## **DEEP LEARNING**

## **WORKSHEET – 5**

## **Solutions**

1. D) All of the above
2. A) Sigmoids do not saturate and hence have faster convergence,

D) Sigmoids are not zero centered; gradient updates go too far in different directions, making optimization more difficult

1. D) None of the above
2. A) True
3. B) Xavier Initialisation
4. A) learning rate shrinks and becomes infinitesimally small
5. B) momentum must be high and learning rate must be low
6. C) when it has many saddle points and flat areas
7. A) ADAM C) NADAM D) RMS Prop
8. C) when it reaches global minimum D) when it reaches a local minima which is similar to global minima (i.e. which has very less error distance with global minima)
9. Convex optimization

Convex optimization means that the function is convex and so the search area is convex. In such circumstances there exists exactly one minima, moreover it is located inside the search area. Any minima finding (correct) algorithm should therefore locate the global minima quite easily.

Non-convex optimization:

Non-convex optimization involves a function which has multiple optima, only one of which is the global optima. Depending on the loss surface, it can be very difficult to locate the global optima.

1. When we optimize neural networks or any high dimensional function, for most of the trajectory we optimize, the critical points (the points where the derivative is zero or close to zero) are saddle points. Thus we can say that Saddle point simultaneously contains a local minimum and a local maximum.
2. The main difference is in classical momentum you first correct your velocity and then make a big step according to that velocity (and then repeat), but in Nesterov momentum, you first make a step into velocity direction and then make a correction to a velocity vector based on a new location (then repeat).
3. The aim of weight initialization is to prevent layer activation outputs from exploding or vanishing during the course of a forward pass through a deep neural network. One other method of initializing weights is to use pre-initialization. The technique involves importing the weights of an already trained network (such as VGG16, AlexNet) and using these as the initial weights of the network to be trained.
4. In neural networks, the output of the first layer feeds into the second layer, the output of the second layer feeds into the third, and so on.

The distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities. This phenomenon is known as internal covariate shift.