



ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

APLICACIÓN DE LAS TÉCNICAS DE OPTIMIZACIÓN ESTADÍSTICA ENFOCADA A LA COMPUTACIÓN GRÁFICA

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE MATEMÁTICO
APLICADO**

GEOCONDA DENNISSE MOLINA MORALES

geoconda.molina@epn.edu.ec

DIRECTOR: MÉNTHOR OSWALDO URVINA MAYORGA

menthor.urvina@epn.edu.ec

19 DE MARZO DE 2024

CERTIFICACIONES

Yo, GEOCONDA DENNISSE MOLINA MORALES, declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Geoconda Dennisse Molina Morales

Certifico que el presente trabajo de integración curricular fue desarrollado por Geoconda Dennisse Molina Morales, bajo mi supervisión.

Ménthor Oswaldo Urvina Mayorga
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el(los) producto(s) resultante(s) del mismo, es(son) público(s) y estará(n) a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Geoconda Dennisse Molina Morales

Ménthor Oswaldo Urvina Mayorga

RESUMEN

(Máximo 250 palabras)

Palabras clave: palabra1, palabra2, palabra3,..., palabra6.

ABSTRACT

(Máximo 250 palabras)

Keywords: palabra1, palabra2, palabra3, ..., palabra6.

Índice general

1. Descripción del componente desarrollado	1
1.1. Descripción del Proyecto	1
1.2. Objetivo general	1
1.3. Objetivos específicos	2
1.4. Alcance	2
2. Marco teórico	4
2.1. Conceptos Fundamentales	4
2.1.1. Agente	4
2.1.2. Entorno	4
2.1.3. Estado	4
2.1.4. Acción	5
2.1.5. Recompensa	5
2.1.6. Política	5
2.2. Rainforcement Learning	5
2.2.1. Ecuacion de Bellman	7
2.2.2. Ecuación de Bellman para procesos estocásticos . . .	8
2.3. Factor de penalización	9
2.4. Q-Learning	10
2.5. Diferencial Temporal	12

2.6. Redes Neuronales Convolucionales	12
3. Metodología	13
3.1. Agente q-learning	13
3.1.1. Implementación del agente	13
3.1.2. Pruebas en entornos	13
3.2. Redes CNN	13
3.2.1. Implementación	13
3.2.2. Pruebas en tornos	13
3.2.3. Integración de las redes CNN al agente q-learning . .	13
3.3. Recolección de datos	13
4. Resultados	14
5. Conclusiones y recomendaciones	15
5.1. Conclusiones	15
5.2. Recomendacione	15
A. Anexos	16
A.1. fff	16
Bibliografía	16

Índice de figuras

Capítulo 1

Descripción del componente desarrollado

1.1. Descripción del Proyecto

Para explicar de mejor manera las aplicaciones de las técnicas de optimización estadística, los modelos estadísticos y los conceptos estadísticos en la computación gráfica, se plantea un proyecto desarrollado con el lenguaje de programación Python, el cual consiste en un agente entrenado por q- learning del cual se derivan las explicaciones de estadística, y de una red neuronal convolucional (CNN), la cual es el modelo estadístico que procesa imágenes y que utiliza técnicas de optimización estadística para aproximar de mejor manera sus resultados, así, el agente q-learner conectado a la red CNN utiliza las salidas del modelo CNN para ajustar su función de calidad y así tomar elecciones que le permita tener un aprendizaje con cada decisión tomada en el entorno en el que se encuentra.

1.2. Objetivo general

Investigar y desarrollar un agente q-learning que incorpore redes neuronales convolucionales para la realización de tareas en entornos gráficos, con el fin de demostrar cómo los matemáticos pueden contribuir al mundo del arte gráfico y la computación gráfica, destacando el papel crucial de la optimización estadística aportada por los matemáticos en

el desarrollo de inteligencias artificiales, específicamente en el campo del deep learning, con el fin de demostrar su relevancia en la mejora de la eficiencia y precisión en la resolución de problemas prácticos.

1.3. Objetivos específicos

- Desarrollar un agente q-learning que integre redes neuronales convolucionales.
- Crear un entorno virtual que permita al agente realizar diversas tareas de las cuales aprende sucesivamente.
- Recopilar datos mediante la observación y registro de las acciones y decisiones tomadas por el agente durante su proceso de aprendizaje en el entorno gráfico.
- Analizar cómo la integración de redes neuronales convolucionales y técnicas de optimización estadística influye en la capacidad de aprendizaje y ejecución de tareas del agente en entornos gráficos, identificando las mejoras significativas en términos de eficiencia y precisión.
- Promover la colaboración entre matemáticos y expertos en el desarrollo de tecnologías avanzadas.

1.4. Alcance

Este estudio se enfocará en el desarrollo e implementación de un agente de aprendizaje basado en el algoritmo Q-Learning, integrado con redes neuronales convolucionales, para la realización de tareas en entornos gráficos. El alcance del estudio abarcará la creación de un entorno virtual que simule situaciones y tareas, permitiendo la observación y registro de las acciones y decisiones tomadas por el agente durante su proceso de aprendizaje.

El período de estudio se extenderá desde la implementación del agente hasta la conclusión del análisis de los datos recopilados, con una duración estimada de cuatro meses.

Los datos utilizados en este estudio serán generados internamente por el entorno virtual y consistirán en registros de las interacciones del agente con su entorno gráfico. No se utilizarán datos de usuarios reales ni se recopilará información personal para este propósito.

Los métodos de análisis incluirán técnicas de optimización estadística para mejorar la eficiencia y precisión del agente en la toma de decisiones, así como la evaluación de su desempeño en la realización de actividades dentro del entorno gráfico. Además, se analizará cómo la integración de redes neuronales convolucionales y técnicas de optimización estadística influye en la capacidad de aprendizaje y ejecución de tareas del agente, identificando mejoras significativas en términos de eficiencia y precisión.

Capítulo 2

Marco teórico

2.1. Conceptos Fundamentales

2.1.1. Agente

Un agente es una entidad que toma decisiones y realiza acciones en un entorno con el objetivo de alcanzar ciertos objetivos o maximizar una recompensa acumulada a lo largo del tiempo. El agente puede ser cualquier sistema o programa que interactúe con su entorno, como un robot, un software de juego, etc.

2.1.2. Entorno

El entorno representa el mundo en el que el agente interactúa y toma decisiones. Puede ser cualquier sistema o situación en la que el agente esté operando, como un videojuego, un robot físico, o un entorno simulado. El entorno proporciona retroalimentación al agente en forma de recompensas o penalizaciones según las acciones que tome.

2.1.3. Estado

El estado es una representación del entorno en un momento dado. Proporciona información sobre la situación actual del agente, incluida

su ubicación, las condiciones del entorno y cualquier otra información relevante para la toma de decisiones.

2.1.4. Acción

Una acción es cualquier movimiento o decisión que el agente puede realizar en respuesta a un estado dado. Las acciones pueden ser discretas o continuas, dependiendo de la naturaleza del problema y de las capacidades del agente.

2.1.5. Recompensa

La recompensa es una señal de retroalimentación que el entorno proporciona al agente en respuesta a una acción específica. La recompensa indica cuán beneficiosa o perjudicial fue la acción tomada por el agente en un estado dado y sirve como guía para el aprendizaje.

2.1.6. Política

La política de un agente es una estrategia que determina qué acción debe tomar el agente en un determinado estado del entorno. Es esencialmente una función que asigna acciones a estados con el fin de maximizar la recompensa esperada a largo plazo. La política puede ser determinista o estocástica, dependiendo de si las acciones son completamente predecibles o tienen cierta aleatoriedad.

2.2. Reinforcement Learning

También conocido como aprendizaje por refuerzo, constituye una rama fundamental dentro del ámbito del aprendizaje automático y el control óptimo. Este enfoque se centra en la toma de decisiones de agentes inteligentes en entornos dinámicos con el objetivo de maximizar la recompensa acumulativa. Junto con el aprendizaje supervisado (Machine Learning) y el aprendizaje no supervisado (Unsupervised Learning), el aprendizaje por

refuerzo conforma uno de los paradigmas básicos del aprendizaje automático.

”A diferencia de los métodos de gradiente de políticas, que intentan aprender funciones que asignan directamente una observación a una acción, Q-Learning intenta aprender el valor de estar en un estado determinado y realizar una acción específica allí.”[2]

En su esencia, el aprendizaje por refuerzo se preocupa por encontrar un equilibrio entre la exploración de territorios inexplorados y la explotación del conocimiento existente, todo ello con el fin de maximizar la recompensa a largo plazo. Es crucial destacar que la retroalimentación en este proceso puede ser incompleta o experimentar ciertos retrasos.

El entorno, en este contexto, se modela mediante un proceso de decisión de Markov (MDP). Muchos algoritmos de aprendizaje por refuerzo emplean técnicas de programación dinámica para abordar este tipo de entorno.

La diferencia fundamental entre los métodos clásicos de programación dinámica y los algoritmos de aprendizaje por refuerzo se encuentra en su aproximación al conocimiento del modelo matemático exacto del proceso de decisión de Markov.

En la programación dinámica clásica, se parte del supuesto de tener un conocimiento preciso y completo del modelo matemático del proceso de decisión de Markov.[4] Esto implica conocer de antemano todas las transiciones posibles entre estados, así como las probabilidades asociadas y las recompensas esperadas. Estos métodos clásicos utilizan esta información para calcular de manera óptima las políticas de decisión, buscando la solución más eficiente para el problema.

En cambio, los algoritmos de aprendizaje por refuerzo no asumen previamente este conocimiento detallado del modelo. En lugar de eso, estos

algoritmos están diseñados para aprender y adaptarse directamente del entorno mediante la interacción continua. Utilizan la experiencia acumulada a lo largo del tiempo para mejorar gradualmente su comprensión del modelo del proceso de decisión de Markov, ajustando sus estrategias de decisión para maximizar la recompensa a largo plazo. Este enfoque es particularmente valioso en situaciones donde obtener información completa y precisa sobre el modelo resulta difícil o impracticable, permitiendo así que los algoritmos de aprendizaje por refuerzo se desempeñen eficazmente en entornos dinámicos y complejos.

2.2.1. Ecuación de Bellman

La ecuación de Bellman en el contexto del aprendizaje por refuerzo es una herramienta conceptual clave que nos ayuda a entender cómo asignar valores a acciones o estados en un entorno dinámico. En términos simples, esta ecuación establece que el valor de una elección o situación se compone de dos partes: la recompensa inmediata y la estimación del valor futuro.

Cuando tomamos una acción o nos encontramos en un estado específico, recibimos una recompensa instantánea. Sin embargo, la ecuación de Bellman reconoce que la toma de decisiones no solo se trata de recompensas inmediatas, sino también de cómo esas decisiones afectan las recompensas a largo plazo. Por lo tanto, el valor de una acción o estado incluye no solo la recompensa inmediata, sino también la expectativa de las recompensas futuras.

Esta expectativa de recompensas futuras se pondera mediante un factor de descuento. Este factor refleja la idea de que las recompensas futuras son menos valiosas que las recompensas inmediatas. Es decir, valoramos más una recompensa hoy que una recompensa de igual cantidad en el futuro.

En este contexto podemos observar la ecuación de Bellman de manera matemática de la siguiente forma[1]:

Sean:

$s = \text{estados}$

$a = \text{acciones}$

$R = \text{recompensa}$

$\gamma = \text{factor descuento}$

Se considera los conjuntos $A = \{a_1, a_2, \dots, a_n\}$ y $S = \{s_1, s_2, \dots, s_n\}$ de acciones y estados, ambos finitos.

Inicialmente se tratará el caso para procesos deterministas, es decir, conocemos el valor $V(s')$ de ante mano para todos los estados futuros, así la ecuación se ve de la siguiente manera:

$$V(s) = \max_a [R(s, a) + \gamma V(s')]$$

Por lo que, se calcula $V(s)$ tomando el máximo de las recompensas obtenidas eligiendo la accion a , en el estado s más el factor de descuento γ por el valor futuro en el estado s' al cual se llega despues de tomar la acción a .

2.2.2. Ecuación de Bellman para procesos estocásticos

Debido a que en la vida real no existen procesos deterministas, se debe incluir la parte estocástica en la ecuación tradicional de Bellman. En un entorno estocástico, las acciones no conducen siempre al mismo resultado; en cambio, hay una probabilidad asociada con cada transición entre estados.

En este contexto, la ecuación de Bellman refleja la idea de que el valor de una elección o estado se calcula tomando en cuenta la recompensa inmediata y la expectativa de recompensas futuras, pero ahora incorpora la aleatoriedad de las transiciones entre estados. Esto significa que, al tomar una acción, no podemos predecir con certeza el próximo estado; en su lugar, debemos considerar las probabilidades asociadas con cada

posible transición.[1]

El factor de descuento sigue siendo relevante en un entorno estocástico y sirve para ponderar las recompensas futuras, reconociendo que la incertidumbre puede afectar la magnitud de esas recompensas.

Sabiendo que el proceso que se está modelando no es un proceso determinista, de la ecuación original de Bellman se modifica los valores $V(s')$ ya que cada decisión tiene una cierta probabilidad, entonces se desea encontrar el valor esperado de tomar una cierta decisión s' es decir, se quiere encontrar el valor: $\sum_{s'}^S P(s, a, s') * V(s')$, el cuál es la suma de los valores futuros $V(s')$ ponderados por la probabilidad de llegar al estado s' tomando la acción a , desde en estado s .

Por lo que, considerando lo antes mencionado en la ecuación inicial se tiene:

$$V(s) = \max_a [R(s, a) + \gamma \sum_{s'}^S P(s, a, s') * V(s')]$$

Es decir, la ecuación inicial se modifica considerando la esperanza de $V(s')$, así:

$$V(s) = \max_a [R(s, a) + \gamma E[V(s')]]$$

2.3. Factor de penalización

Esta técnica implica la aplicación de una penalización o castigo constante a cada paso de tiempo que el agente toma mientras interactúa con su entorno.[3] La idea detrás de la penalización de vida es incentivar al agente a alcanzar su objetivo lo más rápido posible, evitando así comportamientos indeseados que prolonguen innecesariamente el proceso de aprendizaje.

Por otro lado, es importante considerar el impacto de diferentes tipos de recompensas en el aprendizaje por refuerzo. Las recompensas positivas y negativas, junto con las recompensas iguales, desempeñan

roles distintos en la motivación y el comportamiento del agente. Mientras que las recompensas positivas pueden incentivar al agente a repetir ciertos comportamientos, las recompensas negativas pueden desalentar comportamientos no deseados. En algunos casos, todas las acciones pueden tener la misma recompensa, lo que puede influir en la exploración y la explotación del agente.[5]

Finalmente, es crucial considerar cómo la relación entre la recompensa total y la ganancia en la meta puede afectar el comportamiento del agente. Si la recompensa inmediata es significativamente mayor que la ganancia en la meta a largo plazo, el agente puede verse tentado a buscar gratificaciones instantáneas en lugar de perseguir objetivos a largo plazo. Esto puede conducir a comportamientos subóptimos y afectar el proceso de aprendizaje del agente en su conjunto.

2.4. Q-Learning

El Q-Learning es un algoritmo fundamental, que se utiliza para aprender una política óptima de comportamiento en un entorno desconocido y estocástico. La esencia del Q-Learning radica en la estimación de la función de valor de acción óptima, conocida como Q-valor, para cada par de estado-acción posible en el entorno. La función de valor $Q(s, a)$ representa el valor esperado acumulado que se obtiene al tomar la acción a en el estado s y luego seguir una política óptima.

Para comprender mejor el Q-Learning, es esencial entender la actualización de los valores Q utilizando la ecuación de Bellman. La ecuación de Bellman para la función de valor $Q(s, a)$ se define como:

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

Donde:

- $Q(s, a)$ es el valor Q para el estado 's' y la acción 'a'.
- $R(s, a)$ es la recompensa inmediata obtenida al tomar la acción 'a' en

el estado 's'.

- γ es el factor de descuento que determina la importancia relativa de las recompensas futuras.
- s' es el estado resultante después de tomar la acción 'a'.
- $\max_{a'} Q(s', a')$ es el valor máximo de Q para el estado 's' y todas las acciones posibles 'a'.

La ecuación de Bellman establece que el valor Q para un par estado-acción debe ser igual a la recompensa inmediata más el valor Q esperado del mejor próximo estado y acción.[4] La actualización de los valores Q se realiza iterativamente a medida que el agente interactúa con el entorno y recibe retroalimentación en forma de recompensas.

El algoritmo Q-Learning utiliza una estrategia de exploración y explotación para aprender de manera eficiente la política óptima. Durante la fase de exploración, el agente toma acciones aleatorias para explorar el espacio de estados y acciones. Durante la fase de explotación, el agente elige las acciones que tienen el mayor valor Q estimado para el estado actual.

La actualización de los valores Q en el Q-Learning se realiza utilizando la regla de actualización Q , que se define como:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \cdot \left[R(s, a) + \gamma \cdot \max_{a'} Q(s', a') - Q(s, a) \right]$$

Donde:

- α es la tasa de aprendizaje que determina la rapidez con la que se actualizan los valores Q .

Es decir, se refiere a la evaluación de la calidad de las acciones que se podría tomar estando en un estado determinado.

2.5. Diferencial Temporal

Se refiere a la diferencia de las calidades tomadas antes y después

2.6. Redes Neuronales Convolucionales

Capas ocultas

Capas visibles

Capítulo 3

Metodología

3.1. Agente q-learning

3.1.1. Implementación del agente

3.1.2. Pruebas en entornos

3.2. Redes CNN

3.2.1. Implementación

3.2.2. Pruebas en tornos

3.2.3. Integración de las redes CNN al agente q-learning

3.3. Recolección de datos

Capítulo 4

Resultados

Capítulo 5

Conclusiones y recomendaciones

5.1. Conclusiones

5.2. Recomendacione

Capítulo A

Anexos

A.1. fjfj

Referencias bibliográficas

- [1] Richard Ernest Bellman. The theory of dynamic programming, Oct 2008.
- [2] Arthur Juliani. Simple reinforcement learning with tensorflow part 0: Q-learning with tables and neural networks, Aug 2016.
- [3] Volodymyr Mnih et al. Asynchronous methods for deep reinforcement learning. 2016.
- [4] John Schulman et al. High-dimensional continuous control using generalized advantage estimation. 2015.
- [5] John Schulman et al. Trust region policy optimization. 2015.