Curr Year Mid Sem Solution

BITS WILP M Tech Data Science & Engineering (Birla Institute of Technology and Science, Pilani)

Birla Institute of Technology & Science, Pilani
Work Integrated Learning Programmes Division
Second Semester 2020-21
M.Tech. (Data Science and Engineering)
Midsem Examination (Regular)

Course No.	: DSECLZG524
Course Title	: DEEP LEARNING
Nature of Exam	: Open Book
Weightage	: 30%
Duration	: 2 Hours
Date of Examination	: July 10th, 2020

No. of Pages = 3
No. of Questions = 5

Time of Exam: 10 AM – 12 PM

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. [2 + 2.5 + 1.5 **=6 marks]**

Consider the following DNN for image classification of 10 classes. The dataset consists of colour images of size 16x16.

```
# Input Layer
inputs = keras.Input(shape=(##A##))

# Layer 1
x = layers.Dense(30, activation="relu", use_bias=False)(inputs)

# Layer 2
x = layers.Dense(60, activation="relu")(x)

# Layer 3
x = layers.Dense(120, activation="relu", use_bias=False)(x)

# Output Layer
outputs = layers.Dense(##B##, activation=##C##)(x)

model = keras.Model(inputs=inputs, outputs=outputs,
name="model")
model.compile(optimizer = 'adam', loss = ##D##,
metrics=['accuracy'])
```

A. Write the value or code of ##A##, ##B##, ##C## and ##D## to complete the network.

A - (16*16*3,) or (768,)
B - 10
C - softmax
D - categorical cross-entropy

B. If we add a dropout layer of value 0.4 after layer 2-

I.what will be the total number of active neurons in layer 2 during

i. model training

60*(1-0.4) = 36

ii. ii. model testing

60

II. what will be the total number of parameters in the network? What is the change in the number of parameters compared to the original network?

Layer 1- 768*30 = 23,040
Layer 2- 30*60 + 60 = 1,860
Layer 3- 60*120 = 7,200
Output Layer - 120*10 + 10 = 1,210
Total = 23,040 + 1,860 + 7,200 + 1,210 = 33,310
No change in the number of parameters if dropout is added.

C. Assume that the image dataset consists of 5000 train images and 1000 test images. We run for 10 epochs with batch size = 32, learning rate (LR) = 0.01 and sgd optimizer. During model training

I. What will be the size of the batch in the last iteration?

5000 - (156x32) = 8

II. What is the number of times we perform forward propagation?

5000

III. What is the number of times we perform backpropagation? And why?

5000/32 = 156.25 i.e., 157 times backpropagation

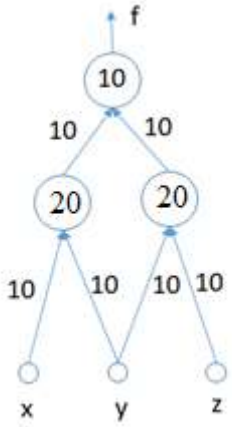
We perform backprop one time per batch taking the average loss of all the images in that batch and finally update the parameters of the network.

Question 2. [2+4**=6 Marks**]

A. Implement the following truth-table as a single-hidden layer MLP with the fewest perceptrons. x, y, z are input binary variables and f is the Boolean function, as specified in the truth table. Perceptrons use step activation function, i.e.,

output = +1 if total input >= bias,
= 0 otherwise.

f (x,y,z) = yz+xy

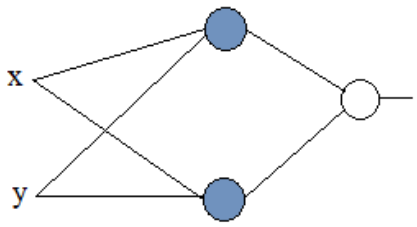


B. Specify all the weights and bias values of the network. Note, weights can be only +10 or -10 or 0, and bias values are multiples of 10 only.

x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Question 3. [3+3=**6 Marks**]

A. Consider the MLP below where each perceptron has a linear activation function where the output z is related to its inputs x and y as z = ax + by + c. We claim that this network can be replaced by a single perceptron with a possibly different linear activation function. Is this claim true or not? Give quantitative justification for your answer.



Let x1, y1 be the outputs of the upper and lower hidden nodes, respectively. So, x1=ax+by+c, and y1=ax+by+c.
So, final output of network, z = ax1+by1+c=a(ax+by+c)+b(ax+by+c)+c
Or, z = $(a^2+ab)x+(b^2+ab)y+(ac+bc+c)$

Thus, a single perceptron with input x, y is enough to implement the given network.

B. We would like to minimize the following quadratic function $0.5*25x^2 + 16*x + 12$ using exactly two steps using gradient-descent with a learning rate $\eta = 1.0$ for the first step. What should the learning rate of the second-step be in order to obtain the minimum in two-steps?

Optimal learning rate $\eta = 1/25$

Question 4. [4 **Marks]**

Use RPROP to perform the minimization of $24 * x^2 + 16x + 12$ starting at x = 2.0 with initial step size 1.0. Take $\alpha = 1.1$ and $\beta = 0.8$. Find minimum of the function and corresponding value of x.

$\frac{df}{dx} = 48x+16$

$\frac{df}{dx}\Big|_{x0=2.0} = +ve$

x = 2.0-1.0=1.0

$\frac{df}{dx}\Big|_{x=1.0} = +ve$

x = 1.0-1.1=-0.1

$\frac{df}{dx}\Big|_{x=-0.1} = +ve$

x = -0.1-1.1*1.1=-1.31

$\frac{df}{dx}\Big|_{x=-1.31} = -ve$

x = -0.1-1.1*1.1*0.8=-1.068

$\frac{df}{dx}\Big|_{x=-1.068} = -ve$

x = -0.1-1.1*1.1*0.8*0.8=-0.8744

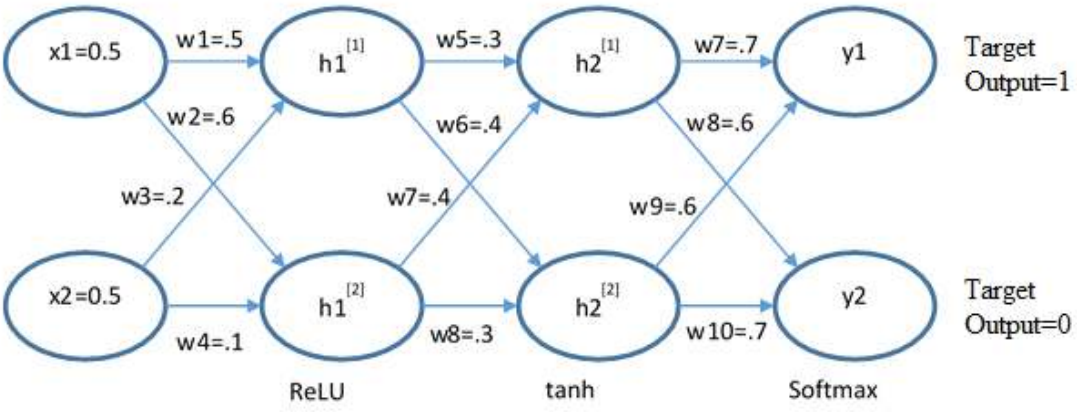
$\frac{df}{dx}\Big|_{x=-0.8744} = -ve$

Follow this process, until df/dx is close to 0.

The function reaches minimum of 12 at x=-0.33.

Question 5. [5+1+2=**8 Marks]**

A. The following network uses ReLU, tanh and softmax activation for hidden layer 1, hidden layer 2, and output layer, respectively. For the given input, calculate the Loss, backpropagate the error, and find new weights for w1 in the next iteration. No bias is used in any node and learning rate is 1.0.



$dh1^{[1]} = 0.35$ $dh1^{[2]} = 0.35$ $h2^{[1]} = 0.245$ $dh2^{[2]} = 0.245$

y1=exp(0.245*1.3)/(exp(0.245*1.3)+exp(0.245*1.3)=0.5

y2= exp(0.245*1.3)/(exp(0.245*1.3)+exp(0.245*1.3)=0.5

Loss L = -d1*log(y1)-d2*log(y2)=-log(y1)=0.693

$dL/dw1 = dL/dz1*dz1/dh2^{[1]} * dh2^{[1]} / dh1^{[1]} * dh1^{[1]} / dw1 +$
$dL/dz2*dz2/dh2^{[1]} * dh2^{[1]} / dh1^{[1]} * dh1^{[1]} / dw1 +$
$dL/dz1*dz1/dh2^{[2]} * dh2^{[2]} / dh1^{[1]} * dh1^{[1]} / dw1 +$
$dL/dz2*dz2/dh2^{[2]} * dh2^{[2]} / dh1^{[1]} * dh1^{[1]} / dw1$

$= dL/dz1*w7 * w5 * (1-h2^{[1]} * h2^{[1]}) * x1 + dL/dz2*w8 * w5 * (1-h2^{[1]} * h2^{[1]}) * x1 +$
$dL/dz1*w9 * w6 * (1-h2^{[2]} * h2^{[2]}) * x1 + dL/dz2*w10 * w6 * (1-h2^{[2]} * h2^{[2]}) * x1$

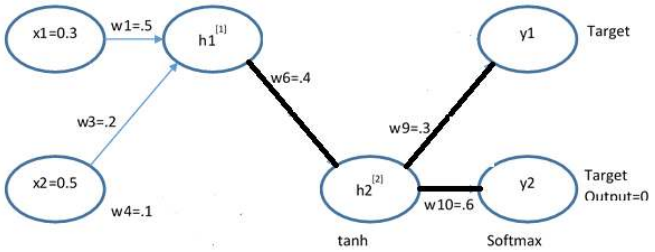
dL/dz1 = dL/dy1*dy1/dz1 = -1/y1*y1*(1-y1)=0.5-1=-0.50
dL/dz2 = dL/dy2*dy2/dz2 = 0

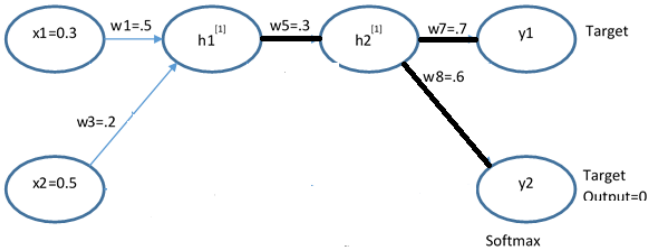
$w1 = 0.5 - (-0.5*w7 * w5 * (1-h2^{[1]} * h2^{[1]}) * x1-0.5*w9 * w6 * (1-h2^{[2]} * h2^{[2]}) * x1)$
=0.5+0.5*0.5*(0.21*(1-0.245*0.245)+0.24*(1-0.245*0.245))=0.6057

B. Assume you have a 50% dropout for Hidden layer 1 and Hidden layer 2.

I. How many unique paths are possible for the above network? 4

II. Identify two unique paths and explain how forward propagation and back propagation is calculated.





Forward calculation will be based on the above two network structure and backward gradient calculation and weight updates will be only based on the above two structures. Other weights will not chance in an iteration.

XXXXXXXXXXX

Birla Institute of Technology & Science, Pilani
Work Integrated Learning Programmes Division
Second Semester 2020-21
M.Tech. (Data Science and Engineering)
Midsem Examination (Makeup)

Course No.	: DSECLZG524
Course Title	: DEEP LEARNING
Nature of Exam	: Open Book
Weightage	: 30%
Duration	: 2 Hours
Date of Examination	: July 24th, 2021

No. of Pages = 3
No. of Questions = 5

Time of Exam: 10 AM – 12 PM

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

A. While selecting different layers of a model for an image classification problem, what all factors do we consider to decide the total number of neurons? Select the correct/incorrect answers and explain.

I) Hardware capacity
II) Inter class variation in the dataset
III) Intra class variation in the dataset
IV) Total number of images in the dataset
V) Number of image augmentations applied on the dataset

Ans. A, B, C, E

A - Number of neurons cannot exceed the memory and hardware capacity.

B, C - More the inter and intra class variations, more the number of neurons required to capture the variation.

E - More the augmentations applied, more the number of neurons required to capture variations in original images and the augmented images.

Number of kernels does not depend on total number of images, as it can be handled by the batch size

B. An image classification problem consists of 4 classes (0, 1, 2, and 3). The output layer of the model consists of a one-hot encoding vector corresponding to each class. The values of each class before softmax activation are 10, 8, 12, and 5.

I) What is the output value of the predicted class post activation?

Class with the pre-activation value of 12 is fed to the loss function. Its output value is $(e^{12})/(e^{10} + e^{8} + e^{12} + e^{5})$ =

II) What is the difference between softmax values of class 1 and class 3?

Difference in softmax values of class 1 and 3 is $(e^8 - e^5)/(e^{10} + e^8 + e^{12} + e^5)$ =

C. Tuning hyperparameters using a test dataset. Is it preferred, if not then why?
Tuning model hyperparameters to a test set means that the hyperparameters may overfit to that test set. If the same test set is used to estimate performance, it will produce an overestimate. Using a separate validation set for tuning and test set for measuring performance provides unbiased, realistic measurement of performance.

D. If there are 8 classes and the classifier predicts each class with equal probability. What will be the cross-entropy loss on any single example?
Cross-Entropy loss simplifies to negative log of the predicted probability for the correct class. At the start of training, we have approximately uniform probabilities for the 8 classes, or 1/8 for each class. So, that means -log(1/8) is what the loss should approximately be at the start.

E. In a scenario while training a neural network for classification, the training loss comes much lower than the validation loss. What do you think is the reason and give two ways to resolve it?
The model is now overfitting, and all of these are valid techniques to address it.
1. Use a network with fewer layers
2. Increase L2 regularization weight
However, since dropout is a form of regularization, then decreasing it will have the opposite effect, as will increasing network size.
1. Decrease dropout probability
2. Increase the size of each hidden layer

Question 2. [2.5+2.5**=5 Marks**]

Consider the expression f(x1,x2) = 2x1x2 + 3x1 + 4x2.

A. Determine for the point (x1, x2) = (0, 0) whether it is (a) a local minima, (b) a local maxima or a saddle point, or none of these.
The point (0, 0) is either a local minimum, local maximum or saddle point since the gradient of f is zero at this point. To determine which of these it is we need to consider the Hessian H which evaluates to

$$\begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}$$

The eigenvalues of the above matrix turn out to be 1 and -1 respectively which means that (0, 0) is a saddle point.

B. Do the same exercise for the point (x1, x2) = (1, 1).

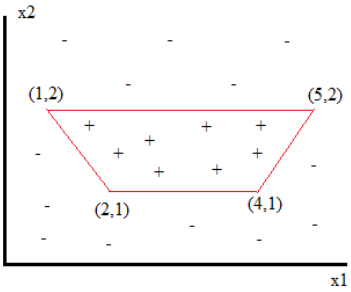
(1, 1) is none of these as the gradient of f is non-zero at this point.

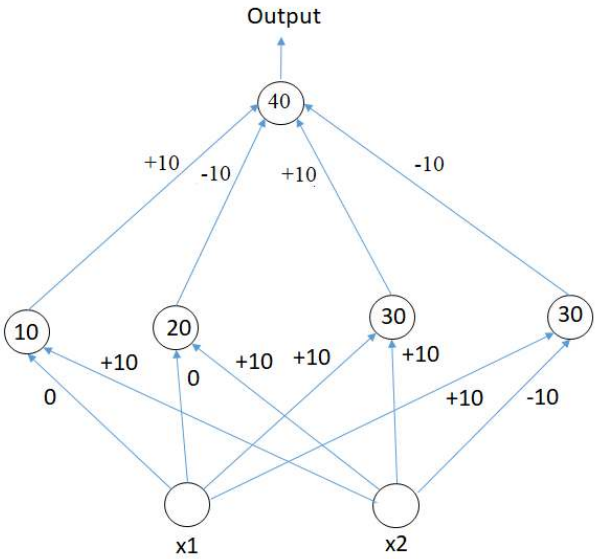
Question 3. [1+1+4=**6 Marks**]

A. What is the minimum number of hidden layers required to implement the following decision boundary?
1

B. What will be minimum number of hidden nodes required? Show the network architecture.
4

C. Organize the hidden nodes from left to right in ascending order of bias values. Use the input x1 as the left node in the input layer. Specify all the weights (only 0, 10 or -10 allowed) and bias (only 0 or multiples of 10). Hidden and Output units use step activation, i.e., output = 1 if total input >=bias, otherwise -1.





Question 4. [4 **Marks**]

Consider the following problem of L1-regularization, i.e., minimize for i=1 to n

$$L_R(\theta_i) = H_{ii}(\theta_i - \theta_i^*)^2 + \alpha \mid \theta_i \mid \text{ where all } H_{ii} > 0,$$

and α is the L1 regularization constant. Find the smallest value of α (in terms of H_{ii} and θ_i^*) so that the optimal regularized solution to the given minimization problem, $\theta_i^R = 0$ for all i.

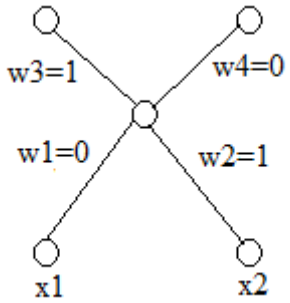
Answer: The regularised optimal solution is as follows: $(\theta_R^*)_i = max(\theta_i^* - \frac{\alpha}{H_{ii}}, 0), if \theta_i^* \geq 0$ and $(\theta_R^*)_i = min(\theta_i^* + \frac{\alpha}{H_{ii}}, 0), if \theta_i^* \leq 0$. If we select α such that $\theta_i^* < \frac{\alpha}{H_{ii}}$ and $-\theta_i^* < \frac{\alpha}{H_{ii}}$, then we can ensure that $(\theta_R^*)_i = 0$ for all i. This condition amounts to picking an α such that $\alpha \geq H_{ii}\|\theta_i^*\|$ for all i.

Question 5. [2+1+2+3=**8 Marks**]

A. Consider a fully connected multilayer perceptron (each hidden node is connected to all inputs and all outputs) with 2 dimensional binary input (0 or 1) and one hidden layer with ReLU activation function, and target output, same as the input (so, input [1,1] is associated with target output [1,1]). What activation function will you use at the output layer? What type of loss function will you use?
Sigmoid activation at output node since outputs are 0-1. Cross entropy is used as the loss function because the output is 0-1.

B. At iteration t, the weights are shown in the following architecture along with the input vector (x1=1, x2=0). What will be value of the loss function at iteration t?
Output of hidden node =0. Reconstructed output = (0.5, 0.5) Target output = (1.0,0.0)
So, loss = -ln(0.5)-ln(0.5)=1.386

C. What will be the weights w1 and w3 in iteration t+1 assuming gradient descent and gradient descent with momentum? Assume learning rate = 0.3 and momentum constant = 0.7. At (t-1), w1=-0.5, w2=0.5, w3=0.5 and w4=-0.5. Assume derivative of ReLU(z)=0 at z=0.



In ordinary gradient descent, change in w3 is proportional to hidden node output. So in this case, w3(t+1)=1.0 if ordinary gradient descent is used.
In ordinary gradient descent, change in w1is proportional to derivative of the hidden node activation function at hidden node output point. Deviative of ReLU(z)=0 at z=0. So in this case, w1(t+1)=0.0 if ordinary gradient descent is used.
For momentum based gradient update, w3(t+1)=1.0+0.7*0.5=1.35, w1(t+1) = 0.0+0.7*0.5=0.35

XXXXXXXXXXXX