

# REINFORCEMENT LEARNING

Imitando l'apprendimento umano



# COME SI IMPARA?









#### **ALGORITMO PER IL COMPUTER**



```
76 WebSpider.py - C:\Python32\WebSpider.py
File Edit Format Run Options Windows Help
from html.parser import HIMLParser
from urllib.request import urlopen
from urllib import parse
class LinkParser(HTMLParser):
   def handle_starttag(self, tag, attrs):
       if tag -- 'a':
           for (key, value) in attrs:
               if key == 'href':
                    newUrl = parse.urljoin(self.baseUrl, value)
                    self.links = self.links + [newUrl]
   def getLinks(self, url):
        self.links = []
        self.baseUrl = url
        response = urlopen(url)
        if response.getheader('Content-Type') == 'text/html':
           htmlBytes = response.read()
            htmlString = htmlBvtes.decode("utf-8")
           self.feed(htmlString)
           return htmlString, self.links
           return "",[]
def spider(url, word, maxPages):
   pagesToVisit = [url]
   numberVisited = 0
   foundWord = False
   while numberVisited < maxPages and pagesToVisit != [] and not foundWord:
       numberVisited = numberVisited +1
        url = pagesToVisit[0]
        pagesToVisit = pagesToVisit[1:]
           print (number Visited, "Visiting:", url)
           parser = LinkParser()
            data, links = parser.getLinks(url)
           if data.find(word)>-1:
               foundWord = True
           pagesToVisit = pagesToVisit + links
           print(" **Failed!**")
   if foundWord:
      print ("The word", word, "was found at", url)
       print ("Word never found")
                                                                      Ln: 47 Col: 33
```



( ) Fr (r) dx ) dt - x ( 45 (x) dt - x 8 (x) + ) (x-a) & R\* (a) du A(x) = \$ b. 7 (lev) 1xU41=1x1+141-1x041 g: X -> X nW (-x/1-9 1) - (9(1-9) -NA) x3+x2+ x3+23+xy2-6-0 163+1+n

### **COME SI IMPARA?**

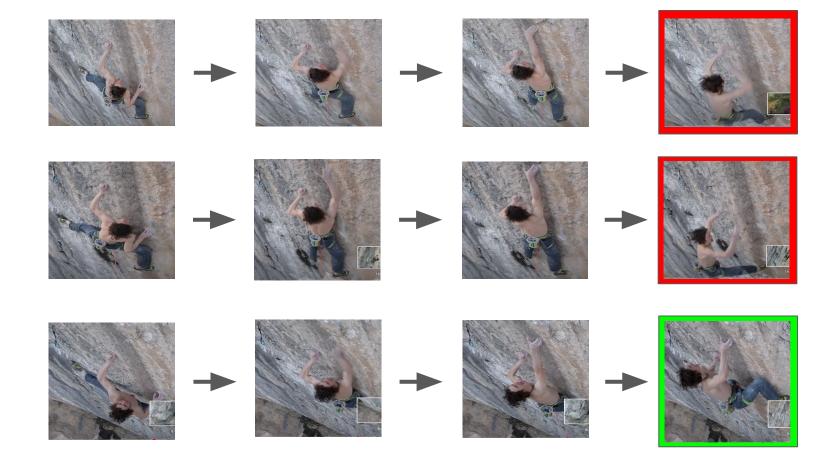






## PROVA ED ERRORE





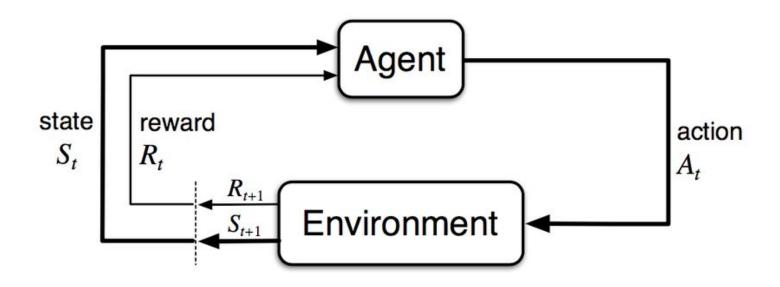
#### PROVA ED ERRORE





#### L'AMBIENTE DEL RL

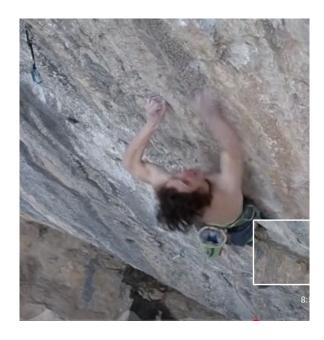


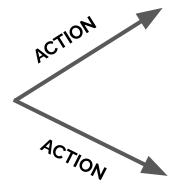


#### LA DINAMICA DEL RL



STATE t





**STATE** *t*+1



REWARD = -1

**STATE** *t*+1



REWARD = +1



$$q_*(s, a) = E[R_a + \gamma \max_{a'} q_*(s', a')]$$

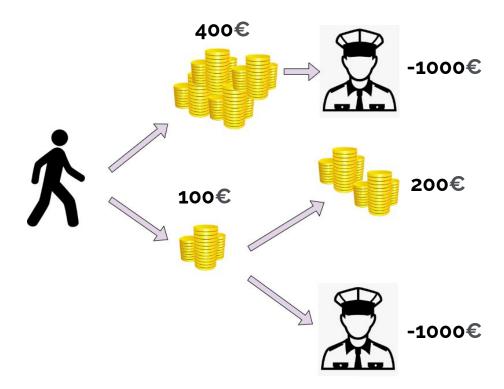
**BELLMAN (1957)** 



$$q_*(s, a) = R_a + \max_{a'} q_*(s', a')$$

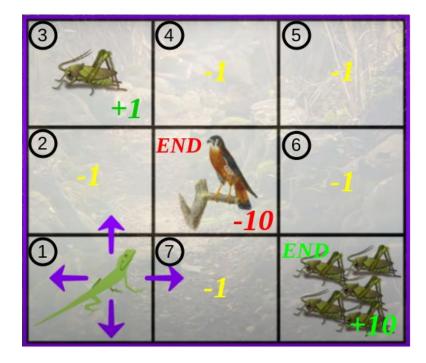


$$q_*(s, a) = R_a + \max_{a'} q_*(s', a')$$



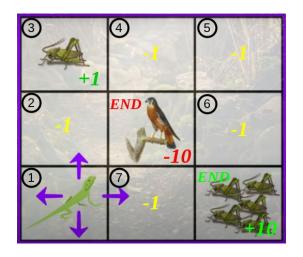


$$q_*(s, a) = R_a + \max_{a'} q_*(s', a')$$



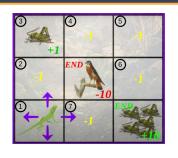


$$q_*(s, a) = R_a + \max_{a'} q_*(s', a')$$



	<b>^</b>	<b>\</b>	1	<b>+</b>
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
<b>⑤</b>	0	0	0	0
6	0	0	0	0
7	0	0	0	0



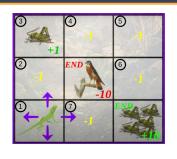


$$q_*(s, a) = R_a + \max_{a'} q_*(s', a')$$

	<b>^</b>	<b>\</b>	1	1
1	0	0	0	0
2	0	0	0	0
<b>)</b> ③	0	0	0	0
4	0	0	0	0
5	0	0	0	0
<b>)</b> ©	0	0	0	0
7	0	0	0	0

	<b>^</b>	<b>\</b>	<b></b>	1
1	-1	0	-1	0
2	+1	0	-10	0
3	0	-1	-1	0
4	0	-10	-1	+1
5	0	-1	0	-1
6	-1	+10	0	-10
7	-10	0	+10	0





$$q_*(s, a) = R_a + \max_{a'} q_*(s', a')$$

	<b>^</b>	<b>\</b>	1	1
<b>(</b>	-1	0	-1	0
2	+1	0	-10	0
<u>©</u>	0	-1	-1	0
4	0	-10	-1	+1
<u>©</u>	0	-1	0	-1
6	-1	+10	0	-10
7	-10	0	+10	0

	<b>^</b>	<b>\</b>	<b>†</b>	1
<b>(</b>	0	0	+9	0
2	+1	0	-10	0
3	0	0	0	0
4	0	-10	-1	+1
5	0	+9	0	0
6	-1	+10	0	-10
7	-10	0	+10	0

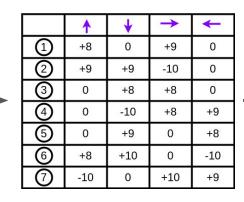


	<b>^</b>	<b>\</b>	<b>†</b>	1
1	0	0	+9	0
2	+1	0	-10	0
3	0	0	0	0
4	0	-10	-1	+1
<u>(5)</u>	0	+9	0	0
6	-1	+10	0	-10
7	-10	0	+10	0

	<b>^</b>	<b>\</b>	<b>†</b>	1
1	0	0	+9	0
2	+1	+9	-10	0
3	0	0	0	0
4	0	-10	+8	+1
<u>(5)</u>	0	+9	0	0
6	+8	+10	0	-10
7	-10	0	+10	+9
	·			

	77	1927		8.0
	<b>^</b>	₩	1	1
1	+8	0	+9	0
2	+1	+9	-10	0
3	0	+8	+7	0
4	0	-10	+8	+1
5	0	+9	0	+7
6	+8	+10	0	-10
7	-10	0	+10	+9

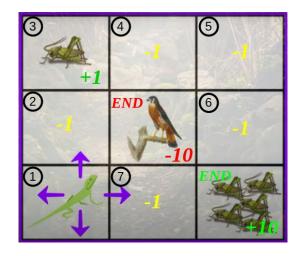
		<b>^</b>	<b>\</b>	1	<b>\</b>
	<b>(</b>	+8	0	+9	0
	2	+9	+9	-10	0
1	3	0	+8	+7	0
	4	0	-10	+8	+9
	<u>(5)</u>	0	+9	0	+7
	6	+8	+10	0	-10
	7	-10	0	+10	+9
				•	•



	<b>^</b>	<b>→</b>	1	<b>+</b>
1	+8	0	+9	0
2	+9	+9	-10	0
3	0	+8	+8	0
4	0	-10	+8	+9
(5)	0	+9	0	+8
6	+8	+10	0	-10
7	-10	0	+10	+9



$$q_*(s, a) = R_a + \max_{a'} q_*(s', a')$$



...si può fare anche per tentativi...

	<b>^</b>	<b>\</b>	<b></b>	+
1	+8	0	+9	0
2	+9	+9	-10	0
3	0	+8	+8	0
4	0	-10	+8	+9
5	0	+9	0	+8
6	+8	+10	0	-10
7	-10	0	+10	+9

#### **SEMPRE APPLICABILE?**





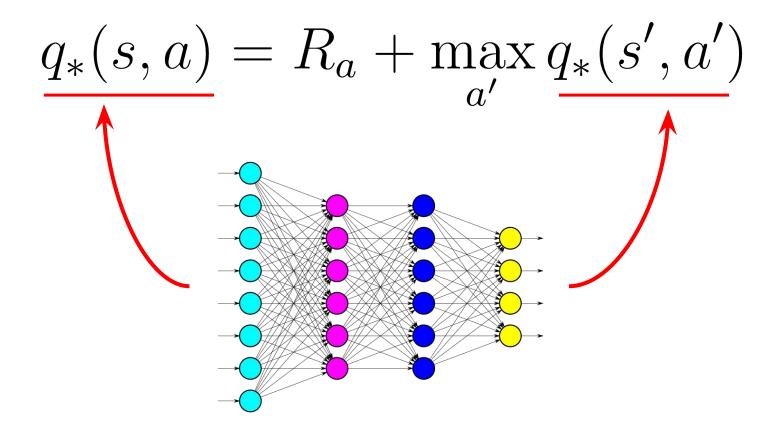




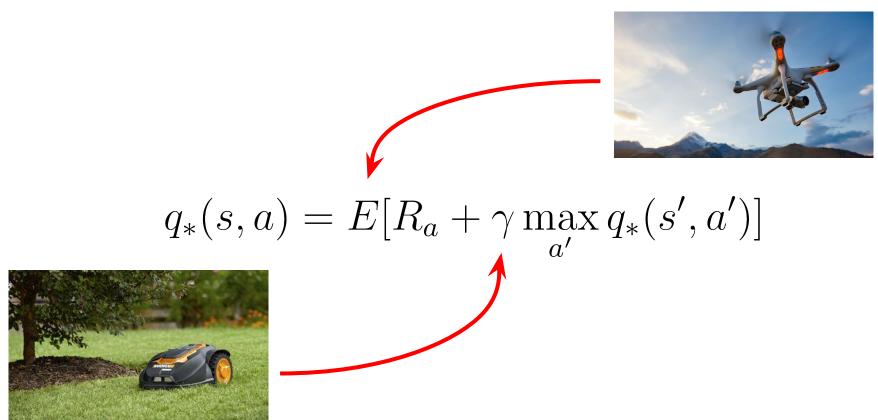
$$\longrightarrow$$
  $|S| = ??$ 

#### **MACHINE LEARNING**









# **DEEPMIND**





ALPHA GO (2016)



ALPHA STAR (2019)



ALPHA FOLD (2021)



# Grazie dell'attenzione!



davide.basso@phd.units.it

