

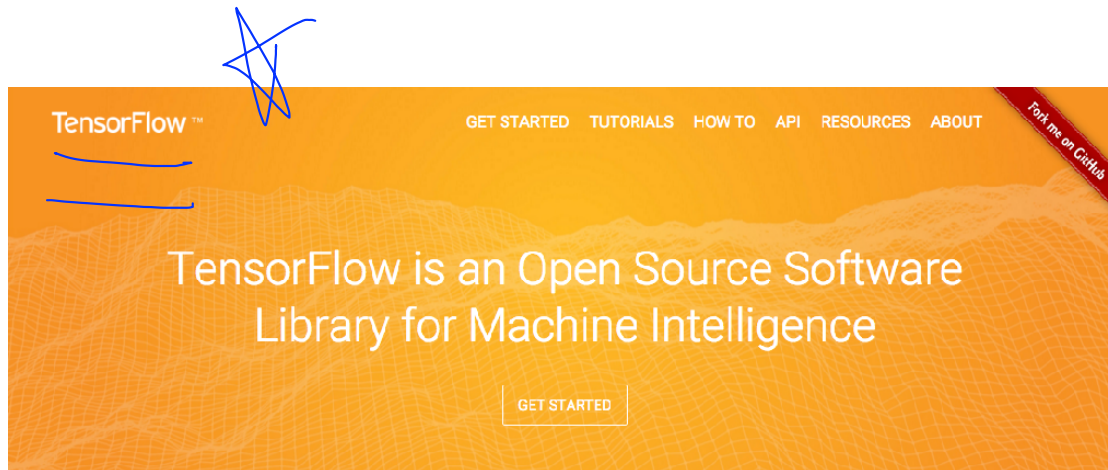


Lab 2: Playing OpenAI Gym Games

Reinforcement Learning with TensorFlow&OpenAI Gym

Sung Kim <hunkim+ml@gmail.com>

Installation



Basic installation steps

- Python
- TensorFlow
 - sudo apt-get install ~~python-pip~~ python-dev
 - pip install tensorflow (or pip install tensorflow-gpu)
- OpenAI Gym
 - sudo apt install cmake
 - apt-get install zlib1g-dev
 - sudo -H pip install gym
 - sudo -H pip install gym[atari]

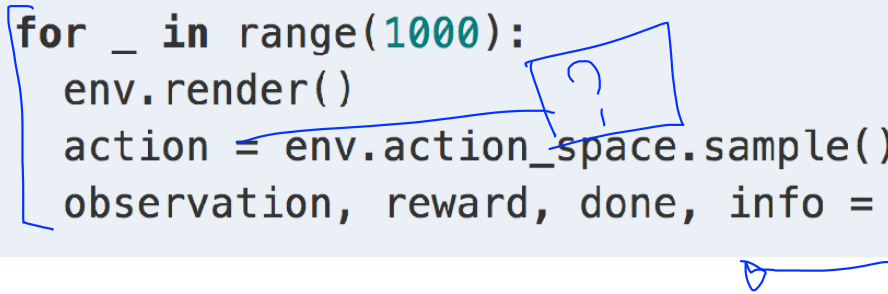
Basic installation - quick checking

```
hunkim@rl-hunkim:~$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow ✓
>>> import gym ✓
>>> █
```

Questions: <https://www.facebook.com/groups/TensorFlowKR/>

Basic OpenAI Gym Environment

```
import gym
env = gym.make("FrozenLake-v0")
observation = env.reset()
for _ in range(1000):
    env.render()
    action = env.action_space.sample() # your agent here (this takes random actions)
    observation, reward, done, info = env.step(action)
```



Python arrow keyin

```
class _Getch:
    def __call__(self):
        fd = sys.stdin.fileno()
        old_settings = termios.tcgetattr(fd)
        try:
            tty.setraw(sys.stdin.fileno())
            ch = sys.stdin.read(3)
        finally:
            termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
        return ch
```

```
inkey = _Getch()
```

```
# MACROS
```

```
LEFT = 0
```

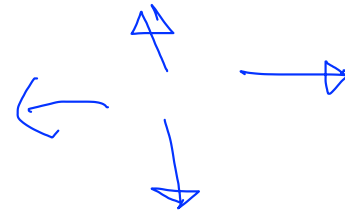
```
DOWN = 1
```

```
RIGHT = 2
```

```
UP = 3
```

```
# Key mapping
```

```
arrow_keys = {
    '\x1b[A': UP,
    '\x1b[B': DOWN,
    '\x1b[C': RIGHT,
    '\x1b[D': LEFT}
```



```
import gym
from gym.envs.registration import register
import sys, tty, termios
```

```
# Register FrozenLake with is_slippery False
register(
    id='FrozenLake-v3',
    entry_point='gym.envs.toy_text:FrozenLakeEnv',
    kwargs={'map_name': '4x4', 'is_slippery': False}
)
```

```
env = gym.make('FrozenLake-v3')
env.render() # Show the initial board
```

```
while True:
    # Choose an action from keyboard
    key = inkey()
    if key not in arrow_keys.keys():
        print("Game aborted!")
        break
    action = arrow_keys[key]
    state, reward, done, info = env.step(action)
    env.render() # Show the board after action
    print("State: ", state, "Action: ", action, "Reward: ", reward, "Info: ", info)

    if done:
        print("Finished with reward", reward)
        break
```

keyin and move

Game play

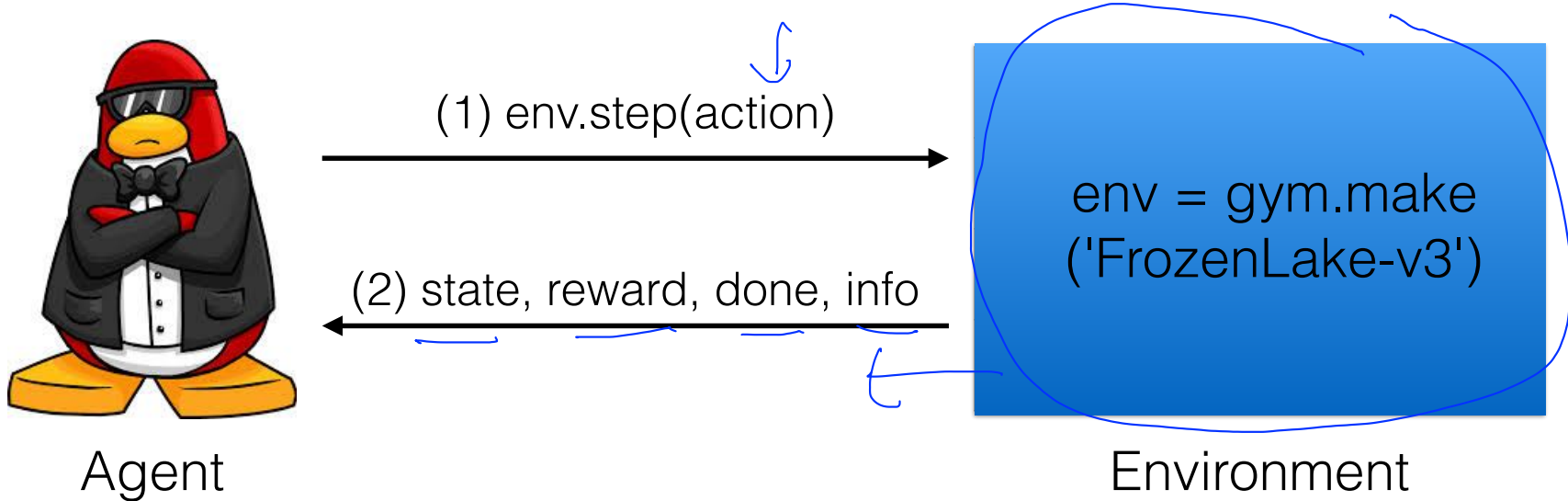
```
print("State: ", state, "Action: ", action, "Reward: ", reward, "Info: ", info)
```

```
Sungs-MacBook-Pro:qlearning hunkim$ python 01_play_frozenlake_det.py
[2016-12-29 20:45:09,107] Making new env: FrozenLake-v3
SFFF
FHFH
FFFH
HFFG

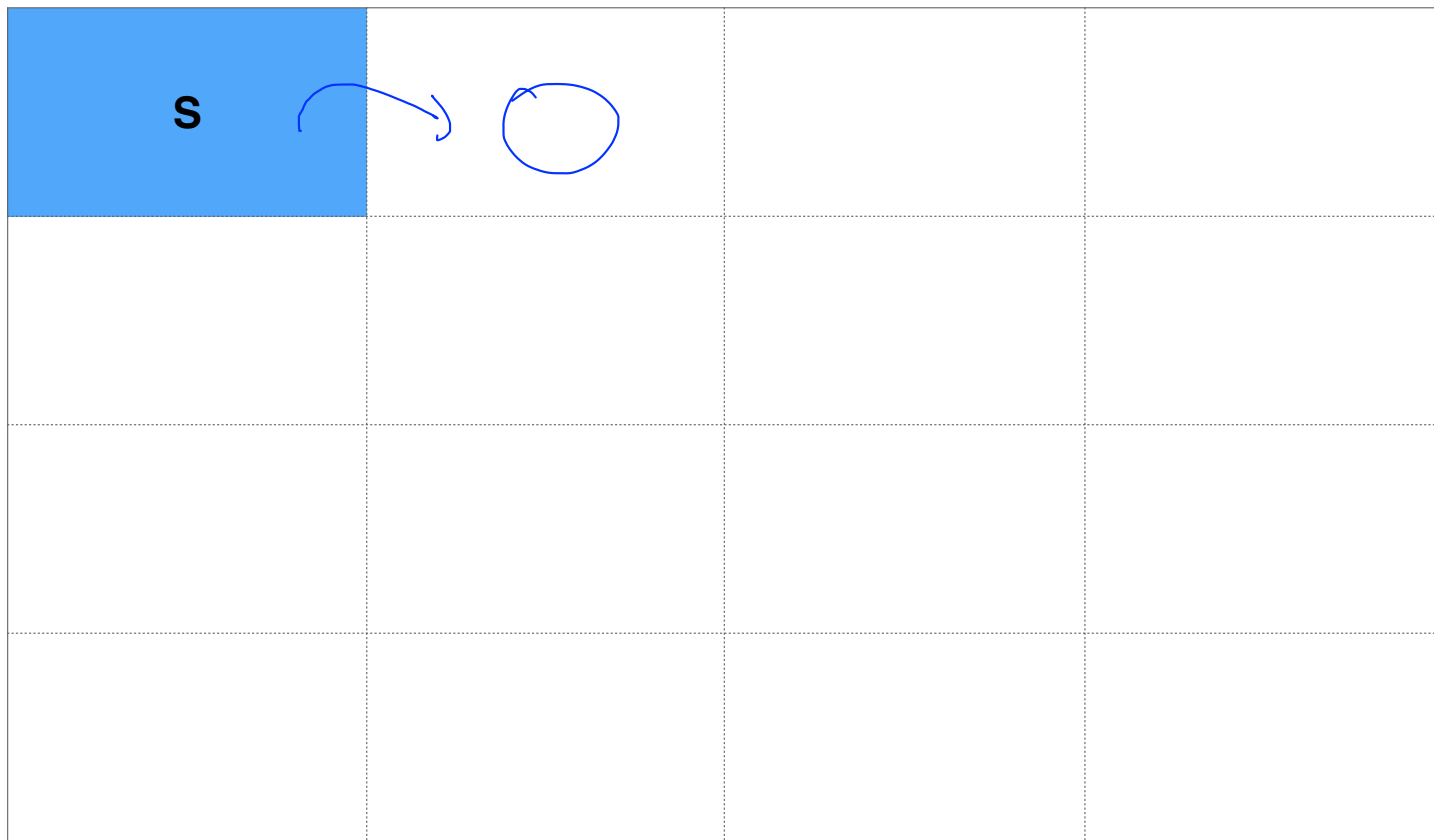
SFFF
FHFH
FFFH
HFFG
(Right)
('State: ', 1, 'Action: ', 2, 'Reward: ', 0.0, 'Info: ', {'prob': 1.0})
SFFF
FHFH
FFFH
HFFG
(Right)
('State: ', 2, 'Action: ', 2, 'Reward: ', 0.0, 'Info: ', {'prob': 1.0})
SFFF
FHFH
FFFH
HFFG
(Down)
('State: ', 6, 'Action: ', 1, 'Reward: ', 0.0, 'Info: ', {'prob': 1.0})
```

* run in terminal. Keyin does not work in PyCharm!

Frozen Lake World (OpenAI GYM)



NEXT: Try Frozen Lake Real Game?



Next
Q-learning (Table)

