UNIVERSITY OF LIVERPOOL

COMP 532
Machine Learning and BioInspired Optimization

RECAP

Dr. Shan Luo

Department of Computer Science

shan.luo@liverpool.ac.uk

Admin info

- Exam
  - COMP 532
  - Machine Learning and BioInspired Learning Optimisation
  - Date: 31 May 2018; Start Time: 10:00; Exam Room: 202 Brodie Tower, Room 107; Duration: 180 mins
  - Chief invigilator:

    Jacopo Castellini (J.Castellini@liverpool.ac.uk)

Admin info

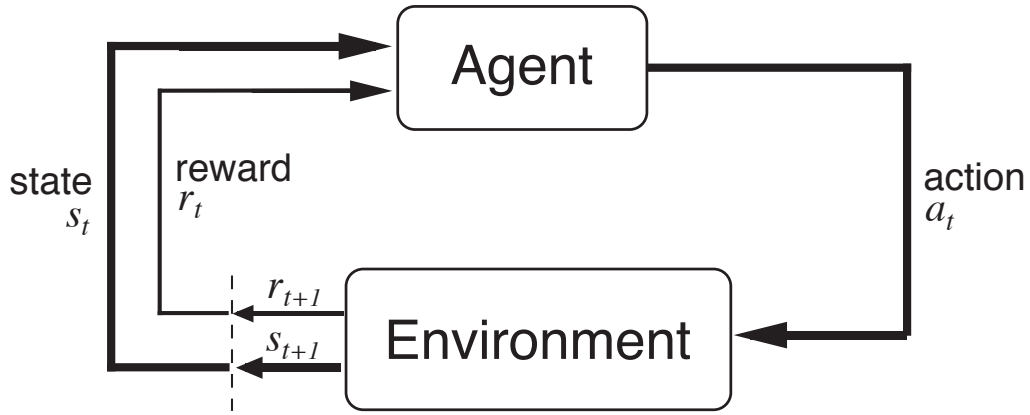
- Office hours
  - Tuesdays 14:00-15:00
  - **BUT not today**
  - Other time: drop me an email for an appointment (shan.luo@liverpool.ac.uk)

Single Agent Reinforcement Learning

- Supervised vs Unsupervised learning
  - Reinforcement Learning: somewhat in between
- Environments:
  - Fully observable vs partially observable
  - Deterministic vs stochastic
  - Episodic vs. sequential
  - Dynamic vs. static
  - Discrete vs. continuous
  - Single agent vs. multi-agent

Single Agent Reinforcement Learning

- Agent-Environment Interface



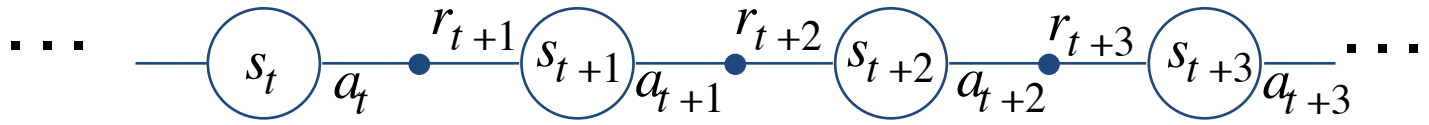
Single Agent Reinforcement Learning

- Markov Decision Process
  - States S
  - Actions A
  - Reward function $S \times A \rightarrow \mathbb{R}$
  - Transition function $S \times A \times S \rightarrow [0, 1]$
  - **Markov Property**: all relevant information is present in the current state

$$Pr\{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, \ldots, r_1, s_0, a_0\}$$

$$= Pr\{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t\},$$

for all s' and r, and all histories $s_t, a_t, r_t, s_{t-1}, a_{t-1}, \ldots, r_1, s_0, a_0$



Single Agent Reinforcement Learning

- Learning goal: collect as much reward as possible in the long run
- **Return:** (discounted) sum of future rewards

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1},$$

where $0 \leq \gamma \leq 1$ is the **discount rate**.

shortsighted $0 \leftarrow \gamma \rightarrow 1$ farsighted

Single Agent Reinforcement Learning

- **State value function $V(s)$**: expected return from state s under some policy
- **State-Action value function $Q(s,a)$**: expected return for taking action a in state s and following policy thereafter

- If you know the optimal value function you can compute the optimal (greedy) policy!

Single Agent Reinforcement Learning

- Finding the optimal value function

Model of the environment?

Bootstrap?

	YES	NO
YES	Dynamic programming	Temporal Difference (TD)
NO	_____	Monte Carlo Methods

Single Agent Reinforcement Learning

- Temporal Difference learning method

  ❑ State TD update rule

  $V(s) \leftarrow V(s) + \alpha\big(r + \gamma V(s') - V(s)\big)$

  ❑ Action-sate TD update rule

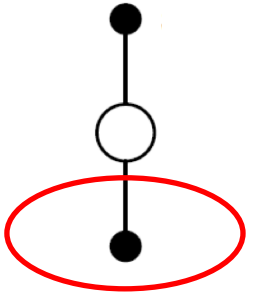
  $\mathrm{Q}(s, a) \leftarrow Q(s, a) + \alpha\big(r + \gamma Q(s', a') - Q(s, a)\big)$

Single Agent Reinforcement Learning

- Temporal Difference learning method
  - ❑ Main algorithms:
    - SARSA (on policy)
    - Q-learning (off policy)
    - R-learning
  - ❑ Action selection:
    - (ε-)Greedy
    - Softmax / Boltzmann

Single Agent Reinforcement Learning

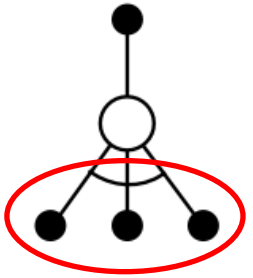
- SARSA vs. Q-learning



Sarsa:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

Q-learning:



$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right]$$

Deep Learning

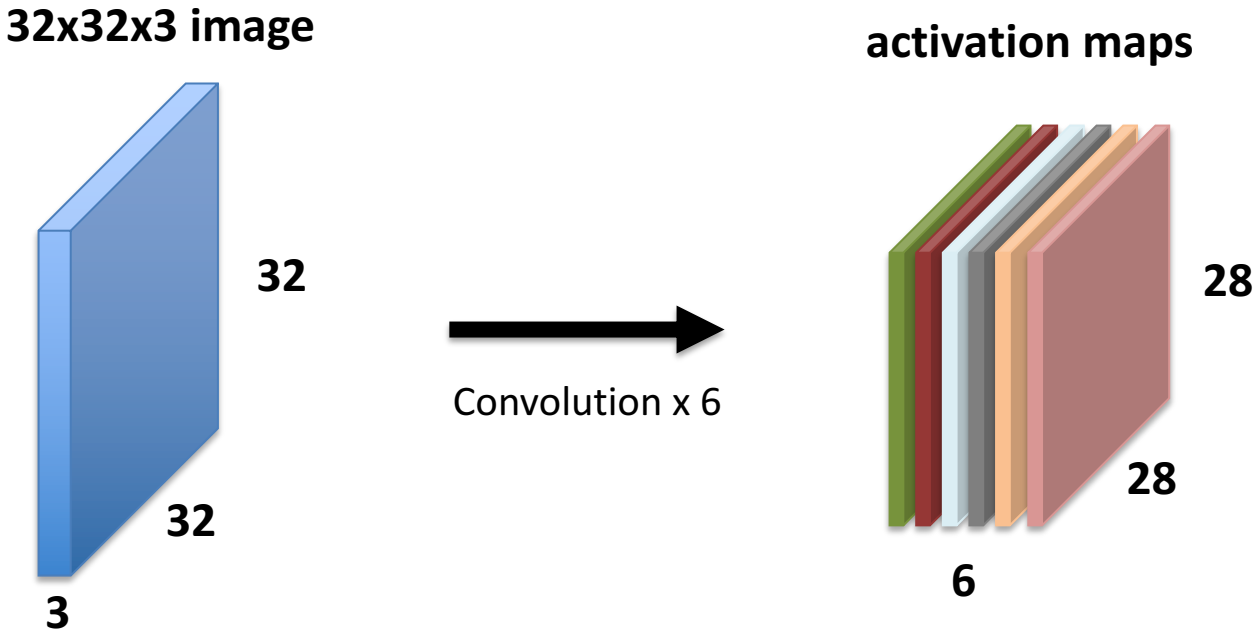
- Artificial Neural Networks
  - Perceptrons (how it works and its limitations)
  - Threshold Logic Units (and how to use them)
  - Linear separability
  - Multi-class classification (a layer of perceptrons)
  - Multi-layer networks
  - Backpropagation (the idea, don't worry about the maths)

Deep Learning

- Convolutional Neural Networks
  - What are they for?
    - Image recognition
    - Other tasks, like Deep Q-learning
  - Why?
    - Learn a hierarchy of features: abstract to complex
    - Similar to how the human visual cortex works

Deep Learning

- Convolutional Neural Networks
  - Convolution layer



Deep Learning

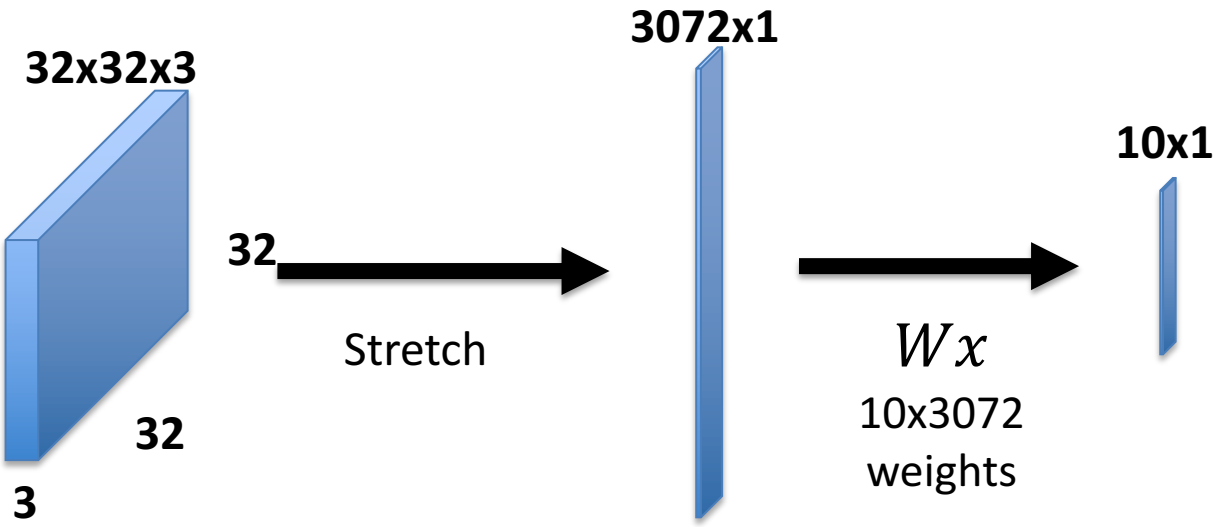
- Convolutional Neural Networks
  - Convolution layer
    - Filter! Reuse of weights.
    - Hyperparameters
      - Number of filters **K**
      - Spatial extent **F**
      - Stride **S**
      - Amount of zero padding **P**
    - Know how to compute output volume dimensions

Deep Learning

- Convolutional Neural Networks
  - Pooling Layer
    - Downsampling, reduce spatial dimensionality
    - Typical: Max Pooling
    - Hyperparameters
      - Spatial extent F
      - Stride S
    - Know how to compute output volume dimensions

Deep Learning

- Convolutional Neural Networks
  - ReLU Layer
  - Fully Connected Layer
    - Produce the final output, just like a normal artificial neural network

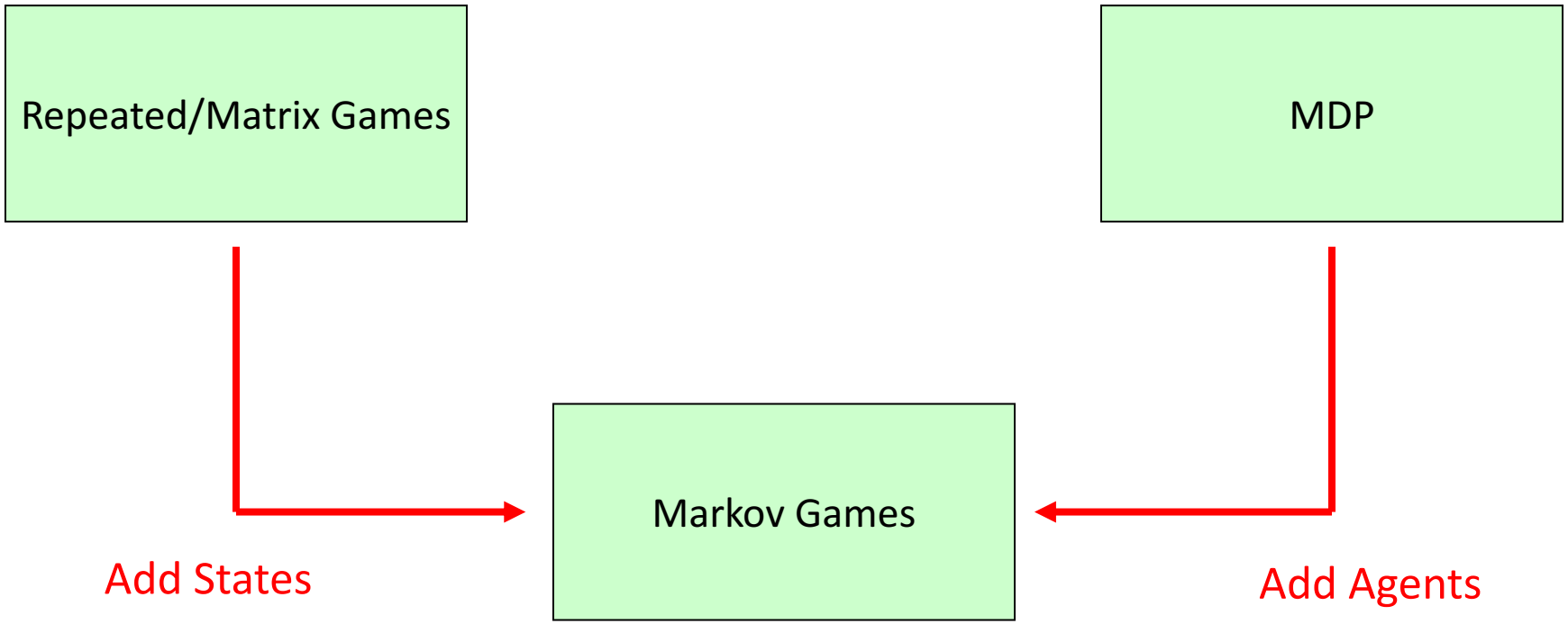


Multiple-Agents Reinforcement Learning

- **Multiple agents learning together**..
  - All agents' actions influence the environment at the same time
- Environment is no longer stationary
  - Markov Property is lost!
- **Cooperative** or **competitive**?

Multiple-Agents Reinforcement Learning

- Stochastic/Markov Games:
- Extension of MDPs to multiple agents
  - **Joint action** defines reward and state transition



Multiple-Agents Reinforcement Learning

- Learning in stochastic games?
  - **Independent Learning** vs **Joint Action Learning**
  - Minimax Q-learning (for zero-sum games)
  - Nash Q-learning (general sum, but needs Nash)
  - Lenient Learning

- **Single state**: matrix games / normal form
  - One shot, perhaps repeated

Multiple-Agents Reinforcement Learning

Game Theory

- Normal form games: payoff matrices
  - Dominant strategies, best response..
  - **Nash equilibrium** = mutual best response

- Assumes rational players that know the game!

- Only explains the outcome, not **how to get there**

Multiple-Agents Reinforcement Learning

- **Evolutionary Game Theory**
  - Players improve over time
  - Modeled as evolving populations

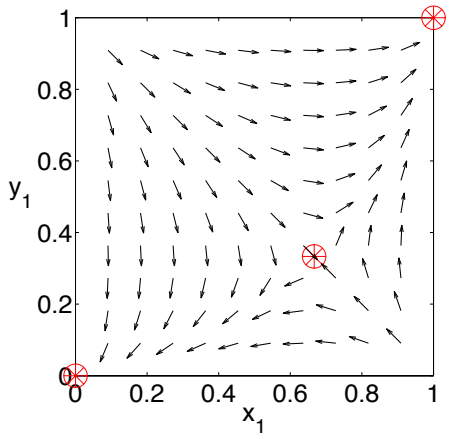
- **Replicator Dynamics**: model the proportional change of types in population as "survival of the fittest"

$$\dot{x}_i = x_i \left[f_i(x) - \sum_j x_j f_j(x) \right]$$

Multiple-Agents Reinforcement Learning

- **Replicator Dynamics** in a matrix game:

$$\dot{x}_i = x_i[(Ay)_i - x^\intercal Ay]$$
$$\dot{y}_i = y_i[(x^\intercal B)_i - x^\intercal By]$$



- **Evolutionarily Stable Strategies**: asymptotically stable fixed points under RD
  - Refinement (subset) of Nash equilibrium!

Multiple-Agents Reinforcement Learning

- The Link between EGT and MARL
  - **Replicator Dynamics** are the continuous time limit of **Cross Learning**
  - Understand this derivation!

- Possible to derive dynamics of various RL algorithms
  - Dynamics predict expected learning behaviour!
  - Can be used to compare RL algorithms
    - in terms of transient behaviour
    - and in terms of convergence

Swarm Intelligence

- Main ingredients of self-organisation:
  - Many interactions, randomness, positive feedback, negative feedback.

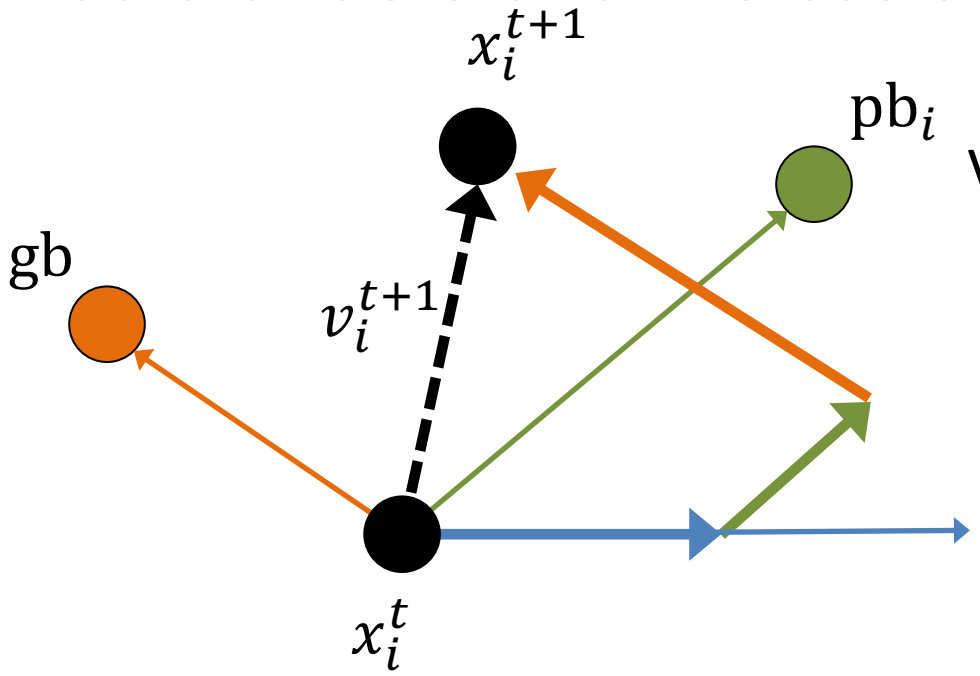
- Main features of Swarm Intelligence (five):
  - Parallelism, stochasticity, adaptivity, use of feedback, autocatalytic.

- Key concept: **Stigmergy**
  - Indirect communication through manipulation of the environment

Swarm Intelligence

Particle Swarm Optimisation

- Loosely based on the flocking of birds
- You should understand the basic algorithm…



Velocity update:

momentum

personal influence

social influence

$$v_i^{t+1} = v_i^t + U(0,\beta_1)\left[\mathrm{pb}_i - x_i^t\right] + U(0,\beta_2)\left[\mathrm{gb} - x_i^t\right]$$

Swarm Intelligence

Ant Colony Optimisation

- Based on the foraging behaviour of ants
  - Deneubourg Bridge experiment

- Understand the main algorithm
  - How ants choose the next path..
    (pheromones and heuristics)
  - Depositing pheromones..
    (only “on the way back”!)
  - Pheromone evaporation
    (to prevent getting stuck in local optimum)

Swarm Intelligence

Stigmergic Landmark Foraging

- Based on honeybees
  - Levy flight and path integration
  - Direct communication!
- Understand the main concepts behind SLF…
  …and its application in robotics.
- Swarm robotics:
  - How to translate biological concepts (e.g. pheromones) to robotic systems?

Artificial Immune Systems

- Know the main algorithms / concepts:
  - **Negative Selection**
    (for anomaly detection, computer security, spam filtering etc)
  - **Clonal Selection**
    (for optimisation)
  - **Somatic Hypermutation**
    (mutation rate inversely proportional to fitness)
- What are the five characteristics of AIS in common with Swarm Systems?

DNA Computing

- Basic procedures:
  - Annealing, PCR, Gel Electrophoresis, Extraction
- Adleman's experiment to solve the HPP/TSP (five steps)
  - Path generation (building blocks)
  - Filtering of correct paths (3 steps)
  - Obtaining the solution
- Understand the problems and limitations

The Bigger Picture

Everything we discussed is related!

- The same ingredients keep coming back:
  - Exploration / exploitation
    (selection / mutation)
  - Trial and error, or many interactions, randomization…
  - Adaptivity
  - Information sharing
    (between interactions, or between individuals / solutions)

ANY QUESTIONS?