**DEEP LEARNING – WORKSHEET 5 ANSWERSHEET**

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

1. D) All of the above
2. C) Sigmoids saturate and kill gradients.
3. C) SoftPlus
4. A) True
5. C) He Normal Initialisation
6. A) learning rate shrinks and becomes infinitesimally small
7. C) momentum and learning rate both must be low
8. B) when it has many local maxima

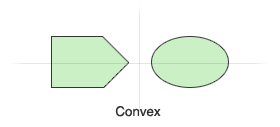
**Q9 and Q10 are MCQs with one or more correct answers. Choose all the correct options.**

1. A) ADAM C) NADAM
2. A) when it reaches local 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)

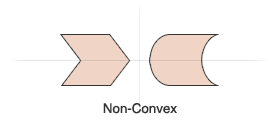
**Q11 to Q15 are subjective answer type question. Answer them briefly.**

1. **Convex Optimization Problems**-A convex optimization problem is a problem where all of the constraints are convex functions, and the objective is a convex function if minimizing, or a concave function if maximizing. **Linear functions are convex**, so linear programming problems are convex problems.  [Conic optimization](https://www.solver.com/conic-optimization) problems -- the natural extension of linear programming problems -- are also convex problems. In a convex optimization problem, the feasible region -- the intersection of convex constraint functions -- is a convex region, as pictured below.



With a convex objective and a convex feasible region, there can be only one optimal solution, which is **globally optimal**.  Several methods -- notably Interior Point methods -- will either find the globally optimal solution, or **prove** that there is no feasible solution to the problem.  Convex problems can be solved efficiently up to very large size.

A non-convex optimization problem is any problem where the objective or any of the constraints are non-convex, as pictured below.



Such a problem may have multiple feasible regions and multiple locally optimal points within each region.  It can take time **exponential** in the number of variables and constraints to determine that a non-convex problem is infeasible, that the objective function is unbounded, or that an optimal solution is the "global optimum" across all feasible regions.

1. **Saddle Points -** “Let us consider the optimization problem in low dimensions vs high dimensions. 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. Saddle points, unlike local minima, are easily escapable." 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.
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 making a step into velocity direction and then make a correction to a velocity vector based on 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. If either occurs, loss gradients will either be too large or too small to flow backwards beneficially, and the network will take longer to converge, if it is even able to do so at all.
4. We define Internal Covariate Shift as 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. When the parameters of a layer change, so does the distribution of inputs to subsequent layers.

These shifts in input distributions can be problematic for neural networks, especially deep neural networks that could have a large number of layers. Batch normalization is a method intended to mitigate internal covariate shift for neural networks.