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

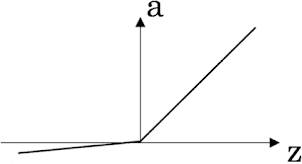
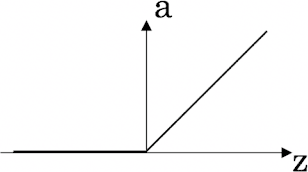
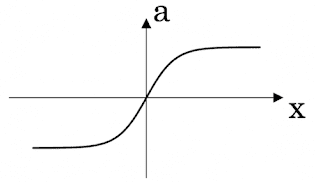
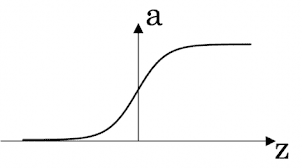
**Ans. C) serially or parallely**

1. Who proposed the first perceptron model and when?
   1. Rosenblatt, 1958 B) McCulloch-pitts, 1958

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

**Ans. A) Rosenblatt, 1958**

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

A)  B) 

C) D)

**Ans. C)**

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?
   1. A unit which does not respond completely to any of the training patterns
   2. The unit which produces the biggest sum-squared error
   3. A unit which doesn’t update during training by any of its neighbour
   4. None of these

**Ans. C) A unit which doesn’t update during training by any of its neighbour**

1. 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

**Ans. B) softmax**

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

**Ans. 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

**Ans. 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?
   1. EarlyStopping B) Dropout

C) checkpoints D) ReduceLROnPlateau

**Ans. A) EarlyStopping, B) Dropout**

1. Why is an RNN used for machine translation, say translating English to Hindi?
   1. It can be trained as a supervised learning problem.
   2. It is strictly more powerful than a Convolutional Neural Network
   3. It is applicable when the input/output is a sequence (e.g., a sequence of words)
   4. RNNs represent the recurrent process of Idea->Code->Experiment->Idea->....

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

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

# 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?

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

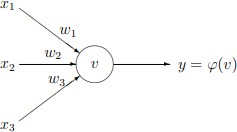
**Ans**. Generally, a large learning rate allows the model to learn faster, at the cost of arriving on a sub-optimal final set of weights. A smaller learning rate may allow the model to learn a more optimal or even globally optimal set of weights but may take significantly longer to train.

When the learning rate is too large, gradient descent can inadvertently increase rather than decrease the training error. When the learning rate is too small, training is not only slower, but may become permanently stuck with a high training error.

If you’re learning rate is too low, you will see a very slow convergence, simple as that. Bengio et al. addressed the issue of vanishing gradients in 1994 and it's generally known that you can always do better if you don't just rely on your intuition with choosing some constant value for the learning rate. The outcome in case when the learning rate is too small will be a [dramatically] increased time of the learning, but eventually you will get the results.

If you’re learning rate is too high, your loss function will grow very fast. It's not unheard of for a norm of gradient of a loss function to be several magnitudes higher than the weights matrix, which also doesn't help the case.

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. Vanishing gradient Problem**

Vanishing gradient problem is a common problem that we face while training deep neural networks.Gradients of neural networks are found during back propagation.

Generally, adding more hidden layers will make the network able to learn more complex arbitrary functions, and thus do a better job in predicting future outcomes. This is where Deep Learning is making a big difference.

Now during back-propagation i.e moving backward in the Network and calculating gradients, it tends to get smaller and smaller as we keep on moving backward in the Network. This means that the neurons in the Earlier layers learn very slowly as compared to the neurons in the later layers in the Hierarchy. The Earlier layers in the network are slowest to train.This is an issue with deep neural networks with large number of hidden layers. Ie, we know updated weight

**W\_new = W\_old — η gradient**

For earlier layers this gradient will be very small. So there will be no significant difference between **W\_new**and **W\_old.**This arises the problem of vanishing gradient descent.This is the problem mainly with sigmoid and tanh functions.So we use RELU based activation functions in training a Deep Neural Network Model to avoid such complications and improve the accuracy .

### **Exploding gradient Problem**

We have discussed about vanishing gradient problem.Now we will get in to exploding gradient problem.Earlier we discussed what happens when our gradient becomes very small.Now we will discuss what will happen if it gets large.

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.The explosion occurs through exponential growth by repeatedly multiplying gradients through the network layers that have values larger than 1.0.This will ultimately led to an total unstable network.

We can understand exploding gradient problem by following techniques:

* We have poor loss on training data
* The change is loss will be large from update to update

Some techniques to overcome exploding gradient descent are:

* Use LSTM in case of RNN where we constantly face exploding gradient problem
* Use weight regularization.

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

**Ans.**

* Epoch – Represents one iteration over the entire dataset (everything put into the training model).
* Batch – Refers to when we cannot pass the entire dataset into the neural network at once, so we divide the dataset into several batches.
* Iteration – if we have 10,000 images as data and a batch size of 200. then an epoch should run 50 iterations (10,000 divided by 50)