Time series modeling and forecasting

Lecture 18 of “Mathematics and AI”

Outline

1. Stochastic processes

   moments, stationarity, autocorrelation, extrapolation

2. Linear models for time-series forecasting

   (vector-)autoregression, moving-average, naïve forecasting

3. Nonlinear models for time-series forecasting

   integrated process, detrending, ARIMA

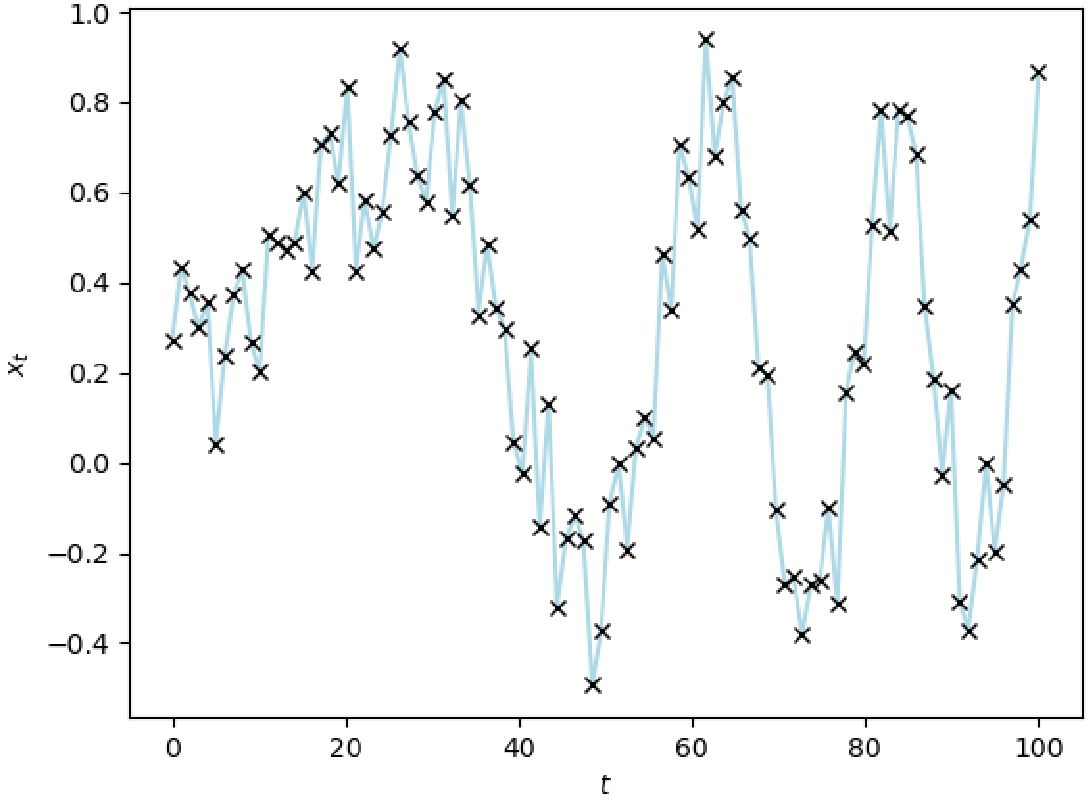
4. Neural networks for time-series forecasting

   recurrent neural networks, long short-term memory

Stochastic processes

Time series data

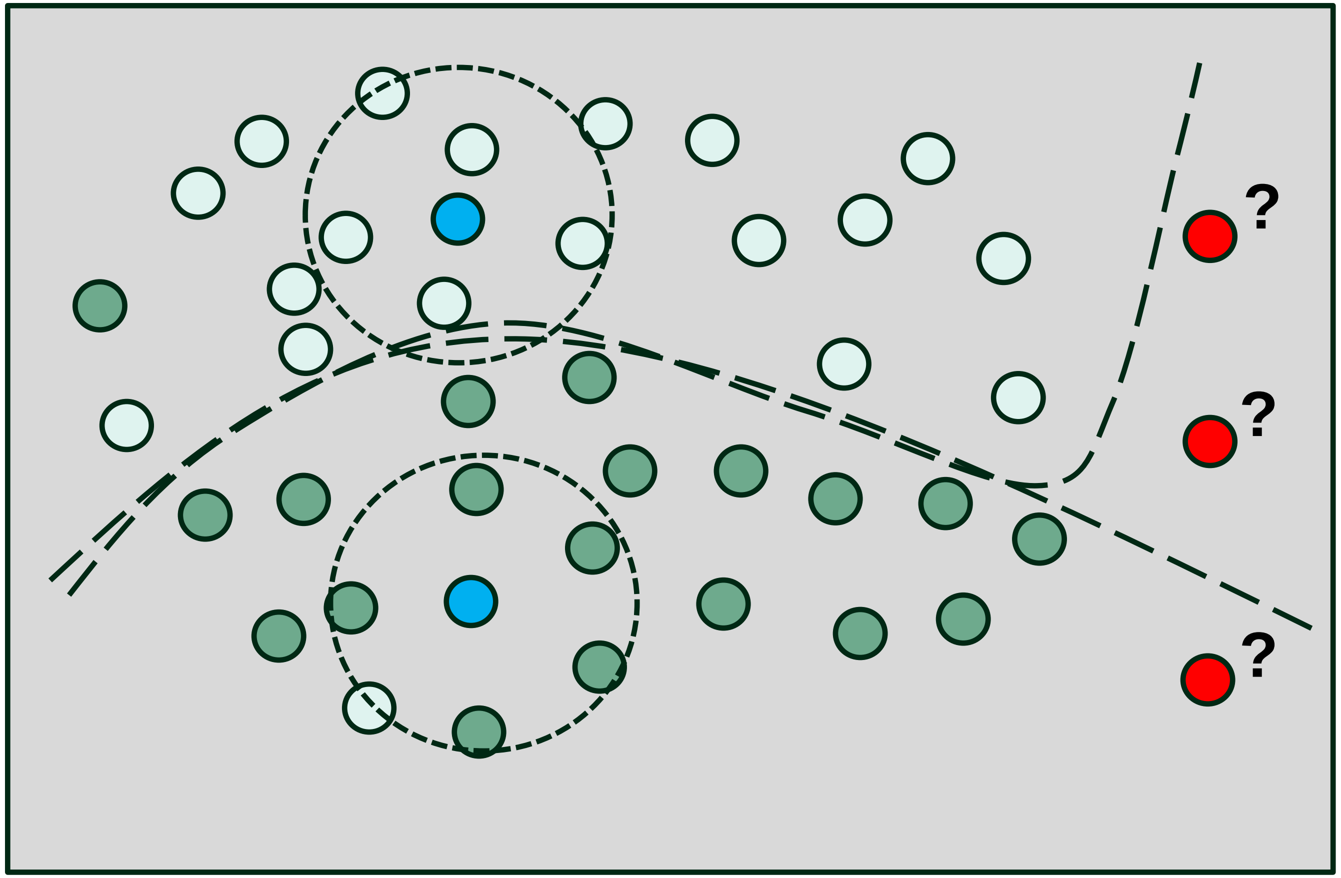
- Sequence of observations
  $x_1, x_2, \dots, x_{t-1}, x_t$
- The order of observations matters!
- Forecast: Prediction of future values
  x_{t^*} with $t^* > t$



Extrapolation vs. interpolation

- Extrapolation tends to be much harder than interpolation in ML
- Important for AI applications

Example: KNN



A stochastic process

Random variable X

defined by prob. distribution $p(x), x \in D_X$

Realization of a random variable

value (can be a vector) $x \in D_X$
drawn from $p(x)$

Stochastic process X_t

defined by joint prob. distribution $p(x_1, x_2, \dots), x_i \in D_X$

Realization of a stochastic process

sequence $(x_1, x_2, \dots, x_t) \in D_x^t$
drawn from $p(x_1, x_2, \dots)$

DARTMOUTH

A stochastic process

Random variable X

defined by prob. distribution $p(x), x \in D_X$

Stochastic process X_t

defined by joint prob. distribution $p\left(\{x_t\}_{t=1,2,\ldots}\right), x_i \in D_X$

Moments of a random variable

Expectation

$$\langle x \rangle = \sum_{x \in D_X} p(x) x$$

Variance (for mean-centered x)

$$\langle x^2 \rangle = \sum_{x \in D_X} p(x)\, x^2$$

Moments of a stochastic process

Expectation

$$\langle x_t \rangle = \sum_{x \in D_X} p_t(x_t) x_t$$

Autocovariance (for mean-centered x_{t_1}, x_{t_2})

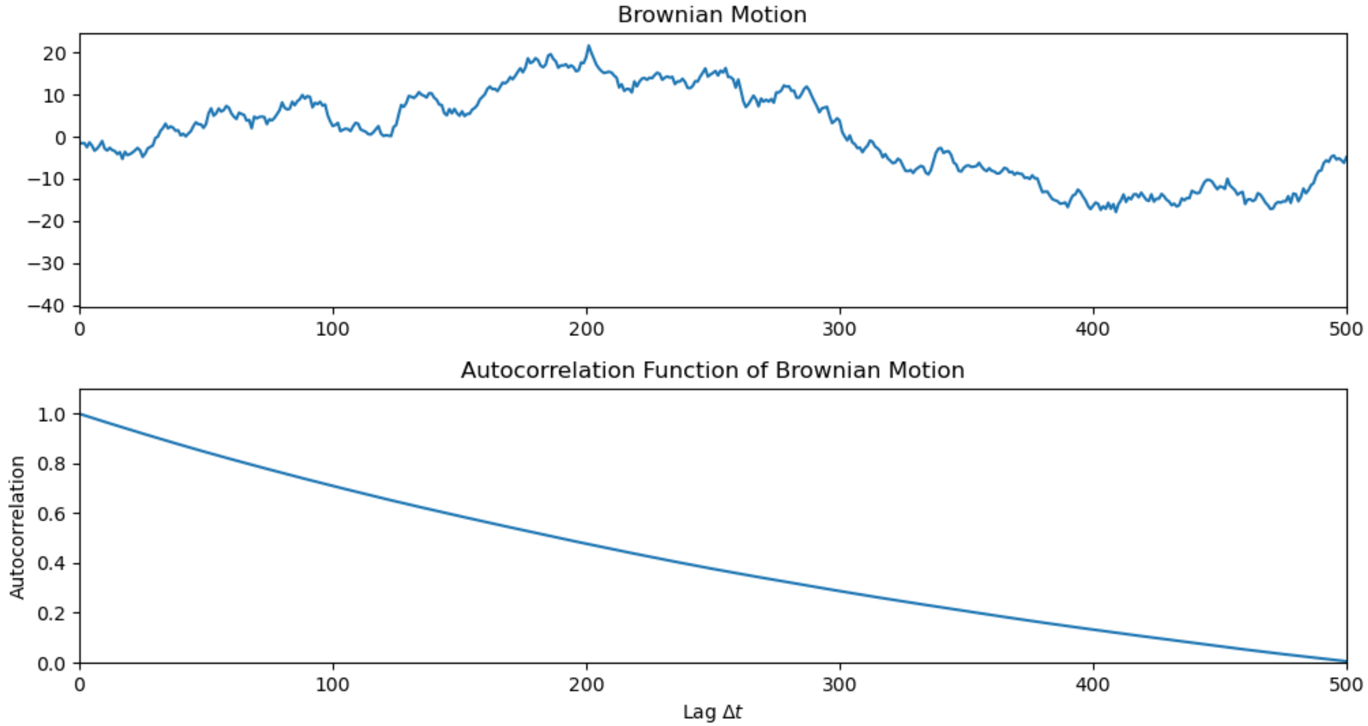
$$\langle x_{t_1} x_{t_2} \rangle = \sum_{x_{t_1}, x_{t_2} \in D_X} p\left(x_{t_1}, x_{t_2}\right) x_{t_1} x_{t_2}$$

Empirical autocorrelation function

- Time series data observed for T time steps with variance $s^2 = \frac{1}{T}\sum_{t=1}^{T}(x_t - \bar{x}_t)^2$
- Autocorrelation r_k function measures correlation between mean-centered x_t and mean-centered x_{t-k}

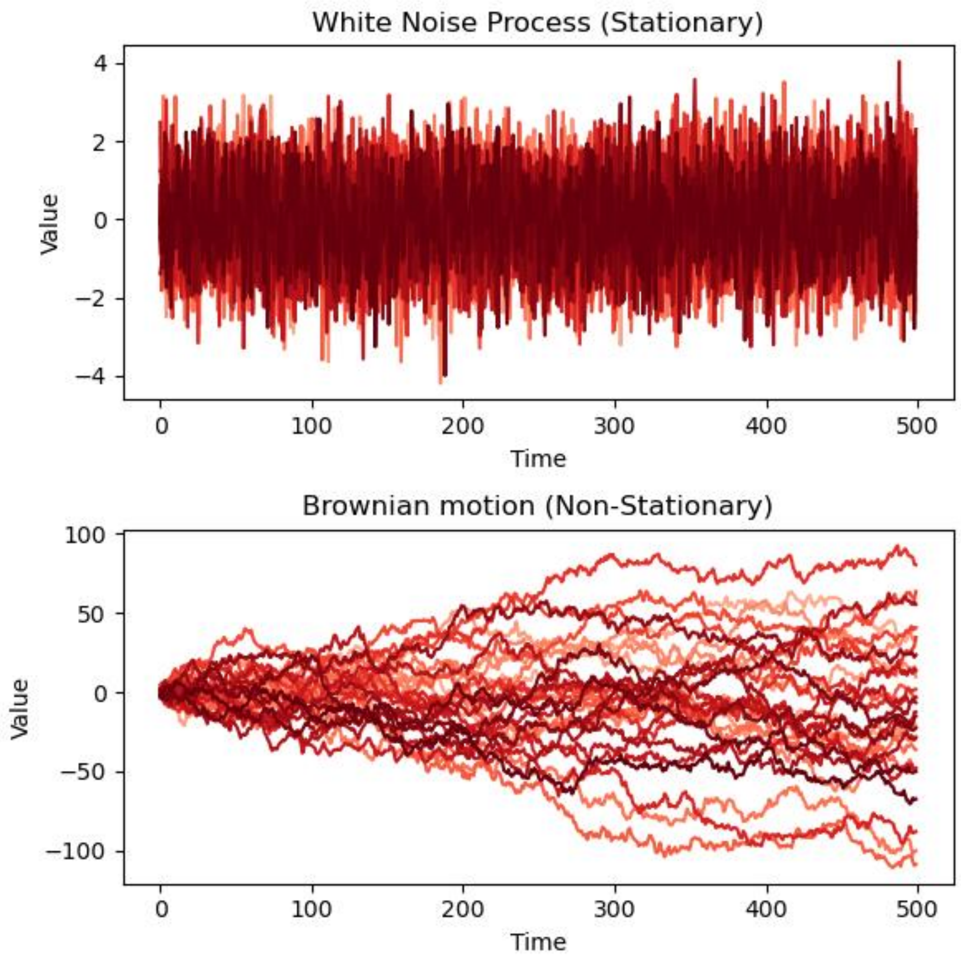
$$r_k = \frac{1}{Ts^2}\sum_{t=k+1}^{T}(x_t - \bar{x}_t)(x_{t-k} - \bar{x}_{t-k})$$

- Typically see downward trend with increasing k



A stationary stochastic process

- Stationary stochastic process
  - Process that is not explicitly time-dependent
  - No trend
  - All moments are independent of t
  - Autocorrelation function only depends on $k = \Delta t$, not t
- Weak stationarity:
  - First and second moments are independent of t



Distribution is normal

iid data generates flat power spectrum

Example 1: Gaussian white noise

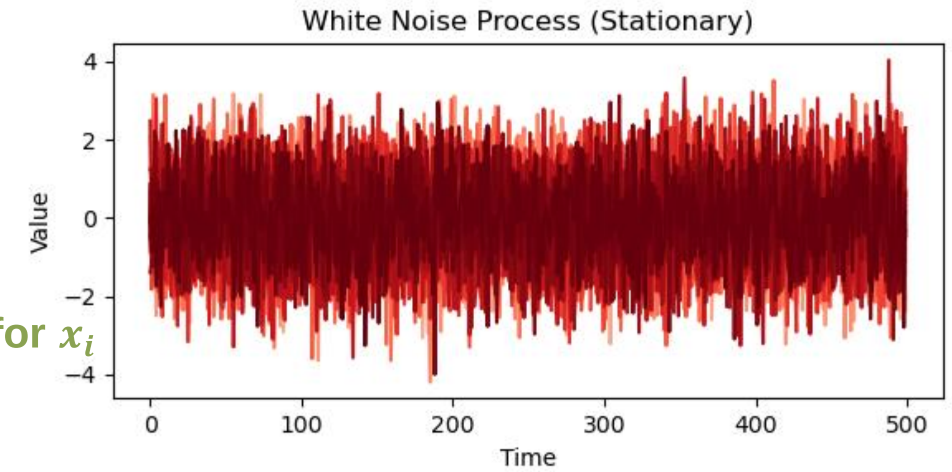
- Sequence $x_1, x_2, \dots, x_t$ of samples drawn from a normal distribution

- Samples are drawn **iid** (**i**dentically **i**ndependently **d**istributed)

Same normal distribution for x_i

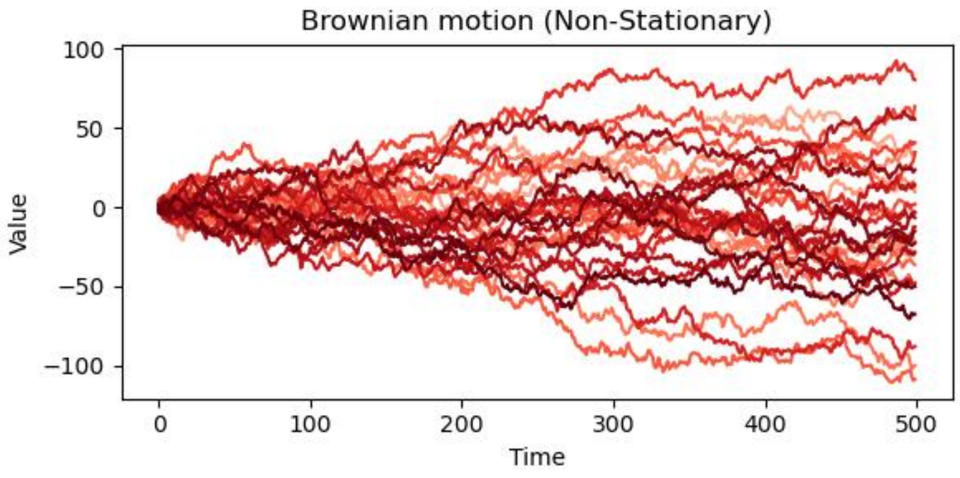
Distribution of x_i does not depend on x_j for $j \neq i$, i.e., $p(x_i, x_j) = p(x_i)\, p(x_j)$

➢ Stationary process



Example 2: Brownian motion

$$x_t = x_{t-1} + \varepsilon_t$$



- Model for particle movement
- Variance increases with t
- Not stationary

Linear models for time-series forecasting

Linear models for time-series modeling

- The state x_t can be expressed a linear combination of previous states:

$$x_t = \varphi_{t,0} + \varphi_{t,1}x_{t-1} + \varphi_{t,2}x_{t-2} + \cdots + \varphi_{t,k}x_{t-k} + \cdots + \varphi_{t,t}x_0$$
$$+\omega_{t,0}\varepsilon_t + +\omega_{t,1}\varepsilon_{t-1} + \omega_{t,2}\varepsilon_{t-2} + \cdots + \omega_{t,k}\varepsilon_{t-k} + \cdots + \omega_{t,t}\varepsilon_0$$

- There are $t + \binom{t+1}{2}$ parameters $\varphi_{i,j}$ and $\binom{t+1}{2}$ parameters $\omega_{i,j}$!
- Make some regularizing assumptions about $\varphi_{i,j}$ and $\omega_{i,j}$

Autoregressive (AR) model

$$x_t = \varphi_0 + \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \cdots + \varphi_p x_{t-p} + \varepsilon_t$$

- Assumptions
  - $\varphi_{i,k} = \varphi_{j,k}$ for all i, j
  - $\varphi_{i,k} = 0$ for all $k > p$
  - $\omega_{t,0} = 1$, and
  - $\omega_{i,k} = 0$ for all other i, k
- Notation
  - AR(1) process has $p = 1$
  - AR(p) process typically has $p > 1$
  - Process is a vector-autoregressive (VAR) process if x_t is vector-valued

Moving-average (MA) model

$$x_t = \varepsilon_t + \omega_1 \varepsilon_{t-1} + +\omega_2 \varepsilon_{t-2} + \cdots + +\omega_p \varepsilon_{t-p}$$

- Assumptions
  - $\omega_{i,k} = \omega_{j,k}$ for all i, j
  - $\omega_{i,k} = 0$ for all $k > p$
  - $\omega_{t,0} = 1$, and
  - $\varphi_{i,k} = 0$ for i, k
- Notation
  - MA(1) process has $p = 1$
  - MA(p) process typically has $p > 1$
  - ARMA process combines AR and MA process
  - VARMA is vector-valued ARMA process

Forecasting with linear models

Fit linear model to training data $x_1, x_2, \ldots, x_t$ → Predict $x_{t+1}, x_{t+2}, \ldots, x_{t+H}$

Prediction is distribution or sample from distribution

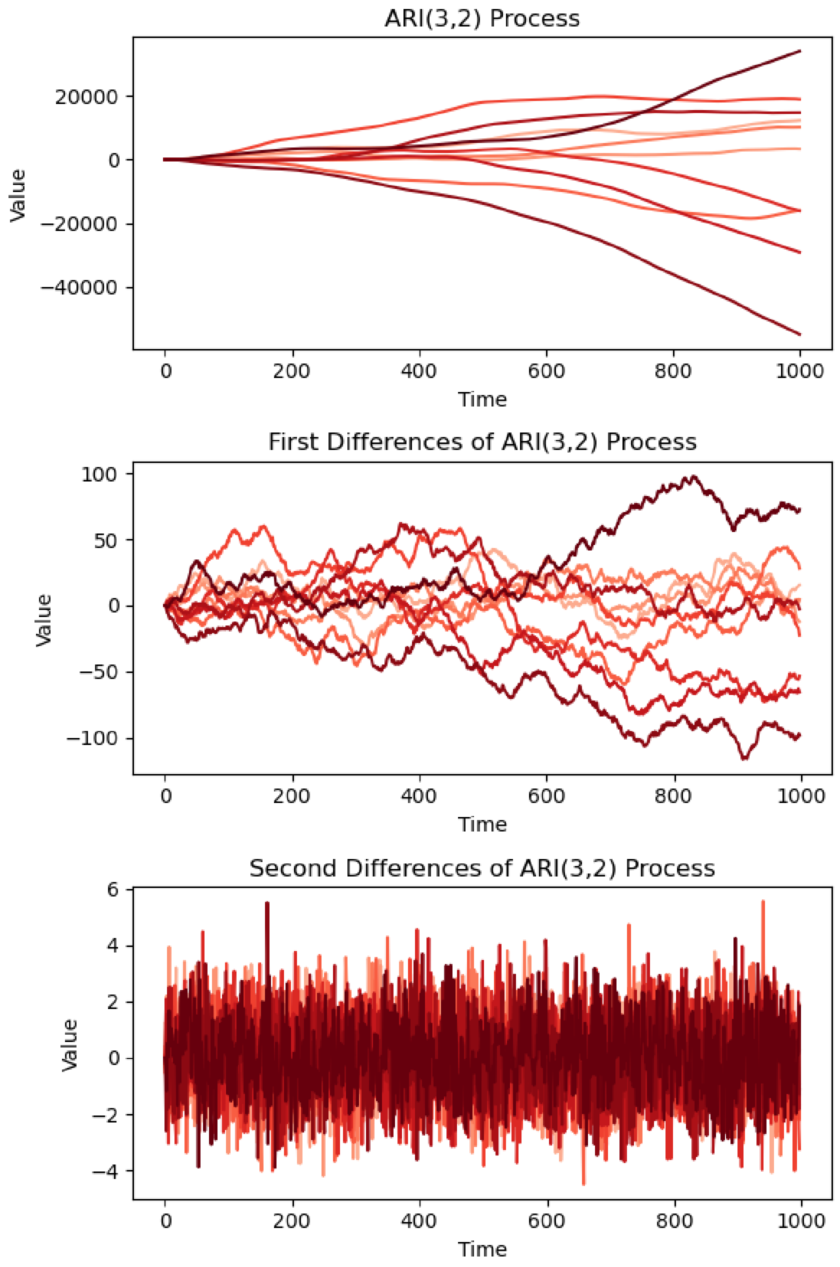
Nonlinear models for time-series forecasting

Integrated processes

- Integrated ARI(1) process
  - x_t is not a linear process, but $\Delta x_t \coloneqq x_t - x_{t-1}$ is a AR process
- Integrated ARIMA(1) process
  - x_t is not a linear process, but $\Delta x_t \coloneqq x_t - x_{t-1}$ is an ARMA process
- Integrated ARIMA(p) process
  - x_t is not a linear process, but $\Delta^p x_t \coloneqq x_t - x_{t-1}$ is an ARMA process

Integrated processes

- Integrated processes can have polynomial trends
- Integrated processes are not stationary
- They can be de-trended via differencing



Nonlinear autoregressive (NAR) processes

$$x_t = f\left(x_{t-1}, x_{t-2}, \ldots, x_{t-p}\right) + \varepsilon_t$$

- Nonlinear function f
- cannot be de-trended via differencing (in general)
- Expressiveness depends on choice of f

Neural networks for time-series forecasting

Recurrent neural networks

- Keep track of parts of the network state from previous inputs to mimic autoregression
- Use recurrent network connections
- Consistent with natural neural network architectures outside the visual system

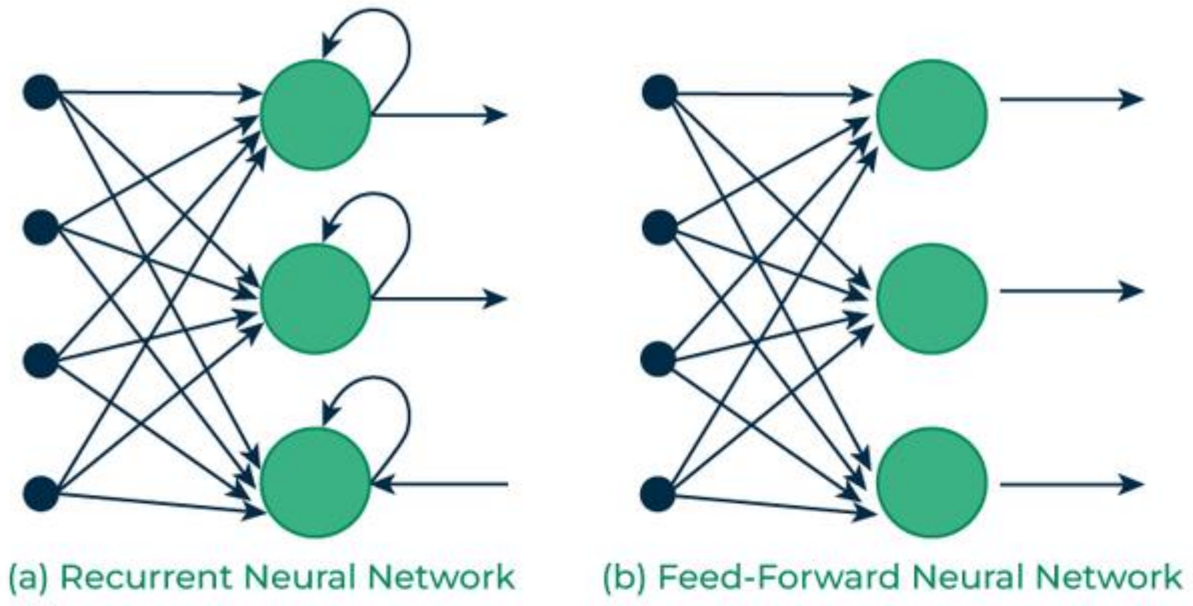
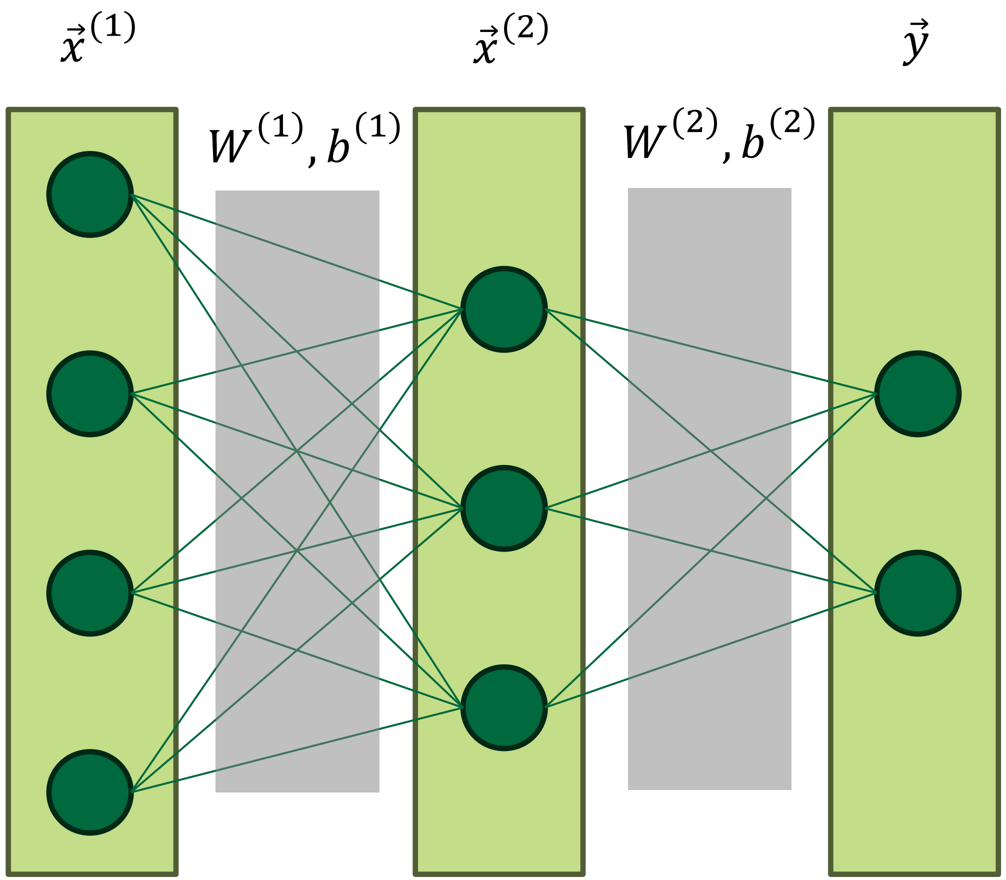


Image source: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

Simple recurrent network

Replace hidden state

$$x_t^{(2)} = \sigma\left(W^{(1)} x_t^{(1)} + b^{(1)}\right)$$



Simple recurrent network

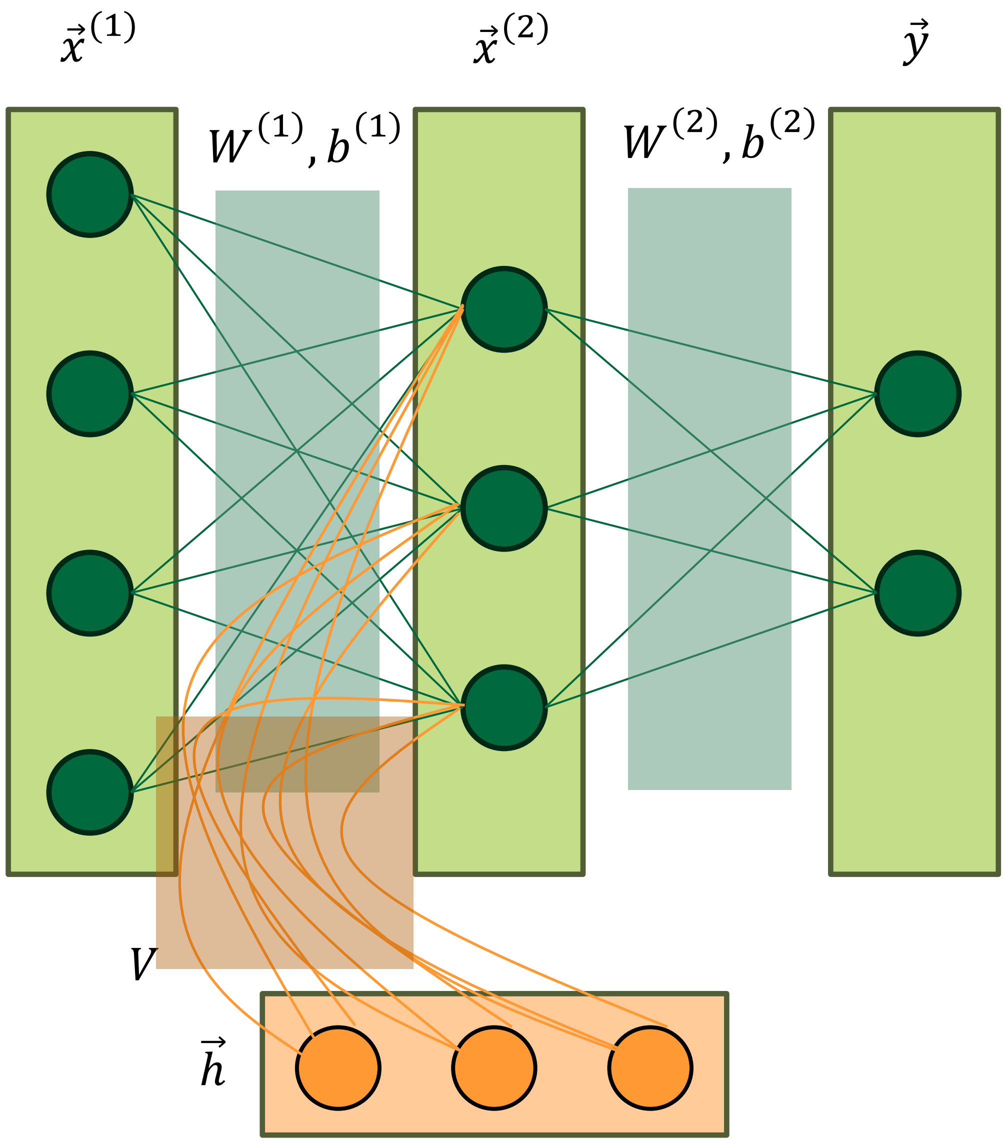
Replace hidden state

$$x_t^{(2)} = \sigma\left(W^{(1)}x_t^{(1)} + b^{(1)}\right)$$

with new hidden state

$$x_t^{(2)} = \sigma_x\left(W^{(1)}x_t^{(1)} + Vh_t + b_x^{(1)}\right)$$

New layer of weights!



DARTMOUTH

Simple recurrent network

Replace hidden state

$$x_t^{(2)} = \sigma\left(W^{(1)} x_t^{(1)} + b^{(1)}\right)$$

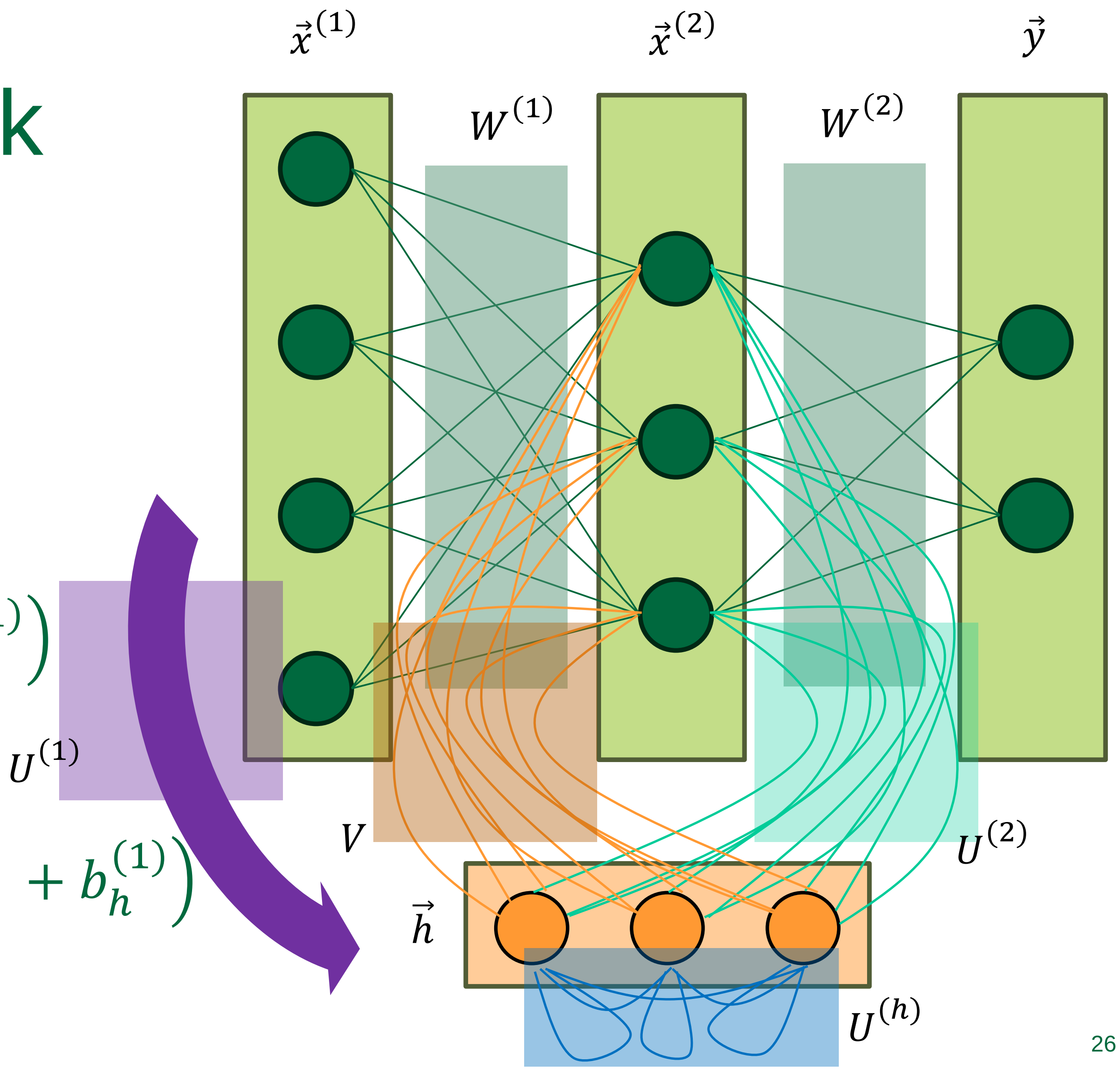
with new hidden state

$$x_t^{(2)} = \sigma_x\left(W^{(1)} x_t^{(1)} + V h_t + b_x^{(1)}\right)$$

With memory state

$$h_t = \sigma_h\left(U^{(1)} x_t^{(1)} + U^{(2)} x_t^{(2)} + U^{(h)} h_t + b_h^{(1)}\right)$$

New layer of weights!



General recurrent unit

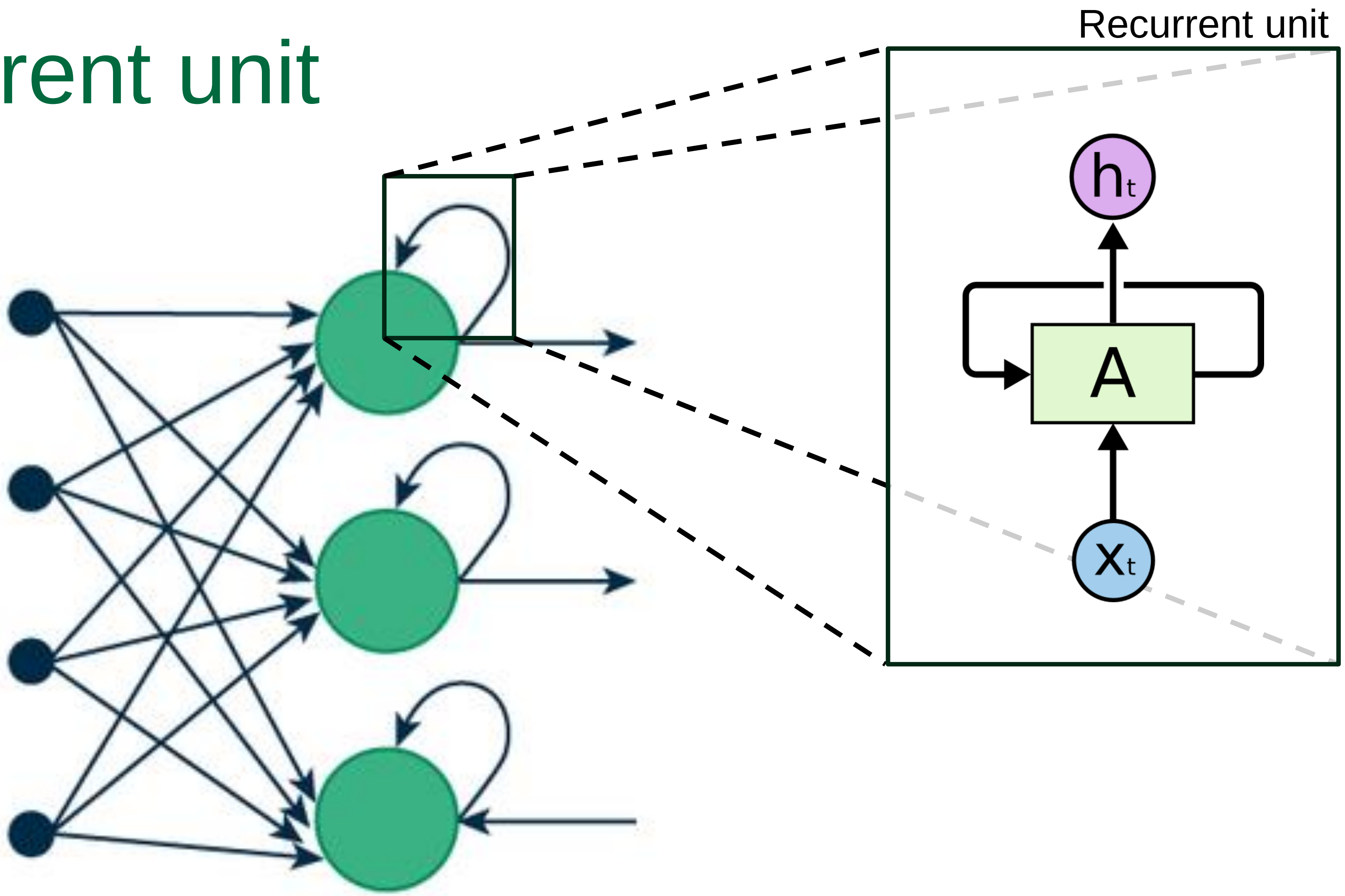
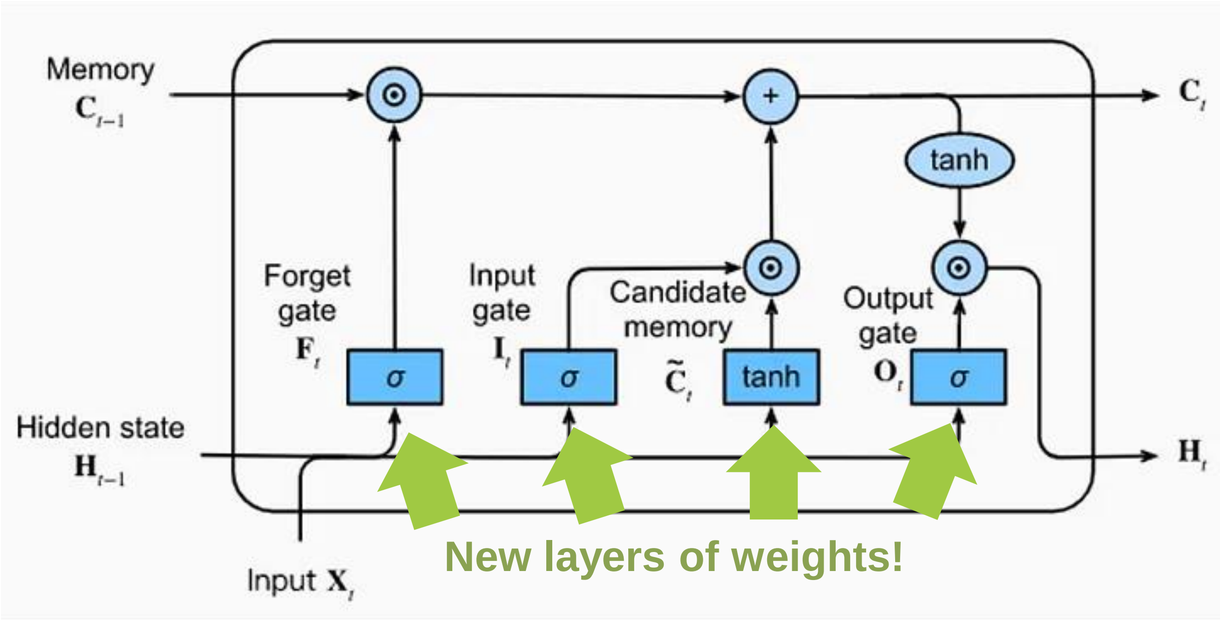
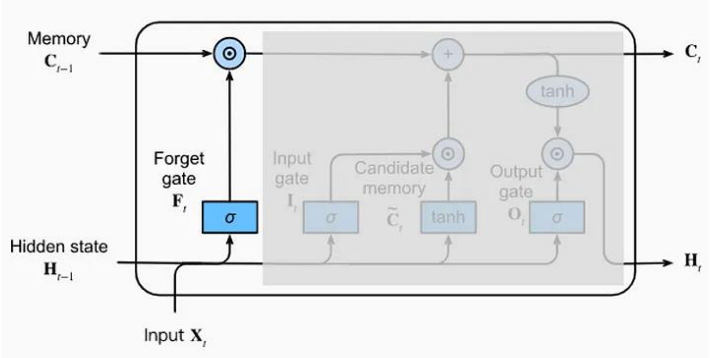


Image sources: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network, https://colah.github.io/posts/2015-08-Understanding-LSTMs

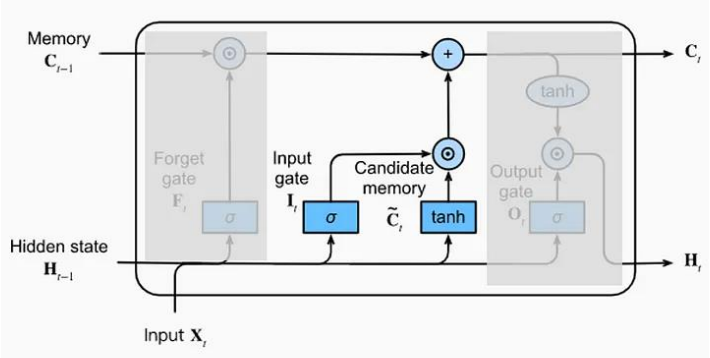
Long short-term memory



Forget gate



Input gate



Output gate

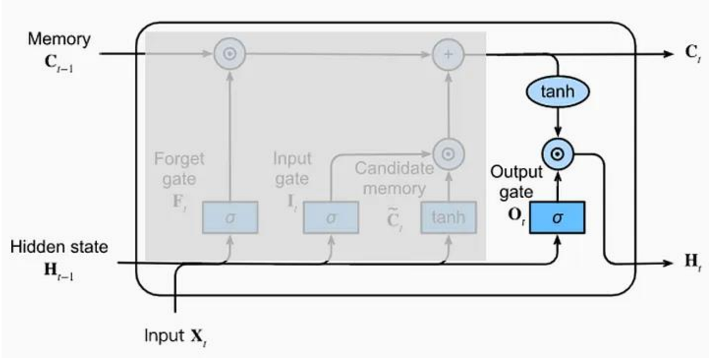


Image source: https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c