



Análisis de Series de Tiempo

Carrera de Especialización en Inteligencia Artificial

Clase ⁶

Ing. Magdalena Bouza, Ing. Carlos German Carreño Romano

Contenido

- Caso de estudio
- Modelos de regresión
 - Análisis de intervenciones
 - Tratamiento de outliers
- Otros modelos estacionales
 - Modelo exponencial
 - Modelo Holt-Winters
- Consultas
- Revisión bibliográfica

Caso de estudio
Pronósticos de la demanda de
capacidad de **Internet**

Sizing Techniques applied to Network Capacity Planning

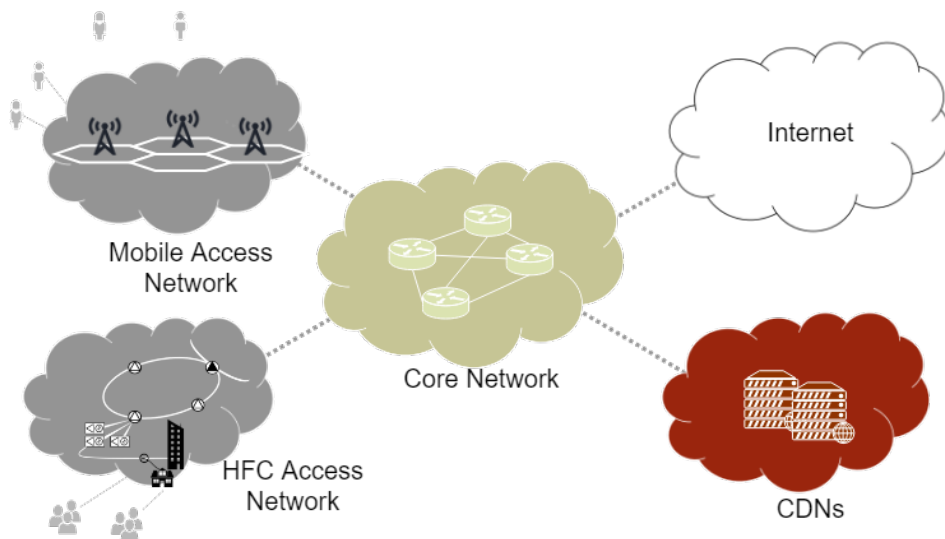
Técnicas de dimensionamiento aplicadas al planeamiento de capacidad de red

- Carlos G. Carreño Romano
- Natalia A. Clivio Velilla

<https://ieeexplore.ieee.org/document/8646077>

Aplicaciones en dimensionamiento de redes

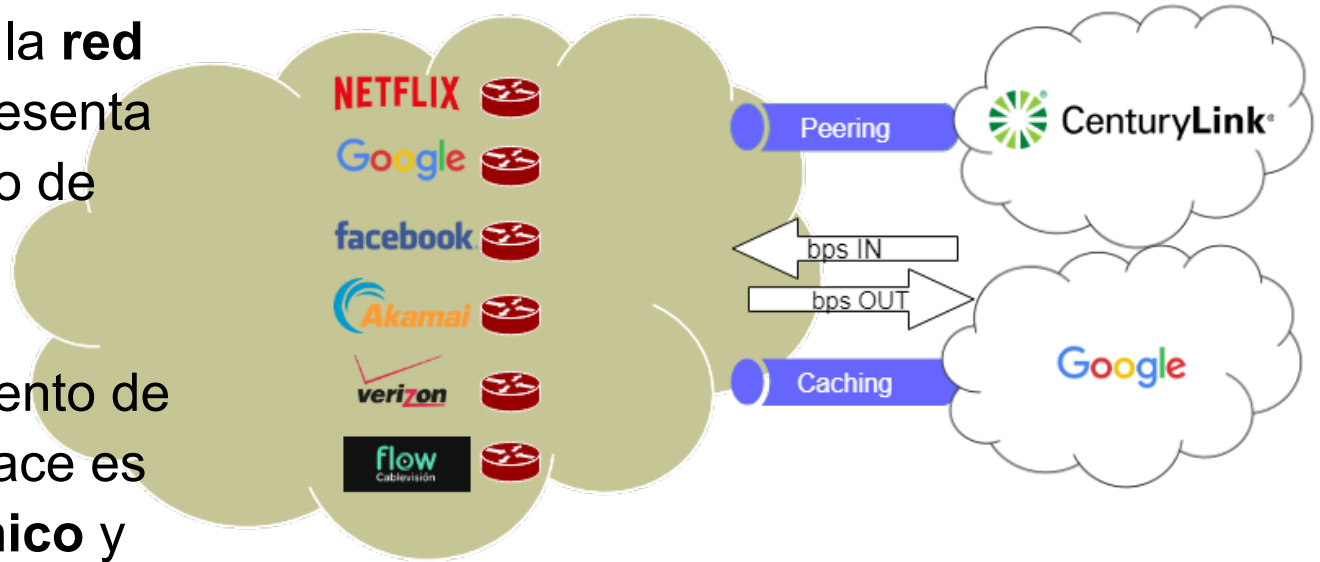
- Proveedores de Servicios de Comunicaciones (CSP)
- Redes de acceso
- Redes Core
- Redes de Distribución de Contenidos (CDN)
- Internet



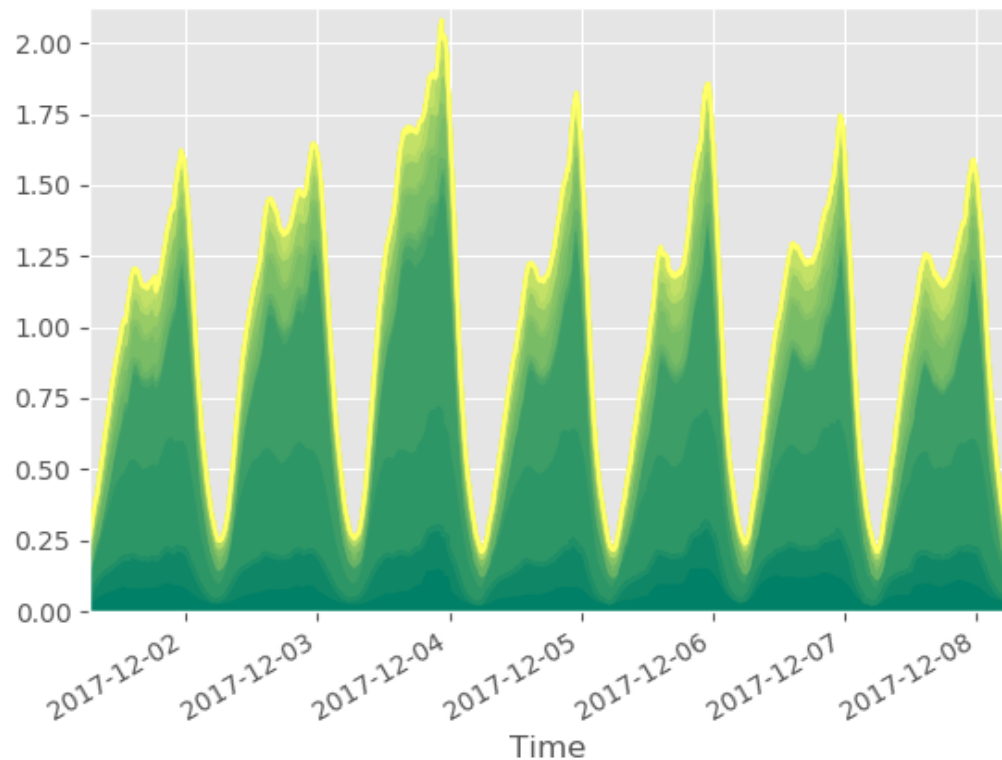
Contexto

Cada enlace entre la **red Core** y las **CDN** presenta una serie de tiempo de tráfico.

Predecir el crecimiento de tráfico en cada enlace es de interés **económico** y **técnico**.



Dataset

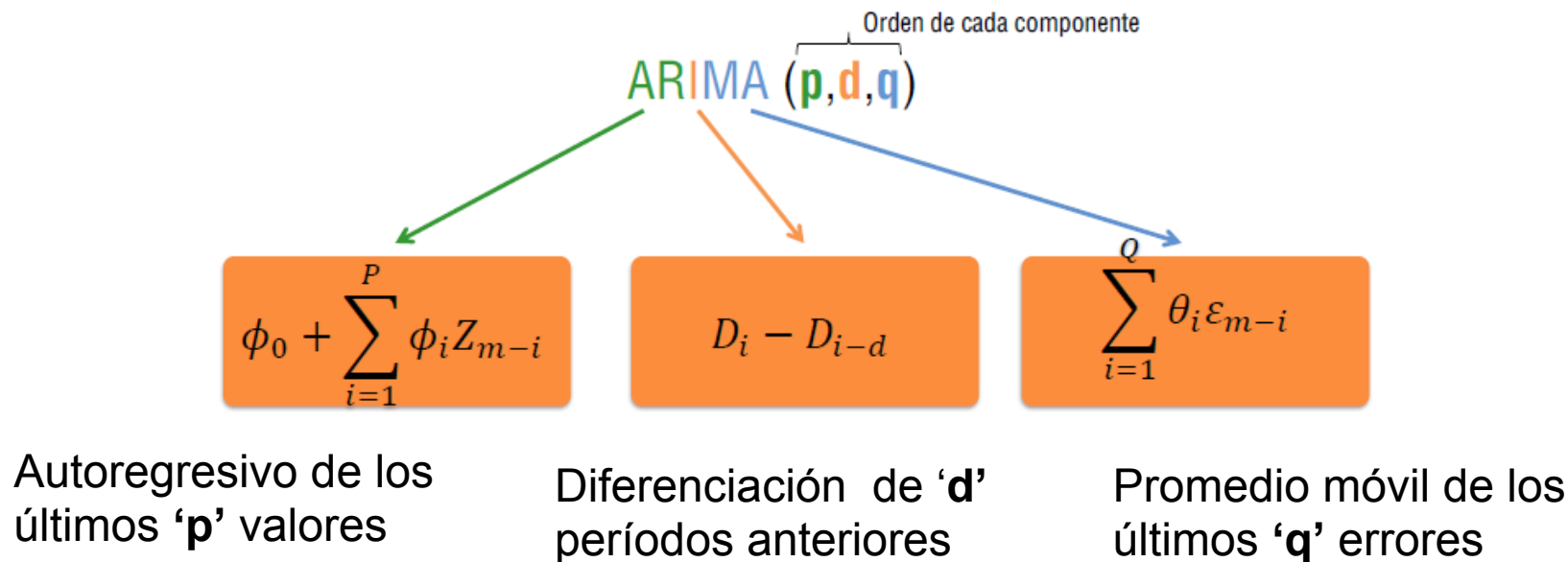


Datos de 1 año con una granularidad de 5 minutos.

Objetivo: Comparar predicciones generadas con ARIMA y LSTM RNN para las trazas de tráfico real de la empresa Telecom Argentina

Modelado usando ARIMA

ARIMA viene de **A**utorregresión **I**ntegrados de **M**edia móvil



Modelado usando ARIMA

Para aplicar modelos ARIMA se suele descomponer la serie, analizando en primer lugar la **tendencia** de la serie, luego la **estacionalidad** y concentrándose en identificar estas componentes filtradas. Hay otras dos componentes que hacen el modelo completo y son las **componentes cíclicas** y las **componentes aleatorias**. El proceso consiste en la descomposición de la serie en forma aditiva o multiplicativa. Usamos en este trabajo la forma aditiva y definimos entonces una serie de tiempo $Y(t)$ como:

$$Y(t) = T(t) + S(t) + C(t) + e(t)$$

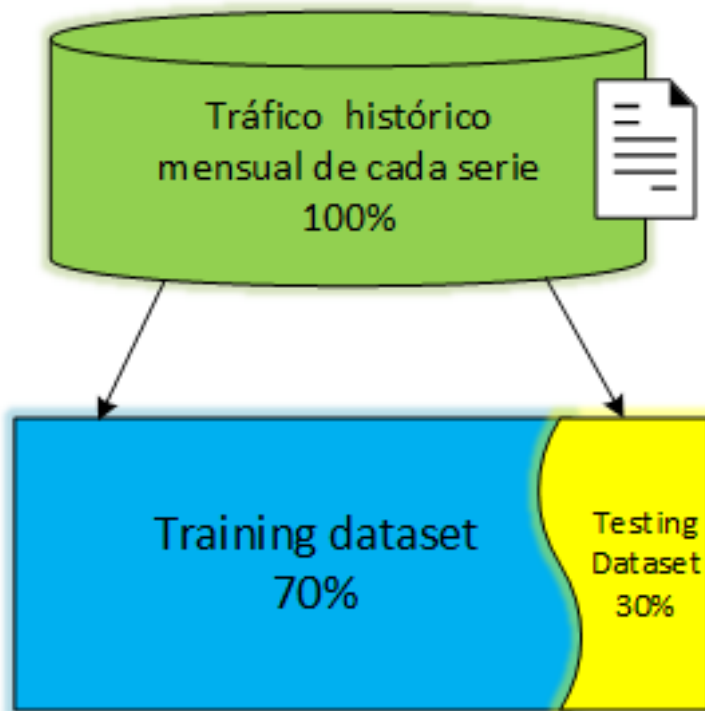
donde:

para tiempo continuo y una serie de tiempo discreto Y_t como:

$$Y_t = T_t + S_t + C_t + e_t$$

- $T(t)$: Tendencia
- $S(t)$: Variación Estacional
- $C(t)$: Componente Cíclica
- $e(t)$: Componente aleatoria

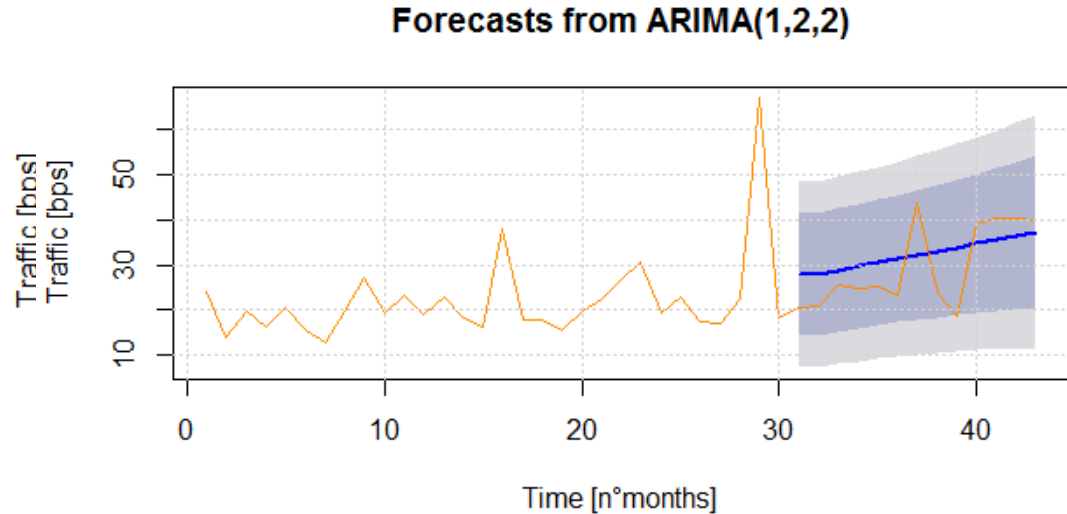
Training and Testing



- Hace falta fraccionar el dataset en dos partes: **training y test**
- En general es útil que las proporciones sean representativas.
- La cantidad y calidad de los datos es un factor siempre presente.
- No todos los algoritmos estadísticos admiten paralelismo.
- Otro factor importante es el nivel de ajuste (**sub fitting** vs. **overfitting**)

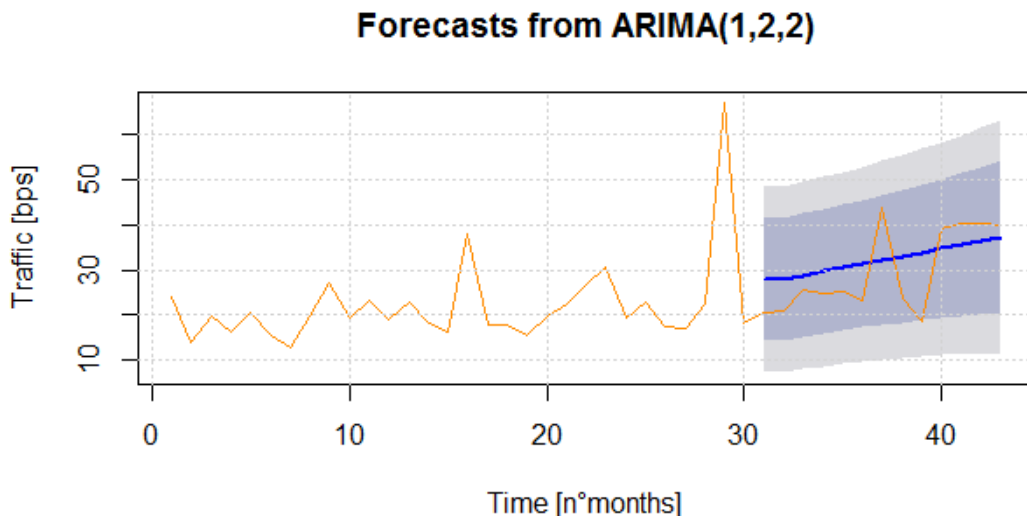
Resultados usando ARIMA

- En azul se grafica la **tendencia**
- En gris oscuro el intervalo de confianza del 95%
- En gris claro el intervalo de 90%
- En naranja el fragmento de testing de la serie



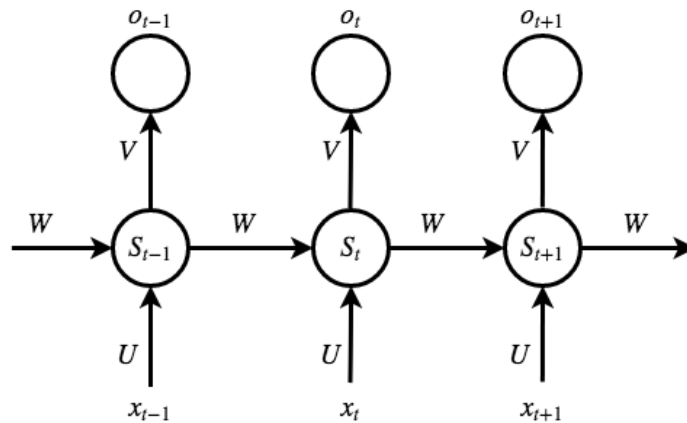
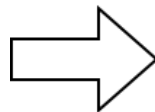
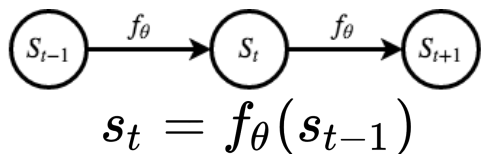
Compromisos

- Los intervalos de confianza pueden ser muy amplios en términos absolutos.
- Si es una variable económica el desvío puede ser demasiado significativo.



Alternativa usando Redes Neuronales

- Usamos redes basadas en estados
- este tipo de redes se llaman Redes Neuronales Recurrentes (RNN)

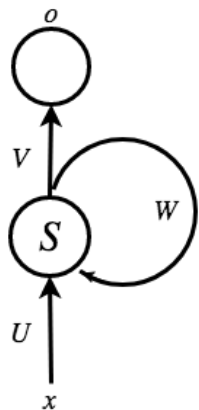
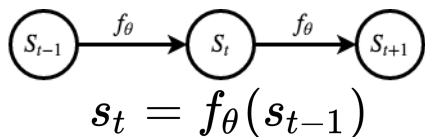


Vanilla Recurrent Neural Network

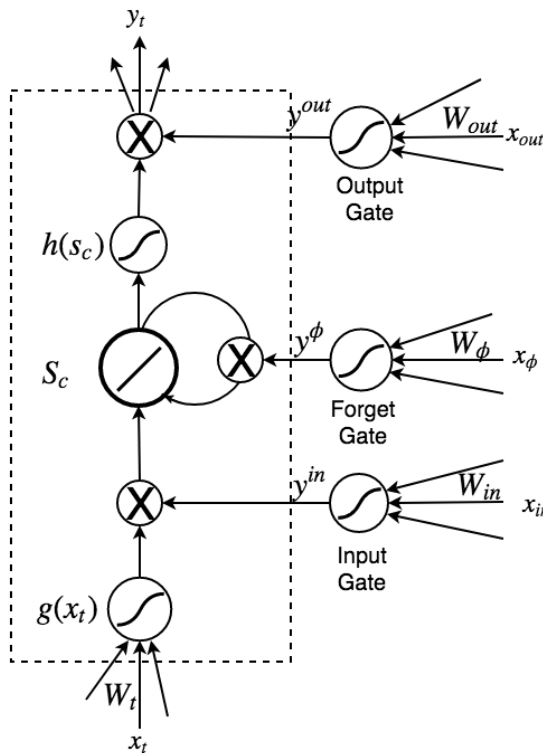
- se puede usar la notación de estados
- se suele usar la topología desplegada

Redes Neuronales LSTM

Notación



- x : entrada
- o : objetivo
- S : estado
- U, V, W : matrices



Celda

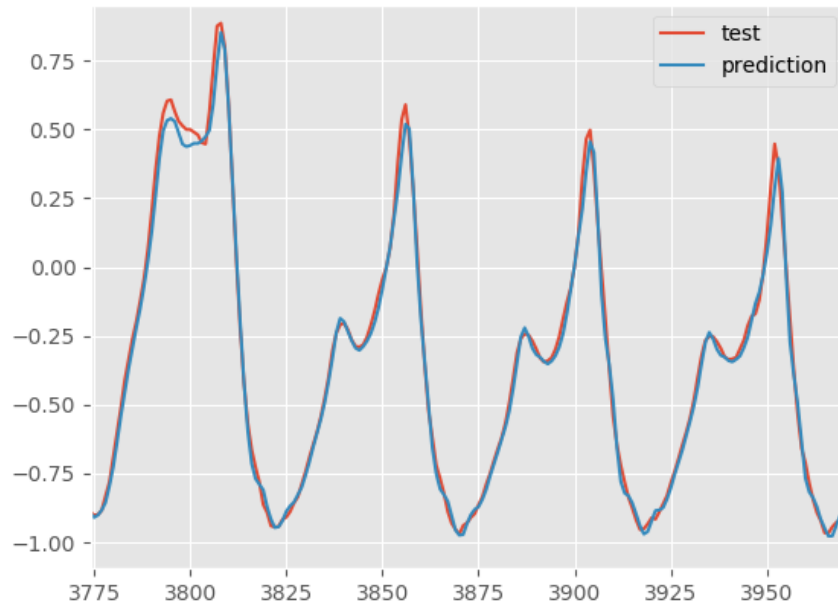
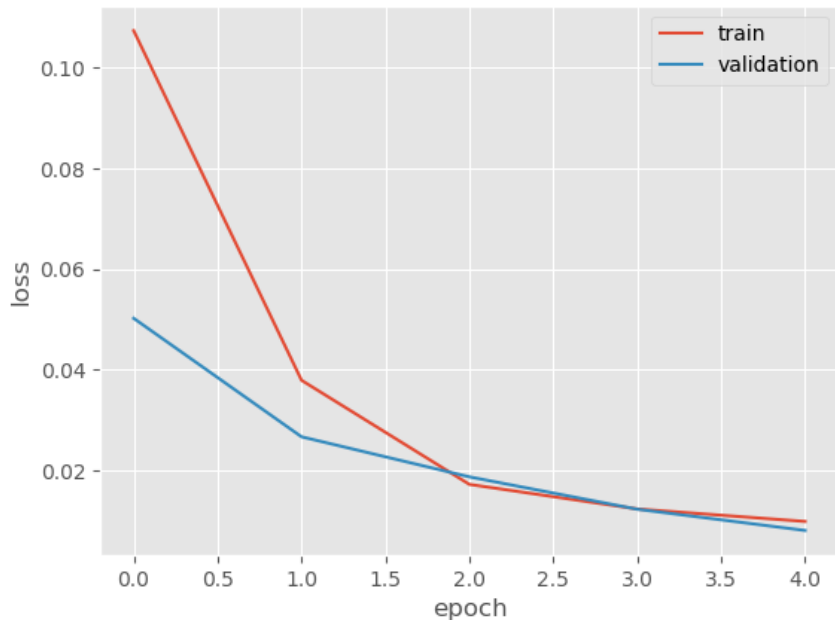
Contiene una entrada y una salida mas tres entradas de control:

- Input gate
- Forget gate
- Output gate

La función core es lineal

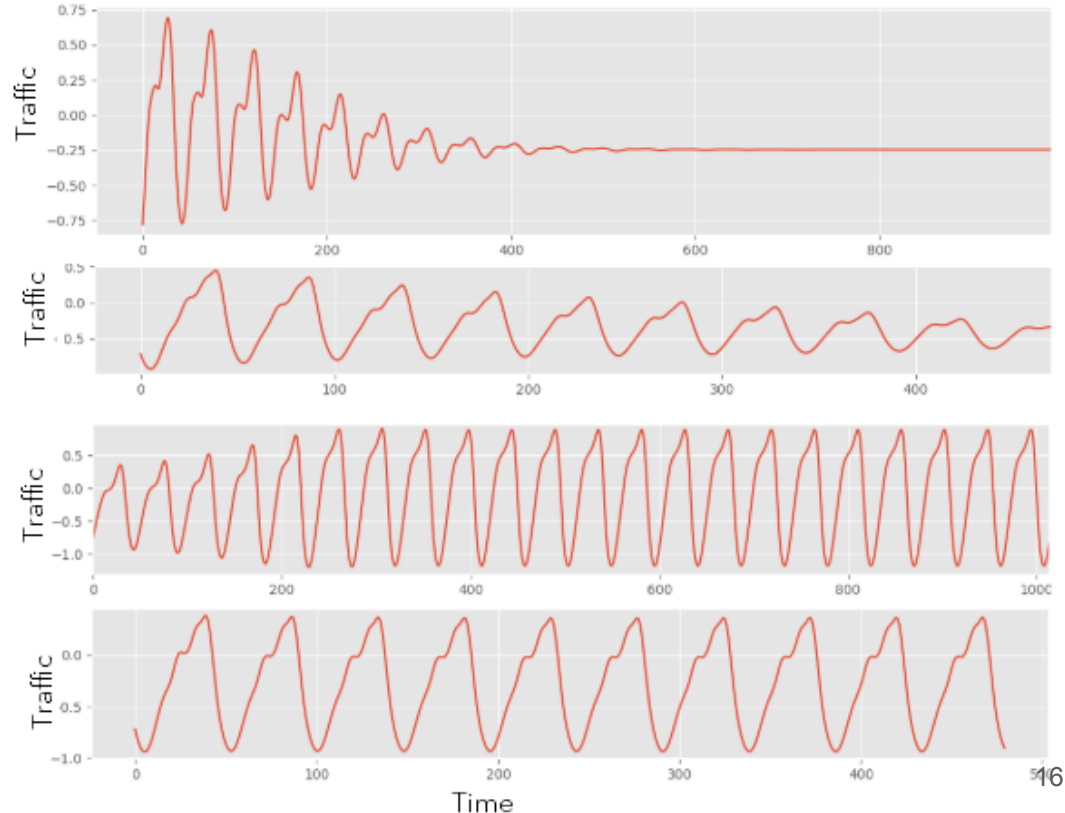
LSTM RNN: entrenamiento y test

Model train vs. validation loss



Predicciones usando ventana deslizante

- La técnica de ventana deslizante consiste en ir realimentando la serie predecida intervalo a intervalo.
- El largo de la ventana, la densidad de datos, la cantidad de períodos muestrados son factores de diseño.



Figuras de mérito

- Como figuras de mérito de los modelos elegimos algunas métricas que sirvan para hacer los modelos comparables.
- El error cuadrático medio, el máximo error absoluto, mínimo error porcentual (absoluto), entre otros.

Métricas de Error Guía Rápida

	Selección de Modelo	Calibración	Toma de Decisiones
MAD			
ET			
RMSE			
MPE			
MAPE			
SFE			
BIAS			
GONA			

Resultados

Training set	ARIMA (p,d,q)	RMSE	MAE	MPE (%)	MAPE (%)	LSTM (inputs,batch, epochs)	RMSE	MAE	MAPE
CDN Google	1,2,1	13,08	10,35	1,16	2,40	144, 144, 5	8.7×10^{-4}	9.2×10^{-3}	< 0.00%
CND Netflix	1,2,1	27,70	19,76	0,22	2,86	144, 144, 5	3.3×10^{-3}	3.1×10^{-3}	< 0.00%
CDN Verizon	1,2,2	9,29	5,28	-1,72	20,63	336(7d), 336,5	2.5×10^{-4}	2.3×10^{-2}	< 0.0%
CDN Akamai	2,2,2	15,25	10,79	-1,54	10,79	336, 336,5	5.7×10^{-4}	6.2×10^{-2}	< 0.0%

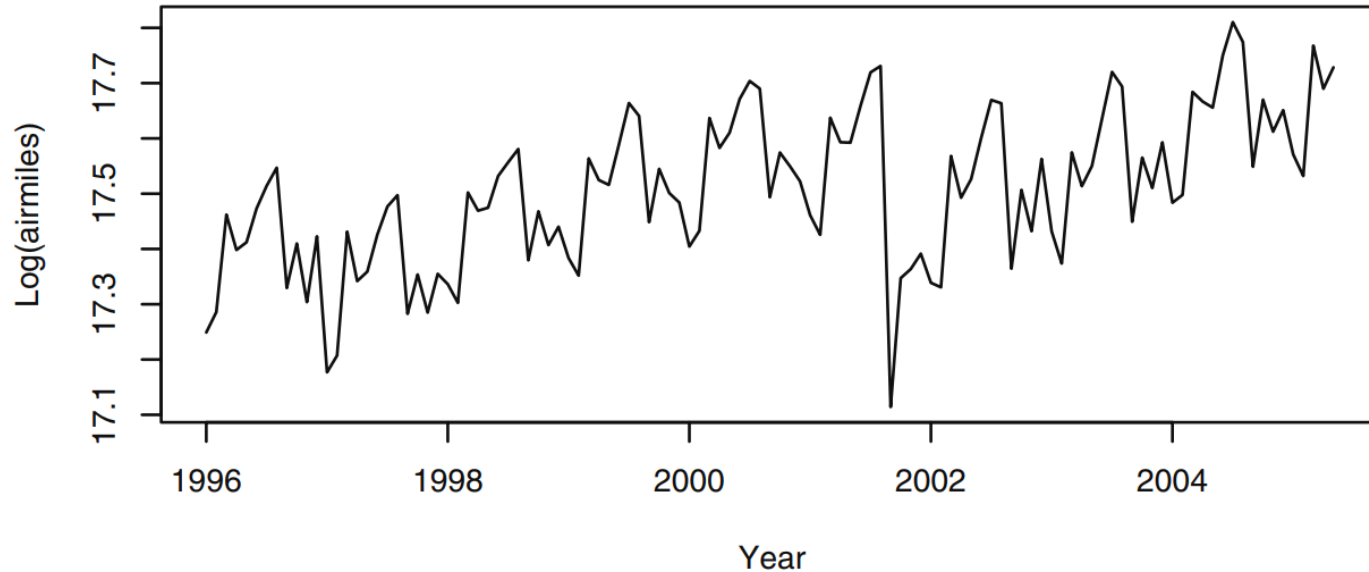
- Componentes estacionales requieren datos interanuales
- Para este dataset:
 - LSTM performa bien en el muy corto plazo, capturando perfectamente los ciclos periódicos
 - ARIMA ofrece tendencias explicables a largo plazo
- Compromiso entre cantidad de datos y método utilizado
- Posibles mejoras extendiendo el tamaño de la red LSTM

Modelos de Regresión

Análisis de intervenciones

Análisis de intervenciones

Exhibit 11.1 Monthly U.S. Airline Miles: January 1996 through May 2005



Análisis de intervenciones

Consideremos el caso de una única intervención:

$$Y_t = \boxed{m_t} + \boxed{N_t}$$

Intervención:
cambio en la media

Proceso natural:
serie de tiempo que puede ser
modelada como un (S)ARIMA

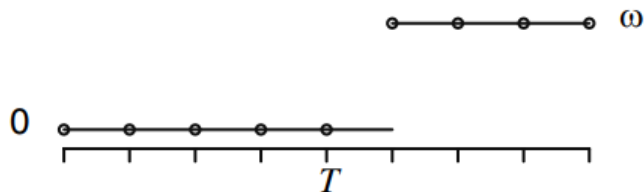
Si la intervención ocurre en un instante T , se asume que $m_t = 0$ para $t < T$ y a $\{Y_t\}_{t < T}$ se la conoce como *datos preintervención* y puede ser usada para modelar N_t .

La estimación de los parámetros de m_t se puede realizar por MV o el enfoque de modelo de estados ya presentados.

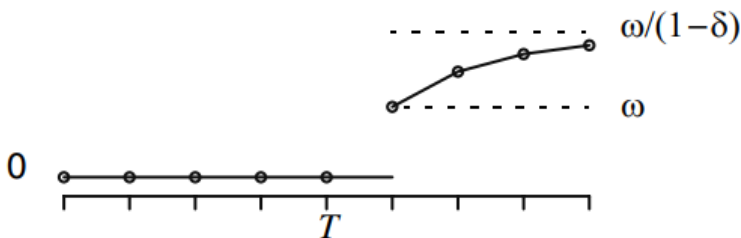
Análisis de intervenciones

Exhibit 11.3 Some Common Models for Step Response Interventions
(All are shown with a delay of 1 time unit)

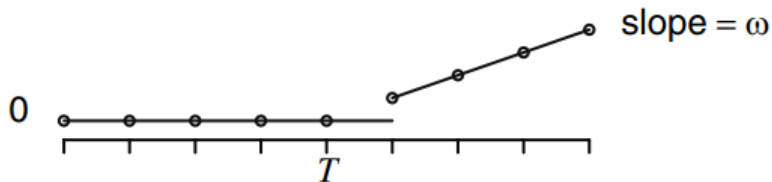
(a) $S_t^T \equiv \mathbf{I}\{t > T\} \quad \omega B S_t^{(T)}$



(b) $\frac{\omega B}{1 - \delta B} S_t^{(T)}$



(c) $\frac{\omega B}{1 - B} S_t^{(T)}$



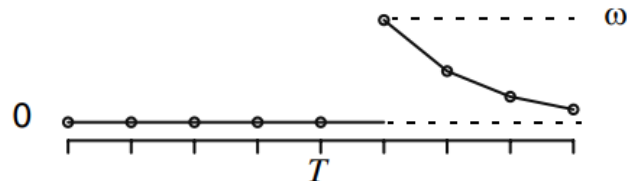
Análisis de intervenciones

Exhibit 11.4 Some Common Models for Pulse Response Interventions
(All are shown with a delay of 1 time unit)

$$[P_t^{(T)} \equiv \delta_T(t)]$$

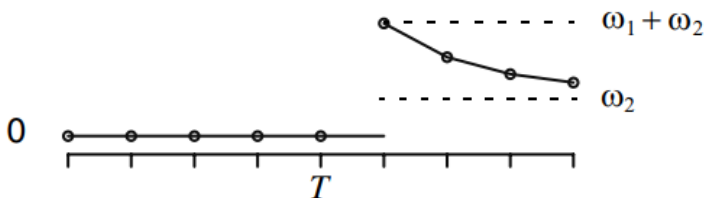
(a)

$$\frac{\omega B}{1 - \delta B} P_t^{(T)}$$



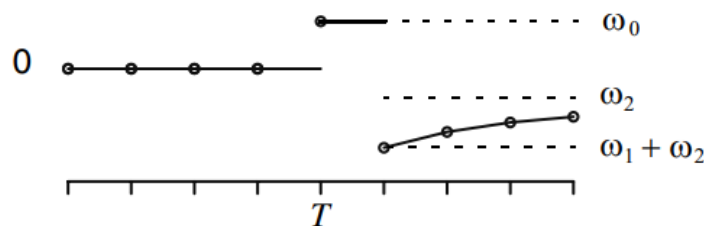
(b)

$$\left[\frac{\omega_1 B}{1 - \delta B} + \frac{\omega_2 B}{1 - B} \right] P_t^{(T)}$$



(c)

$$\left[\omega_0 + \frac{\omega_1 B}{1 - \delta B} + \frac{\omega_2 B}{1 - B} \right] P_t^{(T)}$$

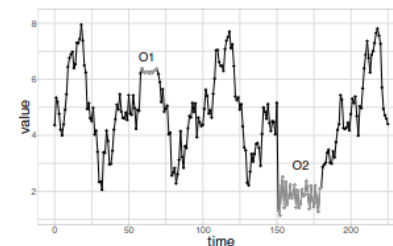
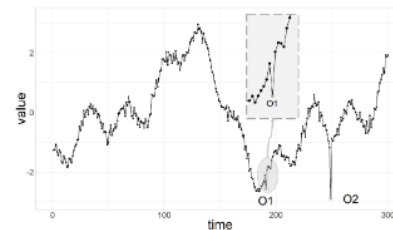
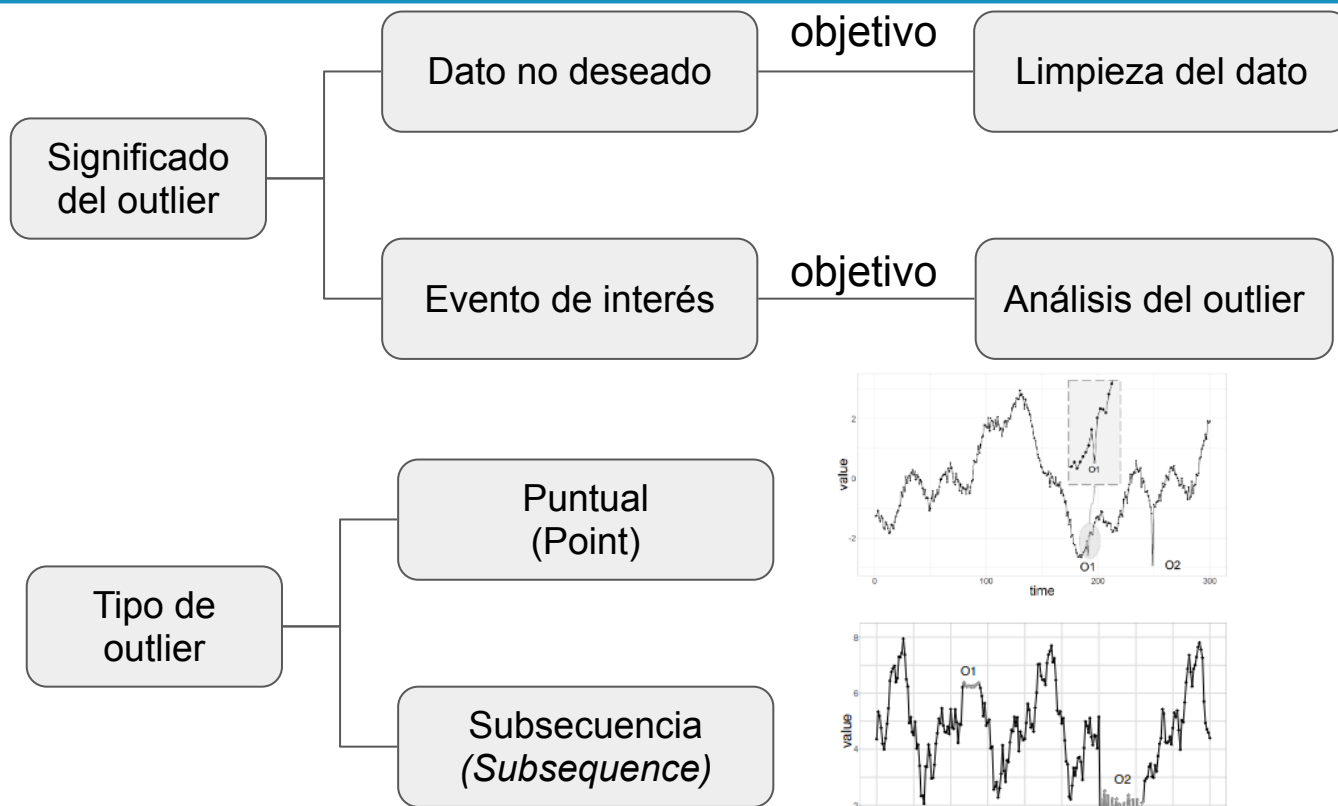


Ejemplo

Ver notebook: [intervention_analysis.ipynb](#)

Análisis de outliers

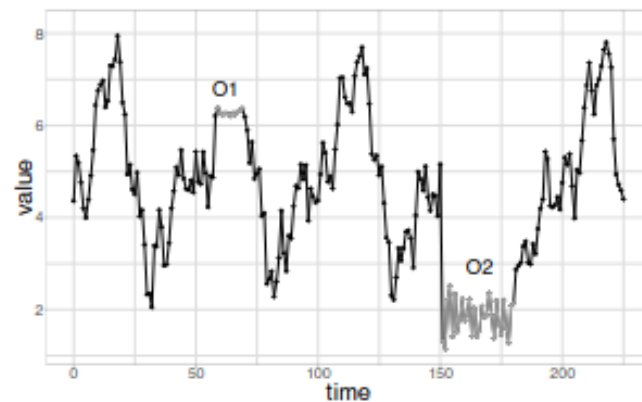
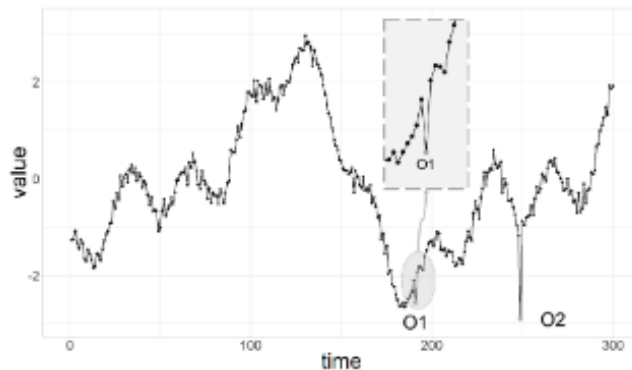
Tipos de outliers



Tipos puntuales

Outliers puntuales: Dato que se comporta de manera inusual en un instante de tiempo específico, comparado con la serie total, o sus puntos vecinos. Son los más comunes.

Outliers de subsecuencia: corresponde a puntos que en conjunto se comportan de forma peculiar respecto del resto de la serie, aunque cada observación de forma individual no resulta necesariamente un outlier.

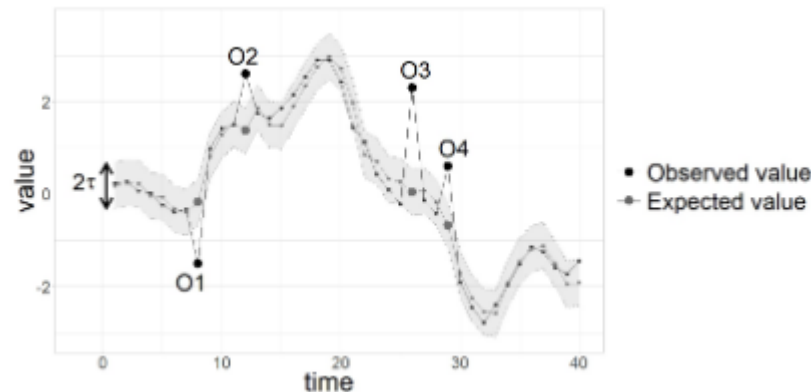


Análisis de Outliers puntuales:

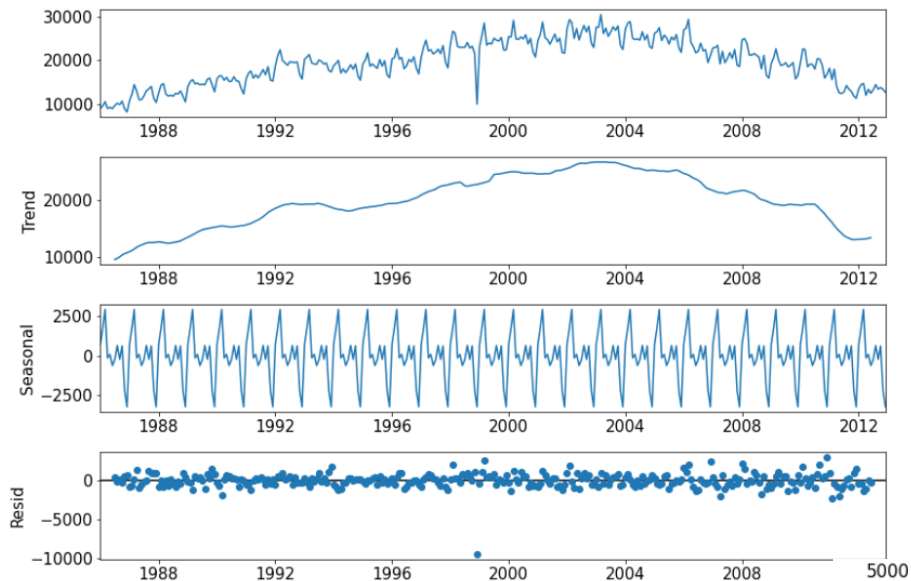
Podemos definir un outlier como un punto que se desvía significativamente de su valor esperado. $|x_t - \hat{x}_t| > \tau$

Hay distintas formas de determinar ese valor esperado y el umbral.

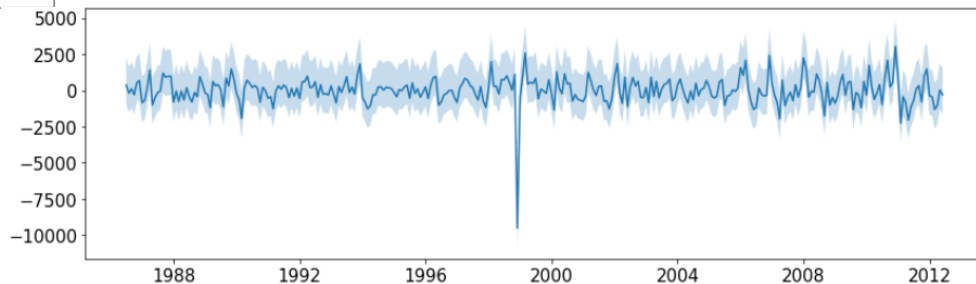
- Modelos constantes a tramos, donde se toman estadísticas como mediana o MAD (mean absolute deviation)
- Identificación de puntos poco probables bajo un cierto modelo o distribución
- Analizando residuos (ej. usando la descomposición STL, o directamente analizando residuos del ARIMA)
- Etc.



Detección de outliers usando la descomposición STL



```
fig = plt.figure(figsize=(14,4))
q975 = residuals.quantile(.975)
q025 = residuals.quantile(.025)
plt.plot(residuals)
plt.fill_between(x=residuals.index,
                 y1=residuals + q025,
                 y2=residuals + q975,
                 alpha=0.25
                )
```



Cómo tratar los outliers

- Usar la media en una ventana para suavizar el outlier
- Modelar el dato basándose en el problema de estudio
- Quitar el punto

Ejemplo

Ver notebook: [outlier_detection.ipynb](#)

Modelos de suavizado exponencial

Suavizado exponencial

El método de suavizado exponencial se suele ajustar bien a series estacionales. Tiene la ventaja de que es simple, requiere menos datos que los modelos ARIMA para predecir y responde bien a cambios en tendencia y estacionalidad de corto plazo.

Los pronósticos obtenidos con este método son promedios ponderados de observaciones pasadas. Los pesos decaen exponencialmente, por lo tanto las últimas observaciones serán las de mayor peso.

Vamos a usar una notación de componentes para describir estos modelos.

Suavizado exponencial simple

Si y_t es la serie de tiempo, la predicción a un paso es:

$$\hat{y}_{t+1|t} = \alpha y_t + \alpha(1 - \alpha)y_{t-1} + \alpha(1 - \alpha)^2 y_{t-2} + \dots$$

donde alpha es el parámetro de suavizado ($0 < \alpha < 1$). Hace falta definir un valor para el momento inicial, es decir $t=1$ al que llamamos n_0 :

$$\hat{y}_{t+1|t} = \alpha y_t + \alpha(1 - \alpha)\hat{y}_{t|t-1}$$

$$\hat{y}_{t+1|t} = \sum_{j=0}^{t-1} \alpha(1 - \alpha)^j y_{t-j} + (1 - \alpha)^t n_0$$

Suavizado exponencial simple

Si usamos la notación de componentes, este modelo simple se puede describir a partir de una componente de *Nivel*:

$$\text{Pronóstico} \quad \hat{y}_{t+h|t} = n_t$$

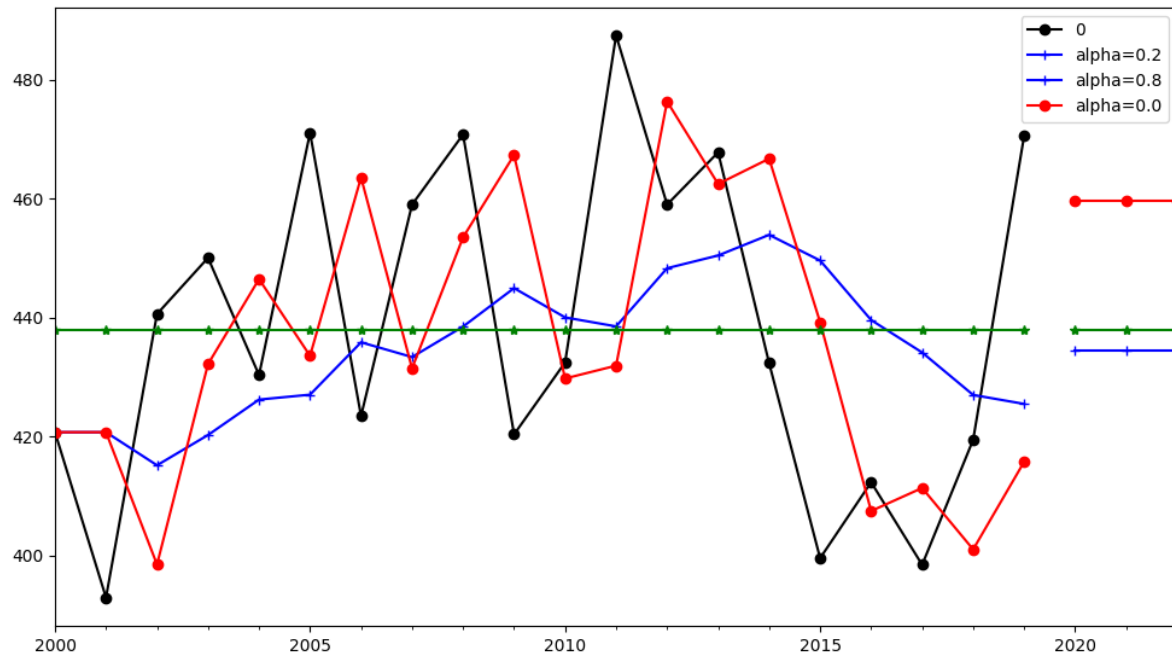
$$\text{Nivel} \quad n_t = \alpha y_t + (1 - \alpha)n_{t-1}$$

En python estos modelos están incluidos en statsmodels.

```
from statsmodels.tsa.api import SimpleExpSmoothing

ins1 = SimpleExpSmoothing(data).fit(smoothing_level=0.2, optimized=False)
```

Suavizado exponencial simple



Suavizado exponencial lineal de Holt

Esta extensión permite trabajar con series que tengan tendencia lineal. Con b_t la pendiente de la tendencia y β el parámetro de suavizado para la tendencia hay una expresión para el pronóstico y dos ecuaciones de componentes:

$$\text{Pronóstico} \quad \hat{y}_{t+h|t} = n_t + hb_t$$

$$\text{Nivel} \quad n_t = \alpha y_t + (1 - \alpha)(n_{t-1} + b_{t-1})$$

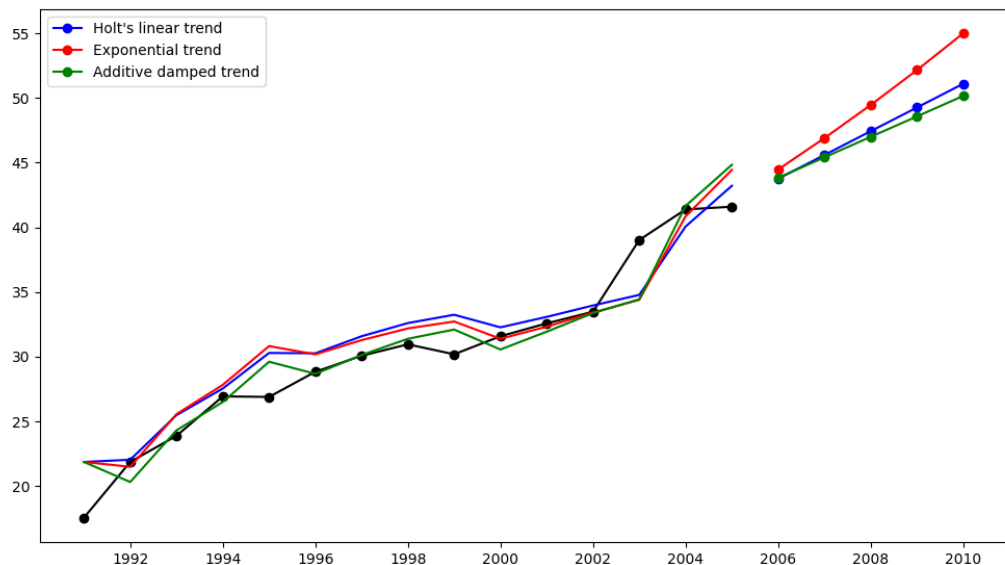
$$\text{Tendencia} \quad b_t = \beta(n_t - n_{t-1}) + (1 - \beta)b_{t-1}$$

En este modelo la tendencia es constante indefinidamente. Para otros casos se define un factor de damping o amortiguación.

Suavizado exponencial lineal de Holt

```
from statsmodels.tsa.api import ExponentialSmoothing, SimpleExpSmoothing, Holt
```

```
fit1 = Holt(air).fit(smoothing_level=0.8, smoothing_slope=0.2, optimized=False)
fcast1 = fit1.forecast(5).rename("Holt's linear trend")
fit2 = Holt(air, exponential=True).fit(smoothing_level=0.8, smoothing_slope=0.4,
                                       optimized=False)
fcast2 = fit2.forecast(5).rename("Exponential trend")
```



Suavizado exponencial de tendencia amortiguada

delta es el parámetro de amortiguación, definido entre (0,1).

$$\textit{Pronóstico} \quad \hat{y}_{t+h|t} = n_t + (\delta + \delta^2 + \dots + \delta^h)b_t$$

$$\textit{Nivel} \quad n_t = \alpha y_t + (1 - \alpha)(n_{t-1} + \delta b_{t-1})$$

$$\textit{Tendencia} \quad b_t = \beta(n_t - n_{t-1}) + (1 - \beta)\delta b_{t-1}$$

Holt - Winters

Holt-Winters resulta una generalización de estos métodos para poder trabajar con series estacionales, agregando una ecuación de suavizado para la componente estacional e_t . Se presenta el método aditivo y el multiplicativo:

Método aditivo:

$$\text{Pronóstico} \quad \hat{y}_{t+h|t} = n_t + hb_t + e_{t+h-m(k+1)}$$

$$\text{Nivel} \quad n_t = \alpha(y_t - e_{t-m} + (1 - \alpha)(n_{t-1} + b_{t-1}))$$

$$\text{Tendencia} \quad b_t = \beta(n_t - n_{t-1}) + (1 - \beta)b_{t-1}$$

$$\text{Estacionalidad} \quad e_t = \gamma(y_t - n_{t-1} - b_{t-1}) + (1 - \gamma)e_{t-m}$$

Holt - Winters

gamma es el parámetro de estacionalidad (0,1), m es el período estacional, por ejemplo 12 meses y $k = \lfloor (h-1)/k \rfloor$ es un índice.

Método multiplicativo:

$$\text{Pronóstico} \quad \hat{y}_{t+h|t} = (n_t + hb_t)e_{t+h-m(k+1)}$$

$$\text{Nivel} \quad n_t = \alpha \frac{y_t}{e_{t-m}} + (1 - \alpha)(n_{t-1} + b_{t-1})$$

$$\text{Tendencia} \quad b_t = \beta(n_t - n_{t-1}) + (1 - \beta)b_{t-1}$$

$$\text{Estacionalidad} \quad e_t = \gamma \frac{y_t}{n_{t-1} + b_{t-1}} + (1 - \gamma)e_{t-m}$$

Holt - Winters

Agregando el factor de damping a este modelo se tiene que:

$$\text{Pronóstico} \quad \hat{y}_{t+h|t} = [n_t + (\delta + \delta^2 + \dots + \delta^h)b_t]e_{t+h-m(k+1)}$$

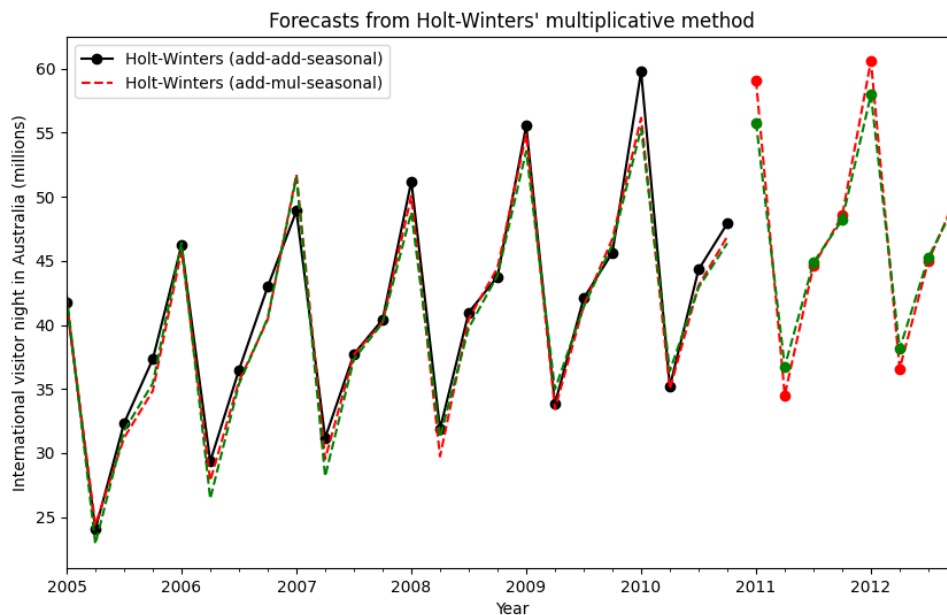
$$\text{Nivel} \quad n_t = \alpha \frac{y_t}{e_{t-m}} + (1 - \alpha)(n_{t-1} + \delta b_{t-1})$$

$$\text{Tendencia} \quad b_t = \beta(n_t - n_{t-1}) + (1 - \beta)\delta b_{t-1}$$

$$\text{Estacionalidad} \quad e_t = \gamma \frac{y_t}{n_{t-1} + \delta b_{t-1}} + (1 - \gamma)e_{t-m}$$

Holt - Winters

```
fit1 = ExponentialSmoothing(  
    aust,  
    seasonal='mul',  
    seasonal_periods=12  
)  
.fit()  
fit2 = ExponentialSmoothing(  
    aust,  
    seasonal='add',  
    seasonal_periods=12  
)  
.fit()
```



Revisión de bibliografía

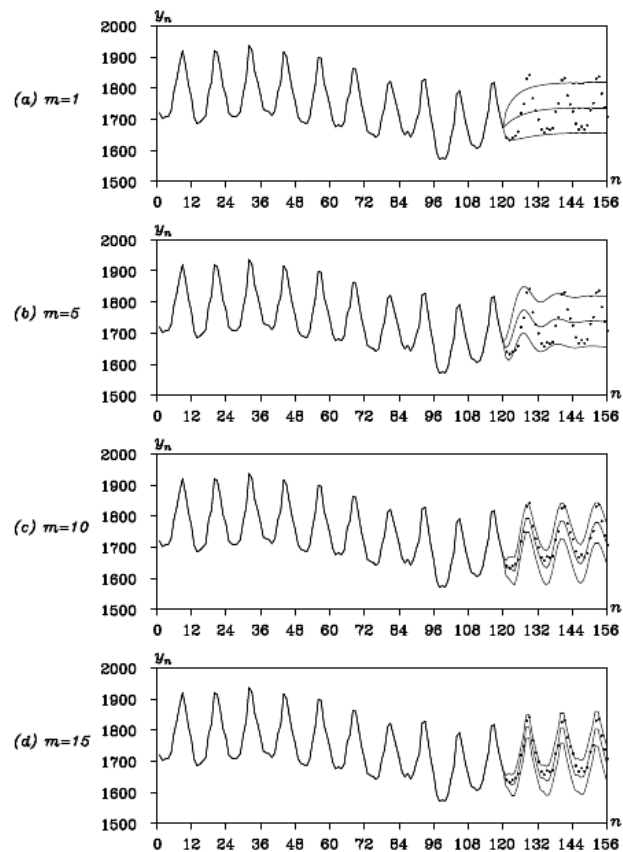


Figure 9.2 Increasing horizon predictive distributions (bold line: mean, thin line: \pm (standard deviation) and \circ : observed value). Orders of the AR models are 1, 5, 10 and 15, respectively.

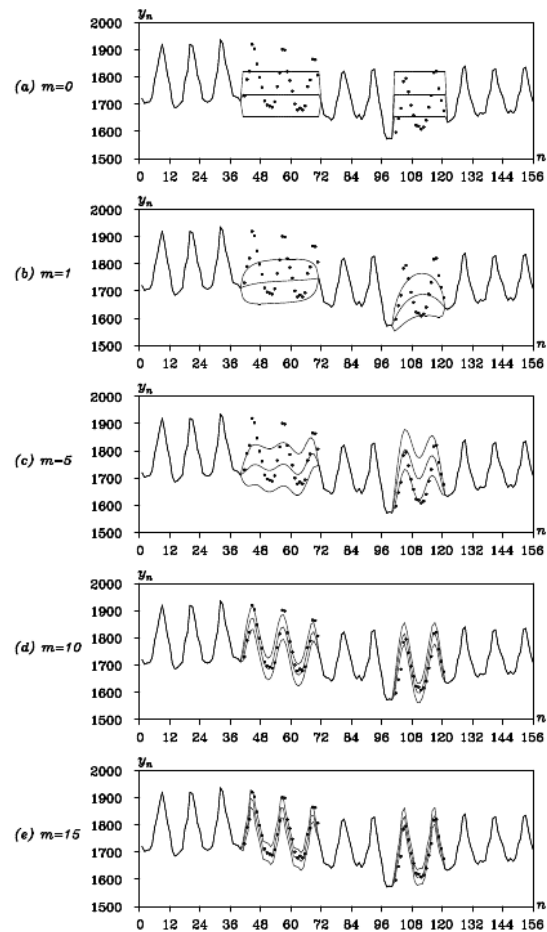


Figure 9.3 Interpolation of missing values (bold line: mean, thin line: \pm (standard deviation) and \circ : observed value). Orders of the AR models are 0, 1, 5, 10 and 15.

Appendix D: The Backshift Operator

Many other books and much of the time series literature use what is called the **backshift operator** to express and manipulate ARIMA models. The backshift operator, denoted B , operates on the time index of a series and shifts time back one time unit to form a new series.[†] In particular,

$$BY_t = Y_{t-1}$$

The backshift operator is linear since for any constants a , b , and c and series Y_t and X_t , it is easy to see that

$$B(aY_t + bX_t + c) = aBY_t + bBX_t + c$$

Consider now the MA(1) model. In terms of B , we can write

$$\begin{aligned} Y_t &= e_t - \theta e_{t-1} = e_t - \theta B e_t = (1 - \theta B) e_t \\ &= \theta(B) e_t \end{aligned}$$

where $\theta(B)$ is the MA characteristic polynomial “evaluated” at B .

Since BY_t is itself a time series, it is meaningful to consider BBY_t . But clearly $BBY_t = BY_{t-1} = Y_{t-2}$, and we can write

$$B^2 Y_t = Y_{t-2}$$

More generally, we have

$$B^m Y_t = Y_{t-m}$$

for any positive integer m . For a general MA(q) model, we can then write

$$\begin{aligned} Y_t &= e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} \\ &= e_t - \theta_1 B e_t - \theta_2 B^2 e_t - \cdots - \theta_q B^q e_t \\ &= (1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q) e_t \end{aligned}$$

or

$$Y_t = \theta(B) e_t$$

where, again, $\theta(B)$ is the MA characteristic polynomial evaluated at B .

For autoregressive models AR(p), we first move all of the terms involving Y to the left-hand side

$$Y_t - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \cdots - \phi_p Y_{t-p} = e_t$$

and then write

$$Y_t - \phi_1 B Y_t - \phi_2 B^2 Y_t - \cdots - \phi_p B^p Y_t = e_t$$

or

[†] Sometimes B is called a **Lag operator**.

Appendix D: The Backshift Operator

$$(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p) Y_t = e_t$$

which can be expressed as

$$\phi(B) Y_t = e_t$$

where $\phi(B)$ is the AR characteristic polynomial evaluated at B .

Combining the two, the general ARMA(p, q) model may be written compactly as

$$\phi(B) Y_t = \theta(B) e_t$$

Differencing can also be conveniently expressed in terms of B . We have

$$\begin{aligned} \nabla Y_t &= Y_t - Y_{t-1} = Y_t - B Y_t \\ &= (1 - B) Y_t \end{aligned}$$

with second differences given by

$$\nabla^2 Y_t = (1 - B)^2 Y_t$$

Effectively, $\nabla = 1 - B$ and $\nabla^2 = (1 - B)^2$.

The general ARIMA(p, d, q) model is expressed concisely as

$$\phi(B)(1 - B)^d Y_t = \theta(B) e_t$$

In the literature, one must carefully distinguish from the context the use of B as a backshift operator and its use as an ordinary real (or complex) variable. For example, the stationarity condition is frequently given by stating that the roots of $\phi(B) = 0$ must be greater than 1 in absolute value or, equivalently, must lie outside the unit circle in the complex plane. Here B is to be treated as a dummy variable in an equation rather than as the backshift operator.

**Ref.: Introduction to Time Series Modeling,
Genshiro Kitagawa**

Bibliografía

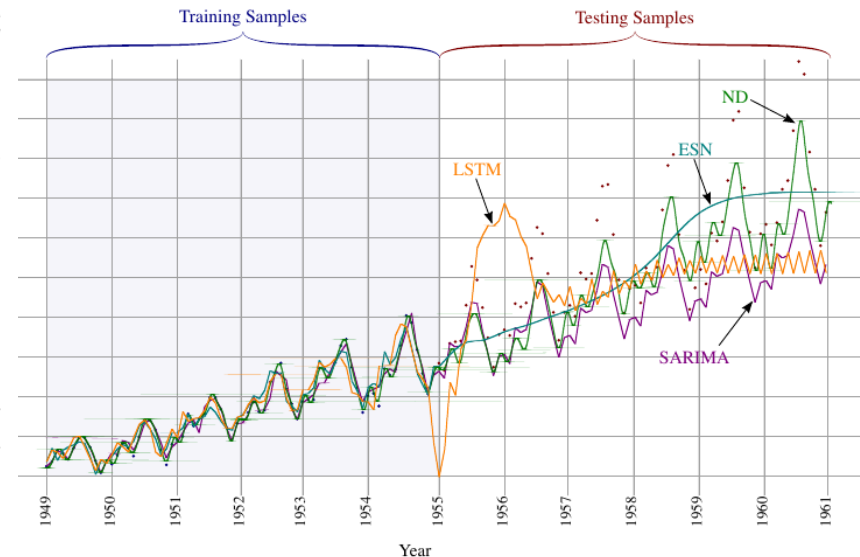
1. [Godfrey2017] Neural Decomposition of Time-Series Data
2. [Chrobok2005] Theory and Application of Advanced Traffic Forecast Methods
3. [Hochreiter1997] LONG SHORT-TERM MEMORY
4. [Gers2000] Learning to Forget: Continual Prediction with LSTM
5. Decompose: <https://towardsdatascience.com/different-types-of-time-series-decomposition-396c09f92693>
6. [Detección de outliers](#)
7. [Tutorial de detección de outliers](#)
8. RPIC2019, Exponential Smoothing: Ref.: <https://otexts.com/fpp2/expsmooth.html>

Neural decomposition

Our model is defined as follows. Let each a_k represent an amplitude, each w_k represent a frequency, and each ϕ_k represent a phase shift. Let t refer to time, and let $g(t)$ be an augmentation function that represents the nonperiodic components of the signal. Our model is defined as

$$x(t) = \sum_{k=1}^N (a_k \cdot \sin(w_k t + \phi_k)) + g(t). \quad (2)$$

Note that the lower index of the sum has changed from $k = 0$ in the DFT to $k = 1$ in our model. This is because ND can account for bias in the augmentation function $g(t)$. Therefore, only N sinusoids are required rather than $N + 2$.



1. [Godfrey2017] Neural Decomposition of Time-Series Data

Advanced Traffic Forecast Methods

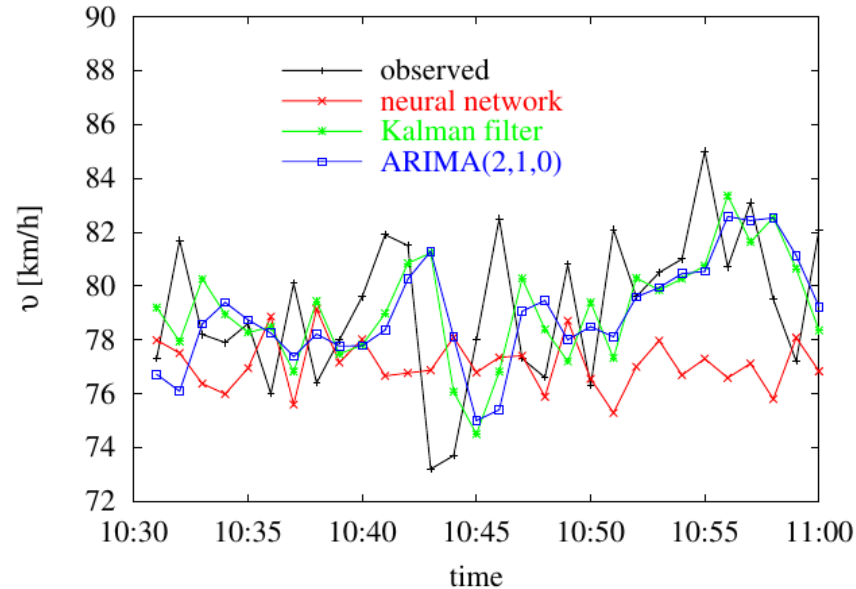
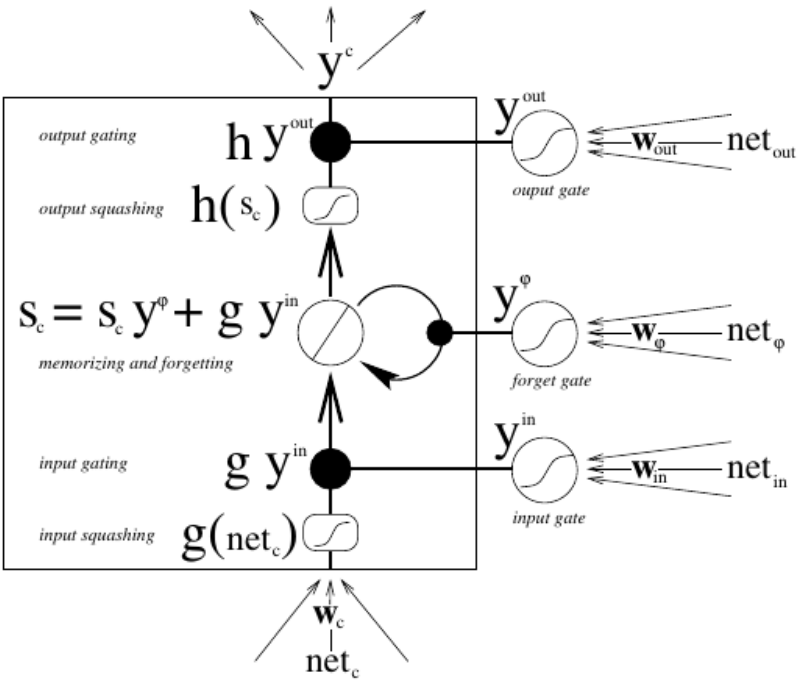
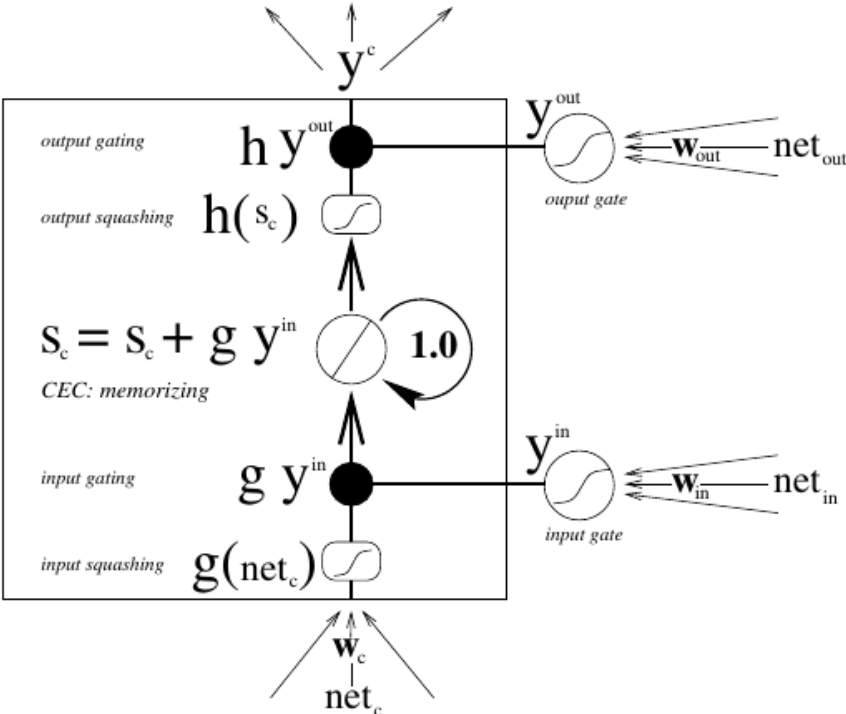


Fig. 3.9. Comparison of a four layer feedforward neural network, a Kalman filter, and an ARIMA(2,1,0) model as one-step-ahead velocity forecast methods by Lee and Choi [1998]. It seems that the velocity fluctuates around a mean value and that no model stands out in forecasting those fluctuations.

Ref.: Theory and Application of Advanced Traffic Forecast Methods, Roland Chrobok, 2005



Consultas