**Deep Learning Worksheet-5 Solution**

1. D (All of the above.)
2. C ( Sigmoid saturate and kill gradients.)
3. C (Soft Plus)
4. A (True)
5. B (Xavier Initialization)
6. A (learning rate shrinks and becomes infinitesimally small)
7. B (momentum must be high and learning rate must be low)
8. C (when it has many saddle points and flat areas)
9. A and C
10. C and D
11. **Convex optimization** there can be only one optimal solution, which is globally optimal or you might prove that there is no feasible solution to the problem, while in **Non-convex optimization** may have multiple locally optimal points and it can take a lot of time to identify whether the problem has no solution or if the solution is global. Hence, the efficiency in time of the convex optimization problem is much better. A convex problem usually is much more easier to deal with in comparison to a non-convex problem which takes a lot of time and it might lead you to a dead end.
12. In low dimensions, it is true that there exists lots of local minima. However in high dimensions, local minima are not really the critical points that are the most prevalent in points of interest. 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**.

The intuition with the saddle point, is that, for a minima located close to the global minima, all directions should be climbing upward; going further downward is not possible. Local minima exist, but are very close to global minima in terms of objective functions, and theoretical results suggest that some large functions have their probability concentrated between the index (the critical points) and the objective function. The index is the fraction of directions moving downward; for all values of index not 0 or 1 (local minima and maxima, respectively), then it is a **saddle point**.

1. 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 making a step into velocity direction and then make a correction to a velocity vector based on new location (then repeat).

Classical momentum:

vW(t+1) = momentum.\*Vw(t) - scaling .\* gradient\_F( W(t) )

W(t+1) = W(t) + vW(t+1)

While Nesterov momentum is this:

vW(t+1) = momentum.\*Vw(t) - scaling .\* gradient\_F( W(t) + momentum.\*vW(t) )

W(t+1) = W(t) + vW(t+1)

1. The aim of pre-weight initialization is to prevent layer activation outputs from exploding or vanishing during the course of a forward pass through a deep neural network.
2. **Internal Covariate Shift** is the change in the distribution of **network** activations due to the change in **network** parameters during training. 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.