COLOMBIA POTENCIA DE LA VIDA

TIC

Actividad 4

Ejemplos y Aplicaciones

TALENTO TECH

AZ | PROYECTOS EDUCATIVOS

UTP Universidad Tecnológica de Pereira

Ejemplos y Aplicaciones

En la actividad de Ejemplos y Aplicaciones, los estudiantes explorarán ejemplos prácticos de Aprendizaje por Refuerzo (RL). Comenzarán con una demostración de la implementación de Q-Learning en un entorno de gridworld simple, donde podrán comprender cómo un agente aprende a navegar y tomar decisiones óptimas en un espacio discreto. Luego, se adentrarán en una discusión sobre las diversas aplicaciones del Aprendizaje por Refuerzo en el mundo real, que abarcan desde juegos hasta robótica y gestión de recursos. A través de ejemplos concretos y casos de uso prácticos, los estudiantes podrán apreciar cómo el Aprendizaje por Refuerzo puede mejorar la capacidad de los agentes para tomar decisiones autónomas en una variedad de contextos y situaciones.

Implementación de Q-Learning en un entorno de gridworld simple

Discusión sobre aplicaciones del Aprendizaje por Refuerzo en el mundo real

Discusión sobre aplicaciones del Aprendizaje por Refuerzo en el mundo real:

En esta parte, se muestran diversas aplicaciones del Aprendizaje por Refuerzo en el mundo real. En juegos, como el ajedrez o los videojuegos, el Aprendizaje por Refuerzo ha demostrado ser efectivo para entrenar agentes que compiten contra humanos o entre sí. En robótica, el Aprendizaje por Refuerzo se utiliza para entrenar robots autónomos que pueden realizar tareas complejas como la navegación en entornos desconocidos o la manipulación de objetos. En gestión de recursos, el Aprendizaje por Refuerzo puede optimizar la asignación de recursos en entornos dinámicos y cambiantes, como la gestión de la energía en edificios inteligentes o la gestión de inventarios en cadenas de suministro. Estos ejemplos ilustran cómo el Aprendizaje por Refuerzo tiene aplicaciones prácticas en una amplia variedad de campos y puede mejorar significativamente la eficiencia y la autonomía en diversas tareas y escenarios del mundo real.

Implementación de Q-Learning en un entorno de gridworld simple:

En esta demostración, los estudiantes aprenderán cómo implementar el algoritmo de Q-Learning en un entorno de gridworld simple. El gridworld es una cuadrícula donde un agente puede moverse entre celdas, con el objetivo de alcanzar una celda objetivo mientras evita obstáculos. A través de la implementación de Q-Learning, los estudiantes entenderán cómo el agente aprende a tomar decisiones óptimas basadas en las recompensas recibidas en cada estado y acción.

Ejercicio 1: Implementación de Q-Learning en un entorno de gridworld simple

```
import numpy as np

# Definición del gridworld

gridworld = np.array([

        [-1, -1, -1,  1],

        [-1, -1, -1, -1],

        [-1, -1, -1, -1],

        [-1, -1, -1, -1]

])


# Definición de las acciones posibles: arriba, abajo, izquierda,

derecha

acciones = [(0, -1), (0, 1), (-1, 0), (1, 0)]


# Implementación de Q-Learning

Q = np.zeros_like(gridworld)


gamma = 0.8

alpha = 0.1

num_episodes = 1000
```

```
for _ in range(num_episodes):

        estado = (0, 0)

        while estado != (0, 3):

        accion = np.random.choice(range(len(acciones)))

        nueva_fila = estado[0] + acciones[accion][0]

        nueva_col = estado[1] + acciones[accion][1]

        if 0 <= nueva_fila < gridworld.shape[0] and 0 <= nueva_col <

gridworld.shape[1]:

                recompensa = gridworld[nueva_fila, nueva_col]

                nuevo_valor = recompensa + gamma * np.max(Q[nueva_fila,

nueva_col])

                Q[estado][accion] = (1 - alpha) * Q[estado][accion] +

alpha * nuevo_valor

                estado = (nueva_fila, nueva_col)


print("Q-values después del entrenamiento:")

print(Q)
```

Ejercicio 2: Aplicación del Aprendizaje por Refuerzo en juegos

```
# Ejercicio 2: Aplicación del Aprendizaje por Refuerzo en juegos

# Ejemplo: Implementación de Q-Learning para un juego simple


import numpy as np


# Definición de las recompensas del juego (datos ficticios)

recompensas = {

        'ganar': 1,

        'perder': -1,

        'empatar': 0

}
```

```
# Implementación de Q-Learning para el juego

Q = {}


def q_learning_juego(estado_actual, accion, nuevo_estado, resultado):

        if estado_actual not in Q:

        Q[estado_actual] = np.zeros(len(acciones))

        if nuevo_estado not in Q:

        Q[nuevo_estado] = np.zeros(len(acciones))

        nuevo_valor = recompensas[resultado] + gamma *

np.max(Q[nuevo_estado])

        Q[estado_actual][accion] = (1 - alpha) *

Q[estado_actual][accion] + alpha * nuevo_valor


# Uso de la función q_learning_juego para actualizar los valores Q

# Se asume que se ha realizado una partida y se tiene información

sobre los estados, acciones, resultados, etc.

# Se llama a la función q_learning_juego con los parámetros adecuados
```

Ejercicio 3: Aplicación del Aprendizaje por Refuerzo en robótica

```
import numpy as np


# Definición del entorno de navegación (datos ficticios)
entorno = np.array([
        [0, 0, 0, 0, 0],
        [0, -1, -1, -1, 0],
        [0, 0, -1, 0, 0],
        [0, -1, -1, -1, 0],
        [0, 0, 0, 0, 0]
])


# Definición de acciones posibles (arriba, abajo, izquierda, derecha)
acciones = [(0, -1), (0, 1), (-1, 0), (1, 0)]
```

Ejercicio 3: Aplicación del Aprendizaje por Refuerzo en robótica

```
# Implementación de Q-Learning
Q = np.zeros((entorno.shape[0], entorno.shape[1], len(acciones)))

gamma = 0.9  # Factor de descuento
alpha = 0.1  # Tasa de aprendizaje
num_episodes = 1000

for _ in range(num_episodes):
        estado = (0, 0)
        while True:
        accion = np.random.choice(range(len(acciones)))
        nueva_fila = estado[0] + acciones[accion][0]
        nueva_col = estado[1] + acciones[accion][1]
        if 0 <= nueva_fila < entorno.shape[0] and 0 <= nueva_col <
entorno.shape[1]:
                recompensa = entorno[nueva_fila, nueva_col]
                nuevo_valor = recompensa + gamma * np.max(Q[nueva_fila,
nueva_col])
                Q[estado[0], estado[1], accion] = (1 - alpha) *
Q[estado[0], estado[1], accion] + alpha * nuevo_valor
                estado = (nueva_fila, nueva_col)
                if recompensa == 1:
                break

print("Valores Q después del entrenamiento:")
print(Q)
```

Ejercicio 4: Aplicación del Aprendizaje por Refuerzo en gestión de recursos

```
import numpy as np

#  Definición de los estados (niveles de inventario), acciones

(órdenes de reabastecimiento) y recompensas (costos, ganancias, etc.)

estados = ['Bajo', 'Medio', 'Alto']

acciones = ['Reabastecer', 'No reabastecer']

recompensas = {

        ('Bajo', 'Reabastecer'): 50,

        ('Bajo', 'No reabastecer'): -10,

        ('Medio', 'Reabastecer'): 30,

        ('Medio', 'No reabastecer'): 0,

        ('Alto', 'Reabastecer'): 10,

        ('Alto', 'No reabastecer'): -20

}
```

```
# Implementación de Q-Learning

Q = {}


gamma = 0.9  # Factor de descuento

alpha = 0.1  # Tasa de aprendizaje

num_episodes = 1000


for _ in range(num_episodes):

        estado_actual = np.random.choice(estados)

        while True:

        accion = np.random.choice(acciones)

        recompensa = recompensas[(estado_actual, accion)]

        if estado_actual not in Q:

                Q[estado_actual] = {}

        if accion not in Q[estado_actual]:

                Q[estado_actual][accion] = 0
```

Ejercicio 4: Aplicación del Aprendizaje por Refuerzo en gestión de recursos

```
            nuevo_estado = np.random.choice(estados)
            max_nuevo_estado = max(Q[nuevo_estado].values()) if
    nuevo_estado in Q else 0
            Q[estado_actual][accion] += alpha * (recompensa + gamma *
    max_nuevo_estado - Q[estado_actual][accion])
            estado_actual = nuevo_estado
            if recompensa == 50 or recompensa == 30 or recompensa == 10:
                break

print("Valores Q después del entrenamiento:")
print(Q)
```

COLOMBIA POTENCIA DE LA VIDA

TIC

TALENTO TECH

AZ | PROYECTOS EDUCATIVOS

UTP Universidad Tecnológica de Pereira