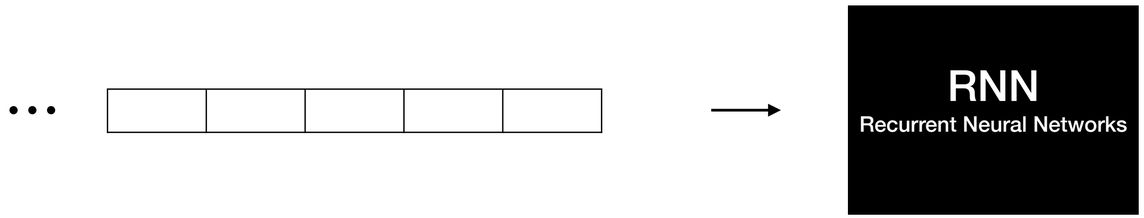
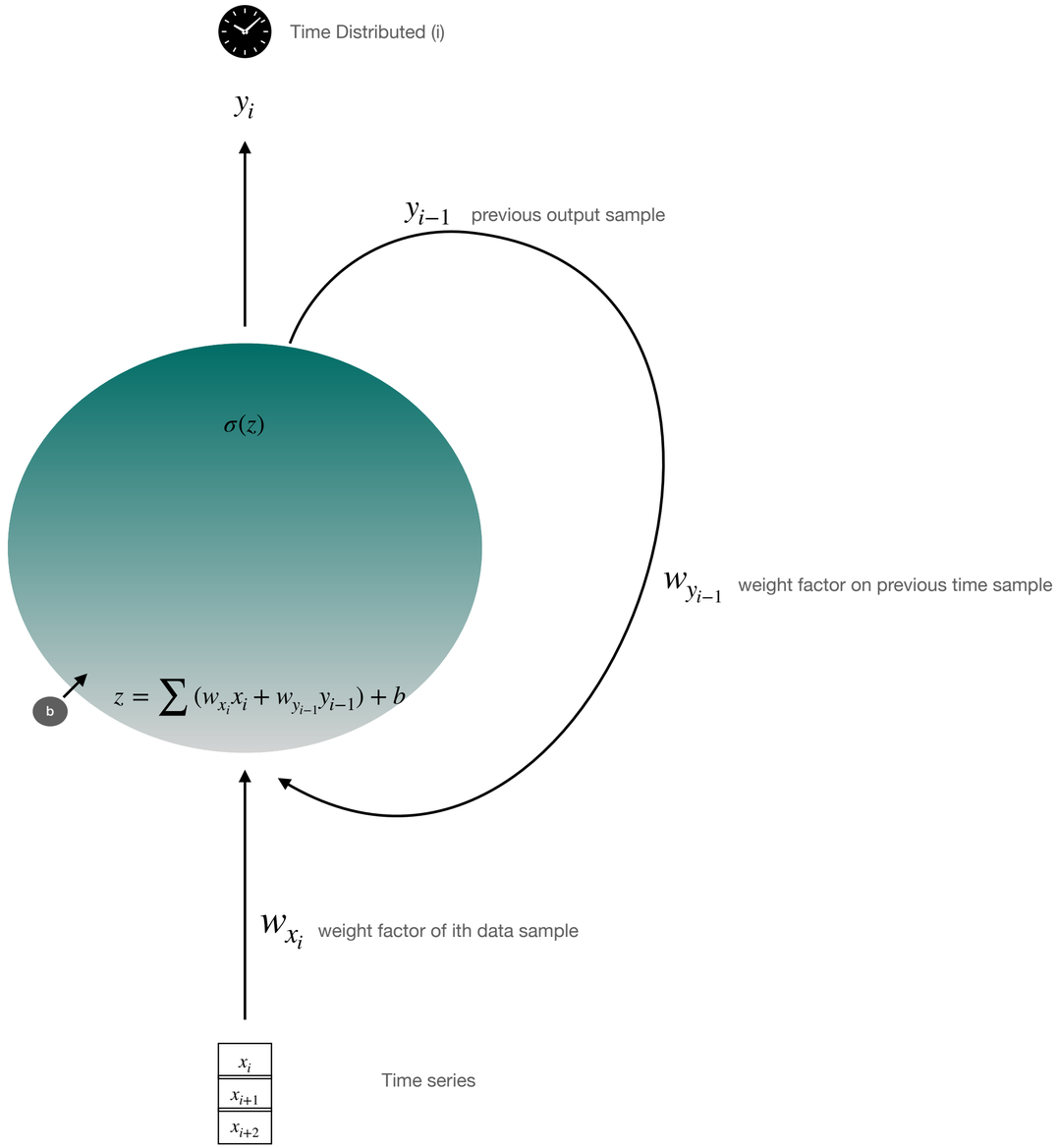
Processing Sequences

Chapter 14

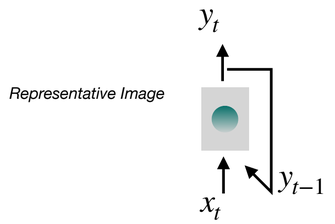
Hector Williams

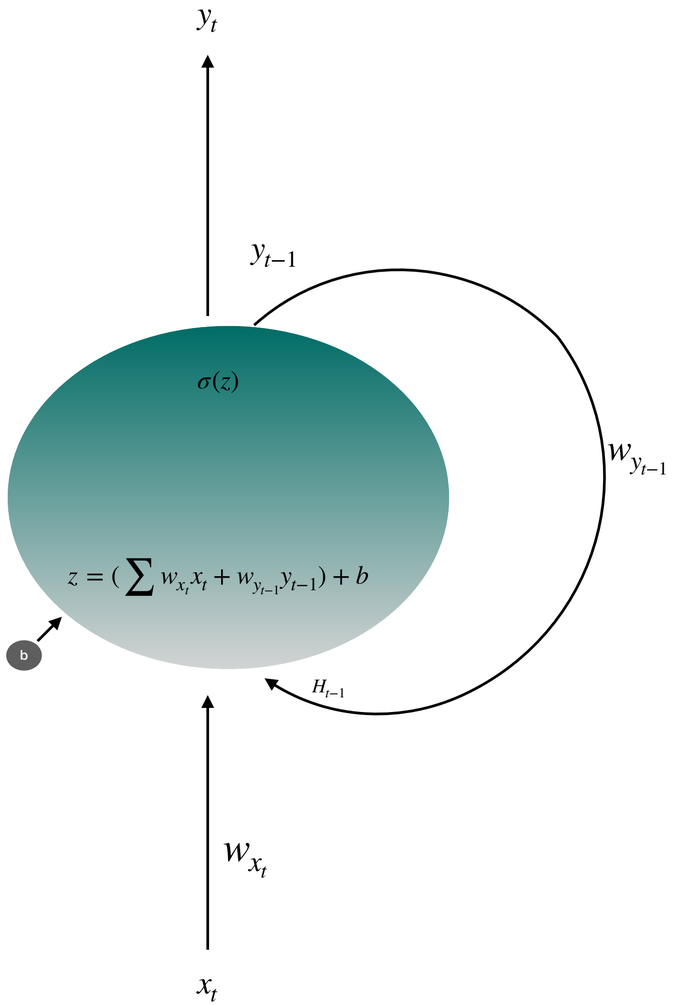

RNN
Recurrent Neural Networks

Recurrent Neuron



Recurrent Neuron



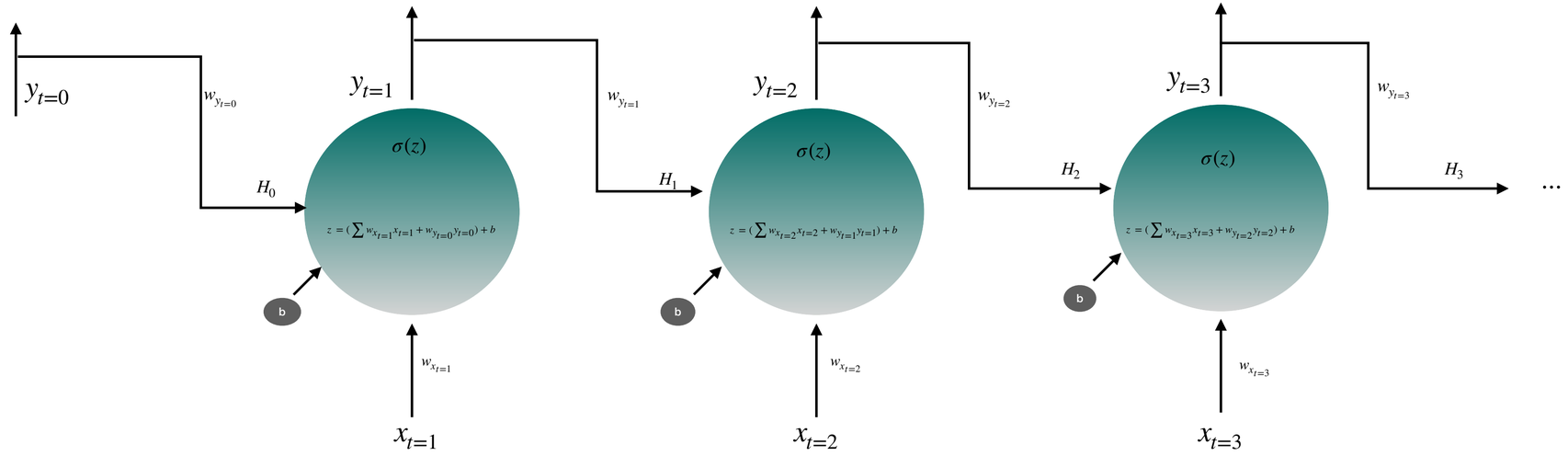


Time Distributed (t)

Equivalence…these distributed outputs act as ROM

y_0	y_1	y_2	y_3

Frame



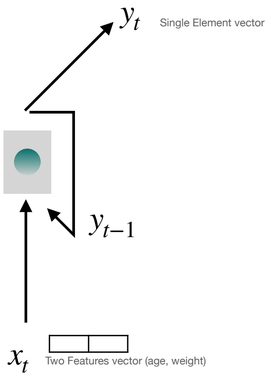
Single neuron being reused

Unrolled over time

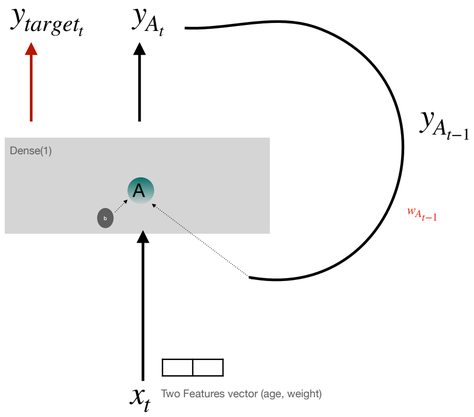
Time

Recurrent Neuron Layer

Basic Representation



Preferred Representation



Bias vector

b_t

b_{A_t}

Output vector

y_t

y_{1_t}

y_{2_t}

Feedback Delay vector

y_{t-1}

$y_{A_{t-1}}$

Input vector

x

x_{age_t}

x_{weight_t}

Input weight vector

w_x

$w_{x_{age}}$

$w_{x_{weight}}$

Feedback weight vector

$w_{y_{t-1}}$

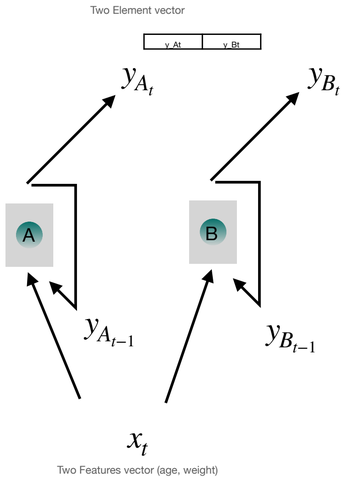
$w_{A_{t-1}}$

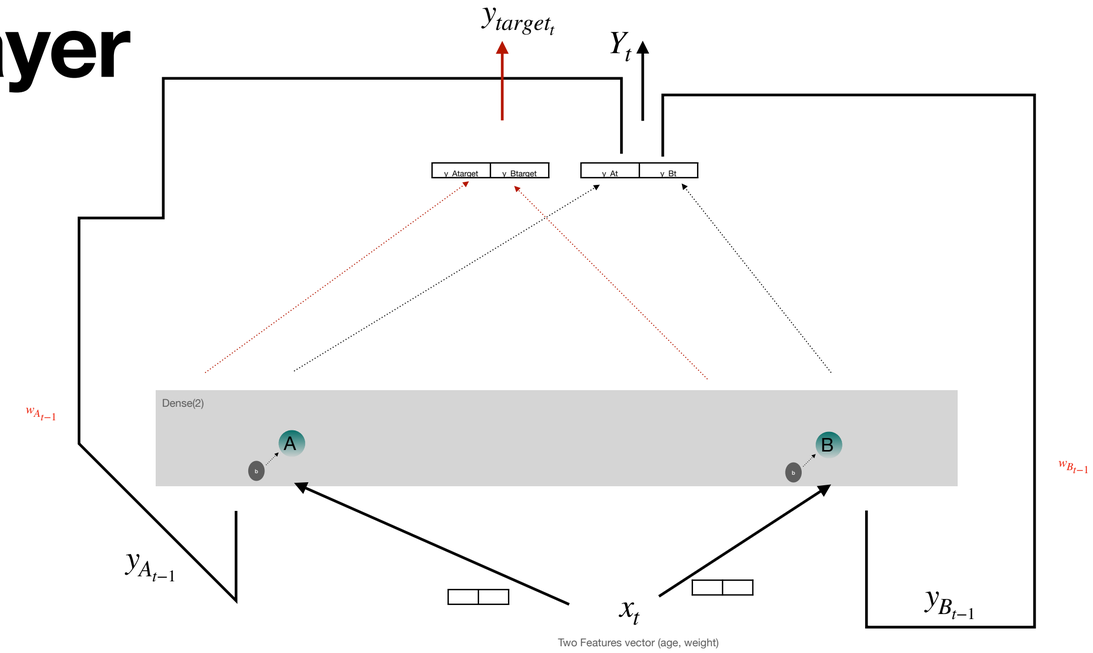
Transpose

$w_{x_{age}}$ $w_{x_{weight}}$ $\times$ x_{age_t} x_{weight_t} $+$ Transpose $w_{A_{t-1}}$ $\times$ $y_{A_{t-1}}$ $+$ b_{A_t} $=$ z_{A_t}

$\sigma(z_{A_t})$ $\rightarrow$ y_{A_t}

Recurrent Neuron Layer





Bias vector

b_t

b_{A_t} b_{B_t}

Output vector

y_t

y_{A_t} y_{B_t}

Feedback Delay vector

y_{t-1}

$y_{A_{t-1}}$ $y_{B_{t-1}}$

Input vector

x

x_{age_t}

x_{weight_t}

Input weight vector

w_x

$w_{A_{x_{age}}}$ $w_{B_{x_{age}}}$

$w_{A_{x_{weight}}}$ $w_{B_{x_{weight}}}$

Feedback weight vector

$w_{y_{t-1}}$

$w_{A_{t-1}}$ $w_{B_{t-1}}$

Transpose

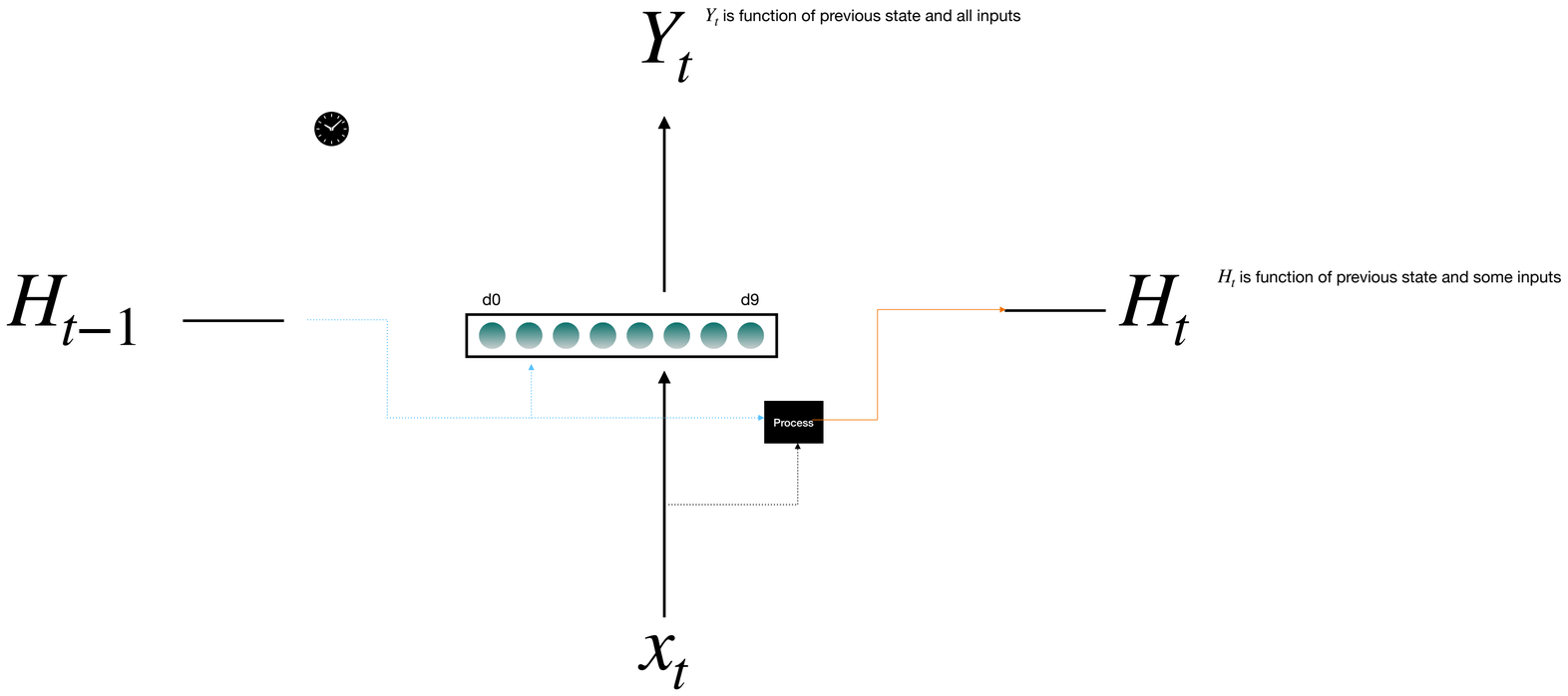
$w_{A_{x_{age}}}$ $w_{A_{x_{weight}}}$ $\times$ x_{age_t} x_{weight_t} $+$ Transpose $w_{A_{t-1}}$ $\times$ $y_{A_{t-1}}$ $+$ b_{A_t} $=$ z_{A_t}

$\sigma(z_{A_t})$ $\longrightarrow$ y_{A_t}

$w_{B_{x_{age}}}$ $w_{B_{x_{weight}}}$ $\times$ x_{age_t} x_{weight_t} $+$ $w_{B_{t-1}}$ $\times$ $y_{B_{t-1}}$ $+$ b_{B_t} $=$ z_{B_t}

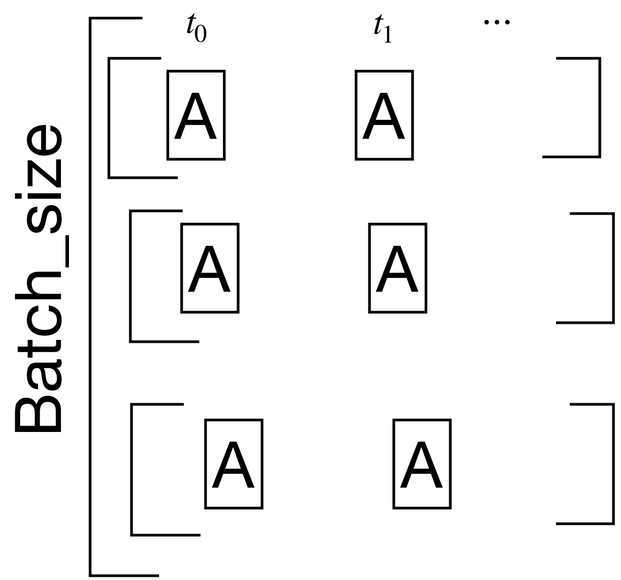
$\sigma(z_{B_t})$ $\longrightarrow$ y_{B_t}

Recurrent Neuron Layer



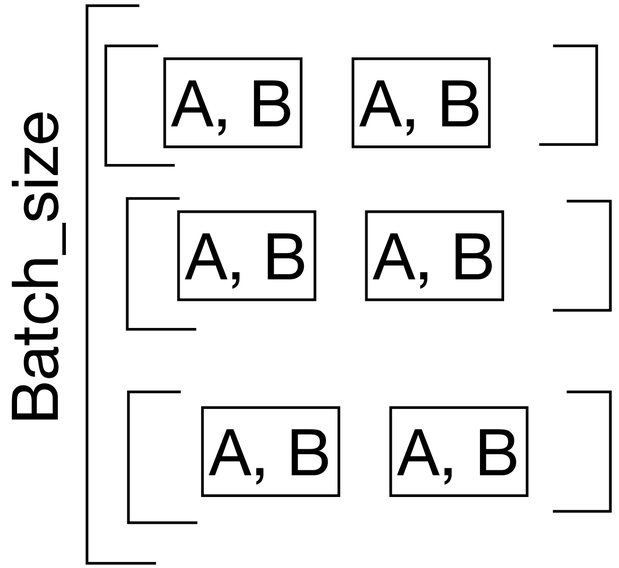
t	y
0	[d0, d1, d2,d3,d4,d5,d6,d7,d8,d9]
1	[d0', d1', d2',d3',d4',d5',d6',d7',d8',d9']
2	[d0", d1'', d2",d3",d4",d5",d6",d7",d8'',d9'']

BatchSize x 2 x 1



One Feature Inputs

BatchSize x 2 x 2



Two Feature Inputs

4 x batchsize x 1

$$\begin{bmatrix} [freq1_0] & [freq1_1] & [freq1_2] \dots \end{bmatrix}$$

$$\begin{bmatrix} [freq2_0] & [freq2_1] & [freq2_2] \dots \end{bmatrix}$$

$$\begin{bmatrix} [offset1_0] & [offset1_1] & [offset1_2] \dots \end{bmatrix}$$

$$\begin{bmatrix} [offset2_0] & [offset2_1] & [offset2_2] \dots \end{bmatrix}$$

Time Series

Parameters

$$\begin{bmatrix} [freq1_0] & [freq1_1] & [freq1_2] \dots \\ [freq2_0] & [freq2_1] & [freq2_2] \dots \\ [offset1_0] & [offset1_1] & [offset1_2] \dots \\ [offset2_0] & [offset2_1] & [offset2_2] \dots \end{bmatrix}$$

Time

$$\begin{bmatrix} time_0 & time_1 & time_2 & \dots \end{bmatrix}$$

Radians

$$time_\Delta = 1/51$$

$$wave_0 = sin((time - [offset1_0]) \times ([freq1_0] \times 10 + 10)) + sin((time - [offset2_0]) \times ([freq2_0] \times 10 + 10)) + 51\_random\_samples$$

$$wave_{dim} = 50x1$$

Waves output

$$\begin{bmatrix} wave_0 & wave_1 & wave_2 & \dots & wave_{999} & wave_{1000} \end{bmatrix}$$

$$wave_{lastvalue} = 1x1$$

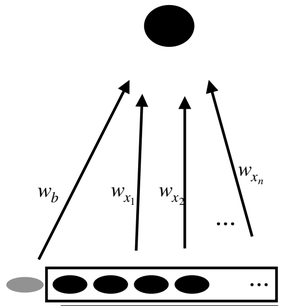
Waves output

$$\begin{bmatrix} wave_{0_{lastvalue}} & wave_{1_{lastvalue}} & wave_{2_{lastvalue}} & \dots & wave_{999_{lastvalue}} & wave_{1000_{lastvalue}} \end{bmatrix}$$

Dense vs Recurrent

Output = 1 neuron

N+1 inputs(parameters) to neuron
(+1 is bias neuron)

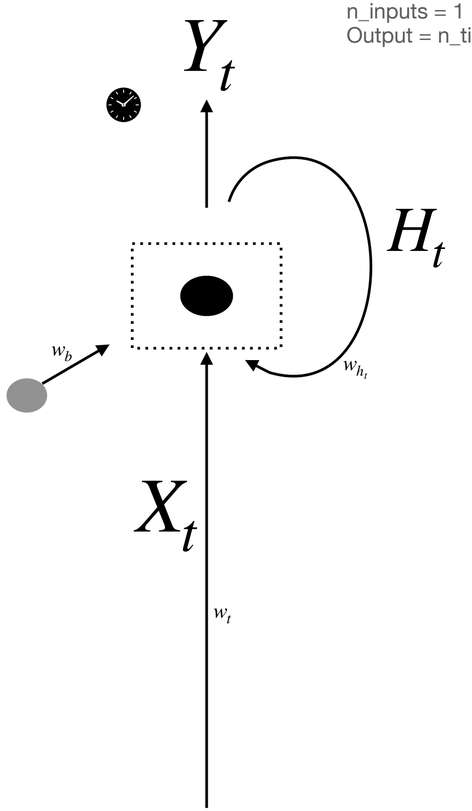


Flatten

Wave
(n=50)

Last time step output captured

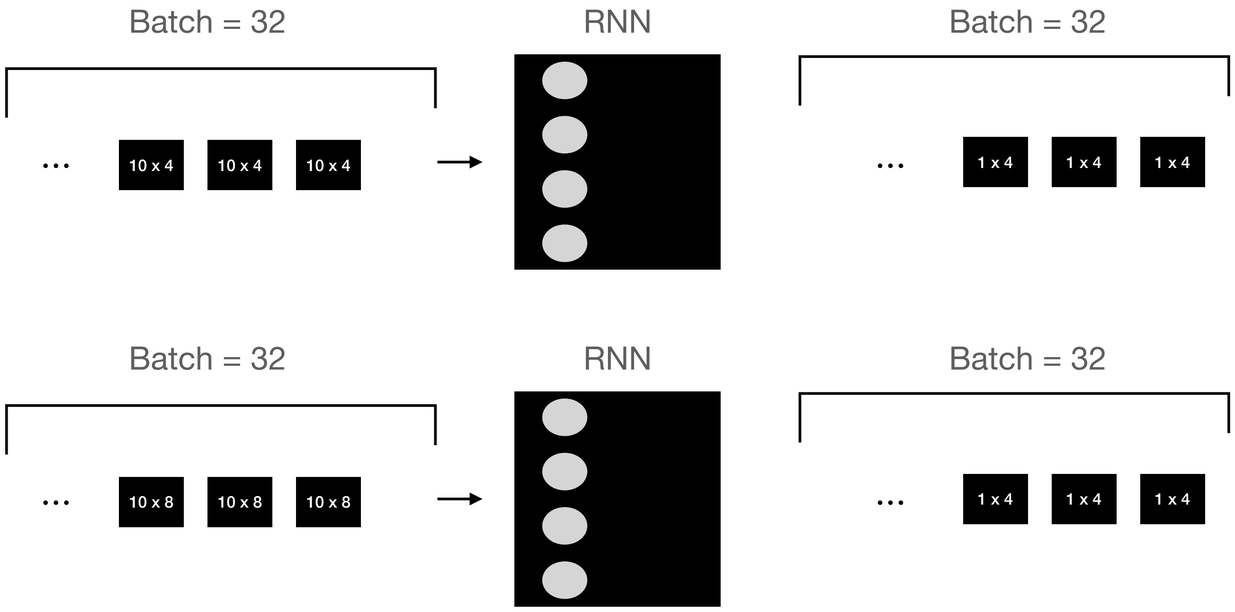
Single RNN Layer (1 units)
3 parameters
n_inputs = 1
Output = n_times_steps x n_neurons



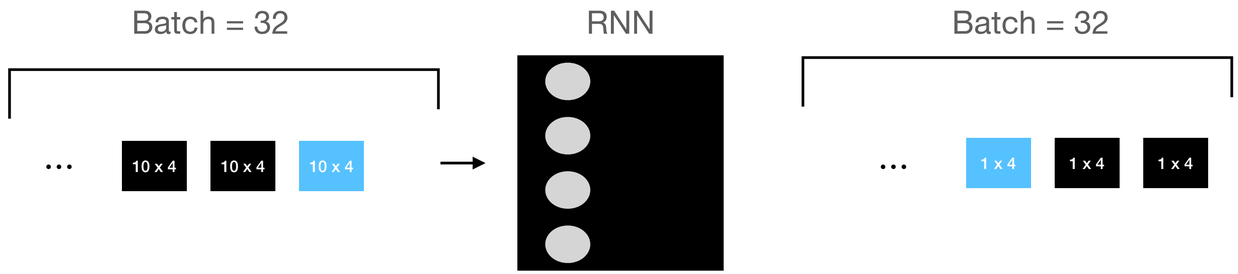
Wave
(n=50)

All time series outputs captured

I/O Recurrent Layer

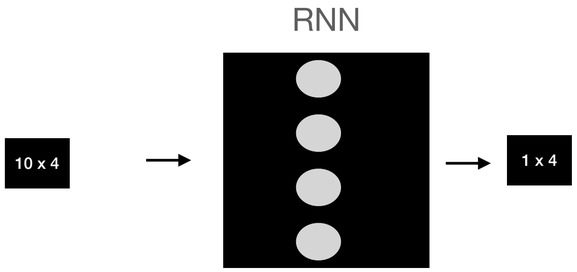


I/O Recurrent Layer

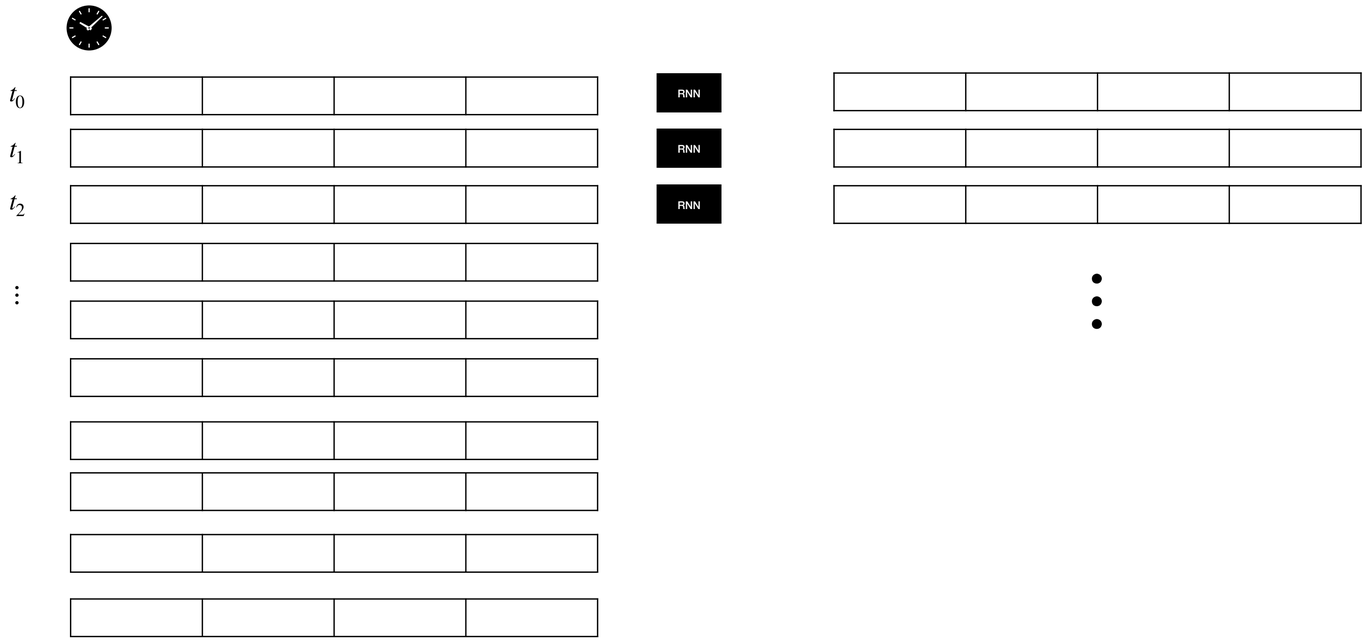


1 of 32 time series in batch

I/O Recurrent Layer



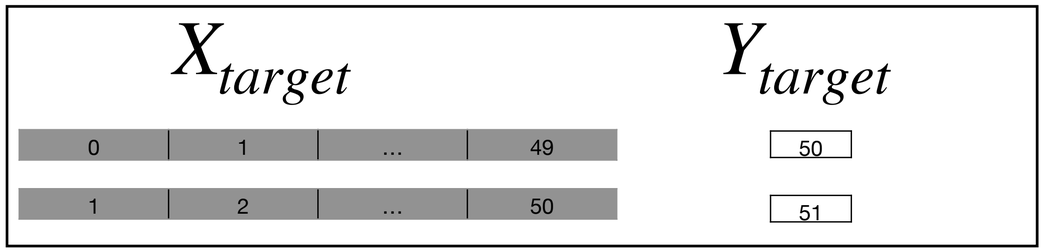
Output for $Y_{batch_{id=0}TimeSeries_{id=0}}$ has size of 10 x 4 ($num\_times\_series \times n\_neurons$)



1 of 32 time series in batch

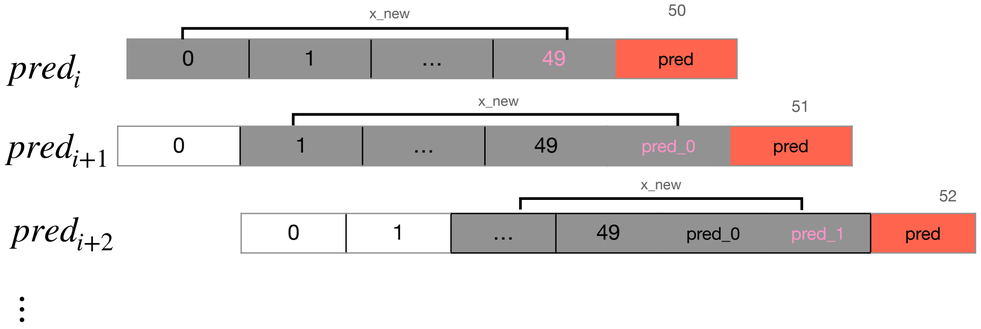
Forecast next sample(s) in Time Series

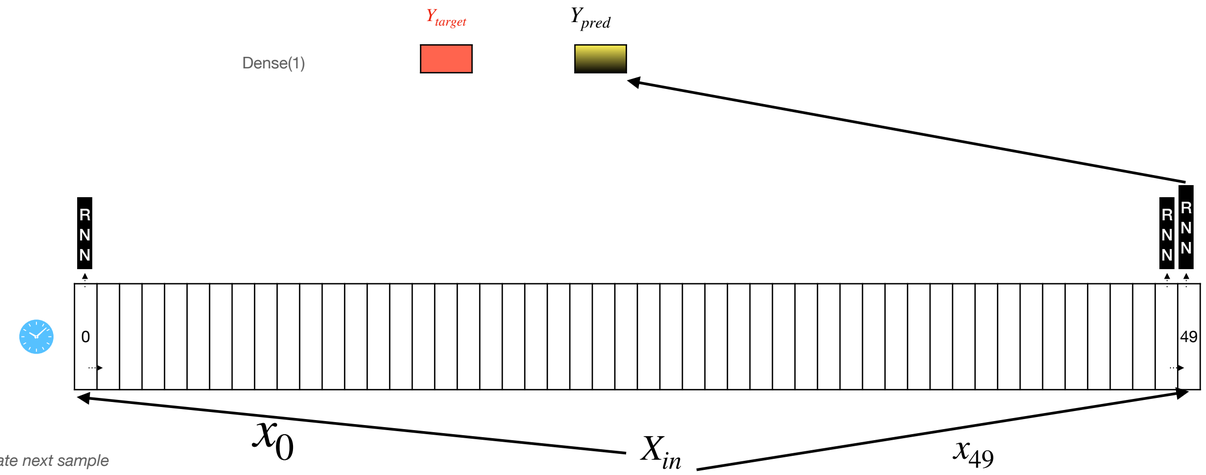
Train



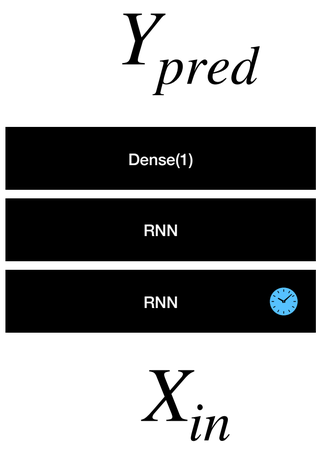
Predict next values consecutively

Forecasts next value after last sample

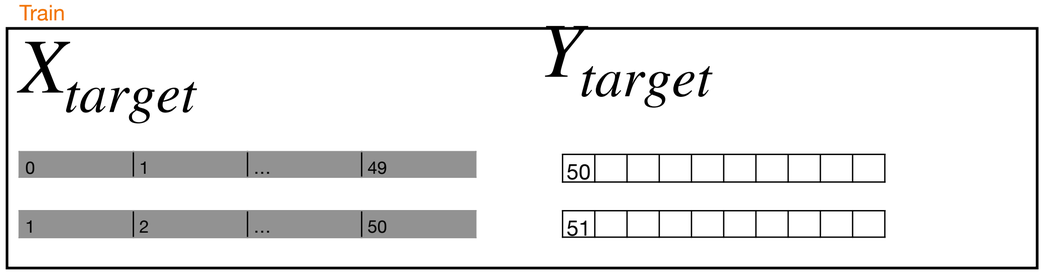


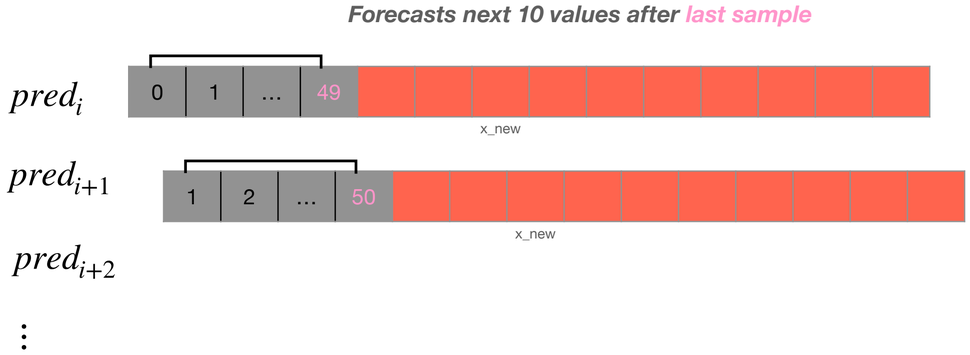


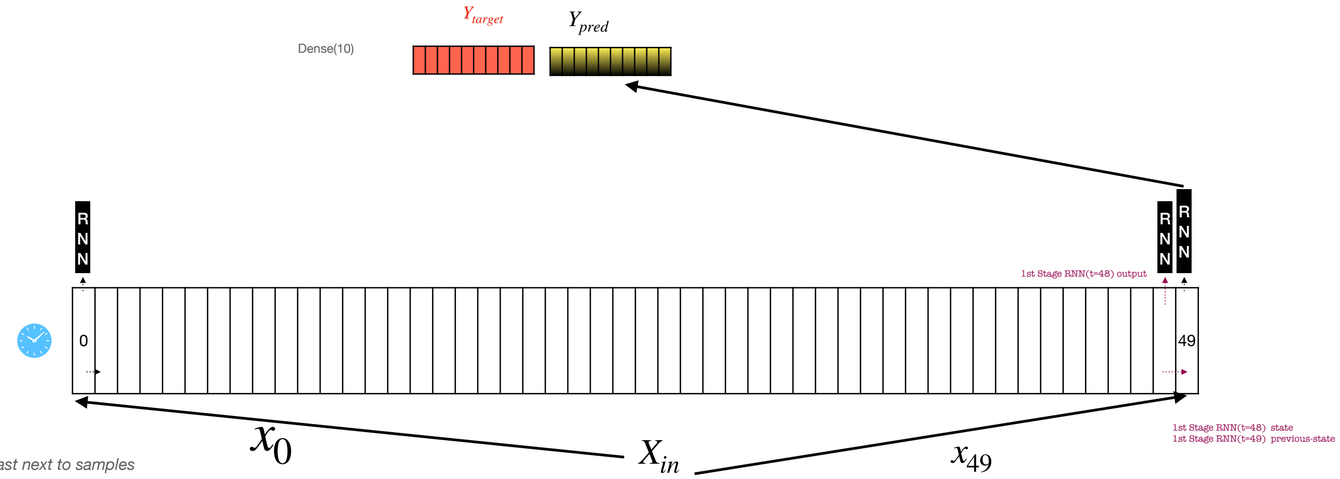
Model learns pattens to estimate next sample

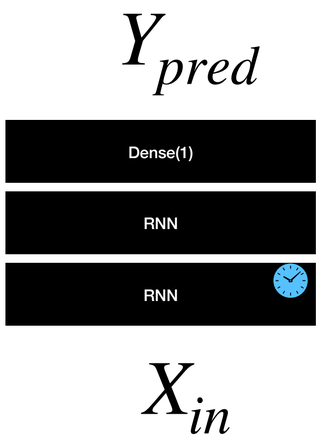


Forecast Several Steps in Time Series





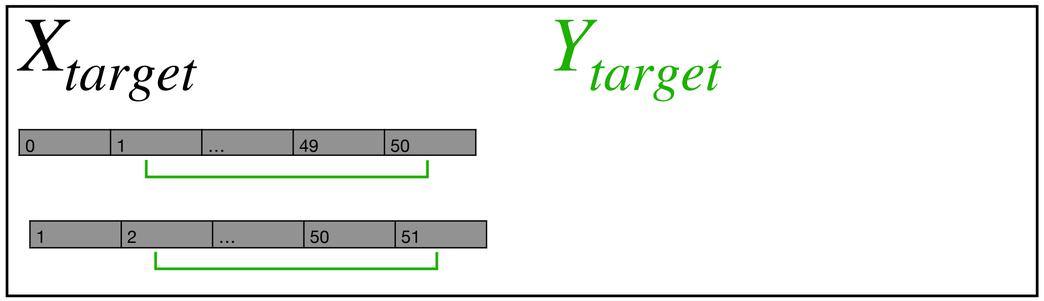




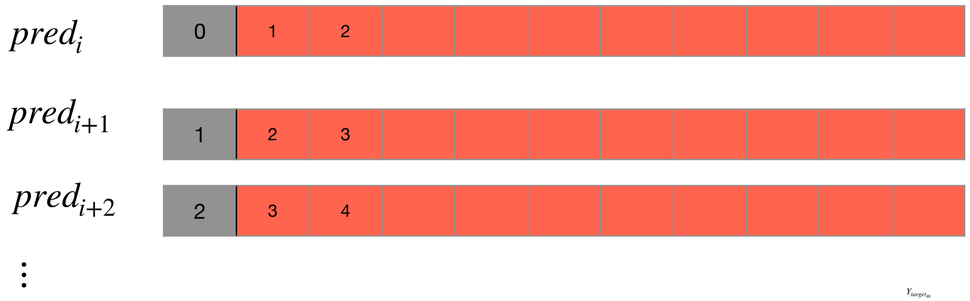
Model learns pattens to estimate last next to samples

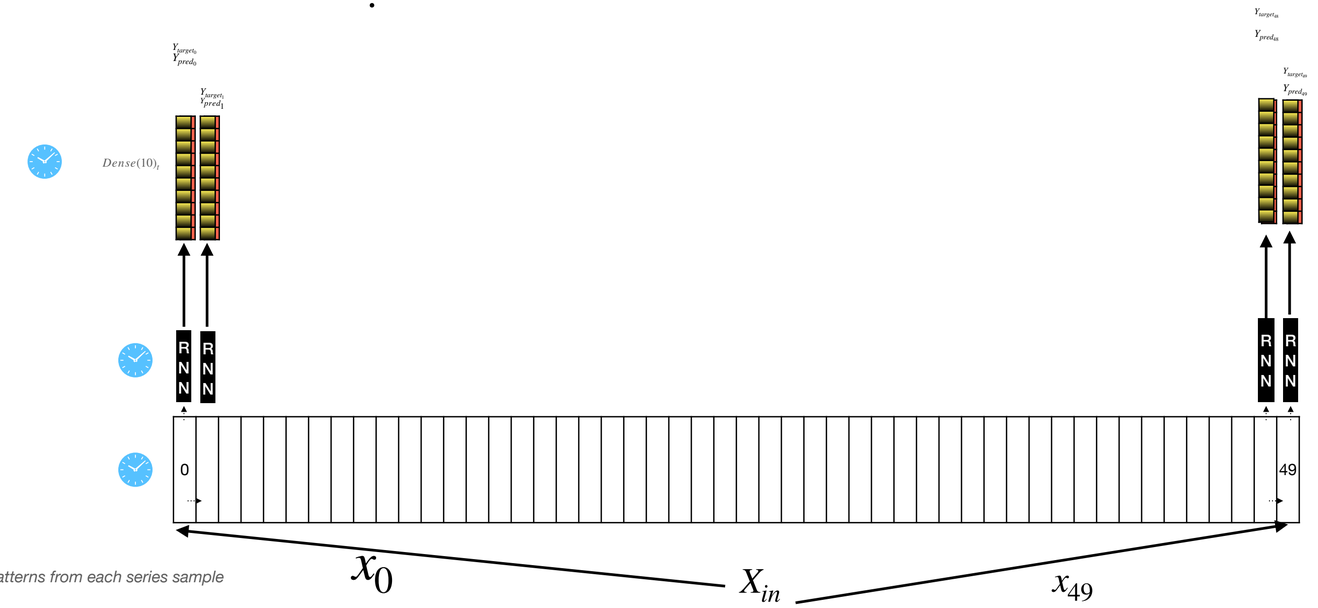
Forecast Several Steps after every step in Time Series

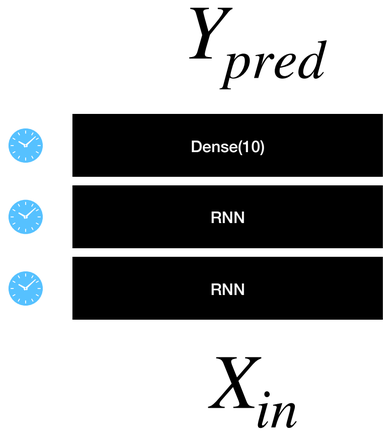
Train



Forecasts next 10 values after every sample

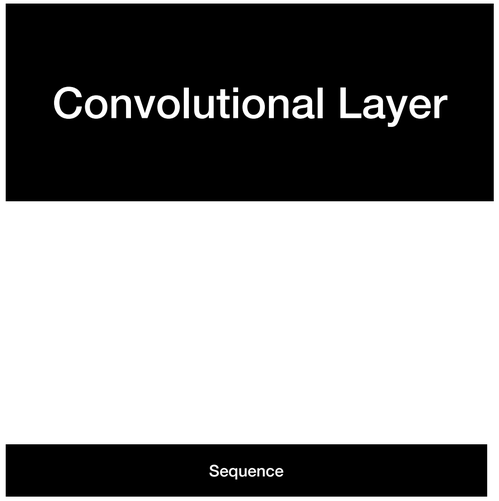




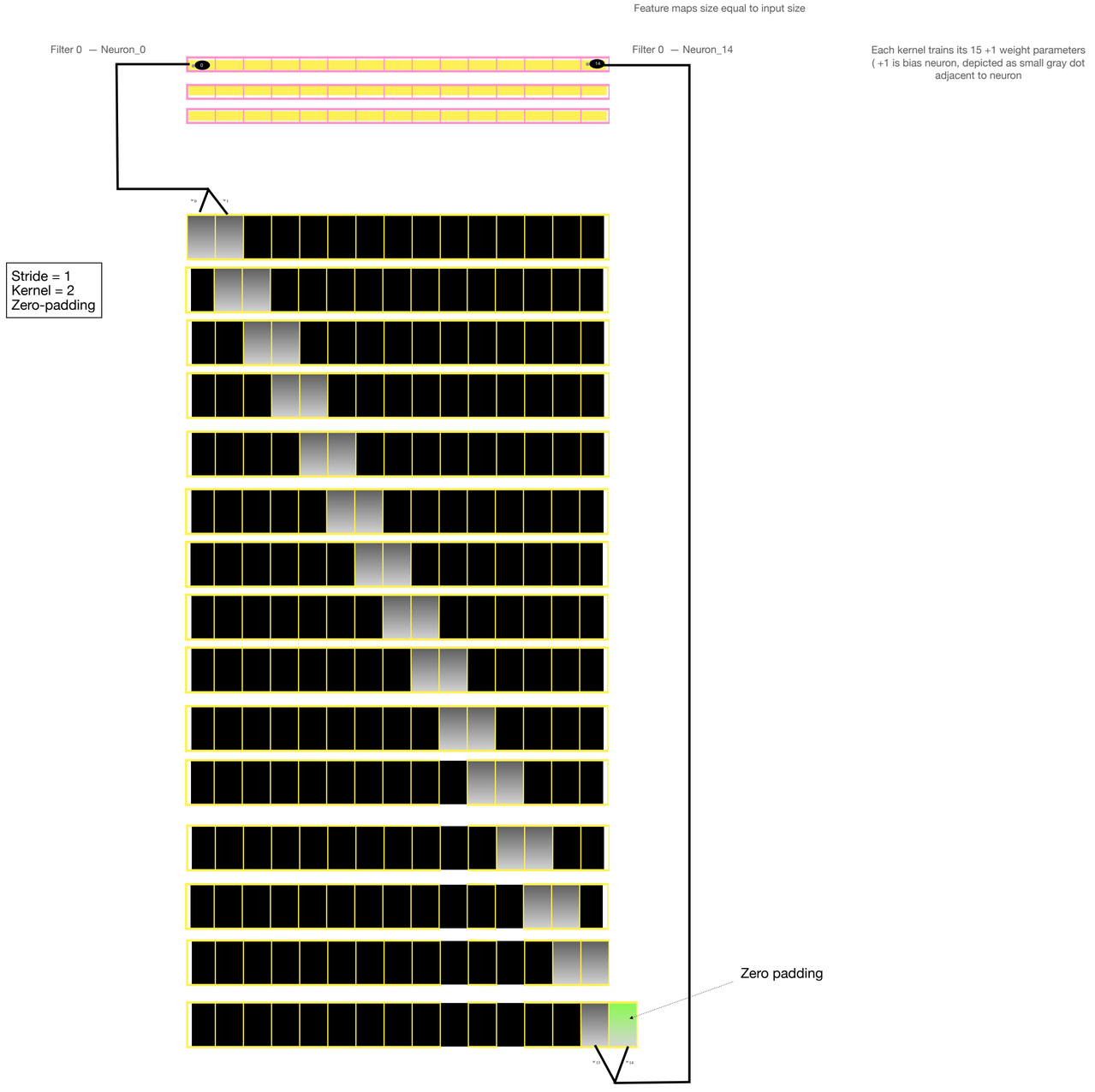


Model learns long patterns from each series sample

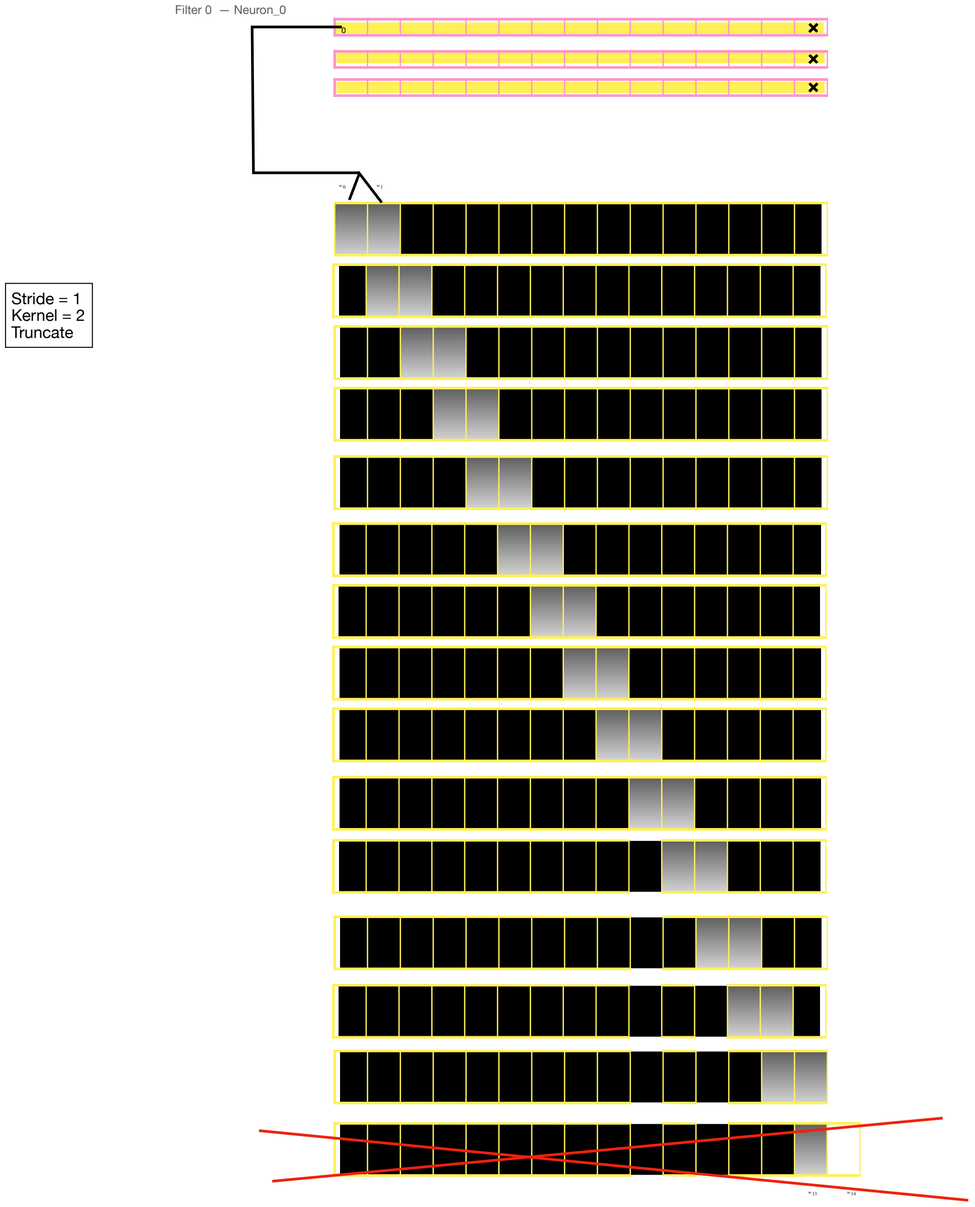
1D Convolution



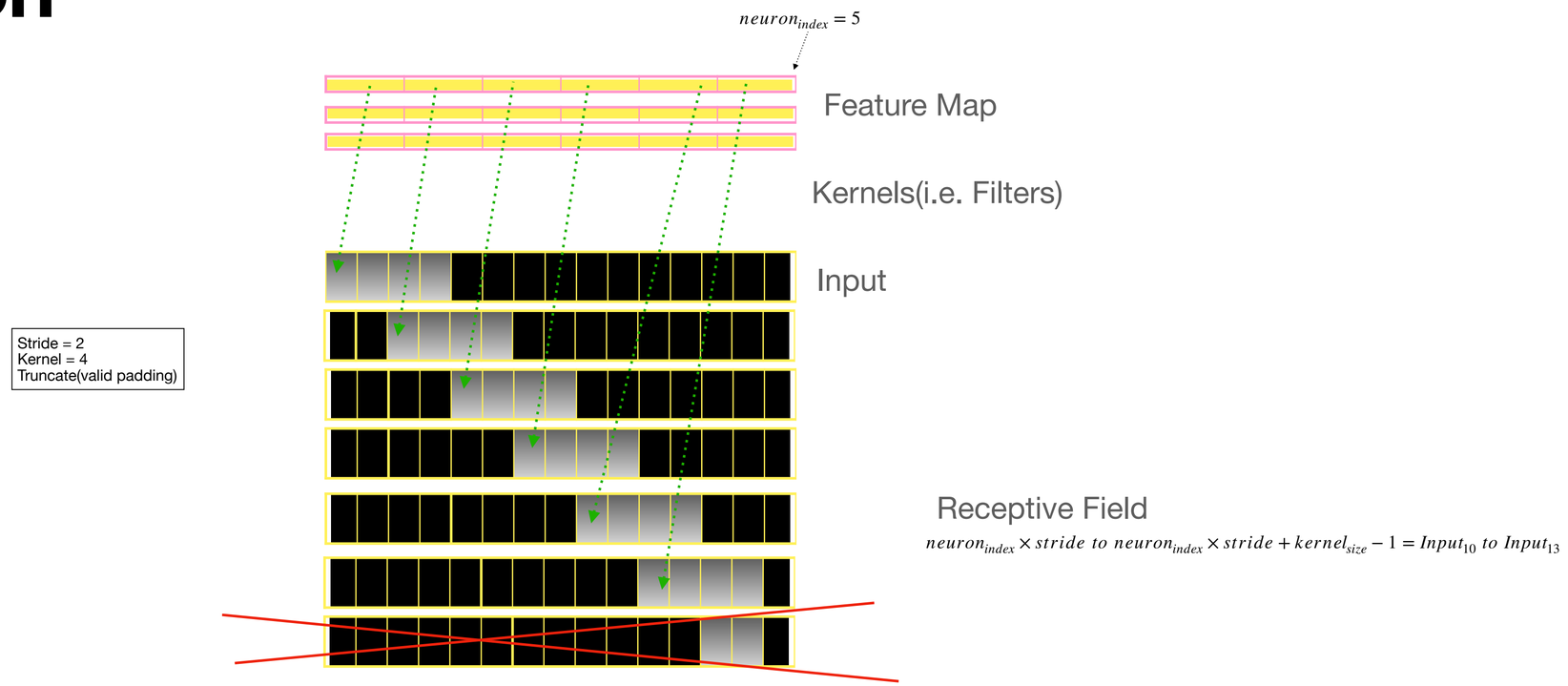
1D Convolution



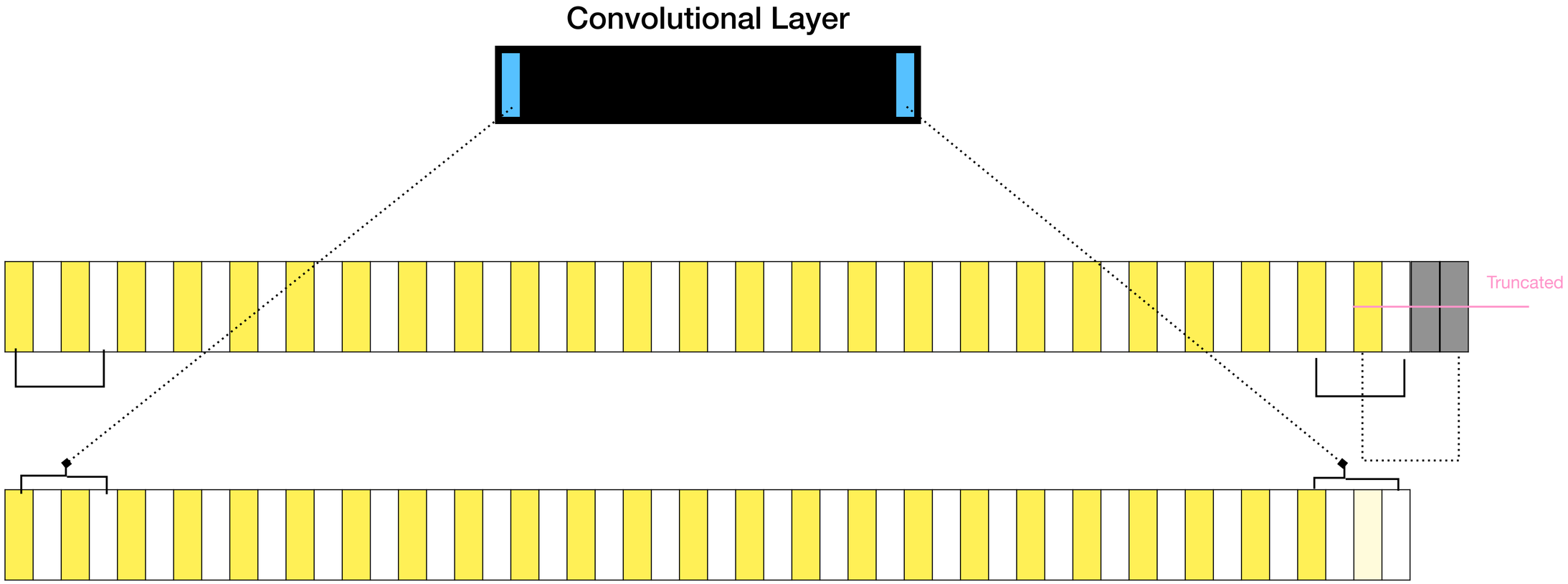
1D Convolution



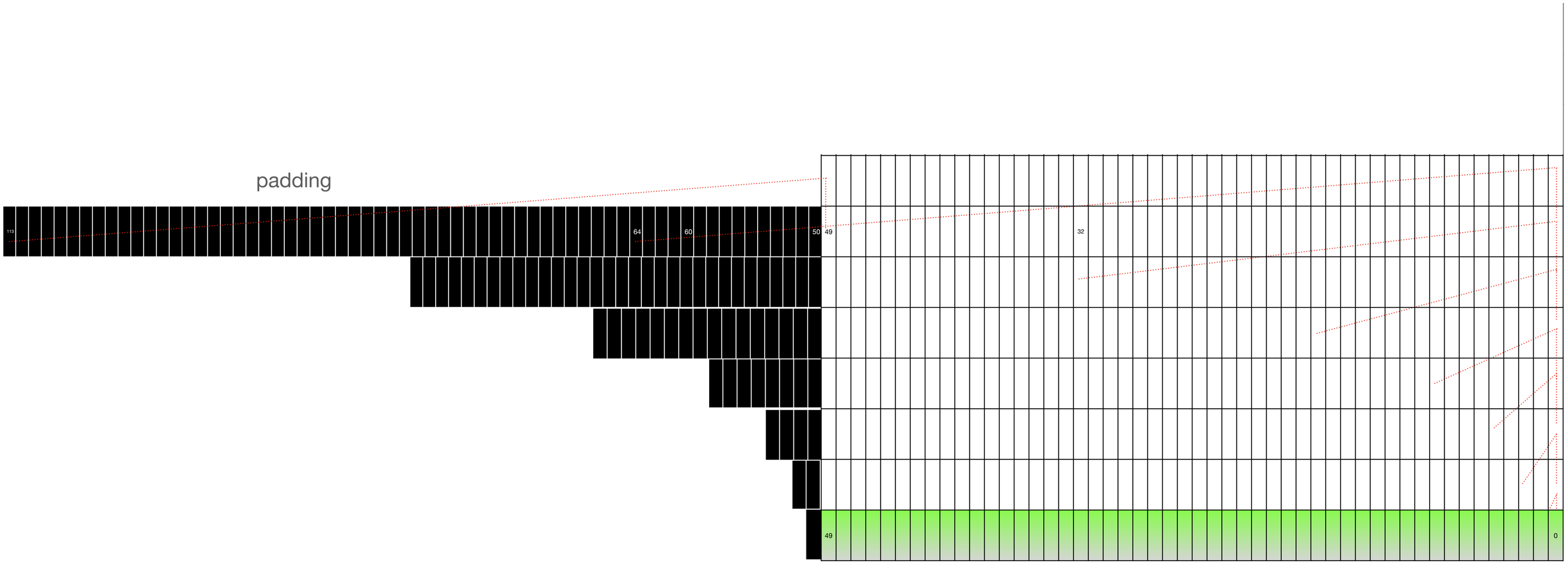
1D Convolution



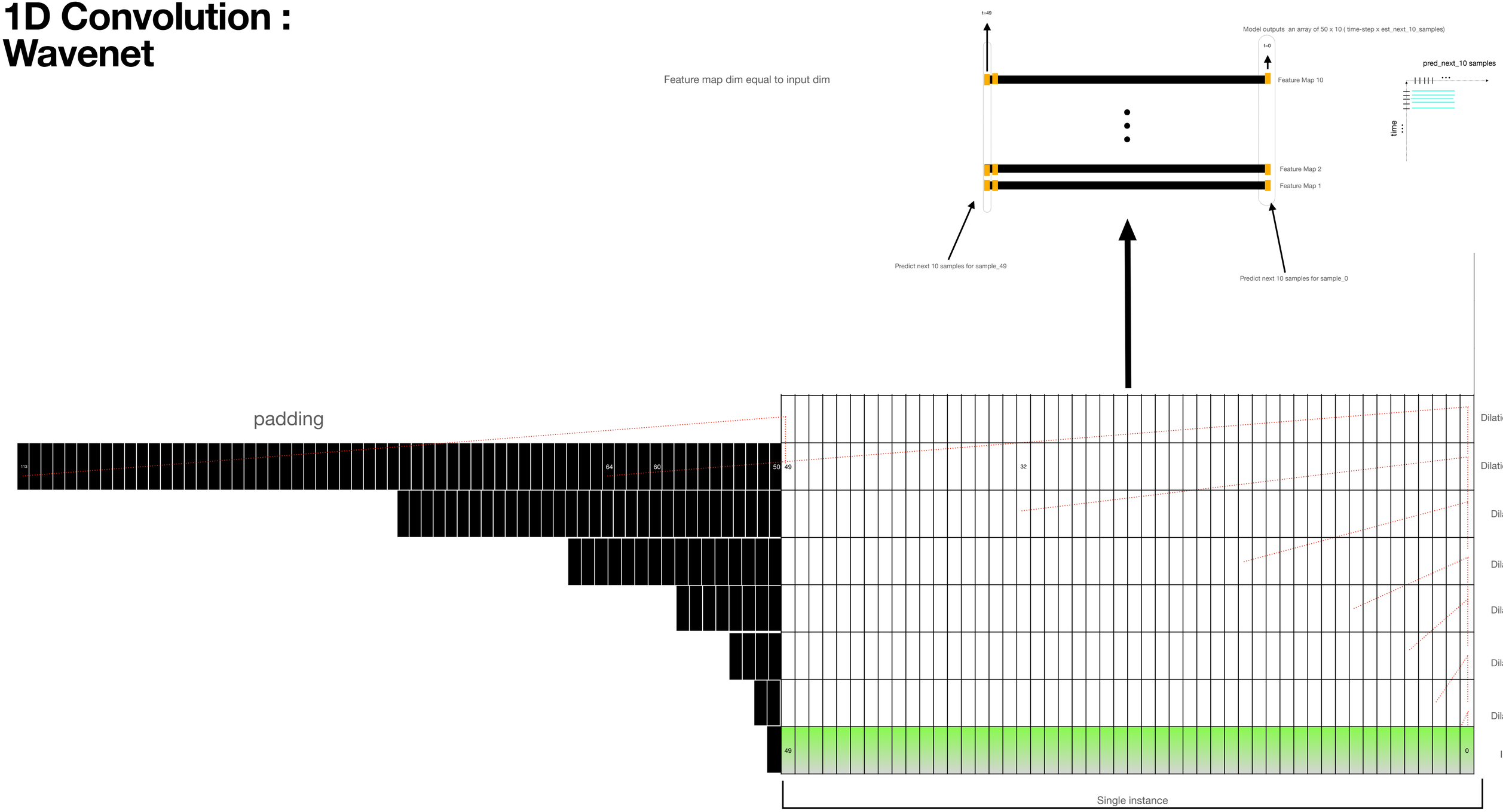
1D Convolution



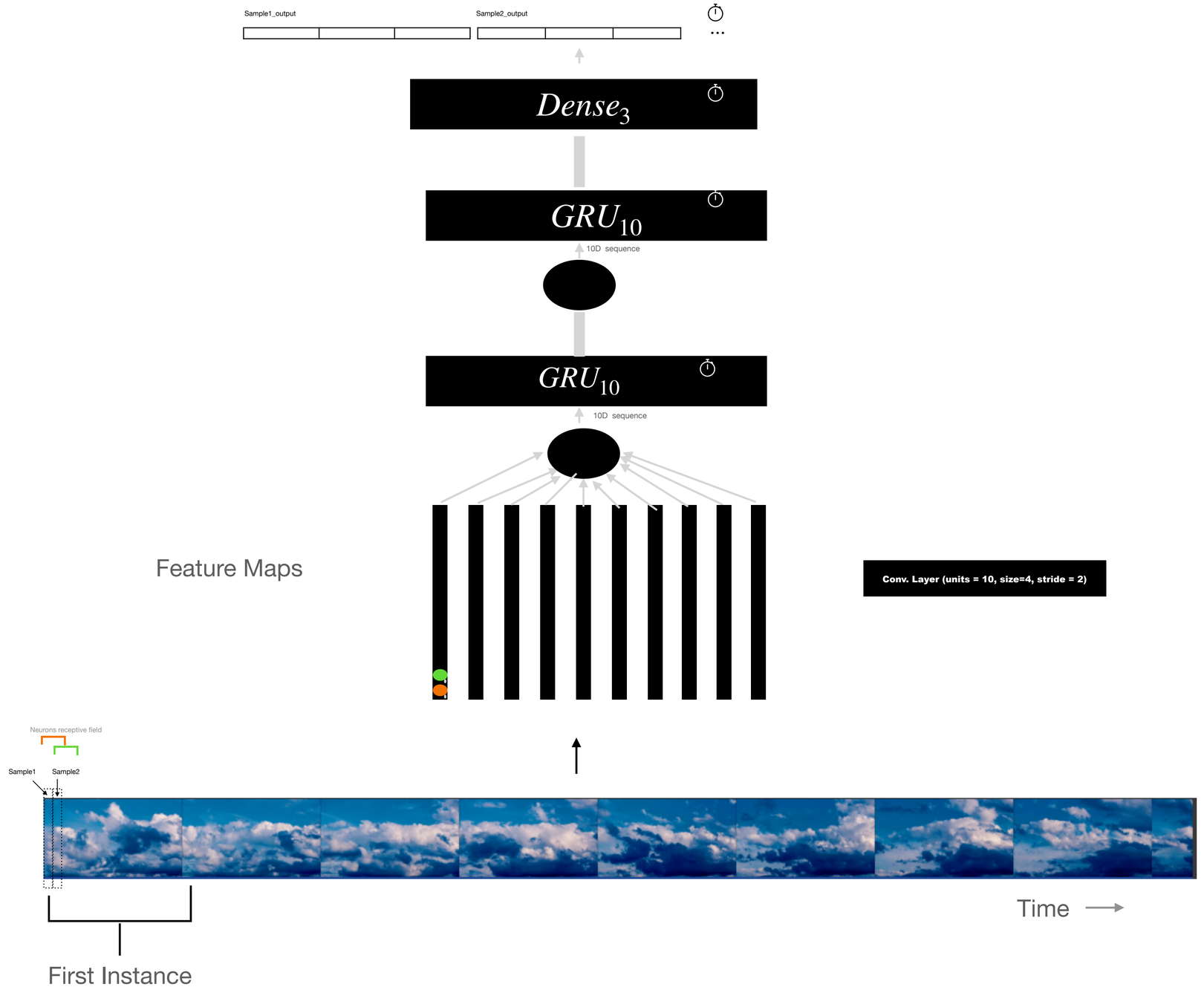
1D Convolution : Wavenet



1D Convolution : Wavenet



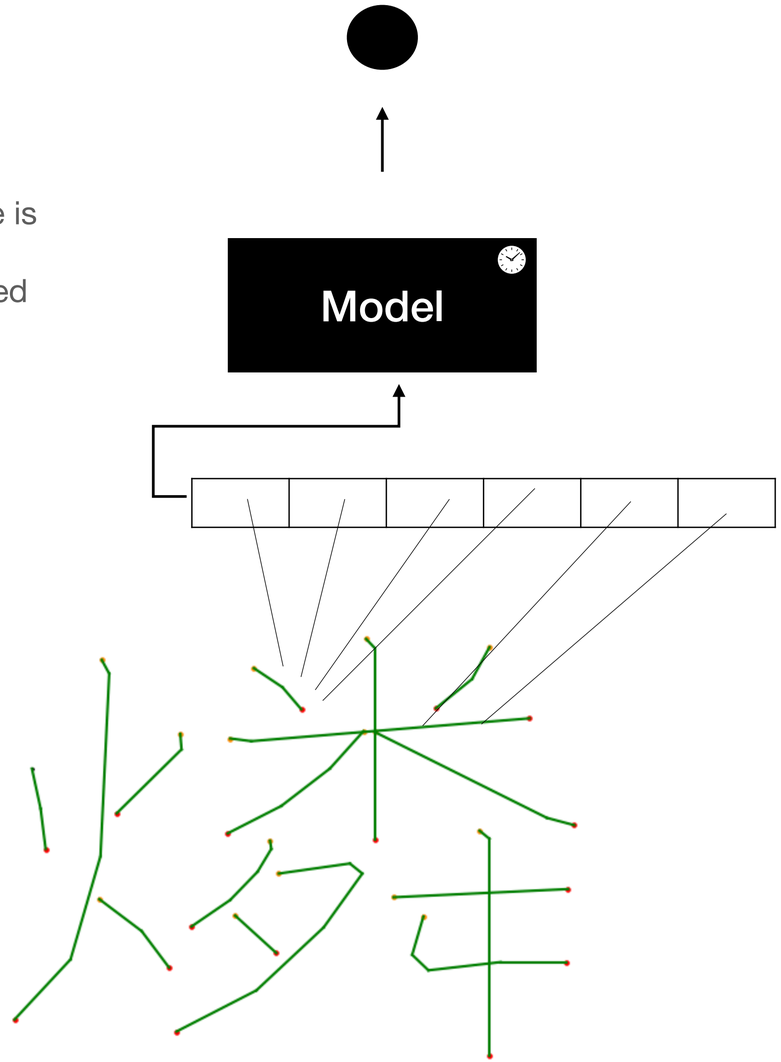
Classify Video



Classification Strokes

Drawing-0 , Not Drawing -1

Max instance sequence length(i.e. sequence is a list of vectors)vary between 100-200 samples. 1D convolutional layers will be used to help learn longer patterns.



Classification Strokes