UNIVERSITY OF LIVERPOOL

# COMP 532
# Machine Learning and BioInspired Optimization

## Recap: Reinforcement Learning and Deep Learning

Dr. Shan Luo

Department of Computer Science

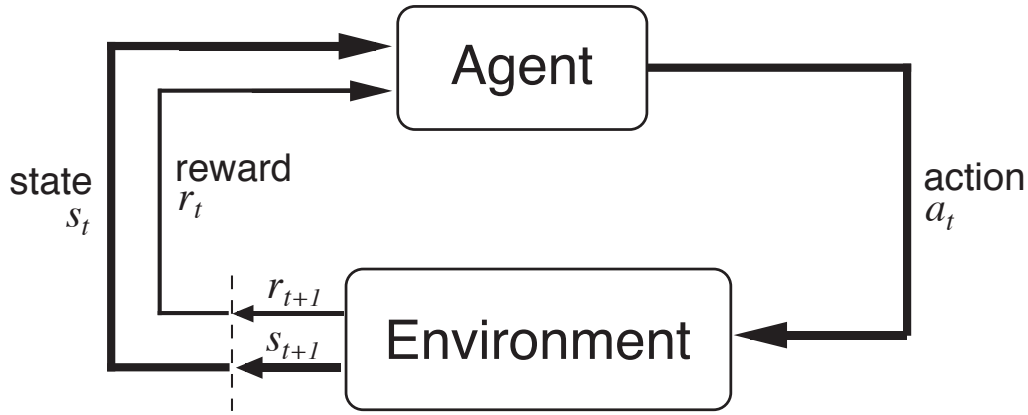
shan.luo@liverpool.ac.uk

# Recap

- 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

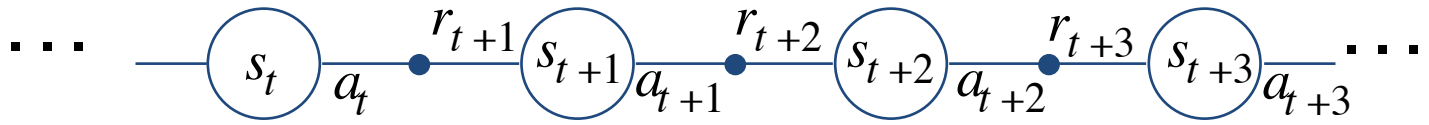
# Recap

- Reinforcement learning
  - Agent-Environment Interface



# Recap

- 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



# Recap

- 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

# Recap

- 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!

# Recap

- Reinforcement learning
  - Finding the optimal value function

Model of the environment?

Bootstrap?

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

# Recap

- 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

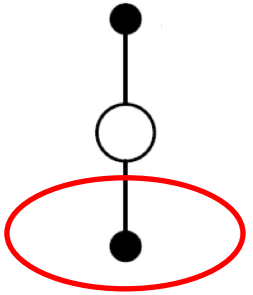
      $\mathrm{Q}(s, a) \leftarrow Q(s, a) + \alpha\big(r + \gamma Q(s', a') - Q(s, a)\big)$

# Recap

- Reinforcement learning
  - Temporal Difference learning method
  - Main algorithms:
    - SARSA (on policy)
    - Q-learning (off policy)
    - R-learning
  - Action selection:
    - ($\varepsilon$-)Greedy
    - Softmax / Boltzmann

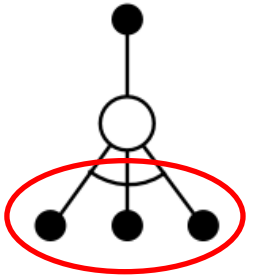
# Recap

- 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:**

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

# Recap

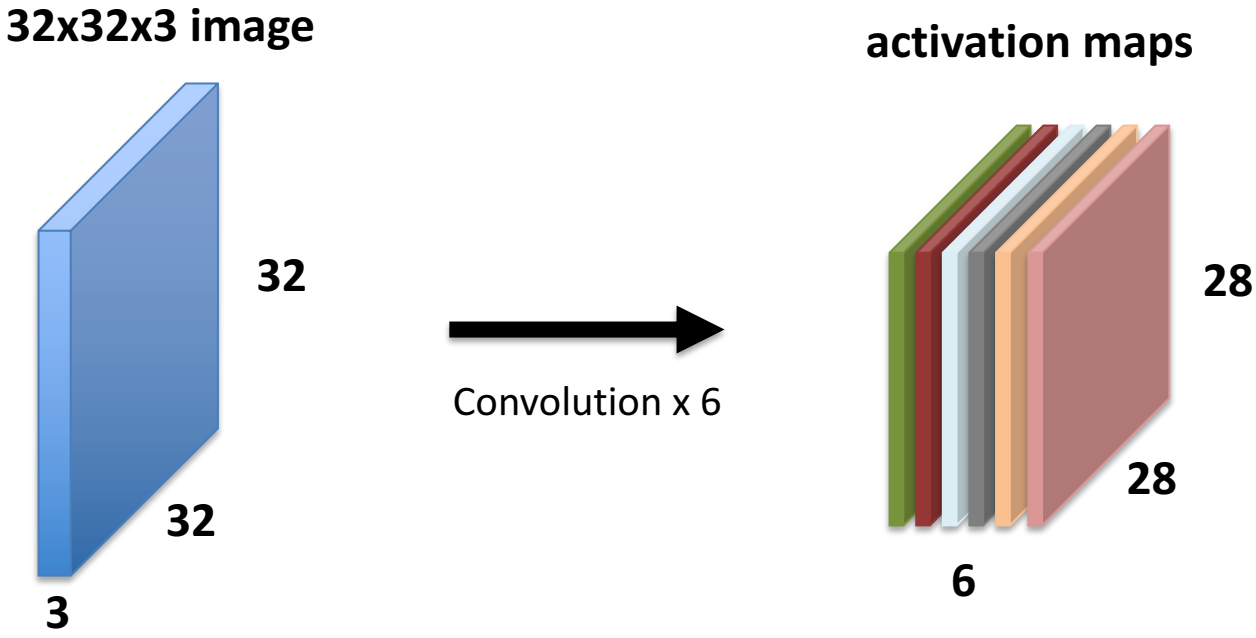
- Artificial Neural Networks
  - Perceptrons (how it works and its limitations)
  - Threshold Logic Units (and how to use them)
  - Linear separability
  - Multi-class classification (a layer of perceptrons)
  - Multi-layer networks
  - Backpropagation (the idea, don't worry about the maths)

# Recap

- 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

# Recap

- Convolutional Neural Networks
  - Convolution layer



# Recap

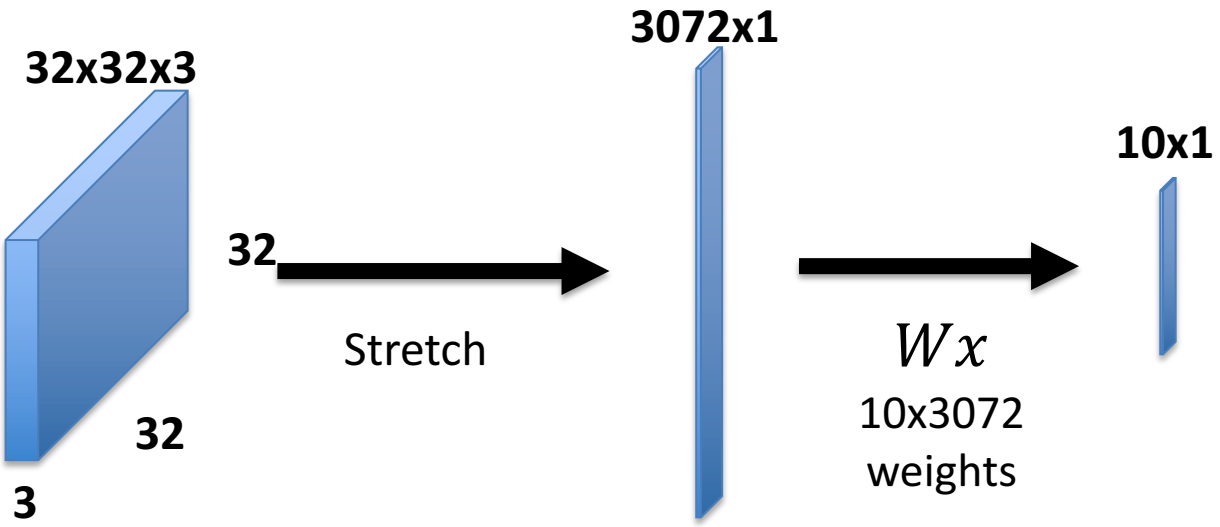
- Convolutional Neural Networks
  - Convolution layer
    - Filter! Reuse of weights.
    - Hyperparameters
      - Number of filters **K**
      - Spatial extent **F**
      - Stride **S**
      - Amount of zero padding **P**
    - Know how to compute output volume dimensions

# Recap

- 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

# Recap

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



# ANY QUESTIONS?