

Reporte de pronósticos de series de tiempo para cuenta corriente

Sandra Thalía España Gómez

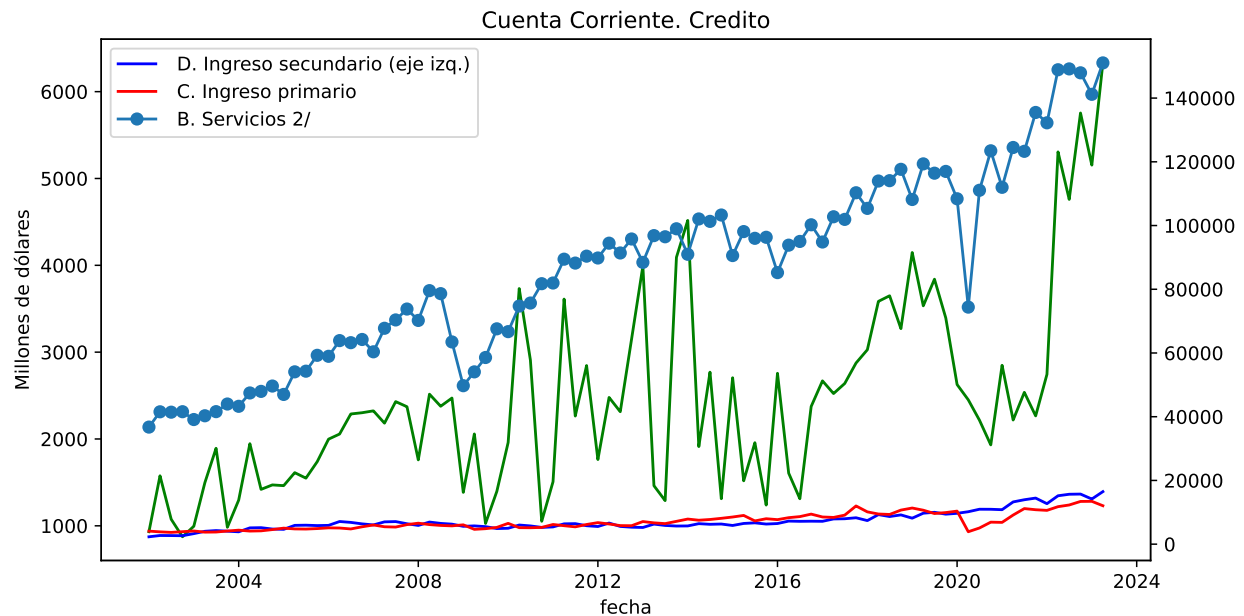
2023-10-13

Este reporte tiene como objeto describir el proceso de creación de modelos econométricos y de redes neuronales usados para pronosticar las series de cuenta corriente, IGAE, inversión fija bruta, tipo de cambio y remesas como regresores

Dado que las cifras de cuenta corriente se encuentran en constante revisión que pueden durar incluso varios años, es necesario guardar las cifras con las que se realizaron los pronósticos, ya que éstas pueden cambiar abruptamente.

Análisis exploratorio de datos (EDA en inglés)

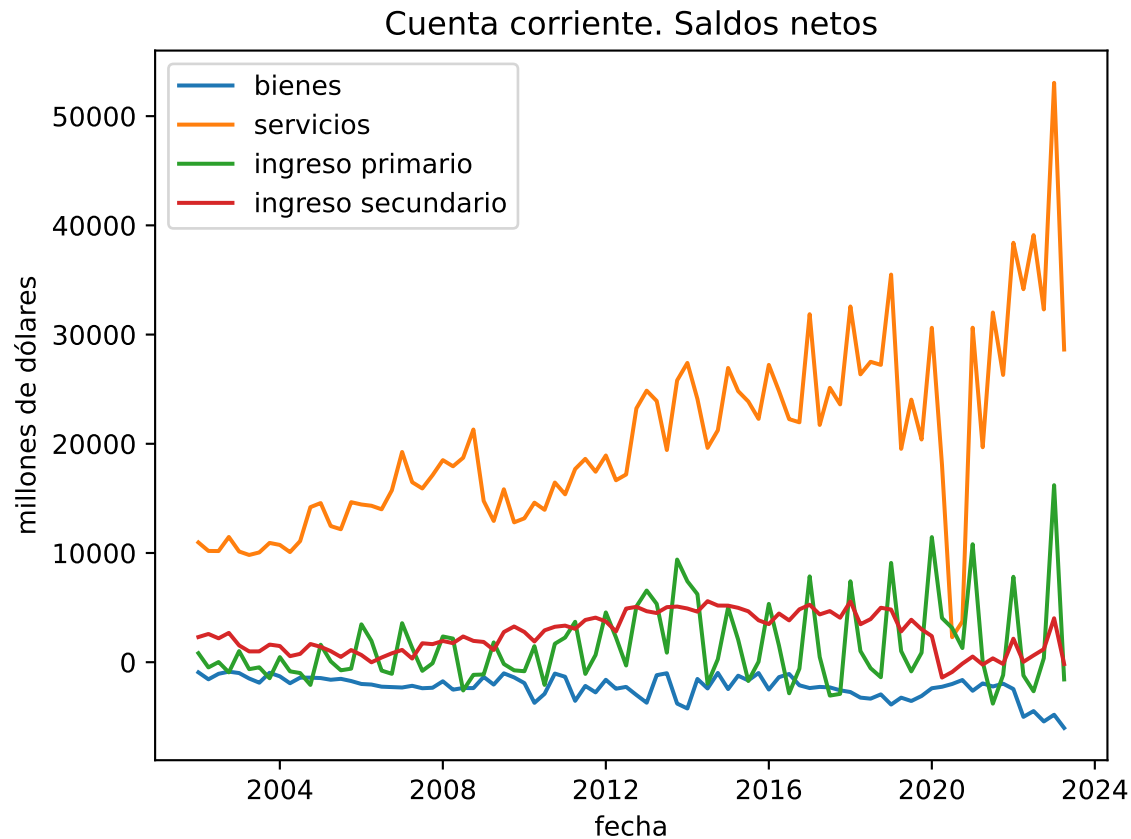
Se observa que la cuenta de crédito de la cuenta corriente más grande proviene de la cuenta de bienes, la cual comprenden exportaciones de mercancías y asciende a más de 175 mil millones de pesos. Por su parte, la segunda cuenta más grande (aunque considerablemente más chica que la cuenta de bienes) es la cuenta de ingresos secundarios, los cuales son casi su totalidad ingresos por remesas y ascienden a poco menos de 23 mil millones de pesos.



Por su parte, la cuenta de servicios primarios de crédito comprende utilidades, dividendos, intereses y remuneración de empleados, y ascendió a poco menos de 23 mil millones de pesos. Por otro lado, las de servicios suman transporte, viajes, servicios empresariales, propiedad intelectual entre los más importantes.

No obstante, en saldos netos (débito - crédito) la cuenta corriente de bienes, servicios e ingresos primarios se encuentra en saldos positivos, mientras que sólo la cuenta de ingresos secundarios se encuentra en saldos

negativos. Por otro lado, se observa un poco de estacionalidad en las cuentas de ingresos primarios e ingresos secundarios.



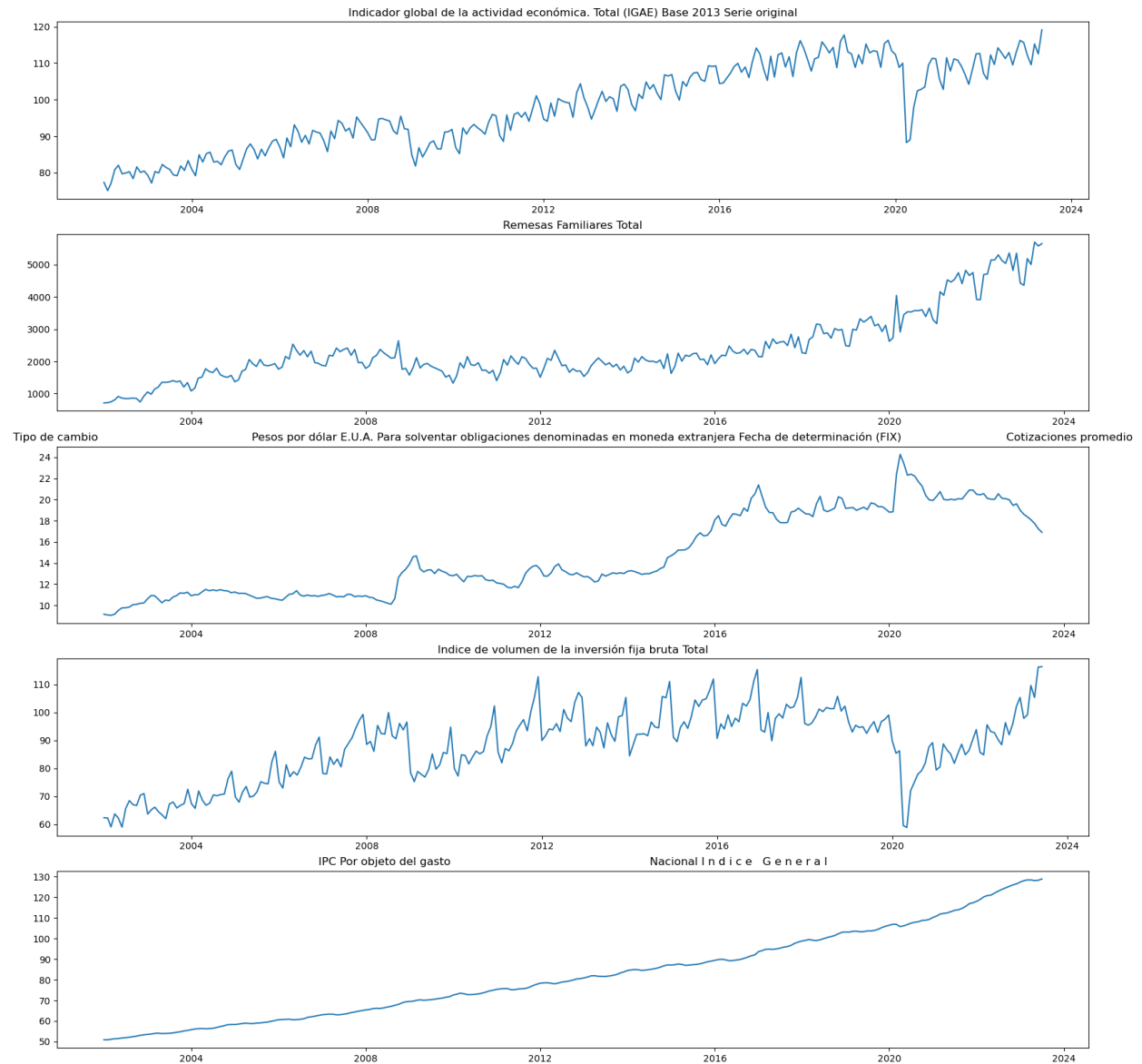
Selección de variables explicativas

Dado que en la cuenta de crédito los bienes fueron la subcuenta más grande con 178 mil millones de dólares, se revisó el análisis de exportaciones e importaciones del atlas de Harvard para México. De acuerdo con dicha base de datos, las principales *exportaciones netas* para México en 2021 fueron en autos y autopartes (~28%), computadoras (~14%), petróleo (~8.3%), otros minerales (~8.5%), así como alimentos, bebidas y agropecuarios (~8%), entre otros.

Por otro lado, las cuentas de ingresos secundarios también incluyen los ingresos por remesas, así como para la cuenta de servicios se incluyeron turismo, propiedad intelectual, que son los principales componentes.

Por todo lo anterior, se incluyeron en el modelo las siguientes variables:

- IGAE base 2013, serie original, mensual, publicada por INEGI.
- Ingresos por remesas en millones de dólares, mensual, publicada por Banco de México.
- Tipo de cambio peso-dólar promedio FIX mensual, publicada por banco de México.
- Índice de volumen de la inversión fija bruta, publicado por el INEGI.
- INPC general, publicado por el INEGI.



Modelos para pronosticar la cuenta corriente.

En este reporte, se utilizaron dos tipos de modelos de pronóstico de series de tiempo: los modelos ARIMA y los modelos de redes neuronales recurrentes (RNN en inglés) y de memoria de corto y largo plazo (LSTM en inglés). Si bien los modelos ARIMA tuvieron un mejor desempeño en la métrica seleccionada RMSE (*root mean squared error* en inglés, raíz de error cuadrático medio) en términos relativos, en términos absolutos las tres metodologías tuvieron un buen desempeño.

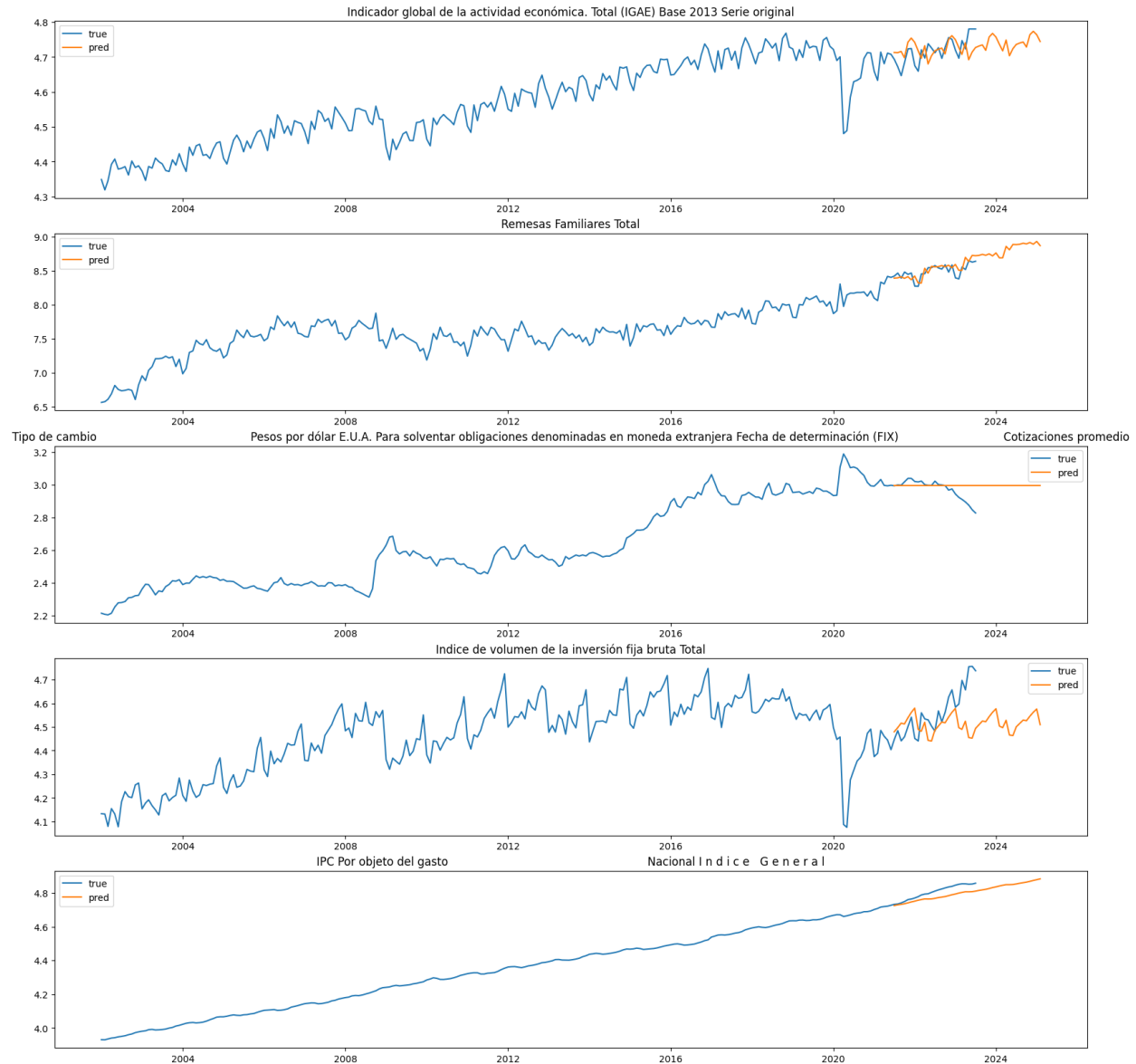
Resumen de RMSE por modelo

IGAE (ARIMA)	Remesas (ARIMA)	USDMXN (ARIMA)	Inversión FB (ARIMA)	INPC (ARIMA)	Cuenta corriente (ARIMA)	Cuenta corriente (RNN)	Cuenta corri- ente (LSTM)
0.029	0.066	0.045	0.100	0.034	0.924	1.209	1.215

Modelos ARIMA

Los modelos auto regresivos de media móvil son de los modelos a los que economistas e incluso científicos de datos suelen recurrir para pronosticar series de tiempo. Si bien existen otros modelos econométricos más sofisticados y con mayor sustento teórico para ciertas variables, los modelos ARIMA son modelos altamente compatibles con las nuevas propuestas metodológicas de aprendizaje profundo, particularmente de redes neuronales.

Dado que las redes neuronales piden pronósticos de los regresores para llevar a cabo el modelo, se parten de los pronósticos generados por los modelos ARIMA para tener una base explicativa para las redes neuronales.



Modelos de redes neuronales

Los modelos de redes neuronales son frecuentemente utilizados en aprendizaje de máquina para variables que tienen cierta relación temporal. En este trabajo, en ambos modelos (RNN y LSTM) se utilizaron capas de perceptrones que aplican una función no lineal a una combinación lineal de los resultados en la capa anterior (o bien, en el caso de la primera capa, de la matriz de insumos).

En ambos modelos, la separación entre entrenamiento y validación no fue aleatorio, al tratarse de una serie de tiempo. De haberlo hecho, erróneamente, el conjunto de entrenamiento hubiera estado contaminado con los datos que tenía como objetivo predecir. Dado que en una serie de tiempo el orden en el que se presentan los datos tiene un significado temporal, el periodo de validación debe consistir en datos que aún no han sido observados por el modelo.

Por otro lado, la matriz de las variables es una matriz relativamente pequeña, por lo que en general se eligieron hiperparámetros de magnitudes relativamente pequeñas.

Redes neuronales recurrentes (RNN en inglés)

En resumen, el modelo que arrojó un menor RMSE fue el siguiente:

Capas (tipo)	Dimensión de salida	Param #
simple_rnn (SimpleRNN)	(N, 32)	1216
dense (Dense)	(N,128)	4224
dense_1 (Dense)	(N,128)	16512
dense_2 (Dense)	(N,64)	8256
dense_3 (Dense)	(N,1)	65

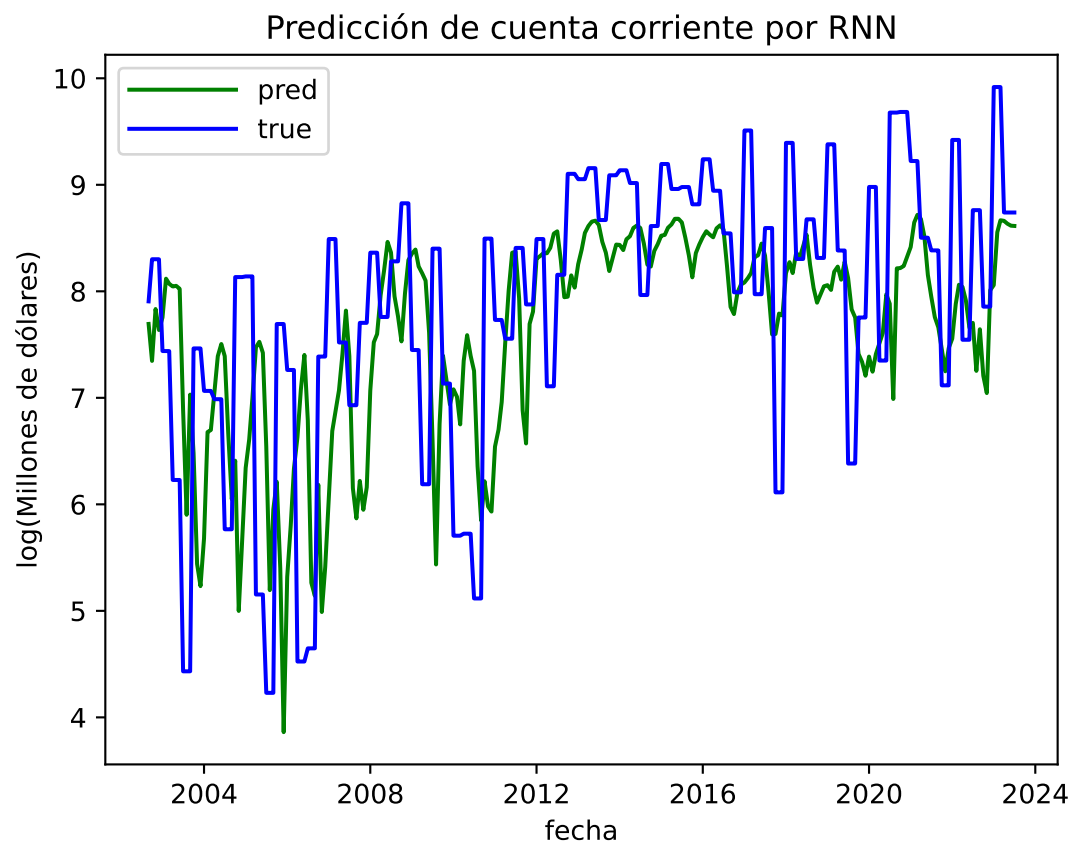
Núm. de parámetros (todos entrenables): 30273.

Destacan las siguientes justificaciones:

- 4 capas: dado que la matriz de series de tiempo es una matriz relativamente sencilla en comparación con otras aplicaciones en redes como detección de imágenes o video, los cuales suelen tener mayor dimensionalidad, se decidió usar un número bajo de capas. De inicio se iban a usar 3 capas, pero se observó que una cuarta capa mejoraba considerablemente el resultado.

-número de neuronas: se buscó que el número de neuronas fueran creciendo o decreciendo gradualmente (en potencia de 2) hasta llegar a la dimensión del insumo o del producto. La convención de varios académicos es utilizar 32, 64 o 128 neuronas en cada capa, dependiendo de la naturaleza de los datos.

- activaciones: sigmoide. Dado que se optó por usar un min-max scaler para que la magnitud de ciertas explicativas no sesgara al modelo, el rango de todas las variables quedó entre cero y uno, por lo que una transformación sigmoide hacía mucho sentido. No obstante, destacó que tanto en el modelo RNN como el LSTM se hizo presente el fenómeno *vanishing gradient* en la variable objetivo, ya que las colas izquierdas o derechas las subestimaba. Esto también pudo haber sido contribuido por el escalador. No obstante, remover el escalador resultaba perjudicial para el RMSE del modelo, por lo que se optó dejarlo a pesar de este inconveniente.
- 120 épocas: se encontró que aumentar el número de épocas a 120 mejoraba. Las épocas van utilizando los pesos de la época precedente (de inicio los pesos son aleatorios) y se van ajustando conforme al gradiente del error de la nueva época hasta finalizar el número de épocas elegido por el usuario. Cuando se seleccionaban 120 épocas, el RMSE de entrenamiento ya no bajaba más, pero tampoco llevaba tantas épocas de haber alcanzado su estado estable.
- Para el tensor de la recursión, se utilizaron cuatro periodos de ajuste (rezagos), los cuales tienen sentido al ser la cuenta corriente una serie trimestral, con factores estacionales encontrados en los modelos ARIMA. Se probó con 4, 6 y 12 periodos de ajuste, siendo 4 periodos de ajuste el modelo que arrojaba menor RMSE. Nota: las recursiones son convoluciones de la matriz de variables explicativas (suma ponderada de rezagos) y convoluciones de los rezagos de la variable objetivo.

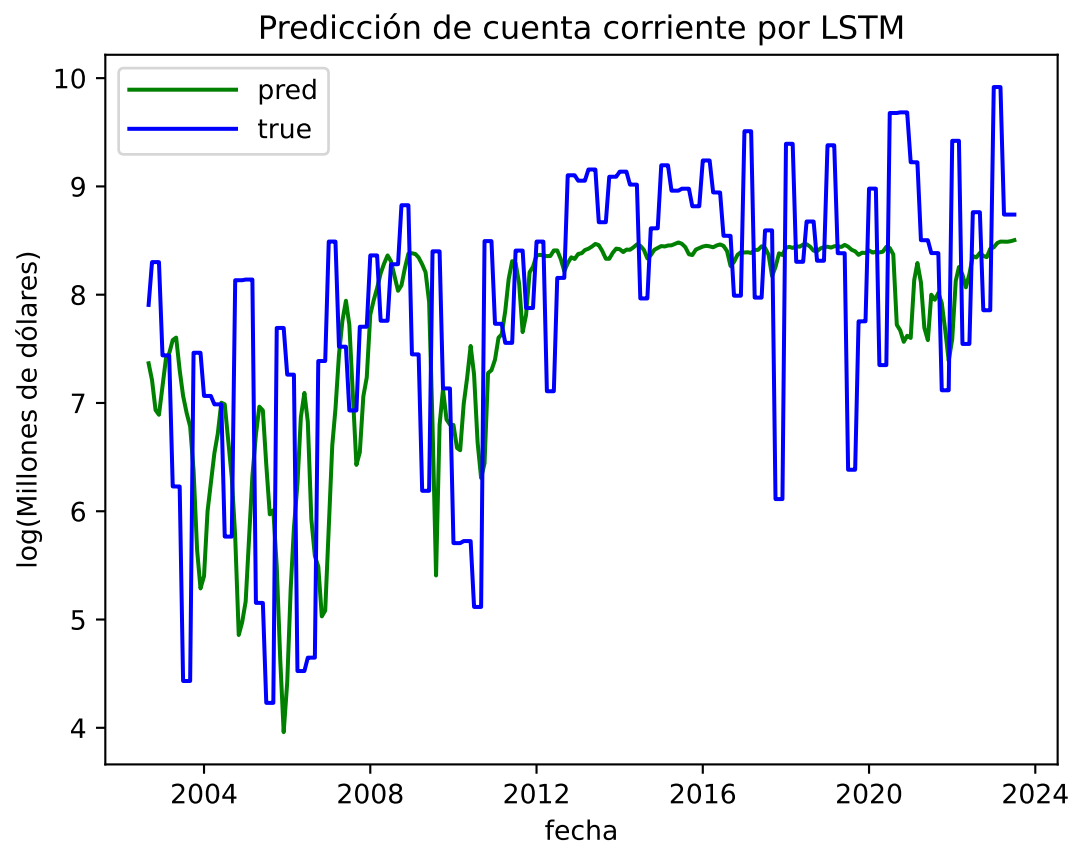


Modelos LSTM

El resumen del modelo LSTM que arrojó mejores resultados en RMSE fue el siguiente:

Capas (tipo)	Dimensión de salida	Param #
lstm (LSTM)	(N, 32)	4864
dense_4 (Dense)	(N, 64)	2112
dense_5 (Dense)	(N, 128)	8320
dense_6 (Dense)	(N, 64)	8256
leaky_re_lu (LeakyReLU)	(N, 64)	0
dense_7 (Dense)	(N, 1)	65

Núm. params: 23617 (todos entrenables)



Explicación de metodología

En general, una red neuronal es un conjunto de capas compuestas por nodos o perceptrones. Dichos perceptrones forman una combinación lineal con pesos iniciados al azar (posteriormente entrenados); a esta combinación se le aplica una transformación no lineal. Dicha transformación a su vez se vuelve insumo para entrenar a la siguiente capa.

El proceso de entrenamiento (a grandes rasgos) de una red neuronal consiste en lo siguiente:

1. Los pesos se inician al azar.
2. Conforme se le van aplicando transformaciones de capas o épocas anteriores, el modelo va evaluando su predicción con los datos reales.
3. El error resultante se usa para ajustar los pesos iniciados al azar, con un factor de ajuste.
4. Esos nuevos pesos se usan en una nueva época.
5. Se vuelve a predecir el error para volver a ajustar los pesos para la época siguiente.

Lo anterior se explica en el siguiente pseudocódigo (Román, 2023).

Algorithm 1 Gradient Descent

```
1: Initialize  $\Omega$  randomly.  
2: for each epoch do  
3:   for each sample do  
4:      $\hat{y}_i = f(x_i; \Omega)$   
5:      $E_i = l(y_i, \hat{y}_i)$   
6:      $\Omega = \Omega - \eta \nabla_{\Omega} E_i$   
7:   end for  
8: end for
```

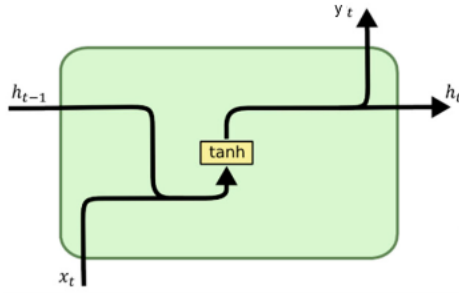
where, $\Omega = \{\omega_i\}$, $l(\cdot)$ is a loss function, and $\nabla_{\Omega} E_i$ is the gradient of the error with respect to the set Ω of parameters.

Dicha metodología tiene ventajas y desventajas. La principal ventaja es poder entrenar un modelo que sería imposible entrenar sin computadora (el número de pesos (parámetros) a entrenar pueden ser varios cientos de miles), aunque algunas desventajas son que pueden haber puntos silla, atorarse en un mínimo local, tener un factor de ajuste insuficiente o no percibir cambios en el límite de su dominio.

RNN

Considérese el siguiente diagrama (Román, 2023):

Recurrent Neural Network: Rumelhart, 1986.



$$\blacktriangleright y_t = h_t = \sigma(\mathbf{w}[\mathbf{x}_t, \mathbf{h}_{t-1}]).$$

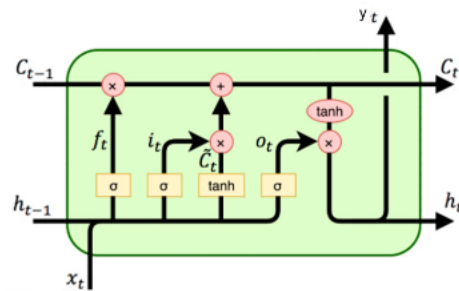
Se observa que en las redes neuronales recurrentes, se aplica una transformación no lineal a la (en este caso) matriz de recurrencia (la combinación lineal entre los rezagos de las explicativas y los rezagos de la variable objetivo o del producto de la capa anterior).

Si la recursión es many-to-many, el producto tanto se pasará a la siguiente capa como se guardará como producto final. En el caso de este estudio no es así, sino many-to-one (sólo tenemos una variable objetivo), pero tenemos varias variables explicativas, por lo que el producto de cada capa no se guarda como producto, sino se usa como insumo de la siguiente capa.

LSTM

Considérese el siguiente diagrama (Román, 2023):

Long Short-term Memory: Hochreiter and Schmidhuber, 1997.



- $\blacktriangleright y_t = h_t = o_t \odot \tanh(C_t),$
- $\blacktriangleright o_t = \sigma(\mathbf{w}_o[\mathbf{x}_t, \mathbf{h}_{t-1}]),$
- $\blacktriangleright f_t = \sigma(\mathbf{w}_f[\mathbf{x}_t, \mathbf{h}_{t-1}]),$
- $\blacktriangleright i_t = \sigma(\mathbf{w}_i[\mathbf{x}_t, \mathbf{h}_{t-1}]),$
- $\blacktriangleright \tilde{C}_t = \tanh(\mathbf{w}_C[\mathbf{x}_t, \mathbf{h}_{t-1}]),$
- $\blacktriangleright C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t.$

Biases are omitted for simplicity.
Also, all variables are vectors.

Think of each section as a gate: input gate, forget gate, output gate, and cell (state) gate.

La descripción es la siguiente: tenemos la transformación de insumo, de olvido y de salida, todas en recurrencias (es decir, convoluciones de la variable objetivo y de las variables explicativas). A la transformación de insumo se le multiplica otra transformación de tangente hiperbólica de una recurrencia; la transformación de olvido es multiplicada por la transformación anterior y el resultado se multiplica point-wise a la recurrencia de salida. Esta multiplicación es considerada como el producto que se usará en las recurrencias de la capa siguiente.

Entonces, la interpretación es la siguiente: los pesos de salida no ven las transformaciones hiperbólicas, mientras que las de entrada son generadas en esa misma capa. Conforme van avanzando el número de capas, la transformación de olvido va recopilando información generada en esa capa e información generada en capas anteriores. Lo anterior implica que va a haber información preponderantemente de las capas más recientes; el efecto de la capa contemporánea sobre el siguiente producto será mayor.

El hecho de que algunos pesos estén sujetos a más transformaciones que otros implica que las variables explicativas, pesos e hiperparámetros tendrán distinta incidencia. Se observa que las explicativas se usan en todas las capas y en todas las transformaciones, por lo que implica que todos los pesos, aunque tengan más o menos transformaciones, aunque se disipen en el tiempo, tendrán muy presentes a las explicativas. Por otro lado, el efecto del tiempo mayor al rezago uno es indirecto y dado que se aplican varias transformaciones de tangente hiperbólica, se puede desvanecer, lo cual podría explicar su desventaja respecto de los modelos autorregresivos.

En otras palabras, se observa que LSTM es una red de redes. Las transformaciones de entrada, de olvido y de salida tienen la misma fórmula que las redes neuronales recurrentes, con diferentes transformaciones. Por ello LSTM es un modelo más sofisticado que RNN.

Referencias

- IBM, *What are neural networks?*, 2022, (<https://www.ibm.com/topics/neural-networks#:~:text=Neural%20networks%2C%20also%20known%20as,neurons%20signal%20to%20one%20another>).
- Román, Edgar, *Recurrent Neural Networks, deep learning*, Instituto Tecnológico Autónomo de México, Department of Computer Science, 2023.
- Román, Edgar, *MLP - Backprop*, Instituto Tecnológico Autónomo de México, Department of Computer Science, 2023.