, 5, "Hi", 5.0, 7]
	L2 = [6.0, 7.0, 'Hello', 8.0]
	
	for elem in L1:
	  for x in L2:
	  if type(elem) == type(x):
	  print(elem, x)
	What will be printed by this program?
	Note: Order of the output also matters.
		2	**L3**	**CO2**

L1- Remember, L2- Understand, L3 – Apply, L4- Analyse, L5 – Evaluate, L6 – Create

6.	Assume that you have task1 and task2 that are expected to run most of the time in your application, which is ported to ESP32. How do you create these tasks using FreeRTOS APIs that would make your application run efficiently? Note: You don't have to give all the parameters of the API invocation. You need to explain which one of the APIs you would use and why?	**2**	**L4**	**CO3**
7.	Give the IoT application scenarios where you would choose NB-IoT over other wireless technologies based on its features.	**2**	**L4**	**CO1**
8.	What are the four important characteristics of WSN? Give a short description on each.	**2**	**L2**	**CO1**

9.	void myFn_char(void) {
	char Arr[] = "abcdefgh";
	char *cp = (char *) Arr;
	
	*(cp + 2) = *(cp + 5);
	cp++;
	*(cp + 2) = *(cp + 5);
	
	printf("Arr: %s\n", Arr);
	}
	What will be printed by the above function?
		2	**L3**	**CO2**

10	In the given C program below, what will be printed? Assume that this is running on a 32-bit system.
	signed short sShortVal = 0xA234;
	signed int sIntVal = sShortVal;
	unsigned short uShortVal = 0xA234;
	unsigned int uIntVal = uShortVal;
	printf("sIntVal = %X, uIntVal = %X\n", sIntVal, uIntVal);
		2	**L4**	**CO3**

11	void inferenceEngine(void) {
	float position = static_cast<float>(inference_count) /
	static_cast<float>(kInferencesPerCycle);
	x_val = position * kXrange;
	y_vals.y_actual = sin(x_val);
	y_vals.y_pred = predict(x_val);
	bool res = xQueueSend(xQHandle, &y_vals,pdMS_TO_TICKS(2000));
	if (!res) {
	printf("MsgQ: Failed to send an item.\n");
	}
	inference_count += 1;
	if (inference_count >= kInferencesPerCycle)
	inference_count = 0;
	}
	i) Why should inference_count be reset?
	ii) What happens if the queue(bool res) is full?
		2	**L2**	**CO3**

12	Explain how different packages such as Keras, TensorFlow and NumPy packages are integrated within the Python environment to enable ML model development.	**2**	**L2**	**CO2**

Part B

Instructions: Answer any two questions. The best two answers will be considered for evaluation.

Sl. No.	Questions	Marks	L1-L6	CO
13.	a) Explain the steps involved in writing and converting a complex wave generation model developed in Python to be run on ESP32 only for inferencing. – 2 marks b) Mention couple of ways you would exploit the features of ESP32 to run the inference engine efficiently on ESP32. – 1 mark Note: Assume TensorFlow and keras are used in the model implementation.	**3**	**L4**	**CO3**
14.	a) Write a note on the rules governing the broadcasting feature of NumPy that works on multi-dimensional arrays of NumPy. – 2 marks b) Give couple of examples explaining the rules – 1 mark	**3**	**L3**	**CO2**
15.	Draw the layered architecture of IoT and give a short description of each of the layers mentioning their responsibilities.	**3**	**L2**	**CO1**

Course Outcomes

1. Choose a suitable wireless protocol based on the problem domain of an IoT product
2. Understand NumPy, TensorFlow framework and the need for TFLite for Edge Devices
3. Identify an IoT solution based on the features supported by ESP32 and FreeRTOS
4. Demonstrate integration of cloud computing platforms with ESP32

Marks Distribution									
L1	L2	L3	L4	L5	L6	CO1	CO2	CO3	CO4
0	11	13	11	0	0	7	17	9	0

L1- Remember, L2- Understand, L3 – Apply, L4- Analyse, L5 – Evaluate, L6 – Create

Answers
Part A

Q1 i) arr1 shape (1, 1, 5)
ii) arr2 shape (1, 2)

Q2.
Row = [0 1 2] - 0.5 marks
Column = [1 0 2] - 0.5 marks
result = [2,4,9] - 1 mark

Q3. i) Item Size: 9
ii) Value: [(1, b'hello')]

Q4.
result shape: (3, 4, 5) – 1 mark
result dim: 3 – 0.5 marks
All the elements in the result array will be filled with 1.0s. – 0.5 marks
Note: Even if 1.0 is not mentioned, instead 1 is given, no marks will be reduced.

Q5. Output is:
Hi Hello
5.0 6.0
5.0 7.0
5.0 8.0

Q6. The API that should be used is xTaskCreatePinnedToCore() for creating both task1 and task2. – 1 Mark

They need to be scheduled on two different cores, coreID given for task1 and task2 need to be different from each other, to use both the cores available in ESP32, since these two are the tasks which are going to be running most of the time in the application, the performance will be better because they run in parallel on both the cores. – 1 Mark.

Q7. NBIoT: Applications that need the following can use this technology
- Long range (< 10 km)
- Lower power of operation
- Applications where device density is higher
- Where there is a requirement of sending data directly to cloud
- Where device complexity required is less and lower data rate is sufficient

Note: Listing the applications as per Session 2A is optional. If any of the four features above are mentioned full marks can be given. Exact data rate or device density values are not expected.

Q8. WSN characteristics: Session 2C
WSN: Wireless Sensor Networks.
The four characteristics are: Redundancy Exploitation, Data-centric Routing,
Data Aggregation and Localized Algorithms. Write a single line note on each**.**

Q9. Arr: abfgefgh - No partial marks.

Q10. sIntVal = FFFFA234, uIntVal = A234

Q11.)i) The system is designed to process data in cycles. Each cycle consists of kInferencesPerCycle samples. When inference_count reaches kInferencesPerCycle, it means one full cycle has been completed, and the system should start the next cycle from the beginning.
ii) When queue is full it prints error message after waiting for 2 seconds.

Q12. **N**umPy is the fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficient. The NumPy library is implemented in C for efficiency and performance.
TensorFlow and Keras are built on top of NumPy for handling numerical data, such as training data (input features), model weights, and operations like matrix multiplication.
Keras APIs provides user friendly interface for building and training machine learning models. TensorFlow. – **Marks will be awarded based on the contents, though not all the above is expected.**

Part B

Q13. The following steps are to be done to train a model and run the inference engine on ESP32.

1. Generate samples of a complex waveform (**$y = 5\ Sin^2(x) + 4\ cos(x)$**) from 0 to 2pi radians.
   Note: The same waveform equation is not a must though this was used in the Lab 6.
2. Randomise the samples using **np.random.uniform** function
3. Split the samples for training and validation
4. Using keras.Sequential add couple of layers with 16 or more nodes on each
5. Using suitable gradient descent algorithm and loss function train the model
6. Generate floating point arrays of all the weights and biases of the model as C header file
7. Write a C code that reads from this header based on number nodes in each layer and process the inputs given to it, for generating the output sine wave on ESP32
8. Use Serial Plotter on ESP32 for displaying the sine wave.
9. The plotting and inferencing tasks can be run on two different cores on ESP32.
10. Need to take care of size of weights and biases do not exceed the maximum memory available on ESP32.
    Note: Even if all the above steps are not given marks can be awarded if the important logical steps are covered.

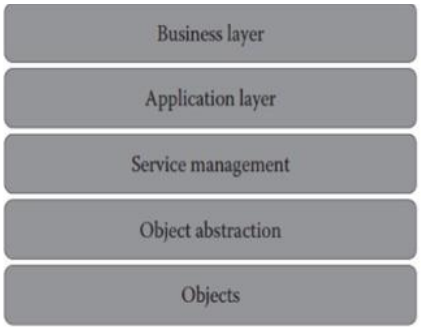
Q14. Element-wise operations on arrays are only valid when the arrays' shapes are either equal or compatible. The equal shapes case is trivial, it includes the stretched array from scalar example.
To determine if two shapes are compatible, NumPy compares their dimensions, starting with the trailing ones and working its way backwards. For example: arrExample[4][5][7][2] from {2} towards [7]
If two dimensions are equal, or if one of them equals 1, the comparison continues. Otherwise, you'll see a ValueError raised.

```
(4, 3)                  (4, 3)
        == padding ==>          == result ==> (4, 3)
(3,)                    (1, 3)
```

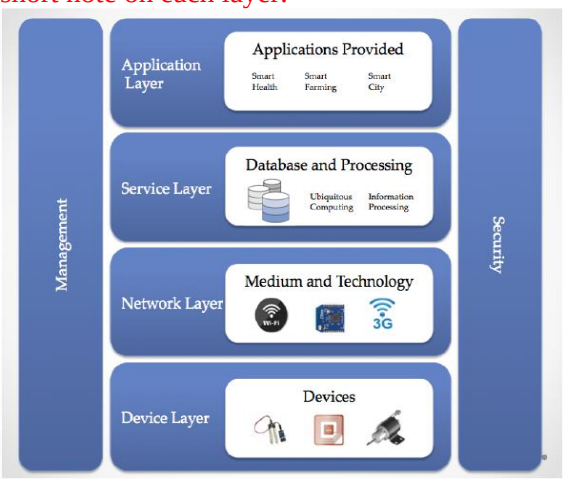
Here's another example, broadcasting between a 3-D and a 1-D array:

```
(3,)                        (1, 1, 3)
           == padding ==>               == result ==> (5, 4, 3)
(5, 4, 3)                   (5, 4, 3)
```

Q15.



Short description about each layer and its responsibilities. You can also draw the below diagram with a short note on each layer.



2. Artificial Intelligence (AI) and Machine Learning (ML) can significantly enhance a fan control system in a smart home by making it more accurate and responsive to varying conditions. like Data-Driven Decision Making, Personalized Control, Improved Accuracy and Efficiency

****