

PRONÓSTICO DE NIVELES CON REDES NEURONALES

Santiago Guizzardi

Instituto Nacional del Agua (INA), Argentina,
santiagooguizzardi@gmail.com

Introducción

La desembocadura del río Paraná en el Río de la Plata ocurre a través de un amplio delta. La hidrodinámica en esta desembocadura es compleja. Está condicionada por las descargas del río Paraná y del río Uruguay y por las variaciones de niveles, consecuencia del efecto de las mareas astronómicas y meteorológicas que afectan el estuario del Río de la Plata (Re et al., 2003).

El pronóstico de series de niveles en las zonas costeras es un insumo esencial para las actividades de gestión y monitoreo del riesgo costero. Convencionalmente, el pronóstico se ha realizado utilizando modelos estadísticos basados en análisis de series de tiempo y modelos numéricos que, mediante una representación matemática, intentan resolver un problema físico (Dinápoli et al, 2020 y Guizzardi et al., 2022). En los últimos años, se han desarrollado con éxito técnicas de inteligencia artificial (IA) para modelar sistemas hidrológicos no lineales. En particular, las redes neuronales artificiales (Seo et al., 2015).

En este trabajo se evalúan distintas arquitecturas de redes neuronales para el pronóstico del nivel del agua en la localidad de San Fernando (Buenas Aires, Argentina), ubicada sobre el río Luján, a menos de 5 km. de la desembocadura en el Río de la Plata (Figura 1). Este punto resulta de interés ya que es un punto de referencia para la población local en el Delta y cuenta con un amplio registro de niveles hidrométricos, los cuales se encuentran influenciados por procesos fluviales y las mareas.

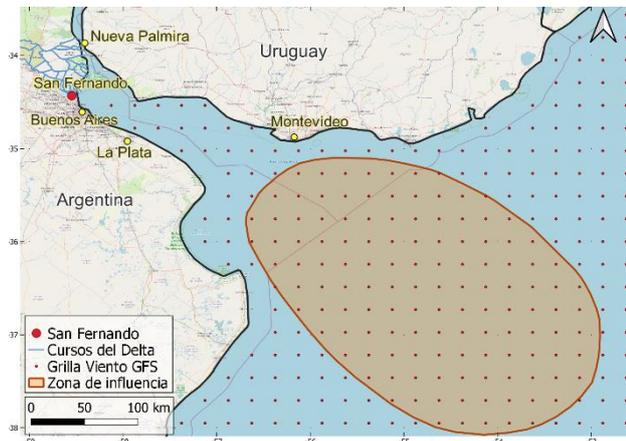


Figura 1.- Área de estudio.

Métodos y Materiales

Estructura de los datos

Los datos que se utilizan para la modelación son: serie de niveles hidrométricos observados en la localidad de San Fernando (Figura 2 a); serie de niveles mareas astronómicas predichas en el mismo punto, proporcionadas por el Servicio de Hidrografía Naval (SHN) (Figura 2 a); campos de viento a 10m de altura, pronosticados por el modelo numérico de predicción meteorológica: Global Ensemble Forecast System (GEFS), proporcionado por el Centro Nacional de Predicción Ambiental (NCEP) de EE.UU. Este modelo se actualiza cuatro veces al día

con predicciones que alcanzan los 16 días. Los datos son de formato grillado con una resolución asociada a un pixel de $0.25^{\circ} \times 0.25^{\circ}$ y una resolución temporal de 3 horas. Las grillas contienen las componentes XY de la velocidad.

La ventana temporal de análisis se extiende desde noviembre del 2020 hasta abril del 2022. Momento desde el que se cuenta con datos de pronósticos de vientos almacenados.

Preprocesamiento de los datos

Para generar una serie temporal a partir de los datos en formato grillado, se realiza un promedio por paso temporal de las celdas contenidas en la “zona de influencia” para los niveles en el río de la Plata (Figura 1). Para este trabajo solo se utilizaron los pronósticos a un día, horizonte temporal en el cual se evalúan los modelos. Es decir, para cada día solo se toma el primer pronóstico brindando. Luego se interpola linealmente para completar a paso horario.

Por último, se agregan dos variables: acumulado de viento en X y en Y (Figura 2 b), que surgen de la suma de las velocidades de viento durante las últimas 24 horas para cada una de las componentes. Con esta variable se busca capturar la permanencia de los vientos. Esto se debe a que el Río de la Plata se ve afectado por ondas de tormenta que generan eventos de crecida y bajante en las costas. Las crecidas se generan cuando soplan vientos fuertes y persistentes del sureste. Por el contrario, las bajantes están asociadas a vientos persistentes que tienen una componente dominante del oeste al noroeste (D’Onofrio et al. 2008).

Se ve afectado

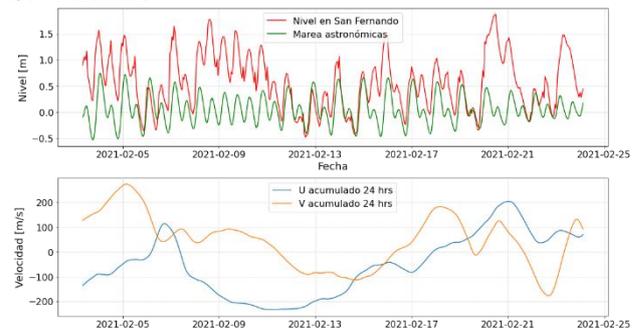


Figura 2.- Datos de entrada a los modelos.

Modelos

Se utiliza una división (70%, 20%, 10%) para los conjuntos de entrenamiento, validación y prueba. Los datos no se mezclan antes de dividirlos. De esta forma, es posible dividir los datos en ventanas de muestras consecutivas para entrenar los modelos. También, se garantiza que los resultados de la validación y prueba sean más realistas y se evalúan sobre los datos recopilados después de entrenar el modelo.

Antes de entrar en la red neuronal se normalizan los datos restando la media y dividiendo por la desviación estándar de los datos de entrenamiento.

Los modelos se desarrollan en lenguaje Python utilizando las librerías de código abierto: Tensorflow y Keras (Abadi, 2016). Esta última es un framework para trabajar con redes neuronales que permite una rápida implementación.

Los modelos se configuran para hacer predicciones de 24 horas

en el futuro basadas en una ventana de muestras consecutivas de las 24 horas del pasado.

Existen dos enfoques para esta tarea: predicciones de disparo único, donde se predice la serie temporal completa en una única salida (Lineal, Denso, CNN, RNN) y predicciones autorregresivas donde el modelo realiza predicciones de un solo paso y su salida se retroalimenta como entrada (AR RNN).

A continuación se describen los modelos utilizados:

Modelo de líneas base: se utiliza este modelo como punto de comparación para los modelos más complejos. Se toma como hipótesis que la serie no cambia considerablemente día a día, es decir, se repite el día anterior, asumiendo que el día siguiente será similar.

Lineal: un modelo lineal simple basado en el último paso de tiempo de entrada.

Denso: modelo que agrega una capa entre la entrada y la salida, dándole más potencia al modelo lineal. Al igual que el anterior, solo se basa solo en un paso de tiempo de entrada.

CNN: un modelo convolucional hace predicciones basadas en un historial de ancho fijo, lo que le permite ver cómo cambian las variables con el tiempo.

RNN: un modelo recurrente puede aprender a usar un largo historial de entradas, si resulta relevante para las predicciones que realiza. El modelo acumula el estado interno durante 24 horas, antes de hacer una predicción para las próximas 24 horas.

AR RNN: Modelo recurrente autorregresivo. Predice la secuencia de salida en pasos de tiempo individuales. La salida de cada modelo se retroalimenta en cada paso y se generan predicciones condicionadas a la anterior.

Resultados

Todos los modelos se entrenan sobre el mismo set de datos. Como función de pérdida, para determinar cuánto se desvían los valores predichos de los valores observados en los datos de entrenamiento y poder optimizar los parámetros, se adopta el error cuadrático medio. Para juzgar el rendimiento de los modelos se utiliza como métrica el error absoluto medio (MAE). En la Figura 3 se presenta el MAE obtenido de cada modelo sobre los datos de validación y prueba.

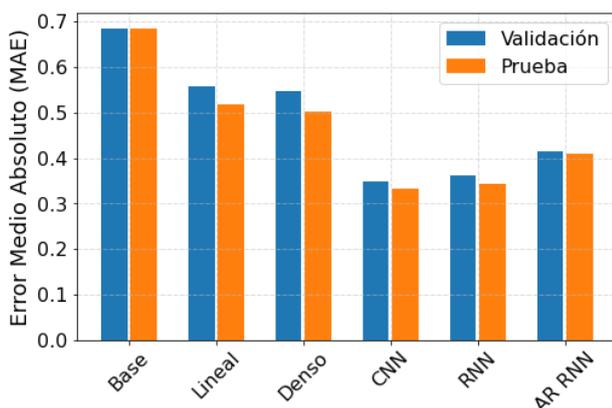


Figura 3- Error medio absoluto sobre datos de validación y prueba.

En la Figura 4 se pueden observar dos ejemplos de salidas para el modelo CNN que obtuvo el menor error medio absoluto entre los modelos utilizados.

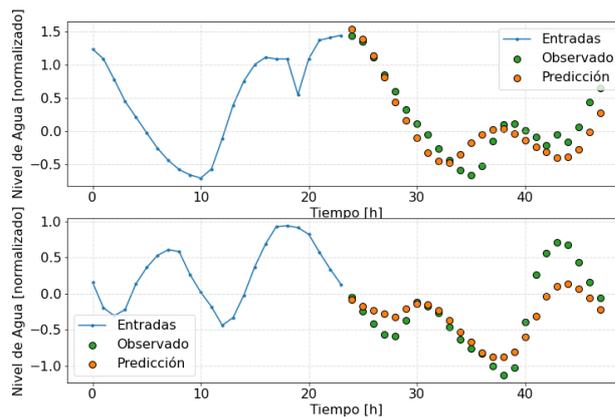


Figura 4.- Ejemplos de salidas del modelo CNN

Conclusión

Comparando los modelos utilizados se obtienen las siguientes conclusiones. El modelo lineal sólo puede capturar una porción de baja dimensión del comportamiento, probablemente basada principalmente en los datos de mareas astronómicas. El modelo Denso, al contar con una capa extra, tiene más potencia que el modelo lineal pero el rendimiento no cambia considerablemente. Esto se debe a que estos modelos se basan solo en un paso de tiempo como entrada al modelo.

Los modelos convolucional (CNN) y recurrente (RNN), tiene la capacidad de ver cómo las variables cambian en el tiempo. Esta capacidad permite un mejor rendimiento que los modelos anteriores, como se refleja en la Figura 3. Por último, el modelo autorregresivo, que en principio parece ser el más potente obtuvo un rendimiento menor que los modelos CNN y RNN. Entonces, estos enfoques más complejos pueden no valer la pena en este problema.

En los próximos pasos se busca profundizar sobre los modelos CNN y RNN buscando hiperparámetros más adecuados para este problema. También, se deben explorar otras posibles variables de entrada al modelo y alternativas para caracterizar los vientos.

Agradecimientos

Este trabajo forma parte del proyecto DELTA PARANA: Estudio Hidrodinámico Integrador del Delta del Río Paraná con Fines Múltiples (<https://www.ina.gov.ar/delta>), dentro del Instituto Nacional del Agua (INA).

Referencias

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). "Tensorflow: Large-scale machine learning on heterogeneous distributed systems". arXiv preprint arXiv:1603.04467.
- Dinápoli, M. G., Simionato, C. G., & Moreira, D. (2020). "Development and validation of a storm surge forecasting/hindcasting modelling system for the extensive Río de la Plata Estuary and its adjacent Continental Shelf". *Natural Hazards*, 103(2), 2231-2259.
- D'Onofrio, E., Oreiro, F., & Fiore, M. (2012). "Simplified empirical astronomical tide model—an application for the Río de la Plata estuary". *Computers & Geosciences*, 44, 196-202.
- Guizzardi, S., Bianchi, J., Cortese, J. E., Uriburu Quirno, M. & Sabarots Gerbec, M. (2022). "Forecast System Implementation in the Paraná Delta". In IAHR world congress 2022.
- Ré, M., & Menéndez, A. N. (2003). "Modelo Hidrodinámico del Río de la Plata y su Frente Marítimo". Informe INA-LHA, 03-21
- Seo, Y., Kim, S., Kisi, O., & Singh, V. P. (2015). "Daily water level forecasting using wavelet decomposition and artificial intelligence techniques". *Journal of Hydrology*, 520, 224-243.