Mix-Frequency Recurrent Neural Network

-Speaker: Stephen Chen-

(Background) Consider a situation like this

1. *We have multiple data sets.*
2. *We have multiple data sets. A bunch of points are missing.*
3. *Not only are they not available, but it also happens randomly.*

Mysterious data cleansing process.

We start to ask ourselves the ultimate question:

- *What if the data we received is just wrong from the beginning?*
- *In this case, then what are we doing?*

What do we want to achieve?

1. *Using multiple sources of data together to predict something moving towards certain direction.*
2. *Not a missing data problem. No fillings, no aggregation.*
3. *As simple as possible + flexible.*

What type of problems we encounter?

1.1 Data comes in with different qualities

- Different frequencies *- Different measurements (units)*

- Amount of noises (accuracies and the verification of sources)

1.2 What does the data really mean? It's not just about numbers.

- Macro and micro structures *- History*

2.1 Aggregation creates loss of information. Filling introduces noise.

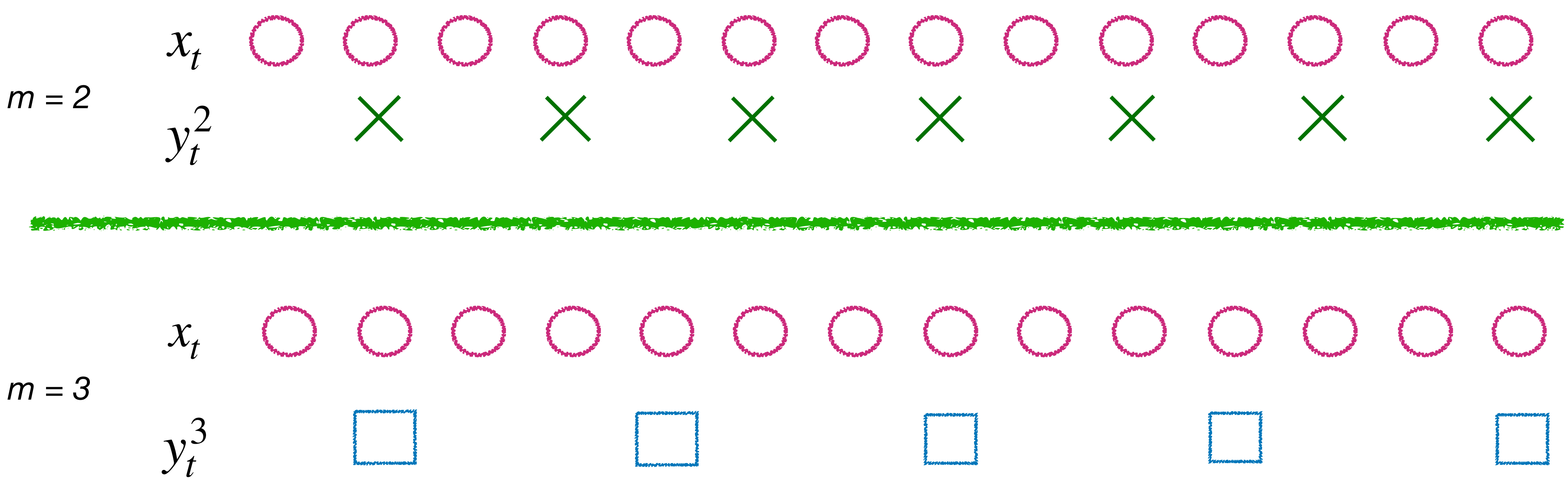
3.1 What type of data should we chose from?

Introduction to one and only notation

Frequency mismatch: m

Simple example when m= 2, 3.
Two types of data: high frequency and low frequency data

High frequency data	*Low frequency data*
: highest frequency data x_t	*: low frequency data* $y_t^{(m^i)}$

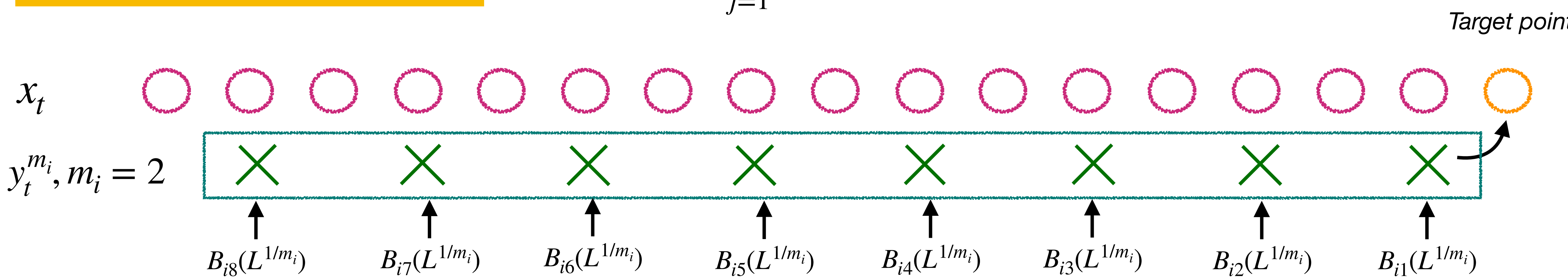


Related work

Mixed Data Sampling (MIDAS) regression models

The MIDAS Touch:
Mixed Data Sampling Regression Models
(2002), Ghysels, Santa-Clara, Valkanov

$$x_{t+1} = \beta_0 + \sum_{j=1}^{L} B_{ij}(L^{1/m_i})y_t^{m_i} + \epsilon_{t+1}$$



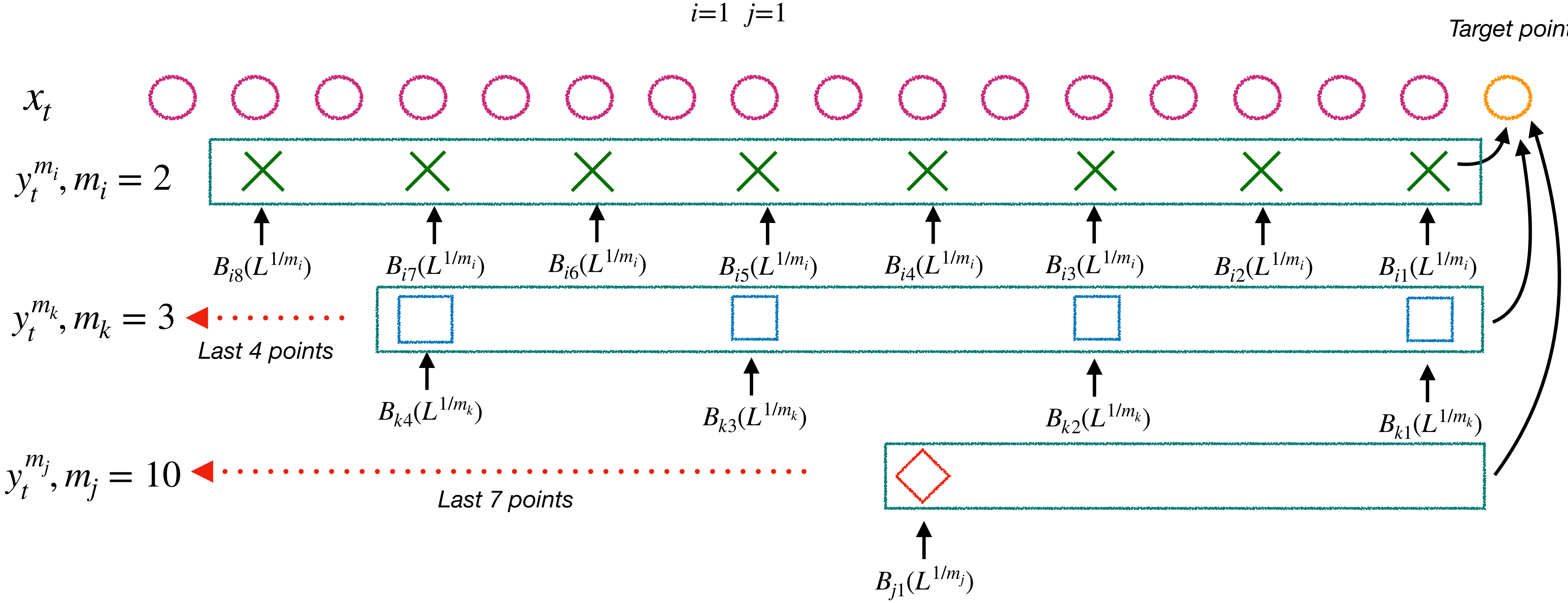
Frequency mismatch m=2, by using the past L (Length) number of y's to predict the next target point.

* *In this case, L=8,* $B_{ij}(L^{1/m_i})$ *Is the coefficient for* $y_t^{m_i}$.

* *The original MIDAS model is used to predict the low frequency data, the model can be modified to predict the high frequency components, which is called Reverse-MIDAS.*

MIDAS with multiple low frequency inputs

$$x_{t+1} = \beta_0 + \sum_{i=1}^{K} \sum_{j=1}^{L} B_{ij}(L^{1/m_i}) y_t^{m_i} + \epsilon_{t+1}$$



* *In this case, K=3, L=8,* $B_{ij}(L^{1/m_i})$ *Is the coefficient for* $y_t^{m_i}$.

Variations of the MIDAS model

Original linear model:

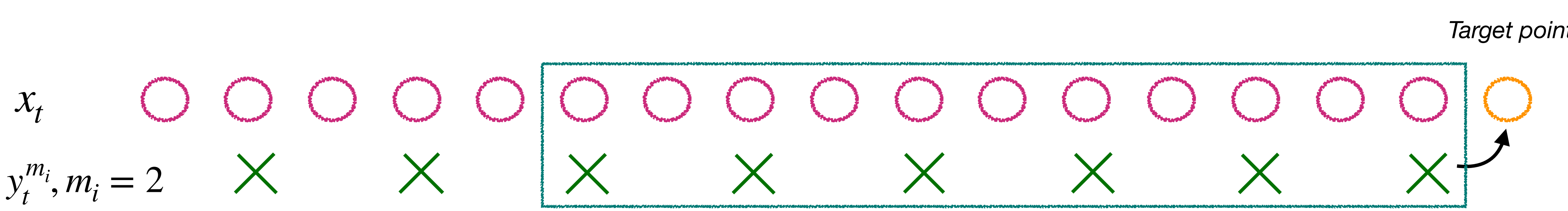
$$x_{t+1} = \beta_0 + \sum_{t=1}^{K} \sum_{j=1}^{L} B_{ij}(L^{1/m_i}) y_t^{m_i} + \epsilon_{t+1}$$

Autoregressive linear model:

$$x_{t+1} = x_t + \beta_0 + \sum_{t=1}^{K} \sum_{j=1}^{L} B_{ij}(L^{1/m_i}) y_t^{m_i} + \epsilon_{t+1}$$

General non-linear model:

$$x_{t+1} = \beta_0 + g\left(\sum_{t=1}^{K} \sum_{j=1}^{L} B_{ij}(L^{1/m_i}) y_t^{m_i} \right) + \epsilon_{t+1}$$

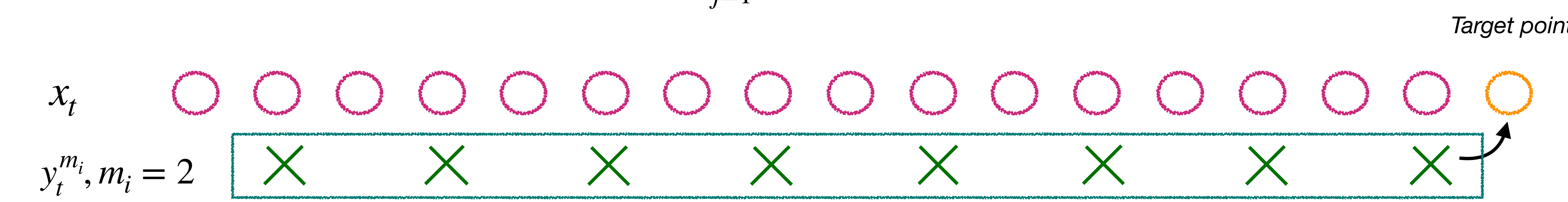


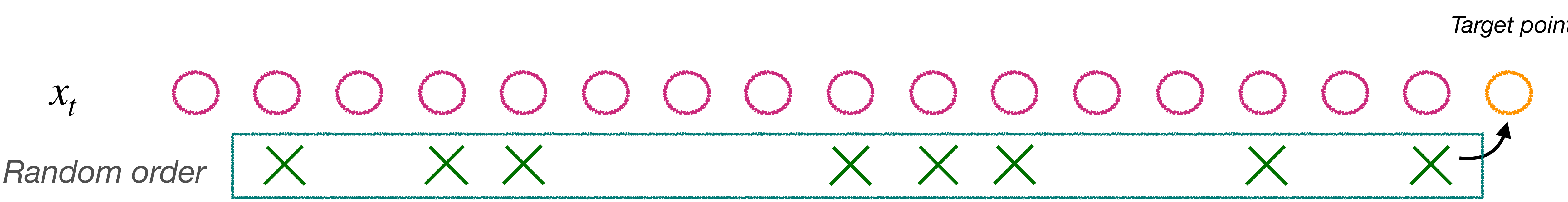
Draw a box, do the math. That's it.

Updates and potential problems with MIDAS model

Original linear model:

$$x_{t+1} = \beta_0 + \sum_{j=1}^{L} B_{1j}(L^{1/m_i})y_t^{m_i} + \epsilon_{t+1}$$





The MIDAS model focuses on the number (length) of y's, it doesn't care about where are the y's.

We will still have the same formulation, the MIDAS model doesn't necessarily has to be under the mixed frequency data context.

Feed-forward Neural Networks

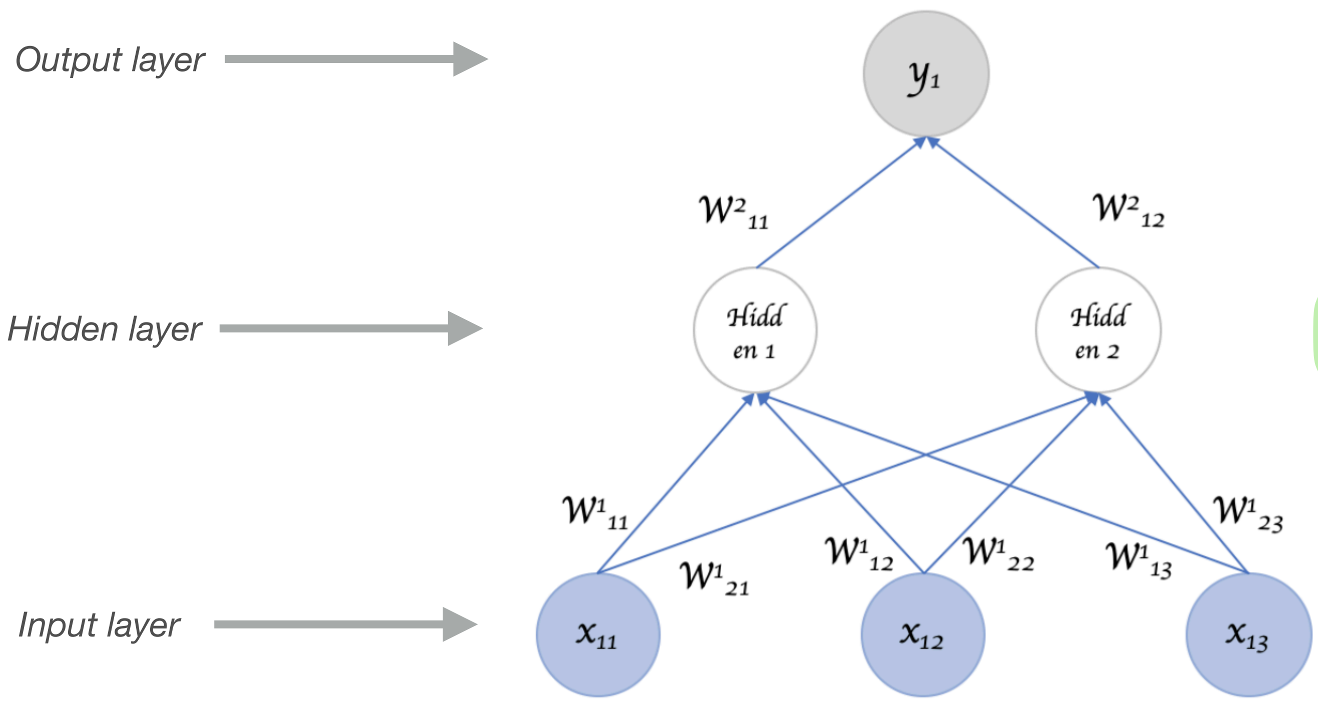
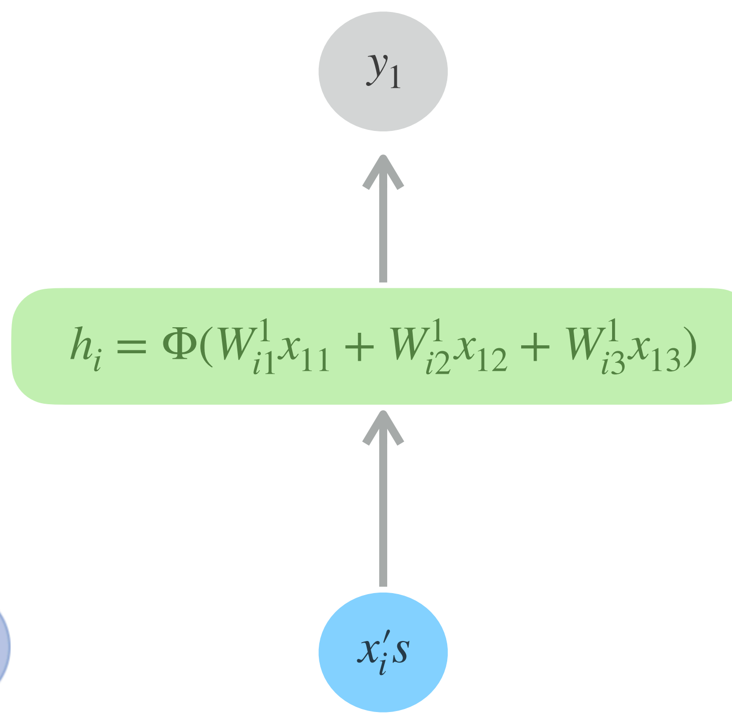


Figure 1: A vanilla network representation, with an input of size 3 and one hidden layer and one output layer of size 1.



Why do we need the activation functions: introducing non-linear properties to realize complex mappings.

** Figure 1, source: https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce*

How do we get Recurrent Neural Networks

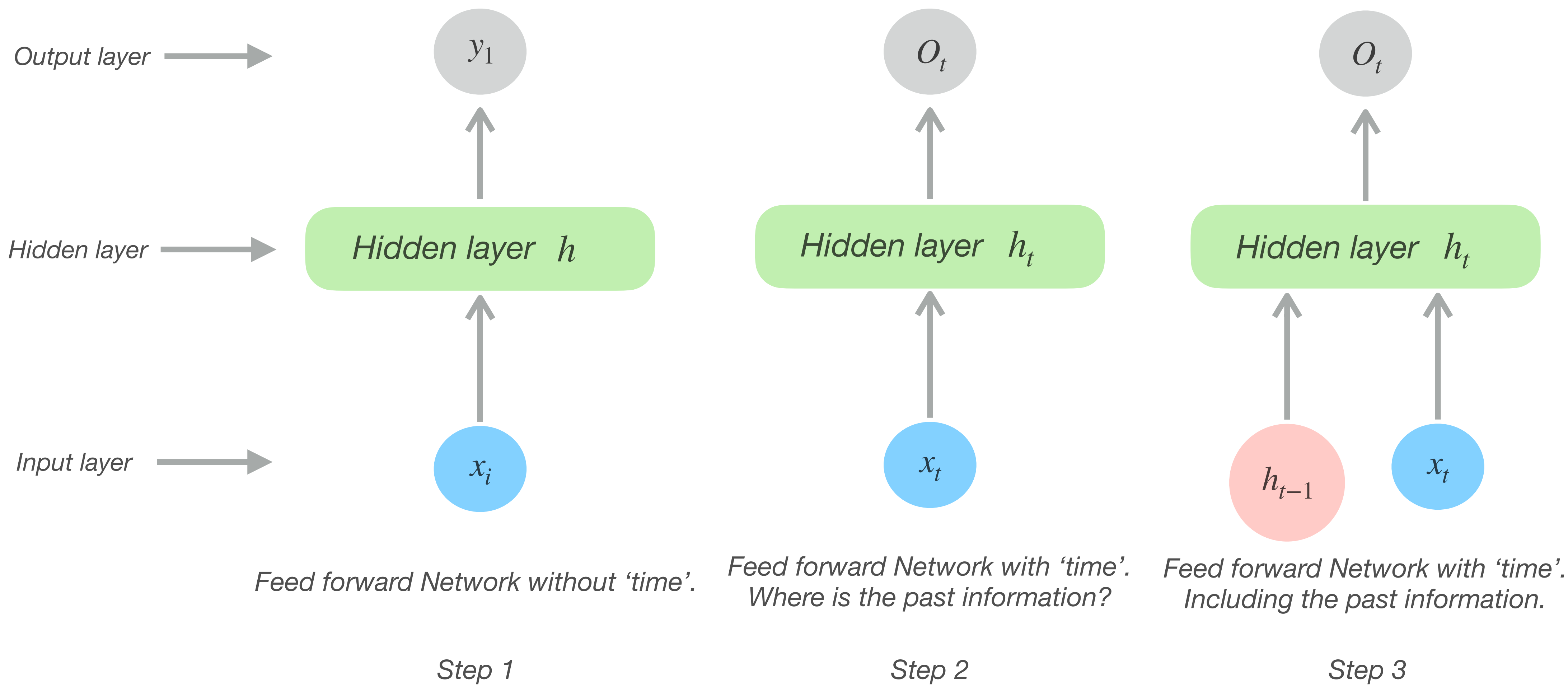


Figure 2: Evolution chain: from feed forward network to recurrent neural network

Recurrent Neural Networks formulation

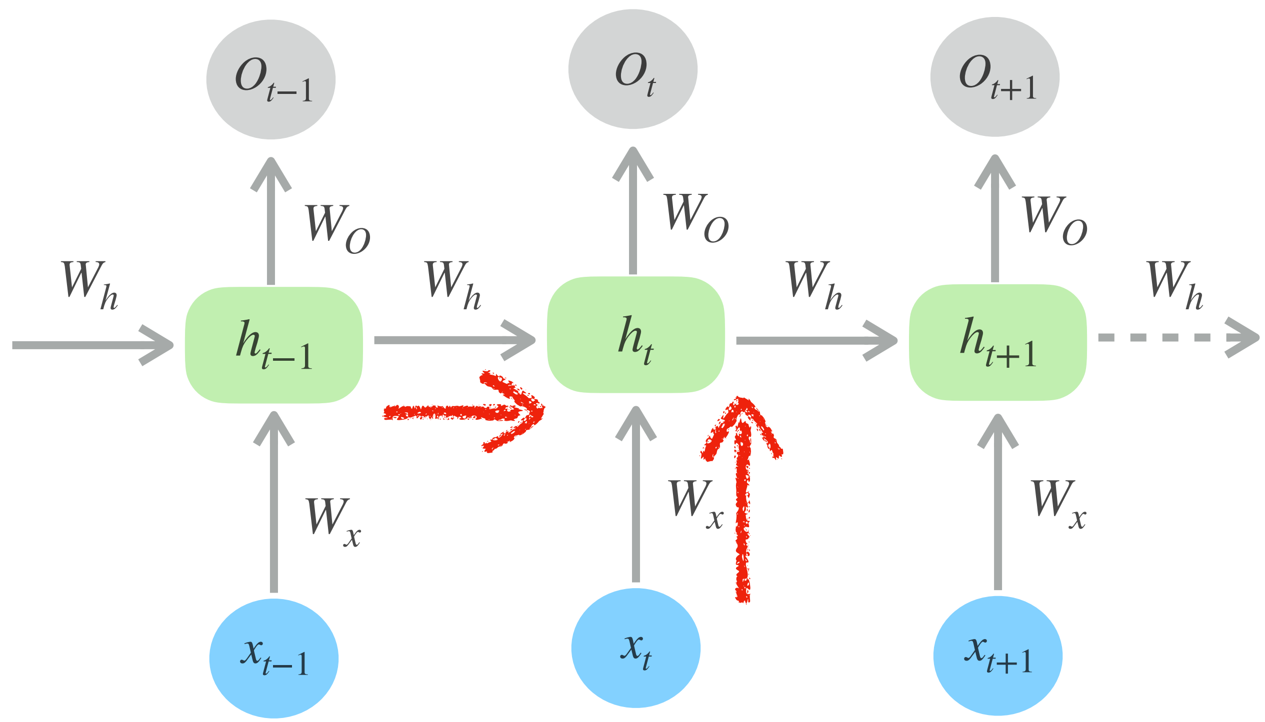


Figure 3: Recurrent Network structure

The hidden state has two sources of inputs:

$$h_t = \Phi(W_h h_{t-1} + W_x x_t + b_h)$$

$$O_t = W_O h_t + b_O$$

Meanwhile the output can be anything. Relevant or irrelevant to the context, for example:

1. *The prediction of* $\hat{x}_{t+1}$ *, or even* $\hat{x}_{t+2}$.
2. *The weather condition for next week.*
3. *Price fluctuations on apples.*

Recurrent Neural Networks formulation

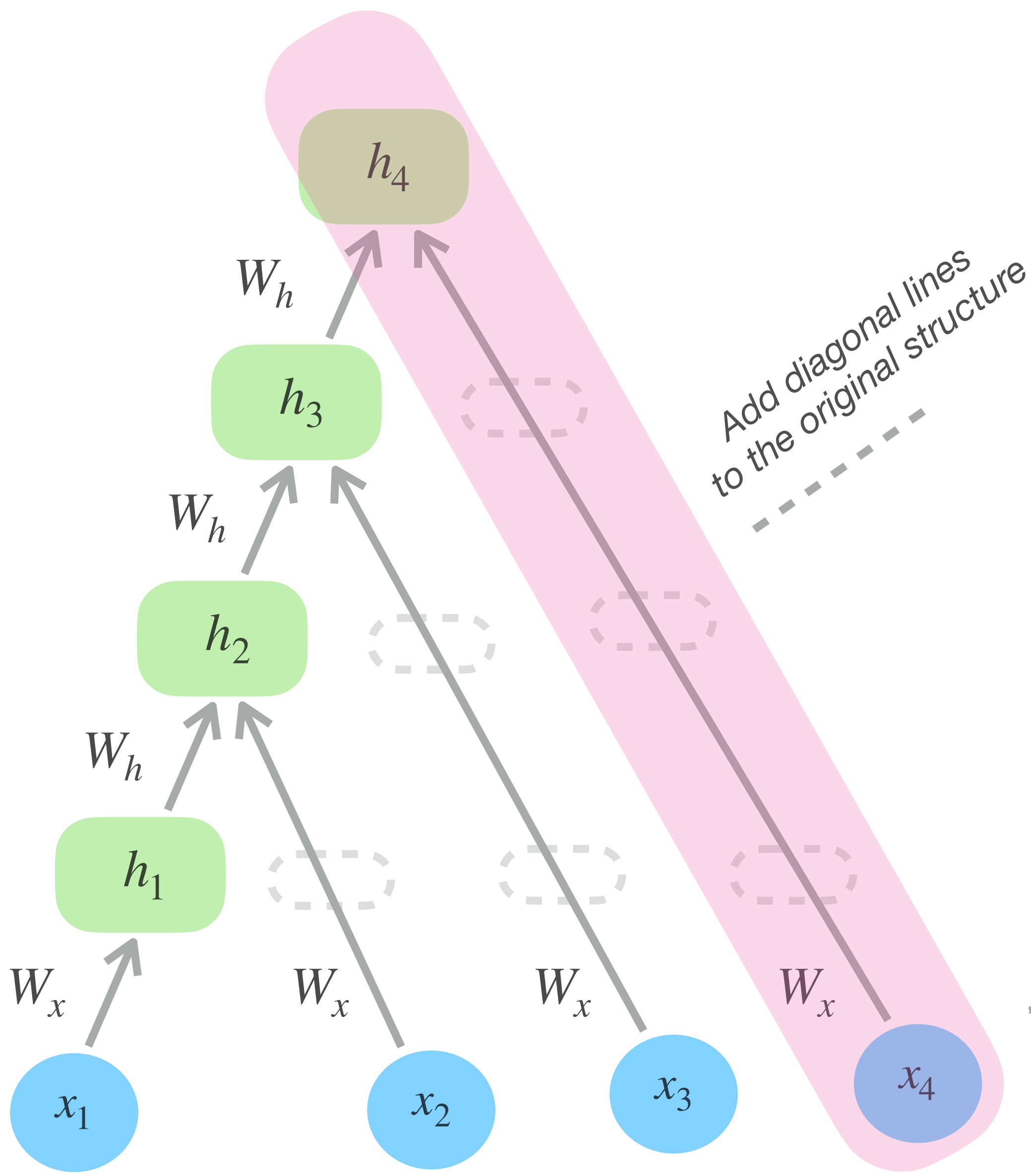


Figure 4: Recurrent Network structure from another point of view.

Do we need to share the parameters across the inputs over time?

Possible leverages:

- *Reducing model's complexity. (computation & overfitting)*

- *Parameter sharing reflects the fact that the model is performing the same task at each step.*

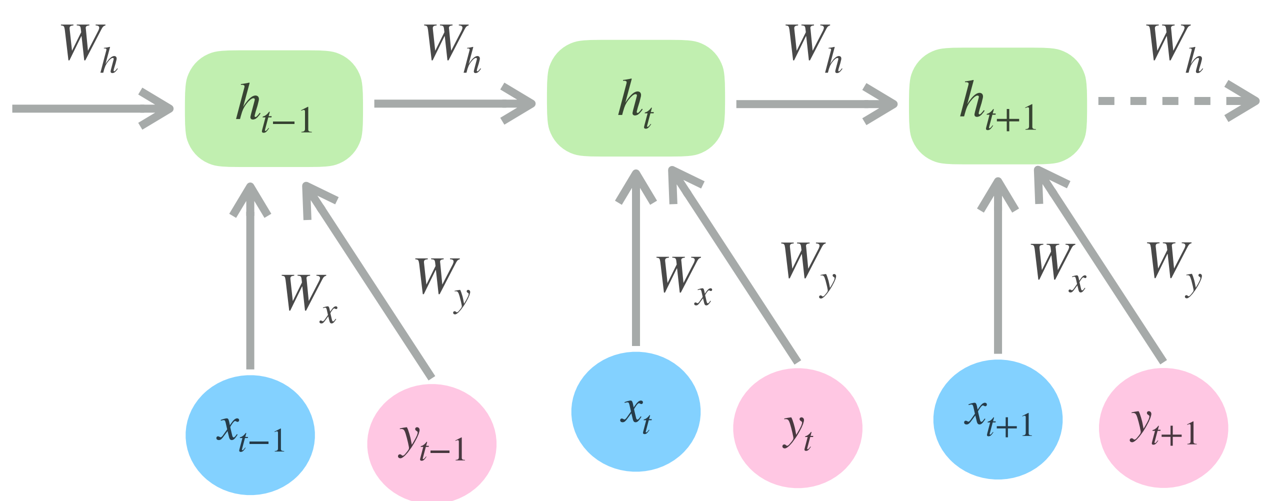
Price paid

- *losing versatility and depth of the model.*

- *Introducing constraints on inputs, it has to be fixed length over time. (* this one is huge *)*

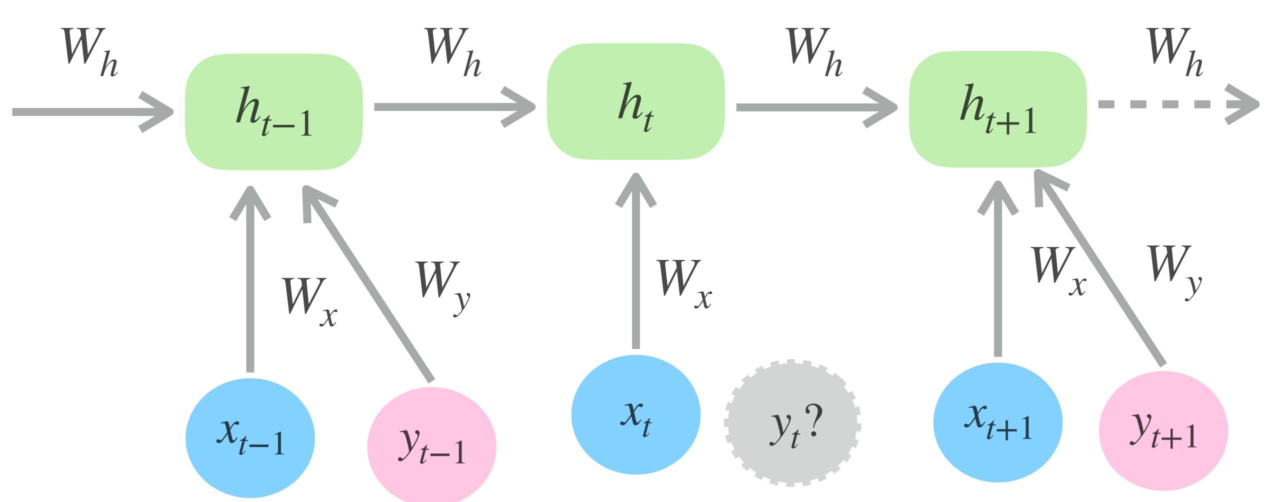
- *** If time length is fixed, the RNN has arbitrary weight, it will have the same structure as a regular feed forward network.*

Recurrent Neural Networks with two inputs (m=2)



$$h_t = \Phi(W_h h_{t-1} + W_x x_t + W_y y_t)$$

Figure 5: Recurrent Network structure with two types of inputs



Can we formulate it in this way?

$$h_t = \Phi(W_h h_{t-1} + W_x x_t)$$

$$h_{t+1} = \Phi(W_h h_t + W_x x_t + W_y y_{t+1})$$

The weights are shared across time, y has missing values.

Figure 6: Recurrent Network structure with two types of inputs
But y occurs every two steps.

Recurrent Neural Networks with two inputs (m=2)

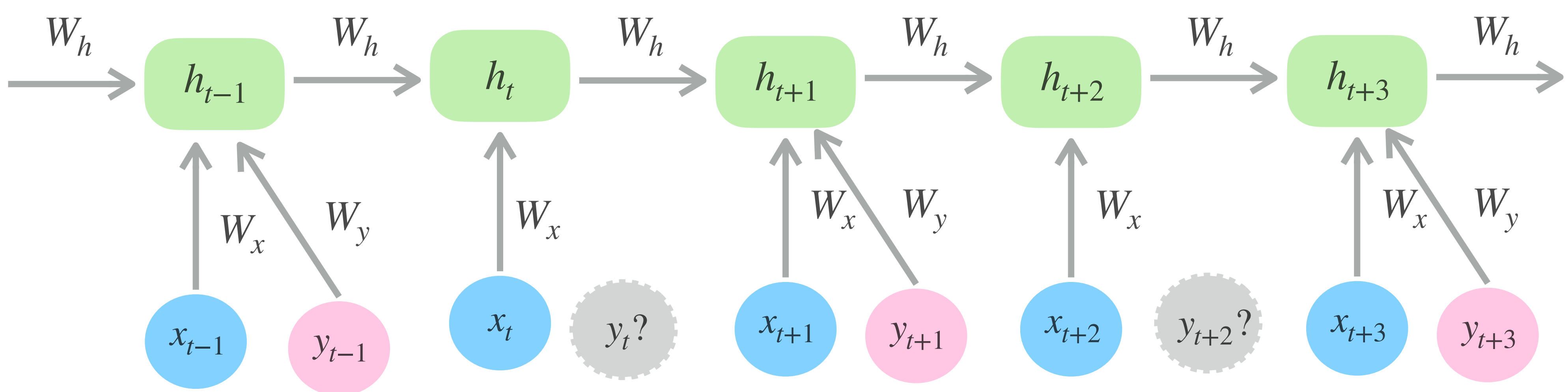


Figure 7: Recurrent Network structure with two types of inputs, y occurs every two steps.

$$h_t = \Phi(W_h h_{t-1} + W_x x_t + W_y y_t^{predict})$$

$$h_{t+1} = \Phi(W_h h_t + W_x x_{t+1} + W_y y_{t+1})$$

- *Two types of filling/prediction that are different: Constant and Zero*
- *Any interpolation methods will not be suitable, must use extrapolate methods. (can not use future information)*
  - *Under this formulation structure, the performance of the model depends on the accuracy of the prediction.*

Recurrent Neural Networks with two inputs (m=2)

Old model with filling/prediction

$$h_t = \Phi(W_h h_{t-1} + W_x x_t + W_y y_t^{predict})$$

$$h_{t+1} = \Phi(W_h h_t + W_x x_{t+1} + W_y y_{t+1})$$

How do we tackle the mixed frequency context?

Why is the data in mixed frequency?

Can the mixed frequency property be utilized in the model?

New model with filling/prediction

$$h_t = \Phi(W_h^1 h_{t-1} + W_x^1 x_t + W_y^1 y_t^{predict})$$

$$h_{t+1} = \Phi(W_h^2 h_t + W_x^2 x_{t+1} + W_y^2 y_{t+1})$$

Two sets of weights to capture different signals.

Old model with **ZERO** filling/prediction

$$h_t = \Phi(W_h h_{t-1} + W_x x_t)$$

$$h_{t+1} = \Phi(W_h h_t + W_x x_{t+1} + W_y y_{t+1})$$

One set of weight sharing across both scenarios

New model with **ZERO** filling/prediction

$$h_t = \Phi(W_h^1 h_{t-1} + W_x^1 x_t)$$

$$h_{t+1} = \Phi(W_h^2 h_t + W_x^2 x_{t+1} + W_y^2 y_{t+1})$$

Two sets of weights to compensate with the missing data

MF-RNN variations with two inputs (m=2)

Original Mixed Frequency RNN:
m=2

$$h_t = \Phi(W_h^1 h_{t-1} + W_x^1 x_t + W_y^1 y_t^{predict})$$

$$h_{t+1} = \Phi(W_h^2 h_t + W_x^2 x_{t+1} + W_y^2 y_{t+1})$$

Autoregressive MF-RNN
m=2

$$h_t = \Phi(W_h^1 h_{t-1} + W_x^{1,1} x_t + W_x^{1,2} x_{t-1} + W_y^1 y_t^{predict})$$

$$h_{t+1} = \Phi(W_h^2 h_t + W_x^{2,1} x_{t+1} + W_x^{2,2} x_t + W_y^2 y_{t+1})$$

Non-linear MF-RNN
m=2

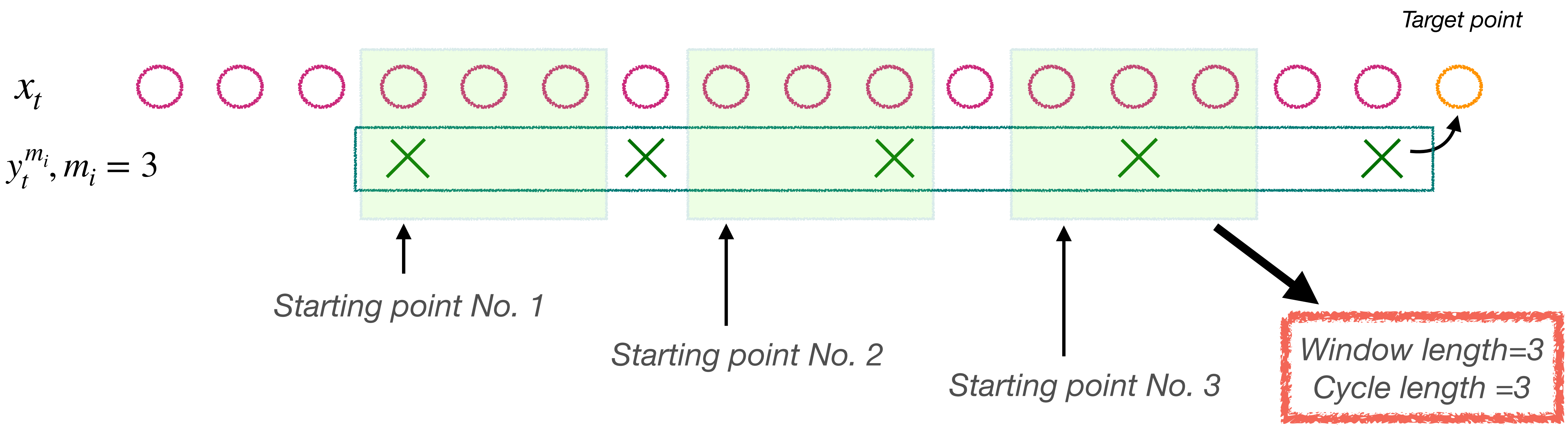
$$h_t = \Phi(W_h^1 h_{t-1} + g(W_x^{1,1} x_t + W_x^{1,2} x_{t-1} + W_y^1 y_t^{predict}))$$

$$h_{t+1} = \Phi(W_h^2 h_t + g(W_x^{2,1} x_{t+1} + W_x^{2,2} x_t + W_y^2 y_{t+1}))$$

General MF-RNN
m=2

$$h_t = \Phi\left(K_h(\theta_0, t) h_{t-1} + g\left(K_x^1(\theta_1, t) x_t + K_x^2(\theta_2, t) x_{t-1} + K_y(\theta_3, t) \hat{y}_t\right)\right)$$

Recurrent Neural Networks with two inputs (m=3)



Mixed Frequency RNN (different starting point):

Starting at No. 2:

$$h_t = \Phi(W_h^1 h_{t-1} + W_x^1 x_t + W_y^1 y_t^{predict})$$

$$h_{t+1} = \Phi(W_h^2 h_t + W_x^2 x_{t+1} + W_y^2 y_{t+1}^{predict})$$

$$h_{t+2} = \Phi(W_h^3 h_{t+1} + W_x^3 x_{t+2} + W_y^3 y_{t+2})$$

Starting at No. 3:

$$h_t = \Phi(W_h^1 h_{t-1} + W_x^1 x_t + W_y^1 y_t^{predict})$$

$$h_{t+1} = \Phi(W_h^2 h_t + W_x^2 x_{t+1} + W_y^2 y_{t+1})$$

$$h_{t+2} = \Phi(W_h^3 h_{t+1} + W_x^3 x_{t+2} + W_y^3 y_{t+2}^{predict})$$

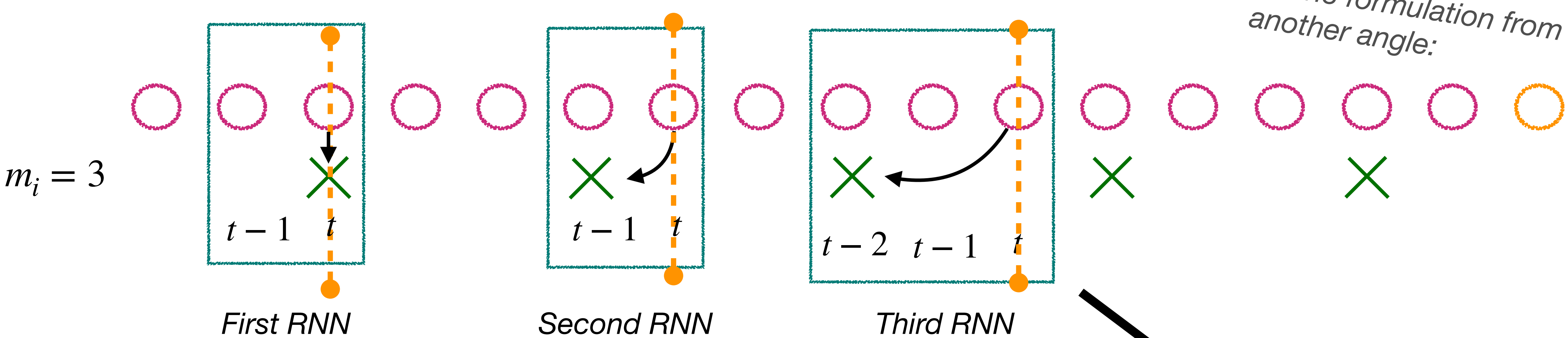
Recurrent Neural Networks with two inputs (m=3), with constant fillings

Constant fillings has two meanings:

- *Predict the present with the information from the past, without any modification.*
- *Carrying the past information to present, and only utilize the information from the past.*

An example with two inputs when m=3:

Think about the formulation from another angle:



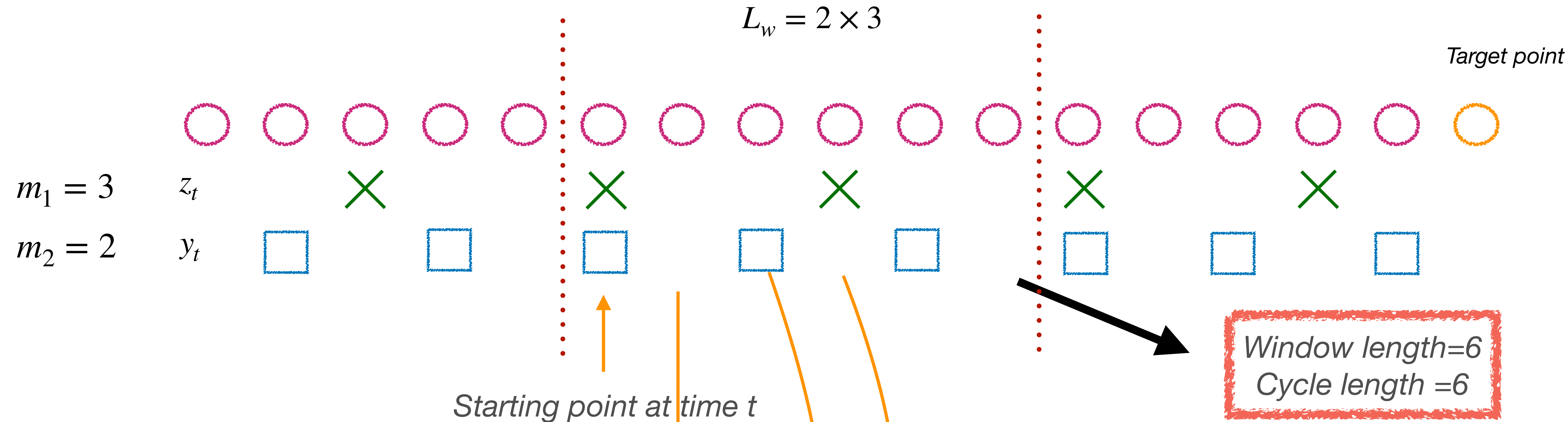
Window length=3
Cycle length =3

First RNN: $h_t = \Phi(h_{t-1}W_1^h + x_tW_1^x + \boxed{y_tW_1^y)}$

Second RNN: $h_t = \Phi(h_{t-1}W_2^h + x_tW_2^x + \boxed{y_{t-1}W_2^y)}$

Third RNN: $h_t = \Phi(h_{t-1}W_3^h + x_tW_3^x + \boxed{y_{t-2}W_3^y)}$

Recurrent Neural Networks with three inputs (m=2, 3)



$$h_t = \Phi(h_{t-1}W_1^h + x_tW_1^x + y_tW_1^y + z_tW_1^z)$$

$$h_{t+1} = \Phi(h_tW_2^h + x_{t+1}W_2^x + y_{t+1}^{predict}W_2^y + z_{t+1}^{predict}W_2^z)$$

$$h_{t+2} = \Phi(h_{t+1}W_3^h + x_{t+2}W_3^x + y_{t+2}W_3^y + z_{t+2}^{predict}W_3^z)$$

$$h_{t+3} = \Phi(h_{t+2}W_4^h + x_{t+3}W_4^x + y_{t+3}^{predict}W_4^y + z_{t+3}W_4^z)$$

Total number of six equations

$$h_t = \Phi(h_{t-1}W_1^h + x_tW_1^x + y_tW_1^y + z_tW_1^z)$$

$$h_{t+1} = \Phi(h_tW_2^h + x_{t+1}W_2^x + 0 + 0)$$

$$h_{t+2} = \Phi(h_{t+1}W_3^h + x_{t+2}W_3^x + y_{t+2}W_3^y + 0)$$

$$h_{t+3} = \Phi(h_{t+2}W_4^h + x_{t+3}W_4^x + 0 + z_{t+3}W_4^z)$$

Total number of six equations

Filling with ZEROS

Recurrent Neural Networks with three inputs

Corollary 3.0.1. *The number of the patterns $N_p \leq N_m$. Where $\#N_m$ is defined as the total least common multiple among all the unique values $m_k, k = 1, ...K$.*

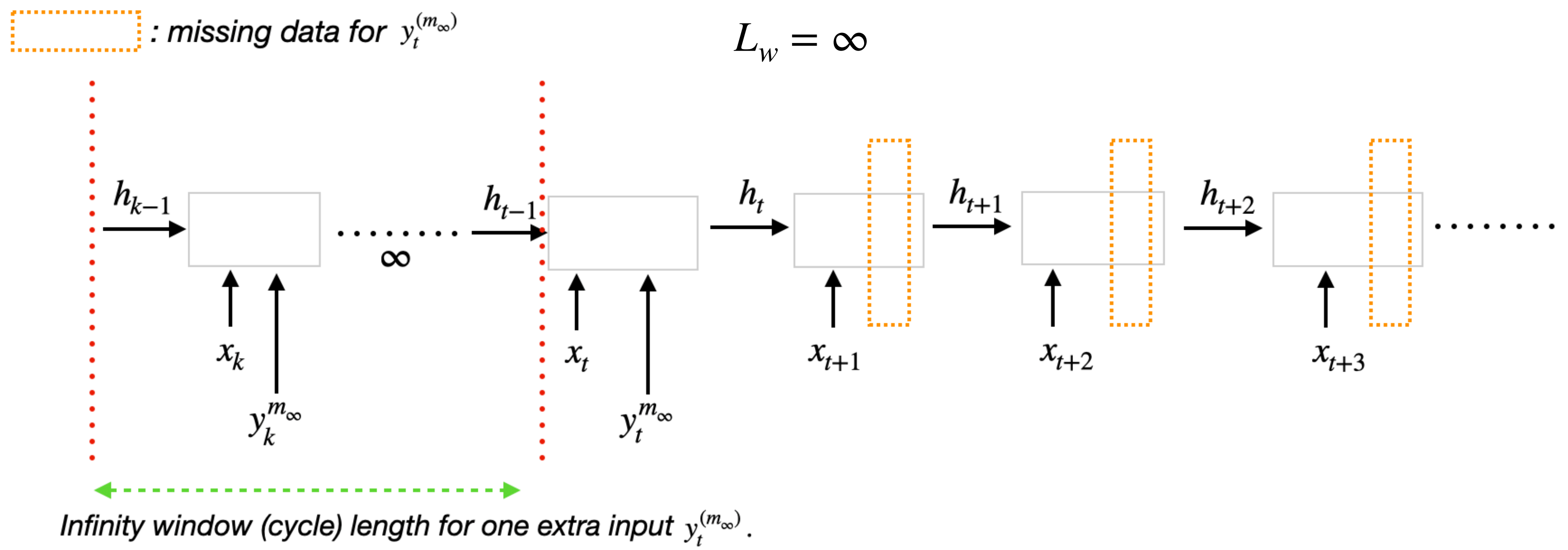
Proof. Consider a special case with two types of exogenous inputs $y_t^{(m_1)}$ and $y_t^{(m_2)}$, where $m_1 = 2, m_2 = 4$. Then $N_p < N_m$, see table [2] below,

Variable List & Combination of Patterns				
time	x_t	$y_t^{(m_1)}$	$y_t^{(m_2)}$	Pattern(s)
$4t$	○	○	○	Pattern 1
$4t+1$	○	○	×	Pattern 2
$4t+2$	○	○	○	Pattern 1 (repeats)
$4t+3$	○	×	×	Pattern 3

Table 2: Patterns with two exogenous variables $y_t^{(m_1)}, y_t^{(m_2)}, m_1 = 2, m_2 = 4$.

This is also a problem, under this formulation, if any exogenous input has a large value of frequency mismatch, the total amount of weights will increase dramatically. For example, m=100.

Recurrent Neural Network with infinity frequency mismatch (extreme)



Formulation 1:

$$h_t = \Phi(h_{t-1}W_1^h + x_tW_1^x + y_t^{m_\infty}W_1^y)$$
$$h_{t+1} = \Phi(h_tW_2^h + x_{t+1}W_2^x + y_{t+1}^{predict}W_2^y)$$
$$h_{t+2} = \Phi(h_{t+1}W_3^h + x_{t+2}W_3^x + y_{t+2}^{predict}W_3^y)$$
$$h_{t+3} = \Phi(h_{t+2}W_3^h + x_{t+3}W_4^x + y_{t+3}^{predict}W_4^y)$$

Infinity sets of weights

Infinity number of equations

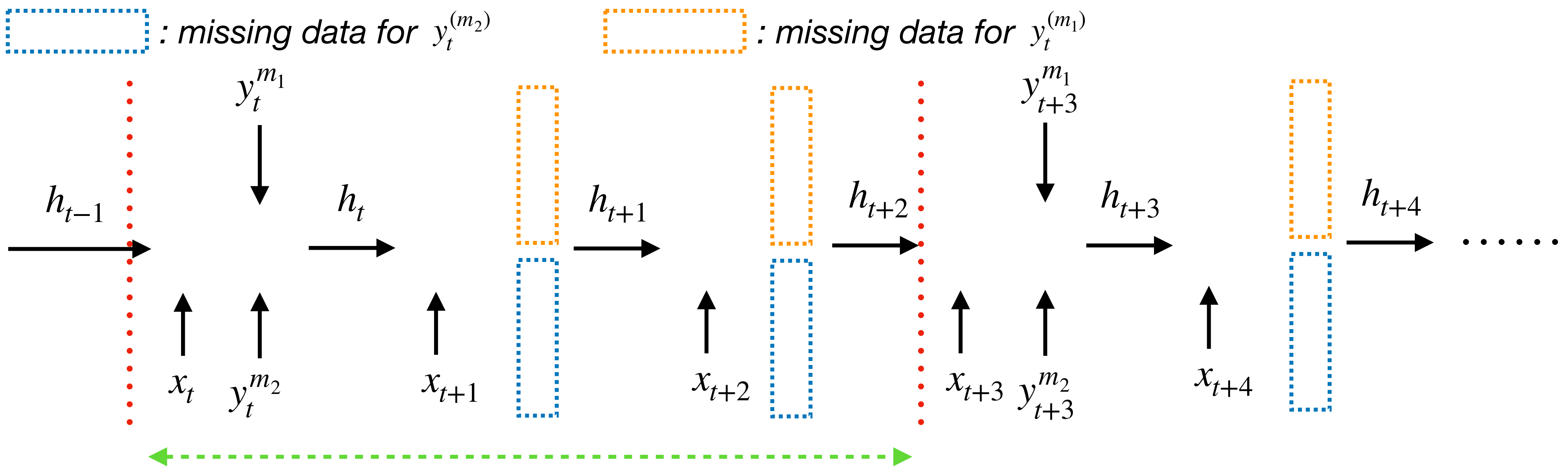
Formulation 2:

$$h_t = \Phi(h_{t-1}W_1^h + x_tW_1^x + y_t^{m_\infty}W_1^y)$$
$$h_{t+1} = \Phi(h_tW_2^h + x_{t+1}W_2^x + y_{t+1}^{predict}W_2^y)$$
$$h_{t+2} = \Phi(h_{t+1}W_2^h + x_{t+2}W_2^x + y_{t+2}^{predict}W_2^y)$$
$$h_{t+3} = \Phi(h_{t+2}W_2^h + x_{t+3}W_2^x + y_{t+3}^{predict}W_2^y)$$

Two sets of weights

Infinity number of equations

Recurrent Neural Network with m=3 frequency mismatch (another formulation)



Window (Cycle) length = 3 for two extra inputs with same frequency mismatch. $y_t^{(m_1)}, y_t^{(m_2)}$. $m_1 = m_2 = 3$.

Formulation 1:

$$h_t = \Phi(h_{t-1}W_1^h + x_t W_1^x + y_t^{m_1} W_1^{y_{m_1}} + y_t^{m_2} W_1^{y_{m_2}})$$

$$h_{t+1} = \Phi(h_t W_2^h + x_{t+1} W_2^x + \hat{y}_{t+1}^{m1} W_2^{y_{m1}} + \hat{y}_{t+1}^{m2} W_2^{y_{m2}})$$

$$h_{t+2} = \Phi(h_{t+1} W_3^h + x_{t+2} W_3^x + \hat{y}_{t+2}^{m1} W_3^{y_{m1}} \hat{y}_{t+2}^{m2} W_3^{y_{m2}})$$

Three sets of weights

Three equations

Formulation 2:

$$h_t = \Phi(h_{t-1}W_1^h + x_t W_1^x + y_t^{m_1} W_1^{y_{m_1}} + y_t^{m_2} W_1^{y_{m_2}})$$

$$h_{t+1} = \Phi(h_t W_2^h + x_{t+1} W_2^x + \hat{y}_{t+1}^{m1} W_2^{y_{m1}} + \hat{y}_{t+1}^{m2} W_2^{y_{m2}})$$

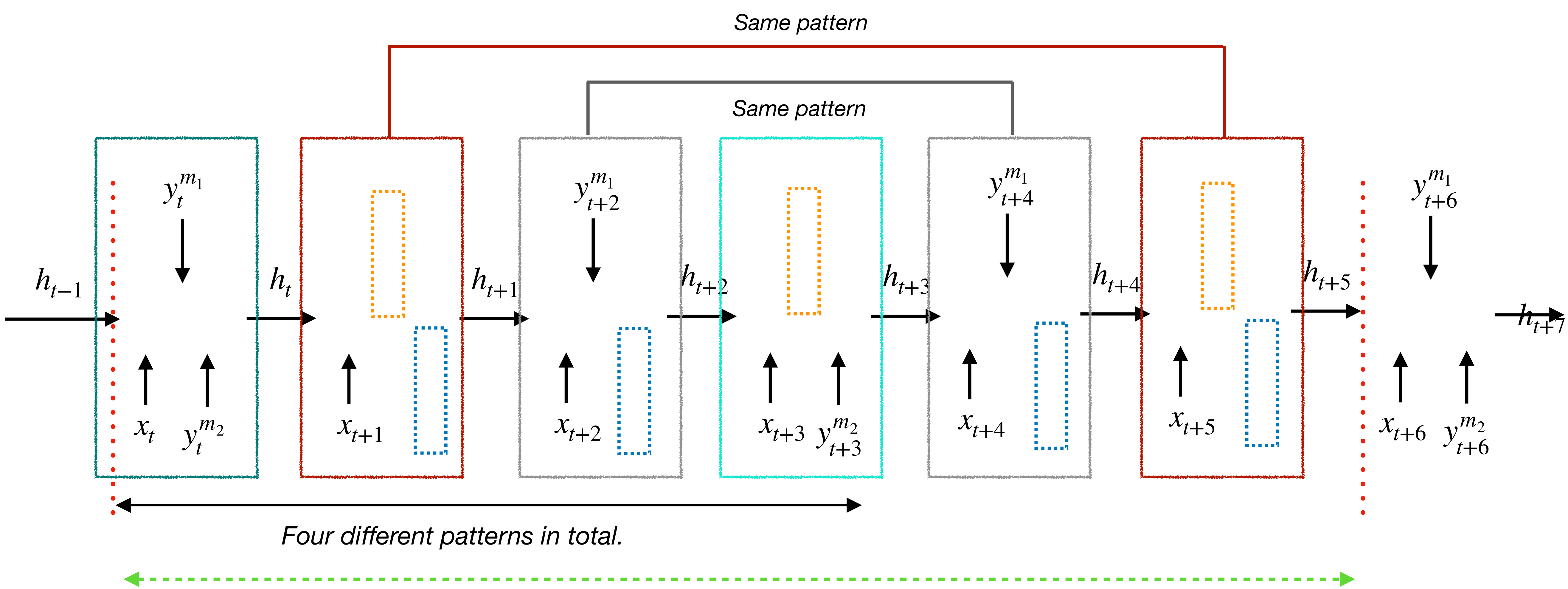
$$h_{t+2} = \Phi(h_{t+1} W_2^h + x_{t+2} W_2^x + \hat{y}_{t+2}^{m1} W_2^{y_{m1}} \hat{y}_{t+2}^{m2} W_2^{y_{m2}})$$

Two sets of weights

Three equations

Recurrent Neural Network with m=2,3 (Pattern formulation)

: missing data for $y_t^{(m_1)}$: missing data for $y_t^{(m_2)}$: RNN cell patterns



Window (Cycle) length = 6 for two extra inputs with frequency mismatch. $y_t^{(m_1)}, y_t^{(m_2)}$. $m_1 = 2, m_2 = 3$. $L_w = 2 \times 3$

Recurrent Neural Network with m=2,3 (Pattern formulation)

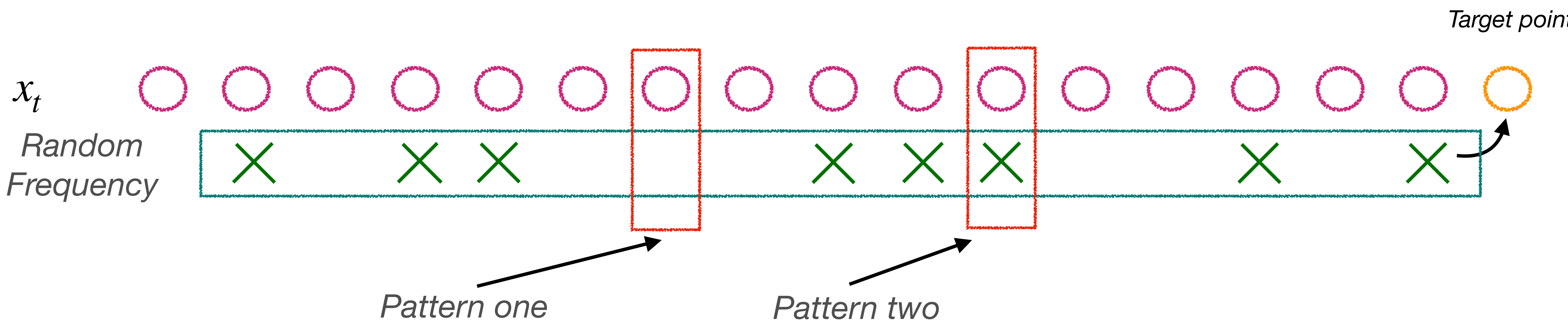
Variable List & Combination of Patterns				
time	x_t	$y_t^{(m_1)}$	$y_t^{(m_2)}$	*Pattern(s)*
$6t$	◯	◯	◯	Pattern 1
$6t+1$	◯	×	×	Pattern 2
$6t+2$	◯	◯	×	Pattern 3
$6t+3$	◯	×	◯	Pattern 4
$6t+4$	◯	◯	×	Pattern 3 (repeats)
$6t+5$	◯	×	×	Pattern 2 (repeats)

Table 1: Patterns with two exogenous variables $y_t^{(m_1)}, y_t^{(m_2)}, m_1 = 2, m_2 = 3$.

Based on the structure of each pattern, at time t, we can model them as,

$$h_t = \begin{cases} \Phi\left(W^h_{L,1} h_{t-1} + W^x_{I,1} x_t + W^{y^{(m_1)}}_{I,1} y_t^{(m_1)} + W^{y^{(m_2)}}_{I,1} y_t^{(m_2)}\right), & \text{Pattern 1} \\ \Phi\left(W^h_{L,2} h_{t-1} + W^x_{I,2} x_t + W^{y^{(m_1)}}_{I,2} y_t^{(m_1)}\right), & \text{Pattern 2} \\ \Phi\left(W^h_{L,3} h_{t-1} + W^x_{I,3} x_t + +W^{y^{(m_2)}}_{I,3} y_t^{(m_2)}\right), & \text{Pattern 3} \\ \Phi\left(W^h_{L,4} h_{t-1} + W^x_{I,4} x_t\right), & \text{Pattern 4} \end{cases}$$

Recurrent Neural Network with Asynchronous frequency (Pattern formulation)



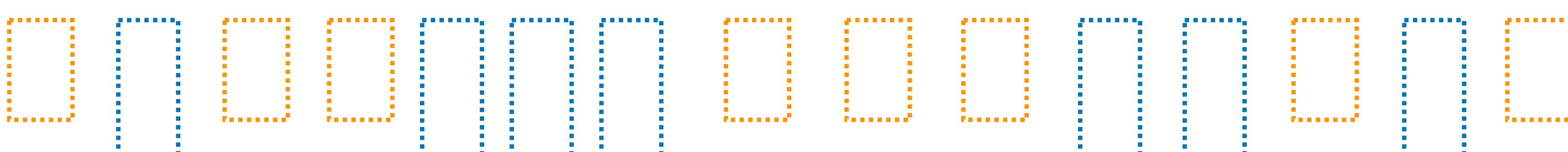
Pattern one cell formulation:

$$h_t = \Phi(W_h^1 h_{t-1} + W_x^1 x_t + W_y^1 y_t^{predict})$$

Pattern two cell formulation:

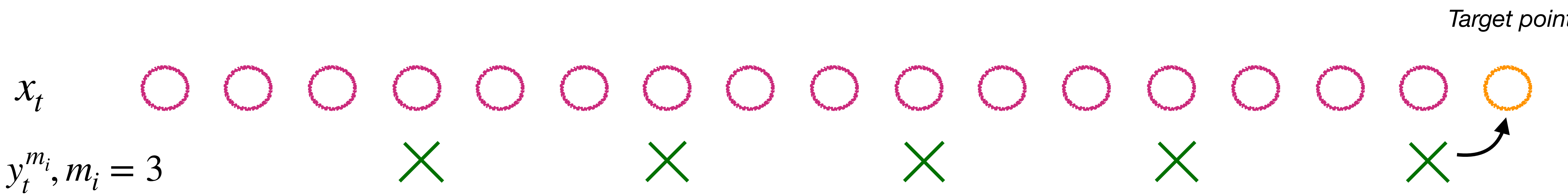
$$h_t = \Phi(W_h^2 h_{t-1} + W_x^2 x_t + W_y^2 y_t)$$

RNN formulation:



This formation method can be extended to situations with asynchronous inputs, as long as all the patterns are developed, which is possible under simple scenarios.

Recurrent Neural Network chain of development (formulations)



Regular recurrent network :

Fixed orders | No missing data | One set of weight

$$h_t = \Phi(W_h h_{t-1} + W_x x_t + W_y \hat{y}_t) \qquad h_{t+1} = \Phi(W_h h_t + W_x x_{t+1} + W_y \hat{y}_{t+1}) \qquad h_{t+2} = \Phi(W_h h_{t+1} + W_x x_{t+2} + W_y \hat{y}_{t+2})$$

Original mixed frequency recurrent network :

Fixed orders | Can fill with zero | Three sets of weight

$$h_t = \Phi(W_h^1 h_{t-1} + W_x^1 x_t + W_y^1 \hat{y}_t) \qquad h_{t+1} = \Phi(W_h^2 h_t + W_x^2 x_{t+1} + W_y^2 \hat{y}_{t+1}) \qquad h_{t+2} = \Phi(W_h^3 h_{t+1} + W_x^3 x_{t+2} + W_y^3 \hat{y}_{t+2})$$

Pattern Mixed frequency recurrent network :

Random orders | Can fill with zero | Two sets of weight

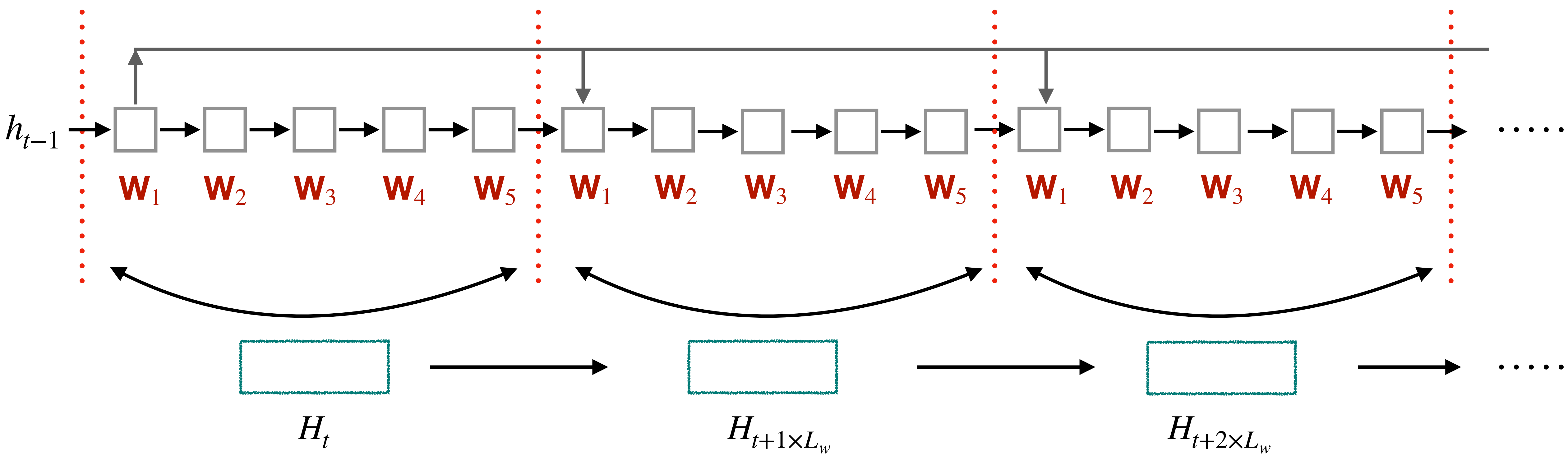
$$h_t = \Phi(W_h^1 h_{t-1} + W_x^1 x_t + W_y^1 \hat{y}_t) \qquad h_{t+1} = \Phi(W_h^2 h_t + W_x^2 x_{t+1} + W_y^2 \hat{y}_{t+1}) \qquad h_{t+2} = \Phi(W_h^2 h_{t+1} + W_x^2 x_{t+2} + W_y^2 \hat{y}_{t+2})$$

The same formulation

Mixed frequency RNN theoretical results (intuition/insight)

MF-RNN aggregated through time:

Simple example $L_w = 6$



This novel idea still remains to be a problem

$$h_{t+6} = \Phi(W_h^1 h_{t+5} + W_x^1 x_{t+6} + W_y^1 \hat{y}_{t+6})$$

$$h_{t+5} = \Phi(W_h^5 h_{t+4} + W_x^5 x_{t+5} + W_y^4 \hat{y}_{t+5})$$

$$h_{t+4} = \Phi(W_h^4 h_{t+3} + W_x^4 x_{t+4} + W_y^3 \hat{y}_{t+4})$$

$$h_{t+3} = \Phi(W_h^3 h_{t+2} + W_x^3 x_{t+3} + W_y^3 \hat{y}_{t+3})$$

How do we formulate the new aggregated network H?

$$h_{t+6} = \Phi(\Phi(\Phi(\Phi(\Phi(\ldots(h_t)))))$$

Mixed frequency RNN theoretical results (Stability results)

Stable recurrent models:

Stable recurrent networks
John Miller & Moritz Hardt (ICLR 2019)

A *recurrent model* is a non-linear dynamical system given by a differentiable *state-transition map* $\phi_w : \mathbf{R}^n \times \mathbf{R}^d \to \mathbf{R}^n$, parameterized by $w \in \mathbf{R}^m$. The hidden state $h_t \in \mathbf{R}^n$ evolves in discrete time steps according to the update rule

$$h_t = \phi_w(h_{t-1}, x_t)\,, \tag{1}$$

Definition 1. *A recurrent model ϕ_w is* stable *if there exists some $\lambda < 1$ such that, for any weights $w \in \mathbf{R}^m$, states $h, h' \in \mathbf{R}^n$, and input $x \in \mathbf{R}^d$,*

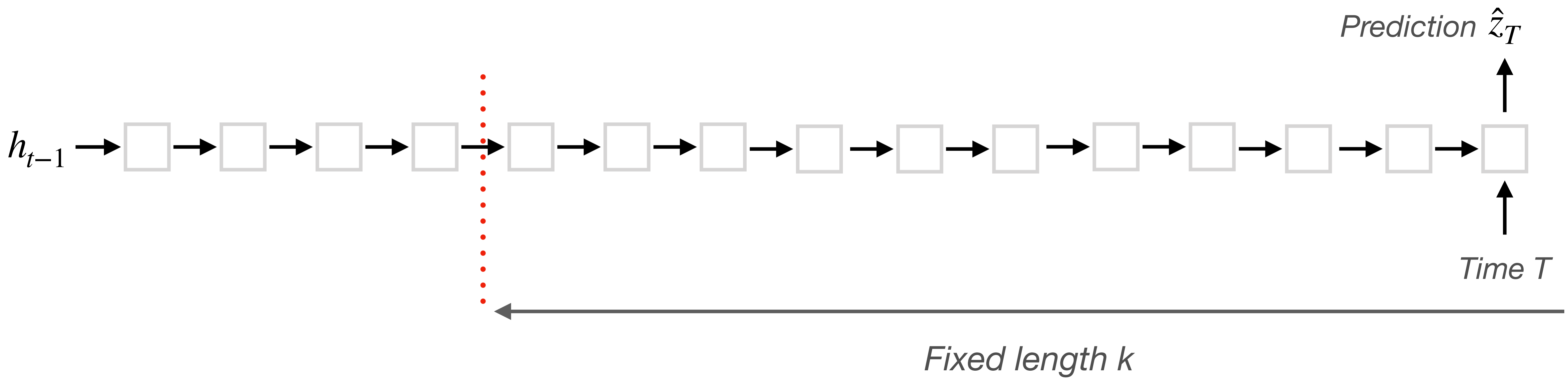
$$\|\phi_w(h, x) - \phi_w(h', x)\| \le \lambda \|h - h'\|\,. \tag{2}$$

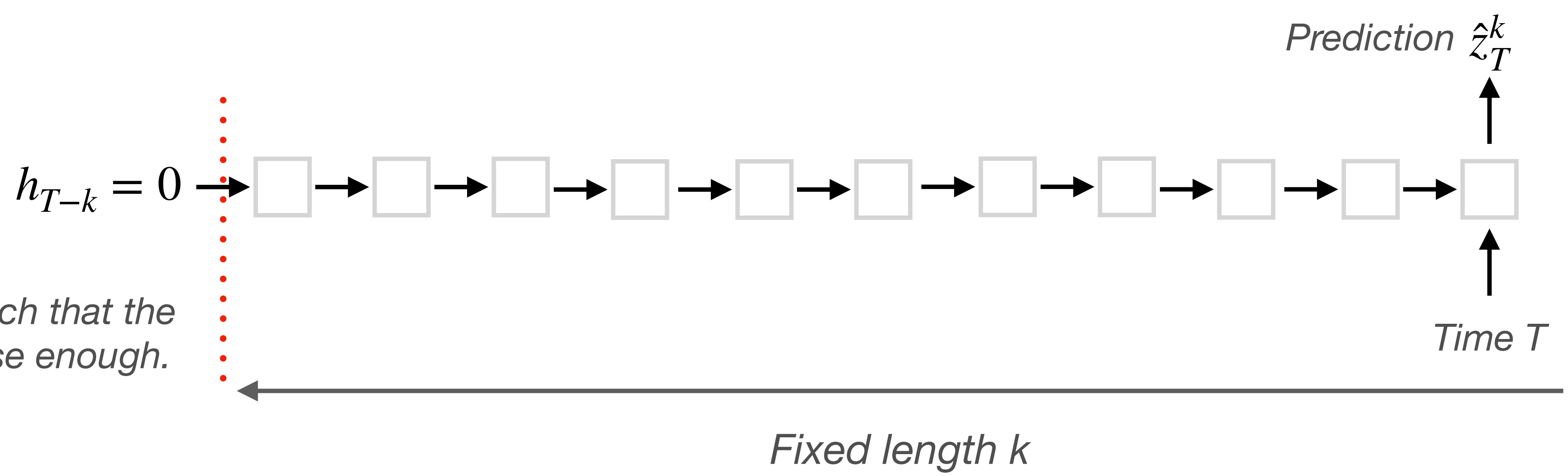
Another way of thinking the definition:

- *The gradient with respect to h will always be under 1.*
- *If we use gradient decent to learn the parameters, the long term gradient will not explode.*

Mixed frequency RNN theoretical results (Stability results)

Stable RNN can be approximated by a fixed length RNN of itself



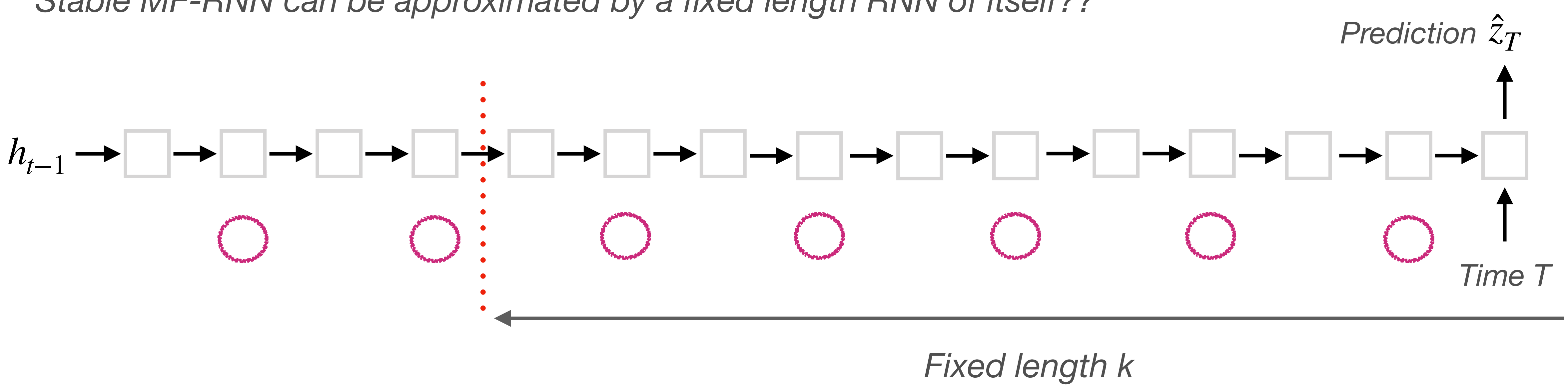


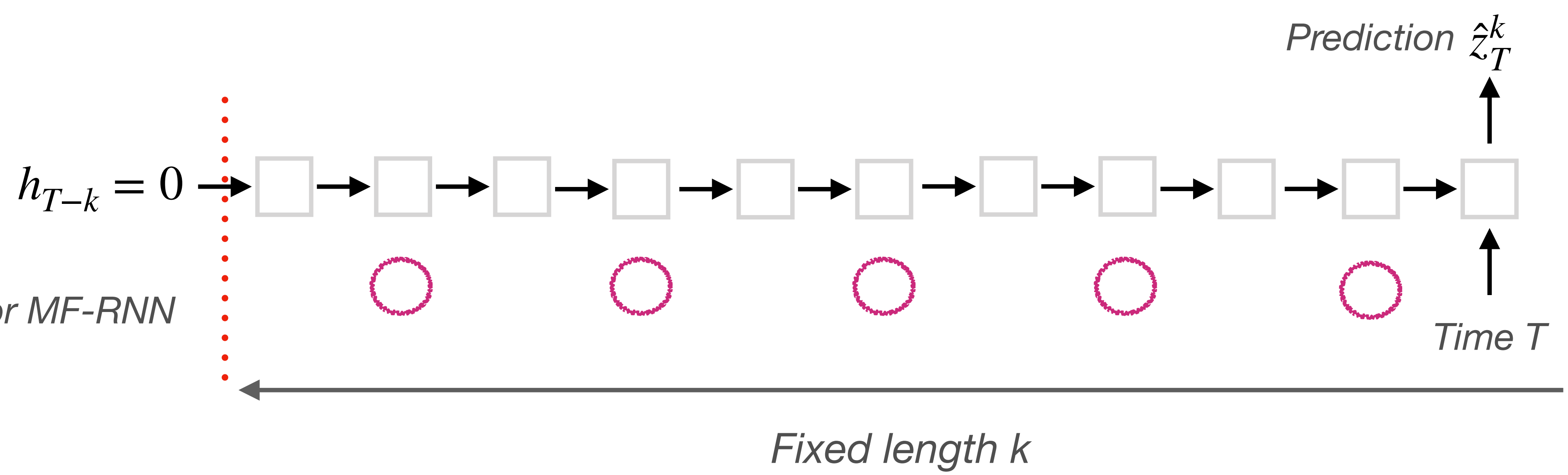
There is a bound for k such that the prediction value are close enough.

$$\|\hat{z}_T^k - \hat{z}^T\| \leq \epsilon$$

Mixed frequency RNN theoretical results (Stability results)

Stable MF-RNN can be approximated by a fixed length RNN of itself??





Is there a similar result for MF-RNN such that

$$\|\hat{z}_T^k - \hat{z}^T\| \leq \epsilon$$

Mixed frequency RNN theoretical results (Stability results)

Stability result for regular RNN:

Theorem 1. *Let p be Lipschitz and smooth. Assume ϕ_w is smooth, λ-contractive, Lipschitz in x and w. Assume the inputs are bounded, and the prediction function f is L_f-Lipschitz. If* $k \geq \Omega(\log(\gamma N^\beta/\varepsilon))$, *then after N steps of projected gradient descent with step size $\alpha_t = 1/t$,* $\|y_T - y_T^k\| \leq \varepsilon$.

Similar result for MF-RNN (m=2):

Theorem 2.4 *Assume the state-transition mapping ϕ_{w_1}, ϕ_{w_2} are smooth and λ_1, λ_2-contractive, Lipschitz in both x and w. Assume the input series are bounded and the prediction function f is L_f-Lipschitz. If the truncation length* $k \geq \max\left\{\frac{2W_n(\log(\lambda_1\lambda_2)\epsilon/2E)}{\log(\lambda_1\lambda_2)}, \frac{2\log\left(\frac{(1-\lambda_1\lambda_2)\epsilon}{2L_f(A+B)}\right)}{\log(\lambda_1\lambda_2)} + 2\right\}$, *under the projected gradient descent condition with step length $\alpha_t = \frac{\alpha}{t}$,* $\|z_t - z_t^k\| \leq \epsilon$.

Regular RNN theoretical results (Convergence results)

Convergence result for regular RNN:

Theorem 1 *Suppose that the error function is given by* (10), *that the weight sequence* $\{\mathbf{w}^k\}$ is generated by the algorithm (14) for any initial value $\mathbf{w}^0$, that Assumptions (A1) and (A2) are valid, and that η is small enough such that (23) below is valid. Then, we have

(a) $E(\mathbf{w}^{k+1}) \leq E(\mathbf{w}^k), \quad k = 0, 1, 2, \ldots;$

(b) *There is* $E^* \geq 0$ *such that* $\lim_{k\to\infty} E(\mathbf{w}^k) = E^*$;

(c) $\lim_{k\to\infty} \|\Delta\mathbf{w}^k\| = 0, \quad \lim_{k\to\infty} \left\|\frac{\partial E(\mathbf{w}^k)}{\partial \mathbf{w}}\right\| = 0.$

Moreover, if Assumption (A3) is also valid, then we have the strong convergence:

(d) *There exists* $\mathbf{w}^* \in \Phi_0$ *such that* $\lim_{k\to\infty} \mathbf{w}^k = \mathbf{w}^*$.

Convergence of gradient method for a fully recurrent neural network
Dongpo Xu, Zhengxue Li, Wei Wu (2010)

L2 loss function is decreasing

Gradient decent (iteration k) to update the weight parameters

η *Is the learning step size*

Assumptions:

Lemma 2.1. *Let* $g : D \subset R^n \to R^1$ *be continuously differentiable on the compact set* $D_0 \subset D$*, and suppose that the set* Ω *of critical points of* g *in* D_0 *is finite. Let* $\{x^k\} \subset D_0$ *be any sequence for which* $\lim_{k\to\infty}(x^k - x^{k+1}) = 0$ *and* $\lim_{k\to\infty} g'(x^k)^T = 0$*. Then* $\lim_{k\to\infty} x^k = x^*$ *and* $g'(x^*)^T = 0$.

Lemma 2.2. *Let* $g : D \subset R^n \to R^1$ *be twice F-differentiable in the open set* $D_0 \subset D$*. Let* $\{x^k\} \subset D_0$ *satisfy* $\lim_{k\to\infty}(x^k - x^{k+1}) = 0$ *and* $\lim_{k\to\infty} g'(x^k)^T = 0$*. If* $\{x^k\}$ *has a limit point* x^* *for which* $H_g(x^*)$ *is non-singular, then* $\lim_{k\to\infty} x^k = x^*$.

Convergence result for MF-RNN:

Theorem 2.9. *Under similar assumptions, there exist a constant C_η, such that if the learning rate $\eta < \frac{1}{C_\eta}$, the loss function $E(\boldsymbol{w}_1^k, \boldsymbol{w}_2^k)$ is a decreasing function w.r.t both w_1, w_2 and there exists a limit value $E^* \geq 0$ such that $\lim_{k\to\infty} E(\boldsymbol{w}_1^k, \boldsymbol{w}_2^k) = E^*$.* (a), (b)

Theorem 2.10. *Under the assumptions of theorem 4.4, the gradient of the loss function converges to 0,* (c)

$$\lim_{k\to\infty} \|\frac{E(\boldsymbol{w}_1^k, \boldsymbol{w}_2^k)}{\partial \boldsymbol{w}_1}\| = 0, \lim_{k\to\infty} \|\frac{E(\boldsymbol{w}_1^k, \boldsymbol{w}_2^k)}{\partial \boldsymbol{w}_2}\| = 0$$

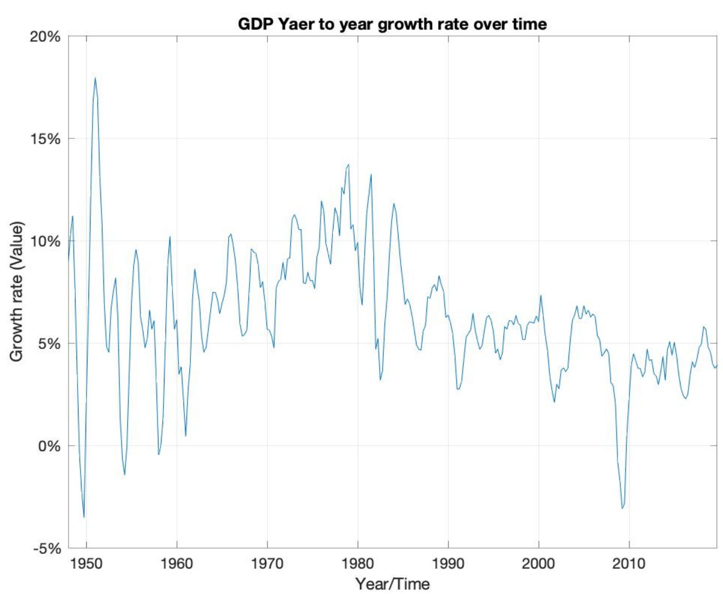
Theorem 2.11. *Under assumption A.3* (d)

$$\lim_{k\to\infty} \boldsymbol{w}_1 = \boldsymbol{w}_1^*$$
$$\lim_{k\to\infty} \boldsymbol{w}_2 = \boldsymbol{w}_2^*.$$

Mixed frequency RNN numerical results (quarterly & monthly)

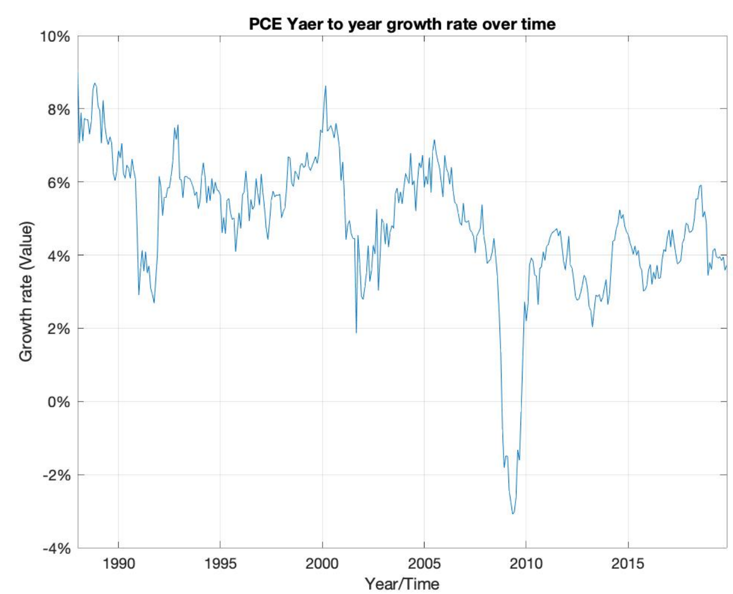
Real data experiment: Low frequency data - (GDP), high frequency data - (PCE)

GDP: gross domestic product *PCE: personal consumption expenditure*



Quarterly data

+



Monthly data

By using the quarterly data, we want to get a better prediction of the next monthly data output

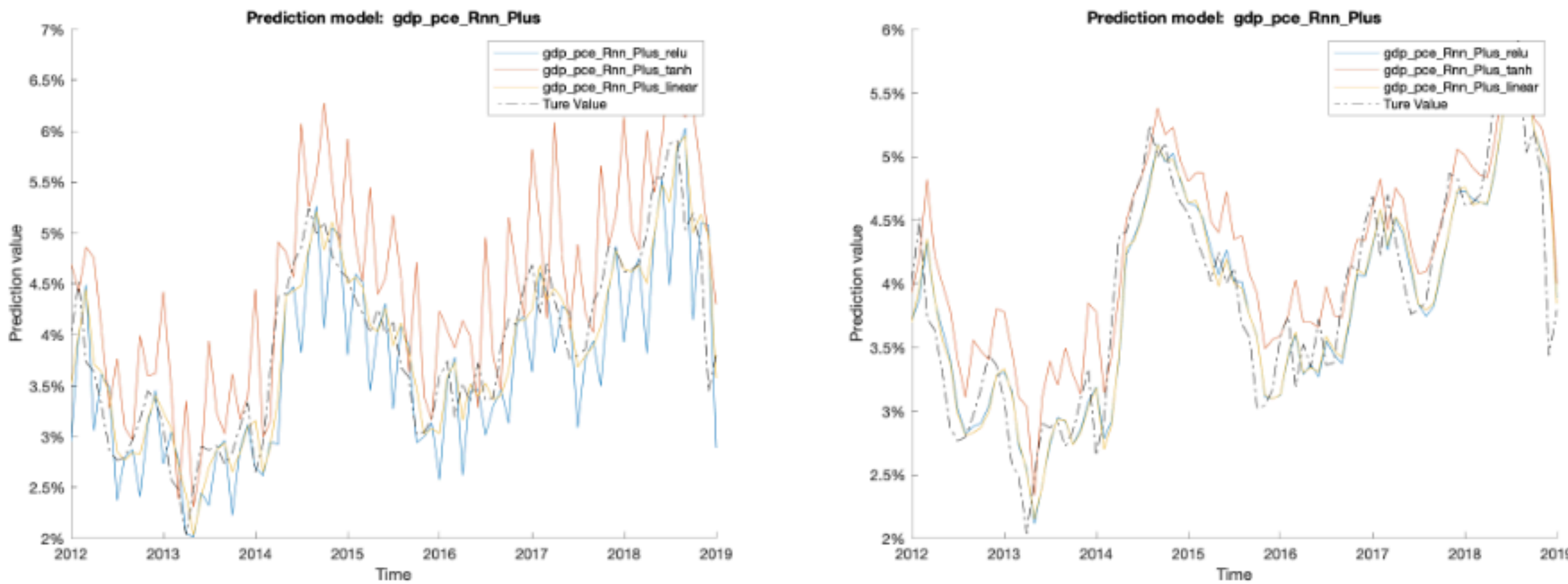
By using the monthly data, we want to get a better prediction of the next quarterly data output

Several interesting points about this experiment:

- Few data points available.
- Data source (every country is different).
- The correlations between the data selected.

MF-RNN the problem of over fitting (quarterly & monthly)

The difference between pattern formulation and generic formulation (both hidden layer d=8)



(a) Generic MF-RNN formulation.

(b) Pattern MF-RNN formulation.

Figure 16: Forecasting monthly Personal Consumption Expenditure (PCE) growth rate with GDP under two types of MF-RNN formulation. Plot (a) has three sets of weights while plot (b) only contains two sets. Blue, red, yellow line represent the results by using ReLU, tanh and linear activation function respectively.

In this case, generic formulation causes over fitting. Pattern formation performs better.

Correlations among data (quarterly & monthly correlation)

The effect of data correlation

HF: Monthly data	LF: GDP	Forecasting monthly data with quarterly variable (GDP) $m = 3$			
		MF Generic Form	MF Pattern Form	RRMIDAS	URMIDAS
Personal Consumption (**PCE**) Expenditure	Relu	0.352	0.144	0.1683*	0.1746**
	tanh	0.640	0.242	0.1683*	0.1746**
	linear	0.155	0.143	0.1683*	0.1746**
Consumer Price (**CPI**) Index	Relu	0.116	0.070	0.0782*	0.0911**
	tanh	0.371	0.079	0.0782*	0.0911**
	linear	0.071	0.072	0.0782*	0.0911**
Inclusive Development (**IDI**) Index	Relu	1.137	0.554	0.5181*	0.4911***
	tanh	0.599	0.927	0.5181*	0.4911**
	linear	0.527	0.523	0.5181*	0.4911**
Unemployment Rate (**UNRATE**)	Relu	0.054	0.0240	0.0241*	0.0263**
	tanh	0.547	0.0240	0.0241*	0.0263**
	linear	0.194	0.0251	0.0241*	0.0263**
Industrial Production (**IPI**) Index	Relu	4.611	3.673	2.9545*	3.5965**
	tanh	7.998	4.111	—	—
	linear	3.372	3.053	—	—

Good!

*, **: Midas Restricted and Unrestricted Models don't contain activation functions, * is added for easier comparison across lines.

Table 8: MSE for UR-Mixed-Frequency RNN and benchmarks comparison. (Low Frequency: GDP, High Frequency: PCE, CPI, IDI, UNRATE)

When data are highly correlated to each other, the pattern RNN provides better results.

If not correlated, it adds noise to the model, linear regression provides better results.

The model works under certain condition.
When we are using the numbers,
we are also using the structures behind the data, which is more important in this case.

R Samans, J Blanke, M Drzeniek, and G Corrigan. The inclusive development index 2018 summary and data highlights. In World Economic Forum, Geneva, Switzerland, 2018

A more granular look at the data shows that GDP per capita is rather weakly correlated with performance on IDI indicators other than labor productivity and healthy life expectancy[1] (and poverty rates in advanced economies). This highlights a **key**

An example for the IDI variable.

Mixed frequency RNN future improvements

Add structure restrictions:

1. For a regular RNN, we can impose structure restrictions on the weight matrix, for example, orthogonality.

On orthogonality and learning recurrent networks with long term dependencies
(2017) Eugene Vorontsov, Chiheb Trabelsi, Samuel Kadoury, Chris Pal.

2. Change the loss function formulation. For example, instead of the L2 loss,

$$loss = \|z_t - \hat{z}_t\|^2 + \lambda \|W^T W - I_h\|$$

★ *3. Limitations on the number of parameters. Instead of having multiple independent sets of weights, the weights can share the same variables. Take the general formulation as an example.*

General MF-RNN
m=2

$$h_t = \Phi\left(K_h(\theta_0, t)h_{t-1} + g\left(K_x^1(\theta_1, t)x_t + K_x^2(\theta_2, t)x_{t-1} + K_y(\theta_3, t)\hat{y}_t\right)\right)$$

Where we can restrict $\theta_1 = \theta_2$

4. Extend to more advanced RNN structure, for example, LSTM, GRU.

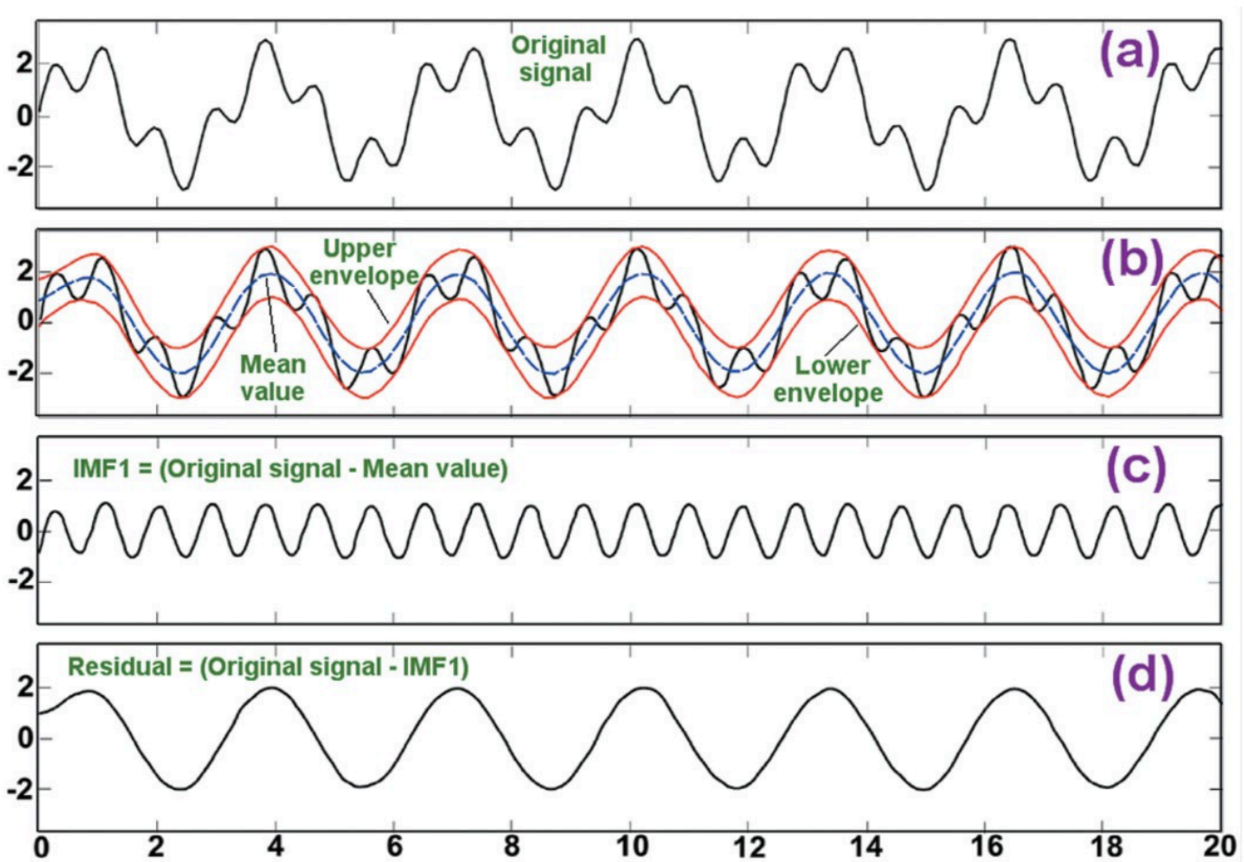
Mixed frequency RNN potential application

Another very interesting proposal:

The MF-RNN was used across multiple data sets to predict. We can also use it for a single data set.

The core idea is to create multiple mixed frequency data sets.

Empirical mode decomposition. (EMD)



The original signal (a) = (c) + (d)

The EMD method (b)

The decomposition results.

Figure source: https://towardsdatascience.com/decomposing-signal-using-empirical-mode-decomposition-algorithm-explanation-for-dummy-93a93304c541

To be continued…

Thank you!