Center for Visual Information Technology
IIIT Hyderabad

Reinforcement Learning - Introduction

Mohit Rathore
mrmohitrathoremr@gmail.com

Sep 7, 2018

Contents

Trivia

- Motivation
- News

Deep Q Learning

- Understanding the terms.
- The approach.
- Flaws with DQN.

Practical examples

- Gym and fromscratchtoml

Trivia

Why should I care?

- At the very core deep learning can solve problem of classification, regression i.e. approximating complex functions.
- Reinforcement learning on the other hand help's us devise an optimum strategy to follow in our feat of maximizing reward.

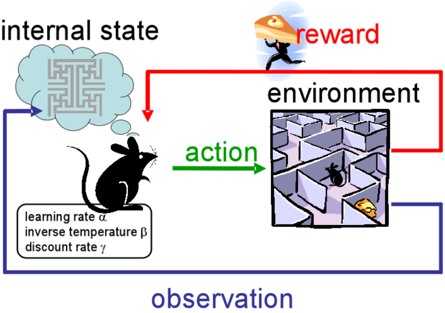
Okay gang, let's see who the advanced AI really is.
IF IF IF IF IF IF IF IF IF IF IF IF IF IF IF IF IF IF IF IF IF IF IF IF

Trivia

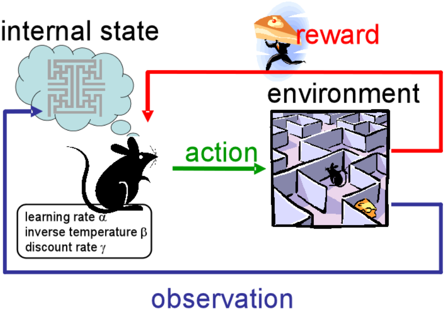
News

- RL Beats Humans at Dota 2. ▸ Link
- AlphaGo beats 18-time world champion Lee Sedol. ▸ Link
- Google has developed an approach to hyperparameter tuning using reinforcement learning that they call AutoML. ▸ Link

- **State** - It can be seen as a variable which directly affect the action taking mechanism. In an environment such as a maze, a state can be seen as the x, y coordinates of the agent.



- **Action** - Performing a particular action in a particular state takes you to the next state.
- The mouse can move in either of the four directions to reach the next block.

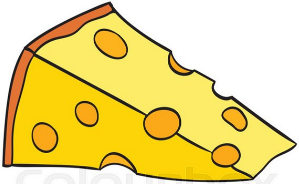


Introduction

Jargons

- **Agent** - It is an abstraction of the entity which acts and learns from previous experiences in hope to maximize reward.
- In this case our agent is - mouse.

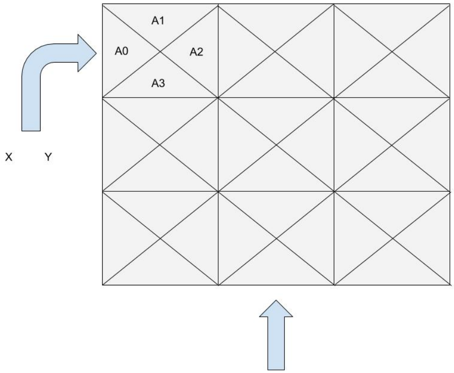
- **Reward** - It is the utility that our agent is trying to maximize.
- For our mouse reward is - cheese.



Introduction

Jargons

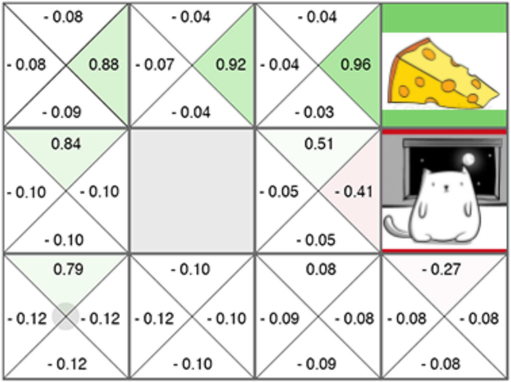
- **Environment** - Environment can be considered as a entity which takes in space-action and returns the next state and reward for that action.



Environment

Reinforcement Learning

Deep Q learning



What can be the best case scenario?

What can be the best case scenario?

Deep Q-Network Training

- Bellman Equation:

$$Q(s,a) = r + \gamma max_{a'} Q(s',a')$$

- Loss function (squared error):

$$L = \mathbb{E}[(\underbrace{\boldsymbol{r + \gamma max_{a'} Q(s',a')}}_{\textbf{target}} - Q(s,a))^2]$$

Why discount the reward by gamma?

Why discount the reward by gamma?

- So that rewards do not diverge as the time steps increase.

Why discount the reward by gamma?

- So that rewards do not diverge as the time steps increase.
- We will be running in circles.

Exploration!

- We take the optimum action but sometimes we take a random action with exploration probability ϵ so that surroundings are investigated for possible rewards.
- We decay this ϵ over time exponentially.

$$\epsilon = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min}) * \exp^{(-\lambda * t)}$$

Reinforcement Learning

Deep Q learning

- **Explore** - Take random actions and create a queue of (current state, action, next state, reward).

- **Explore** - Take random actions and create a queue of (current state, action, next state, reward).
- **Train** - Sample from this queue and use the data to create a training data for our model to learn from.

- **Explore** - Take random actions and create a queue of (current state, action, next state, reward).
- **Train** - Sample from this queue and use the data to create a training data for our model to learn from.
  - This should be done keeping in mind that we tend to converge the bellman equation.

Reinforcement Learning

Deep Q learning

- **Explore** - Take random actions and create a queue of (current state, action, next state, reward).
- **Train** - Sample from this queue and use the data to create a training data for our model to learn from.
    - This should be done keeping in mind that we tend to converge the bellman equation.
    - Taking a random action with a exploration probability (which decrease over time).

- **Explore** - Take random actions and create a queue of (current state, action, next state, reward).
- **Train** - Sample from this queue and use the data to create a training data for our model to learn from.
  - This should be done keeping in mind that we tend to converge the bellman equation.
  - Taking a random action with a exploration probability (which decrease over time).
  - The new variables (state, action, next state, reward) acquired are pushed into the queue.

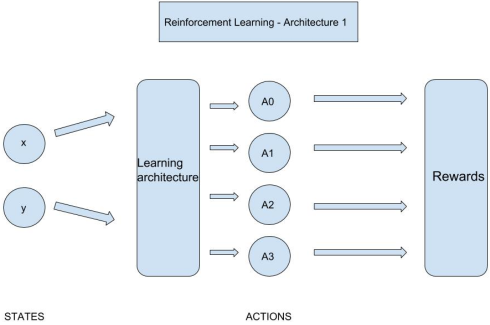
Reinforcement Learning

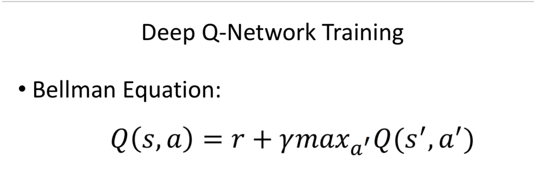
Deep Q learning

- **Explore** - Take random actions and create a queue of (current state, action, next state, reward).
- **Train** - Sample from this queue and use the data to create a training data for our model to learn from.
  - This should be done keeping in mind that we tend to converge the bellman equation.
  - Taking a random action with a exploration probability (which decrease over time).
  - The new variables (state, action, next state, reward) acquired are pushed into the queue.
- **Play** - The model is tested by keeping the exploration probability as zero and stopping learning from experiences.

Reinforcement Learning

Deep Q learning





Deep Q-Network Training

- Bellman Equation:

$$Q(s,a) = r + \gamma max_{a'} Q(s',a')$$

There is a flaw in this approach.

There is a flaw in this approach.

- Our Q values shift but the target value also shifts.

Reinforcement Learning

Deep Q learning

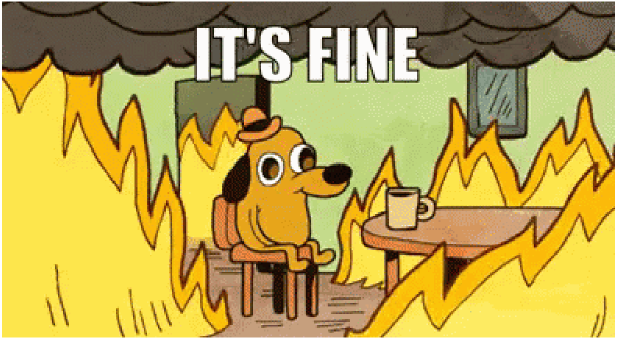
There is a flaw in this approach.

- Our Q values shift but the target value also shifts.
- Due to the max in the formula for setting targets, the network suffers from maximization bias, possibly leading to overestimation of the Q function's value and poor performance.

$$\underbrace{Q(s,a)}_{\text{TD target}} = r(s,a) + \underbrace{\gamma Q(s', \underbrace{argmax_a Q(s',a)}_{\text{DQN Network choose action for next state}})}_{\text{Target network calculates the Q value of taking that action at that state}}$$

Like most all machine learning algorithms all this comes at a price.

Like most all machine learning algorithms all this comes at a price.



It uses a lot of system resources!

Reinforcement Learning

Breakout

Breakout by deepmind ▸ Link

Examples

openAI gym and fromscratchtoml

openAI gym provides an environment for your agent to learn and play around. ▸ Link

Code for machine learning algorithms from scratch - ▸ Link

Thank You