Introduction to Artificial Intelligence

COMP307/AIML420

Neural Engineering and Evolutionary Computation: Tutorial

Dr Fangfang Zhang

fangfang.zhang@ecs.vuw.ac.nz

COMP307/AIML420 Week 4 (Tutorial)

1. Announcements
   - Assignment 2 (**12%**)
   - Due on 27th April (the week after teaching break)
   - Helpdesk: Monday to Friday, 4-5pm
   - Helpdesk available during the teaching break

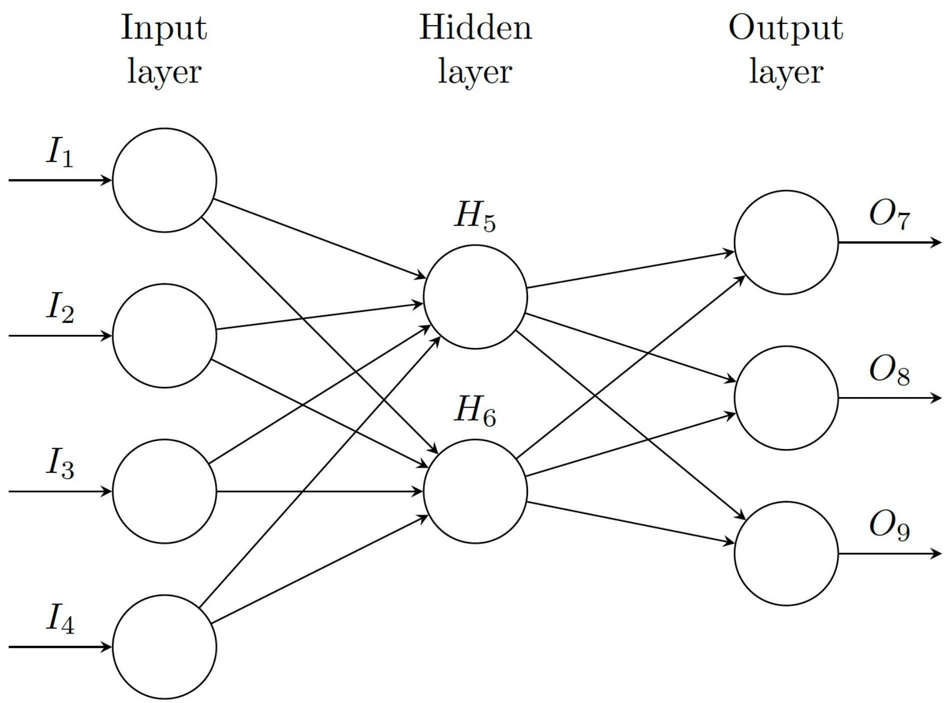
2. Neural Engineering

   Part 1

3. Evolutionary Computation
   - Genetic programming

   Part 2

Neural Engineering



Weight Update Frequency

- All the weights are updated **after** one feedforward pass and one backward propagation/pass
- Frequency of weight update = Frequency of passes

- **Online learning**: a pass for each training instance
- **Batch learning**: a pass for a batch (a subset of training instances)
  - weight change is the sum of the changes for all the instances in the batch
- **Offline learning**: a pass for all the training instances
  - Weight change is the sum of the changes for all training instances

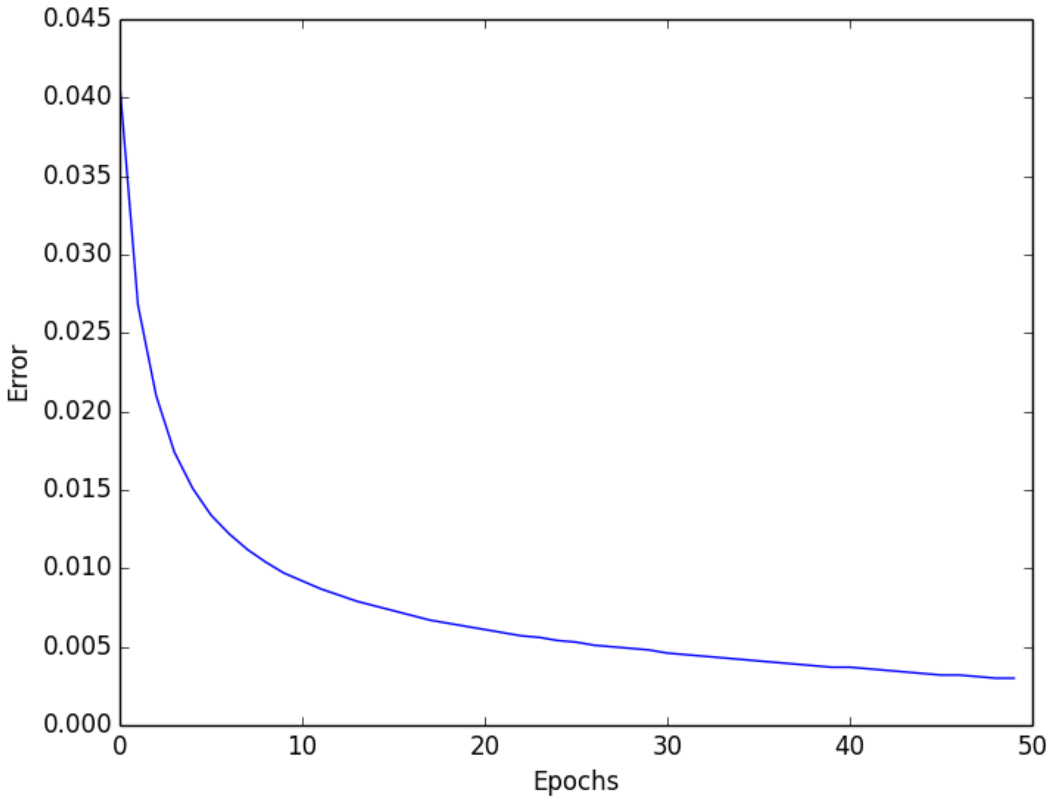
Weight Update Frequency

- Assuming a weight $w = 0.2$
- 4 training instances
- Online learning
  - Instance 1, $\Delta w = 0.1$, $w \to 0.3$
  - Instance 2, $\Delta w = 0.05$, $w \to 0.35$
  - Instance 3, $\Delta w = 0.03$, $w \to 0.38$
  - Instance 4, $\Delta w = 0.01$, $w \to 0.39$
- Offline learning
  - Instance 1, $\Delta w = 0.1$, $w = 0.2$ unchanged
  - Instance 2, $\Delta w = 0.08$, $w = 0.2$ unchanged
  - Instance 3, $\Delta w = -0.03$, $w = 0.2$ unchanged
  - Instance 4, $\Delta w = 0.05$, $w = 0.2$ unchanged
  - $w \to 0.2 + 0.1 + 0.08 - 0.03 + 0.05 = 0.4$

Weight Update Frequency

- Epoch: period when all the training instances are used once
- #Iterations = #passes

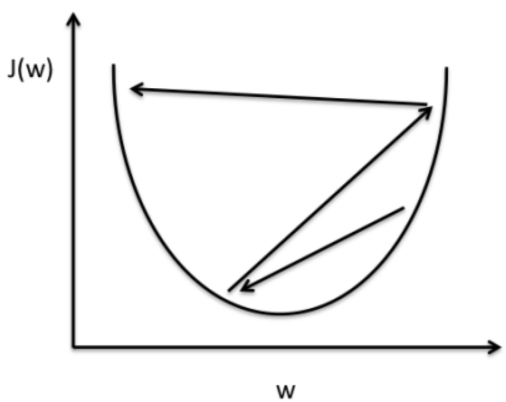
- 1000 training instances, batch size = 500, then need 2 iterations to complete **one** epoch



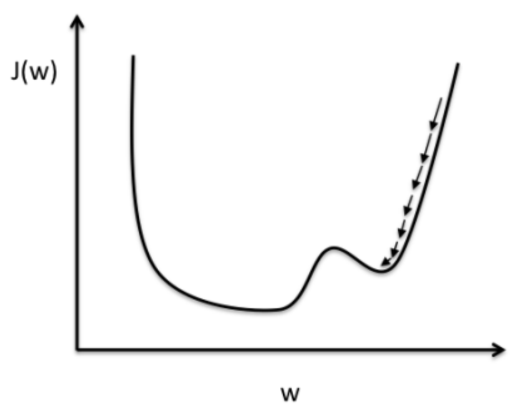
Learning Rate

- Large learning rate may cause oscillating behaviour
- Small learning rate may cause slow convergence
- 0.2 is a good starting point in practice

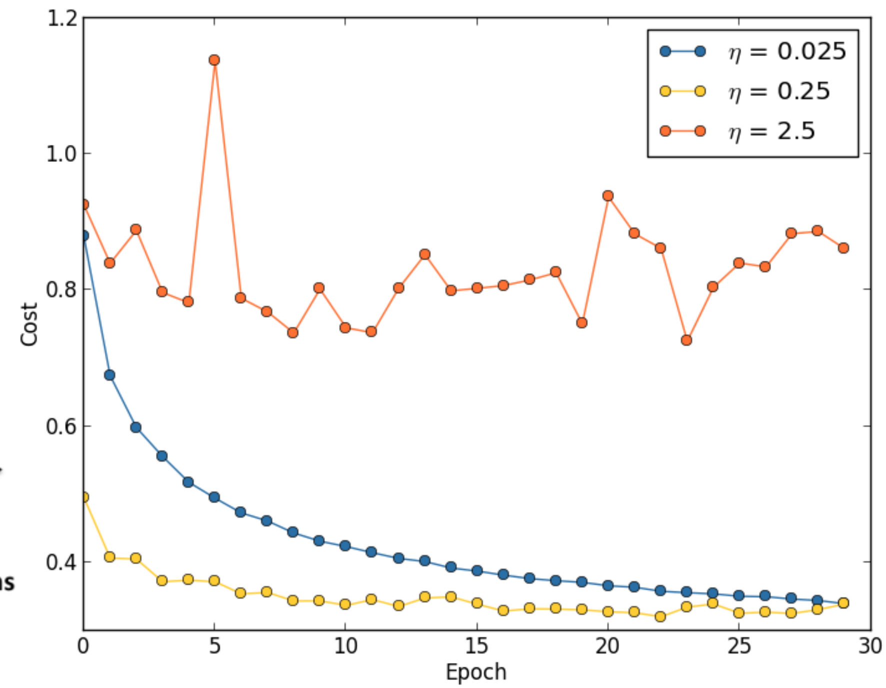
$$\boxed{\Delta w_{i\rightarrow j} = \boldsymbol{\eta} o_i o_j (1 - o_j)\beta_j}$$



Large learning rate: Overshooting.



Small learning rate: Many iterations until convergence and trapping in local minima.



Evolutionary Computation

- Evolutionary algorithms
  - Genetic algorithms (the biggest branch)
  - Evolutionary programming
  - Evolutionary strategies
  - Genetic Programming (Koza, 1990s, fast growing area)

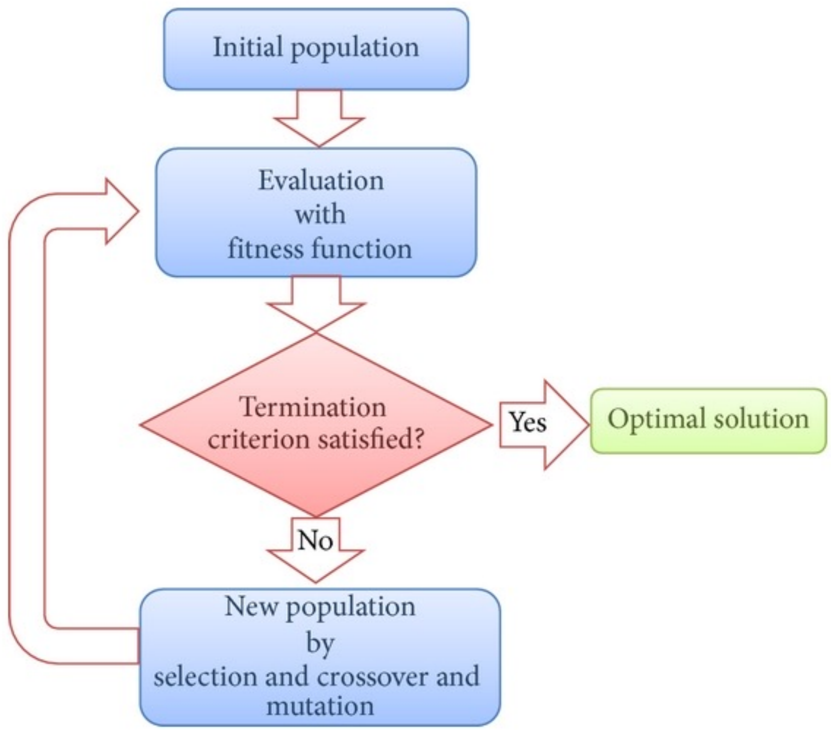
- Swarm intelligence
  - Ant colony optimisation
  - Particle swarm optimisation (PSO)
  - Artificial immune systems

- Other techniques
  - Differential evolution
  - Estimation of distribution algorithms
  - …

Evolutionary Computation

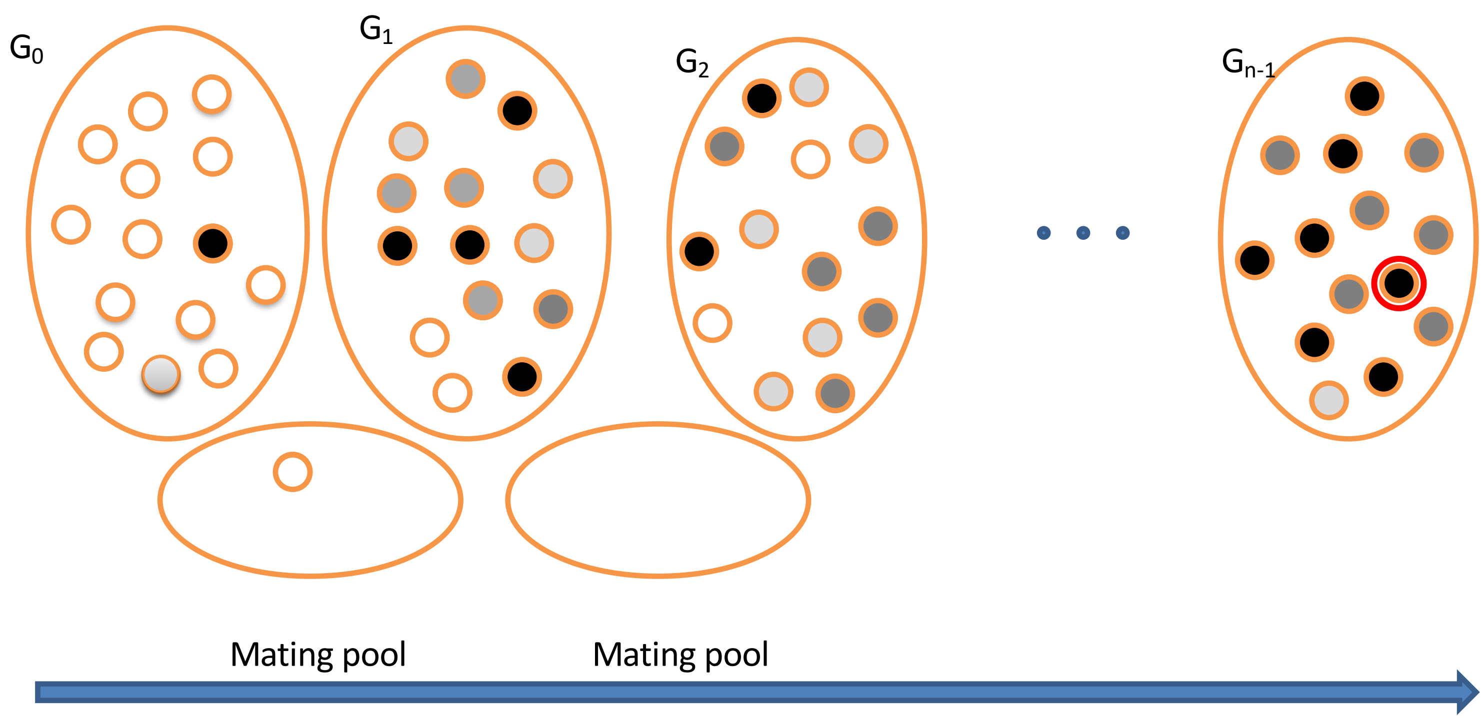
- Three steps of learning

Define function → goodness of function → pick the best function



Evolutionary Algorithms

- Darwinian biological evolution principle
  - Representation is problem dependent
  - Fitness function: goodness measure
  - Selection: fitter individuals are more likely to survive and reproduce
  - Genetic operators: to generate new individuals (crossover, mutation)



A Basic Genetic Algorithm

- Randomly initialise a population of chromosomes

- **Repeat until** stopping criteria are met:
  - Construct an empty new population
  - **Repeat until** the new population is full:
    - Select **two** parents from the population by roulette wheel selection
    - Apply crossover to the two parents to generate two children
    - Each child has a probability (mutation rate) to undergo mutation
    - Put the two children into the new population
  - **End Repeat**
  - Move to the new population (new generation)
- **End Repeat**

- Output the best individual from the final population

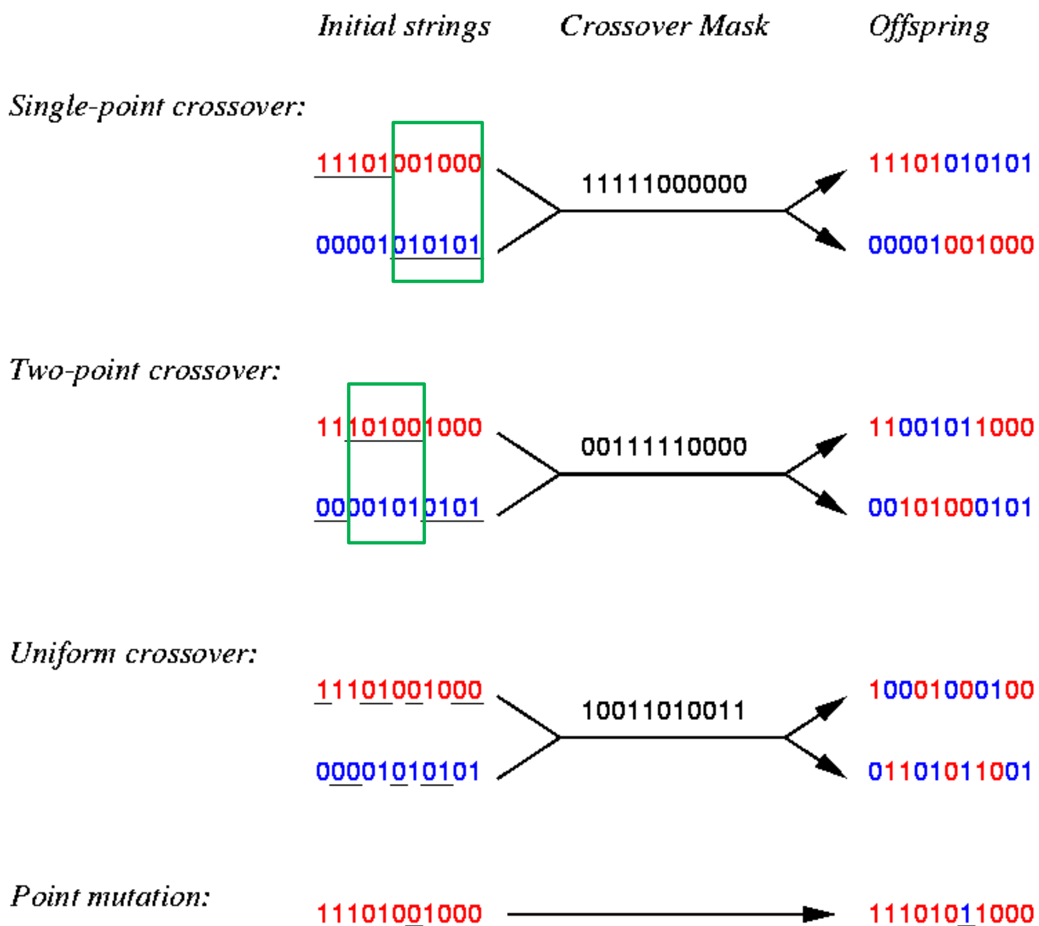
A Simple GA Example

- **OneMax Problem**
  - Target to (11111…1)
  - More zeros means worse: far away from the target
  - Simple "benchmark" problem!

- Representation: bit string
- Fitness function: $1 + \sum x_i$ (the larger, the better)
- Crossover: single-point crossover
- Mutation: point mutation

Genetic Algorithm

- Representation: individuals are binary strings
- An individual is also called a chromosome



A Simple GA Example

- 10 bits (Optimal fitness = 11)
- population size = 20
- Run for 10 generations

```
At generation 0 average fitness is 6.0, best fitness is 9
At generation 1 average fitness is 6.65, best fitness is 10
At generation 2 average fitness is 6.8, best fitness is 11
At generation 3 average fitness is 6.9, best fitness is 9
At generation 4 average fitness is 6.45, best fitness is 9
At generation 5 average fitness is 6.95, best fitness is 9
At generation 6 average fitness is 7.3, best fitness is 11
At generation 7 average fitness is 6.65, best fitness is 10
At generation 8 average fitness is 6.25, best fitness is 8
At generation 9 average fitness is 6.6, best fitness is 8
```

Keep elites (i.e., best ones) to the next generation!!!

Summary

- Neural engineering
- Evolutionary computation

- Next week

--- Genetic programming (next Monday)

--- GP for Regression and Classification (next Tuesday)