RV University

School of Computer Science Engineering

B.Tech Degree Examination – With answer keys - Updated

Semester : V

Course Code : CS3100

Course Title : IoT and Edge Computing

Duration : 2 Hours   **Max. Marks: 30**

Part A: Attempt all the questions in this section. All the questions carry 2 marks each.

Sl. No.	Part A: Questions	Marks	L1-L6	CO
1.	Give the output printed by the code given below. arr1 = np.array([[[[2, 3, 5, 8]]]]) arr2 = np.array([1,2]) print("arr1 shape ", arr1.shape) print("arr2 shape ", arr2.shape)	2	L2	2
2.	What is the output of this program and values in the result array? A = np.ones((3, 6, 1)) B = np.zeros((1, 1, 5)) result = A * B print("result shape: ", result.shape) print("result dim: ", result.ndim)	2	L4	2
3.	Give the IoT application scenarios where you would choose LTE-M over other wireless technologies, based on its features.	2	L4	1

4.	Give the output printed by the code given below. Assume that this is running on a 32-bit system.
	signed short sShortVal = 0xB567;
	signed int sIntVal = sShortVal;
	unsigned short uShortVal = 0xB567;
	unsigned int uIntVal = uShortVal;
	printf("sIntVal = %X, uIntVal = %X\n", sIntVal, uIntVal);
		2	L4	3

5.	What is eager execution? Quote any two advantages of eager execution in Machine learning.	2	L2	3
6.	What are trainable parameters in a neural network? - **1 Mark** Consider a three-layer neural network model with all of them being dense connections, the number of nodes in each layer are as shown below: **Layer 1 (input): 6 neurons** **Layer 2: 4 neurons** **Layer 3 (output): 2 neurons** What is the total number of trainable parameters here? – **1 Mark** **Note**: No marks will be awarded if the steps are not shown.	2	L3	2
7.	Does the **overfitting** issue in a neural network get resolved by **decreasing** the number of weights and biases? Justify your answer.	2	L3	2

Sl. No.	Part A: Questions	Marks	L1-L6	CO
8.	What is the rank of each of the following tensor types? a) A scalar b) A vector c) Matrix M below $M = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	2	L2	2
9.	Why is **TensorFlow Lite** suitable for running machine learning models on mobile and embedded devices? - **1Mark** Give two differences between the original model and its TFLite version.	2	L2	3
10	What are the memory locations accessed when these instructions execute on ESP32? Is this a data read or write operation? **int *ip = 0x3FFF0004; int val = *ip;**	2	L4	3
11	What is **one-hot encoding**? Explain it with one sample input parameter of car evaluation model or any other model of your choice.	2	L2	2
12	What is **AI Hub** of **Qualcomm**? Where will you be able to use it in your Machine Learning model development project?	2	L2	5

Part B

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

Sl. No.	Questions	Marks	L1-L6	CO
13.	1. Explain the steps involved in writing and converting a sine wave generation model developed in Python to be run on ESP32 only for inferencing. 2. Mention couple of ways you would exploit the features of ESP32 to run the inference engine efficiently on ESP32. Note: Assume TensorFlow and keras are used in the model implementation.	3	L4	3
14.	Briefly explain the functioning of MQTT protocol and some of its benefits.	3	L2	4

15.	What do **vTaskSuspend(taskHandle)** and **vTaskResume(taskHandle)** API do?
	 What will be the state of the **task1** process when the below function is executed by another task in the system, on ESP32?
	 myFn(void){
	   vTaskSuspend(task1Handle);
	   printf("State of task1 %d\n", eTaskGetState(task1Handle));
	   delay(1000);
	   vTaskResume(task1Handle);
	   printf("State of task1 %d\n", eTaskGetState(task1Handle)); }
		3	L3	3

Question No		Marks
Q1.	**arr1 shape (1, 1, 1, 4)** **arr2 shape (2,)**	**2**
Q2.	**result shape: (3, 6, 5)** **result dim: 3**	**2**
Q3.	**LTE-M** - **LTE-M uses the existing LTE voice network for communicating data generated by IoT devices.** - **Low power M2M communications.** - **Longer battery life of devices (10 years are more)** - **It can co-exist with other Telecom technologies 2G/3G/4G** - **It has security and privacy features** - **Higher data rate (around 1Mbps)** - **Medium latency of a few 10s of milliseconds.** **Note; Any four points are sufficient to get full marks.**	**2**
Q4.	**sIntVal = FFFFB567, uIntVal = B567**	**2**
Q5.	**Eager execution** is a mode of execution in machine learning frameworks, like TensorFlow 2.0, where operations are evaluated immediately as they are called, rather than being deferred until later, as in the traditional static graph execution mode which was followed in Tensor Flow 1.0. – **1.0 mark** In this mode, as the instructions are executed the results are output, without the need of defining the Data Flow Graph prior to running the ML models. **This is very advantages from the point of view of the following:** **D**eveloping ML models, debugging them, it is similar to the way Python Interpreter works, understanding error messages are easier. – **Any two pints – 1.0 mark.**	**2**
Q6.	**Trainable parameters** in a neural network are the **weights** and **biases** that the model learns or updated during the training process – **1 mark** **Layer 1 (input): 6 neurons, Layer 2: 4 neurons and Layer 3 (output): 2 neurons.** No. of **weights** and **biases** in the 1st layer: **6 + 6 = 12** No. of **weights** between the first and the second layer = **6×4 = 24** and **Biases** in the 2nd layer = **4** Total weights and biases = **24 + 4 = 28** Number of **weights** between the second and third layer = **4×2 = 8** and **Biases** in the 3rd layer = **2** Total weights and biases = **8 + 2 = 10** Total weights and biases or trainable parameters = 12 + **28 + 10 = 50 – 1 Mark** **Note: Only when the steps and the final answers are correct full mark is to be given.** **Note**: The above answer is assuming a single input, feeding into 6 neurons in the 1st layer. If you have misunderstood the 1st layer of 6 neurons as input layer, the answer is**: 6*4 + 4 + 4*2 + 2 = 38** trainable parameters is also considered to be correct.	**2**

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

Question No		Marks
Q7.		**2**
Q8.	**Yes**. **Overfitting** occurs when a neural network learns to memorize the training data rather than generalizing from it, leading to poor performance on new, unseen data. Reducing the number of weights and biases can help mitigate overfitting by simplifying the model **Note**: If the answer is **Yes – 1 mark**, justification is correction **another 1 mark**.	**2**
Q9.	A Scalar, Rank is 0 A Vector Rank is 1 Matrix M rank is 2	**2**
Q10.	**TensorFlow Lite (TFLite)** is specifically designed to run machine learning (ML) models on mobile and embedded devices in a way that addresses the memory **and compute** constraints of these platforms. It is an optimized, lightweight version of TensorFlow aimed at providing efficient and fast inferencing on edge devices. The following key factors make TensorFlow Lite suitable for such platforms: **(any two points are enough)** 1. Reduced Model Size 2. Optimized for Edge Computing 3. Low Power Consumption 4. Fast and Low-Latency Inference	**2**
Q11.	Since an integer is 32-bit value in ESP32, the address locations accessed are: **0x3FFF 0004 to 0x3FFF 0007**. This is a read operation because value stored in the memory location pointed to by the int pointer ip is being assigned to the integer variable val.	**2**
Q12.	One-hot encoding is a technique used to convert categorical data into a binary matrix or vector that can be used as an input to a Deep learning network. In the **car evaluation example**, each categorical feature (like price, maint, doors, etc.) has multiple possible values, which are represented as binary arrays where each possible value corresponds to a specific position in the array. If a particular category is active, its position in the array is set to 1, and all other positions are set to 0. If there are four categories of Price: **low, med, high, vhigh**, then if a sample Price value happens to be **high**, the **one-hot encoded value would be (0, 0, 1, 0)**	**2**
	AI Hub is a platform developed by Qualcomm Technologies to streamline the deployment of ML models on devices powered by Qualcomm Snapdragon platforms which include mobile and tablet devices and development kits. It offers a comprehensive ecosystem for optimizing, validating, and deploying machine learning models across various applications, including mobile, automotive, and Internet of Things (IoT) solution, including image classification, GenAI models.	**2**

Part B

The following steps are to be done to train a model and run the inference engine on ESP32.
1. Generate samples of a sine wave from 0 to 2pi radians.

Question No		Marks
Q13.	2. Randomize 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.	3
Q14.	**MQTT** (Message Queuing Telemetry Transport) is a lightweight, publish-subscribe messaging protocol designed for low-bandwidth, high-latency, or unreliable networks, making it ideal for IoT (Internet of Things) applications **Benefits of MQTT: Any two are sufficient:** 1. **Lightweight and efficient**: MQTT minimizes the resources required by clients and network bandwidth. 2. **Bidirectional communication**: MQTT facilitates communication between devices and servers, supporting publishing and subscribing. It also allows broadcasting messages to groups of devices. 3. **Scalability**: MQTT can scale to support millions of devices or "things" in an IoT or IIoT ecosystem. 4. **Quality of Service (QoS) levels**: MQTT specifies different QoS levels to ensure reliable message delivery. 5. **Persistent sessions**: MQTT supports persistent sessions between devices and servers, reducing reconnection time over unreliable networks. 6. **Security features**: MQTT supports TLS encryption for message confidentiality and authentication protocols for client verification.	3
Q15.	vTaskSuspend(task1Handle) suspends the task1 and puts it in the Suspend state until another process calls vTaskResume(task1Handle). When another process calls vTaskResume(task1Handle)., the suspended task1 is resumed and put in the Ready Queue. The state of the task1 after the vTaskSuspend(task1Handle) is Suspend and the state of the task1 immediately after vTaskResume(task1Handle) would be Ready.	3

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
5. Demonstrate the need for AI accelerators for IoT in the ML domain

Marks Distribution										
L1	L2	L3	L4	L5	L6	CO1	CO2	CO3	CO4	CO5
0	15	7	11	0	0	2	12	14	3	2

Signature of Paper Setter   **Signature of Scrutiniser**

Signature of the Dean