COMP815 Nature Inspired Computing

Overview of the Course

Timetable and Assessment

- Assessment in *Assignment* Section
  Part 1: coding + analysis
  Part 2: project report + portfolio

- Course Format: 1 hour Lecture + 1 hour Workshop
- Timetable and Deadlines in *Announcements*

- Q&A: preferably *Teams*

Any Question so far?

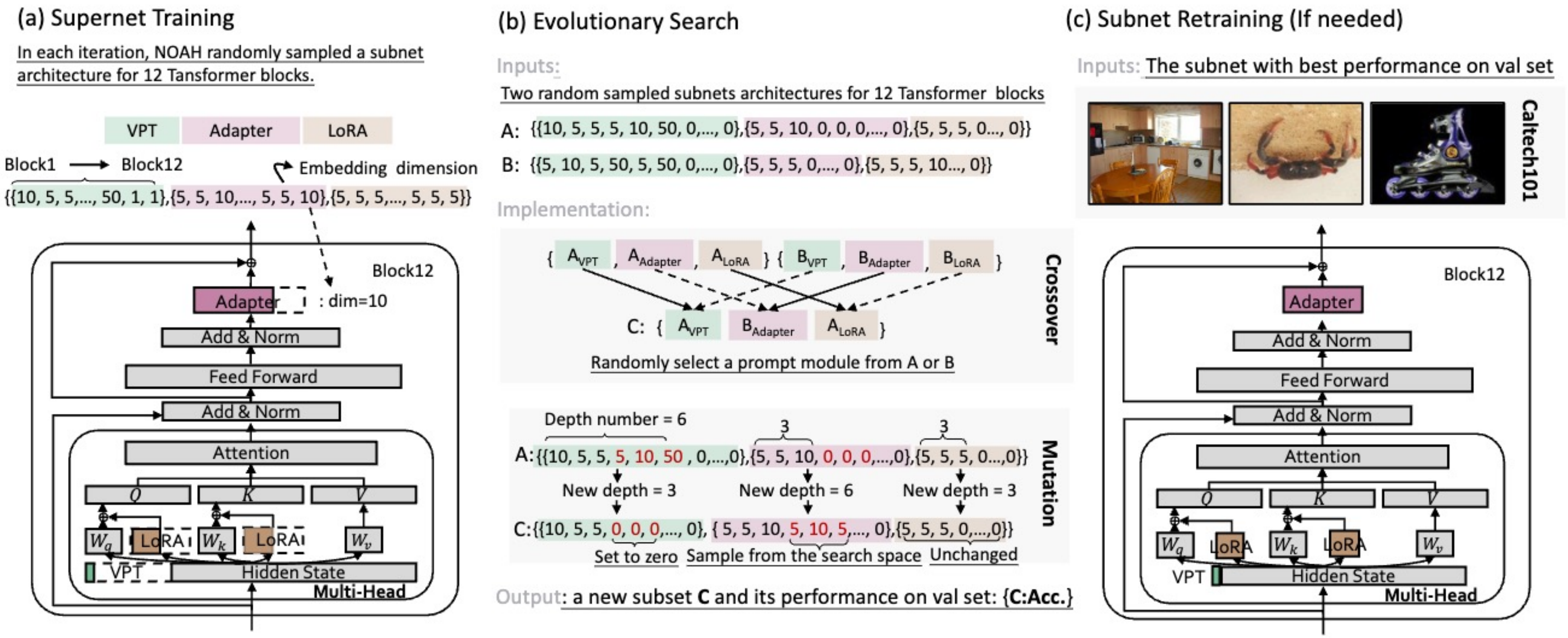
Course Objectives

- Introduce a different kind of algorithms to solve problems that are difficult to solve using conventional algorithms
- Algorithms that are inspired by nature that contains an element of randomness
- Focus on
  - Evolutionary and genetic algorithms   Biologically inspired
  - Swarm algorithms   Socially inspired
  - Neural Networks   Brain inspired

Motivating Videos/Demos

Evolutionary Search for Neural Networks

– Neural Prompt Search, *Nanyang Technological University*



Motivating Videos/Demos

Swarm Intelligence

Swarm Drones [credit to DSO National Laboratories, Singapore]:
Link

Five Principles of Swarm Intelligence [credit to Alfonso]:
Link

Motivating Videos/Demos

Artificial Neural Networks

How AI mimics Human Brain [credit to TheAIAwakening]:
Link

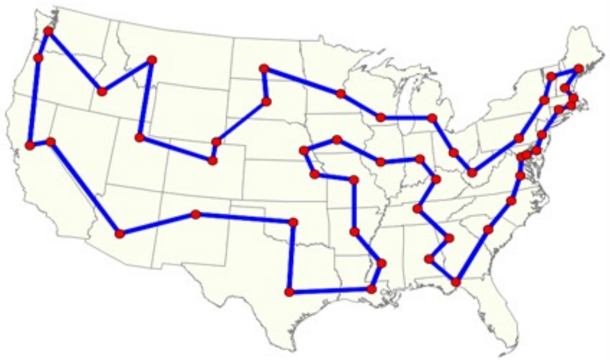
Any Question so far?

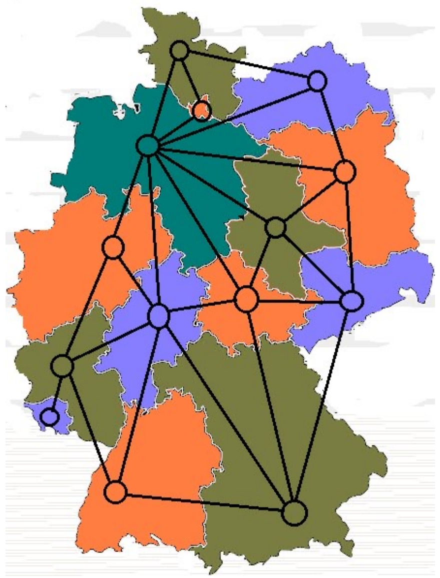
COMP815 Nature Inspired Computing

Problem Formulation

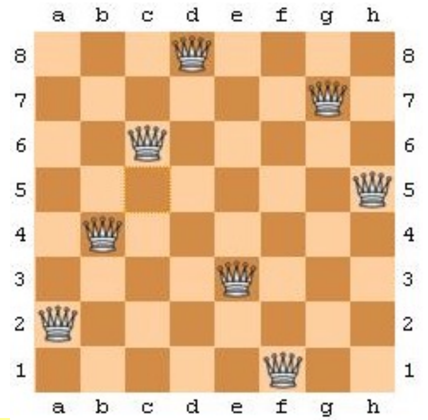
Examples of Problems

- Timetabling
- Scheduling
- Engineering design





- Generic problems with many applications:
  - Travelling salesman problem (TSP)
  - Eight-queens or N-queens problem
  - Graph colouring problem



Further Examples

- Machine learning problems
  - Determining a statistical model for a given set of data
  - Training an artificial neural network
- Models are used for prediction or inference or control
  - Object recognition
  - Adaptive control
  - Outcomes of changes in economic policies

These problems can be viewed as optimization problems

A search through the feasible space for the best solution

Mathematical Description

Objective function

Minimize $f(\boldsymbol{x})$

parameters

$\boldsymbol{x} = [x_1, x_2, \dots, x_N]$

Subject to $\boldsymbol{x} \in M$

Feasible set

M is finite = discrete optimization

M is uncountably infinite = continuous optimization

Constraints

The feasible set M is often given by a system of equations and inequalities:

Equality constraints	$g_j \leq 0$	$j = 1,2, \ldots, m$
Inequality constraints	$h_k = 0$	$k = 1,2, \ldots, n$

Problem Classification

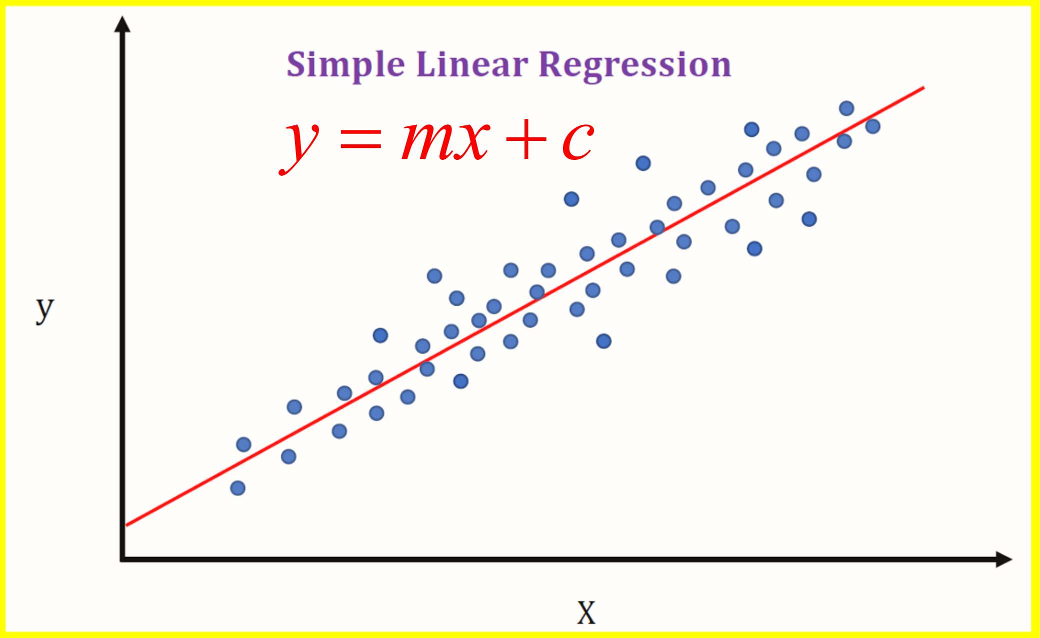
	Objective function	
Constraints	**Yes**	**No**
Yes	Constrained optimization problem	Constraint satisfaction problem
No	Unconstrained optimization problem	

Linear Regression

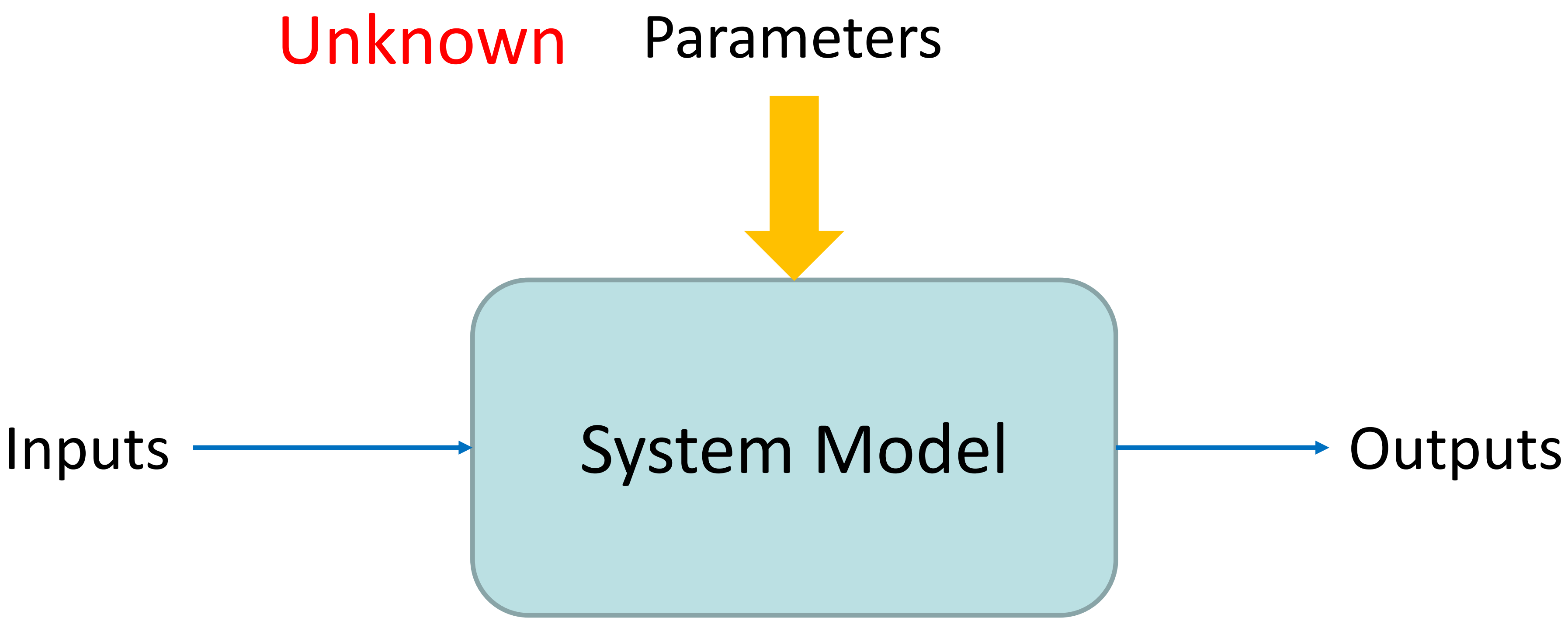
Given a set of data $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
Find their linear dependence

$$\underbrace{\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} c \\ m \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}}_{y}$$

$$\min_{A} \parallel XA - y \parallel$$

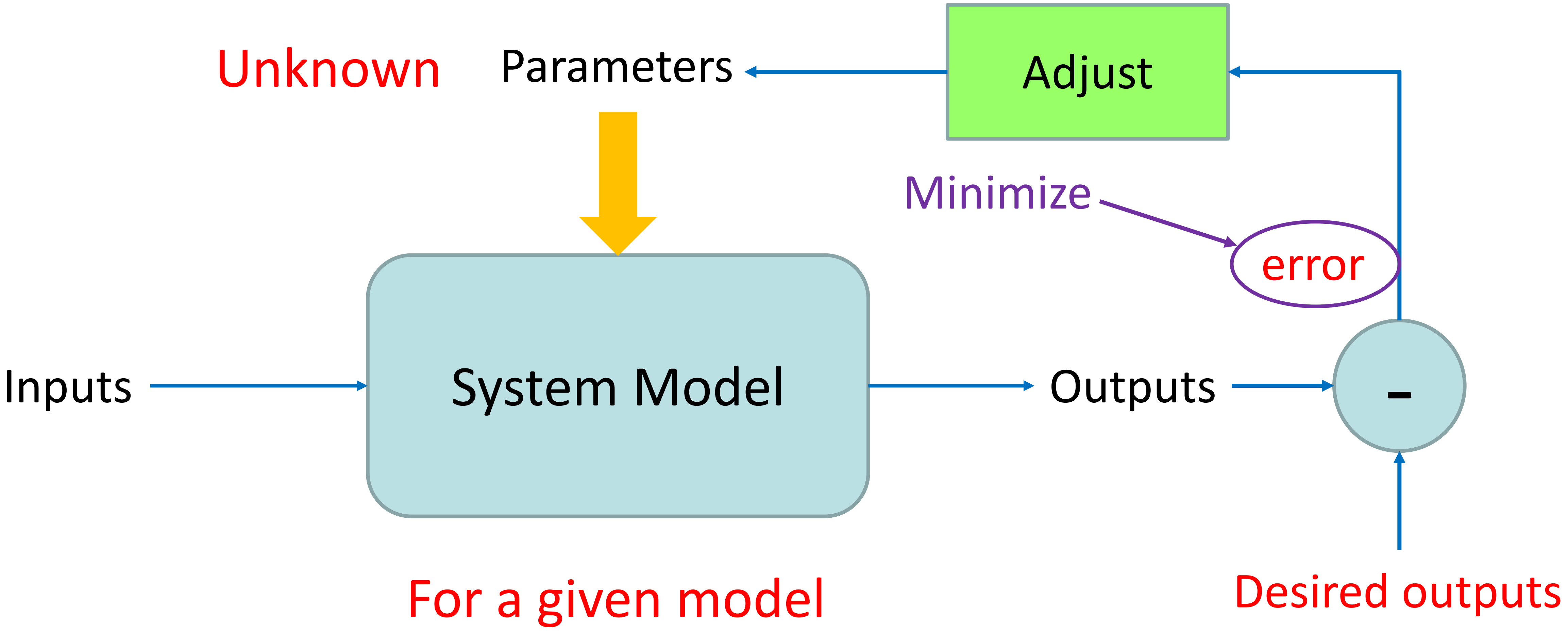


Modelling Problem



For a given model

Optimization Problem



Levels of Difficulty of Problems

- P: problem can be solved in polynomial time (a polynomial function of the size of the problem)
- NP: problem can be solved, and any solution can be verified in polynomial time by another algorithm
- NP-complete: problem is NP and any other problem in NP can be reduced to this problem in polynomial time by an algorithm
- NP-hard: problem is at least as hard as any other NP-complete problem, but solution may not be verified in polynomial time

Many practical problems turn out to be NP-complete

Almost impossible to find the optimum solution

Approximate solutions

Summary

- Assessments, Timetable, Key dates
- Scope of this course
- Optimisation problem