Name : Naman Dave
ID : 201801439

My individual contribution.

Programming:

1. Problem 2: Retrieving approximated functions for max speed, handling and acceleration (or RPMs) of an F1 race car as well as their weights in tuning.
2. Problem 4: Complete programming of the problem including HW enabling and HW-accelerator configuration and GPU/CUDA-NVIDIA function programming of matrices as per need.
3. Problem 5: Complete programming of both the parts and their errors as well as efficiencies.

Documentation:

PPT slides(1-20)

My learning from the course and project:

A. Project:
    1. Problem 1: How the modified Newton Rapson's method can be used to implement methods that finds extrema(s) and then maxima(s) and minima(s) explicitly and then the most fundamental optimizer of artificial neural nets i.e. the gradient descent which is the dominant front phase of today's technical developments which are in Artificial Intelligence.
    2. Problem 2: Use of secant method in real world problems (and the reason)

3. Problem 3: Comparing some of the interpolating methods with some of the benchmarking ML models and learned by the accuracy comparison
4. Problem 4: Optimizing(time) the most optimized algorithms learned in the course further by choosing the right choice of HW, HW-accelerators, and sequence of operations which optimizes the algorithms inline through processors' architectures and specialized architectures for specific operations cumulatively (Matrix operations).
5. Problem 5: Obtaining the values of some the fundamental constants and comparing them with the exact values stored optimally in the systems

B. Course:

I learned a lot from this course, especially the different methods to solve particular problems for e.g. finding polynomial coefficients that satisfy the given data points.

Other than learning the theory, I learned how to derive a better solution to the given problem by using slight variation of the solution existing and then further following the same steps to arrive the best solution possible, in short I like the way we arrive at the nearmost(most optimized) method to solve a particular method. For example: we started solving a system of linear equation by taking them

as product Ax = b problem, then initially we solved it by inverting x directly, then we slightly changed the inverting process to a set of triangle reduction which we called as gauss elimination, then we arrived at forming A = LU matrices and solving them in linear time, then we slightly change this method to solve problem iteratively instead of explicit operations to gain the stability and confirmed accuracy also from simple(Richardson’s method) to the most optimized one(Gauss Seidel/SoR method).