Défis en Intelligence Artificielle

Défi 3 : L'IA pour l'analyse et la prévision de séries temporelles (II/III)

Souhaib Ben Taieb

University of Mons

December 15, 2022

UMONS
Université de Mons

Faculté
des Sciences

Overview

Last week recap

Statistical forecasting methods

- AR, MA and ARIMA models
- Model estimation, validation and selection
- (S)ARIMA models
- Auto.arima

AI forecasting models

- Challenges in training AI forecasting models (recap)
- Time-series (cross-)validation
- Some multi-step forecasting strategies
- (Deep) Neural Networks

Outline

Last week recap

Statistical forecasting methods

AI forecasting models

Last week recap

Course organization

Introduction to time series forecasting

Concepts and tools for time series data

- Time series patterns
- Time series plots
- The autocorrelation function
- Stationarity
- Transformations
- Time series decomposition

Methods and tools for time series forecasting

- Overview of forecasting methods
- Challenges in training AI models for forecasting
- Some simple forecasting methods
- Residual diagnostics
- Evaluating forecast accuracy

Overview of forecasting methods

- Simple forecasting methods
  - Often used as (naive) baselines
- Statistical/time series models
  - Good performance for many real-world applications
  - Often used as (stronger) baselines for AI models
- AI/Machine learning methods
  - Time series forecasting can be reduced to a regression problem
  - Use any AI learning algorithm for regression
  - Specific AI architectures for sequential data

Outline

Outline

Last week recap

Statistical forecasting methods

- AR, MA and ARIMA models
- Model estimation, validation and selection
- (S)ARIMA models
- Auto.arima

AI forecasting models

- Challenges in training AI forecasting models (recap)
- Time-series (cross-)validation
- Some multi-step forecasting strategies
- (Deep) Neural Networks

Autoregressive (AR) models

An **autoregressive (AR) model** of order p can be written as

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t,$$

where ε_t is white noise.

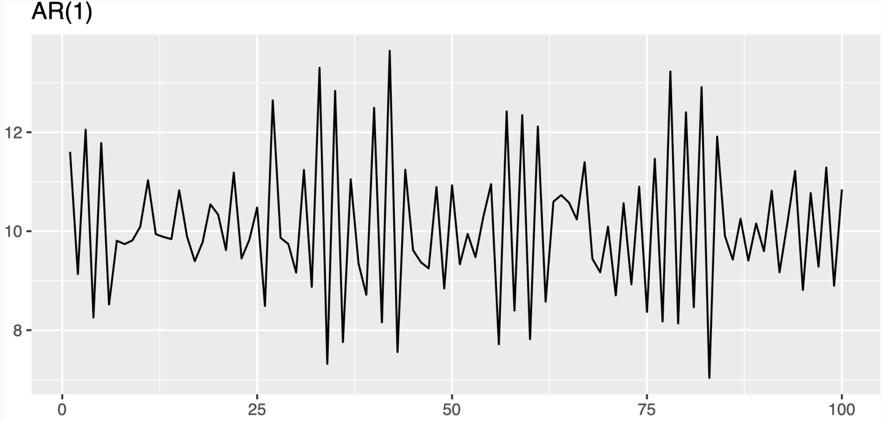
This is like a **multiple regression** but with **lagged values** of y_t as predictors./inputs.

Some **constraints** on the values of the coefficients are required to obtain a **stationary** model.

AR(1) example

$$y_t = 18 - 0.8y_{t-1} + \varepsilon_t$$

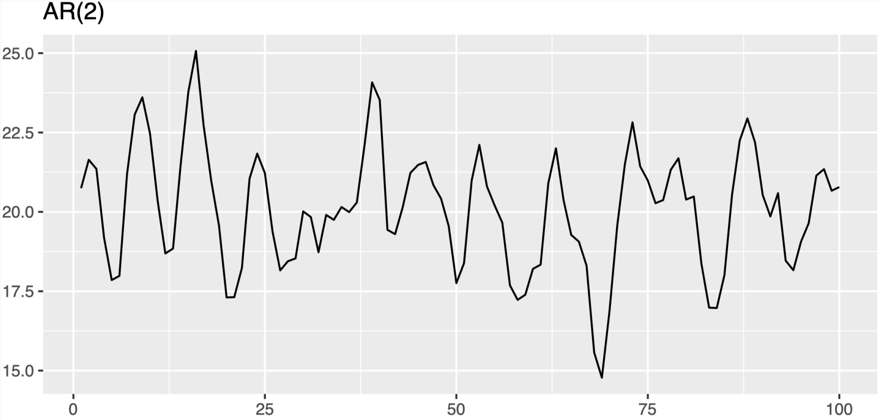
$$\varepsilon_t \sim N(0, 1), \quad T = 100.$$



AR(2) example

$$y_t = 8 + 1.3y_{t-1} - 0.7y_{t-2} + \varepsilon_t$$

$$\varepsilon_t \sim N(0, 1), \qquad T = 100.$$



Moving Average (MA) models

A **moving average** (MA) model order p can be written as

$$y_t = c + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \cdots + \theta_q\varepsilon_{t-q},$$

where ε_t is white noise.

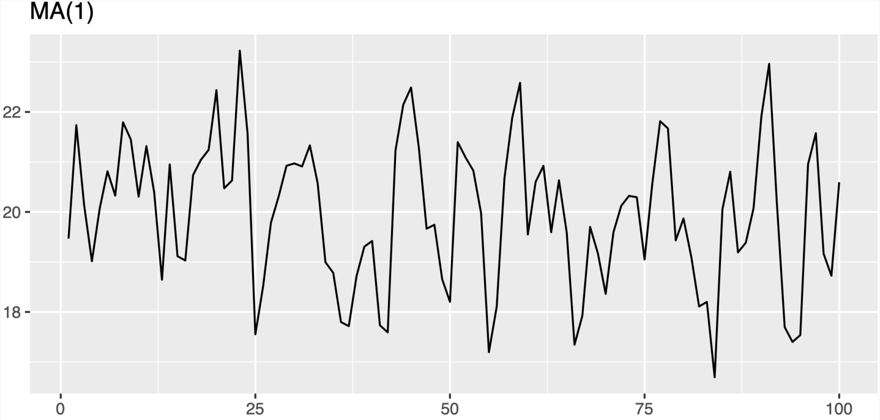
Rather than using past values of the forecast variable in a regression, an MA model uses **past forecast errors** in a regression-like model.

Some **constraints** on the values of the coefficients are required to obtain a **stationary** model.

MA(1) example

$$y_t = 20 + \varepsilon_t + 0.8\varepsilon_{t-1}$$

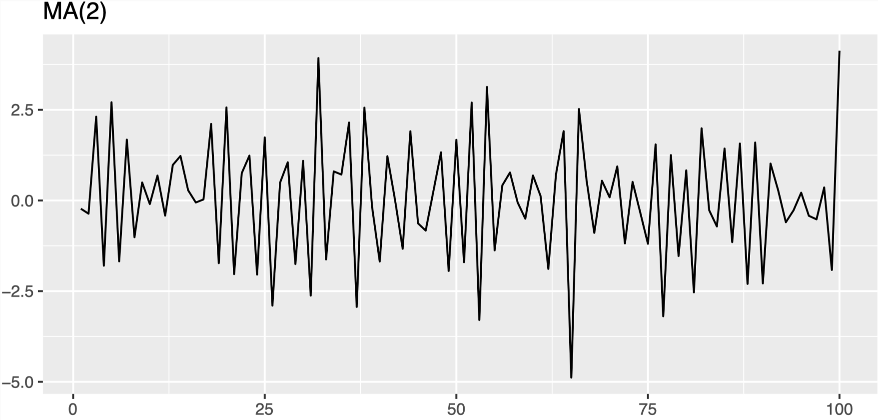
$$\varepsilon_t \sim N(0, 1), \quad T = 100.$$



MA(2) example

$$y_t = \varepsilon_t - \varepsilon_{t-1} + 0.8\varepsilon_{t-2}$$

$$\varepsilon_t \sim N(0, 1), \quad T = 100.$$



Autoregressive Moving Average (ARMA) models

An **autoregressive moving average** (ARMA) model of order p and q can be written as

$$
\begin{aligned}
y_t = c &+ \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} && (1)\\
&+ \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t, && (2)
\end{aligned}
$$

where ε_t is white noise.

- Predictors include both **lagged values** and **lagged errors**.
- Conditions on coefficients are required to ensure **stationarity**.

Outline

Last week recap

Statistical forecasting methods

- AR, MA and ARIMA models
- Model estimation, validation and selection
- (S)ARIMA models
- Auto.arima

AI forecasting models

- Challenges in training AI forecasting models (recap)
- Time-series (cross-)validation
- Some multi-step forecasting strategies
- (Deep) Neural Networks

Maximum Likelihood Estimation (model fitting)

- If the model order is given (i.e. p and q), we need to **estimate** the parameters

$$c, \phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q$$

- Maximum Likelihood Estimation (MLE) finds the values of the parameters which **maximise the probability** of obtaining the data that we have observed.
- MLE maximizes the **likelihood function**:

$$L = L(c, \phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q; y_1, y_2, \ldots, y_T).$$

- For ARMA models, MLE is similar to **least squares estimation**.

-

Information criteria (model validation and selection)

- For classical time series models, **model validatio**n is performed using **information criteria**, which are **asymptotically** equivalent to some form of (time-series) cross-validation.
- For ARMA models, the **Akaike's Information Criterion (AIC)** can be written as

$$\mathrm{AIC} = \mathrm{AIC}(p, q; y_1, y_2, \ldots, y_T) = -2\log(L) + 2(p + q + k + 1),$$

  where
  - L is the likelihood of the data
  - $k = 1$ if $c \neq 0$ and $k = 0$ if $c = 0$
  - the last term is the number of parameters in the model (including σ^2, the variance of the residuals)

-

Information criteria (model validation and selection)

- The **corrected AIC (AICc)** can be written as

$$\mathrm{AICc} = \mathrm{AIC} + \frac{2(p+q+k+1)(p+q+k+2)}{T-p-q-k-2}.$$

- The **Bayesian Information Criterion (BIC)** can be written as

$$\mathrm{BIC} = \mathrm{AIC} + [\log(T) - 2](p+q+k+1).$$

- Good models are obtained by **minimising** the AIC, AICc or BIC.

Outline

Last week recap

Statistical forecasting methods

- AR, MA and ARIMA models
- Model estimation, validation and selection
- (S)ARIMA models
- Auto.arima

AI forecasting models

- Challenges in training AI forecasting models (recap)
- Time-series (cross-)validation
- Some multi-step forecasting strategies
- (Deep) Neural Networks

Autoregressive Integrated Moving Average (ARIMA) models

- If we combine **differencing** with **autoregression** and a **moving average** model, we obtain a non-seasonal **ARIMA** (AutoRegressive Integrated Moving Average) model:

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

  where $y_t^{'}$ is the differenced series (one or multiple differences).

- We call it an **ARIMA**(p, d, q) model where
  - p is the order of the autoregressive part;
  - d is the degree of first differencing involved;
  - q is the order of the moving average part.

- Many particular cases:
  - White noise model: ARIMA$(0, 0, 0)$ with no constant
  - Random walk model: ARIMA$(0, 1, 0)$ with no constant
  - AR(p): ARIMA$(p, 0, 0)$
  - MA(q): ARIMA$(0, 0, q)$

Seasonal ARIMA (SARIMA) models

ARIMA	$\underbrace{(p, d, q)}$	$\underbrace{(P, D, Q)_m}$
	$\uparrow$ Non-seasonal part of the model	$\uparrow$ Seasonal part of of the model

where m is the number of observations in each season.

Outline

Last week recap

Statistical forecasting methods

- AR, MA and ARIMA models
- Model estimation, validation and selection
- (S)ARIMA models
- Auto.arima

AI forecasting models

- Challenges in training AI forecasting models (recap)
- Time-series (cross-)validation
- Some multi-step forecasting strategies
- (Deep) Neural Networks

Software

pmdarima.arima.auto_arima

```
pmdarima.arima.auto_arima(y, X=None, start_p=2, d=None, start_q=2, max_p=5, max_d=2, max_q=5, start_P=1, D=None, start_Q=1, max_P=2, max_D=1, max_Q=2, max_order=5, m=1, seasonal=True, stationary=False, information_criterion='aic', alpha=0.05, test='kpss', seasonal_test='ocsb', stepwise=True, n_jobs=1, start_params=None, trend=None, method='lbfgs', maxiter=50, offset_test_args=None, seasonal_test_args=None, suppress_warnings=True, error_action='trace', trace=False, random=False, random_state=None, n_fits=10, return_valid_fits=False, out_of_sample_size=0, scoring='mse', scoring_args=None, with_intercept='auto', sarimax_kwargs=None, **fit_args)
```
[source] [source]

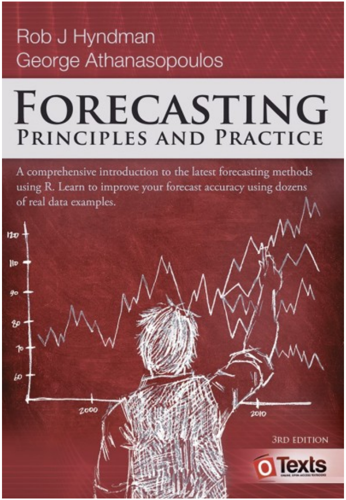
- https://alkaline-ml.com/pmdarima/modules/classes.html
- https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.AutoARIMA.html#pmdarima.arima.AutoARIMA
- https://alkaline-ml.com/pmdarima/tips_and_tricks.html

Software

-

```
                           SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1000
Model:               SARIMAX(2, 0, 2)   Log Likelihood               -1398.466
Date:                                   AIC                           2806.931
Time:                                   BIC                           2831.470
Sample:                             0   HQIC                          2816.258
                               - 1000
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.7012      0.075      9.383      0.000       0.555       0.848
ar.L2         -0.2353      0.060     -3.910      0.000      -0.353      -0.117
ma.L1          0.6982      0.072      9.724      0.000       0.557       0.839
ma.L2          0.3858      0.051      7.513      0.000       0.285       0.486
sigma2         0.9578      0.042     22.699      0.000       0.875       1.041
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                 3.83
Prob(Q):                              0.99   Prob(JB):                         0.15
Heteroskedasticity (H):               0.89   Skew:                             0.14
Prob(H) (two-sided):                  0.28   Kurtosis:                         3.09
===================================================================================
```

Reference



https://otexts.com/fpp3

Outline

Last week recap

Statistical forecasting methods

AI forecasting models

- Challenges in training AI forecasting models (recap)
- Time-series (cross-)validation
- Some multi-step forecasting strategies
- (Deep) Neural Networks

Outline

Challenges in training AI models for forecasting

- (Statistically) dependent data
  - (naive) shuffling can destroy the temporal dependence structure
  - **Time-series training/test split needed**
- Non-stationarity
  - Validity of the training/test split?
  - Transformations to stabilize mean and variance
- Specific patterns: seasonality, trend, cycle, etc
  - Can the model capture these patterns?
- Arrow of time: use past observations to predict future values
- **Multi-step ahead forecasting, i.e. sequential predictions**

Outline

Last week recap

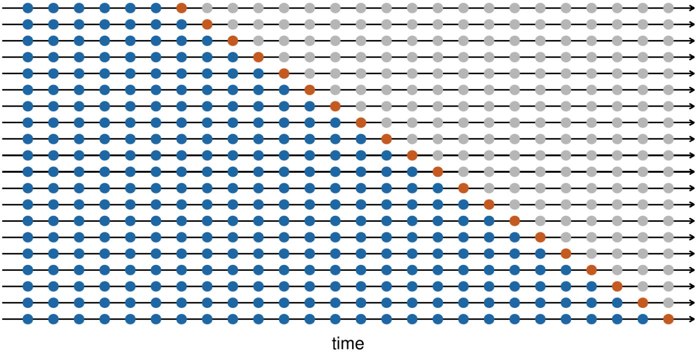
Statistical forecasting methods

- AR, MA and ARIMA models
- Model estimation, validation and selection
- (S)ARIMA models
- Auto.arima

AI forecasting models

- Challenges in training AI forecasting models (recap)
- Time-series (cross-)validation
- Some multi-step forecasting strategies
- (Deep) Neural Networks

Time-series cross-validation



Time-series cross-validation

$$\underbrace{y_1, y_2, y_3, y_4, y_5, y_6}_{\text{Training}}, \underbrace{y_7, y_8, y_9, y_{10}}_{\text{Validation}}$$

- $y_1, y_2, y_3, y_4, y_5, y_6 \longrightarrow y_7$
- $y_1, y_2, y_3, y_4, y_5, y_6, y_7 \longrightarrow y_8$
- $y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8 \longrightarrow y_9$
- $y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9 \longrightarrow y_{10}$

Also called the **rolling-origin** approach.

Time-series cross-validation - Example with lag $p = 3$

X			y
y_{t-2}	y_{t-1}	y_t	y_{t+1}
y_1	y_2	y_3	y_4
y_2	y_3	y_4	y_5
y_3	y_4	y_5	y_6
y_4	y_5	y_6	y_7

X			y
y_{t-2}	y_{t-1}	y_t	y_{t+1}
y_1	y_2	y_3	y_4
y_2	y_3	y_4	y_5
y_3	y_4	y_5	y_6
y_4	y_5	y_6	y_7
y_5	y_6	y_7	y_8

X			y
y_{t-2}	y_{t-1}	y_t	y_{t+1}
y_1	y_2	y_3	y_4
y_2	y_3	y_4	y_5
y_3	y_4	y_5	y_6
y_4	y_5	y_6	y_7
y_5	y_6	y_7	y_8
y_6	y_7	y_8	y_9

X			y
y_{t-2}	y_{t-1}	y_t	y_{t+1}
y_1	y_2	y_3	y_4
y_2	y_3	y_4	y_5
y_3	y_4	y_5	y_6
y_4	y_5	y_6	y_7
y_5	y_6	y_7	y_8
y_6	y_7	y_8	y_9
y_7	y_8	y_9	y_{10}

The validation set approach

$$\underbrace{y_1, y_2, y_3, y_4, y_5, y_6}_{\text{Training}}, \underbrace{y_7, y_8, y_9, y_{10}}_{\text{Validation}}$$

$$y_1, y_2, y_3, y_4, y_5, y_6 \longrightarrow y_7$$
$$\longrightarrow y_8$$
$$\longrightarrow y_9$$
$$\longrightarrow y_{10}$$

X			y
y_{t-2}	y_{t-1}	y_t	y_{t+1}
y_1	y_2	y_3	y_4
y_2	y_3	y_4	y_5
y_3	y_4	y_5	y_6
y_4	y_5	y_6	y_7
y_5	y_6	y_7	y_8
y_6	y_7	y_8	y_9
y_7	y_8	y_9	y_{10}

Outline

Last week recap

Statistical forecasting methods

- AR, MA and ARIMA models
- Model estimation, validation and selection
- (S)ARIMA models
- Auto.arima

AI forecasting models

- Challenges in training AI forecasting models (recap)
- Time-series (cross-)validation
- Some multi-step forecasting strategies
- (Deep) Neural Networks

Multi-step forecasting - recursive strategy ($H = 3$)

$$y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10} \rightarrow \hat{y}_{11}, \hat{y}_{12}, \hat{y}_{13}$$

X			y
y_{t-2}	y_{t-1}	y_t	y_{t+1}
y_1	y_2	y_3	y_4
y_2	y_3	y_4	y_5
y_3	y_4	y_5	y_6
y_4	y_5	y_6	y_7
y_5	y_6	y_7	y_8
y_6	y_7	y_8	y_9
y_7	y_8	y_9	y_{10}
y_8	y_9	y_{10}	$\hat{y}_{11}$
y_9	y_{10}	$\hat{y}_{11}$	$\hat{y}_{12}$
y_{10}	$\hat{y}_{11}$	$\hat{y}_{12}$	$\hat{y}_{13}$

This is the strategy used by classical statistical time series models (e.g. ARIMA).

Multi-step forecasting - direct strategy ($H = 3$)

$$y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10} \rightarrow \hat{y}_{11}, \hat{y}_{12}, \hat{y}_{13}$$

X			y
y_{t-2}	y_{t-1}	y_t	y_{t+1}
y_1	y_2	y_3	y_4
y_2	y_3	y_4	y_5
y_3	y_4	y_5	y_6
y_4	y_5	y_6	y_7
y_5	y_6	y_7	y_8
y_6	y_7	y_8	y_9
y_7	y_8	y_9	y_{10}
y_8	y_9	y_{10}	$\hat{y}_{11}$

X			y
y_{t-2}	y_{t-1}	y_t	y_{t+2}
y_1	y_2	y_3	y_5
y_2	y_3	y_4	y_6
y_3	y_4	y_5	y_7
y_4	y_5	y_6	y_8
y_5	y_6	y_7	y_9
y_6	y_7	y_8	y_{10}
y_8	y_9	y_{10}	$\hat{y}_{12}$

X			y
y_{t-2}	y_{t-1}	y_t	y_{t+3}
y_1	y_2	y_3	y_6
y_2	y_3	y_4	y_7
y_3	y_4	y_5	y_8
y_4	y_5	y_6	y_9
y_5	y_6	y_7	y_{10}
y_8	y_9	y_{10}	$\hat{y}_{13}$

Multi-step forecasting - multi-output strategy ($H = 3$)

$$y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10} \rightarrow \hat{y}_{11}, \hat{y}_{12}, \hat{y}_{13}$$

X			y		
y_{t-2}	y_{t-1}	y_t	y_{t+1}	y_{t+2}	y_{t+3}
y_1	y_2	y_3	y_4	y_5	y_6
y_2	y_3	y_4	y_5	y_6	y_7
y_3	y_4	y_5	y_6	y_7	y_8
y_4	y_5	y_6	y_7	y_8	y_9
y_5	y_6	y_7	y_8	y_9	y_{10}
y_8	y_9	y_{10}	$\hat{y}_{11}$	$\hat{y}_{12}$	$\hat{y}_{13}$

$\rightarrow$ The multi-output strategy requires a model that can deal with multiple outputs, e.g. neural networks.

Outline

Last week recap

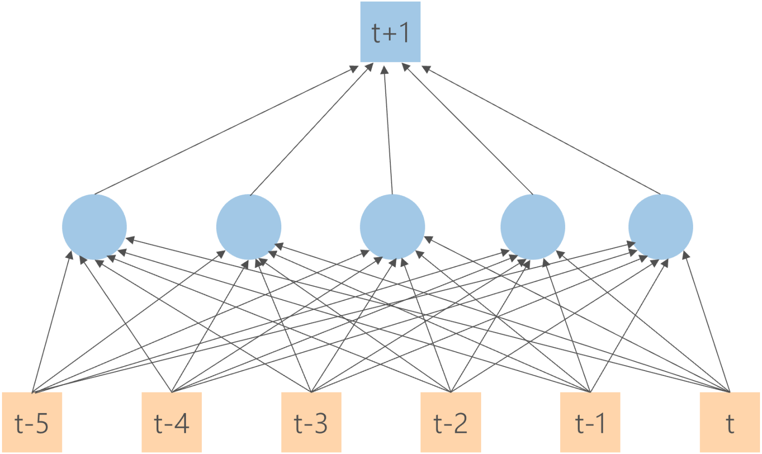
Statistical forecasting methods

AR, MA and ARIMA models
Model estimation, validation and selection
(S)ARIMA models
Auto.arima

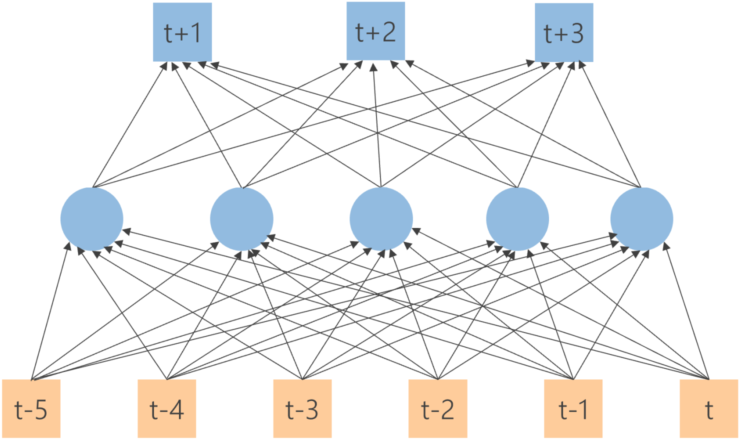
AI forecasting models

Challenges in training AI forecasting models (recap)
Time-series (cross-)validation
Some multi-step forecasting strategies
(Deep) Neural Networks

Single output network



Multi-output network



Neural network hyperparameters

Hyperparameters are adjustable parameters that define the model architecture and govern the learning process. By contrast, other parameters (such as node weights) are derived via model training.

- Architecture
  - Types of layers, number of layers, layer order, number of neurons per layer, layer activations, etc.
- Optimization
  - Opitmizer, weight initialization, learning rate, batch size, number of epochs, stopping criterion, etc.
- Loss function
  - Loss function, form of regularization, etc.
- ...

Hyperparameter tuning: search across various hyperparameter configurations and select the configuration that results in best (out-of-sample) performance