Lecture 8:
GAN and Reinforcement Learning

Yanbin Liu

Auckland University of Technology

May 7, 2024

Table of Contents

GAN:Generative Adversarial Network

GAN: A Two-Player Game

- GANs estimate generative models via an adversarial process.
- GANs simultaneously train two models: A generative model G and a discriminative model D.
- Model G captures the data distribution.
- Model D estimates the probability that a sample came from the training data.
- The training procedure for G is to maximize the probability of D.
- This framework corresponds to a *minimax* two-player game.

Goodfellow, I. et al. (2014) Generative adversarial nets, In NIPS, pp. 2672-2680.

GAN: An Example

- In GANs, the generative model can be thought as analogous to a team of counterfeiters who produce fake currency and use it without detection.
- In GANs, the discriminative model is analogous to the police who detect the counterfeit currency.

Goodfellow, I. et al. (2014) Generative adversarial nets, In NIPS, pp. 2672–2680.

GAN: An Example

- In GANs, competition in this game drives both teams to improve their methods till the counterfeits are indistinguishable.
- In a simple GAN, for example, generative model generates samples through a multilayer perceptron, the discriminative model is also a multilayer perceptron.

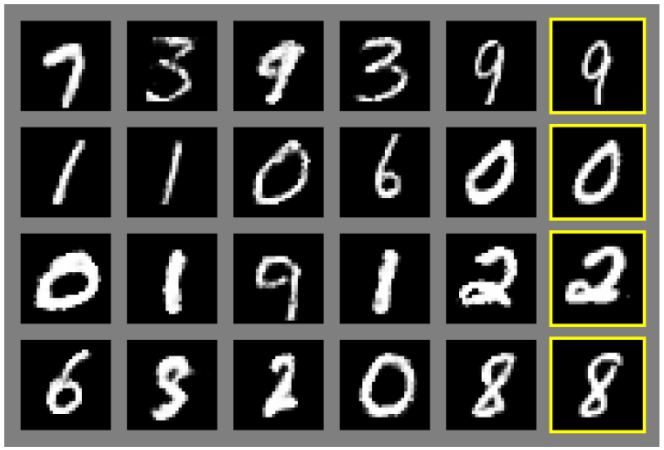
Goodfellow, I. et al. (2014) Generative adversarial nets, In NIPS, pp. 2672–2680.

GAN: A Simple Model

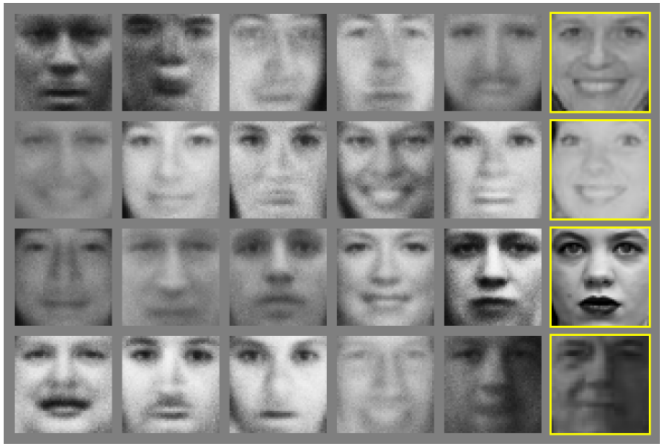
- We generate samples from the generative model by using only forward propagation.
- We train both models by using only the highly successful backpropagation and dropout algorithms.
- A large network randomly drops out nodes during training for reducing overfitting and improving generalization errors.

Goodfellow, I. et al. (2014) Generative adversarial nets, In NIPS, pp. 2672–2680.

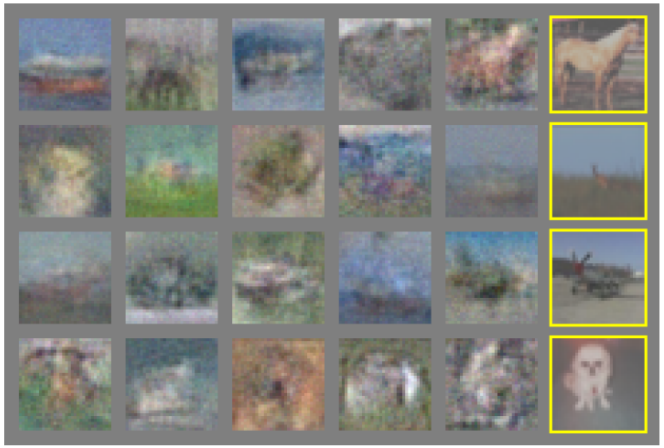
GAN Using Datasets: MNIST, TFD, CIFAR-10



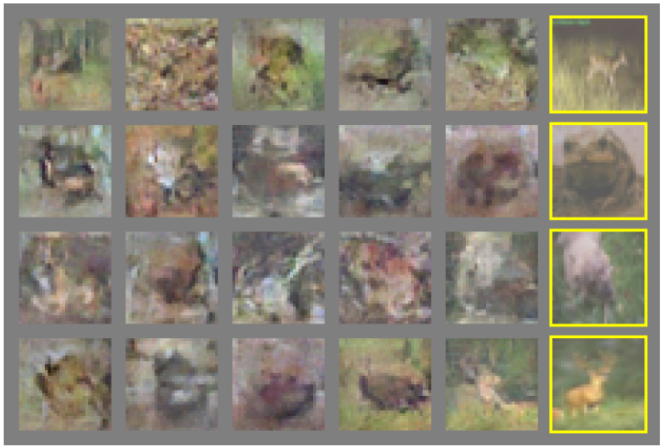
a)



b)



c)



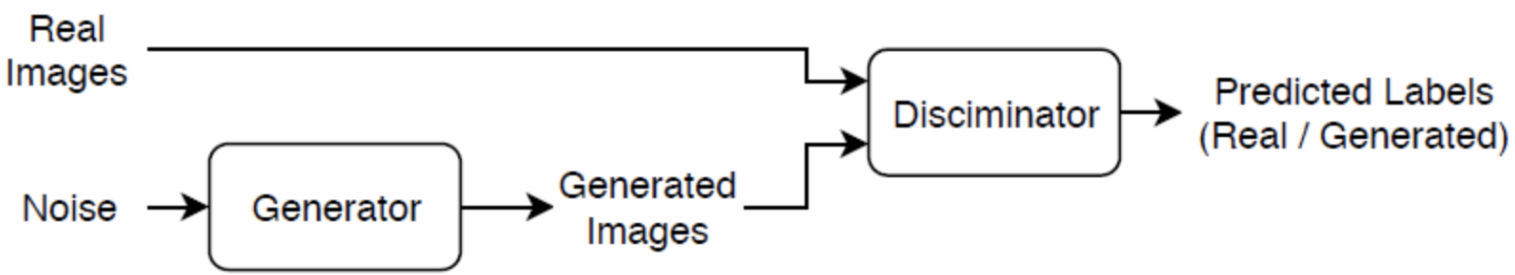
d)

Goodfellow, I. et al. (2014) Generative adversarial nets, In NIPS, pp. 2672–2680.

GAN:Generative Adversarial Network

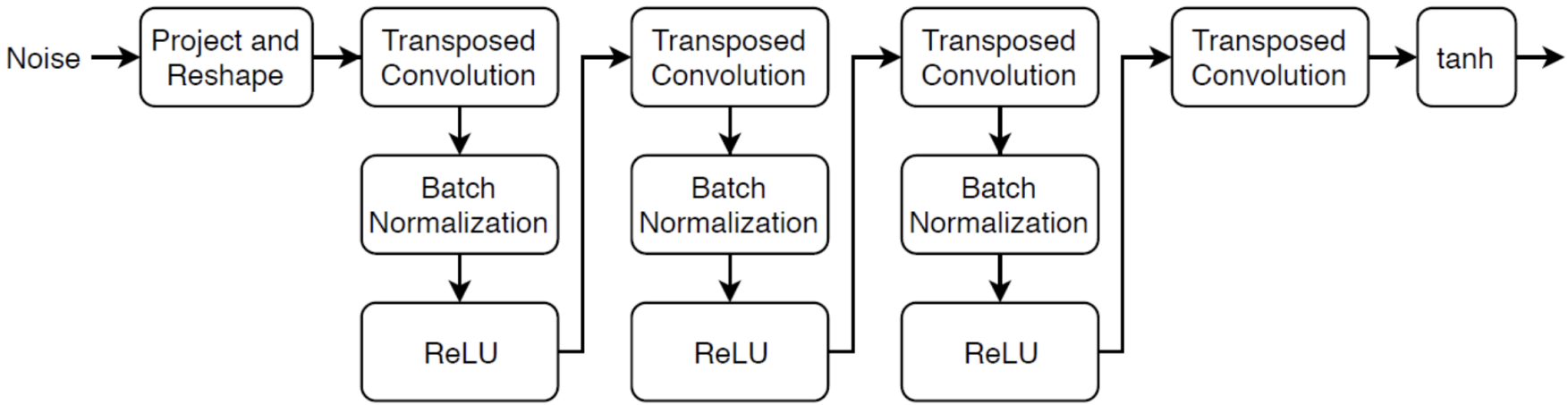
MATLAB GAN: Architecture

- MATLAB GAN is a deep learning network that can generate data with similar characteristics as the input real data.
- **Generator**: Given a random vector as input, this network generates data with the same structure as the training data.
- **Discriminator**: Given batches of data, containing observations from both the training data, and generated data from the generator, this network classifies the observations as “real” or “generated”.



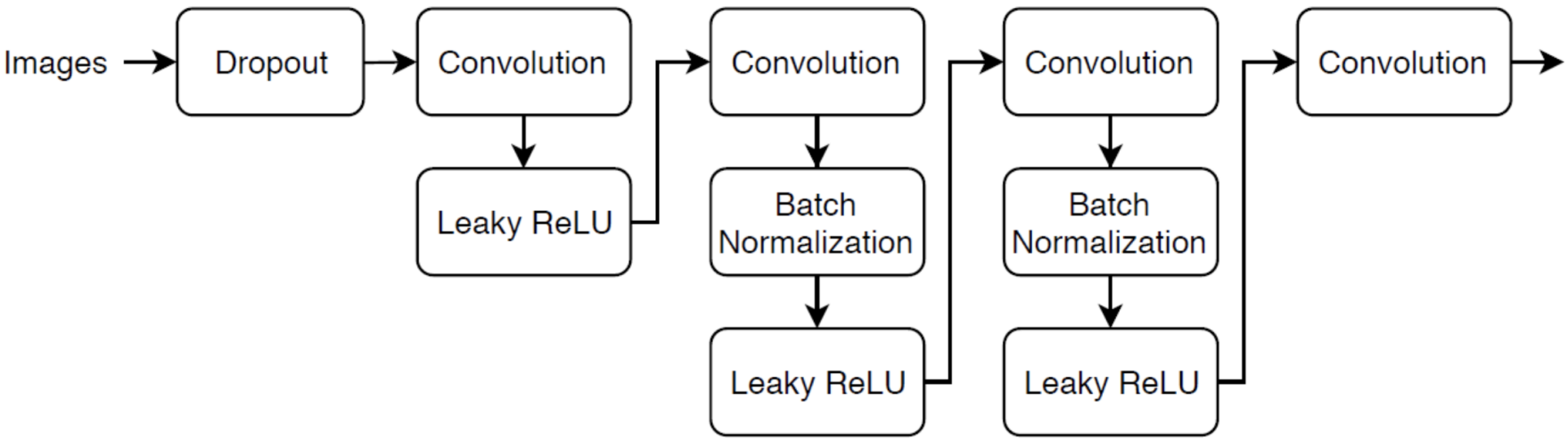
web: https://www.mathworks.com/help/deeplearning/examples/train-generative-adversarial-network.html

MATLAB GAN: Architecture of Generator



web: https://www.mathworks.com/help/deeplearning/examples/train-generative-adversarial-network.html

MATLAB GAN: Architecture of Discriminator



web: https://www.mathworks.com/help/deeplearning/examples/train-generative-adversarial-network.html

MATLAB GAN Loss Functions

- The loss function for the generator is given by

$$L_G = \text{mean}(\log(\sigma(\hat{Y}_G)))$$

where $\sigma(\cdot)$ denotes the sigmoid function, $\hat{Y}_G$ stands for the output of the discriminator with generated data input, mean is the mean function.

- The loss function for the discriminator is given by

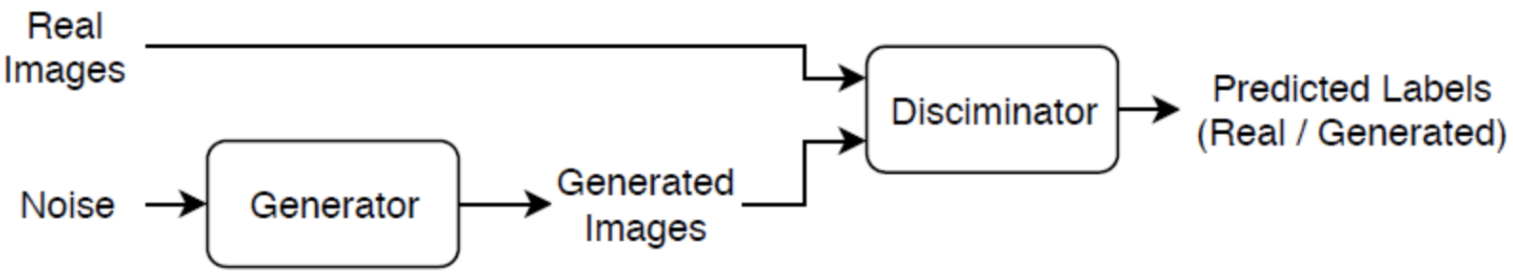
$$L_D = -\text{mean}(\log(\sigma(\hat{Y}_D))) - \text{mean}(\log(1 - \sigma(\hat{Y}_G)))$$

where $\hat{Y}_D$ denotes the output of the discriminator with real data input, mean is the mean function.

web: https://www.mathworks.com/help/deeplearning/examples/train-generative-adversarial-network.html

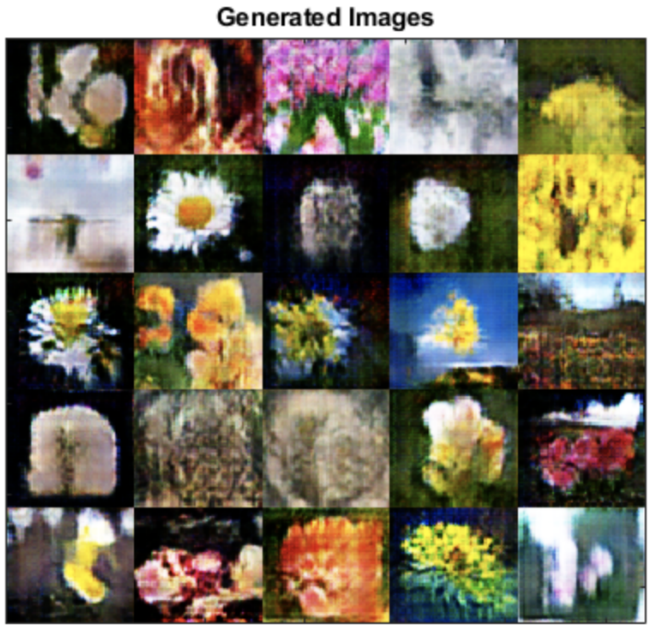
MATLAB GAN: Training

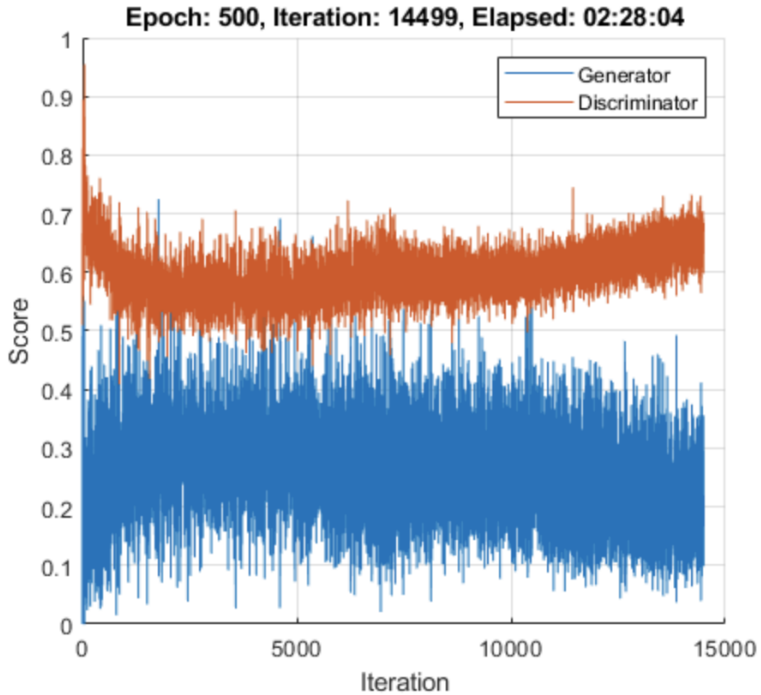
- Train the generator to generate data that "fools" the discriminator.
- Train the discriminator to distinguish between real and generated data.
- Optimize the performance of the generator, maximize the loss of the discriminator when given generated data.
- Optimize the performance of the discriminator, minimize the loss of the discriminator when given batches of both real and generated data.

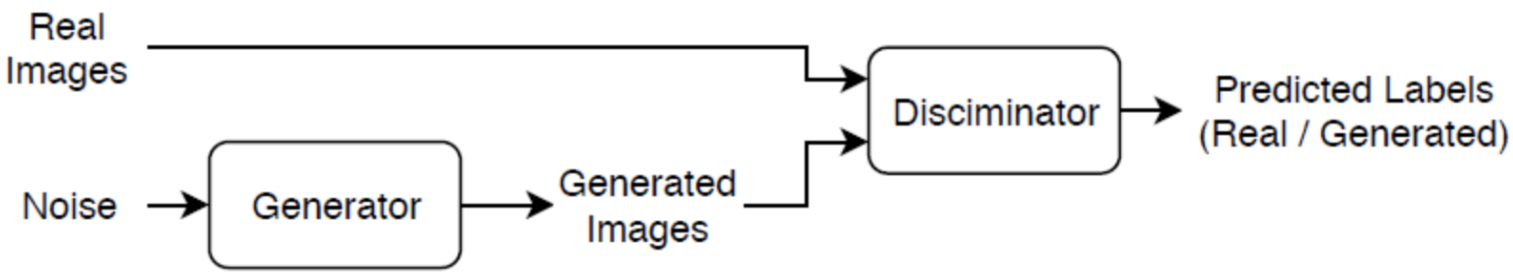


web: https://www.mathworks.com/help/deeplearning/examples/train-generative-adversarial-network.html

MATLAB GAN: Results

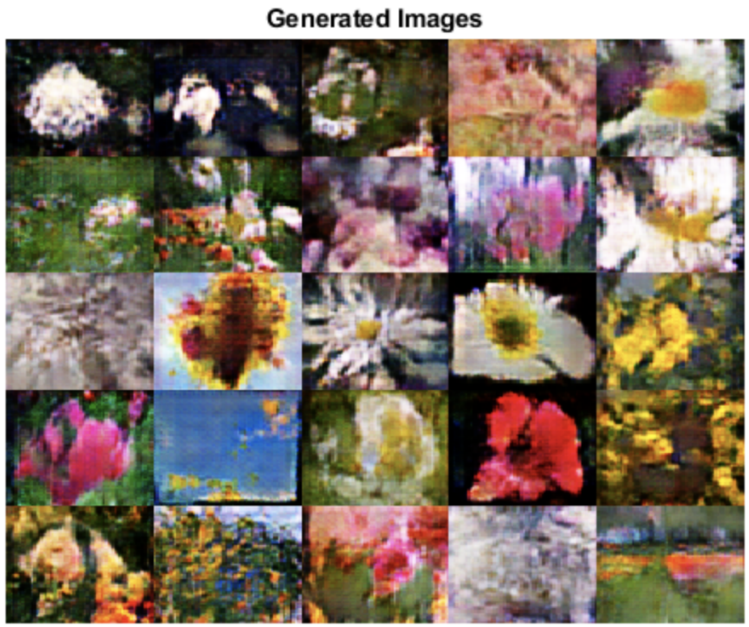


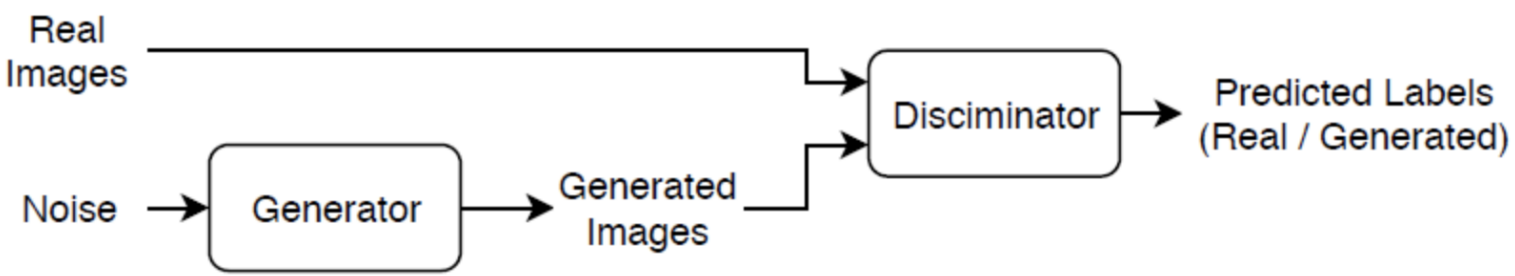




web: https://www.mathworks.com/help/deeplearning/examples/train-generative-adversarial-network.html

MATLAB GAN: Results





web: https://www.mathworks.com/help/deeplearning/examples/train-generative-adversarial-network.html

Questions?

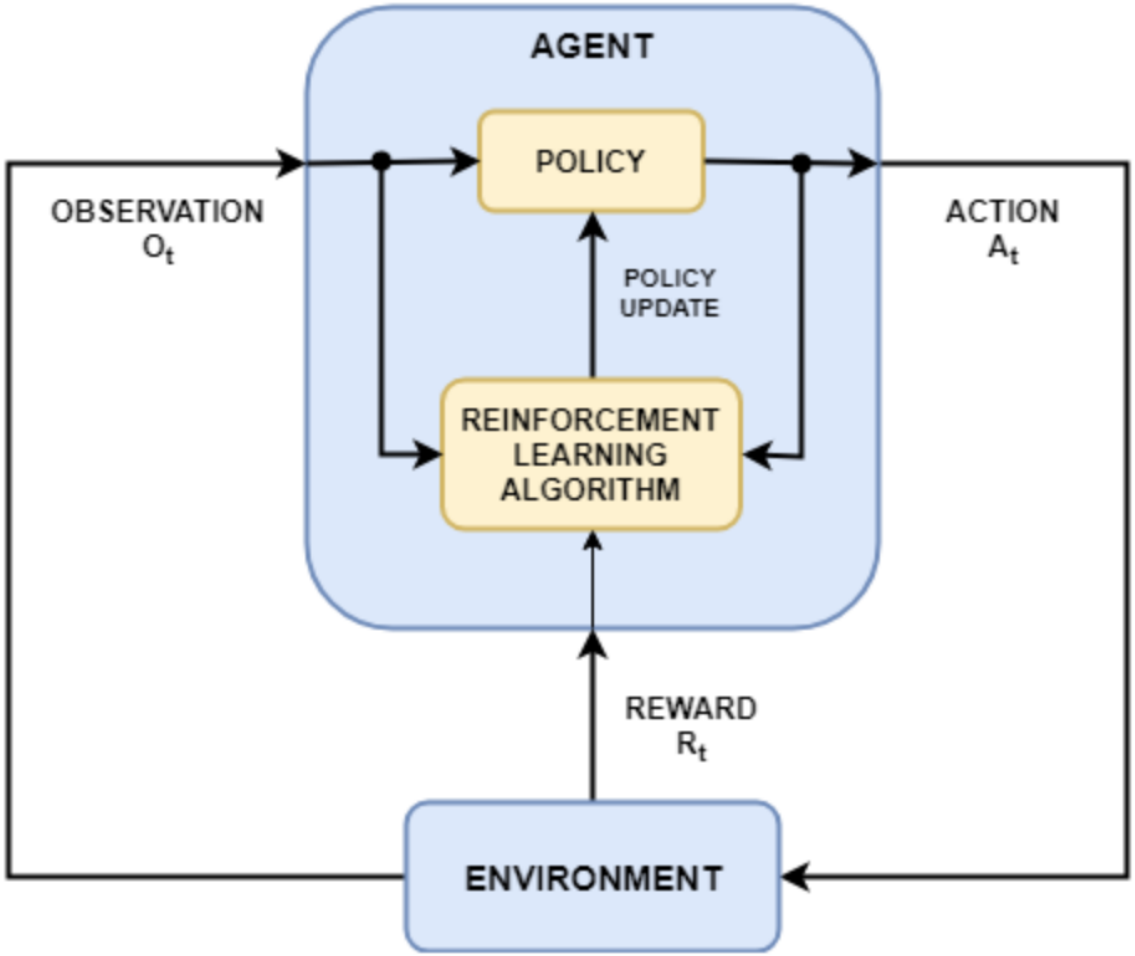
Reinforcement Learning

MATLAB Reinforcement Learning

- Reinforcement learning is a goal-directed computational approach where a computer learns to perform a task by interacting with an unknown dynamic environment.
- This learning approach enables a computer to make a series of decisions to maximize the cumulative reward for the task without human intervention and without being explicitly programmed to achieve the task.

web:https://www.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html

MATLAB Reinforcement Learning



web:https://www.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html

MATLAB Reinforcement Learning

- The goal of reinforcement learning is to train an agent to complete a task within an unknown environment.
- The agent receives observations and a reward from the environment and sends actions to the environment.
- The reward is a measure of how successful an action is with respect to completing the task goal.
- The agent contains two components: A policy and a learning algorithm.
- The policy is a mapping that selects actions based on the observations from the environment.
- .....

web:https://www.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html

MATLAB Reinforcement Learning

- .....
- Typically, the policy is a function approximator with tunable parameters, such as a deep neural network.
- The learning algorithm continuously updates the policy parameters based on the action, observations, and reward.
- The goal of the learning algorithm is to find an optimal policy that maximizes the cumulative reward received during the task.
- Reinforcement learning refers to an agent learning the optimal behavior through repeated trial-and-error interactions with the environment without human involvement.

web:https://www.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html

MATLAB Reinforcement Learning Workflow

The general workflow for training an agent using reinforcement learning includes the following steps:

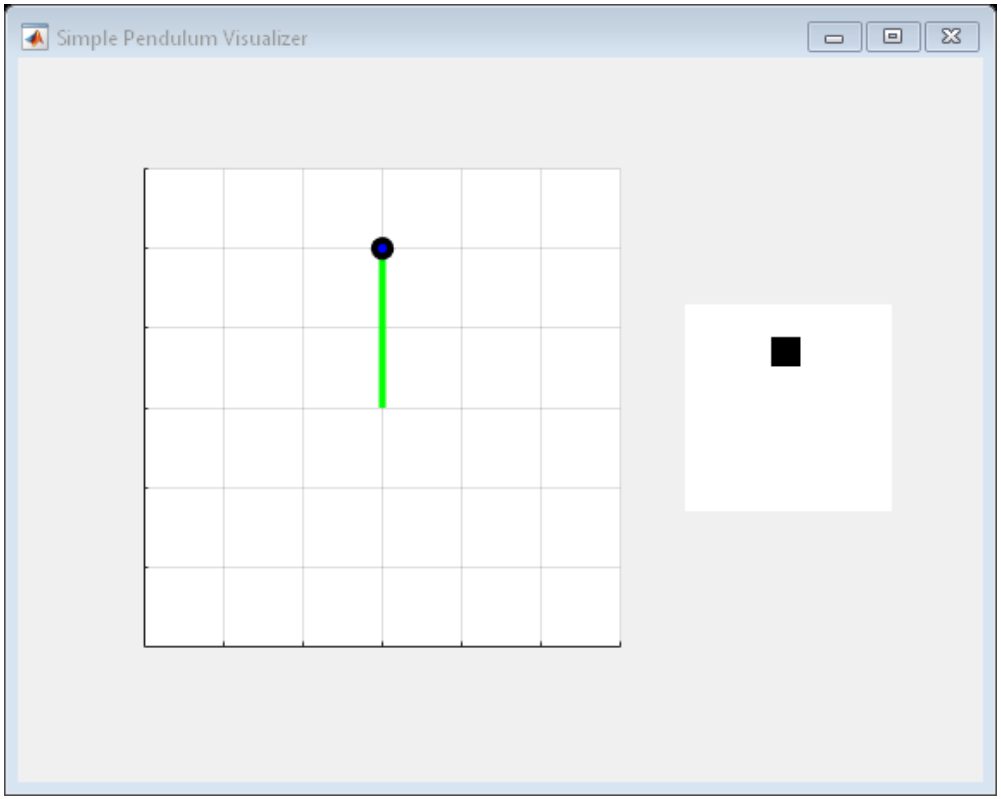
- Formulate Problem: Define the task for the agent to learn.
- Create Environment: Define the environment within which the agent operates.
- Define Reward: Specify the reward signal that the agent uses to measure its performance.
- Create Agent: Create the agent.
- Train Agent: Train the agent policy representation.
- Validate Agent: Evaluate the performance of the trained agent.
- Deploy Policy: Deploy the trained policy representation.

web:https://www.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html

MATLAB Reinforcement Learning Workflow

web:https://www.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html

MATLAB Reinforcement Learning: Balance Pendulum



web:https://au.mathworks.com/help/deeplearning/ug/train-ddpg-agent-to-swing-up-and-balance-pendulum-with-image-observation.html

MATLAB Reinforcement Learning: Pendulum

https://au.mathworks.com/matlabcentral/fileexchange/57882-reinforcement-learning-example-pendulum-controller-w
s_tid=FX_rc1_behav

MATLAB Reinforcement Learning: Basic Grid World

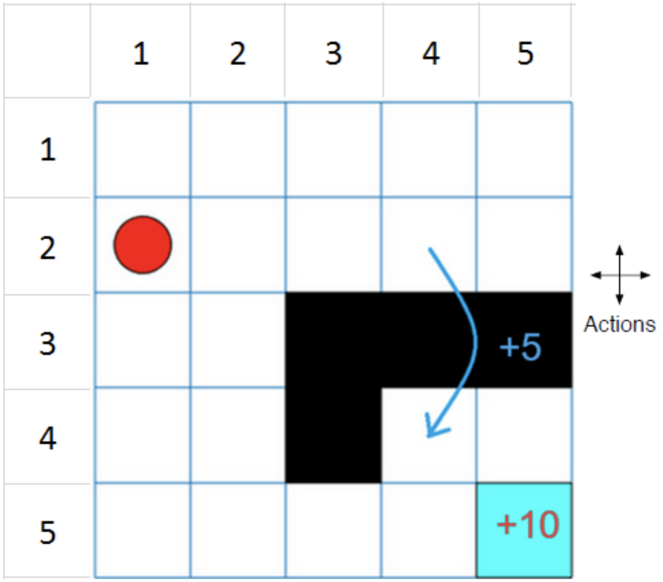
The agent begins from cell [2,1] (second row, first column).

The agent receives a reward +10 if it reaches the terminal state at cell [5,5] (blue).

The environment contains a special jump from cell [2,4] to cell [4,4] with a reward of +5.

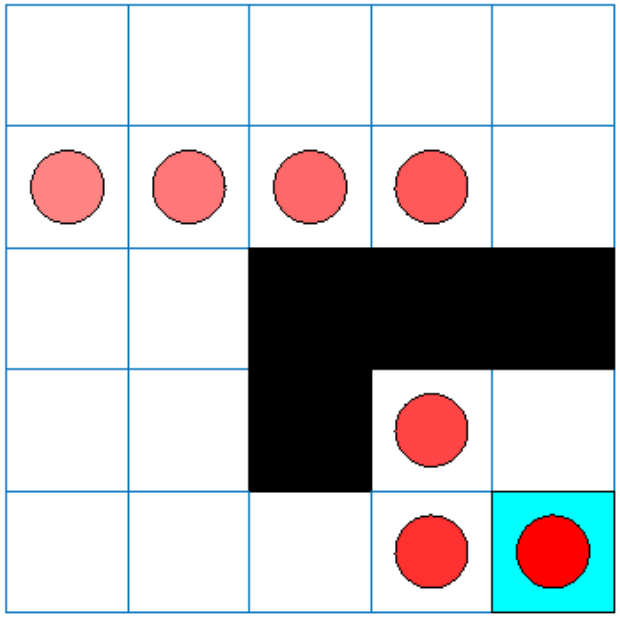
The agent is blocked by obstacles (black cells).

All other actions result in −1 reward.



web:https://www.mathworks.com/help/reinforcement-learning/ug/train-q-learning-agent-to-solve-basic-grid-world.html

MATLAB Reinforcement Learning: Basic Grid World



web:https://www.mathworks.com/help/reinforcement-learning/ug/train-q-learning-agent-to-solve-basic-grid-world.html

Questions?

GAN and Reinforcement Learning

Learning Objectives

- Design and analyse algorithms of deep neural networks.
- Demonstrate advanced understanding of the state-of-the-art in the practice of deep learning.