**WORKSHEET-2**

**DEEP LEARNING**

1. Operations in the neural networks can performed \_\_\_\_\_\_\_?

Answer: C) serially or parallel

2. Who proposed the first perceptron model and when?

Answer: A) Rosenblatt, 1958

3. Which one of these plots represents a ReLU activation function?

Answer: D)

4. In a simple artificial neural network with 5 neurons in the input layer, 8 neurons in the hidden layer and 3 neurons in the output layer. What is the size of the weight matrices between hidden-output layers and input-hidden layers?

Answer: B) [8×3], [5×8]

5. What is a dead unit in a neural network?

Answer: D) None of these

6. Which of the following functions can be used as an activation function if we wish to predict the probabilities of n classes such that sum of all n probabilities is equal to 1?

Answer: A) sigmoid

7. The amount of output of one unit received by another unit depends on what?

Answer: D) weights

8. What is asynchronous update in neural networks?

Answer: B) output units are updated sequentially

9. Which of the following techniques can be used to reduce overfitting in a neural network?

Answer: A) EarlyStopping B) Dropout D) ReduceLROnPlateau

10. Why is an RNN used for machine translation, say translating English to Hindi?

Answer: A) It can be trained as a supervised learning problem.

C) It is applicable when the input/output is a sequence (e.g., a sequence of words

11. The output of a perceptron is calculated as follows:

y = f(b + ∑wiXi )

Where f(x) is the activation function. If you want to build a perceptron which gives an output for linear regression, what will be the activation function you would use?

Answer: To build a perceptron which gives an output for linear regression, we will use linear activation function, it has the equation similar to as of a straight line i.e. y = ax, No matter how many layers we have, if all are linear in nature, the final activation function of last layer is nothing but just a linear function of the input of first layer.

12. What will happen if we use very large or very small learning rates?

Answer: The learning rate is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. Choosing the learning rate is challenging as a value too small may result in a long training process that could get stuck, whereas a value too large may result in learning a sub-optimal set of weights too fast or an unstable training process. Larger learning rates result in rapid changes and require fewer training epochs. A learning rate that is too large can cause the model to converge too quickly to a suboptimal solution.

13. Below is a diagram if a single artificial neuron:

The node has three inputs x = (x1, x2, x3) that receive only binary signals (either 0 or 1). How many different input patterns this node can receive? What if the node had four, five inputs? Can you give a formula that computes the number of binary input patterns for a given number of inputs?

Answer: Possible inputs for three inputs:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| X2 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| X3 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

Here, for three inputs we got 8 possible outputs i.e. 23

If, we have four inputs, we will get 16 possible combination of input value i.e 24

So, we can derive 2n = where n is the number of input

14. What Are Vanishing and Exploding Gradients?

Answer: Vanishing Gradients: The term vanishing gradient refers to the fact that in a feedforward network (FFN) the backpropagated error signal typically decreases (or increases) exponentially as a function of the distance from the final layer. A problem with training networks with many layers (e.g. deep neural networks) is that the gradient diminishes dramatically as it is propagated backward through the network. The error may be so small by the time it reaches layers close to the input of the model that it may have very little effect. As such, this problem is referred to as the “vanishing gradients” problem. In a network of n hidden layers, n derivatives will be multiplied together. If the derivatives are large then the gradient will increase exponentially as we propagate down the model until they eventually explode, and this is what we call the problem of exploding gradient. Alternatively, if the derivatives are small then the gradient will decrease exponentially as we propagate through the model until it eventually vanishes, and this is the vanishing gradient problem. In the case of exploding gradients, the accumulation of large derivatives results in the model being very unstable and incapable of effective learning.

15. What Is the Difference Between Epoch, Batch, and Iteration in Deep Learning?

Answer: Epoch is when an ENTIRE dataset is passed forward and backward through the neural network only ONCE.

Batch: Total number of training examples present in a single batch. As we cannot pass entire dataset to neural network, so we divide the data into number of batches.

Iteration: It is the number of batches needed to complete one epoch.