



Tarea 3: aprendizaje de modelos de conducción vehicular en 2D

Introducción

En esta tarea tendrá la oportunidad de experimentar con el uso de redes neuronales recurrentes para el procesamiento de datos secuenciales. En particular, deberá entrenar redes recurrentes para modelar y predecir el comportamiento de conductores de vehículos en múltiples escenarios reales. Para el desarrollo se utilizará el framework Pytorch sobre la plataforma [Colab](#) de Google. Dado que existe abundante material disponible en línea para colaborar con el desarrollo de la tarea, se espera que todo recurso externo utilizado, sea este código, librerías o papers, esté debidamente indicado.

Set de datos

La fuente primaria de datos para entrenar los modelos será el set de datos **Interaction**, descrito [acá](#). Este conjunto considera múltiples escenarios de conducción, donde para cada uno se registran múltiples variables a lo largo del tiempo para cada uno de los vehículos involucrado. Para obtener los datos, descargue el archivo disponible [acá](#), y descomprima el contenido utilizando la contraseña que será enviada por correo.

Cada escenario estudiado se encuentra descrito en una serie de archivos con extensión `.csv`, donde se encuentran tabuladas las variables. En particular, se registran posiciones, velocidades y ángulo de giro, entre otros. Para entender el formato y la dinámica de los escenarios, se recomienda utilizar la herramienta de visualización provista por los desarrolladores del conjunto de datos, disponible [acá](#).

Para crear sets de entrenamiento, validación y test independientes, considere trayectorias completas de vehículo y no partes de estas, es decir, de todos los vehículos participantes en los escenarios analizados, reserve una proporción de estos para los conjuntos indicados anteriormente.

Modelos

Para esta tarea, debe utilizar modelos de redes recurrentes como los descritos en el capítulo 3 del curso, sin limitante en relación al tipo de capas que se pueden utilizar (recurrente, convolucional, densa, *dropout*, *batchnorm*, etc). Se recomienda revisar bibliografía relacionada para esto. Considere además preprocesar las entradas de acuerdo a lo utilizado en la literatura (por ej., restar la media, normalizar en el rango $[0,1]$, etc.). No hay problema en basarse completamente en modelos propuestos previamente en la literatura o en tutoriales. En cualquier caso, debe justificar sus elecciones.

Actividades a realizar

Para la tarea se espera se espera que realice al menos las siguientes actividades:

- Entrenamiento simple: seleccione un escenario y entrene un modelo recurrente que permita predecir para al menos un *time step* en el futuro, la posición, velocidad y ángulo de rotación de un vehículo. Como variables de entrada puede definir a su gusto la ventana temporal y espacial que utilizará, es decir, la cantidad de intervalos de tiempo previos que utilizará (ventana fija, variable o sin ventana), y de qué otros vehículos utilizará información (ninguno, misma pista adelante, misma pista adelante y atrás, vecindad, etc). Reporte, además de la métrica de error principal en el set de test, series de tiempo para una trayectoria de prueba, donde se muestre la desviación con respecto a la trayectoria original para las distintas variables predichas.
- Transferencia: utilizando el modelo entrenado en el ítem anterior, seleccione un nuevo escenario y evalúe el rendimiento del modelo en él, usando una transferencia directa (modelo no se modifica, solo se evalúa) y fine-tuning. Compare los rendimientos con el de un modelo entrenado desde cero en este nuevo escenario.
- Comparación: Para la cantidad de escenarios que seleccione (1 o más), compare el rendimiento de un modelo recurrente con el de al menos una arquitectura distinta (densa o convolucional, por ejemplo). Analice las diferencias y discuta sobre el motivo de estas.
- Bonus: Muestre los resultados de los ítems anteriores utilizando una visualización gráfica. Puede basarse en la herramienta de visualización disponible con los datos.

Desarrollo y entrega

La tarea puede desarrollarse de manera individual o en parejas, utilizando el framework Pytorch para Python. Se recomienda utilizar la plataforma Google Colab con el fin de facilitar la instalación de librerías. Esta plataforma permite utilizar gratuitamente una GPU para el entrenamiento por intervalos de 12 horas continuos. En el *notebook* desarrollado debe ir tanto el código como un informe (preferiblemente intercalados), donde se expliquen los pasos realizados, se analicen los resultados y se planteen conclusiones. La entrega de la tarea tiene como fecha límite el jueves 01 de julio a las 23:59, a través del buzón que se habilitará en el sitio del curso. Para fines de corrección, se revisará la última versión entregada.

Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por “copia” o “plagio” se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.