Birla Institute of Technology & Science, Pilani
Work Integrated Learning Programmes Division
Second Semester 2019-20
AIML Certification Program
Comprehensive Examination (Regular)

Course No.	: AIML_Oct_2019_PCAMZC241
Course Title	: DEEP LEARNING and ANN
Nature of Exam	: Closed Book
Total Marks	: 30
Duration	: 2 Hours
Date of Exam:	September 13, 2020 Time of Exam:

No. of Pages = 4
No. of Questions = 5

Note: Assumptions made if any, should be stated clearly at the beginning of your answer. Show your rough work to get partial credit, when appropriate.

Question 1. **[**1x5 + 2= **7 marks]**

Refer to the partial code of a CNN implementation of image classification and answer following questions. Input is gray-scale image of height = 224, width = 224, and output corresponds to class labels. Total number of classes is 10.

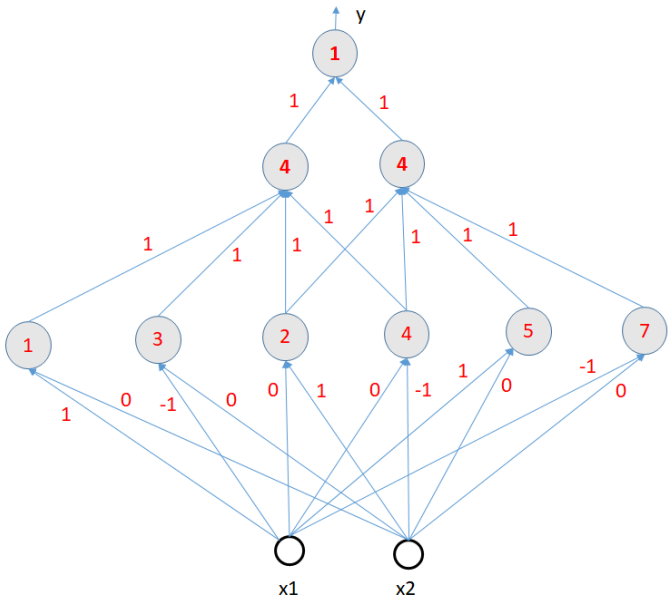
```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense

model = Sequential()

============================== Level 1

model.add(Conv2D(input_shape=(XXXaXXX),filters=64,kernel_size=(3,3),padding="same",
activation="relu"))
model.add(Conv2D(filters=64,kernel_size=(3,3),padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(XXXbXXX(filters=128, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))  ---
LAYER 4

model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))

model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

============================ Level 2

model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
```

```
========================== Level 3

model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(0.5))

========================== Level 4

model.add(Dense(XXXdXXX))
model.add(Activation('sigmoid'))

model.compile(loss=XXXcXXX, optimizer='rmsprop',metrics=['accuracy'])
```

A. Fill in the input shape (**XXXaXXX**)
B. Fill in the conv layer type to be used: (**XXXbXXX**)
C. Fill in the Loss function to be used: (**XXXcXXX**)
D. At what level we will have to use the Flatten Layer
    a. ========== 1
    b. ========== 2
    c. ========== 3
    d. ========== 4
E. Calculate the number of neurons in the final dense Layer
F. Calculate the number of parameters in layer 4 labelled above

Question 2. [1x3+3+1= **7 Marks**]

Refer to the following multilayer perceptron network with two hidden layers. All nodes use a step activation function, i.e., output = 1 if total input >= bias, else output=0. Bias at each node is indicated inside the node.



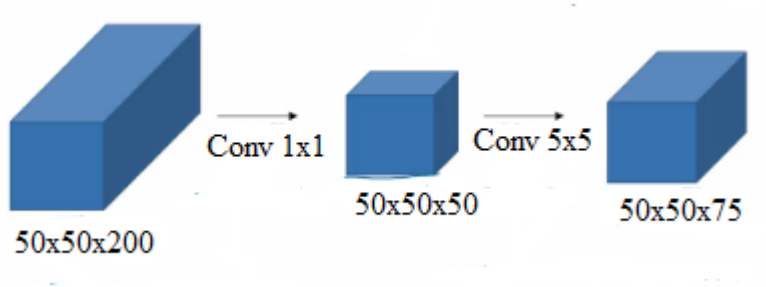
A. What will be the output y, if (x1,x2)=(2,3), (x1,x2)=(6,3), and (x1,x2)=(4,3)
B. Express output y as a decision rule of input (x1, x2). Recall, decision rule is expressed as an *if-then-else* statement, e.g., y = (x1> 2) OR (x2 < 7) AND (x1 + x2 = 7)

C. Can this decision rule be realized using a multilayer perceptron with one hidden layer? If yes, how many hidden node will be needed in that one hidden layer?

Question 3. [0.5 x 2 + 2 + 1 + 1 **= 5 Marks]**

A. In the following GoogLeNet step, 1x1 operators are first used to process 50x50 images of depth 200 and produce 50x50 images of depth 50, and then 5x5 operators are used to output final 50x50 images of depth 75.
   a) What is the padding size used in the first step and padding size in the last step?
   b) How many multiplication operations are needed in the first and final steps?

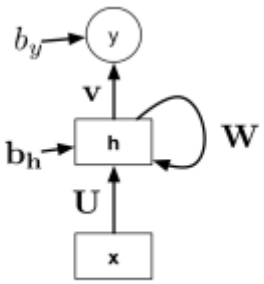
ii.



B. How is the vanishing gradient problem mitigated in deep layers of Resnet?
C. In faster R-CNN for object detection and classification, what does the multi-role loss function optimize in the region proposal network (RPN) module?

Question 4. [1 x 4 + 1 + 1 = 6 **marks]**

Refer to the following recurrent neural network with one input unit **x**, two hidden units **h**, and one output unit y. All of the units use the hard threshold activation function, i.e., output is 1 if total weighted input is >=bias, else 0.



Note, at time t, $\mathbf{h}_t$ = step($\mathbf{W}\mathbf{h}_{t-1}$ + $\mathbf{U}\mathbf{x}$ - $\mathbf{b}_h$) and y_t = step($\mathbf{v}\mathbf{h}_t$ - b_y). **U**, **v** and **W,** b_y and $\boldsymbol{b}_{\boldsymbol{h}}^{\boldsymbol{T}}$ are given by

$\mathbf{v}$ = [1 -1] and b_y = 0.5

$\boldsymbol{b}_{\boldsymbol{h}}^{\boldsymbol{T}} = [-0.5 \quad 0.5]$

W= $\begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$

$$\mathbf{U}= \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

A. The inputs are presented to this RNN as binary sequences from left to right. What will be the output string for the input string {0, 1, 0, 0}. Assume initial hidden state values are 0's.

B. What is the key limitation of RNN overcome by the long short term memory (LSTM) model?

C. Suppose an RNN is used to predict next character in the input string 'abracadabra' (no upper case needs to be considered). If one hot encoding is used for representing input and output, how many nodes will be needed in the input/output layer? If characters are presented from left to right, write the first input and target output vectors.

Question 5. Answer the following questions [1x5=**5 Marks]**

A. For what choice of activation functions in hidden and output nodes, autoencoder weights represent the principal component vectors?
B. Gradient ascent is used in training the generative module of the generative adversarial network (GAN). True/False
C. What types of convolution are used in the pooling layers of CNN based GANs?
D. Why is KL divergence used for regularization in variational autoencoders?
E. A dataset **X** is given. During training, how are the input and output data specified for a denoising autoencoder?