**qLearn.ipynb:**

{

"cells": [

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": [

"import gym\n",

"from random import randint\n",

"import numpy as np"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": [

"env = gym.make(\"MountainCar-v0\")"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": [

"def discretizar(valor):\n",

" aux = ((valor - env.observation\_space.low) / (env.observation\_space.high - env.observation\_space.low))\*20\n",

" return tuple(aux.astype(np.int32))"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": [

"q\_table = np.random.uniform(low = -1, high = 1, size = [20,20,3])"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": [

"taza\_aprendizaje = 0.1\n",

"factor\_descuento = 0.95\n",

"episodios = 5000\n",

"listado\_recompensas = []\n",

"\n",

"for episodio in range(episodios):\n",

" estado = discretizar(env.reset())\n",

" final = False\n",

" recompensa\_total = 0\n",

"\n",

" while not final:\n",

"\n",

" if randint(0,10) > 2:\n",

" accion = np.argmax(q\_table[estado])\n",

" else:\n",

" accion = randint(0,2)\n",

"\n",

" nuevo\_estado, recompensa, final, info = env.step(accion)\n",

" q\_table[estado][accion] = q\_table[estado][accion] + taza\_aprendizaje \* (recompensa + factor\_descuento \* np.max(q\_table[discretizar(nuevo\_estado)]) - q\_table[estado][accion])\n",

" estado = discretizar(nuevo\_estado)\n",

" recompensa\_total += recompensa\n",

"\n",

" if (episodio + 1) % 500 == 0:\n",

" env.render()\n",

"\n",

" listado\_recompensas.append(recompensa\_total)\n",

" if (episodio + 1) % 100 == 0:\n",

" print(f'Espisodio {episodio + 1} - Recompensa: {np.mean(listado\_recompensas)}')\n",

"\n",

" env.close()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.9.13"

},

"orig\_nbformat": 4

},

"nbformat": 4,

"nbformat\_minor": 2

}