**WORKSHEET-2**

**DEEP LEARNING**

# Q1 to Q8 are MCQs with only one correct answer. Choose the correct option.

1. Operations in the neural networks can performed ?
   1. serially B) parallely

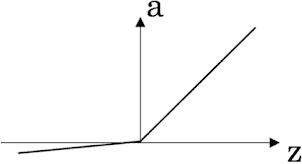
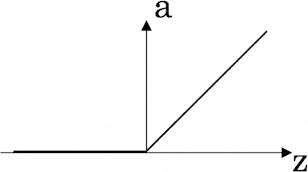
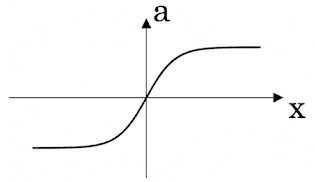
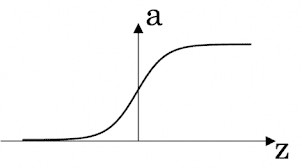
C) serially or parallely D) None of the above

1. Who proposed the first perceptron model and when?

A) Rosenblatt, 1958 B) McCulloch-pitts, 1958

C) John Hopfield, 1982 D) McCulloch-pitts, 1982

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

A)  B) 

C) D)

1. 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?

A) [3×8], [5×8] B) [8×3], [5×8]

C) [5×8], [8×5] D) [8×3], [5×3]

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

A) A unit which does not respond completely to any of the training patterns

1. The unit which produces the biggest sum-squared error
2. A unit which doesn’t update during training by any of its neighbour
3. None of these
4. 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?
   1. sigmoid B) softmax

C) tanh D) ReLU

1. The amount of output of one unit received by another unit depends on what?
   1. output unit B) input unit

C) activation values D) weights

1. What is asynchronous update in neural networks?
   1. output units are updated parallely B) output units are updated sequentially

C) either sequentially or parallely D) None of the above

# Q9 and Q10 are MCQs with one or more correct answers. Choose all the correct options.

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

A) EarlyStopping B) Dropout

C) checkpoints D) ReduceLROnPlateau

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

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

B) It is strictly more powerful than a Convolutional Neural Network

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

D) RNNs represent the recurrent process of Idea->Code->Experiment->Idea->....

# Q11 to Q15 are subjective answer type question. Answer them briefly.

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

*n*

*y*  *f* (*b*   *wi xi* )

*i* 1

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?

Ans: A neural network without an activation function is essentially just a linear regression model. The activation function does the non-linear

**Linear Function :-**

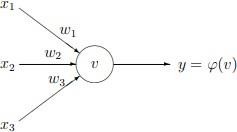
* **Equation :**Linear function has the equation similar to as of a straight line i.e. **y = ax**
* No matter how many layers we have, if all are linear in nature, the final activation function of last layer is nothing but just a linear function of the input of first layer.
* **Range :** -inf to +inf
* **Uses : Linear activation function** is used at just one place i.e. output layer.
* **Issues :**If we will differentiate linear function to bring non-linearity, result will no more depend on *input “x”* and function will become constant, it won’t introduce any ground-breaking behavior to our algorithm.

**For example :** Calculation of price of a house is a regression problem. House price may have any big/small value, so we can apply linear activation at output layer. Even in this case neural net must have any non-linear function at hidden layers.

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

Ans: 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.

1. 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?



1. What Are Vanishing and Exploding Gradients?

Ans: An error gradient is the direction and magnitude calculated during the training of a neural network that is used to update the network weights in the right direction and by the right amount.

In deep networks or recurrent neural networks, error gradients can accumulate during an update and result in very large gradients. These in turn result in large updates to the network weights, and in turn, an unstable network. At an extreme, the values of weights can become so large as to overflow and result in NaN values.

The explosion occurs through exponential growth by repeatedly multiplying gradients through the network layers that have values larger than 1.0.

In recurrent neural networks, exploding gradients can result in an unstable network that is unable to learn from training data and at best a network that cannot learn over long input sequences of data.

How do You Know if You Have Exploding Gradients?

There are some subtle signs that you may be suffering from exploding gradients during the training of your network, such as:

The model is unable to get traction on your training data (e.g. poor loss).

The model is unstable, resulting in large changes in loss from update to update.

The model loss goes to NaN during training.

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

Ans: **A batch** is the complete dataset. Its size is the total number of training examples in the available dataset.Mini-batch size is the number of examples the learning algorithm processes in a single pass (forward and backward).A Mini-batch is a small part of the dataset of given mini-batch size.

**Iterations** is the number of batches of data the algorithm has seen (or simply the number of passes the algorithm has done on the dataset).

**Epochs** is the number of times a learning algorithm sees the complete dataset. Now, this may not be equal to the number of iterations, as the dataset can also be processed in mini-batches, in essence, a single pass may process only a part of the dataset. In such cases, the number of iterations is not equal to the number of epochs.

