NOMBRE: Rodrigo Ogalde Lisboa SECCIÓN: 1

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2613 — Inteligencia Artificial — 1' 2023

Tarea 4 – Respuesta Pregunta 1.5

En esta implementación en particular, el overfitting puede ser útil porque el conjunto de datos de entrenamiento representa todas las posibles combinaciones de estados de juego. Es decir, el conjunto de entrenamiento es el universo completo de todas las posibles situaciones que el agente puede encontrar durante el juego. Por lo tanto, "aprender de memoria" los datos en este caso, derivará en que la red neuronal aprenderá la acción más óptima para cada posible estado del juego.
Esto no es el caso para la mayoría de las aplicaciones, ya que en la mayoría de los casos, el conjunto de entrenamiento es una muestra del universo de las posibles entradas y que buscamos que el modelo sea capaz de generalizar a partir de una muestra para poder hacer predicciones precisas sobre datos no revisados con anterioridad.

a) Predicción del precio de las viviendas: Si entrenamos un modelo para predecir el precio de las viviendas basándonos en características como el tamaño, la ubicación, el número de habitaciones, etc., no queremos que nuestro modelo se ajuste demasiado a los datos de entrenamiento. Esto se debe principalmente a que si queremos predecir los precios de las casas que se vendan a futuro, pueden ser que no sean exactamente iguales a las casas en nuestro conjunto de datos de entrenamiento. Si nuestro modelo está demasiado ajustado a los datos de entrenamiento, puede que no sea capaz de hacer predicciones precisas sobre estas nuevas casas.

b) Detección de spam en correos electrónicos: Si entrenamos un modelo para detectar spam en correos electrónicos, no queremos que nuestro modelo se ajuste demasiado a los datos de entrenamiento. Esto se debe a que los spammers cambian constantemente sus tácticas y los correos electrónicos de spam que se envían en el futuro pueden no ser exactamente iguales a los correos electrónicos de spam en nuestro conjunto de datos de entrenamiento. Si nuestro modelo está demasiado ajustado a los datos de entrenamiento, puede que no sea capaz de detectar correctamente estos nuevos tipos de spam.

La función de activación softmax se utiliza en la capa de salida de la red neuronal porque proporciona una distribución de probabilidad. En otras palabras, la función softmax toma un vector de números reales y los transforma en valores de probabilidad que suman 1. Esto es útil en este caso porque queremos que la red neuronal nos proporcione la probabilidad de cada posible acción que el agente puede tomar. La acción con la mayor probabilidad será la acción seleccionada por el agente.

En cuanto a la estructura de la red neuronal,

- Gato: El modelo contiene cinco capas en total. La primera capa es la capa de entrada, que tiene 4 neuronas correspondientes a las 4 características de entrada (las posiciones del gato y del ratón). Luego, hay cuatro capas ocultas con 50, 100, 500 y 900 neuronas respectivamente. Finalmente, la capa de salida tiene 5 neuronas, correspondientes a las 5 posibles acciones que el agente puede tomar.
- Ratón: El modelo contiene cuatro capas en total. La primera capa es la capa de entrada, que tiene 4 neuronas correspondientes a las 4 características de entrada (las posiciones del gato y del ratón).

Luego, hay tres capas ocultas con 50, 100, y 500 neuronas respectivamente. Finalmente, la capa de salida tiene 5 neuronas, correspondientes a las 5 posibles acciones que el agente puede tomar.

La elección del número de capas y neuronas en cada capa fue un proceso de prueba y error. En este caso, se eligió un modelo con varias capas ocultas y un número creciente de neuronas en cada capa para permitir que la red capture relaciones más complejas en los datos.

NOMBRE: Rodrigo Ogalde Lisboa SECCIÓN: 1

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2613 — Inteligencia Artificial — 1' 2023

Tarea 4 – Respuesta Pregunta 2.4

- ¿Qué rol cumple la tasa de descuento en Q-Learning?
  La tasa de descuento, en este caso denotada como gamma, determina cuánto valor le da el agente a las recompensas futuras en comparación con las recompensas inmediatas. Un valor de gamma cercano a 0 hará que el agente valore las recompensas inmediatas sobre las futuras, mientras que un valor de gamma cercano a 1 hará que el agente valore las recompensas futuras de manera similar a las recompensas inmediatas. Es decir, la tasa de descuento determina cuánto "mira hacia el futuro" el agente.

- ¿Qué tasa de descuento te dio mejores resultados? ¿Por qué crees que fue así?
  Luego de realizar varios entrenamientos, logré notar que tener una taza de descuento baja es mejor, pero toma un mayor tiempo para que aprenda el modelo y pueda jugar con mejor precisión.
  Dentro de los entrenamientos, pude notar que al tener una baja taza de descuento permitió al modelo probar distintos movimientos "más arriesgados", lo que derivó en que tuviera más oportunidades en lograr un avance a que si hubiera sido con una taza alta de entrenamiento.

- ¿Para qué sirve el learning rate? ¿Qué valor te entregó mejores resultados? Comenta al respecto. El learning rate, en este caso denotado como alpha (), determina cuánto se actualiza el valor Q en cada iteración del algoritmo Q-Learning. Un valor de alpha de 0 significa que el valor Q no se actualiza en absoluto, mientras que un valor de alpha de 1 significa que el valor Q se actualiza completamente con la nueva información obtenida. Un valor de alpha intermedio permitirá que el valor Q se actualice parcialmente, conservando algo de la información antigua y agregando algo de la nueva información. El valor del learning rate con los que obtuve mejores resultados fue con uno intermedio (aproximadamente 0.5), ya que a partir de esto, le permitió al modelo guardar información útil como los "dead-ends" o lugares que son más probables que se encuentre atrapado. Tambien logré notar que con un valor bajo el modelo logra aprender de la misma manera, pero de una forma más lenta, lo que le toma más iteraciones para lograr el mismo resultado que si estuviera con un valor intermedio.

- Explica cómo diseñaste el reward de cada agente y la lógica detrás de tu decisión. Piensa en otro reward alternativo para cada agente (no la implementes), entrega una ventaja y una desventaja en comparación con el reward implementado.
  El reward para el gato se basa en la distancia al ratón. Si el gato captura al ratón, recibe una recompensa alta. Si se acerca al ratón, recibe una recompensa positiva. Si la distancia al ratón se mantiene igual, la recompensa es neutra, y si la distancia al ratón aumenta, la recompensa es negativa. Esto incentiva al gato a moverse hacia el ratón y capturarlo.
  Para el ratón, la lógica es similar pero invertida. Si el ratón es capturado, recibe una recompensa negativa. Si se aleja del gato, recibe una recompensa positiva. Si la distancia al gato se mantiene igual, la recompensa es neutra, y si la distancia al gato disminuye, la recompensa es negativa. Esto incentiva al ratón a alejarse del gato.
  Una alternativa podría ser dar una recompensa basada en el tiempo que el ratón logra evitar al gato, en lugar de la distancia al gato. Esto podría incentivar al ratón a moverse de manera más impredecible

y hacer que sea más difícil para el gato capturarlo. Sin embargo, también podría incentivar al ratón a moverse en círculos o en un patrón repetitivo, lo que podría no ser el comportamiento deseado.

NOMBRE: Rodrigo Ogalde Lisboa SECCIÓN: 1

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2613 — Inteligencia Artificial — 1' 2023

Tarea 4 – Respuesta Pregunta 3

- ¿Cuál de los dos modelos es más eficiente en términos de tiempo de entrenamiento? ¿Por qué crees que es así? El modelo de Aprendizaje Reforzado parece ser más eficiente en términos de tiempo de entrenamiento. En el mismo lapso de 40 minutos, el agente de Aprendizaje Reforzado fue capaz de completar 15100 juegos, mientras que el agente de Red Neuronal solo completó 1100 juegos. Esto puede deberse a que el Aprendizaje Reforzado puede aprender más rápidamente de sus errores y ajustar su comportamiento en consecuencia, mientras que las Redes Neuronales pueden requerir más tiempo para ajustar sus pesos y aprender de los errores.

- ¿Cuál de los dos modelos tuvo un mejor desempeño en el juego? ¿Qué métricas utilizas para cuantificar su desempeño? El modelo de Aprendizaje Reforzado tuvo un mejor desempeño en el juego. Una métrica clave para cuantificar el desempeño es el número medio de pasos por juego. En general, el agente de Aprendizaje Reforzado requirió menos pasos para completar un juego en comparación con el agente de Red Neuronal, lo que indica una mayor eficiencia.

- ¿Qué sucede si el tamaño del mapa cambia? ¿Cuál de los dos modelos maneja mejor este cambio y por qué? El modelo de Aprendizaje Reforzado puede ser más adaptable a los cambios en el tamaño del mapa. Esto se debe a que el Aprendizaje Reforzado puede aprender y ajustar su comportamiento en función de las recompensas y penalizaciones que recibe, lo que podría permitirle adaptarse mejor a los cambios en el entorno.

- Menciona dos diferencias observables en el comportamiento de cada agente, por comportamiento nos referimos a secuencias de movimientos o situaciones que frecuenta el agente. Luego de haber realizado varios procesos de entrenamiento pudimos notar que el agente de Aprendizaje Reforzado tomó decisiones más eficientes que llevaron a juegos más cortos, mientras que el agente de Red Neuronal realizó más exploración del entorno, lo que resultó en juegos más largos.

- Por último, reflexiona y explica cuál de los dos modelos recomendarías para desarrollar agentes en juegos de caza más complejos y por qué. Basándome en los resultados obtenidos de los entrenamientos, recomendaría el modelo de Aprendizaje Reforzado para desarrollar agentes en juegos de caza más complejos. Este modelo demostró ser más eficiente en términos de tiempo de entrenamiento y desempeño en el juego. Además, el Aprendizaje Reforzado tiene la ventaja de poder adaptarse a entornos cambiantes, lo que podría ser útil en juegos más complejos donde las condiciones pueden variar. Sin embargo, es importante tener en cuenta que la elección del modelo también puede depender de otros factores, como la estructura del juego y las capacidades de computación disponibles.