Johns Hopkins Engineering

Applied Machine Learning for Mechanical Engineers

Optimization, Part 1, D

Johns Hopkins
Whiting School of Engineering

Introduction to Optimization Algorithms

- By the end of this lecture you will be able to:
  - Describe some issues with gradient calculations.
  - Describe concepts behind some mathematical optimization problems.
  - Describe concepts behind some nature-inspired optimization problems.

Introduction to Optimization Algorithms

- Gradients calculation costs
  - Complicated objective functions.

$$f(x_1, x_2) = s\left(s\left(\sin\left(\frac{x_1}{x_2}\right) \times \sin\left(\frac{x_2}{x_1}\right)\right)\right) \qquad (1-9)$$
$$where\ s(x) = \frac{1}{1+e^{-x}}$$

  - More number of variables.
  - Computation and memory costs.

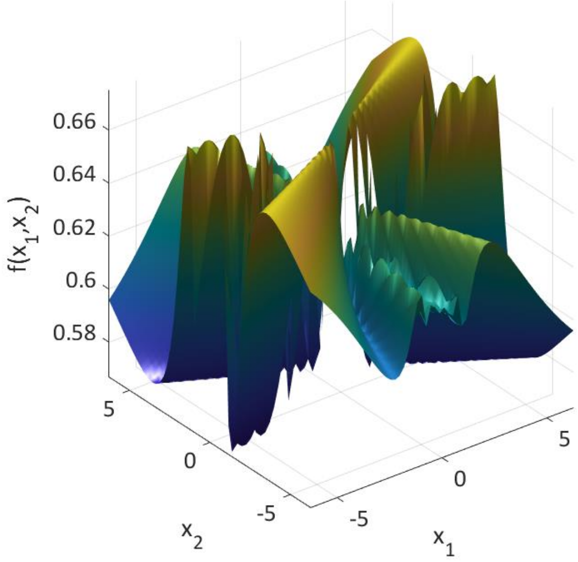

Figure 1-6 An example of a complicated objective function

Introduction to Optimization Algorithms

- Mathematical optimization algorithms are less computation-intensive, but usually iterative approaches to get to optimum solution.

- Some mathematical optimization algorithms comprise a gradient function and a penalty function to estimate the optimum solution iteratively.

  - The gradient function is to make the iterative algorithm gradually closer to the optimum solution.

  - Penalty function is to minimize any violation of constraints during the iterative procedure.

Introduction to Optimization Algorithms

- Neural Dynamic Optimization

Gradient Part Penalty Part

$$\boldsymbol{X}^s = \int \left\{ \boxed{-\nabla f(\boldsymbol{X}^{s-1})} \boxed{- R\left[\sum_{j=1}^{J} g_j^+(\boldsymbol{X}^{s-1}) \times \nabla g(\boldsymbol{X}^{s-1}) + \sum_{v=1}^{V} h_v(\boldsymbol{X}^{s-1}) \times \nabla h_v(\boldsymbol{X}^{s-1})\right]} \right\} ds \qquad (1-10)$$

where $\boldsymbol{X}^s$ is the solution at the s^{th} iteration, $f(\boldsymbol{X}^{s-1})$ is the value of objective function at $\boldsymbol{X}^{s-1}$, R is the penalty parameter, $g_j^+(\boldsymbol{X}^{s-1}) = max\{0, g_j(\boldsymbol{X}^{s-1})\}$, $g_j(\boldsymbol{X}^{s-1}) \leq 0$ is the j^{th} inequality constraint, and $h_v(\boldsymbol{X}^{s-1}) = 0$ is the v^{th} equality constraint. This integration is being solved in each iteration numerically using methods such as Runge-Kutta.

- More iterations make $\boldsymbol{X}^s$ closer to the optimum solution.

Adeli, H., & Park, H. S. (1995). A neural dynamics model for structural optimization—theory. Computers & structures, 57(3), 383-390]

Introduction to Optimization Algorithms

- Nature-inspired optimization algorithms are inspired by natural phenomena such as evolutionary theories, chemical reactions, animal behaviors, etc.

  - Genetic algorithm (GA) is inspired by the process of natural selection based on the concept of Darwin's theory of evolution.

  - Artificial bee colony (ABC) algorithm is inspired by foraging behavior of honeybees.

  - Particle Swarm Optimization (PSO) inspired by social behaviors and movements of animal, fish, or birds.

Introduction to Optimization Algorithms

- GA
  - A higher chance of survival of the bests in a population of a generation to generate the next generation.
    - Those tigers that run faster have a chance of getting more foods/prey and survive to create the next generation of tigers.
    - Those deers that run faster have a chance of not being hunted by the tigers and survive to create the next generation of deers.

Introduction to Optimization Algorithms

- GA
  - “Best” means best fitness (objective) value.
  - Say $N = 1$ in Eq. 1-1, then $\boldsymbol{x} = [x_1]$ is a chromosome of a tiger. A chromosome is made of binary genes.

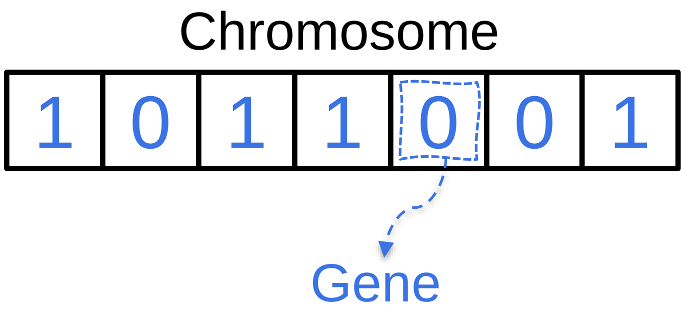

Figure 1-7 An example of a chromosome representation in GA

Introduction to Optimization Algorithms

- GA
  - Population of 8 tigers (chromosomes) along with their corresponding fitness (objective) values.
  - Say F_2, F_3, F_5, F_8 are the smallest fitness values amongst all 8 chromosomes.

A Population of 8 Chromosomes

$F_1 = f(\boldsymbol{x}_1) \leftarrow$	0	1	0	0	0	0	1
$F_2 = f(\boldsymbol{x}_2) \leftarrow$	1	0	1	1	0	1	1
$F_3 = f(\boldsymbol{x}_3) \leftarrow$	0	0	1	0	1	0	1
$F_4 = f(\boldsymbol{x}_4) \leftarrow$	1	1	1	1	0	0	1
$F_5 = f(\boldsymbol{x}_5) \leftarrow$	0	1	0	1	0	1	1
$F_6 = f(\boldsymbol{x}_6) \leftarrow$	1	1	1	0	1	1	0
$F_7 = f(\boldsymbol{x}_7) \leftarrow$	1	0	0	1	0	0	1
$F_8 = f(\boldsymbol{x}_8) \leftarrow$	0	0	1	1	0	0	1

Figure 1-8 An example of 8 chromosomes to form a population in GA

Introduction to Optimization Algorithms

- GA
  - Crossover
  - Mutation
  - Iterative

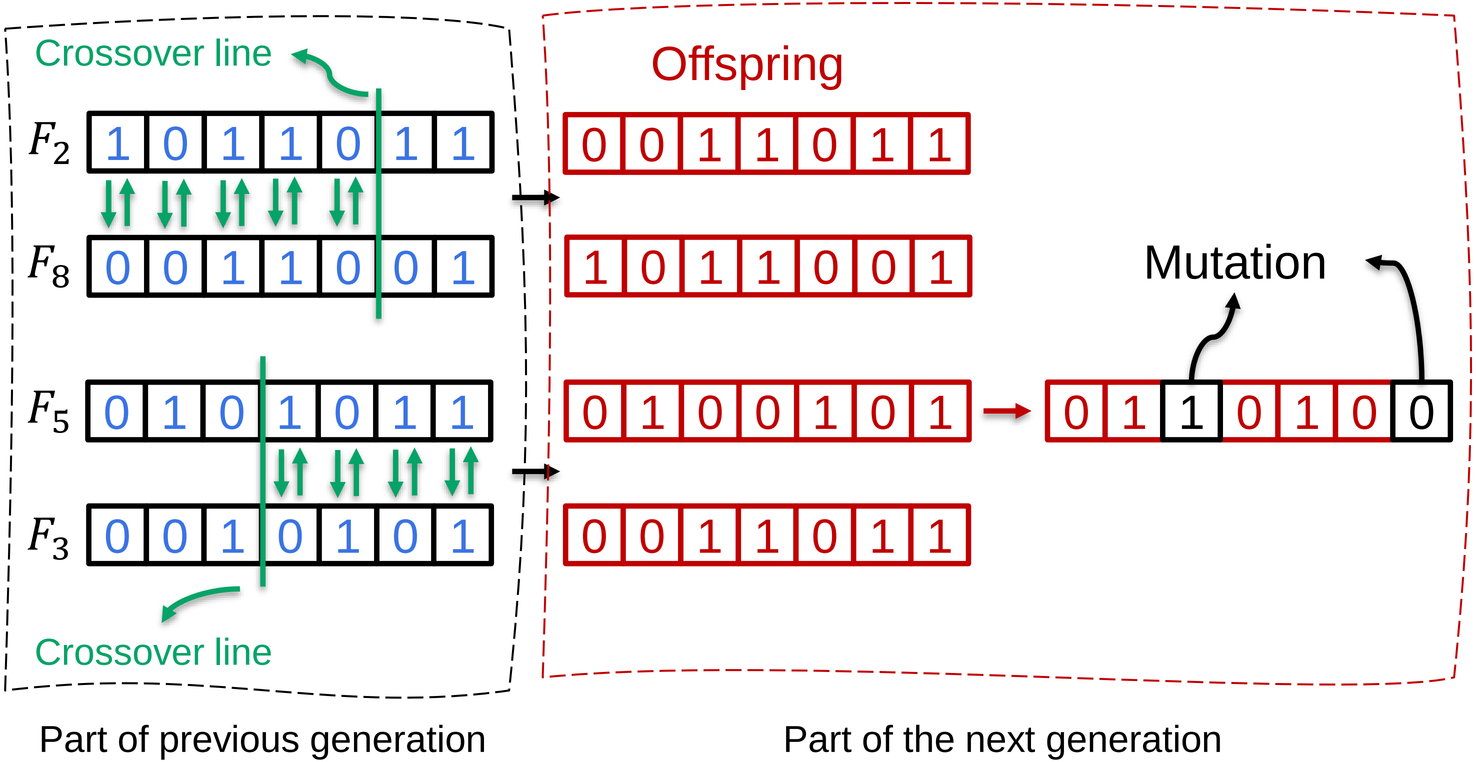

Figure 1-9 An example of crossover and mutations to create the population of the next generation using the best chromosomes from the previous generations

Introduction to Optimization Algorithms

- Mathematical and nature-inspired optimization algorithms.
  - Computationally efficient.
  - Number of parameters to manage their efficiencies.
  - Iteratively getting close to optimum solution.
  - Rely on initial point (e.g. $\boldsymbol{X}^{s=0}$) for large number of variables; might be trapped in a local optima.

Introduction to Optimization Algorithms

- In this lecture, you learned about:
  - Some issues with gradient optimization.
  - Concepts behind some mathematical optimization problems.
  - Concepts behind some nature-inspired optimization problems.
- In the next module, we will go over available programming packages for optimization problems.

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING