Learning to Predict Without Looking Ahead: World Models Without Forward Prediction

C. Daniel Freeman, Luke Metz, David Ha

Alumno	Varela, Agustina Belén
Padrón	99238
Materia	Simulación
Código de materia	75.26 - 95.19

Introducción

El artículo "Learning to Predict Without Looking Ahead: World Models Without Forward Prediction" pone un enfoque innovador dentro del campo del deep learning y la inteligencia artificial. Este trabajo da una nueva perspectiva en la cual los agentes operan en entornos complejos, donde las observaciones no son siempre completas o constantes.

Uno de los conceptos claves es el "observational dropout", encargado de limitar la frecuencia en la que el agente puede observar el entorno, obligándolo de esta forma a desarrollar nuevos modelos internos capaces de estimar y completar la información faltante. Diferenciándose de esta forma con los enfoques tradicionales que dependen de modelos predictivos detallados.

Tradicionalmente, se utilizaron múltiples enfoques para el aprendizaje de modelos y el control de la inteligencia artificial, donde bastaba con que el modelo captara las características esenciales del entorno, desde técnicas simples hasta métodos más avanzados como con redes neuronales. Sin embargo, lo interesante en este proyecto es poner una perspectiva novedosa, acerca de que ya no es necesario que los modelos predictivos sean perfectos, o que reciban toda la información del contexto para que el agente aprenda a realizar su tarea eficientemente. Para demostrarlo, se llevaron a cabo gran cantidad de pruebas en distintos escenarios como el problema del cart-pole, el grid world e incluso carreras de autos.

Trabajo relacionado

En el aprendizaje de modelos y control en inteligencia artificial, hubo un interés muy grande en utilizar modelos del mundo (world model) para acelerar el aprendizaje. ¿Pero qué son los modelos del mundo y cómo funcionan? Son representaciones internas que un agente crea para simular y predecir el comportamiento del entorno en el que opera. Estos modelos permiten al agente tomar decisiones basadas en una versión imaginada del mundo real, lo que les ayuda a planificar y anticiparse a las consecuencias de sus acciones sin tener que interactuar directamente con el entorno en todo momento.

Por lo tanto, los enfoques tradicionales se basan en ajustar modelos con datos obtenidos del entorno real. Los primeros trabajos en esta área emplearon parametrizaciones de modelos simples, como parámetros ajustables para la identificación de sistemas. Con el tiempo, avanzó hacia parametrizaciones más flexibles como aproximadores de funciones, incluso el uso de redes neuronales.

A pesar de los avances en el modelado predictivo, estos métodos se suelen aplicar principalmente a dominios relativamente simples ya que los errores en el modelo del mundo pueden acumularse y causar problemas.

Este nuevo enfoque propone un concepto conocido como "aprendizaje de representaciones" en el que el modelo del mundo emerge como resultado de la resolución de una tarea específica. El impulso para mejorar el modelo se basa en la optimización de caja negra. Proponiendo algoritmos evolutivos y robustos, y se adaptan a restricciones mientras resuelven la tarea dada.

Cuando un modelo del mundo es suficientemente bueno

El método “observational dropout” consiste en dejar de lado la restricción a un agente de aprender políticas sin necesidad de interactuar directamente con el mundo real, y permitirnos que el agente cambie periódicamente entre observaciones reales y observaciones simuladas generadas por un modelo mundial.

Al tratar este problema como un MDP (Proceso de Decisión de Markov) que se utiliza para modelar decisiones en situaciones donde los resultados son

parcialmente aleatorios y dependen de las decisiones del agente, el agente puede utilizar su modelo del mundo para generar observaciones basadas en estados anteriores. Sin embargo, cuando el agente recibe datos reales, el modelo interno se resetea, sincronizándose nuevamente con la realidad. Haciendo de esta forma una mezcla entre la “observación real” y la creada por él.

Este enfoque sugiere que, aunque el modelo del mundo aprendido no sea completamente preciso en sus predicciones o en la estimación de parámetros del entorno real, aún puede ser útil para desarrollar políticas que funcionen bien. En esencia, el modelo del mundo no necesita ser perfectamente predictivo para ayudar a encontrar una política efectiva, destacando que el sesgo en la estructura del modelo influye significativamente en la facilidad de optimización de la política final.

Aunque la simplicidad de este enfoque puede no ser aplicable a entornos más complejos del mundo real, la lección clave es que un modelo del mundo puede ser inexacto, siempre y cuando sus errores no perjudiquen la efectividad de la política que se desea optimizar.

Proceso de Decisión de Markov

Como se mencionó anteriormente, el método de “observational dropout”, dentro de los modelos del mundo en el aprendizaje por refuerzo, se propone una estrategia en la que un agente alterne entre observaciones reales y simuladas para aprender una política más efectiva, el proceso es descripto utilizando un Proceso de Decisión de Markov (MDP), por lo que entraremos más en detalle del mismo.

En este caso, el MDP se expande a un espacio de estados aumentado que incluye tanto estados reales como simulados por el modelo del mundo. El agente, con cierta probabilidad, decide entre actualizarse con transiciones del mundo real o con transiciones generadas por su modelo interno del mundo. Esta decisión se controla por una variable aleatoria denominada probabilidad de "peek".

Cuando el agente observa el mundo real, su modelo interno se resetea para sincronizarse con la realidad. La función de recompensa del MDP modificado permanece igual a la del MDP original, ya que el objetivo sigue siendo optimizar el desempeño en el entorno real.

La restricción en las observaciones hace que la optimización sea más difícil, lo cual puede llevar a un peor desempeño en la tarea subyacente. Sin embargo, este enfoque puede impulsar el aprendizaje del modelo del mundo de manera similar a como la evolución impulsa los modelos internos en otras configuraciones.

El MDP se puede representar de la siguiente forma:

- Variables de estados y transiciones:

  - s ∈ S: Representa los estados del sistema o entorno en el que el agente opera, es decir del sistema original. S es el conjunto de todos los posibles estados.
  - $s^{t+1} \sim P(s^t, a^t)$: El estado se distribuye de acuerdo con una función de transición p, que depende del estado actual (s^t) y de la acción (a^t) que el agente elige en ese estado, representando al del modelo.

- Recompensa:

  - $R(s^t, a^t, s^{t+1})$: Esta función asigna una recompensa al agente en función del estado actual (s^t) y de la acción (a^t) y el siguiente estado s^{t+1}.

La transición se alterna entre los estados reales y los del modelo del mundo con una probabilidad p de la siguiente manera:

$$P'(a^t, (s')^t) = \begin{cases} (s_{orig}^{t+1}, s_{orig}^{t+1}), & \text{if } p < r \\ (s_{orig}^{t+1}, s_{model}^{t+1}), & \text{if } p \geq r \end{cases}$$

El objetivo es entrenar tanto al agente ($\pi(s, \theta)$) como al modelo del mundo ($M(s, a^t, \theta)$) para maximizar la recompensa en este MDP aumentado. La técnica de optimización empleada utilizando redes neuronales para parametrizar el modelo del mundo y la política es “Reinforce” basada en población debido a su simplicidad y efectividad en la obtención de buenos resultados en diversas tareas.

También se destaca que el modelo del mundo desarrollado puede no ser directamente interpretable, y que la política aprendida podría utilizar características complejas extraídas del modelo. Aunque este método desafía las convenciones típicas sobre la necesidad de observaciones continuas y completas

del mundo real, su eficacia está limitada por la capacidad del agente para procesar y utilizar de manera efectiva las observaciones simuladas.

Políticas emergentes de modelos del mundo entrenados con “observational dropout”

Previamente se mencionó el experimento de balanceo de cart-pole, ahora se va a desarrollar un nuevo experimento, balanceo de cart-pole swing up, donde en este caso no sólo hay que mover el carro de izquierda a derecha en una pista mientras se mantiene un péndulo en la posición vertical sobre el carro, sino que en este caso el péndulo comienza colgando hacia abajo, y el objetivo es que el agente mueva el carro de tal manera que el péndulo se balancee hacia arriba y se mantenga equilibrado en una posición vertical.

El experimento del balanceo del cart-pole, puede ser resuelto trivialmente con una política lineal simple. Sin embargo, en cart-pole swing up como es una tarea más compleja, al aplicar “observational dropout” una política lineal no es suficiente, por lo que se requiere que el agente aprenda dos cosas fundamentales:

1. **Incorporar energía al sistema** para elevar el péndulo.
2. **Disminuir energía** para equilibrar el péndulo una vez que se acerca a la posición de equilibrio inestable en posición vertical.

La dinámica del sistema se modela por un conjunto de ecuaciones no lineales que describen la aceleración del carro y el ángulo del péndulo:

$$[\ddot{x}, \ddot{\theta}] = F(x, \theta, \dot{x}, \dot{\theta})$$

El agente en este experimento recibe observaciones incompletas del entorno, por lo que debe aprender un modelo del mundo para llenar esos vacíos de la observación. Y de esta forma, se pudo comprobar que surgen modelos del mundo que permiten que la política funcione bien en el entorno de cart-pole swing up.

Para entrenar al agente se lo hizo con diferentes probabilidades de peek (p), frecuencia con la que se le permite al mismo observar el entorno real. Dando los siguientes resultados:

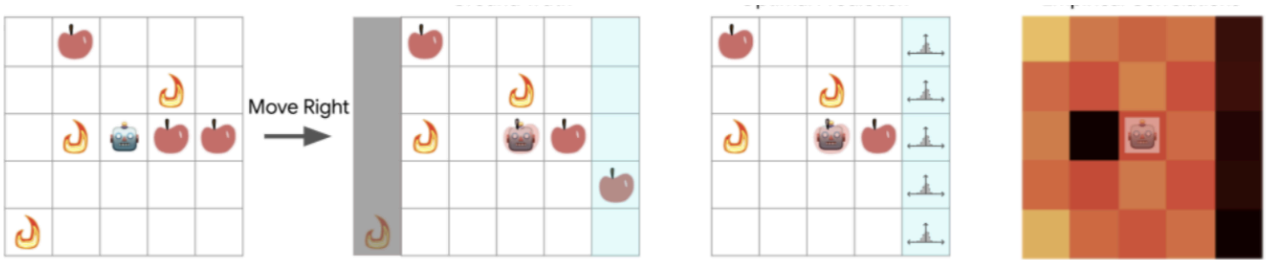
- El agente es capaz de levantar y equilibrar el cart-pole en la mayoría de los casos, con puntaje acumulado 500.
- El agente es capaz de resolver la tarea cuando observa solo un décimo de los marcos, p=10.
- Incluso con una probabilidad de peek mas baja, p=5, el agente resuelve la tarea la mitad de las veces.

Con todo esto se pudo observar que incluso con baja probabilidad de peek, un porcentaje significativo de los modelos del mundo aprendidos pueden ser utilizados para entrenar políticas que se comporten correctamente bajo una dinámica real, aunque no resuelvan completamente la tarea.

En comparación con un enfoque basado en modelos explícitos, donde se entrena un modelo para predecir la siguiente observación usando un conjunto de datos recogidos de un agente que se entrena sin modelo. En este caso con el enfoque basado en “observational dropout” mostró superar al modelo aprendido explícitamente en tareas de transferencia del cart-pole, pero desaparece al escalar el tamaño del modelo predictivo. En el entorno generado por el modelo del mundo, aprende a levantar la polea, pero sólo la mantiene en equilibrio durante un período de tiempo corto. Esto refleja que el modelo del mundo no necesita predecir con exactitud las observaciones para mantener el equilibrio, sino que se enfoca principalmente en aprender las partes más difíciles de la tarea, como levantar la poléa.

Sesgos inductivos de los modelos del mundo, en un entorno “grid world”

Este estudio examina cómo los sesgos inductivos de los modelos del mundo influyen en su rendimiento en un entorno de "grid world". En este caso, el entorno es un problema donde hay una cuadrícula que contiene manzanas y fuegos, las primeras otorgan recompensas mientras que los segundos penalizaciones, por lo que debe saber buscar y evitar a la hora de moverse, ya que puede hacerlo en cuatro direcciones o mantenerse en su lugar.



1

Se comparan dos arquitecturas de modelos del mundo:

- Completamente conectada, con mayor capacidad de modelado, pero genera menos sesgo. Si bien se puede aprender a predecir de manera óptima si se entrena con un objetivo supervisado, su rendimiento es inferior en este entorno específico.
- Convolucional con menos capacidad pero más sesgos, capturando patrones de tarea de buscar y evitar de forma más eficiente, superando al modelo completamente conectado.

En conclusión, a pesar de tener mayor capacidad, el modelo completamente conectado es superado por el modelo convolucional en la tarea de búsqueda y prevención. El modelo convolucional aprende transiciones más fiables y predictivas, especialmente en las direcciones claves (abajo y a la derecha), lo cual es crucial para el desempeño del agente.

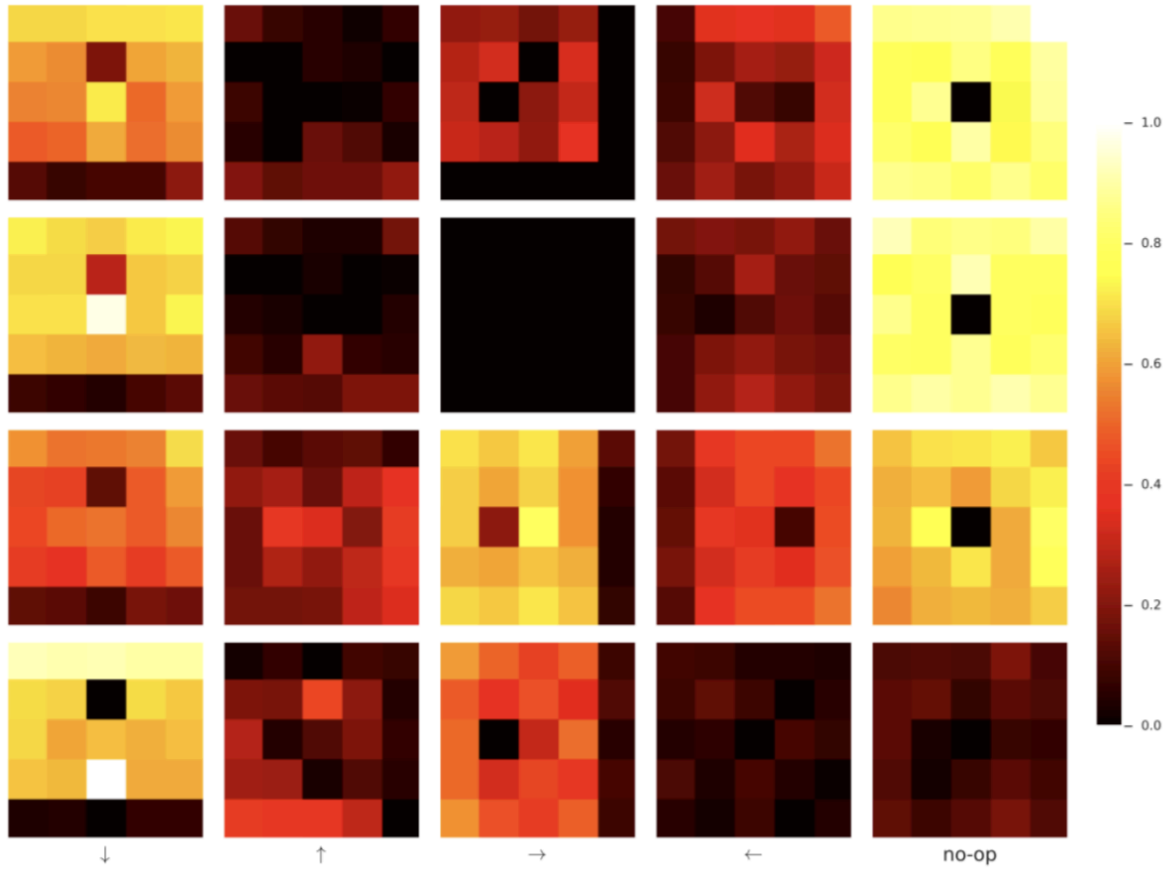
Se observó que los modelos convolucionales, especialmente aquellos entrenados con una probabilidad de observación real (peek probability) cercana al 50%, les permite aprender transiciones más precisas y útiles para predecir correctamente el próximo estado del entorno. Mientras que los modelos completamente conectados mostraron mapas de transición menos interpretables, aunque las políticas entrenadas con ellos alcanzaron un rendimiento cercano al de los modelos convolucionales.

En base a todo eso se llega a que en tareas de búsqueda en entornos parcialmente observables, el sesgo inductivo (es decir, las suposiciones inherentes en la arquitectura del modelo) es clave para que el modelo del mundo sea efectivo. Mientras que en los modelos convolucionales, con el sesgo adecuado, superan a los modelos completamente conectados en la tarea de predecir correctamente las transiciones del entorno.

---

[1] La imagen ejemplifica un caso de la grilla en la que el agente se encuentra en el centro de la misma y hay manzanas y fuegos a su alrededor. Cómo comienza a moverse y cómo esos movimientos se ven en píxeles representando qué tan certeros son los mismos.

Para dar un caso más claro, podemos observar la siguiente imagen en la que se representan matrices de correlación para varias arquitecturas convolucionales, donde se analiza cómo el modelo predice las transiciones. Los colores en las matrices indican la correlación entre las predicciones del modelo y la realidad, donde el amarillo representa una alta correlación (predicciones precisas) y el rojo oscuro o negro indica baja correlación (predicciones menos precisas o incorrectas).



[2]

En este caso se puede observar principalmente un pixel oscuro adyacente al agente en muchas de las direcciones, indicando cómo el modelo falla en predecir una manzana desplazándose hacia esa dirección. Este error es consecuencia de que el modelo convolucional trata las posiciones de manera uniforme, sin distinguir entre diferentes ubicaciones específicas, aplicando el mismo filtro a toda la cuadrícula sin adaptarse a los cambios y ubicación del agente. Por ello es muy importante ajustar la arquitectura del modelo del mundo teniendo en cuenta las transiciones del entorno, fundamental para las tareas de buscar y evitar.

[2] Imagen que representa distintos casos de movimientos y su evolución en el modelo en píxeles.

Carrera de autos

En este experimento se busca cómo un modelo del mundo puede ayudar a un agente a conducir en un juego de carreras sin observar en su totalidad el mundo real. El objetivo es evaluar hasta qué punto el modelo del mundo puede compensar la falta de observación directa y qué tan útiles son las representaciones internas que aprende para facilitar su forma de conducir.

En este caso, el agente debe conducir en pistas generadas aleatoriamente, recorriendo el mayor número de baldosas en el menor tiempo posible. Para ello se tienen observaciones en forma de imágenes en alta resolución y acciones del agente que controlan la dirección, acelerador y freno del auto. Se utiliza Autoencoder Variacional (VAE) para comprimir las observaciones en vectores de menor dimensión, que luego el agente usa para tomar decisiones.

Se entrena un modelo del mundo que predice cambios en el vector del VAE, basados en la observación previa y la acción que se realizó. Permitiendo al agente generar predicciones sobre el entorno, incluso cuando no tiene acceso directo a las observaciones. Este modelo es evaluado en su capacidad para predecir movimientos locales del coche y anticipar curvas en la pista, usando como referencia las observaciones reales.

Se comienza a disminuir la probabilidad de observación directa, el agente sigue siendo capaz de continuar con la tarea sin inconvenientes, confirmando que el modelo del mundo funciona correctamente para llenar los vacíos de observación. Se debe tener en cuenta que en aquellos casos donde la observación es demasiado baja, la efectividad disminuye ya que no es capaz de rellenar completamente la falta de información.

Para probar si las características aprendidas son realmente útiles, se entrena una política lineal que sólo utiliza las salidas de capa oculta del modelo, donde los resultados muestran que con una probabilidad de observación menor tienen más probabilidad de aprender representaciones útiles.

Es por eso que se demuestra como un modelo del mundo en este caso puede aprender de representaciones internas permitiendo al agente realizar tareas más

complejas como conducir un auto en una pista de carreras, incluso con un acceso muy limitado a las observaciones.

Conclusión:

Después de todo este análisis se pueden encontrar distintos puntos claves para comprender mejor cómo los modelos del mundo funcionan cuando se entrenan con observational dropout en tareas de aprendizaje por refuerzo.

La eficiencia de los modelos del mundo pueden ser efectivos cuando se optimizan para maximizar la recompensa total. Si bien en entornos simples no representan el mundo de manera perfecta, facilitan mucho el aprendizaje como para resolver ciertas tareas.

Se analizaron los distintos casos:

- Caso Cart-Pole: los modelos del mundo aprendidos logran que el péndulo se mantenga derecho, pero no se mantiene de forma perfecta el equilibrio, es decir maneja muy bien las tareas más 'complicadas'.
- Caso Grid World: los modelos del mundo realizan mapas de cambio de bits de manera confiable pero en algunas direcciones.
- Caso Cart Racing: el modelo del mundo es muy útil incluso con una observación parcial, aunque no capture completamente todos los movimientos, la capacidad para predecir cambios de dirección y anticipar curvas fue muy efectiva. Aún siendo imperfecto, puede facilitar el aprendizaje de forma efectiva.

Es decir, a pesar de estos detalles, ningún caso fue lo suficientemente grave como para perjudicar el rendimiento del agente, sino en algunos casos estos puntos de error resultaron irrelevantes para el desempeño. Modificando la complejidad de los modelos del mundo se podría maximizar su eficacia y aprovechar plenamente su potencial.

Análisis práctico

Para continuar con el análisis del tema se llevaron a cabo varias ejemplificaciones de los distintos casos y modelos en el notebook.

En un principio se decidió hacer un modelo de mundo genérico que pueda funcionar para los distintos casos que se querían probar como cart-pole, Grid World, y Cart Racing. Por lo que se creó la función “create_world_model
“ y “train_world_model” destinada para su entrenamiento.
Se creó junto a ellos “data“ encargada de hacer la recolección de datos del entorno para que aprenda y poder interactuar con el entorno.
Luego, se realizó también la política para que tome los distintos estados y pueda producir una acción con las funciones “create_policy_model” y “train_policy“ para su entrenamiento.

Una vez realizadas todas las funciones genéricas que nos van a funcionar para los distintos casos, se fueron realizando las pruebas esperadas para ellos:

- **Caso Cart-Pole**

El entrenamiento se está trabajando con 10 épocas con 32 lotes de datos en cada época. En este caso lo que buscamos comparar es el resultado de "loss" ya que la pérdida, se encarga de medir qué tan bien el modelo está prediciendo el siguiente estado. Un valor más bajo indica una mejor precisión.

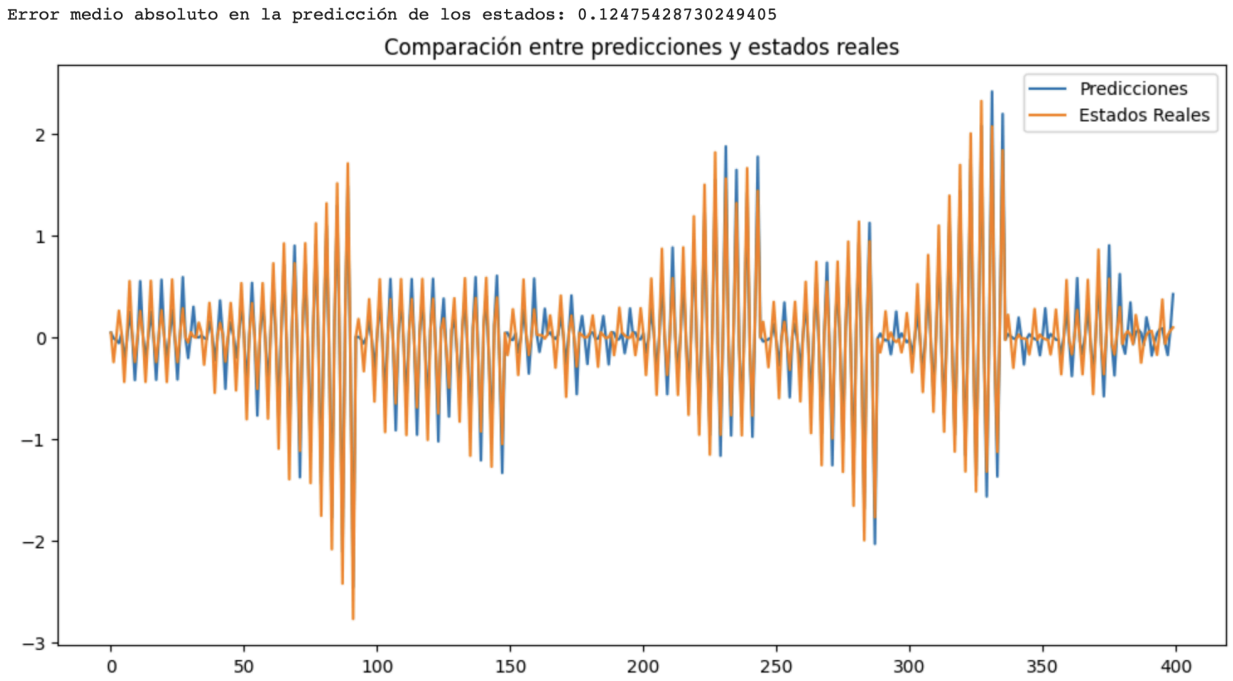
Como se puede observar a lo largo de las distintas épocas, la pérdida disminuye con cada época. Indicando que el modelo está mejorando su capacidad para predecir el siguiente estado. La pérdida en la última época recibida es 0.0310. Esto indica que, después de 10 épocas, el modelo logró una pérdida relativamente baja, lo que es una buena señal de que el modelo aprendió a predecir el siguiente estado con cierta precisión.

Por lo tanto, en un problema como el Cart-Pole, el objetivo es que el modelo del mundo aprenda a predecir cómo cambia el estado del entorno en función de las acciones tomadas. Lo que se espera es que haya reducción en la pérdida a lo largo de las épocas.

Estos resultados son consistentes con los enfoques de modelos del mundo mencionados en la monografía. En particular, los modelos del mundo para el cart-pole suelen mostrar una disminución de la pérdida durante el

entrenamiento, y este comportamiento es el esperado y por lo tanto coincide con el recibido anteriormente.

Se decide realizar un gráfico para interpretar mejor los datos:

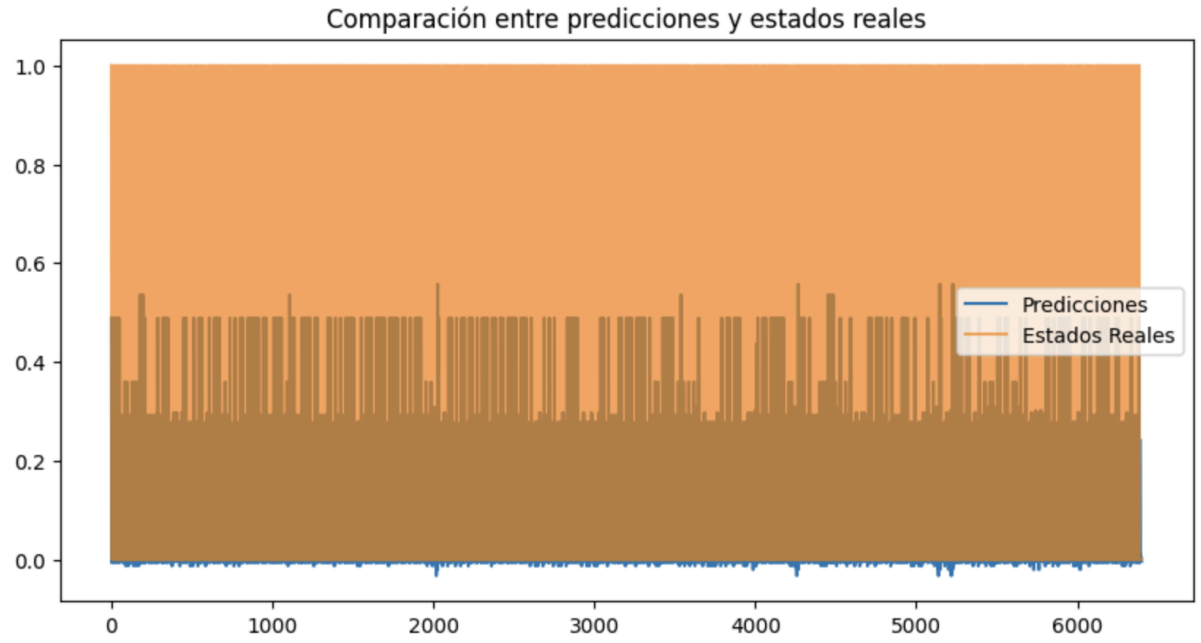


En el mismo se pueden observar varios puntos clave:

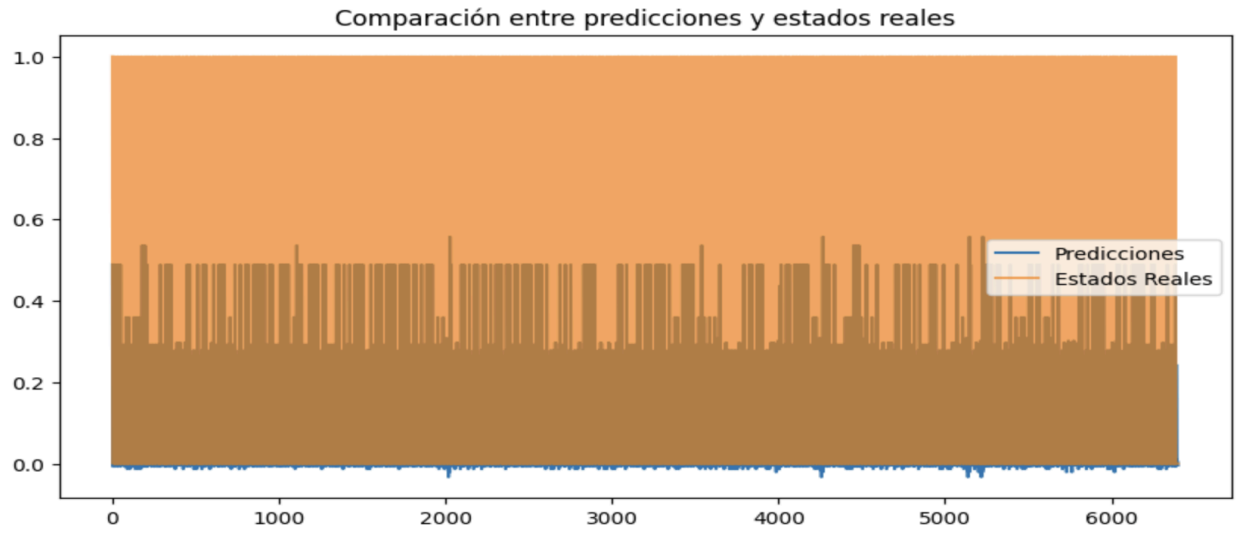
- El error medio absoluto de 0.12 en este caso. Es decir, las predicciones del modelo del mundo difieren de los estados reales en 0.12 unidades.
- En el gráfico, las líneas de color azul representan las predicciones del modelo, mientras que las naranjas representan los estados reales.
- En algunos puntos, las predicciones del modelo coinciden con los estados reales. Si bien en el resto no coinciden tanto, se sigue manteniendo la estructura y el comportamiento general del entorno.
- Estos resultados sugieren que el modelo del mundo capta la dinámica general del entorno, pero tiene dificultades para predecir con precisión en situaciones de mayor variabilidad.
- Concluimos en que tal como lo estudiado, los modelos del mundo tienden a tener un rendimiento razonable, pero no perfecto.

- **Caso Grid World**

Este caso también se decidió implementarlo con 10 épocas y 32 pasos, se realizó el código para llevar a cabo sus pruebas y en los resultados descriptos se puede observar como la pérdida (loss) va disminuyendo a medida que pasan las épocas, representando que el modelo está aprendiendo y mejorando, yendo hacia una mejor solución. En estos casos de prueba se obtuvo un Error medio absoluto en la predicción de los estados de 0.09 y el resultado fue representado con la siguiente gráfica:



Para tener una comparación más precisa, se realizó nuevamente el experimento pero ahora con 50 épocas, donde se puede observar que hay una gran mejora entre la primera época hasta la número 13. A partir de este punto se estabiliza la pérdida, por lo que el modelo aprendió la mayoría de los patrones de los datos, y a partir de ese momento, las mejoras son muy pequeñas como si el modelo estuviera llegando a un punto de convergencia. Tal como se puede observar la gráfica recibida de este experimento es muy similar a la del paso anterior:



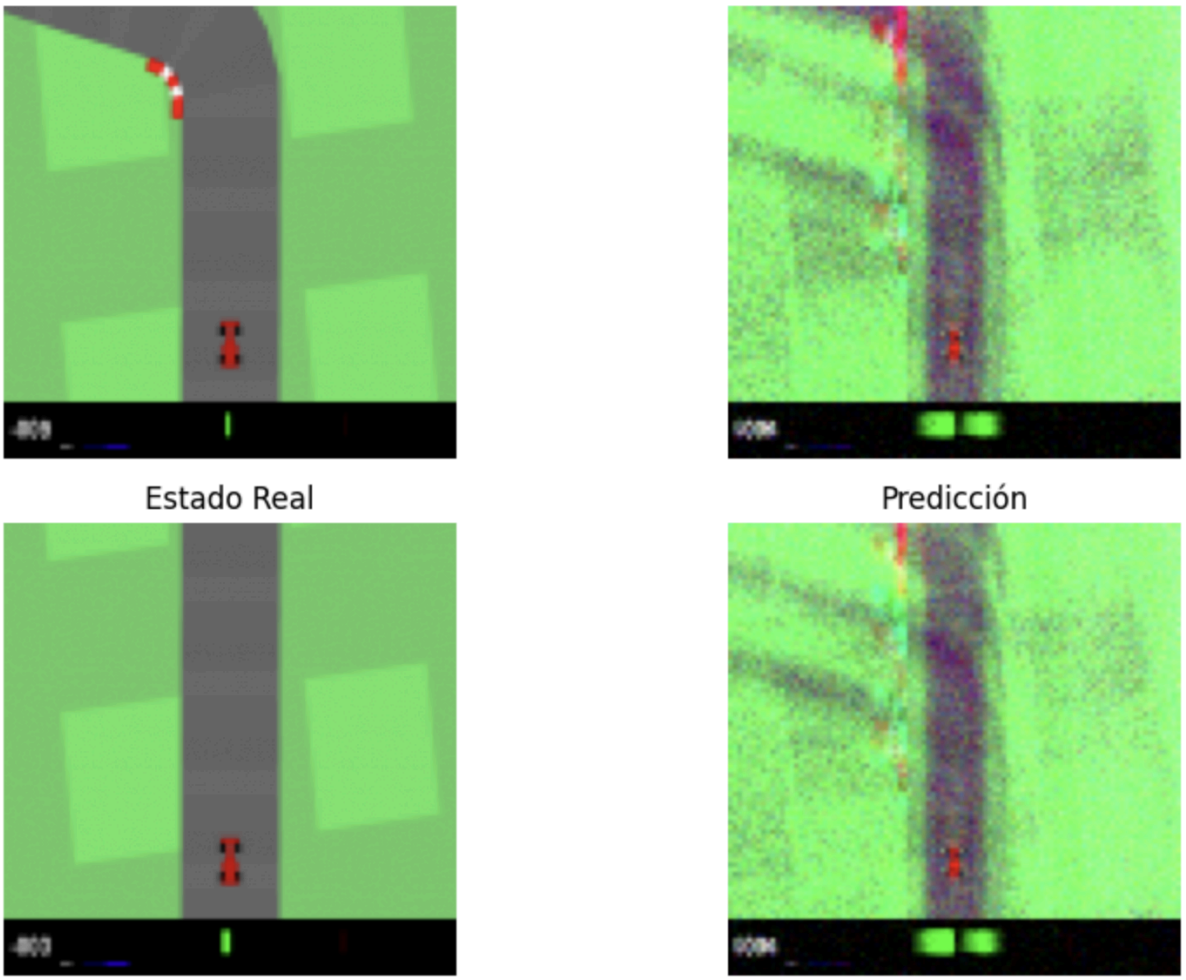
Por lo tanto, para este modelo, lo más eficiente es reducir el número de épocas a alrededor de 15, ya que después de este punto, no se observa una mejora significativa, lo que permite ahorrar tiempo y recursos sin afectar el rendimiento del modelo.
Nuevamente, los resultados obtenidos se alinean con lo esperado, ya que representa cómo el modelo del mundo puede aprender del entorno en un número bajo de épocas, alcanzando el punto de convergencia donde las mejoras son pequeñas. Representa a su vez que el modelo realmente está aprendiendo y reconociendo los estados y acciones eficientemente.

- **Caso Car Racing**

Finalmente para el caso de car racing, se decidió modificar el modelo inicial y la recolección de datos ya que se requería de más dimensiones para poder trabajar con imágenes. En este caso los resultados fueron con foco en los valores de loss, al disminuir la pérdida, se indica que el modelo está aprendiendo a predecir los estados con mayor precisión durante el entrenamiento. Para algunos casos hay ciertos picos, es decir que el modelo tiene algunas dificultades en algunos casos que se deben a la complejidad del entorno y a la variabilidad en los datos.

El MAE da aproximadamente 0.158, lo que quiere decir que la diferencia entre las predicciones del modelo y los estados reales ronda las 0.158 unidades.

Para interpretar mejor, se representó gráficamente los resultados obteniendo lo siguiente:



En estas imágenes si bien a simple vista muestran un contorno bastante similar entre el Estado real y la predicción, observandolas más en profundidad se encuentra por ejemplo que empiezan a fallar en situaciones más complejas como cuando se realizan curvas, donde se empieza a perder o distorsionar la imagen. Incluso en casos donde es recto se empieza a simular como si realmente hubiera una curva o si iniciara a realizarla. Se puede observar como los resultados si bien fueron mejorando a lo largo del entrenamiento, se puede continuar ajustando el modelo para llegar a una mejor precisión.