1. C

2. A

3. C

4. B

5. C

6. B

7. D

8. B

9. ABD

10. AC

11.

If we want to build a perceptron to give outputs of a linear regression problem, we should use a linear activation function, so that f(WTX+b)= WTX+b.

12.

If the learning rates are too small, the gradient descent will take a lot of time to converge(given the loss function is a convex function) and the model may over-fit. Also, if the function is non-convex instead, it may get stuck in some local minima or a saddle point.

On the other hand, if the learning rate is too high, then the solution of gradient descent might diverge instead of converging or keep jumping around the minima.

13.

The number of different input patterns are 23.

Given d input nodes, the formula for different number of input patterns can be given as 2d.

14.

In back propagation, when we are trying to update weights in earlier layers, a lot of gradient terms get multiplied to form dL/dW. Now, if we using sigmoid or tanh activation function, the gradient values will lie between [0,1]. So, a lot of terms combined together makes dL/dW tend to zero, hence there is close to no change in weights in initial layers. This is called problem of vanishing gradients.

Similarly, if the value of each gradient term is >1, they combine to form a large value of dL/dW, thus creating a lot of change in weights, which may create our solution to diverge instead of converging. This is problem of exploding gradients.

15.

An epoch is completed when our model has seen each and every point in our dataset once, means we have passed (forward propagated) and back propageted our full set of training data through our network.

Iterations are the number of batches we need to process in order to complete one epoch.

Batch is the number of points we process together in forward and backward propagation. Until and unless we have huge volume of RAM, we can’t process our entire training data in one go, therefore we divide it into batches.