Computational Intelligence

Unit # 12

Artificial Immune Systems (Source: Wikipedia)

- The field of Artificial Immune Systems (AIS) is concerned with abstracting the structure and function of the immune system to computational systems, and investigating the application of these systems towards solving computational problems from mathematics, engineering, and information technology.
- AIS is a sub-field of Biologically-inspired computing, and Natural computation, with interests in Machine Learning and belonging to the broader field of Artificial Intelligence.

History

- AIS began in the mid 1980s with Farmer, Packard and Perelson's (1986) and Bersini and Varela's papers on immune networks (1990).
- However, it was only in the mid 90s that AIS became a subject area in its own right.

Inspiration

- AIS is inspired by the working of immune systems.
- The immune system is comprised of an intricate network of specialized tissues, organs, cells and chemical molecules.
- The capabilities of the natural immune system include the ability to recognize, destroy and remember an almost unlimited number of pathogens (foreign or non-self objects that enter the body, including viruses, bacteria, multi-cellular parasites, and fungi), and also to protect the organism from misbehaving cells in the body.

Working of Immune System

- To assist in protecting the organism, the immune system has the capability to distinguish between self and non-self.
- Critically, the system does not require exhaustive training with negative (non-self) examples to make these distinctions, but can identify a pathogen as being non-self even though it has never been encountered before.

Working of Immune System (Cont'd)

- Both the innate and acquired immune systems are compromised of a variety of molecules, cells and tissues.
- The most important cells are leukocytes (white blood cells) which can be divided into two major categories: phagocytes and lymphocytes.
- Lymphocytes circulate constantly through the blood, lymph, lymphoid organs and tissue spaces.
- A major component of the population of lymphocytes is made up of B and T cells.

Working of Immune System (Cont'd)

- These cells are capable of recognizing and responding to certain antigen (foreign molecules) patterns presented on the surface of pathogens.
- Antigens are not the invading pathogens themselves, rather they are molecular signature expressed by the invading pathogen.
- The control of adaptive immunity can be divided into two branches: *humoral immunity* which is controlled by B-cells, and *cellular immunity* which is controlled by T-cells.

Key Terms

Component	Definition
Pathogens	Foreign bodies including viruses, bacteria, multi-cellular parasites, and fungi.
Antigens	Foreign molecules expressed by a pathogen that trigger an immune system response.
Leukocytes	White blood cells, including phagocytes and lymphocytes (B and T cells) for identifying and killing pathogens.
Antibodies	Glycoproteins (protein+carbohydrate) secreted into the blood in response to an antigenic stimulus that neutralise the antigen by binding specifically to it.

Destruction Process

- Antigen-secreting pathogen enters the body.
- B-cells are activated by the foreign antigen.
- With help of T-cells, B-cells undergo cloning and mutation.
- Plasma B-cells secrete immunoglobulins which attach to the antigen.
- Marked antigens are attacked by the immune system.
- Memory of the antigen is maintained by B memory cells.

Metaphor

- Once the immune system detects a pathogen, B-cells begin to clone at a rate proportional to their affinity to the antigen that stimulated them.
- Adopting this metaphor in order to design an optimization algorithm, the antibody can be considered as a potential solution, the antigen is a test dataset, and the degree of binding or fit between the antibody and the antigen represents the fitness or the quality of the solution.
- The objective, therefore, is to start from an initial population of solutions, test them against the dataset, and, using the algorithm iteratively, improve the quality of the solutions in the population.

AIS Algorithm

- Create an initial random population P of solution vectors (antibodies).
- Select a subset F of the solutions from P, biasing the selection process towards better solutions.
- For each member of F (the parents), create a set of clones, with better members of F producing more clones.
- Mutate each of these clones, in inverse proportion of their parent's fitness. Better solutions are mutated less.

AIS Algorithm (Cont'd)

- Select a subset of the newly generated solutions S.
- Create a number of newly created random solutions R.
- Replace poorer members of P with better solutions from S and R.

AIS Algorithm

(Source: http://www.artificial-immune-systems.org)

```
input   : S = set of patterns to be recognised, n the number of worst elements to select for removal
output  : M = set of memory detectors capable of classifying unseen patterns

begin

  Create an initial random set of antibodies, A

  forall  patterns in S   do
    Determine the affinity with each antibody in A
    Generate clones of a subset of the antibodies in A  with the highest affinity.
      The number of clones for an antibody is proportional to its affinity
    Mutate attributes of these clones to the set A , and place a copy of the highest
      affinity antibodies in A  into the memory set, M
    Replace the n lowest affinity antibodies in A with new randomly generated antibodies
  end

end
```


AIS Algorithm (For Assignment)

- Create an initial random population P of size **10**.
- Select a subset F (**size 4**) from P, biasing the selection process towards better solutions. (**use either FPS or RBS**)
- Create a set of clones from F, with better members of F producing more clones. (**use FPS or RBS**)
- Mutate each of these clones, in inverse proportion of their parent's fitness. Better solutions are mutated less.

AIS Algorithm (For Assignment)

- Select a subset of the newly generated solutions S. (**size 2 using Binary Tournament)**
- Create a number of newly created random solutions R. (**size 2**)
- Replace poorer members of P with better solutions from S and R. (**now you would have 10+2+2 = 14 solutions**).
- Apply **truncation** to pick the top 10 solutions.