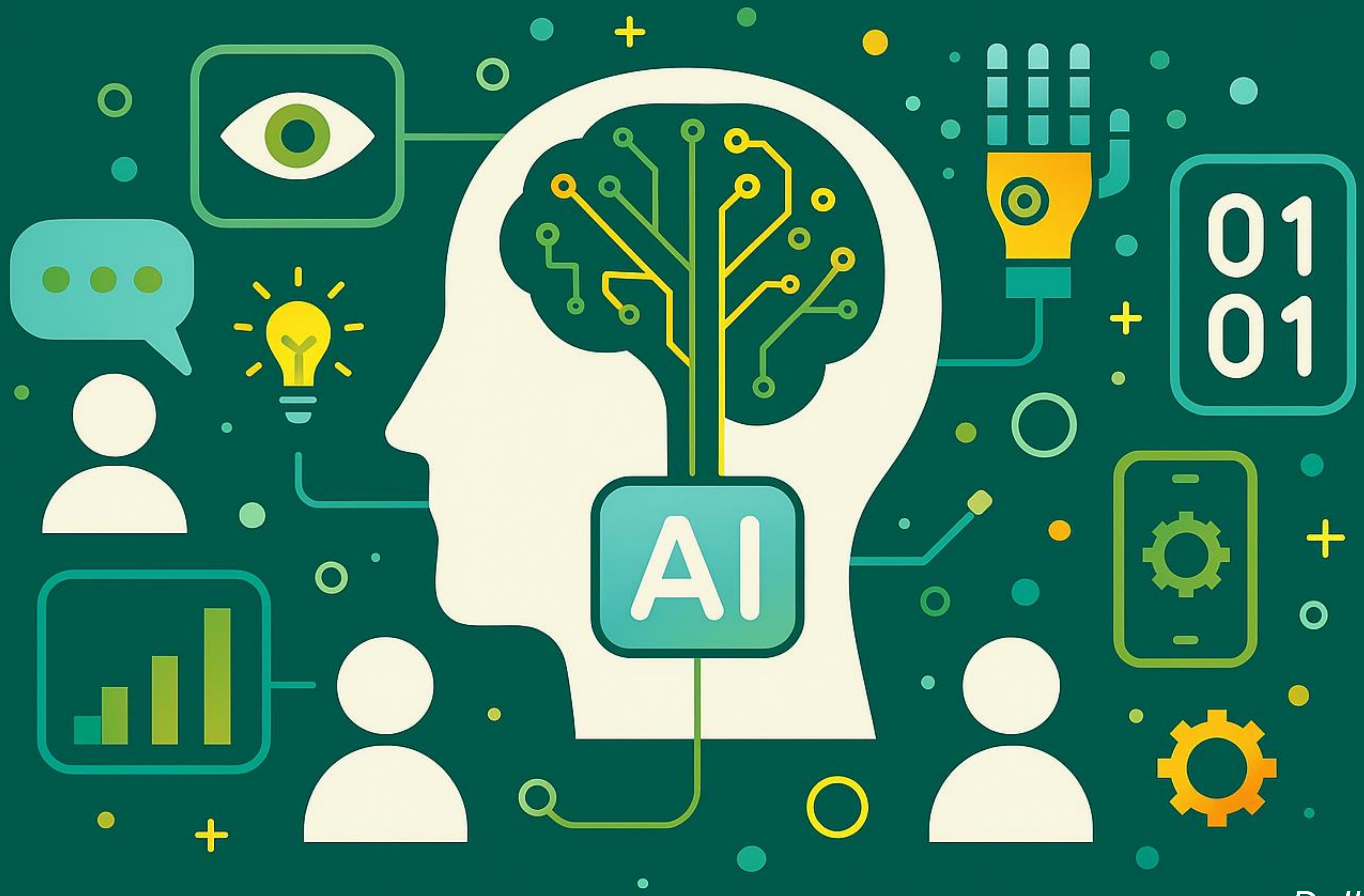
Time series analysis



Dr Ibrahim Alsaggaf

Attribution Requirement

When using, sharing, or adapting this document for any individual, group, or organisation, proper citation of the original author is required. Please cite as follows:

Alsaggaf, I. (2025) *Introduction to Artificial Intelligence*. Available at: https://github.com/ibrahimsaggaf/Introduction-to-Artificial-Intelligence (Accessed: [insert date]).

Content

- Time series data
- Long-Term-Short-Memory (LSTM)
- Q&A

Lab session: Forecasting a multivariate time series

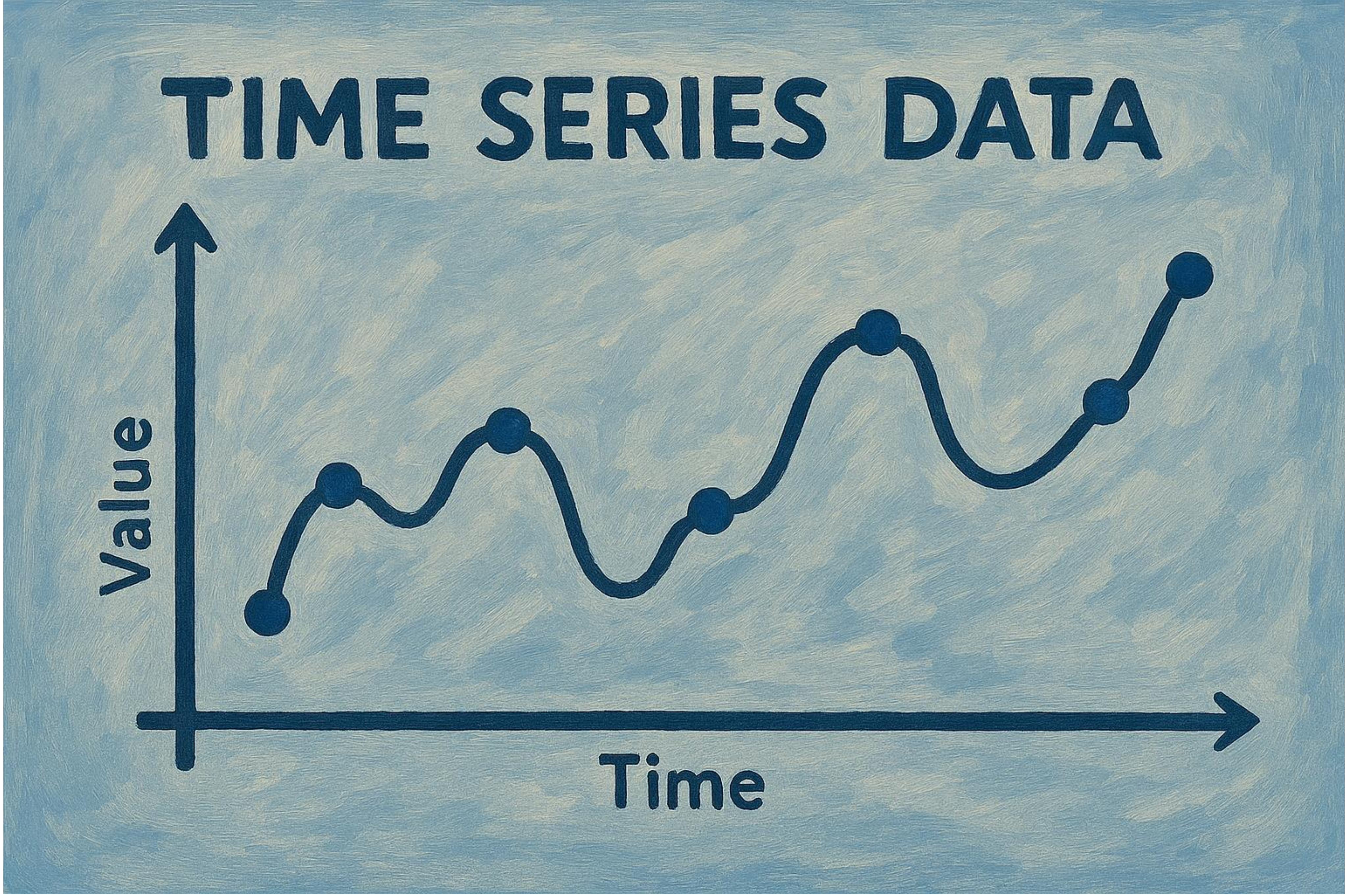
Time series data

1. Is time series data considered a distinct type of data?

2. Does the data change over time?

3. Does the sequence or order of data points in the series matter?

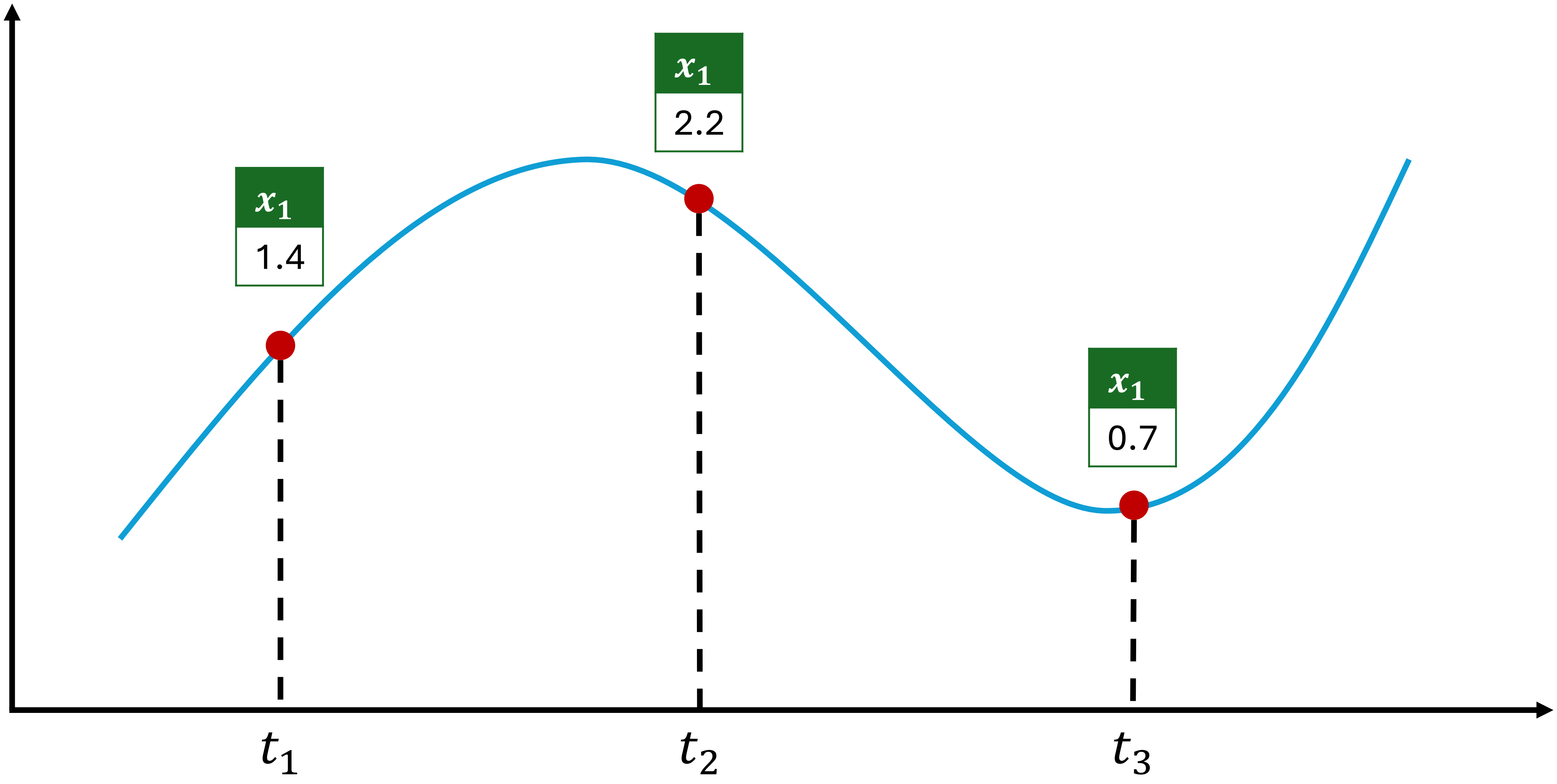
4. Is time series data considered structured or unstructured?



Generated using https://copilot.microsoft.com

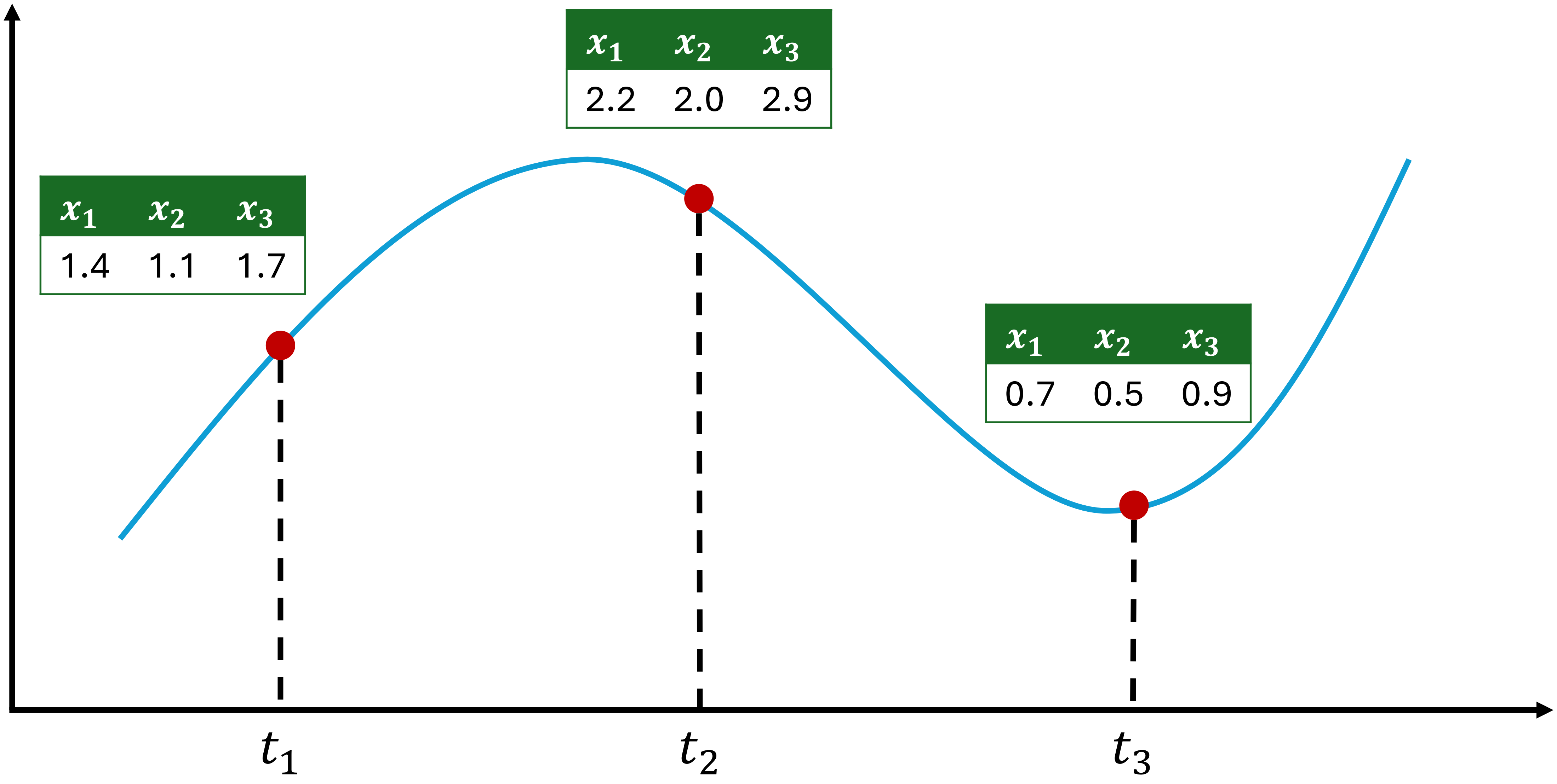
Time series data

Univariate



Time series data

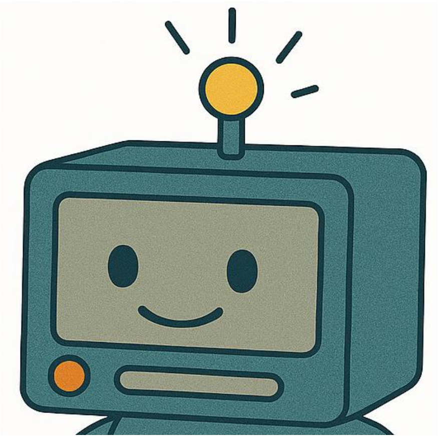
Multivariate



Time series data

Forecasting (Supervised learning)

Past Present



AI agent

?

Future

Forecasting (Supervised learning)

Univariate

Time	x_1
t_1	1.4
t_2	2.2
t_3	0.7
t_4	1.6

$$y \approx h(X)$$

Time series data

Forecasting (Supervised learning)

Univariate

Time	x_1
t_1	1.4
t_2	2.2
t_3	0.7
t_4	1.6

$$\cancel{Y} \approx h(X)$$

Forecasting (Supervised learning)

Univariate

Time	x_1	y
t_1	1.4	2.2
t_2	2.2	0.7
t_3	0.7	1.6
~~t_4~~	~~1.6~~	~~?~~

$$x_1 = \{t_1, \dots, t_{n-1}\}$$
$$y = \{t_2, \dots, t_n\}$$

y is x_1 but shifted by one time step (lag=1).

Vectors concatenation

$\vec{A}$	0.0	1.0	2.0	3.0				
$\vec{B}$	0.5	0.5	0.5	0.5				
$Concat(\vec{A}, \vec{B})$	0.0	1.0	2.0	3.0	0.5	0.5	0.5	0.5

Element-wise addition

$\vec{A}$	0.0	1.0	2.0	3.0
$\vec{B}$	0.5	0.5	0.5	0.5
$\vec{A} + \vec{B}$	0.5	1.5	2.0	3.5

Element-wise product

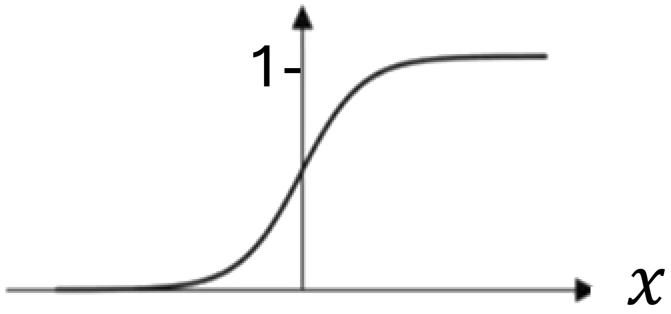
$\vec{A}$	0.0	1.0	2.0	3.0
$\vec{B}$	0.5	0.5	0.5	0.5
$\vec{A} \odot \vec{B}$	0.0	0.5	1.0	1.5

Element-wise product

$\vec{A}$	1.3	2.2	1.7	3.1	
$\vec{B}$	0.1	0.1	0.8	0.8	Control vector
$\vec{A} \odot \vec{B}$	0.13	0.22	1.36	2.48	

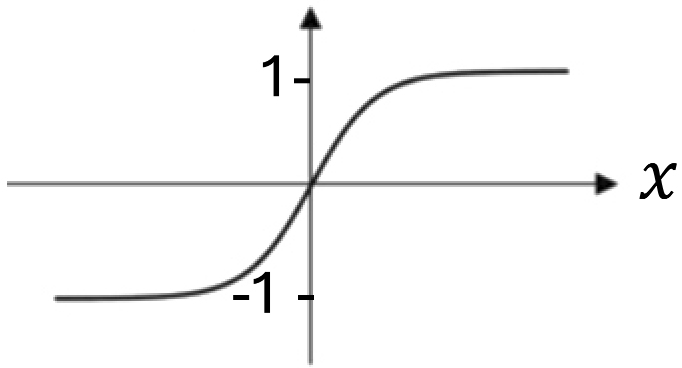
Activation functions

$sigmoid(x)$



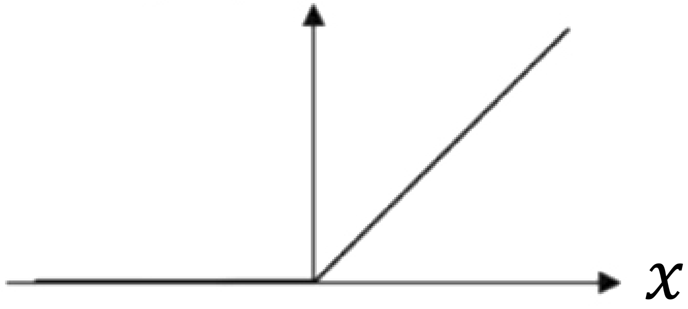
$[0, 1]$

$tanh(x)$



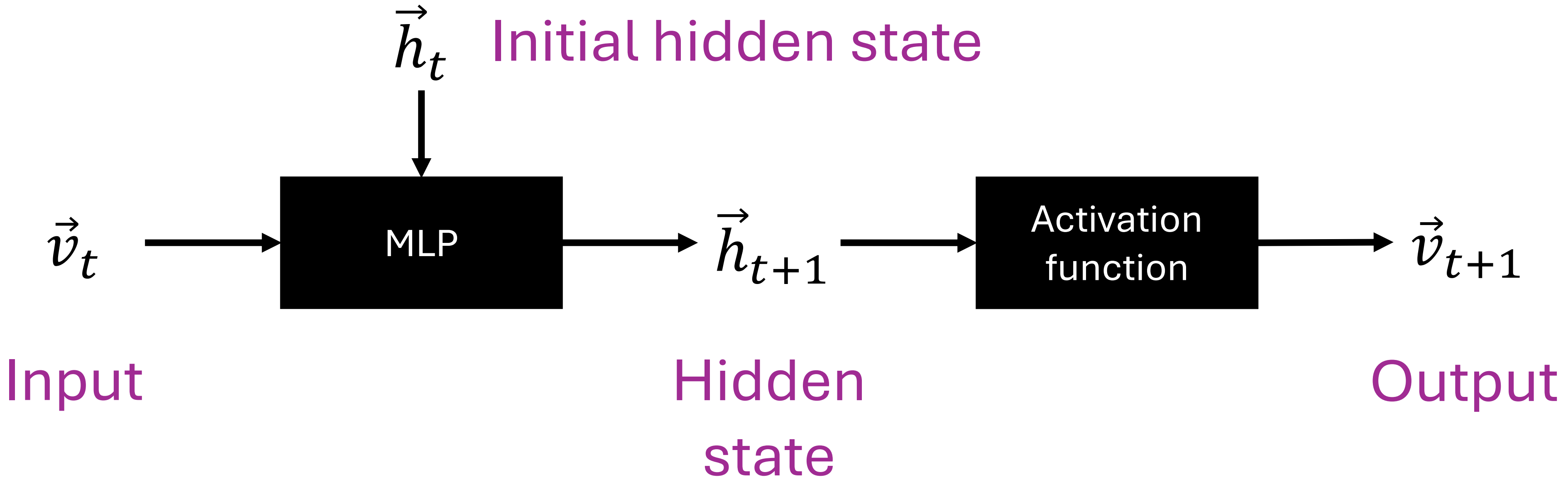
$[-1, 1]$

$relu(x)$

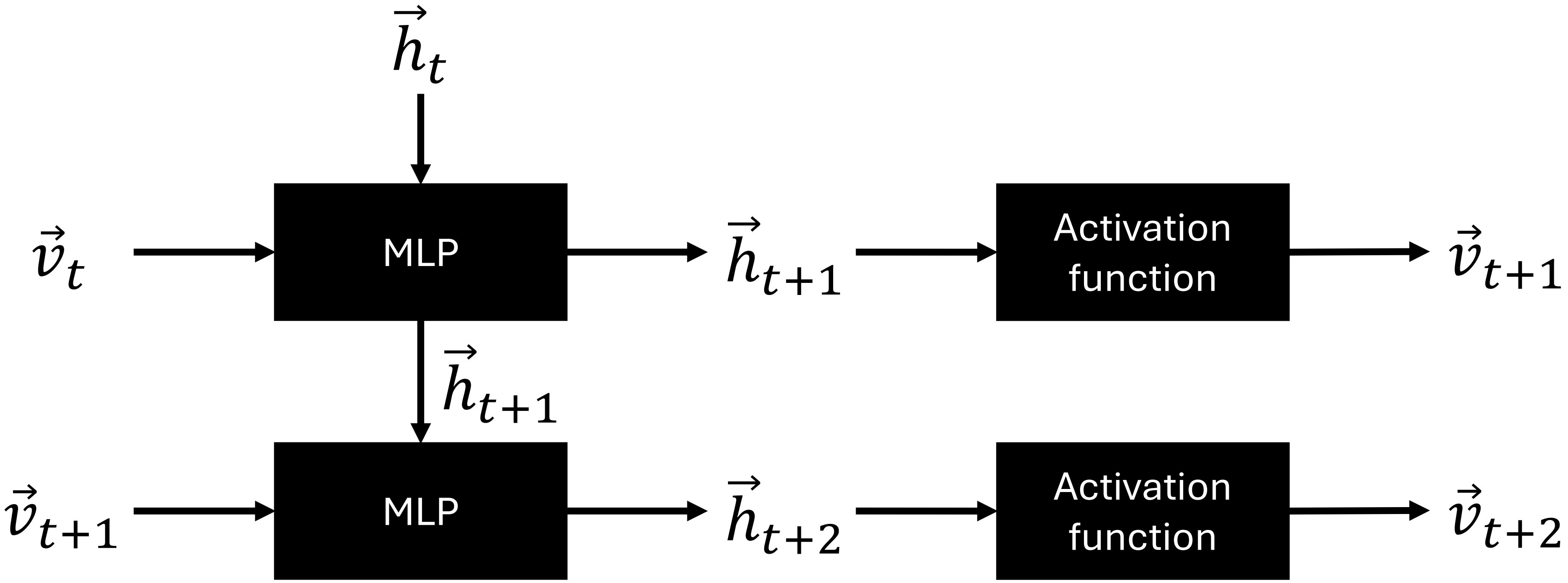


$\max(x, 0)$

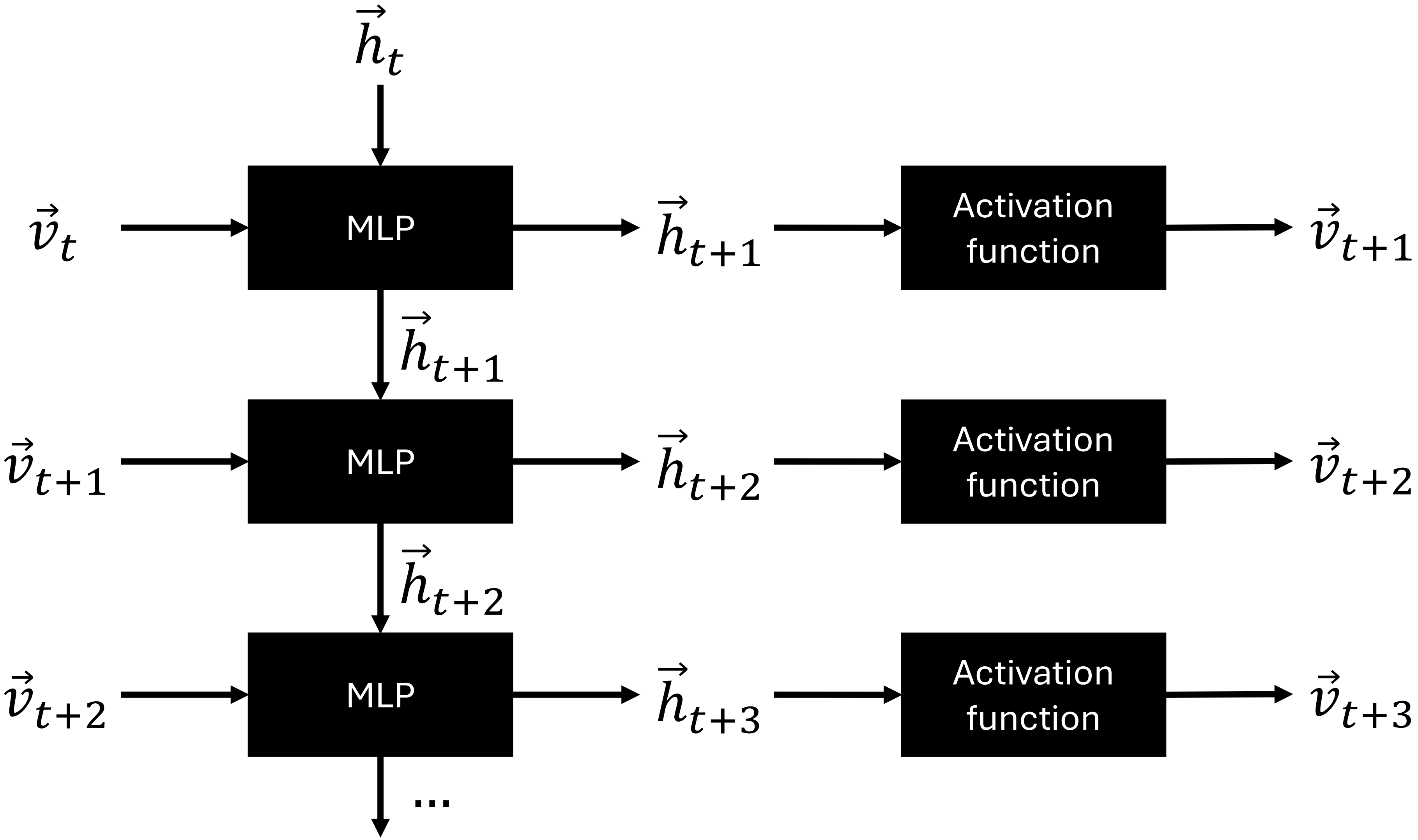
Recurrent Neural Networks



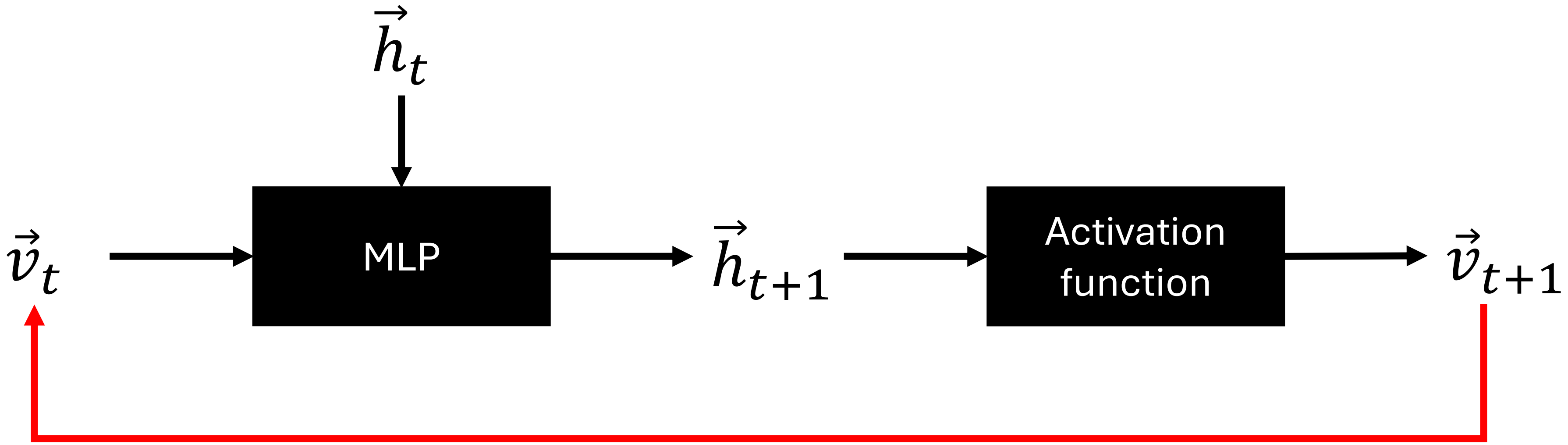
Recurrent Neural Networks



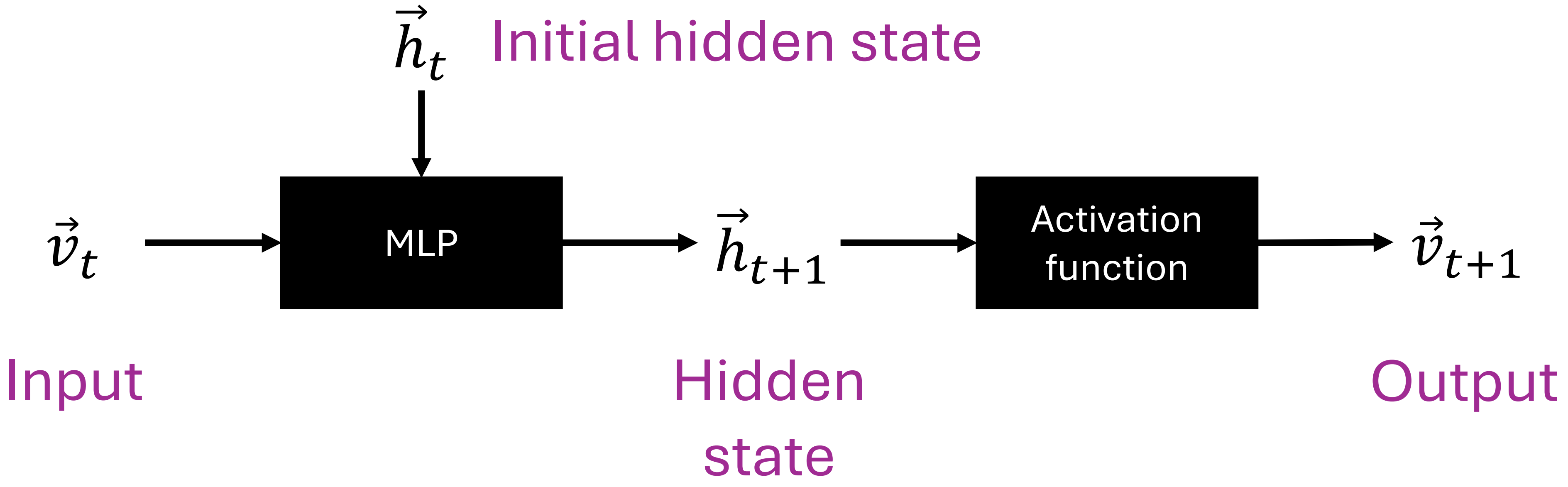
Recurrent Neural Networks



Recurrent Neural Networks

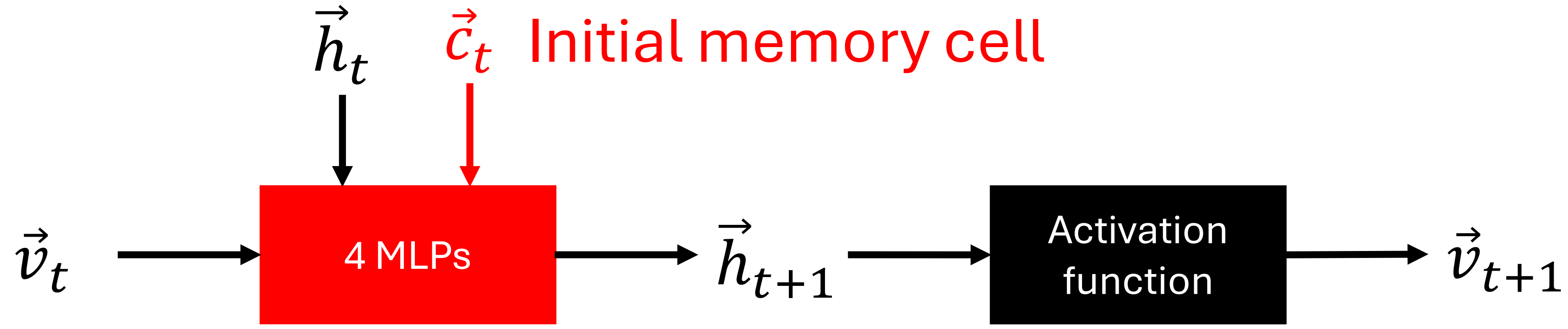


LSTM



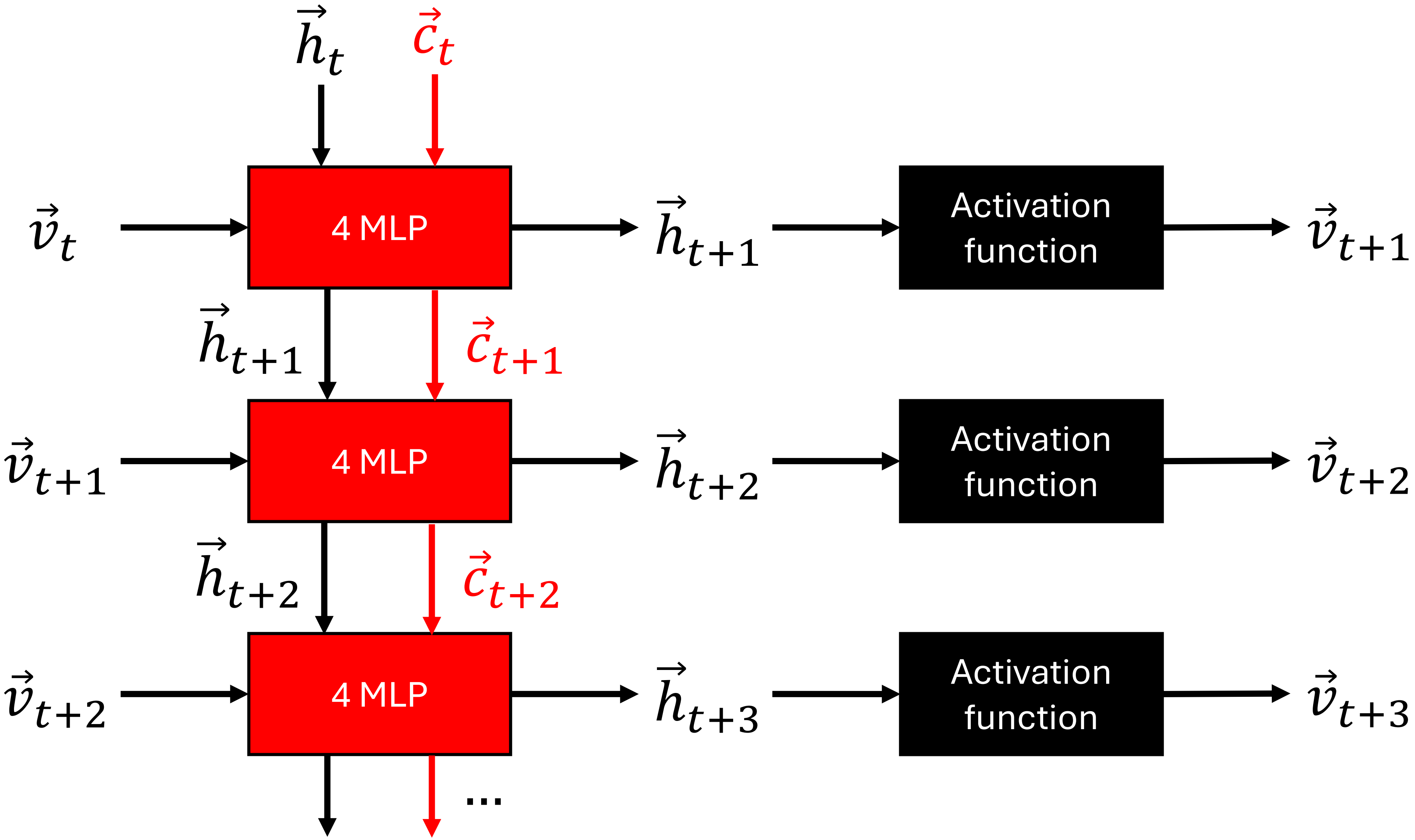
$\vec{h}_t$
Initial hidden state
$\vec{v}_t$
MLP
$\vec{h}_{t+1}$
Activation function
$\vec{v}_{t+1}$
Input
Hidden state
Output

LSTM



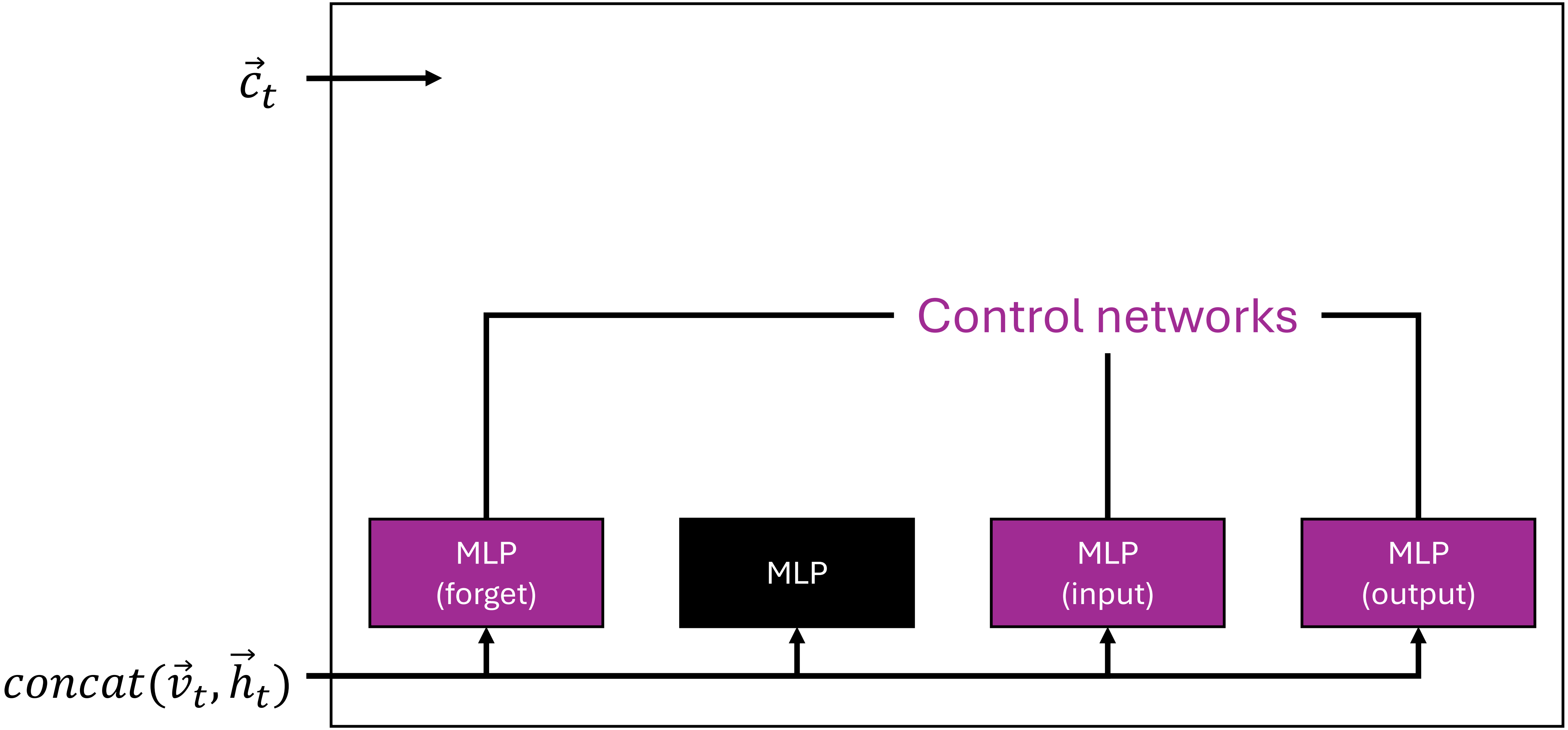
$\vec{h}_t$
$\vec{c}_t$ Initial memory cell
$\vec{v}_t$
4 MLPs
$\vec{h}_{t+1}$
Activation function
$\vec{v}_{t+1}$

LSTM

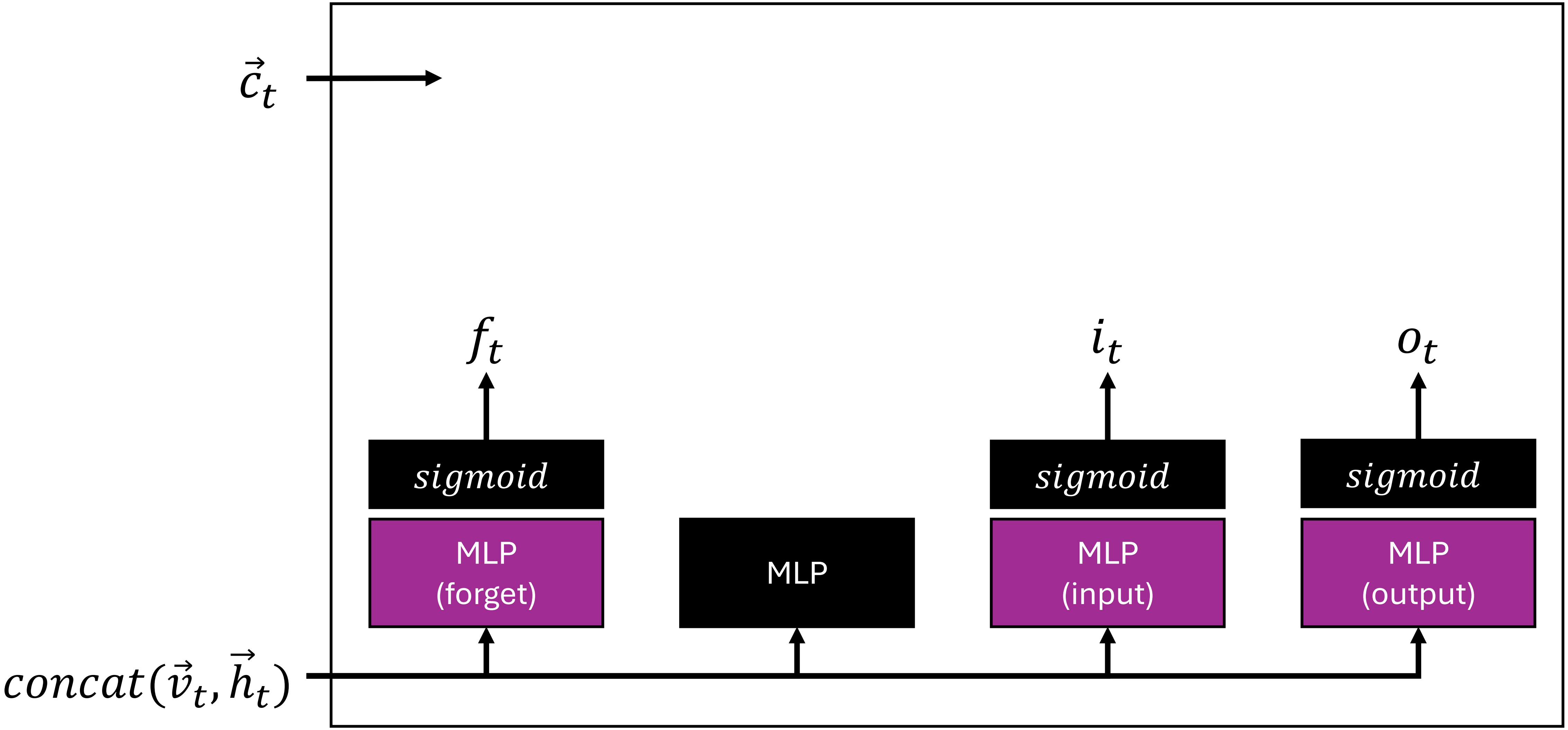


$\vec{h}_t$
$\vec{c}_t$
$\vec{v}_t$
4 MLP
$\vec{h}_{t+1}$
Activation function
$\vec{v}_{t+1}$
$\vec{h}_{t+1}$
$\vec{c}_{t+1}$
$\vec{v}_{t+1}$
4 MLP
$\vec{h}_{t+2}$
Activation function
$\vec{v}_{t+2}$
$\vec{h}_{t+2}$
$\vec{c}_{t+2}$
$\vec{v}_{t+2}$
4 MLP
$\vec{h}_{t+3}$
Activation function
$\vec{v}_{t+3}$
...

LSTM

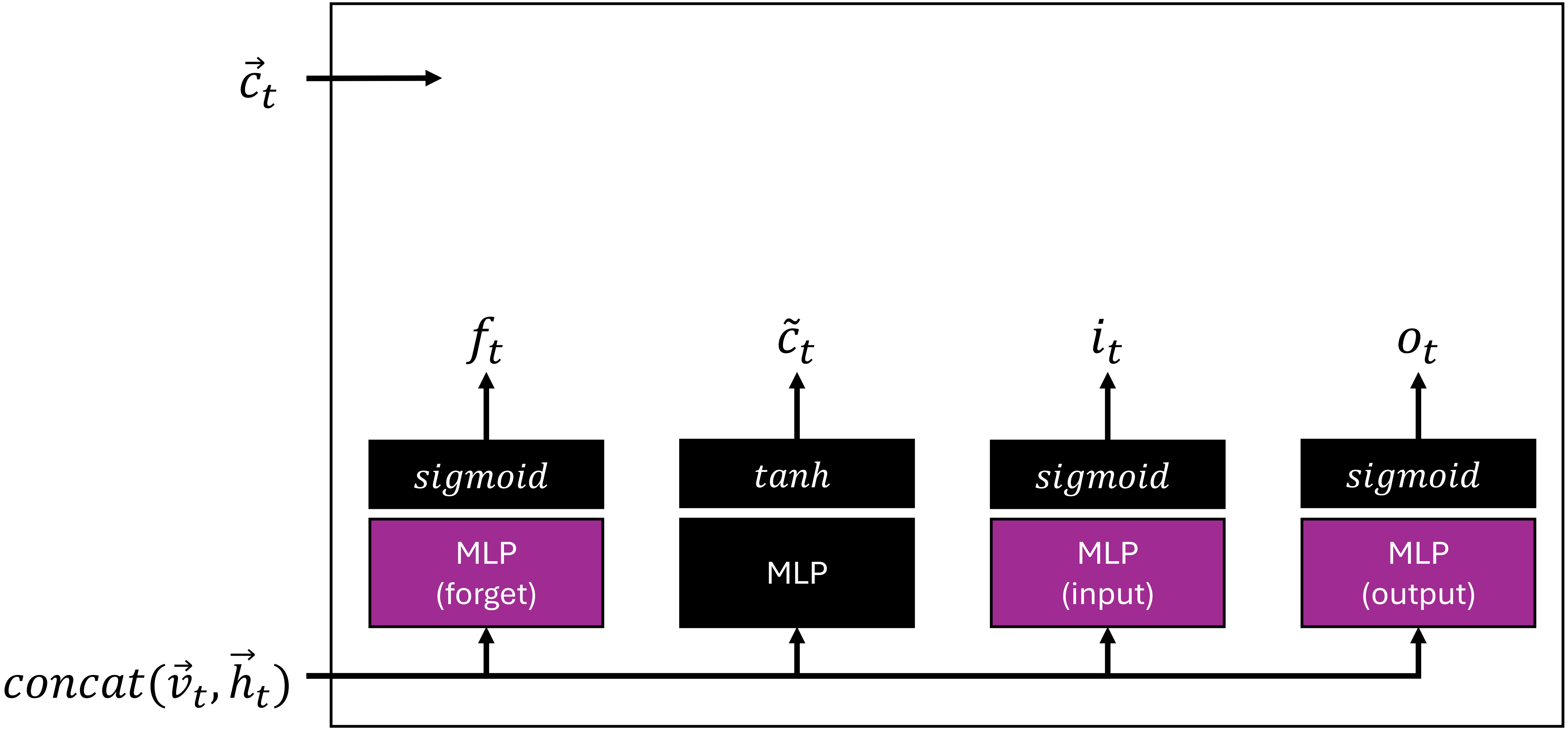

$\vec{c}_t$
Control networks
MLP (forget)
MLP
MLP (input)
MLP (output)
$concat(\vec{v}_t, \vec{h}_t)$

LSTM



$\vec{c}_t$
f_t
i_t
o_t
sigmoid
sigmoid
sigmoid
MLP
(forget)
MLP
MLP
(input)
MLP
(output)
$concat(\vec{v}_t, \vec{h}_t)$

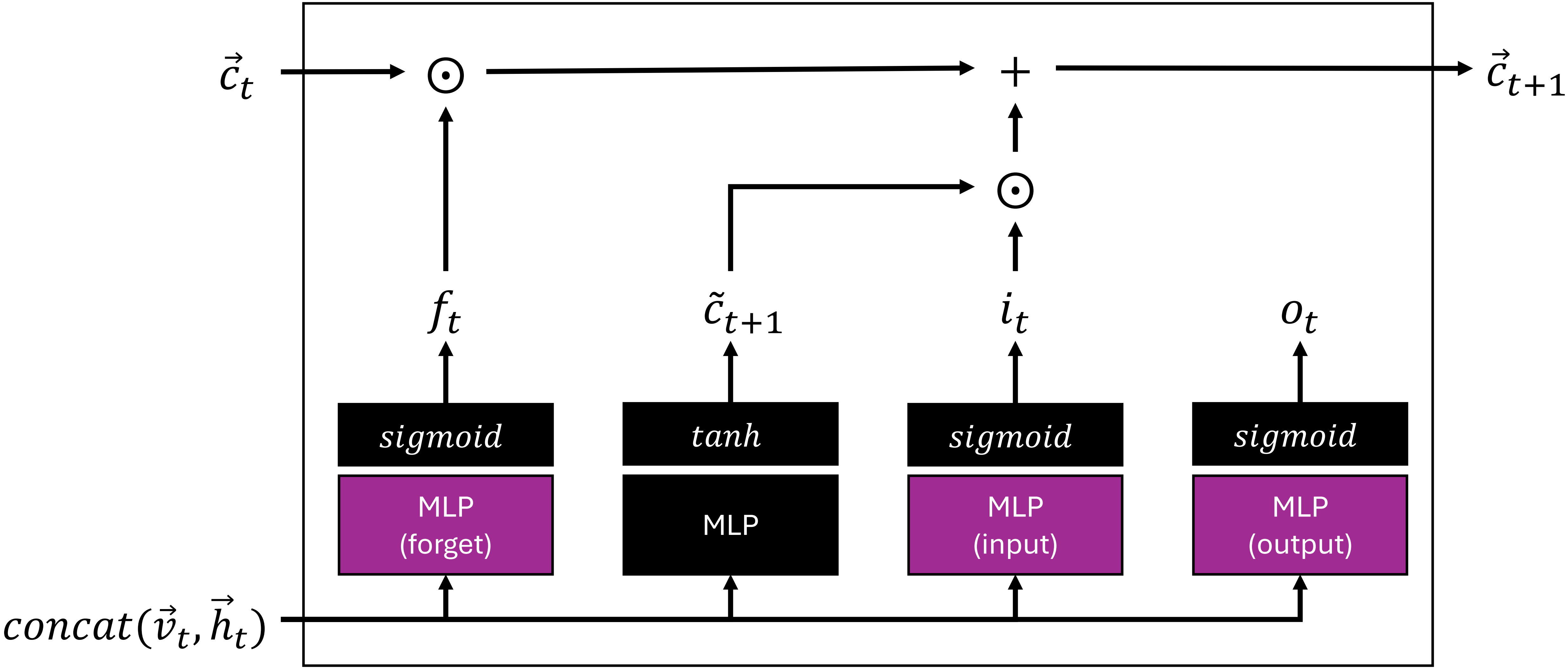
LSTM



$\vec{c}_t$
f_t
$\tilde{c}_t$
i_t
o_t
$sigmoid$
$tanh$
$sigmoid$
$sigmoid$
MLP (forget)
MLP
MLP (input)
MLP (output)
$concat(\vec{v}_t, \vec{h}_t)$

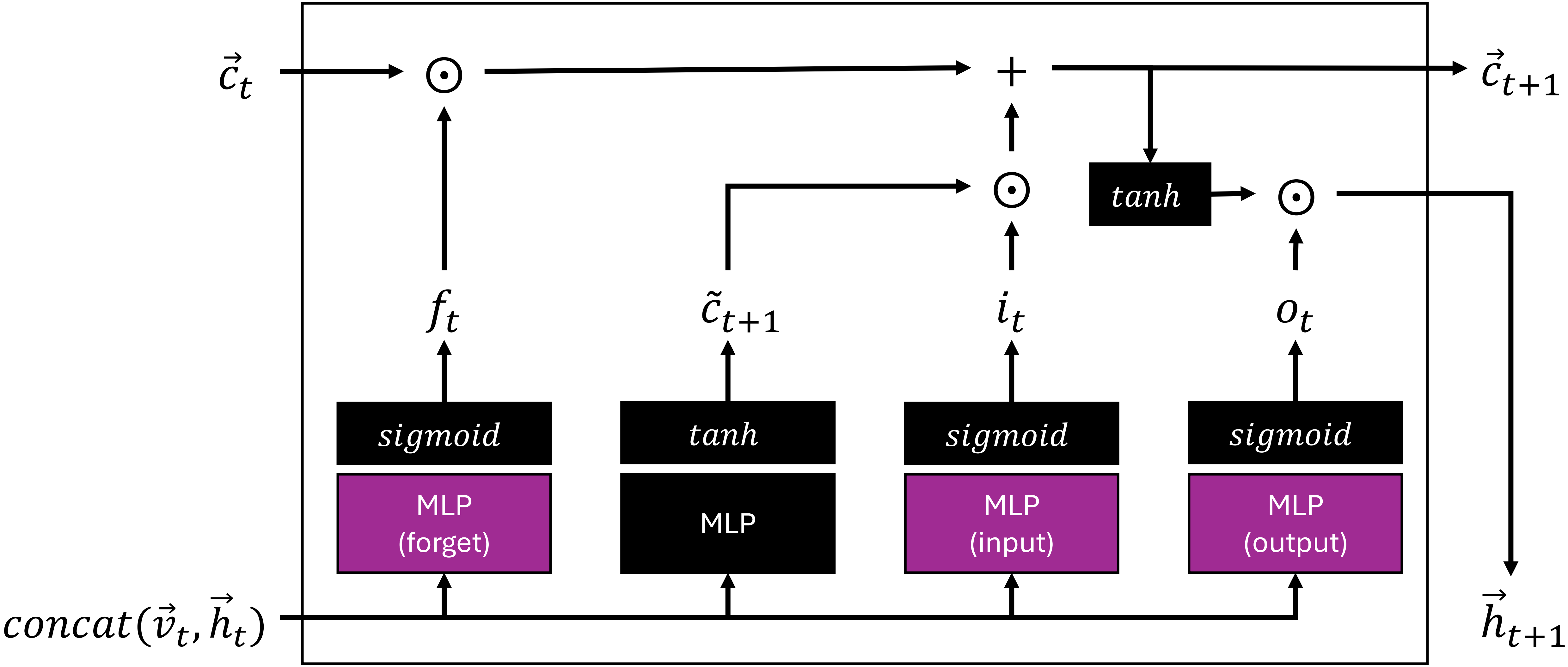
LSTM

$$\vec{c}_{t+1} = f_t \odot \vec{c}_t + i_t \odot \tilde{c}_{t+1}$$



LSTM

$$\vec{h}_{t+1} = o_t \odot tanh(\vec{c}_{t+1})$$



LSTM

0 $\vec{v}_t, \vec{c}_t, \vec{h}_t$

1 $\vec{c}_{t+1} = f_t \odot \vec{c}_t + i_t \odot \tilde{c}_{t+1}$

2 $\vec{h}_{t+1} = o_t \odot tanh(\vec{c}_{t+1})$

3 $\vec{v}_{t+1} = activation(\vec{h}_{t+1})$

Q&A

Lab Time

Lab 5: Forecasting a multivariate time series

Lab 5: Dataset

Electricity Transformer Dataset (ETDataset)

In this Github repo, we provide several datasets could be used for the long sequence time-series problem. All datasets have been preprocessed and they were stored as `.csv` files. The dataset ranges from 2016/07 to 2018/07.

	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
0	2016-07-01 00:00:00	5.827	2.009	1.599	0.462	4.203	1.340	30.531000
1	2016-07-01 00:15:00	5.760	2.076	1.492	0.426	4.264	1.401	30.459999
2	2016-07-01 00:30:00	5.760	1.942	1.492	0.391	4.234	1.310	30.038000
3	2016-07-01 00:45:00	5.760	1.942	1.492	0.426	4.234	1.310	27.013000
4	2016-07-01 01:00:00	5.693	2.076	1.492	0.426	4.142	1.371	27.787001

Figure 3. A demo of the ETT data.

Field	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
Description	The recorded **date**	High UseFul Load	High UseLess Load	Middle UseFul Load	Middle UseLess Load	Low UseFul Load	Low UseLess Load	**Oi**l Temperature (target)

Step 1

- Download the Lab5 directory from the GitHub repository
  https://github.com/ibrahimsaggaf/Introduction-to-Artificial-Intelligence
- Open the Lab5 directory in Visual Studio Code.
- The Lab5 directory contains 5 files:
  - main.py
  - model.py
  - network.py
  - utils.py
  - requirements.txt

Take your time examining these files.

Step 2

- Download the Electricity Transformer dataset (ETTm1.csv) from https://github.com/zhouhaoyi/ETDataset/tree/main
- Move the csv file into the lab5 directory:
    - ETTm1.csv($\approx 10\ MB$)

Step 3

- Create and activate a virtual environment under the name "lab5_env" (see Lab 1)
- Install the below libraries inside the virtual environment using a requirements file (see Lab 4):
  - Scikit-learn
  - Deep learning library Pytorch
  - Visualisation library Matplotlib

Step 4

- Run the command:
  python main.py

This command runs a forecasting task on a multivariate time series:

1. Load and scale the ET dataset.
2. Split the data considering time, where the training set includes past and present, whilst the testing set represents the future.
3. Create a LSTM model.
4. Train the model on the training set, then predict both training and testing sets.
5. Plot both real and predicted target series (OT) and save the figure.

Step 5

- Inspect the printed output along with the generated figure labelled "forecasting.jpg".
- Investigate two hyperparameters:
  - ❑ Try different values of *number_of_layers* (2, 3, 4)
  - ❑ Try different values of *number_of_epochs* (100, 500, 1000)

  Inspect the generated figure with every change and select the combination that obtained the best performance.

Lab 5: Forecasting a multivariate time series

Congrats!

[✓] Split time series data

[✓] Train a LSTM model using multivariate time series data

[✓] Conduct simple hyperparameter search

Quiz 4

Q1: Which of the following sampling methods works best for univariate time series data?

A) Stratified sampling
B) K-fold cross validation
C) Random sampling
D) None of the above

Q2: In this lab, what conclusions can be drawn from observing the generated figure?

A) Time series data is difficult to learn because it requires learning time-dependent patterns.

B) Tabular data is more difficult to learn compared to time series data

C) The figure suggests that the model successfully learnt time series patterns

D) All of the above

Reading list

Time Series Forecasting Made Simple (Part 4.1): Understanding Stationarity in a Time Series
https://towardsdatascience.com/time-series-forecasting-made-simple-part-4-1-understanding-stationarity-in-a-time-series/

Seasonality Detection in Time Series Data
https://www.geeksforgeeks.org/machine-learning/seasonality-detection-in-time-series-data/