Análisis y predicción de series temporales usando redes neuronales recurrentes

Emilio Barragán Rodríguez

ETSII - Universidad de Sevilla

15 de julio de 2021

Índice

- Series temporales
  - Clasificación
  - Componentes
- Inteligencia Artificial
  - Aprendizaje Automático
  - Redes Neuronales
- Deep Learning
  - RNN
  - LSTM
  - GRU
- Análisis y predicción de series temporales
  - Estudio de la estacionaridad
  - Transformación de una serie no estacionaria a estacionaria
  - Framework para problemas de series temporales
  - Implementación y desarrollo: Casos COVID-19 en España
- Conclusiones

Introducción

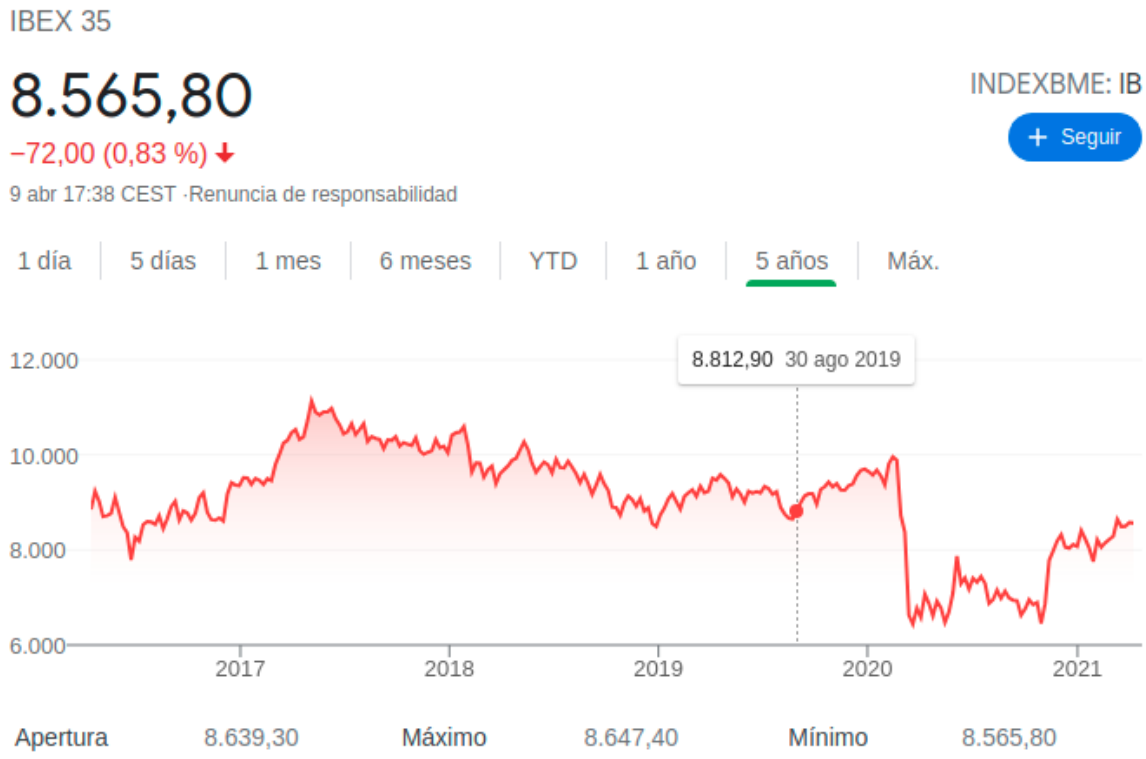
- Contexto:
  - Necesidad de predecir el futuro
  - Aumento de capacidad de cómputo y datos
- Objetivo del presente documento:
  - Series temporales
  - RNN

Índice

- Series temporales
  - Clasificación
  - Componentes
- Inteligencia Artificial
  - Aprendizaje Automático
  - Redes Neuronales
- Deep Learning
  - RNN
  - LSTM
  - GRU
- Análisis y predicción de series temporales
  - Estudio de la estacionaridad
  - Transformación de una serie no estacionaria a estacionaria
  - Framework para problemas de series temporales
  - Implementación y desarrollo: Casos COVID-19 en España
- Conclusiones

Series Temporales

- Sucesión de observaciones
- Eje X tiempo, eje Y valores



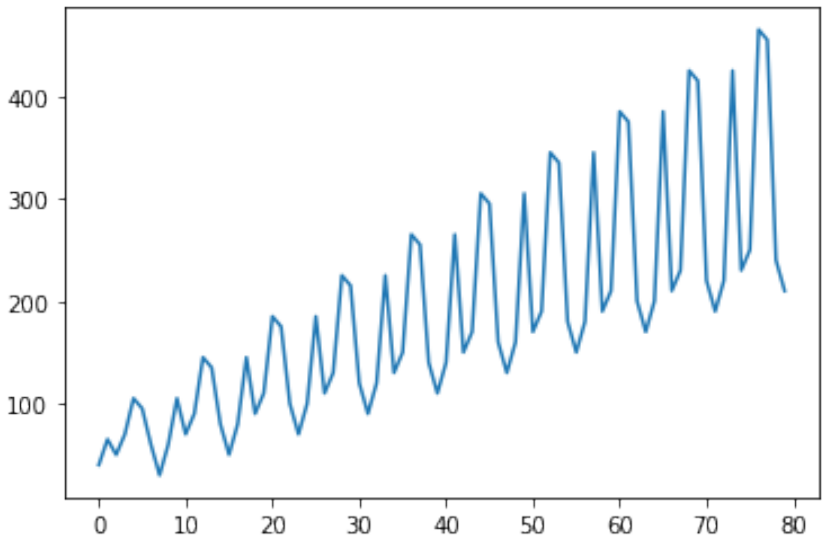
Clasificación

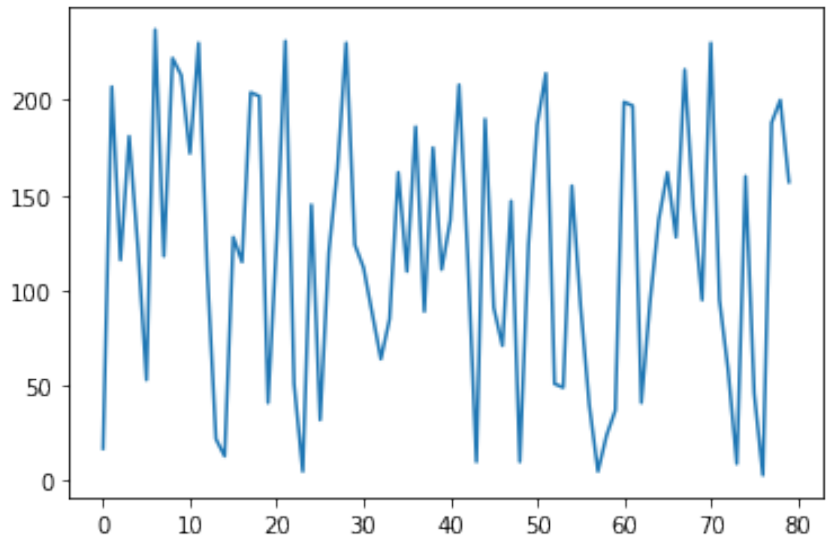
- Estacionarias:
  - $E[X_t] = \mu \ \ \forall t \in T$
  - $Var(X_t) = \sigma^2 \ \ \forall t \in T$
  - $Cov(X_t, X_{t+k}) = \gamma_k \ \ \forall t \in T, \forall k \in \{1..|T| - t\}$
- No estacionarias:
  - Tendencia
  - Media variable
  - Estacionalidad

Componentes

- Variables:
  - Univariable
  - Multivariable
- Nivel:
  - Estable
  - Inestable
  - Circunstancial
- Tendencia:
  - Creciente
  - Decreciente
- Estacionalidad
- Ruido

Componentes: Ejemplos





Índice

- Series temporales
  - Clasificación
  - Componentes
- Inteligencia Artificial
  - Aprendizaje Automático
  - Redes Neuronales
- Deep Learning
  - RNN
  - LSTM
  - GRU
- Análisis y predicción de series temporales
  - Estudio de la estacionaridad
  - Transformación de una serie no estacionaria a estacionaria
  - Framework para problemas de series temporales
  - Implementación y desarrollo: Casos COVID-19 en España
- Conclusiones

Inteligencia Artificial

- ¿Qué es?: “La ciencia e ingenio de hacer máquinas inteligentes”
- Etapas:
  - Entusiasmo (1952-1969)
  - Realidad (1966-1973)
  - Sistemas basados en conocimiento (1969-1979)
  - Industria (1980-Actualidad)

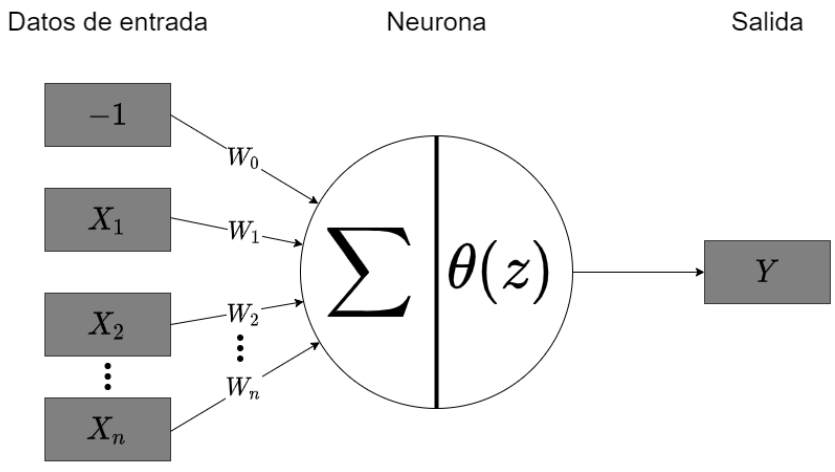
Aprendizaje Automático

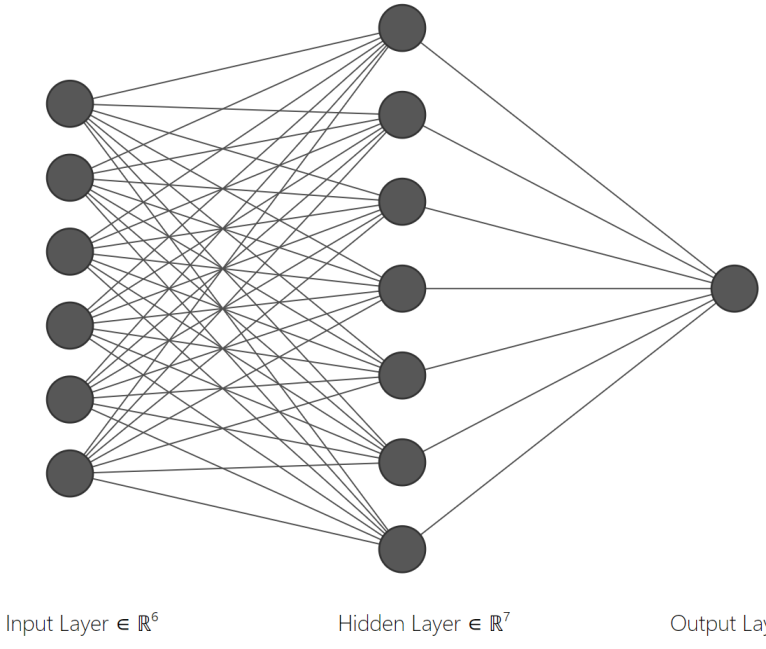
- ¿Qué es?: Rama de la inteligencia artificial. Se centra en intentar que las máquinas “aprendan”.

- Tipos:
  - Aprendizaje no supervisado
  - Aprendizaje supervisado
  - Aprendizaje por refuerzo

Redes Neuronales

- ¿Qué son?: Modelo concreto del campo del machine learning. Tratan de emular en cierta manera el cerebro humano, imitando las neuronas, conexiones entre ellas, etc.



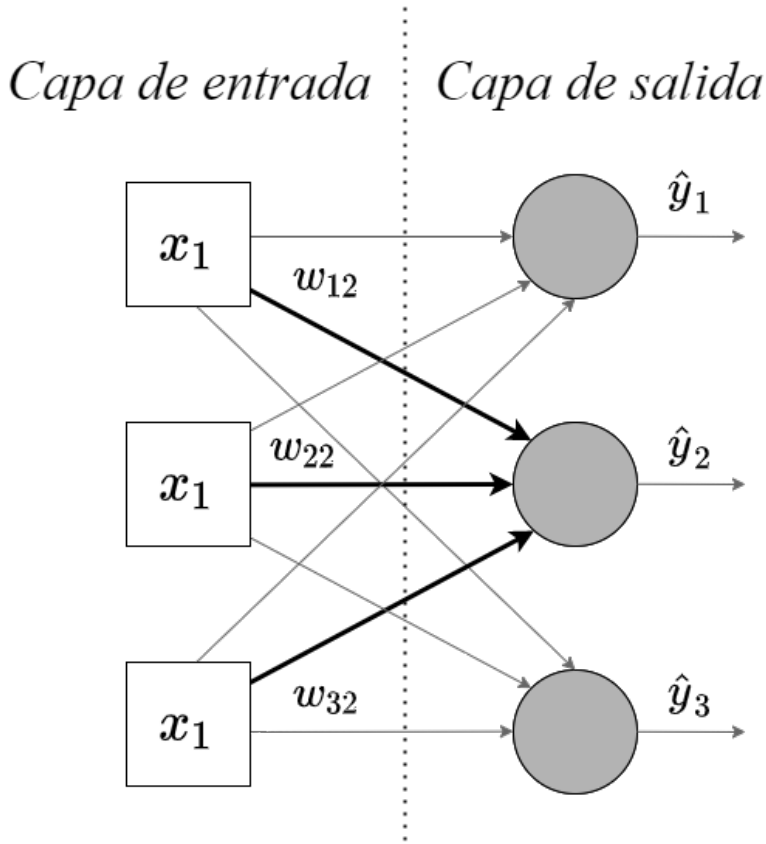


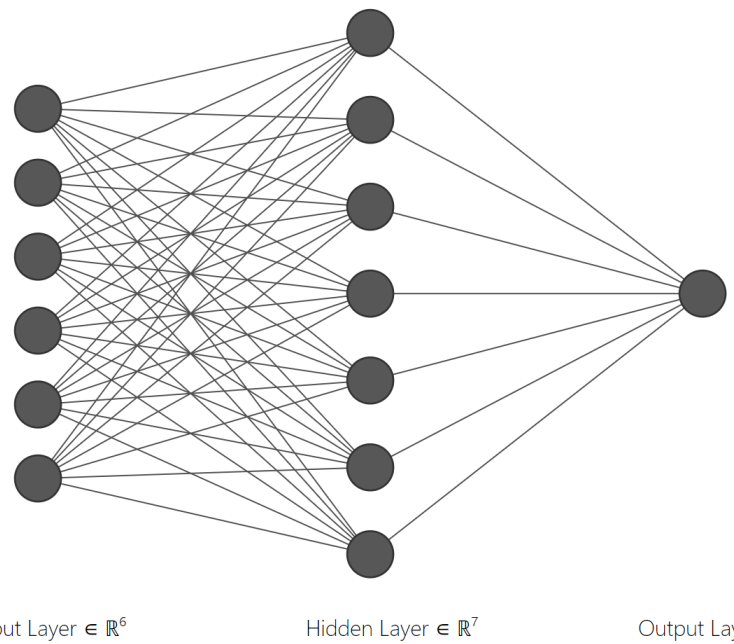
Redes Neuronales: Aprendizaje

$$w_{i,j}^{(siguiente)} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i$$

$$w^{L-1}(t+1) = w^{L-1}(t) - \alpha \frac{\partial C}{\partial w^{(L-1)}}$$

$$b^{L-1}(t+1) = b^{L-1}(t) - \alpha \frac{\partial C}{\partial b^{(L-1)}}$$





Índice

- Series temporales
  - Clasificación
  - Componentes
- Inteligencia Artificial
  - Aprendizaje Automático
  - Redes Neuronales
- Deep Learning
  - RNN
  - LSTM
  - GRU
- Análisis y predicción de series temporales
  - Estudio de la estacionaridad
  - Transformación de una serie no estacionaria a estacionaria
  - Framework para problemas de series temporales
  - Implementación y desarrollo: Casos COVID-19 en España
- Conclusiones

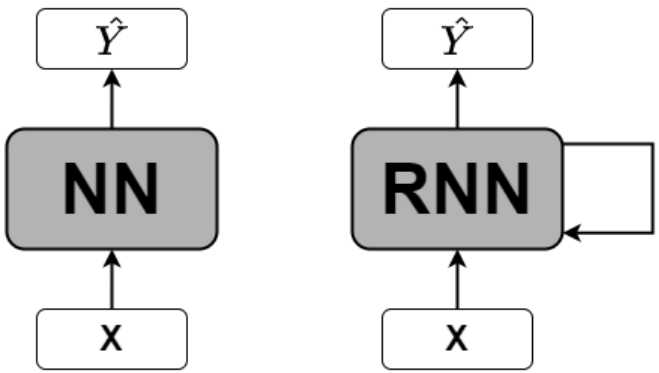
Deep Learning

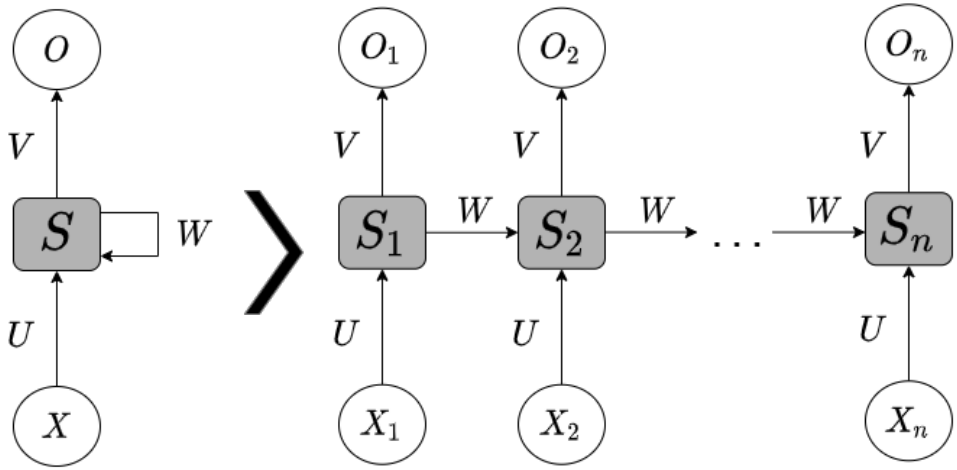
- ¿Qué es?: Un subcampo del aprendizaje automático. En él se estudian redes neuronales que son estructuralmente más complejas que el simple perceptrón multicapa.

- Origen e hitos:
  - Anterior a los años 70
  - Auge en 2011-2012
  - Google Brain

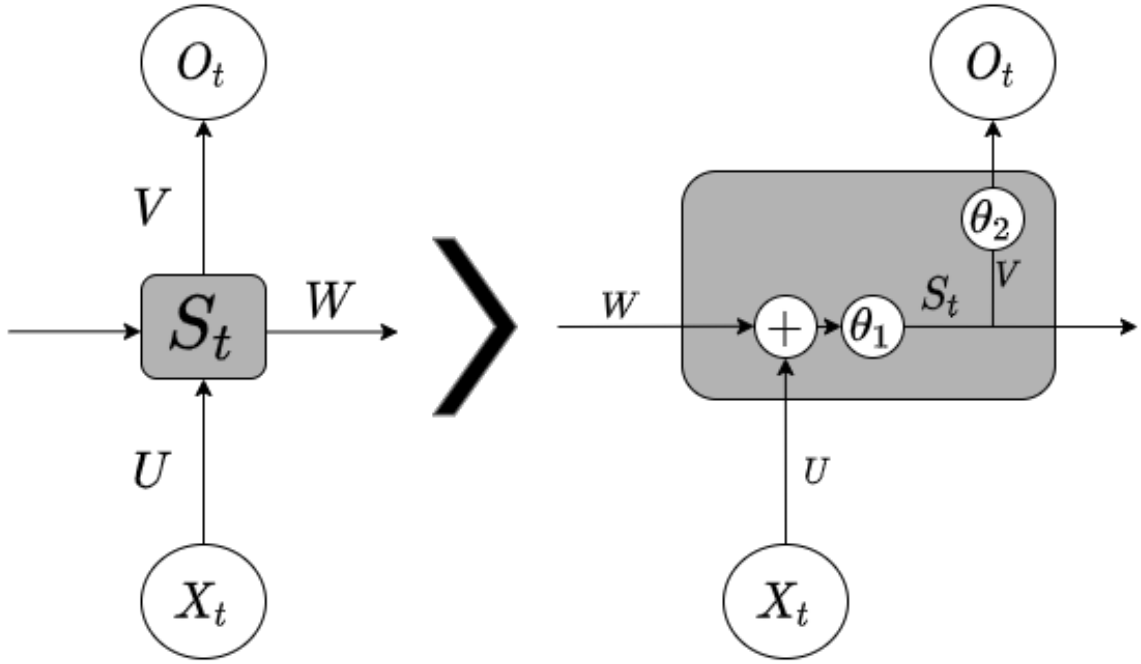
RNN

- Idea: Hacer uso de información secuencial.





RNN: Estructura y aprendizaje



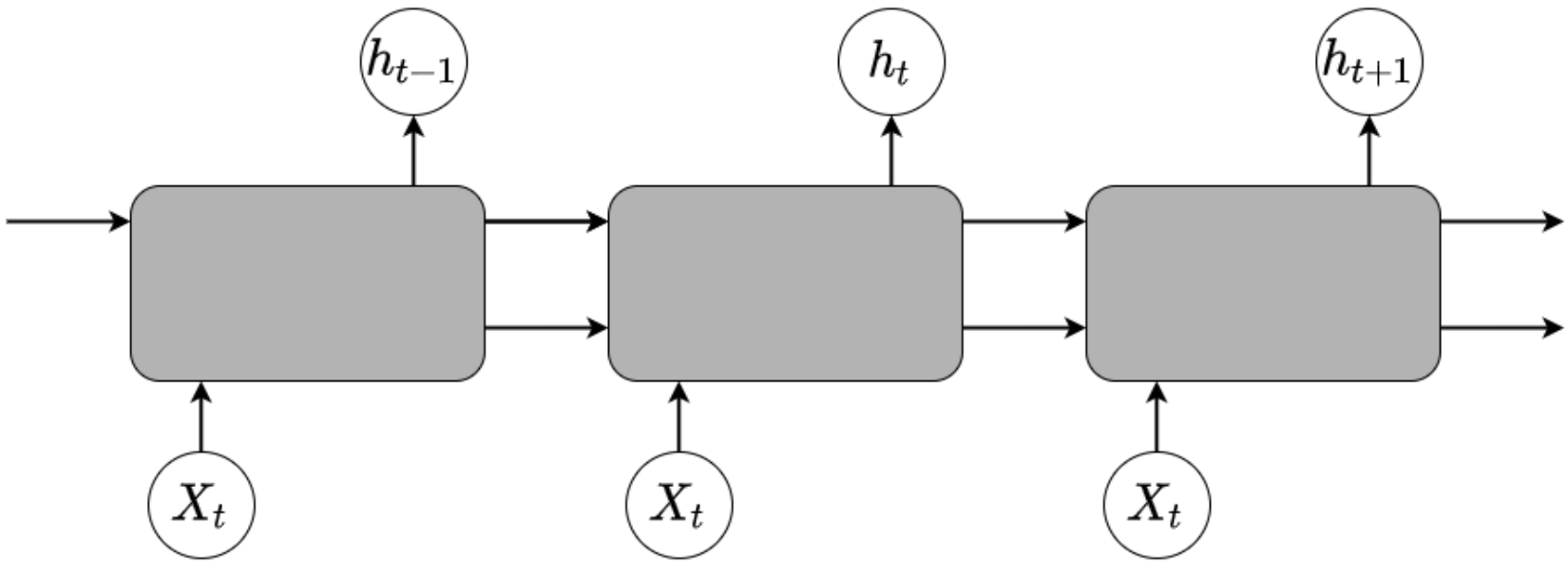
$$\boldsymbol{W_i(t+1)} = W_i(t) - \alpha \frac{\partial C_i}{\partial W_i}$$

$$\boldsymbol{V_i(t+1)} = V_i(t) - \alpha \frac{\partial C_i}{\partial V_i}$$

$$\boldsymbol{U_i(t+1)} = U_i(t) - \alpha \frac{\partial C_i}{\partial U_i}$$

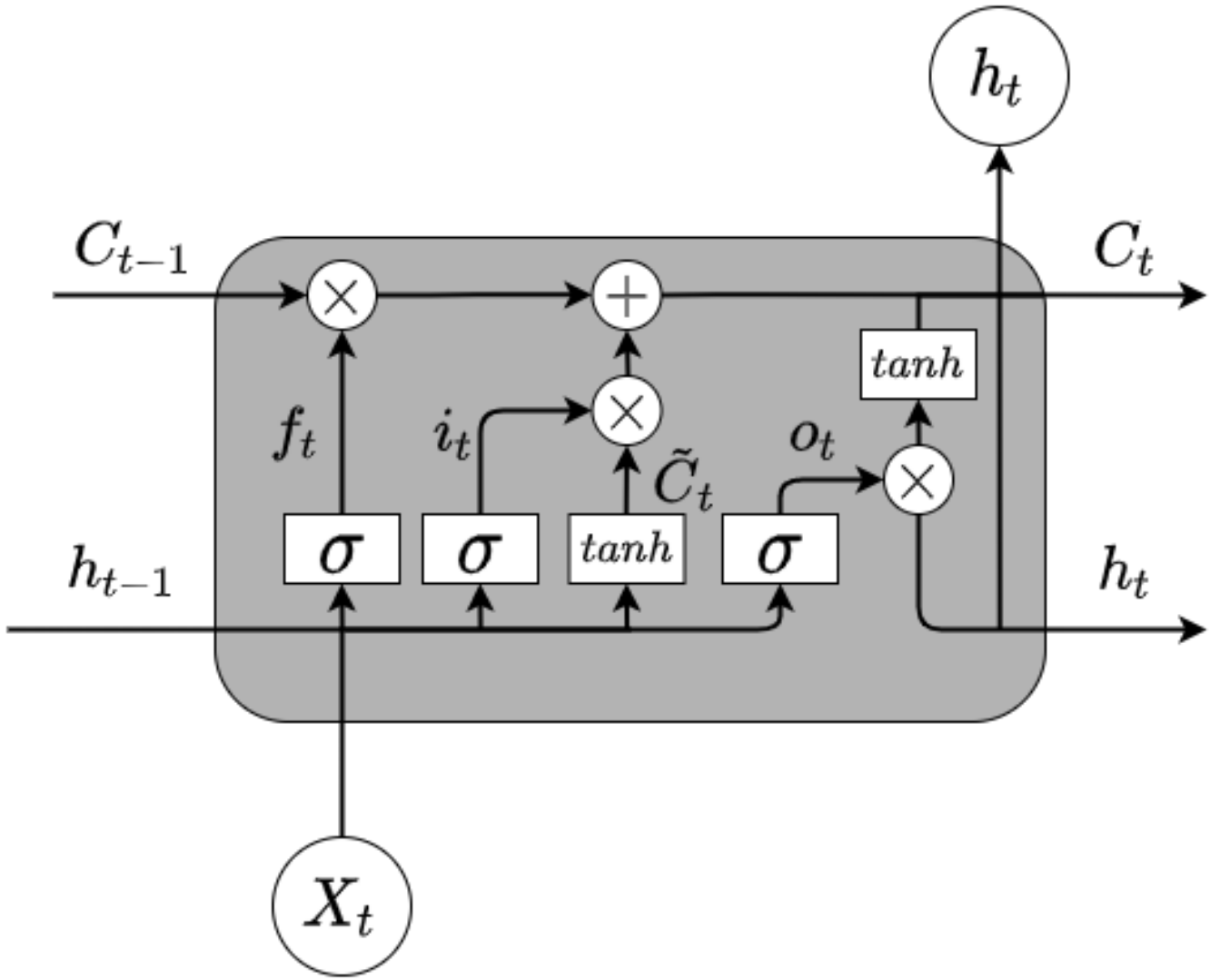
LSTM

- ¿Qué son?: Tipo de red RNN que pueden aprender dependencias a largo plazo.



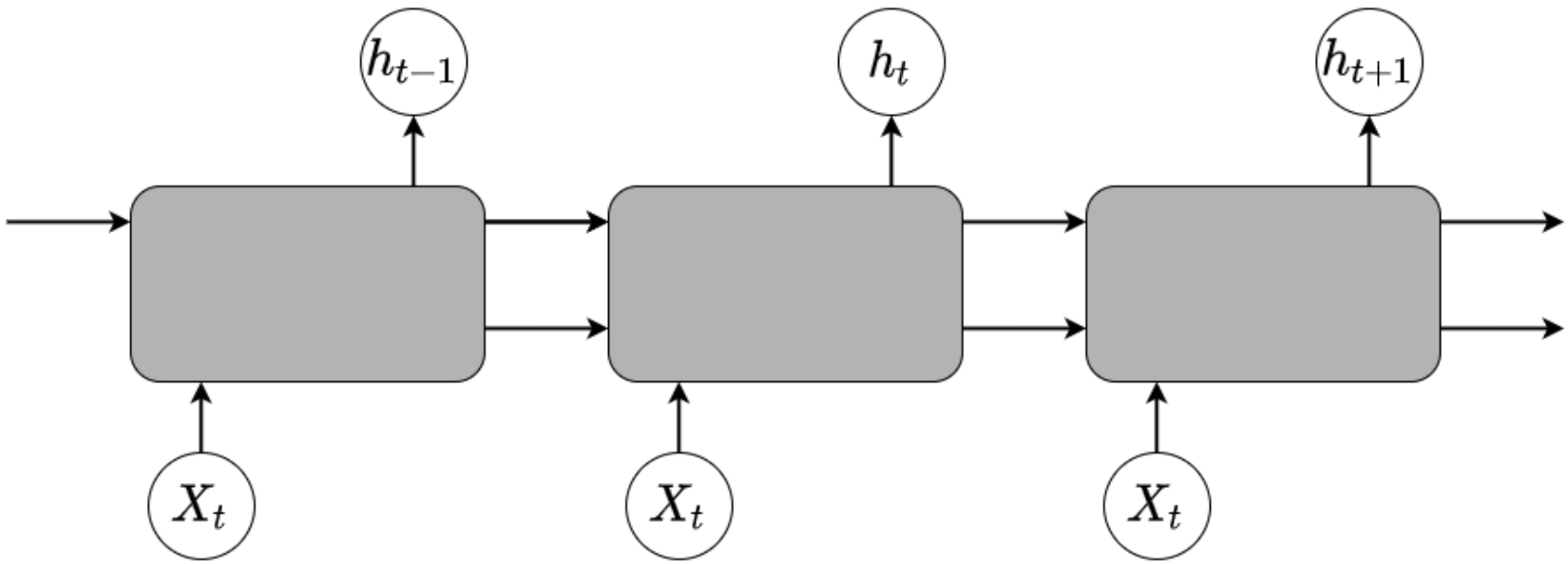
LSTM

$$\boldsymbol{C_t} = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$



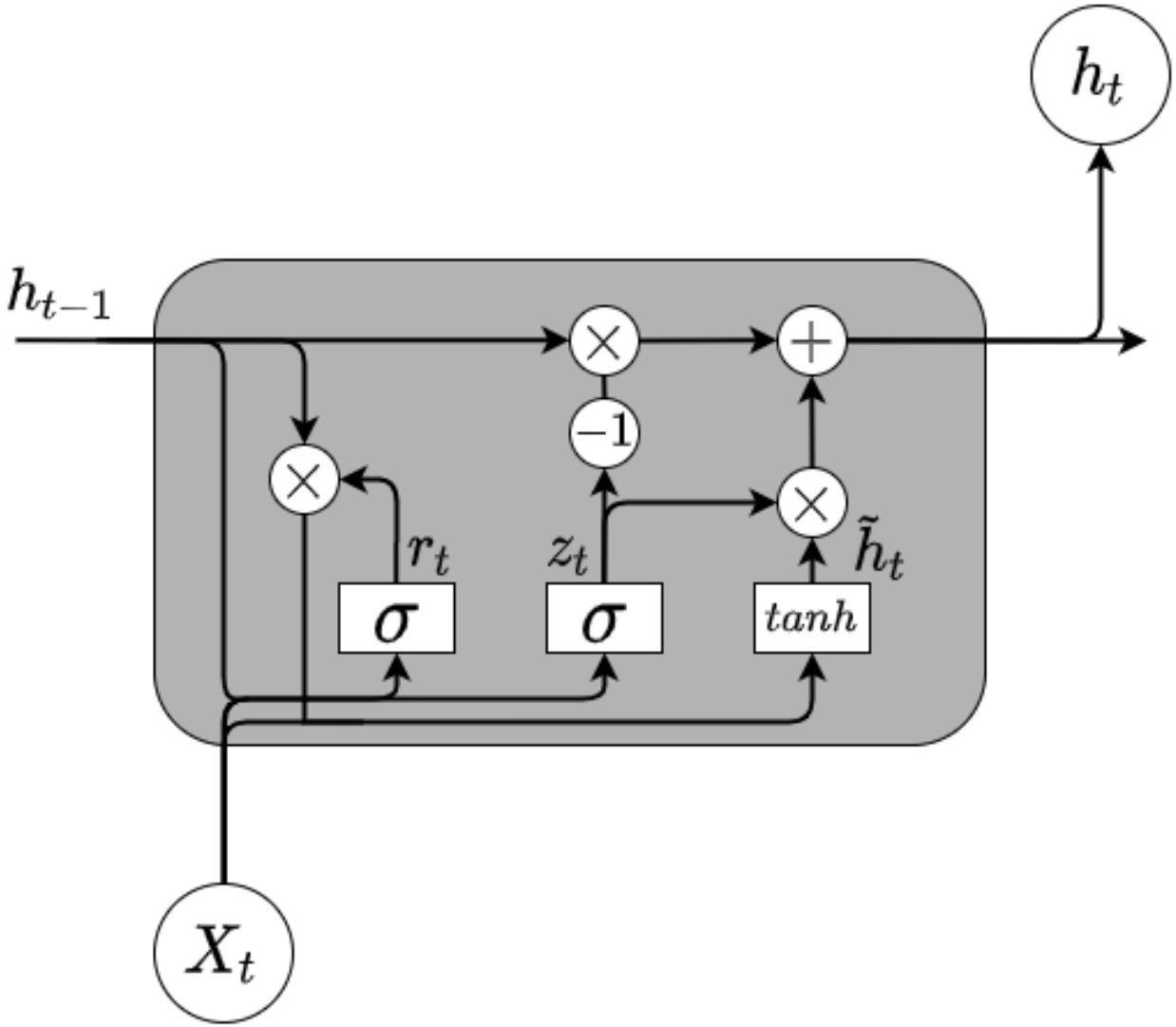
GRU

- ¿Qué son?: Variación de las redes LSTM. Estas combinan la puerta de olvido y la puerta de entrada en una sola puerta de actualización.



GRU

$$\boldsymbol{h_t} = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t$$



Índice

Series temporales
- Clasificación
- Componentes

Inteligencia Artificial
- Aprendizaje Automático
- Redes Neuronales

Deep Learning
- RNN
- LSTM
- GRU

Análisis y predicción de series temporales
- Estudio de la estacionaridad
- Transformación de una serie no estacionaria a estacionaria
- Framework para problemas de series temporales
- Implementación y desarrollo: Casos COVID-19 en España

Conclusiones

Análisis y predicción de series temporales

- Métodos para estudiar la estacionaridad
- Método para transformar no estacionaria a estacionaria
- Aplicación ejemplo real: COVID-19

Estudio de la estacionaridad

- Contraste de Dickey-Fuller:
  - H_θ: La serie tiene una raíz unitaria.
  - H_1: La serie no tiene raíz unitaria.

- Contraste de Kwiatkowski-Phillips-Schmidt-Shin:
  - H_θ: La serie no tiene una raíz unitaria.
  - H_1: La serie tiene raíz unitaria.

- Caso 1: Ambos tests dicen que la serie no es estacionaria.
- Caso 2: Ambos tests dicen que la serie es estacionaria.
- Caso 3: El test KPSS dice que es estacionaria y el test ADF dice que no es estacionaria
- Caso 4: El test KPSS dice que no es estacionaria y el test ADF dice que es estacionaria.

Transformación de una serie no estacionaria a estacionaria

Una serie temporal se diferencia (o se resta) de la siguiente manera:

$$difference(t) = observation(t) - observation(t-1)$$

Se puede revertir haciendo:

$$inverted(t) = differenced(t) + observation(t-1)$$

- Desfase
- Orden de diferencia

Framework para problemas de series temporales

1. Entrada vs. Salida.
2. Endógeno vs. Exógeno.
3. Regresión vs. Clasificación.
4. Desestructurado vs. Estructurado.
5. Univariable vs. Multivariable.
6. Un paso vs. Varios pasos.
7. Estático vs. Dinámico.
8. Homogéneo vs. Heterogéneo.

Implementación y desarrollo: Casos COVID-19 en España

- Tecnologías:
  - Python (Pandas, Numpy, Sklearn, statsmodel, joblib, math)
  - Tensorflow (Keras)
  - Google Colaboratory

```
import math
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import ScalarFormatter
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM, GRU
from keras.models import load_model
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import MinMaxScaler
from statsmodels.tsa.stattools import kpss
from statsmodels.tsa.stattools import adfuller
import joblib
```

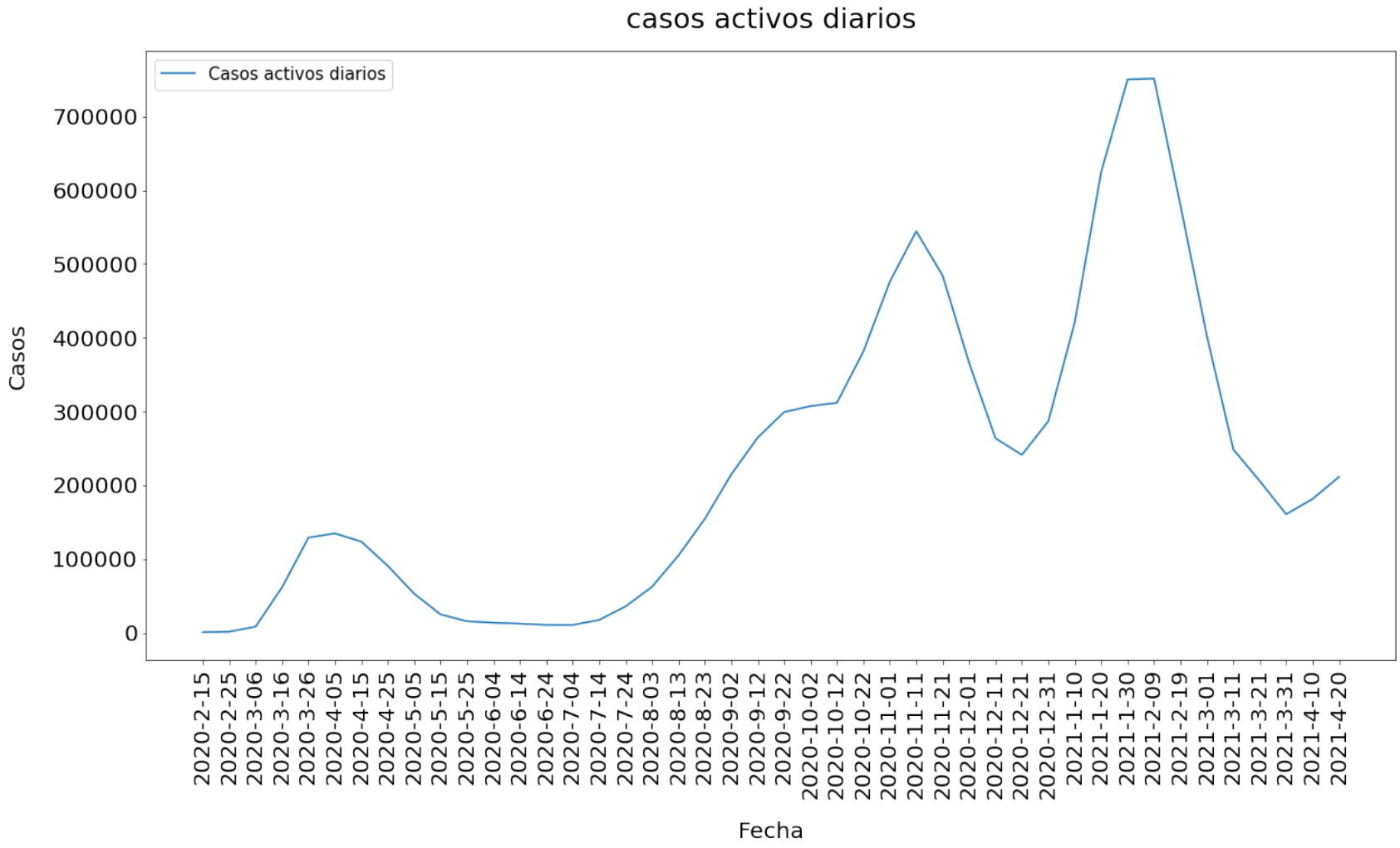
Implementación y desarrollo: Casos COVID-19 en España

- Dataset:
  - 7 columnas y 96825 filas (a fecha 29 de Abril de 2021)
  - Comprendidos entre las fechas 15-02-2020 y 20-4-2021, con una observación por día

	date	country	cumulative_total_cases	daily_new_cases	active_cases	cumulative_total_deaths	daily_new_deaths
81864	2020-2-15	Spain	710.0	NaN	708.0	0.0	NaN
81865	2020-2-16	Spain	736.0	26.0	734.0	0.0	NaN
81866	2020-2-17	Spain	764.0	28.0	762.0	0.0	NaN
81867	2020-2-18	Spain	792.0	28.0	790.0	0.0	NaN
81868	2020-2-19	Spain	810.0	18.0	808.0	0.0	NaN
...	...	...	...	...	...	...	...
82299	2021-4-25	Spain	3481969.0	7949.0	226645.0	77689.0	49.0
82300	2021-4-26	Spain	3488469.0	6500.0	227837.0	77738.0	49.0
82301	2021-4-27	Spain	3496134.0	7665.0	231312.0	77855.0	117.0
82302	2021-4-28	Spain	3504799.0	8665.0	233886.0	77943.0	88.0
82303	2021-4-29	Spain	3514942.0	10143.0	243892.0	78080.0	137.0

440 rows × 7 columns

Implementación y desarrollo: Casos COVID-19 en España



Entendiendo nuestro problema

1. **Entrada vs. Salida**: Datos de los casos nuevos diarios de covid de los últimos 7 días VS. Predicción para los 7 días siguientes.
2. **Endógeno vs. Exógeno**: Endógeno.
3. **Regresión vs. Clasificación**: Regresión.
4. **Desestructurado vs. Estructurado**: Estructurado.
5. **Univariable vs. Multivariable**: Univariable.
6. **Un paso vs. Varios pasos**: Varios pasos.
7. **Estático vs. Dinámico**: Estático.
8. **Homogéneo vs. Heterogéneo**: Homogénea.

Estudio de la estacionaridad

```
(1.3258295596855243, 0.01, 18, {'10%': 0.347, '5%': 0.463, '2.5%': 0.574, '1%': 0.739})
KPSS Test Statistic: 1.33
5% Critical Value: 0.46
p-value: 0.01
ADF Test Statistic: -1.92
5% Critical Value: -2.87
p-value: 0.32
```

```
data_spain['Difference'] = data_spain['active_cases'].diff()
```

```
(0.09984500996522702, 0.1, 18, {'10%': 0.347, '5%': 0.463, '2.5%': 0.574, '1%': 0.739})
KPSS Test Statistic: 0.10
5% Critical Value: 0.46
p-value: 0.10
ADF Test Statistic: -4.82
5% Critical Value: -2.87
p-value: 0.00
```

Creación de los modelos

```python
def create_lstm_model(X, y, path):
    model = Sequential()
    model.add(LSTM(25, activation='tanh', input_shape=(X.shape[1], X.shape[2]),
                   dropout=0.2))
    model.add(Dense(10))
    model.add(Dense(10))
    model.add(Dense(y.shape[1]))
    model.compile(optimizer='adam', loss='mse')

    # entrenamos el modelo
    model.fit(X, y, epochs=70,batch_size=16, verbose=0, shuffle=False)
    model.summary()
    model.save(path)

def create_gru_model(X, y, path):
    model = Sequential()
    model.add(GRU(25, activation='tanh', input_shape=(X.shape[1], X.shape[2]),
                  dropout=0.2))
    model.add(Dense(10))
    model.add(Dense(10))
    model.add(Dense(y.shape[1]))
    model.compile(optimizer='adam', loss='mse')

    # entrenamos el modelo
    model.fit(X, y, epochs=70,batch_size=16, verbose=0, shuffle=False)
    model.summary()
    model.save(path)
```

Creación de los modelos

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 25)	2700
dense (Dense)	(None, 10)	260
dense_1 (Dense)	(None, 10)	110
dense_2 (Dense)	(None, 7)	77

Total params: 3,147
Trainable params: 3,147
Non-trainable params: 0

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 25)	2100
dense_3 (Dense)	(None, 10)	260
dense_4 (Dense)	(None, 10)	110
dense_5 (Dense)	(None, 7)	77

Total params: 2,547
Trainable params: 2,547
Non-trainable params: 0

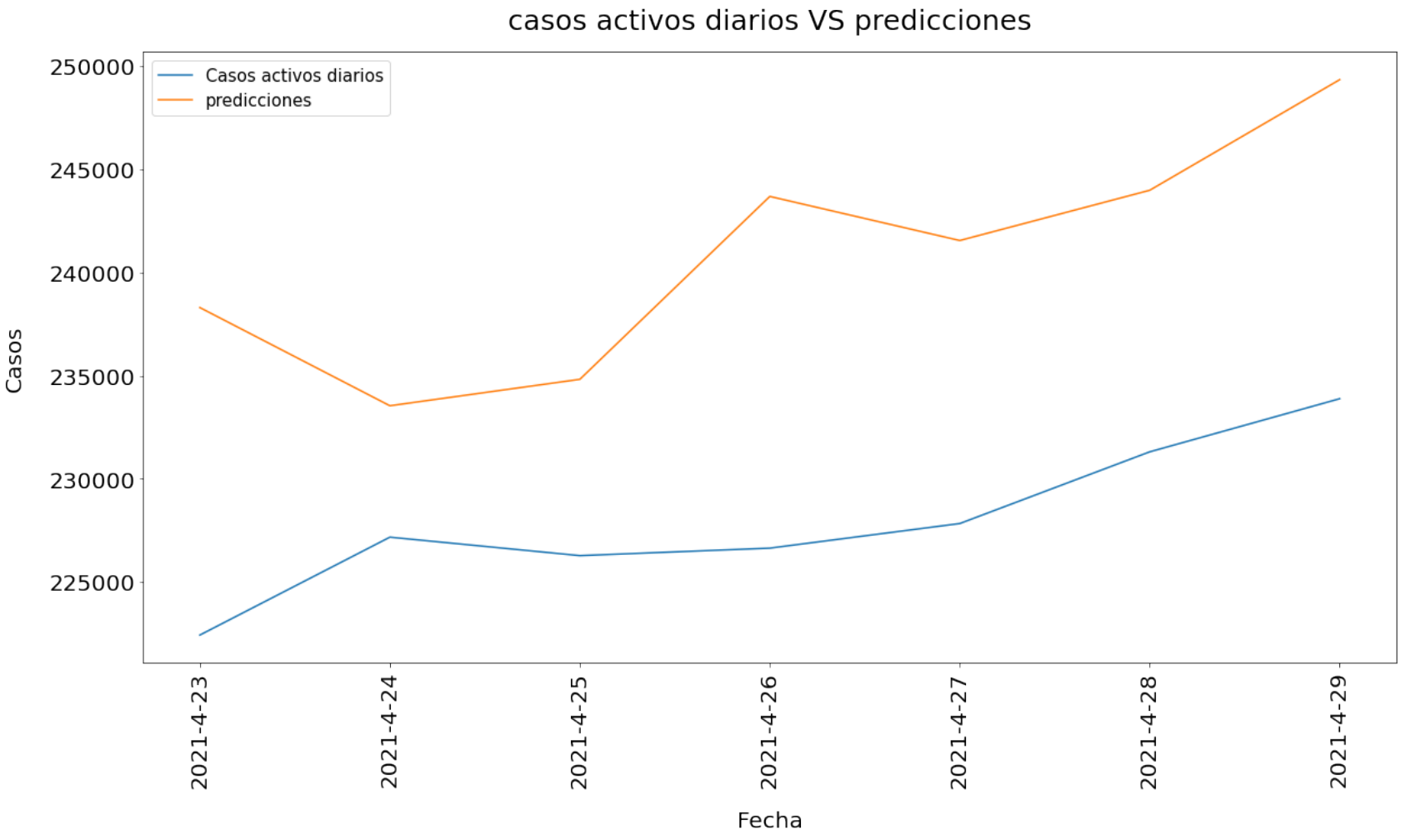
Resultados

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

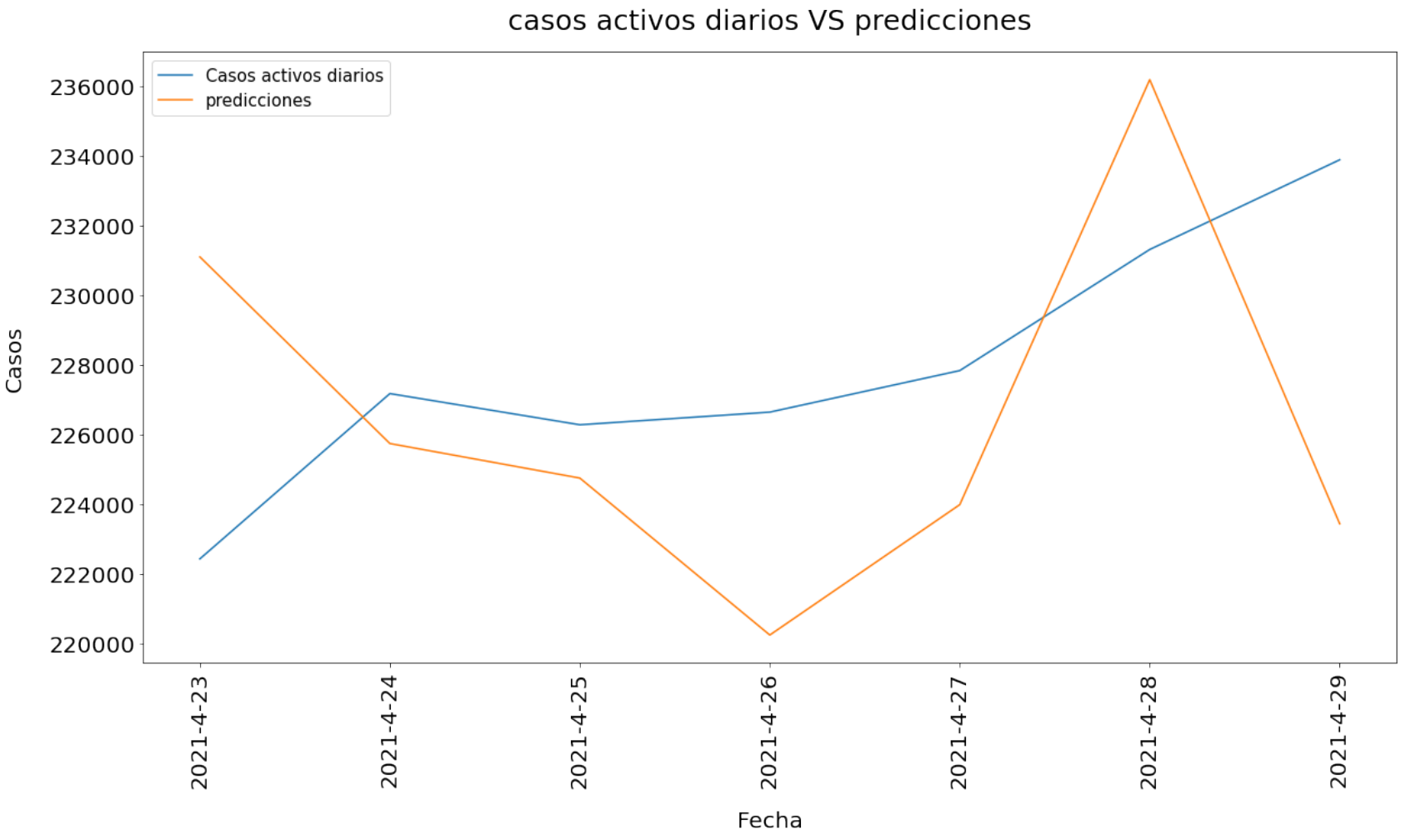
RMSE LSMT: 13327.89

RMSE GRU: 6187.47

Resultados


casos activos diarios VS predicciones
Casos activos diarios
predicciones
250000
245000
240000
235000
230000
225000
Casos
2021-4-23
2021-4-24
2021-4-25
2021-4-26
2021-4-27
2021-4-28
2021-4-29
Fecha

Resultados


casos activos diarios VS predicciones
Casos activos diarios
predicciones
Casos
236000
234000
232000
230000
228000
226000
224000
222000
220000
2021-4-23
2021-4-24
2021-4-25
2021-4-26
2021-4-27
2021-4-28
2021-4-29
Fecha

Índice

- Series temporales
  - Clasificación
  - Componentes
- Inteligencia Artificial
  - Aprendizaje Automático
  - Redes Neuronales
- Deep Learning
  - RNN
  - LSTM
  - GRU
- Análisis y predicción de series temporales
  - Estudio de la estacionaridad
  - Transformación de una serie no estacionaria a estacionaria
  - Framework para problemas de series temporales
  - Implementación y desarrollo: Casos COVID-19 en España
- Conclusiones

Conclusiones

- Rendimiento de las LSTM vs. GRU
- Dificultad de predicción de COVID-19

- Consideraciones futuras:
  - Sistemas caóticos
  - Mayor complejidad en LSTM y GRU
  - Otras arquitecturas RNN