

# Temporal Data Mining for Healthcare Data

Dr Chang Xu

School of Computer Science

Reference: Healthcare Data Analytics, Chapter 11

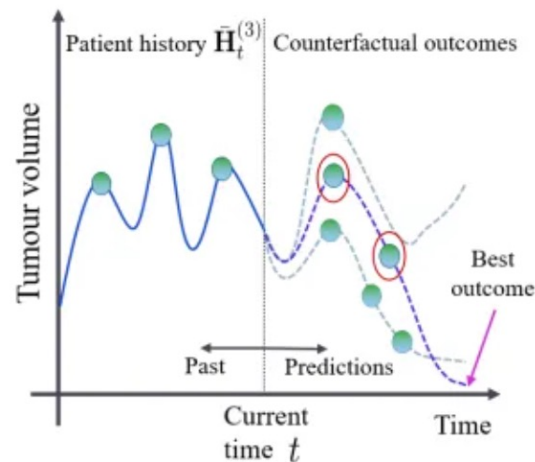


THE UNIVERSITY OF  
SYDNEY



# Temporal data in healthcare

- Healthcare data almost always contain time information.
- Such as the time for diagnosing and treatments, the duration of the symptoms.
- By exploring the temporal information could provide complicate analysis for diseases and treatments that cannot be achieved by only analysis the condition in one single time point.



# Electronic health records (EHR)

- EHR contain longitudinal patient health information, including demographics, laboratory test results, medication orders, medical diagnoses, procedures, progress notes, radiology reports, etc.
- All these information called “**medical events**” and each event will come with one timestamp.
- Mining the temporal dimension of EHR data is extremely promising as it may reveal patterns that enable a more precise understanding of disease manifestation, progression, and response to therapy.

# EHR data are:

**Multivariate:** A large number of clinical variables might be measured for a single patient (e.g., white blood counts, creatinine values, cholesterol levels, etc.).

Handy patients enterprise edition

File Edit View Help

David (8 month and 10 days)  
John (2 years and 3 months)  
Mother: Teacher  
Father: Financial advisor  
Parents: Married

Last: Anderson P  
First: David Boy  
Birth: 5 January 2009  
Age: 8 month and 10 days Patient nb: 3

Forms  
Meeting (Doctor)  
Full status (Doctor)  
Assistant  
Billing  
Reports  
Statistics

SOAP Sum T  
R-V T, P, PC  
Admission Agenda

Sheets  
O: Neurologic  
O: Vascular  
O: Cardiac  
O: Respiratory  
O: Abdomen  
Exams  
Radiology  
Summary  
Patient documents  
Letter

Meetings  
2 month checkup 5 Mar 09 2m.0d  
1 month checkup 5 Feb 09 1m.0d  
Respiratory problem 22 Jan 09 17d  
10 days checkup 13 Jan 09 8d  
Control for return at home 9 Jan 09 4d  
Birth 5 Jan 09 0d

Diagnosis  
General  
My Diagnosis  
Social

New documents  
- Abdomen palpat 15 Sep 2009  
- Cardiac auscul 15 Sep 2009  
To do  
Send checkup

Notes  
Father ask many questions, add 10 minutes to consultation

Current doctor Dr Herman

Menu 1 Menu 2 Menu 3 Search

## Digestive

Thursday, 22 Jan 2009

Digestive inspection  
Normal

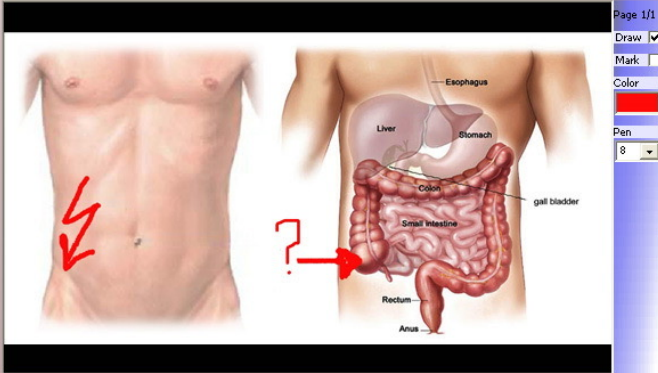
Digestive auscultation  
Normal abdomen noises

Digestive palpation  
Little pain on the right lower area

Liver  
No hepatomegaly.

Rectal

Page 1/1  
Draw  
Mark  
Color  
Pen  
8

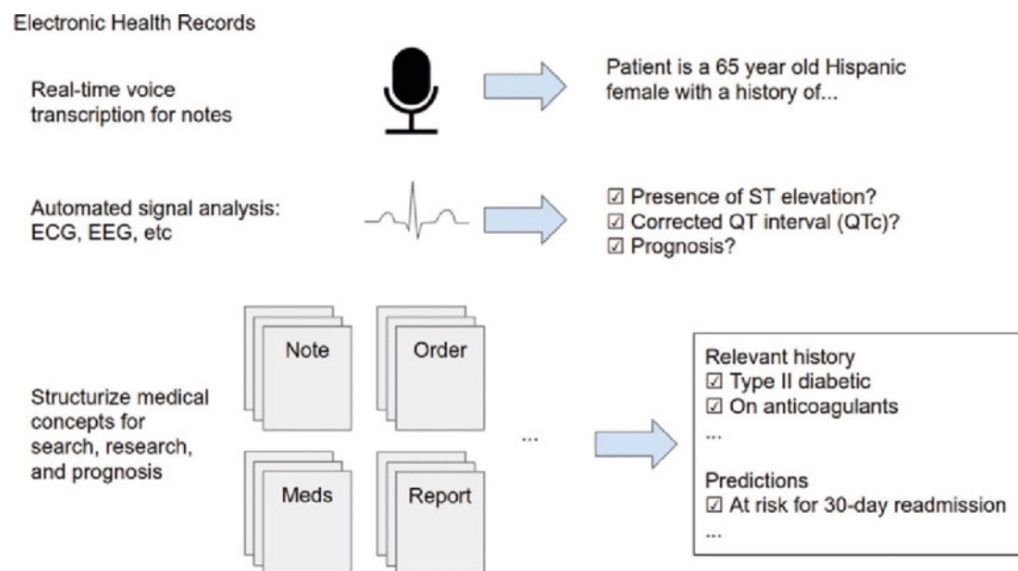


Documents manager

Previous page Next page

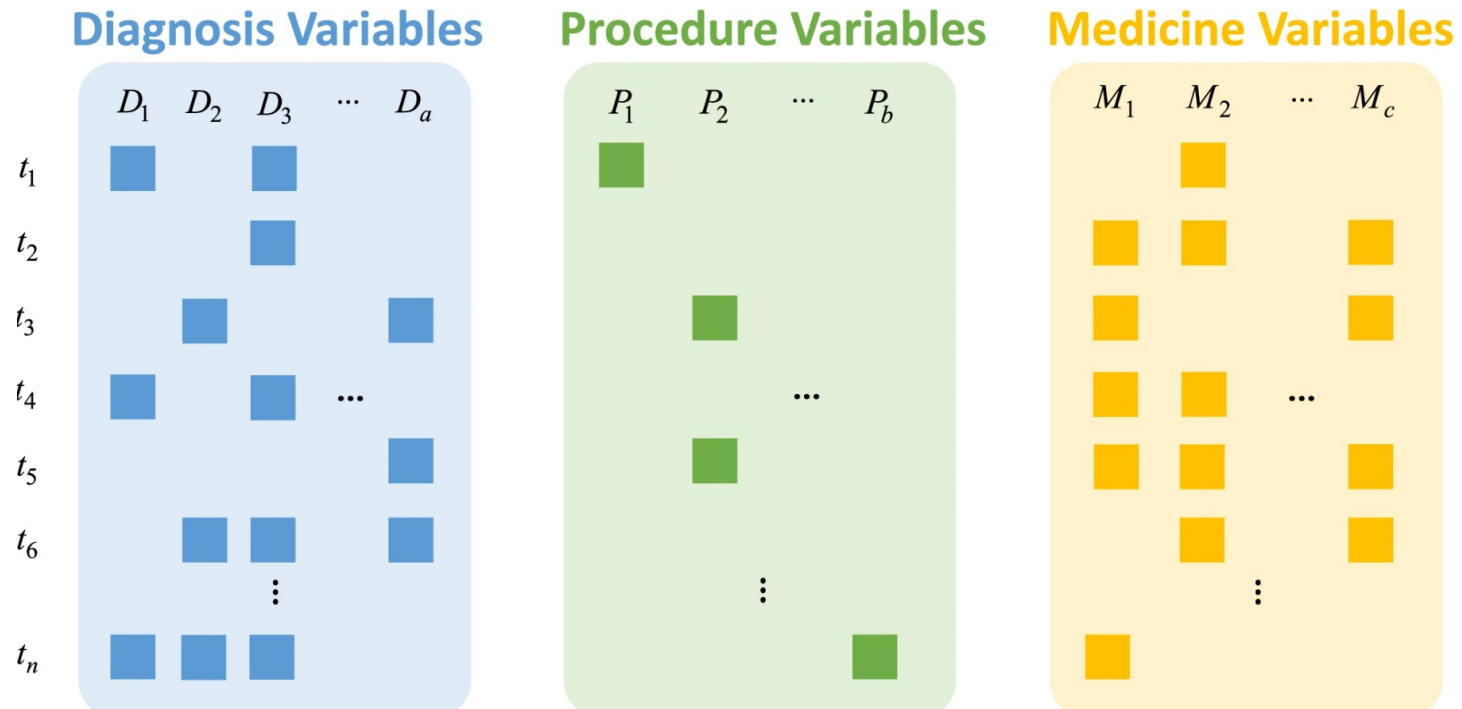
# EHR data are:

***Heterogeneous:*** The data contain multiple types of events; some events have numeric values (e.g., lab results), some events have categorical values (e.g., diagnosis/procedure codes), and some events may even have time durations (e.g., medications orders are usually associated with a time interval during which the patient should take the medication).



# EHR data are:

***Irregular in time:*** The variables are measured asynchronously at irregular time intervals (i.e., data are collected whenever a patient visits a healthcare facility). The time intervals at which the variables are measured can greatly vary between different patients as well as within a specific patient.



## EHR data are:

***Sparse:*** The data contain a lot of unknown/missing values because patients do not undergo all examinations every time.

# Training Data

	Admin Data (Dense)				Other Diagnoses (Sparse)	Procedures (Sparse)
	Age	LoS <sup>1</sup>	R.P. <sup>2</sup>	Diag. ...	...	...
0	33	2	0	A53.1	0 0 1 0 0 0 1 ... 0	0 1 0 0 0 0 0 0 0 0 ... 0
1	56	14	1	C77.3	0 0 0 0 0 0 0 ... 1	0 0 1 0 0 0 0 0 0 0 ... 0
.	73	3	0	Z02.2	1 0 0 0 0 0 0 ... 0	0 0 0 0 0 0 0 0 0 0 ... 1
.	44	1	0	O61.1	0 0 0 0 0 0 0 ... 0	0 0 0 0 0 0 0 1 0 0 ... 0
.	61	3	0	B63.9	0 0 0 0 1 0 0 ... 0	0 0 0 0 0 0 0 1 0 0 ... 0
.	29	1	0	A02.9	0 1 0 0 0 0 0 ... 0	0 0 0 0 0 1 0 0 0 0 ... 0
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
n	.	.	.	.	.	.

# Sensor Data

***Physiological parameters:*** This data are often collected for critically ill patients in intensive care units (ICU). Examples of physiological parameters are temperature, blood pressure, oxy-gen saturation, respiration rate, and heart rate.

***Electrocardiogram (ECG):*** The recording of the electrical activity of the heart over a period of time. ECG translates impulses generated by the polarization and depolarization of cardiact issue into a waveform, which are analyzed for the detection of arrhythmias and heart-related disorders.

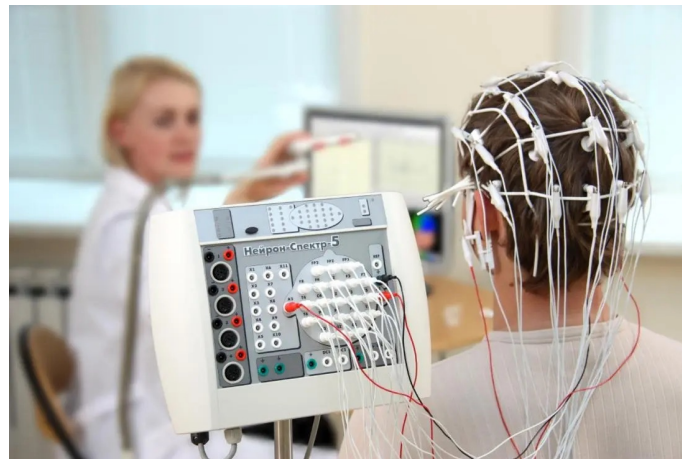
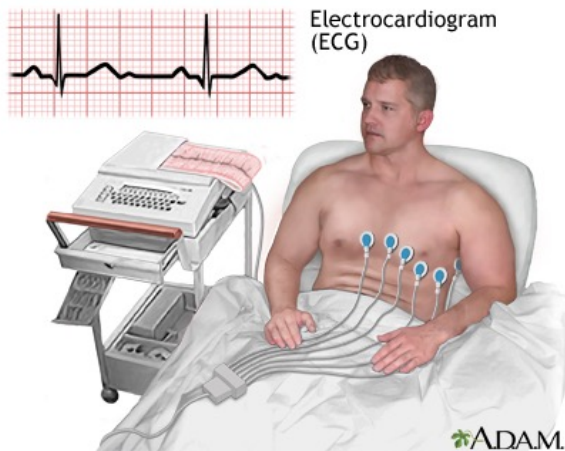
***Electroencephalogram (EEG):*** The recording of the brain's spontaneous electrical activity over a period of time (typically 20–30 minutes). The activity is detected by electrodes placed on the scalp. EEG is often used for studying neurological disorders and assessing cognitive functions.



# Sensor Data

Sensor data are usually represented as numeric time series (the events are homogenous) that are regularly measured in time at a high frequency (e.g., physiological parameters are typically recorded at few Hz while EEG are recorded at several kHz).

Sensor data for a specific subject are measured over a much shorter period of time (usually several minutes to several days) compared to the longitudinal EHR data (collected across the life span).



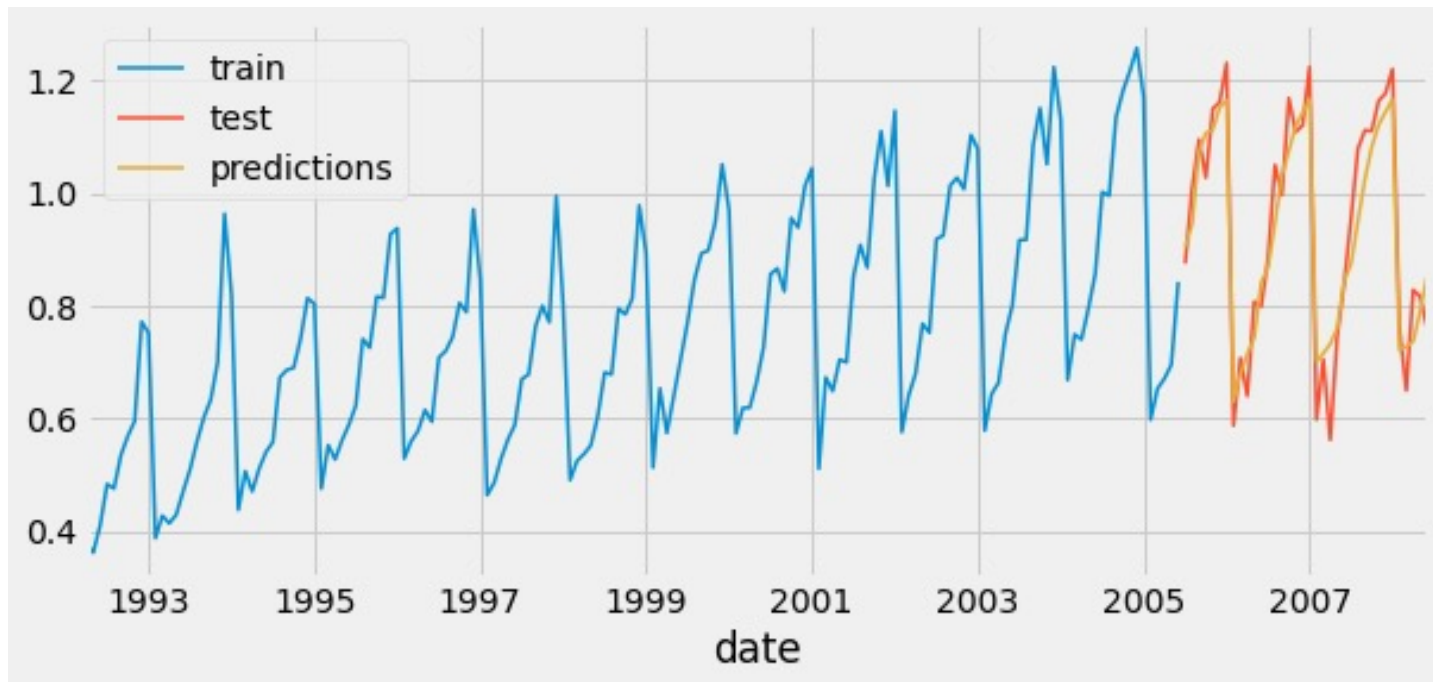
# Time Series Forecasting

**Autoregression** is a time series model that uses observations from previous time steps as input to a regression equation to predict the value at the next time step.

# Machine learning for forecasting

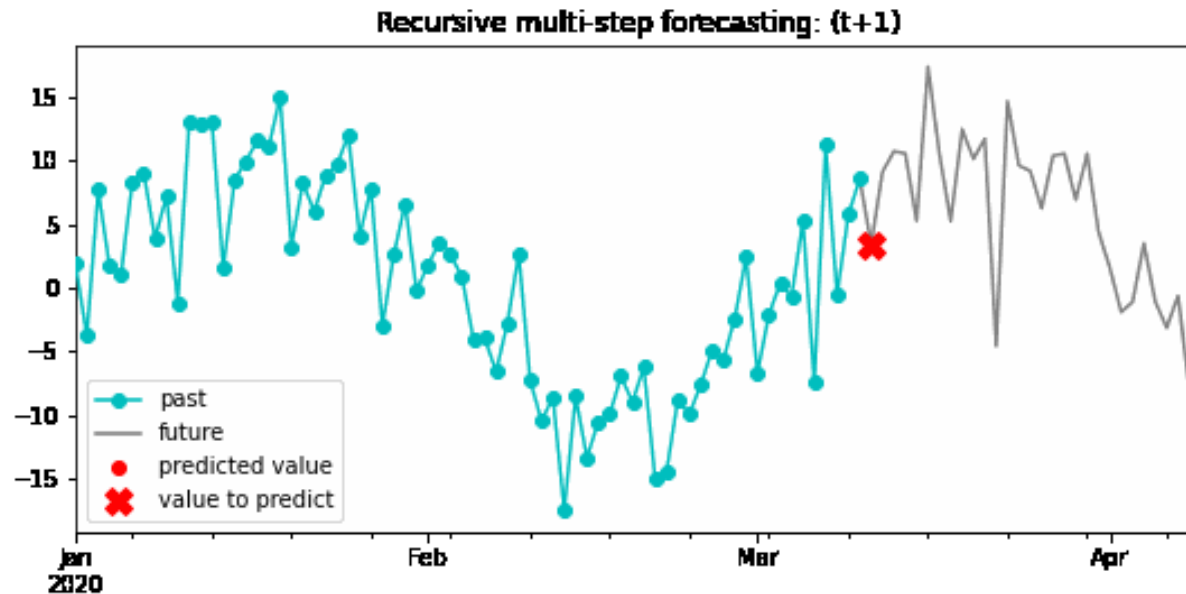
A time series is available with the monthly expenditure (millions of dollars) on corticosteroid drugs that the Australian health system had between 1991 and 2008.

It is intended to create an autoregressive model capable of predicting future monthly expenditures.



## Time series forecasting

A time series is a succession of chronologically ordered data spaced at equal or unequal intervals. The forecasting process consists of predicting the future value of a time series, either by modeling the series solely based on its past behavior (autoregressive) or by using other external variables.



An autoregression model makes an assumption that the observations at previous time steps are useful to predict the value at the next time step.

# Data Preparation

In order to apply machine learning models to forecasting problems, the time series has to be transformed into a matrix in which each value is related to the time window (lags) that precedes it.

In a time series context, a lag with respect to a time step  $t$  is defined as the values of the series at previous time steps. For example, lag 1 is the value at time step  $t-1$  and lag  $m$  is the value at time step  $t-m$ .

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

*Time series transformation into a matrix of 5 lags and a vector with the value of the series that follows each row of the matrix.*

**5 lag time window – A hyperparameter!**

# Data Preparation

This type of transformation also allows to include additional variables.

Time series										X						y
1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	f	6
										2	3	4	5	6	g	7
										3	4	5	6	7	h	8
										4	5	6	7	8	i	9
										5	6	7	8	9	j	10

*Time series transformation including an exogenous variable.*

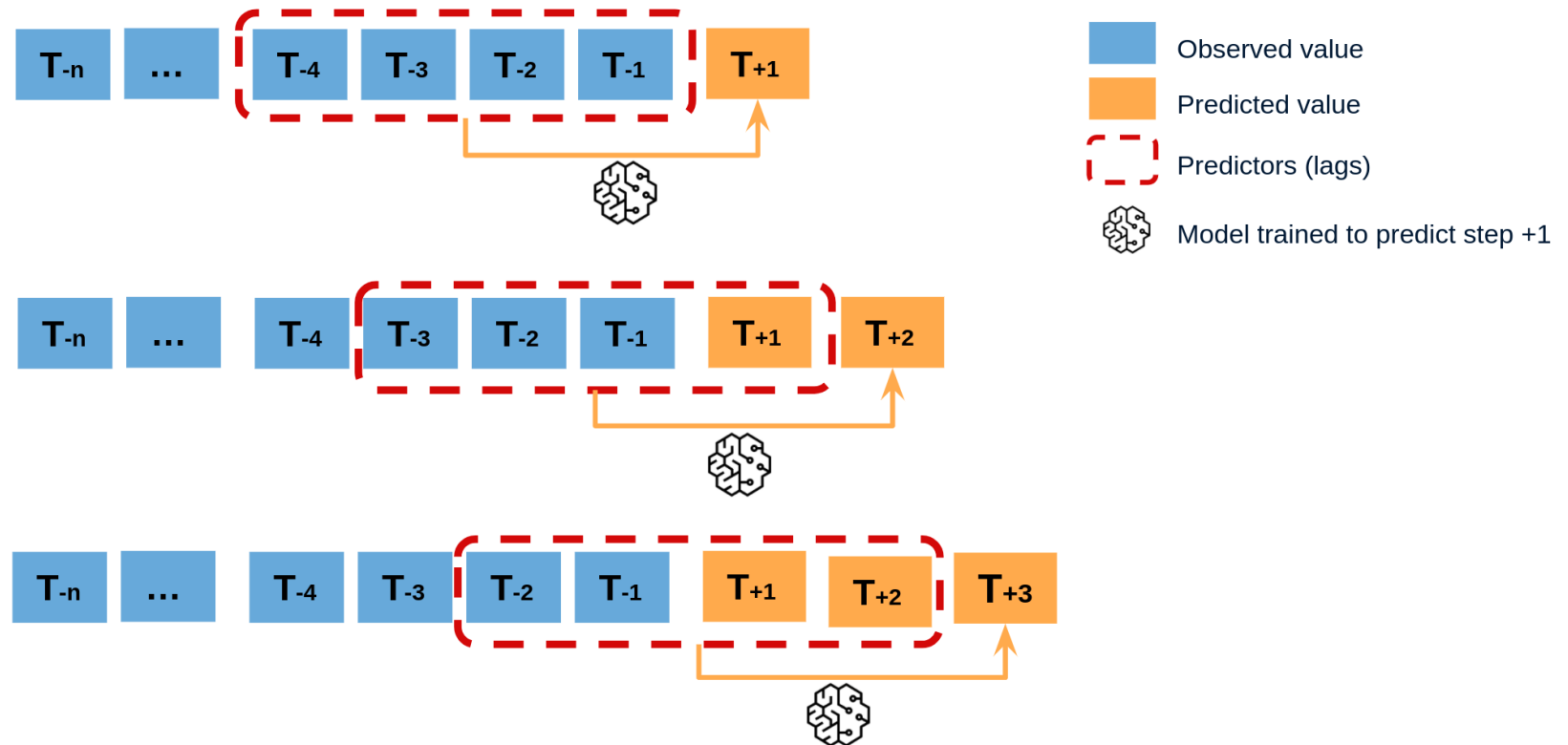
Once data have been rearranged into the new shape, any regression model can be trained to predict the next value (step) of the series.

During model training, every row is considered a separate data instance, where values at lags 1, 2, ...  $p$  are considered as features to predict the target quantity of the time series at time step  $p+1$ .

# Multi-Step Time Series Forecasting

When working with time series, it is seldom needed to predict only the next element in the series ( $t+1$ ). Instead, the most common goal is to predict a whole future interval ( $t+1$ ), ..., ( $t+n$ ) or a far point in time ( $t+n$ ).

## Recursive multi-step forecasting

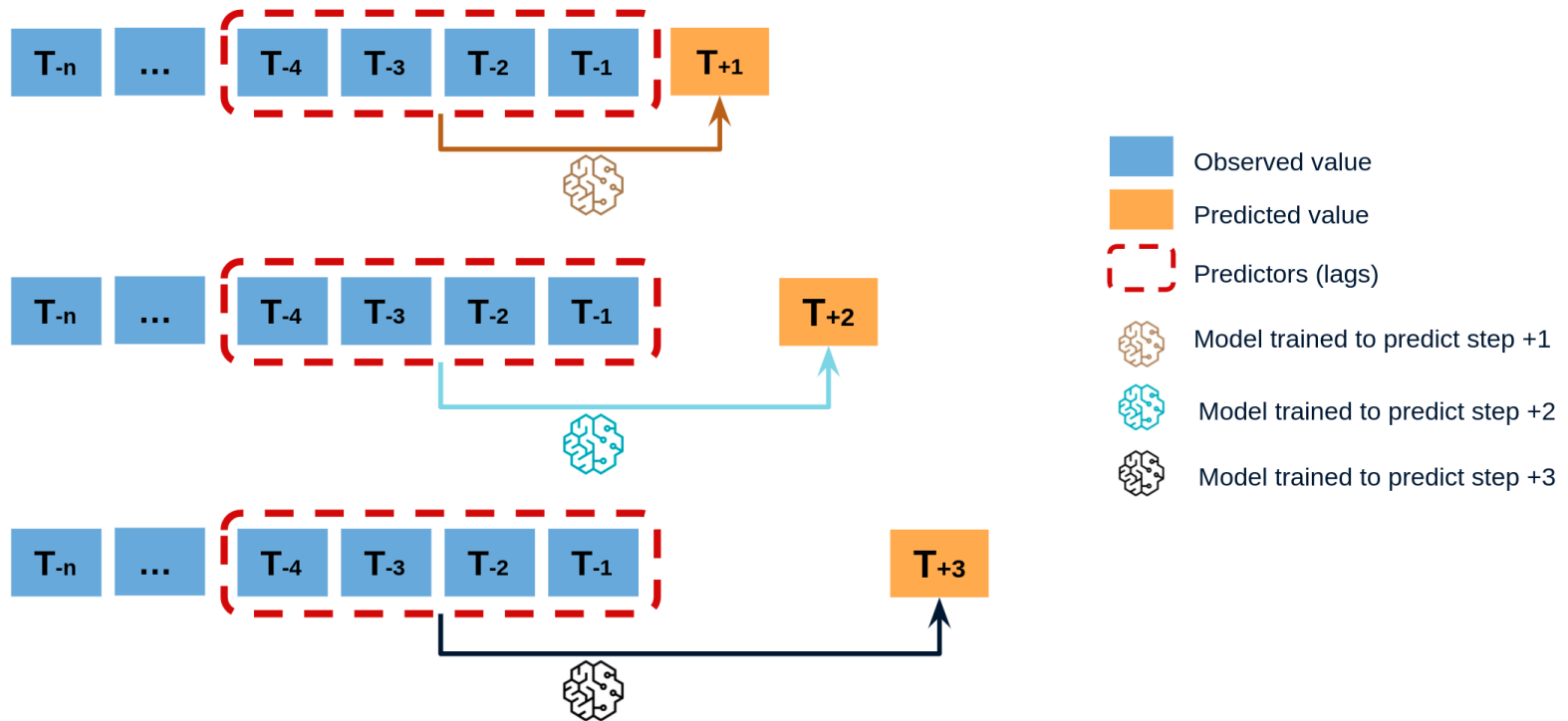


*Recursive multi-step prediction process diagram to predict 3 steps into the future using the last 4 lags of the series as predictors.*

# Multi-Step Time Series Forecasting

When working with time series, it is seldom needed to predict only the next element in the series ( $t+1$ ). Instead, the most common goal is to predict a whole future interval ( $t+1$ ), ..., ( $t+n$ ) or a far point in time ( $t+n$ ).

## Direct multi-step forecasting



*Direct multi-step prediction process diagram to predict 3 steps into the future using the last 4 lags of the series as predictors.*



# Regression

Time series									
1	2	3	4	5	6	7	8	9	10

Exogenous variable									
a	b	c	d	e	f	g	h	i	j

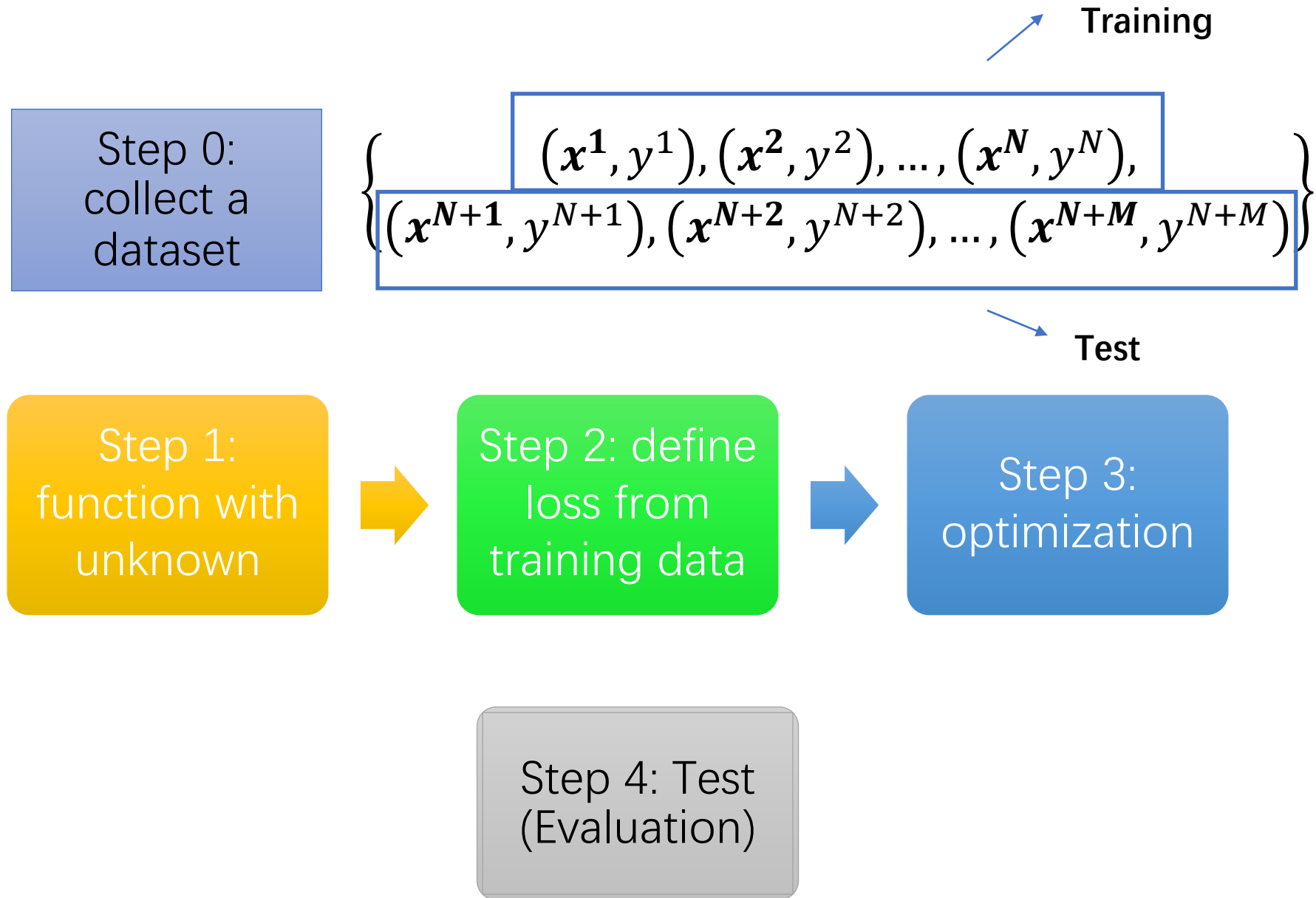
X						y
1	2	3	4	5	f	6
2	3	4	5	6	g	7
3	4	5	6	7	h	8
4	5	6	7	8	i	9
5	6	7	8	9	j	10

Once data have been rearranged into the new shape, any regression model can be trained to predict the next value (step) of the series.

A regression model is required for autoregressive prediction.

There are multiple regression are available in Sklearn, such as linear regression and random forest.

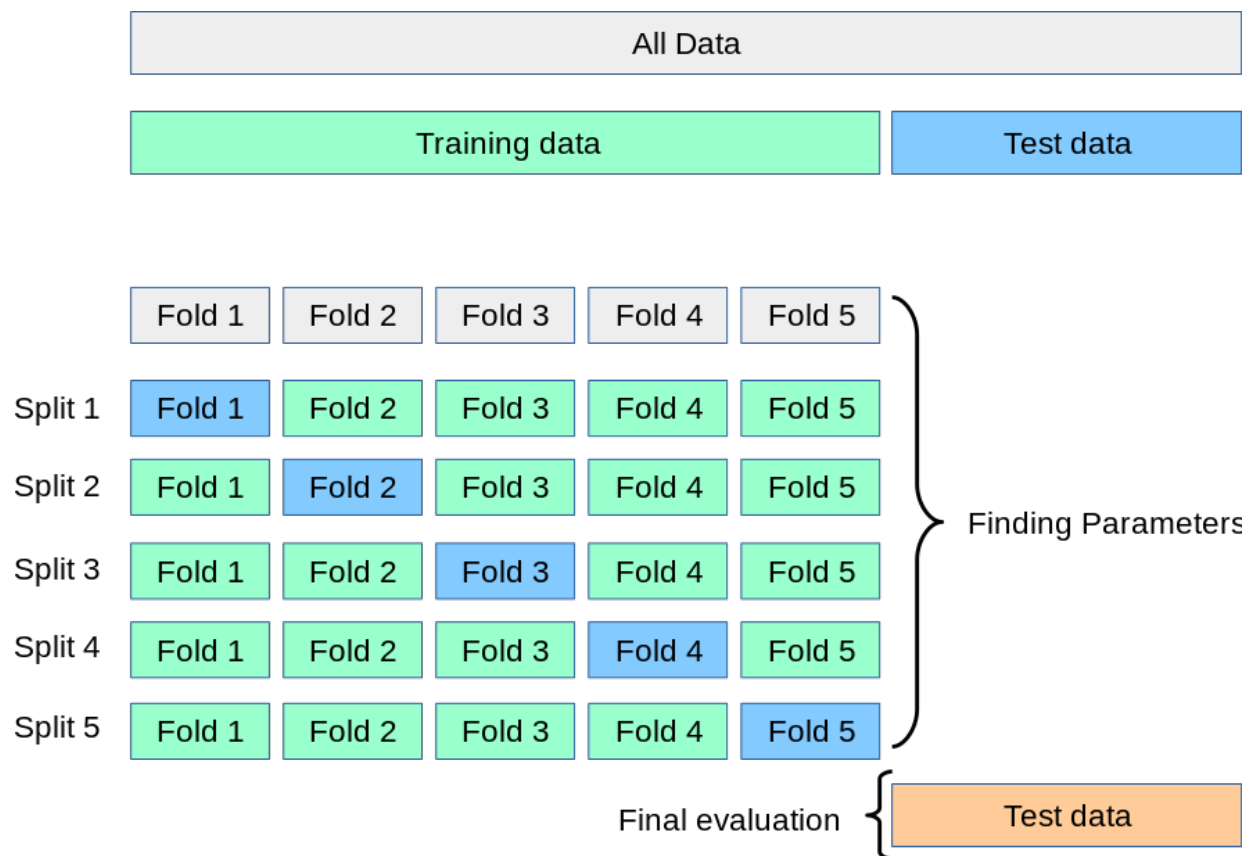
# Machine Learning is so simple .....



# Cross-validation

Training proceeds on the training set, after which evaluation is done on the validation set, and when the experiment seems to be successful, final evaluation can be done on the test set.

However, by partitioning the available data into three sets, we drastically reduce the number of samples which can be used for learning the model, and the results can depend on a particular random choice for the pair of (train, validation) sets.



A model is trained using  $k-1$  of the folds as training data;

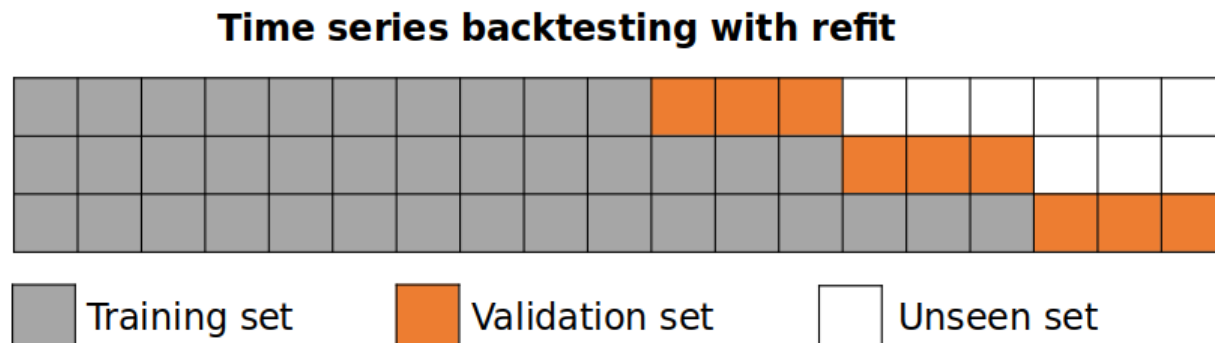
The resulting model is validated on the remaining part of the data.

# Cross-validation of time series forecasting

**Backtesting** is a term used in modeling to refer to testing a predictive model on historical data. Backtesting involves moving backward in time, step-by-step, in as many stages as is necessary. Therefore, it is a special type of cross-validation applied to previous period(s).

## 1. Backtesting with refit and increasing training size (fixed origin)

The model is trained each time before making predictions. With this configuration, the model uses all the data available so far. It is a variation of the standard cross-validation but, instead of making a random distribution of the observations, the training set increases sequentially, maintaining the temporal order of the data.

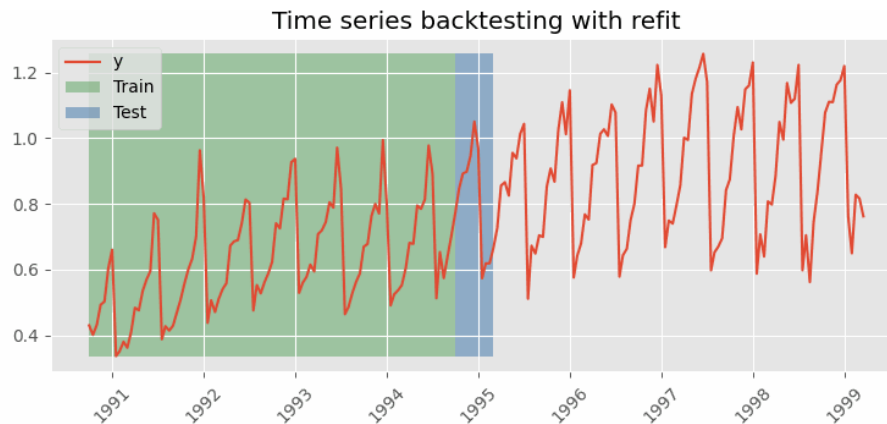


# Cross-validation of time series forecasting

**Backtesting** is a term used in modeling to refer to testing a predictive model on historical data. Backtesting involves moving backward in time, step-by-step, in as many stages as is necessary. Therefore, it is a special type of cross-validation applied to previous period(s).

## 1. Backtesting with refit and increasing training size (fixed origin)

The model is trained each time before making predictions. With this configuration, the model uses all the data available so far. It is a variation of the standard cross-validation but, instead of making a random distribution of the observations, the training set increases sequentially, maintaining the temporal order of the data.

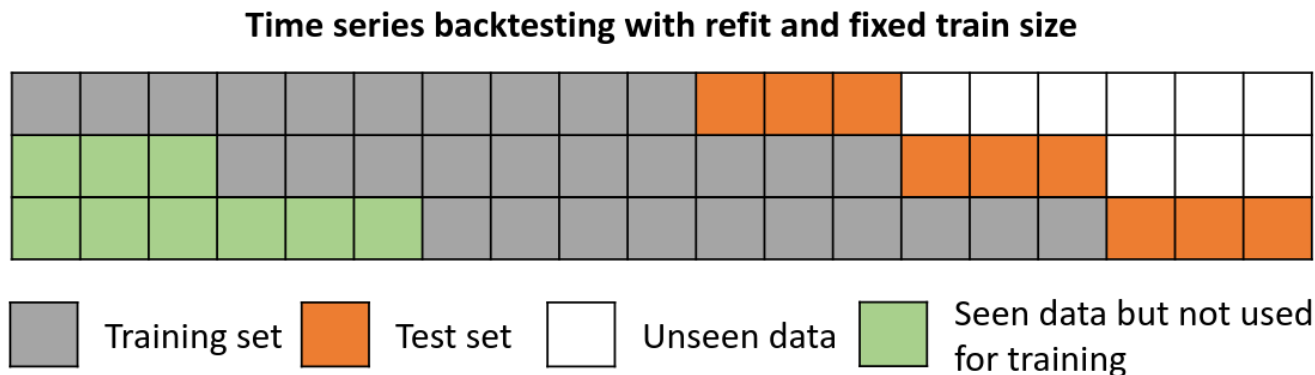


# Cross-validation of time series forecasting

**Backtesting** is a term used in modeling to refer to testing a predictive model on historical data. Backtesting involves moving backward in time, step-by-step, in as many stages as is necessary. Therefore, it is a special type of cross-validation applied to previous period(s).

## 2. Backtesting with refit and fixed training size (rolling origin)

A technique similar to the previous one but, in this case, the forecast origin rolls forward, therefore, the size of training remains constant. This is also known as time series cross-validation or walk-forward validation.

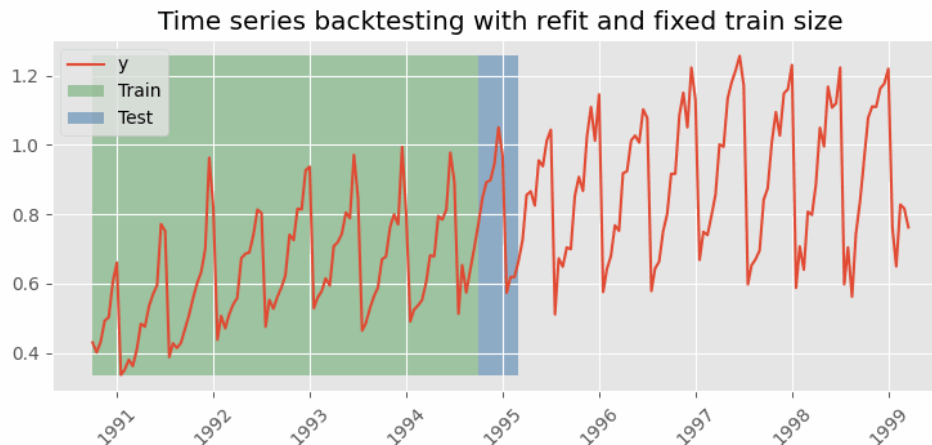


# Cross-validation of time series forecasting

**Backtesting** is a term used in modeling to refer to testing a predictive model on historical data. Backtesting involves moving backward in time, step-by-step, in as many stages as is necessary. Therefore, it is a special type of cross-validation applied to previous period(s).

## 2. Backtesting with refit and fixed training size (rolling origin)

A technique similar to the previous one but, in this case, the forecast origin rolls forward, therefore, the size of training remains constant. This is also known as time series cross-validation or walk-forward validation.

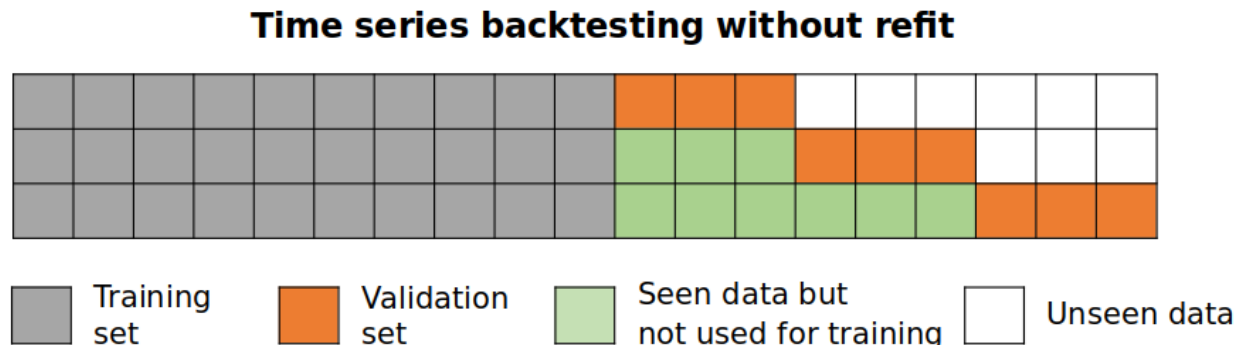


# Cross-validation of time series forecasting

**Backtesting** is a term used in modeling to refer to testing a predictive model on historical data. Backtesting involves moving backward in time, step-by-step, in as many stages as is necessary. Therefore, it is a special type of cross-validation applied to previous period(s).

## 3. Backtesting without refit

After an initial train, the model is used sequentially without updating it and following the temporal order of the data. This strategy has the advantage of being much faster since the model is trained only once. However, the model does not incorporate the latest data available, so it may lose predictive capacity over time.



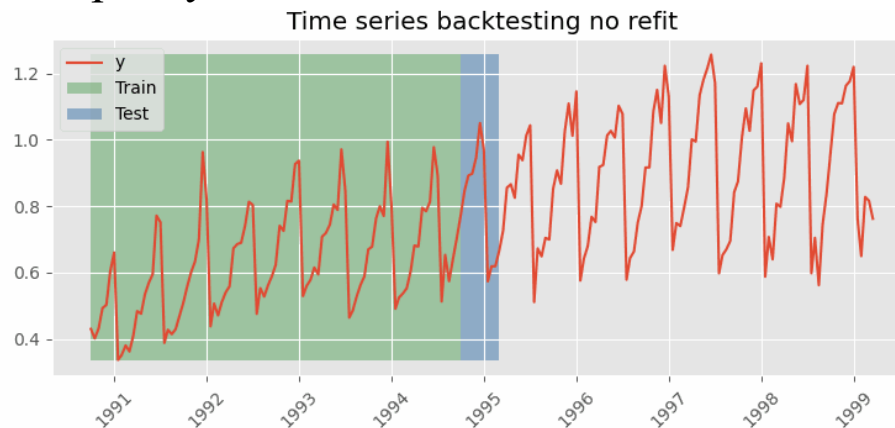


# Cross-validation of time series forecasting

**Backtesting** is a term used in modeling to refer to testing a predictive model on historical data. Backtesting involves moving backward in time, step-by-step, in as many stages as is necessary. Therefore, it is a special type of cross-validation applied to previous period(s).

## 3. Backtesting without refit

After an initial train, the model is used sequentially without updating it and following the temporal order of the data. This strategy has the advantage of being much faster since the model is trained only once. However, the model does not incorporate the latest data available, so it may lose predictive capacity over time.



# Time series analysis of EHRs

Recurrent neural network (RNN) models convert clinical event sequences and related time-stamped data into pathways relevant to early detection of disease.

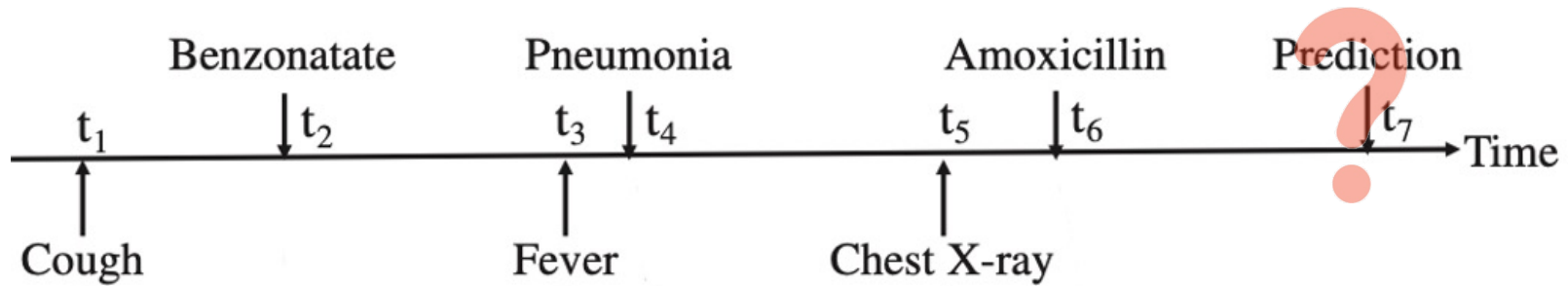
## EHRs

Before diagnosis of a disease, an individual's progression mediated by **pathophysiologic changes** distinguishes those who will eventually get the disease from those who will not.

Detection of **temporal event sequences** that reliably distinguish disease cases from controls may be particularly useful in improving predictive model performance.

EHR data are highly complex, given the structure and breadth of information captured (spanning provider behavior, care utilization, treatment pathways, and patient disease state) and irregular sampling frequency.

# EHRs

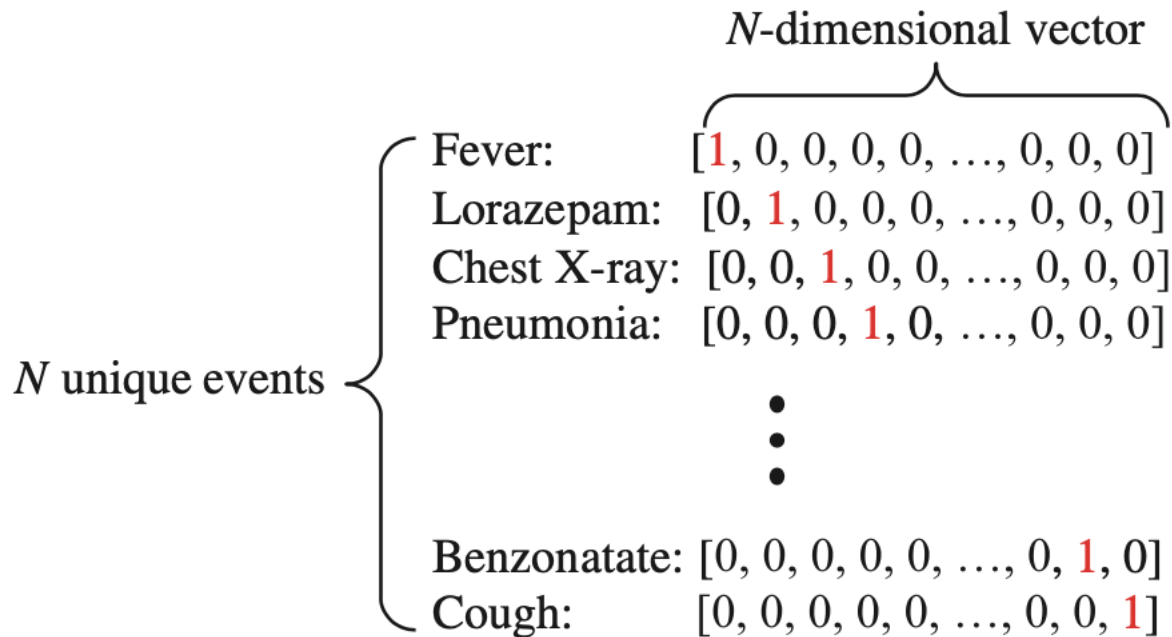


Most predictive modelling work using EHR data rely on aggregate features (e.g., event count and event average).

Temporal relations among disaggregated features (e.g., medication ordered at one time and procedure performed at another) are not captured using these methods.

# Represent clinical events in EHR data

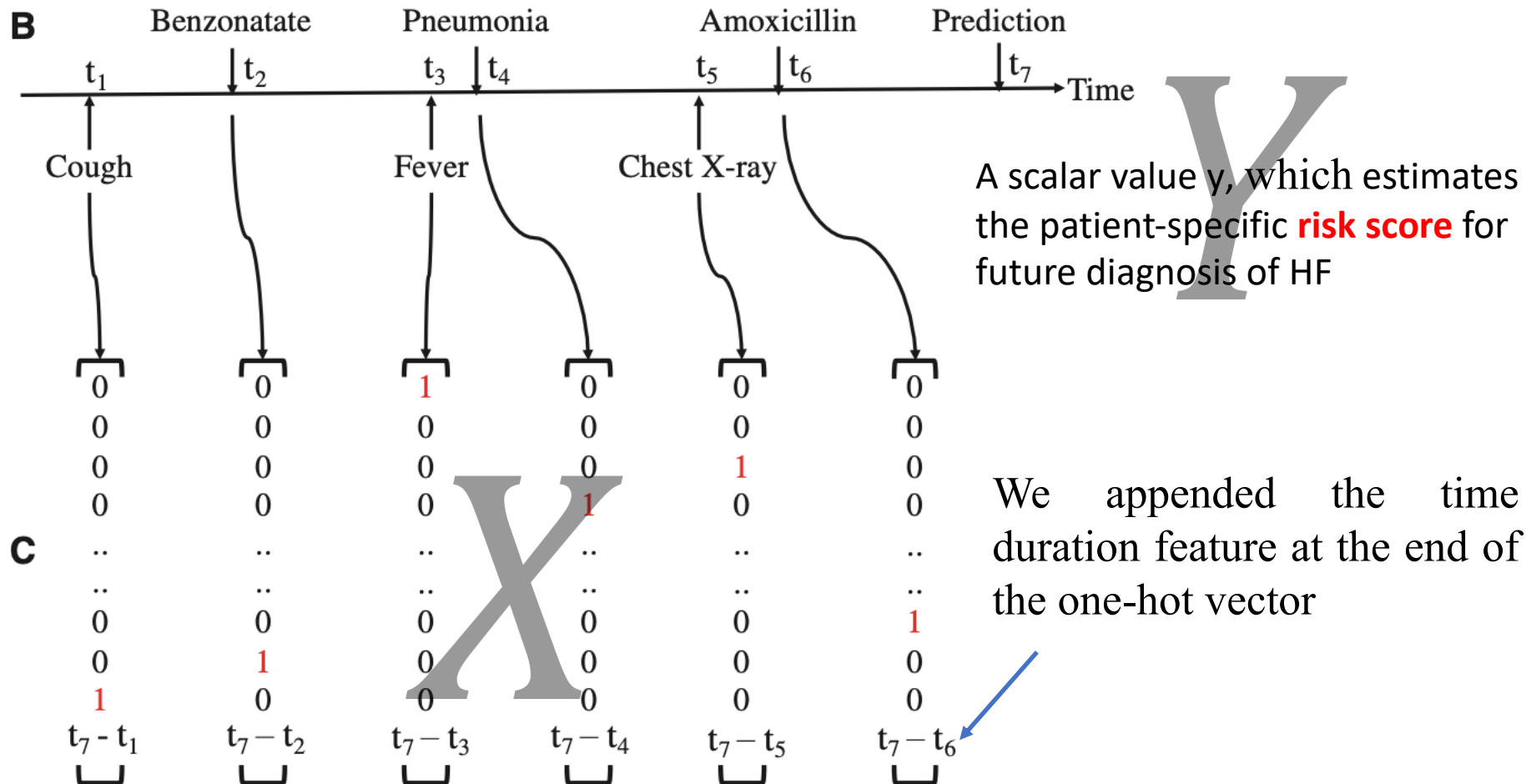
To represent clinical events in EHR data as computable event sequences, we adopted the one-hot vector format.



*Each of the  $N$  unique clinical events was represented as an  $N$ -dimensional vector, where one dimension is set to 1 and the rest are 0.*

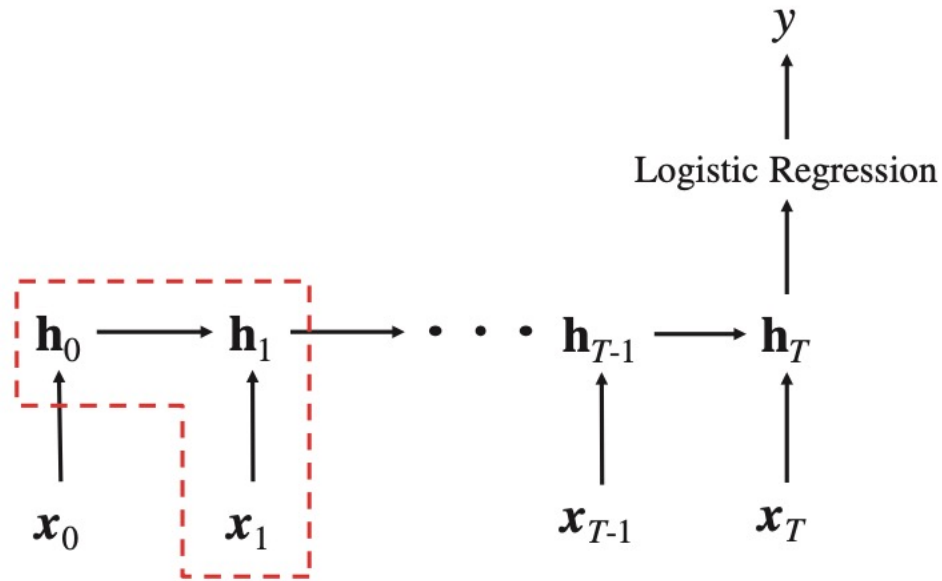
# Data preparation

Using these one-hot vectors, a sequence of clinical events can be converted to a sequence of one-hot vectors.



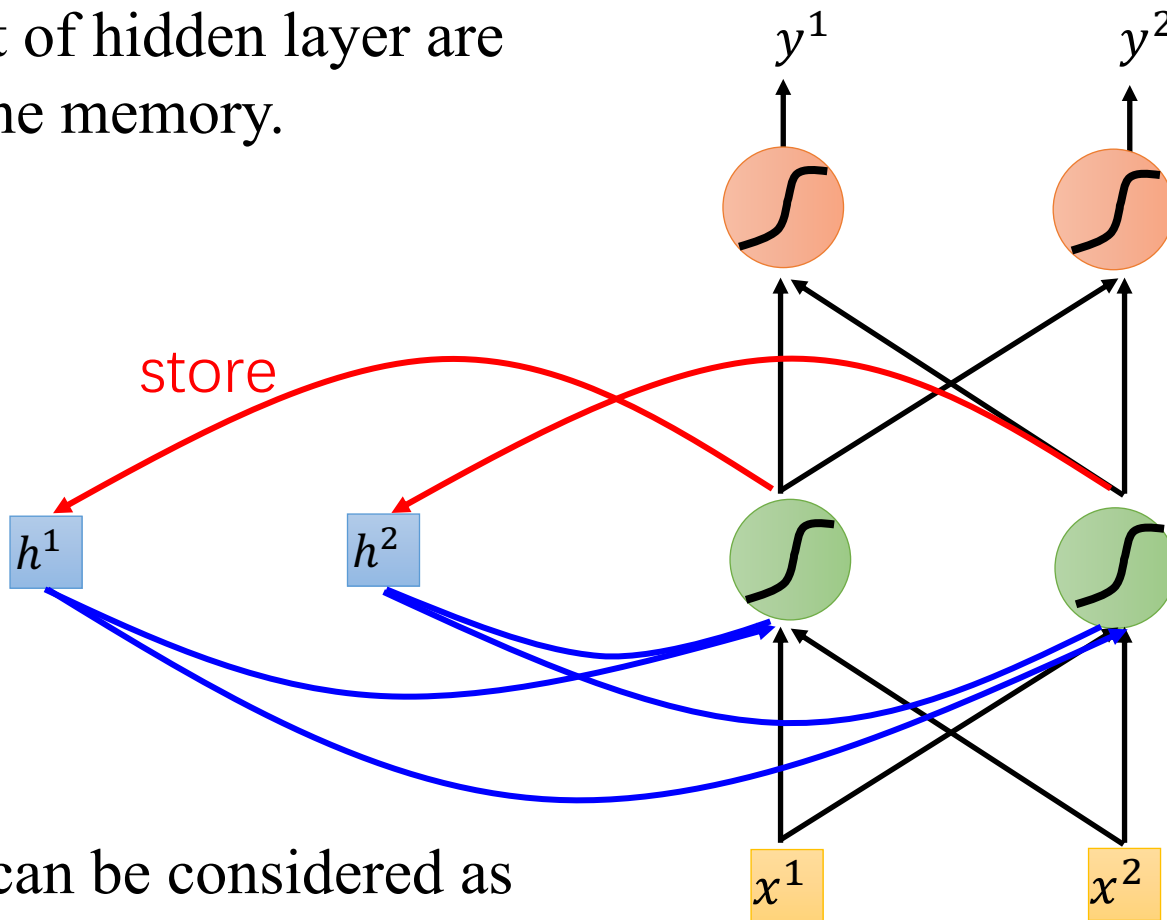
# Recurrent neural network

Given a sequence of clinical visits of length  $T$ , the RNN accepts an input vector  $x_t$  (in our base case, one-hot vectors representing clinical codes) at each timestep  $t$ , while storing information in a single hidden layer  $h$  whose state changes over time ( $h_{t-1}, h_t, h_{t+1}$ ). After seeing the entirety of clinical events, we make the final prediction  $y$  based on the hidden state vector  $h_T$



# Recurrent neural network

The output of hidden layer are stored in the memory.



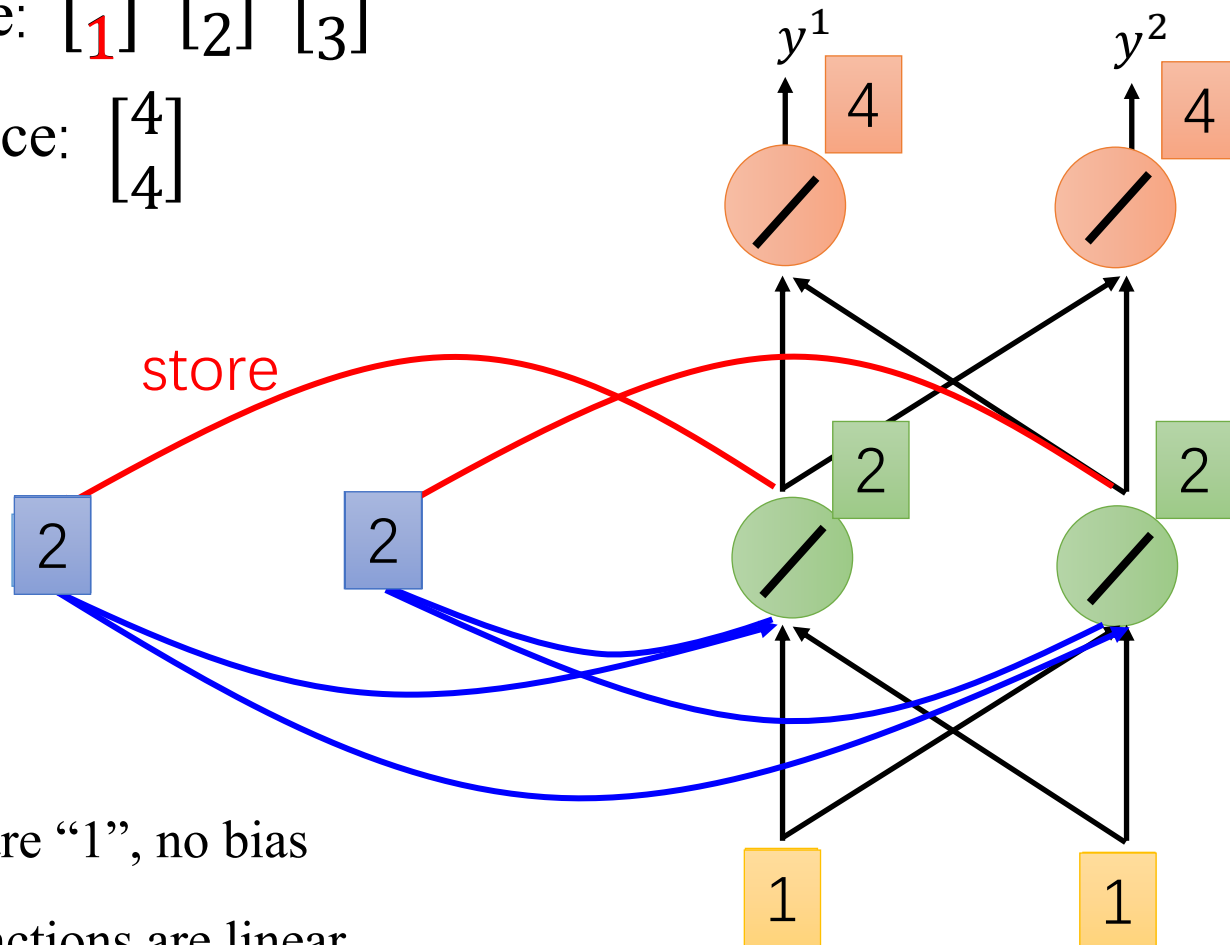
Memory can be considered as another input.



# Recurrent neural network

Input sequence:  $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$   $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$   $\begin{bmatrix} 3 \\ 3 \end{bmatrix}$

Output sequence:  $\begin{bmatrix} 4 \\ 4 \end{bmatrix}$



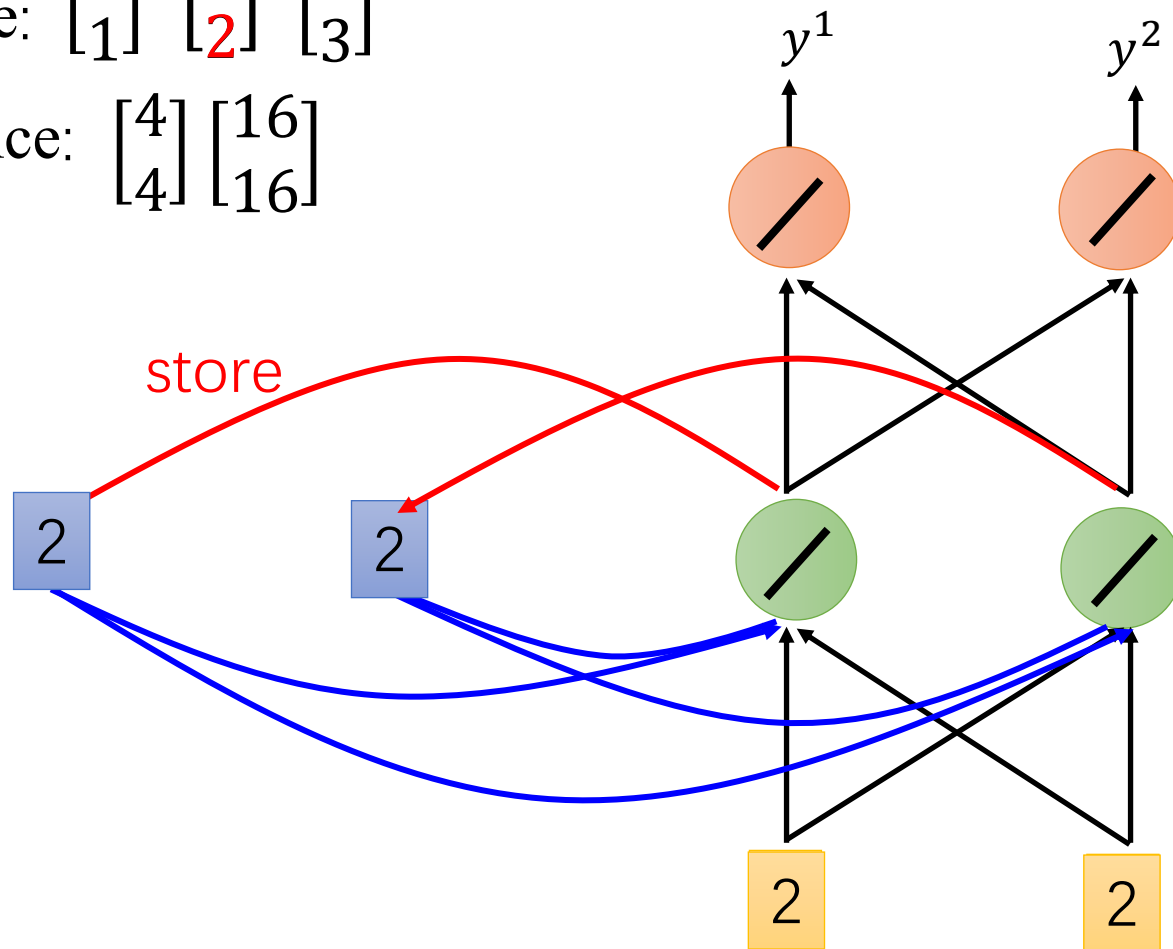
All the weights are “1”, no bias

All activation functions are linear

# Recurrent neural network

Input sequence:  $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$   $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$   $\begin{bmatrix} 3 \\ 3 \end{bmatrix}$

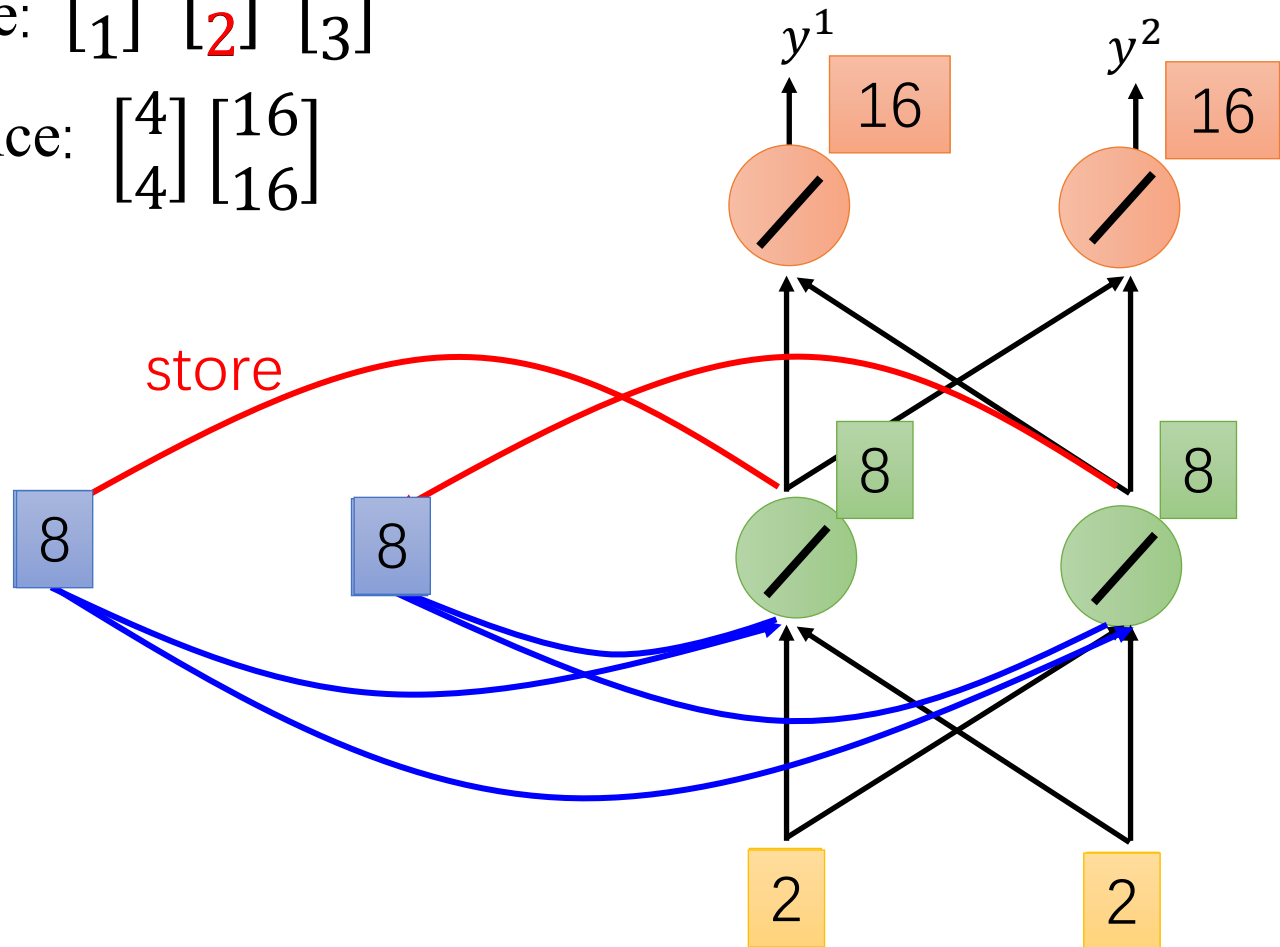
Output sequence:  $\begin{bmatrix} 4 \\ 4 \end{bmatrix}$   $\begin{bmatrix} 16 \\ 16 \end{bmatrix}$



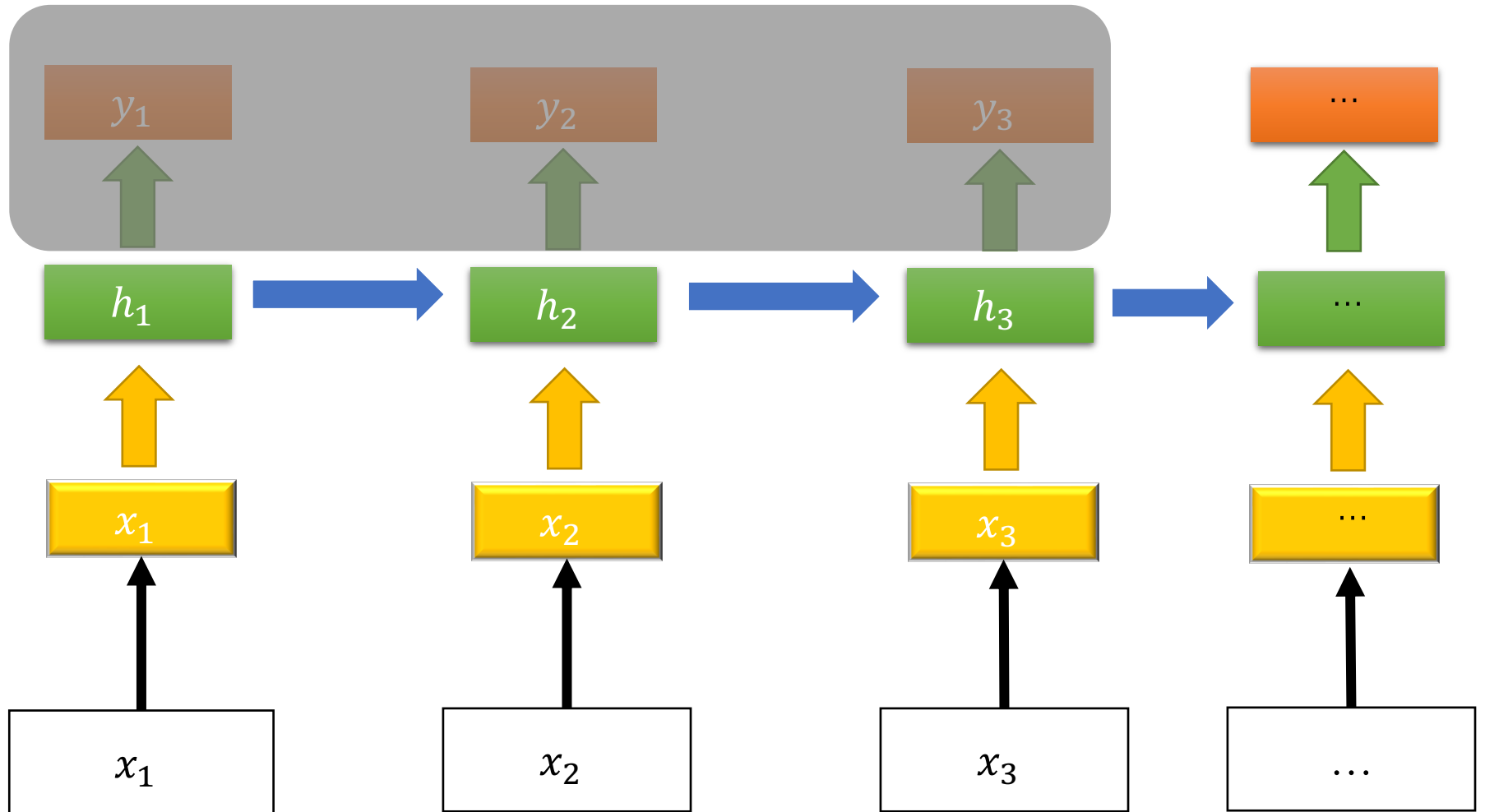
# Recurrent neural network

Input sequence:  $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$   $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$   $\begin{bmatrix} 3 \\ 3 \end{bmatrix}$

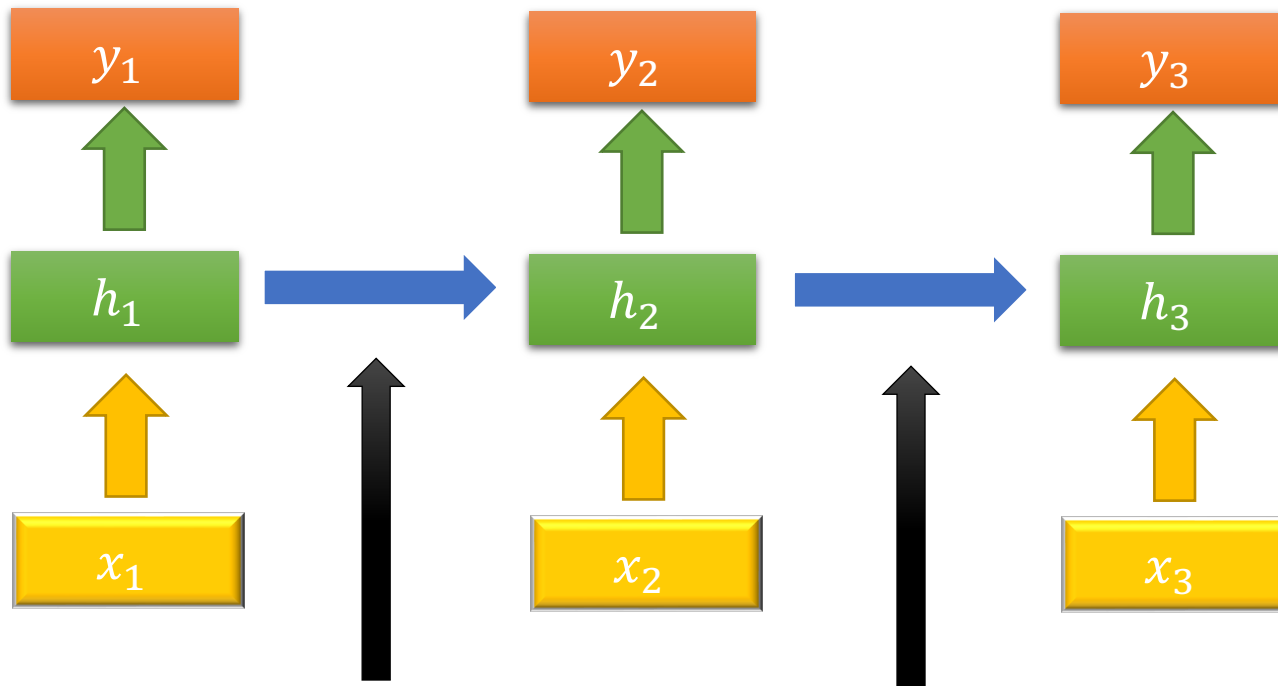
Output sequence:  $\begin{bmatrix} 4 \\ 4 \end{bmatrix}$   $\begin{bmatrix} 16 \\ 16 \end{bmatrix}$



# Recurrent neural network



# Recurrent neural network



Memory can be considered as another input.

# Recurrent neural network

Formally

$x_t$  is the input

$h_t$  is the hidden state

$y_t$  is the output

$h_0 = \vec{0}$

$h_t = \tanh(w_{hh}h_{t-1} + w_{hx}x_t + b_h)$

$y_t = w_{yh}h_t$

