Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería Eléctrica
IEE3664 – Control Inteligente

Tarea 4: Aprendizaje Reforzado (2025-S2)

Hint: Revisar la sección *Nota de la tarea* en el anexo.

Problema 1: Avance del Proyecto (1 Punto)

Para esta entrega se espera que presenten un avance equivalente al menos al 30 % del proyecto final. Esto implica incluir una descripción clara y detallada de la metodología que están utilizando, así como resultados preliminares que evidencien el progreso.

Ejemplos de avances aceptables (además de la descripción de la metodología):

- Haber recopilado y preprocesado el conjunto de datos que utilizarán, acompañado de un análisis básico o visualizaciones.
- Haber implementado la arquitectura inicial del modelo y mostrar los primeros gráficos de pérdida, recompensas o métricas de desempeño.
- Tener un entorno de simulación funcionando, con las primeras pruebas de interacción entre un controlador de prueba o baseline y el entorno.

Problema 2: RL Tabular (2 Puntos)

En este problema tendrá que resolver el laberinto *Frozen Lake* de la librería *gymnasium* [1] con métodos tabulares. El script de ayuda *p2.py* muestra como importar el ambiente, y el script *p2_functions.py* tiene algunas funciones de ayuda tal como la implementación del algoritmo *value iteration*. Utilizando este ambiente usted deberá:

a) Analizar cómo cambia la función de valor óptima al variar (uno a la vez):

- El parámetro *success_rate* de 1/3 a 1 y luego a 1/5.
- El factor de descuento γ de 0.99 a 0.95

Comente las razones de los cambios observados en la función de valor.

b) Analizar cómo cambia la política óptima al variar (uno a la vez):

- *reward_schedule* de (1, 0, 0) a (1, 0, -1) (ahora penalizamos cada movimiento).
- El parámetro *is_slippery* a False.

Comente las razones de los cambios observados en la política.

c) Implementar First-visit Monte Carlo (FVMC) y Every-visit Monte Carlo (EVMC) utilizando las mejores prácticas (**Hint**: revise la clase 24) y grafique los valores estimados para cada posición del laberinto y el retorno promedio (puede graficar un promedio móvil del retorno a través de los episodios para más claridad). Responda:

- ¿Cuál método converge más rápido y por qué?
- ¿Cuál método presenta mayor varianza en las estimaciones de valor? ¿A qué se debe?

d) Implementar el método SARSA y Q-learning utilizando las mejores prácticas y grafique los valores estimados para cada posición del laberinto y el retorno promedio. Realice las siguientes comparaciones:

- Compare la velocidad de convergencia de ambos métodos y el retorno promedio obtenido.
- Mantenga el rate de exploración ϵ en 1. Compare los valores y los retornos obtenidos por ambos métodos ¿A qué se debe esta diferencia?
- En teoría ¿Cuál de los dos métodos es más conservador? ¿Por qué?

[1] `https://gymnasium.farama.org/environments/toy_text/frozen_lake/`

Problema 3: Deep Q-learning (3 Puntos)

Tras una vida de excesos, Alan Brito comenzó una mañana como cualquier otra: tres panes, una torta y un café con cuatro cucharadas de azúcar. Sin embargo, algo no estaba bien. A mitad del día empezó a sentirse inusualmente cansado y débil, tanto que decidió (por primera vez en mucho tiempo) ir al médico. Después de varios exámenes, el doctor le dio el diagnóstico: Contra todo pronóstico, y pese a lo que cualquiera habría pensado basándose en su dieta, Alan Brito no tenía diabetes Tipo 2, sino diabetes Tipo 1 [2]

Una vez más, Alan Brito recurre a usted por sus conocimientos en control inteligente para que lo ayude a manejar su glucosa sanguínea mediante un páncreas artificial entrenado con aprendizaje reforzado. Para ello pone a su disposición un simulador del sistema glucorregulatorio de personas con diabetes Tipo 1 [3], con el objetivo de resolver el problema de control de **glucosa postprandial** (es decir, después de comer).

El control postprandial consiste en determinar el bolo de insulina óptimo para cada paciente y para cada ingesta de comida, de manera que se cumplan los siguientes objetivos:

- L_1: Reducir el valor máximo alcanzado por la glucosa después de comer (evitar hiperglicemias).
- L_2: Aumentar el valor mínimo alcanzado por la glucosa después de comer (evitar hipoglicemias).
- L_3: Minimizar la desviación, respecto al nivel basal de glucosa del paciente, al final del periodo postprandial (6–8 horas).

Generalmente, se considera que los valores seguros de glucosa se encuentran entre 80 y 180 mg/dL, por lo que las métricas L_1 y L_2 pueden evaluarse tanto respecto a estos límites como respecto al valor basal individual (valor de glucosa en equilibrio sin comidas) de cada paciente. La figura 1 muestra los puntos importantes de la respuesta postprandial.

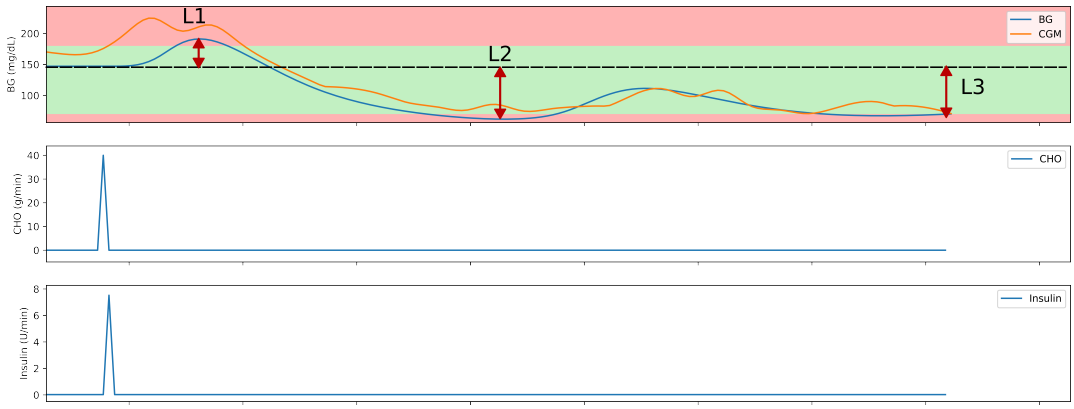

Figura 1: Respuesta Postprandial

Para realizar esta tarea, se le entrega algunos scripts a medias con el algoritmo Deep Q-learning que otro ingeniero, Eugenio Epsilon, no logró terminar. Los scripts son los siguientes:

- *dqn.py*: Algortimo DQN terminado a medias
- *replay_memory.py*: Tiene el buffer de *Experience Replay* correctamente implementado
- *p3.py*: Contiene un loop de aprendizaje donde, para un paciente escogido al azar y una comida de tamaño aleatorio, el controlador debe determinar el bolo de insulina adecuado en el momento de la ingesta. Para este problema, usted podrá leer directamente el valor de glucosa en sangre (omitiremos el uso del sensor CGM), accesible mediante `info['bg']`. Asimismo, podrá consultar el tamaño de la comida utilizando `info['meal']`. Por otro lado, se asume que la insulina basal (la insulina necesaria para mantener al paciente en su nivel de glucosa basal) es conocida para cada paciente y se administra en cada paso de tiempo. Este valor también está disponible para su lectura. Cada episodio durará 8 horas.

[2] `https://www.mayoclinic.org/es/diseases-conditions/type-1-diabetes/symptoms-causes/syc-20353011`
[3] `https://github.com/jxx123/simglucose`

Utilizando estos scripts ustede deberá:

a) Completar la función `calc_action()` dentro del script *dqn.py*, considerando que el bolo máximo de insulina es 10 unidades. (**Hint:** Recuerde que DQN trabaja en un espacio de acciones discreto)

b) Modificar la función `learn()` en *dqn.py* para implementar la variante Double DQN, que deberá activarse únicamente cuando el parámetro `double_dqn = True`.

c) Considerando las restricciones del problema (su controlador DQN solo actuará una vez en todo el periodo y la información asociada a L_1, L_2 y L_3), defina:

- Una representación de estado que, de forma aproximada, satisfaga la propiedad de Markov.
- Una función de reward adecuada al objetivo de control.
- Las transiciones relevantes que deben almacenarse en el buffer de *Experience Replay* mediante la función `collect_experience()`.
- El criterio para decidir cuándo debe aprender el modelo (ejecución de la función `learn()`): ¿En cada paso? ¿Al final del episodio? ¿En ambos? ¿Por qué?

d) Entrene su agente de DQN por un número determinado de episodios que usted determine conveniente y grafique la retorno acumulado, la pérdida de su modelo y el retorno acumulado por paciente a través de todos los episodios de entrenamiento. Compare los resultados entre DQN regular y *Double DQN*.

e) Con el mejor agente obtenido, grafique:

- El bolo de insulina determinado por el controlador para cada individuo, considerando comidas de tamaño 30, 70 y 120 gramos.
- La respuesta de la glucosa en cada caso y para cada individuo.

Recurde mostrar el retorno acumulado para cada caso.

Anexo

Nota de la tarea

La nota de esta tarea estará dada por las condiciones del siguiente **sistema experto**:

- IF $N_{T_4} \geq 4$ THEN $NF_{T_4} = \text{máx}(N_{T_4}, NP)$
- IF $N_{T_4} < 4$ THEN $NF_{T_4} = N_{T_4}$

Donde:

- NT_4: Nota preliminar de esta tarea
- NP: Nota promedio de las tareas anteriores, es decir $NP = \frac{N_{T_1}+N_{T_2}+N_{T_3}}{3}$
- NF_{T_4}: Nota final de esta tarea.

Informe

El informe debe ser realizado en Latex, utilizando el template en Canvas. Sin embargo, el Problema 1 podrá entregarse en otro documento con el formato del proyecto (dos columns estilo IEEE).

Código

- **Todo su código debe ser realizado en python version** $\geq$ 3.11. Para la implementación de sus redes neuronales **debe utilizar la librería Pytorch**.
- **Está estrictamente prohibido el uso de Chatbots para completar su tarea. Si el profesor o los ayudantes determinan que utilizó un chatbot durante el desarrollo la tarea se calificará con Nota 1 (Ver diapositivas de la primera clase para el detalle de los lineamientos del curso en este tema).**