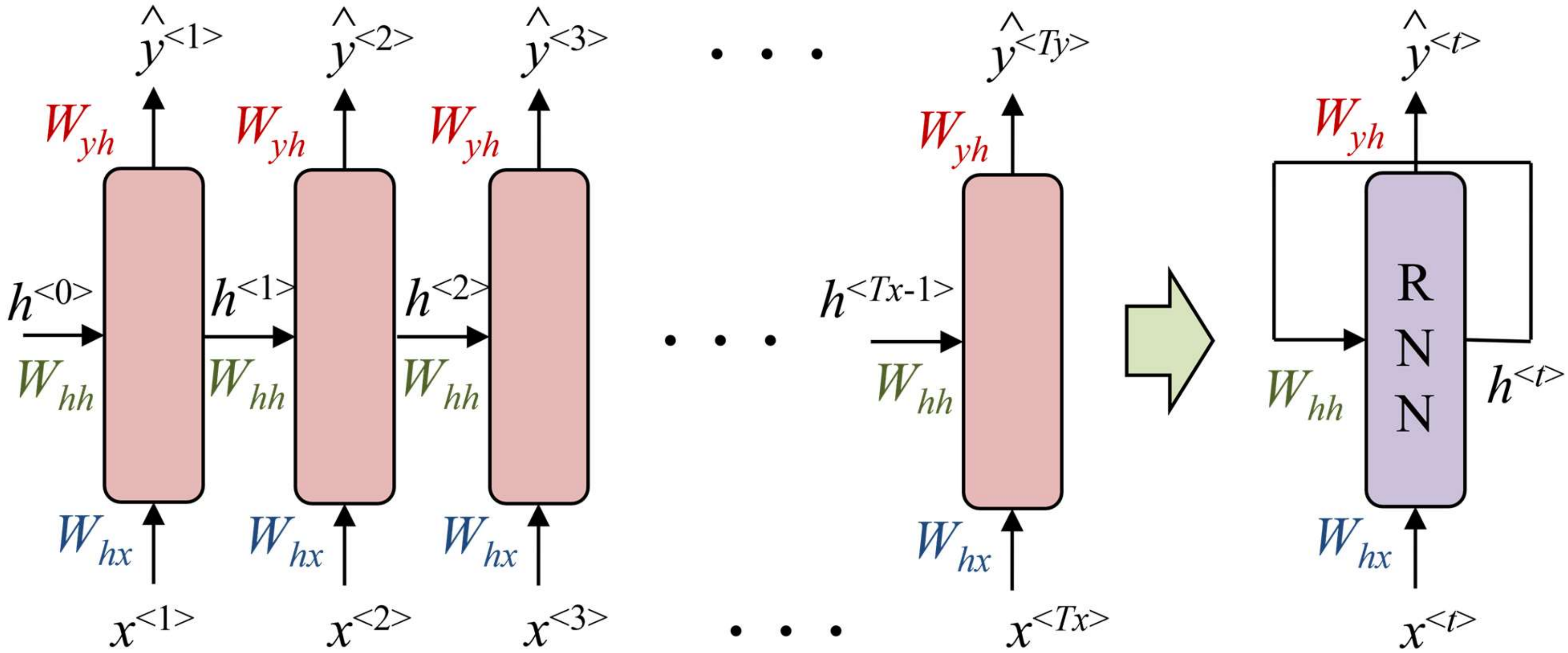
SEQUENCE MODELS

Dr.Varodom Toochinda

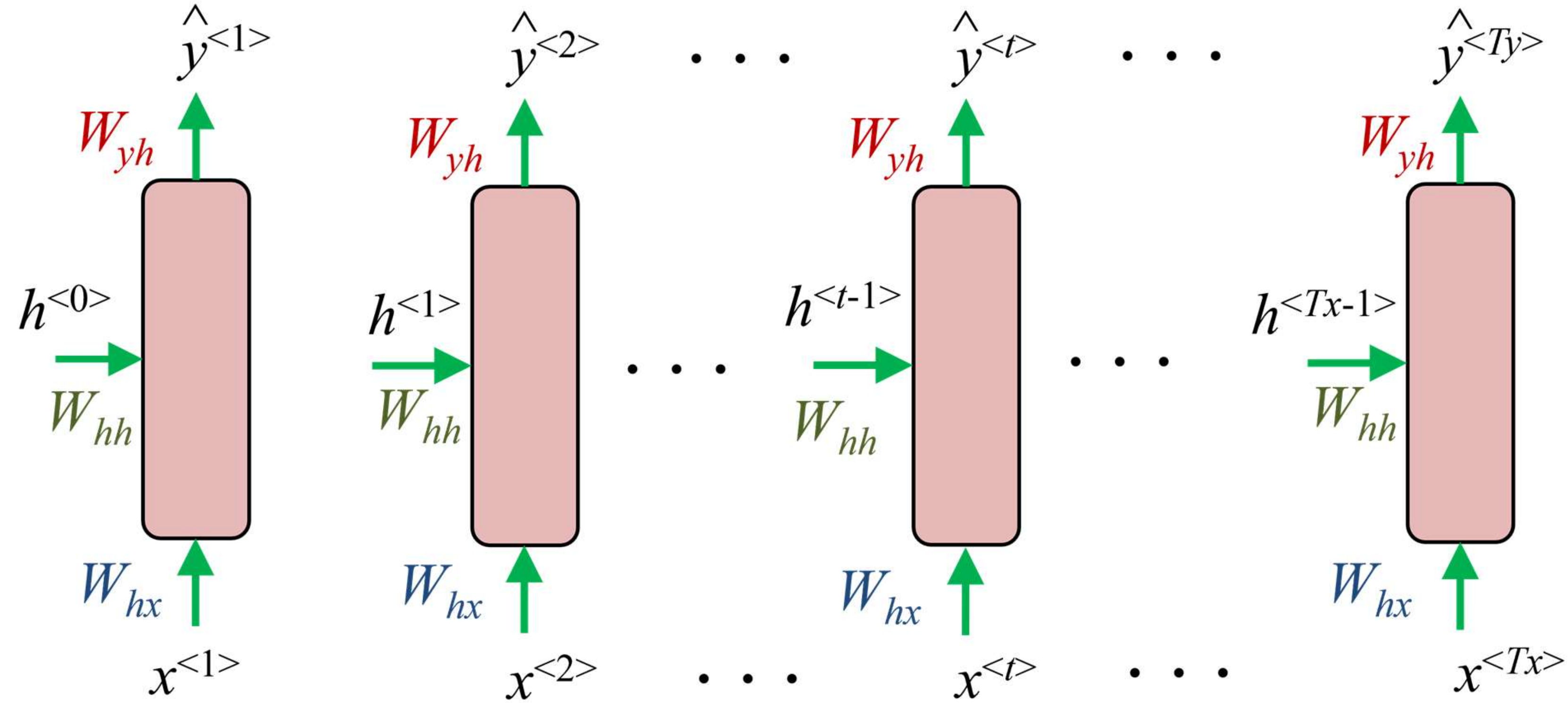
Dept. of Mechanical Engineering

Kasetsart University



RNN (RECURRENT NEURAL NETWORKS)

RNN FORWARD PROPAGATION



$$h^{<t>} = f(W_{hh}h^{<t-1>} + W_{hx}x^{<t>} + b_h)$$

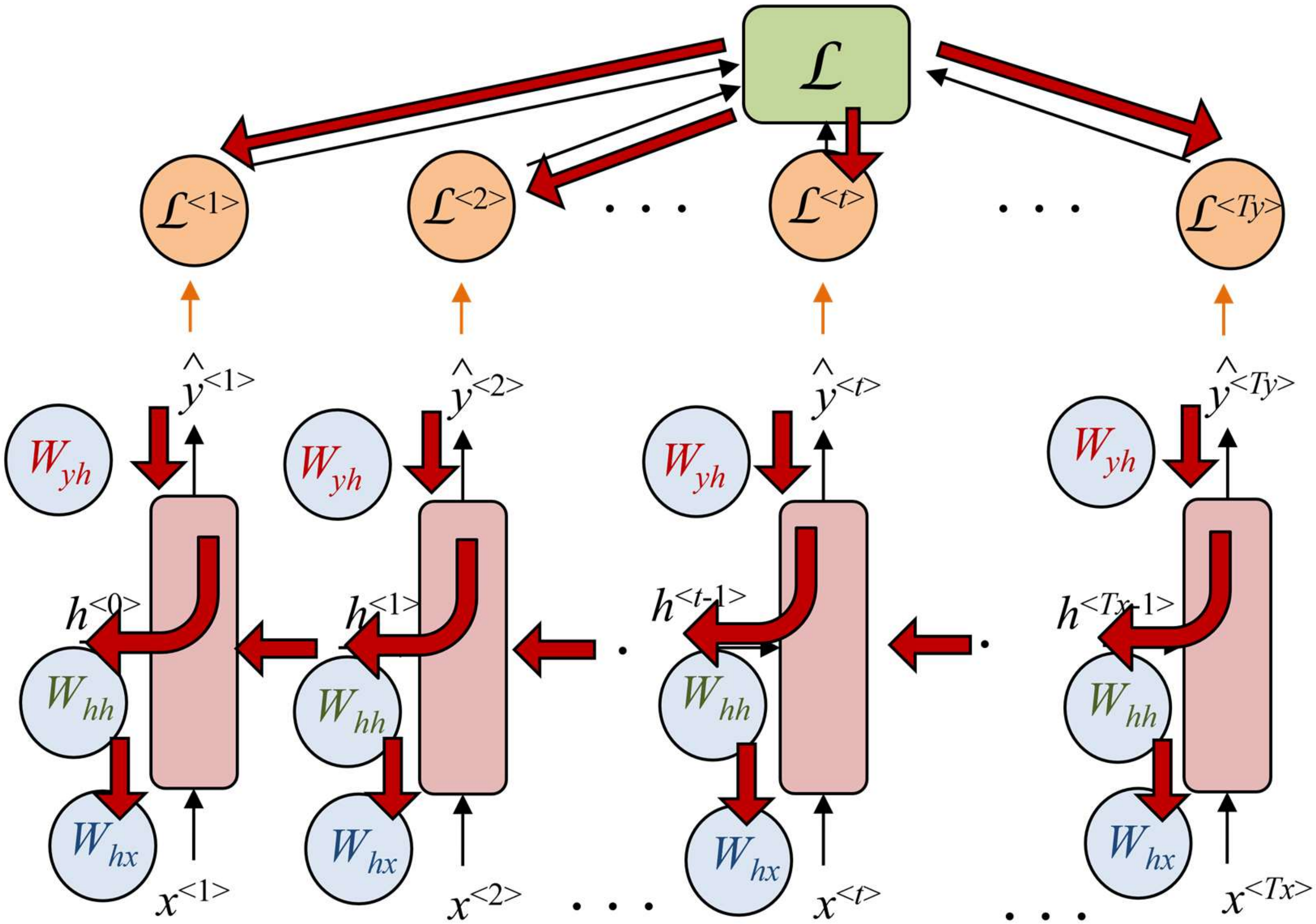
$$\hat{y}^{<t>} = g(W_{yh}h^{<t>} + b_y)$$

$$W_h = \begin{bmatrix} W_{hh} & W_{hx} \end{bmatrix}$$

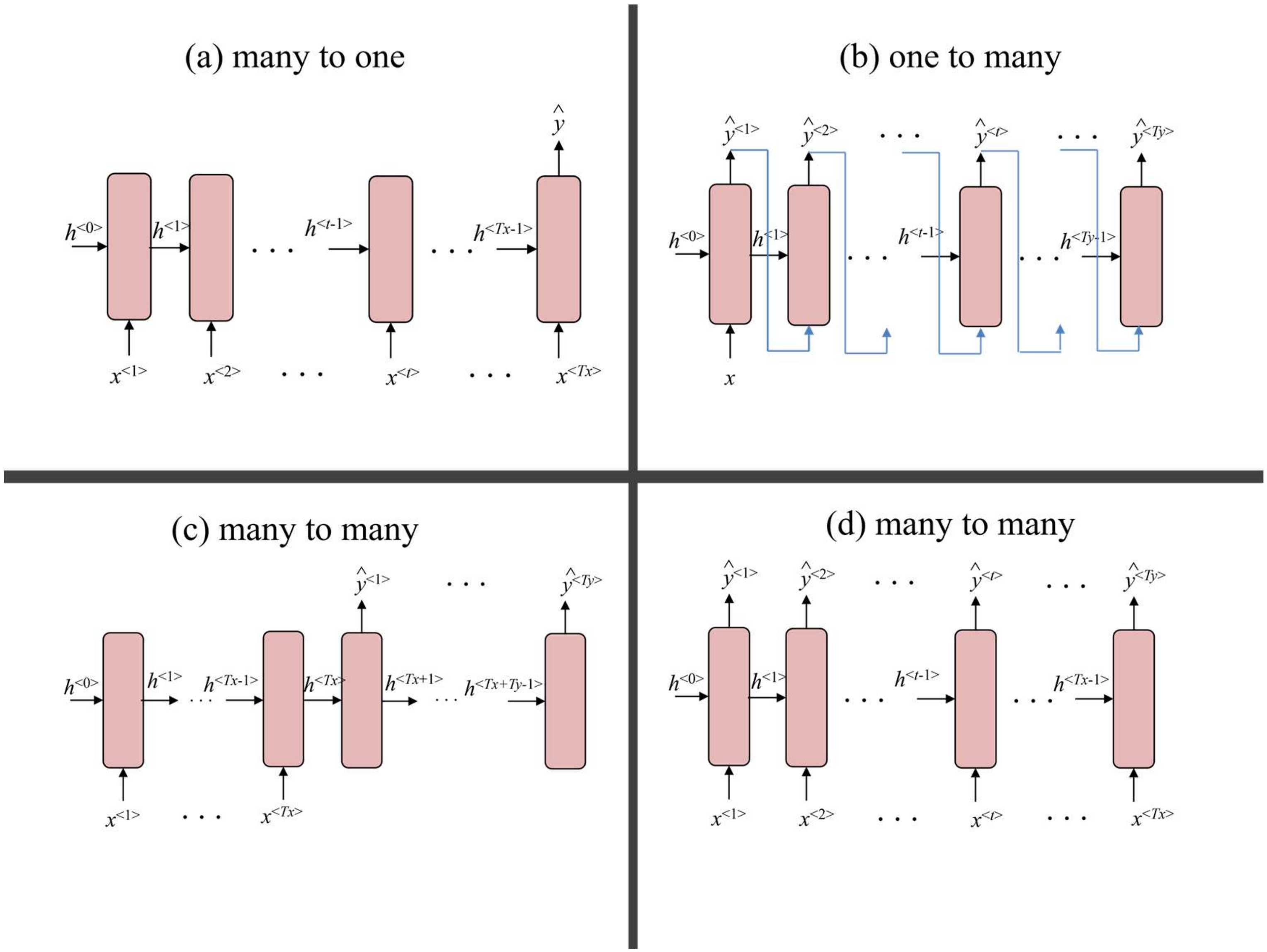
$$[h^{<t-1>}, x^{<t>}] = \begin{bmatrix} h^{<t-1>} \\ x^{<t>} \end{bmatrix}$$

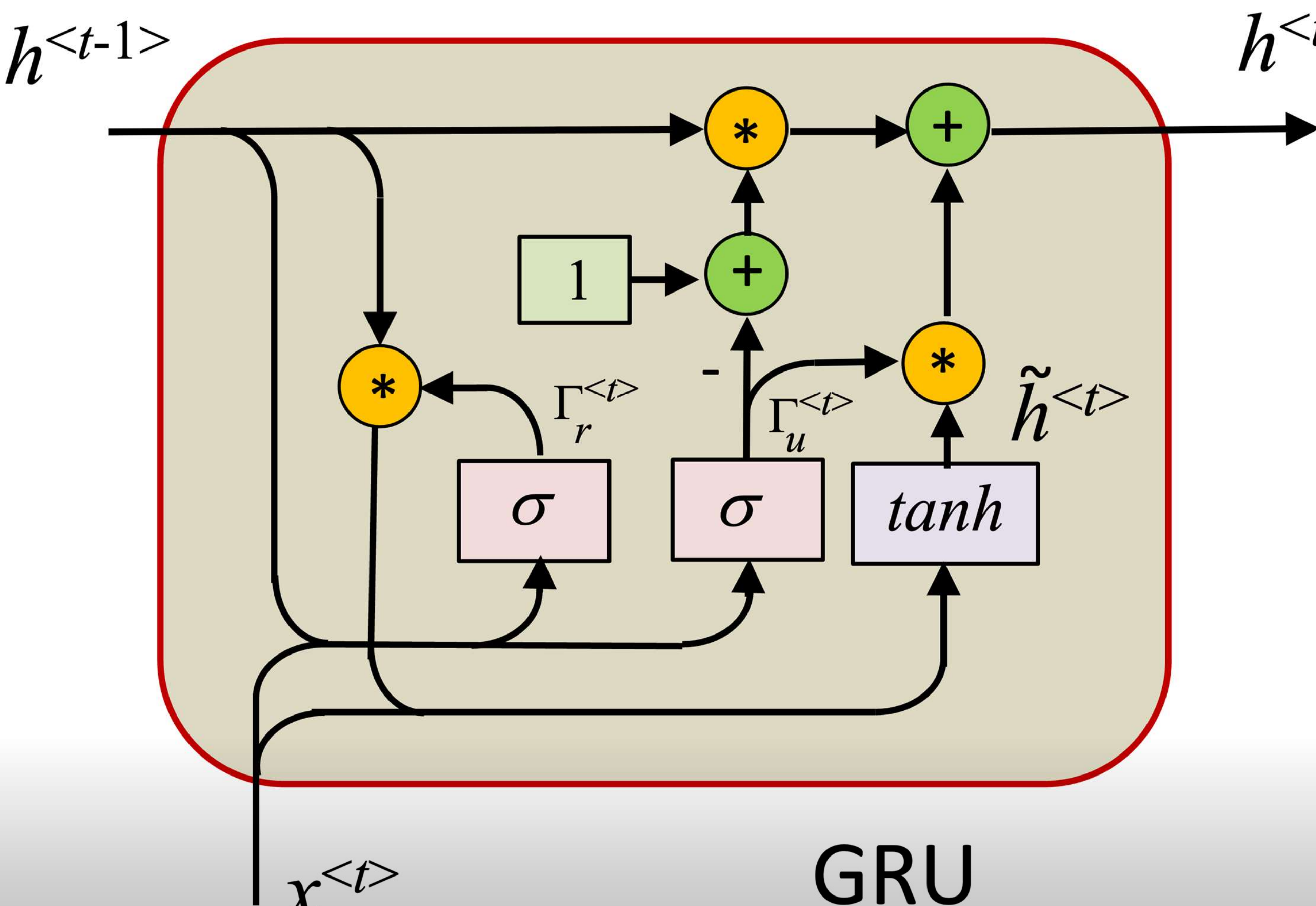
$$h^{<t>} = f(W_h[h^{<t-1>}, x^{<t>}] + b_h)$$

RNN BACKWARD PROPAGATION



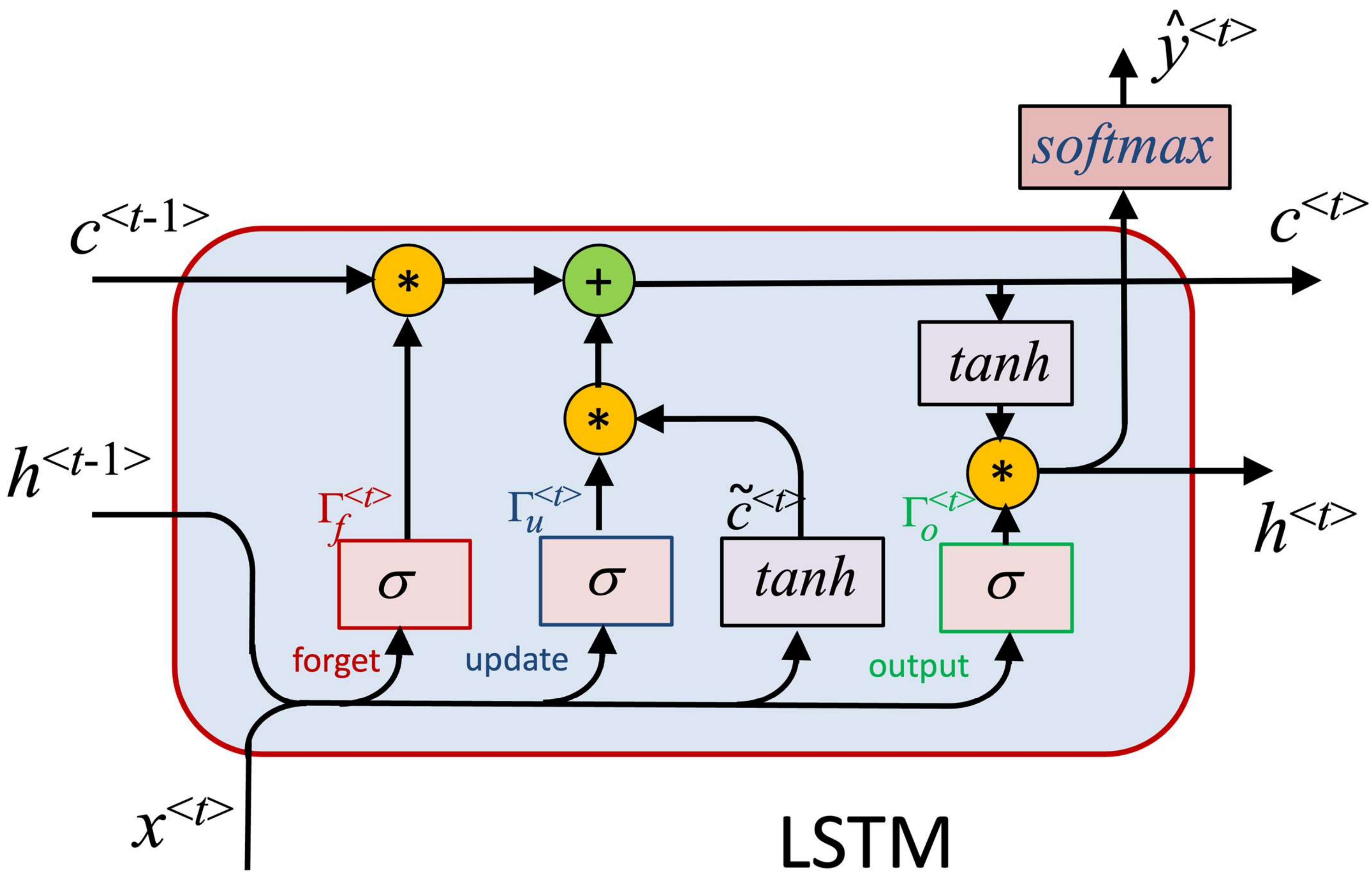
RNN ARCHITECTURES

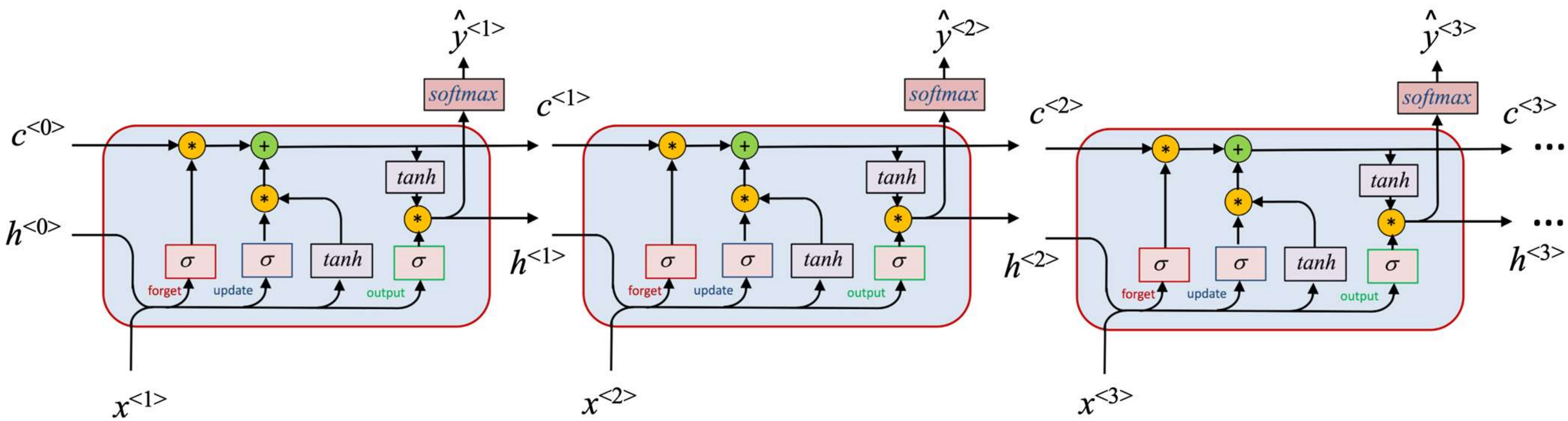




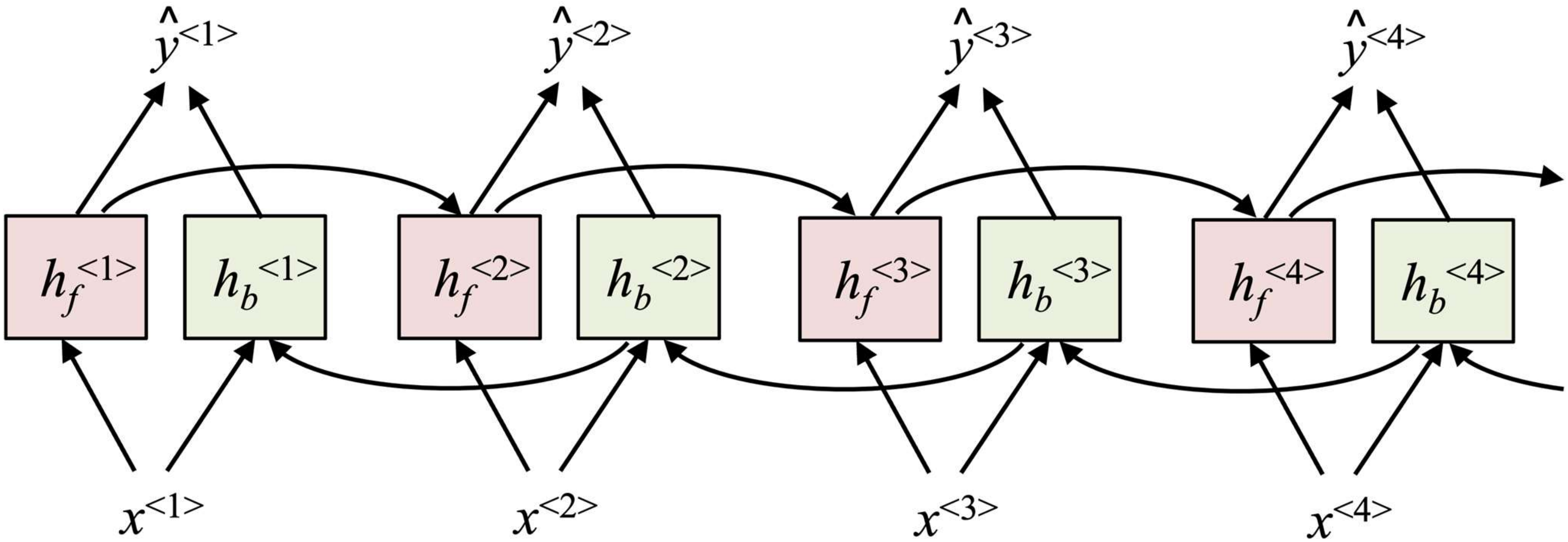
GRU (GATED RECURRENT UNIT)

LSTM (LONG SHORT TERM MEMORY)



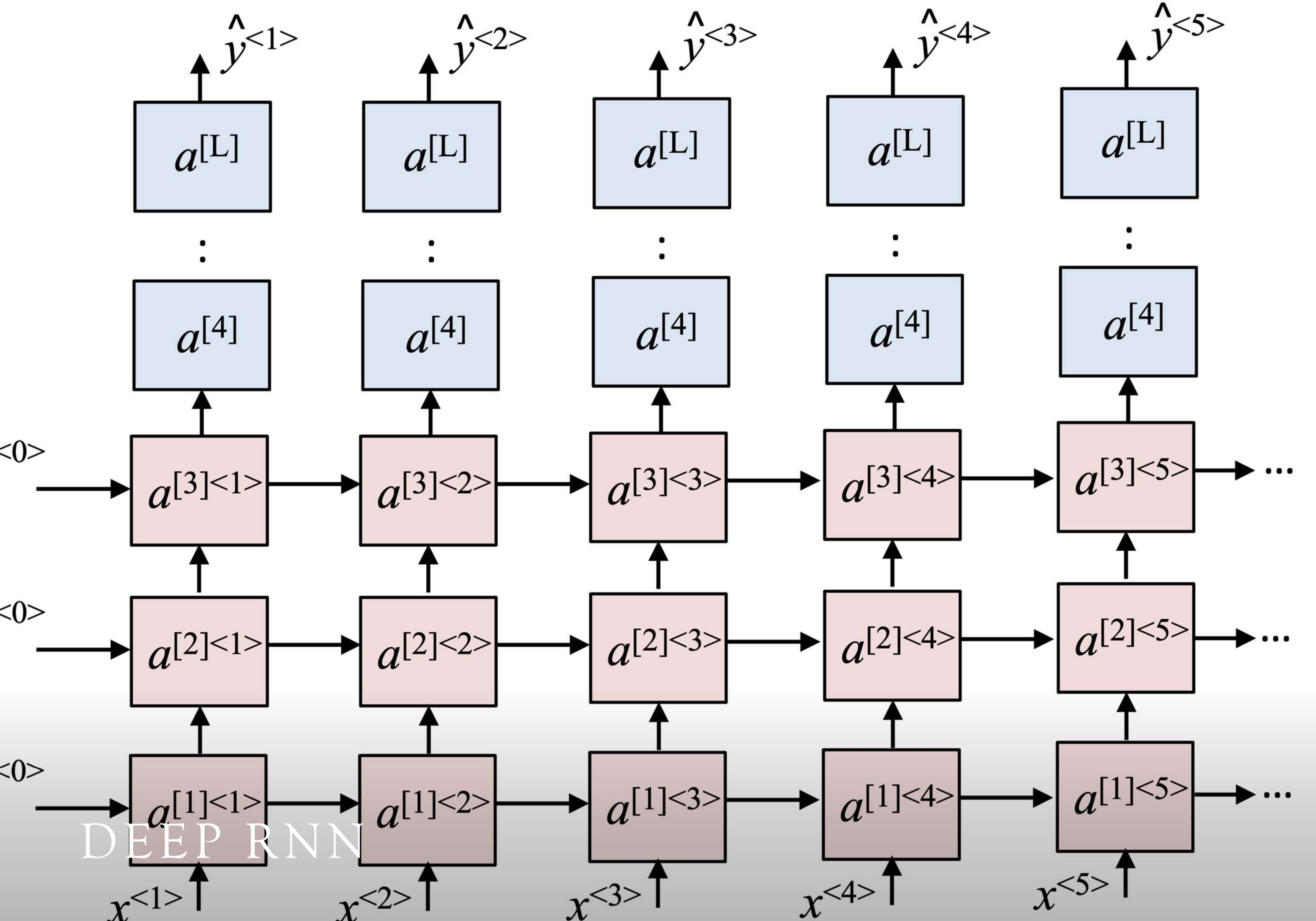


LSTM UNROLLED



BIDIRECTIONAL RNN

DEEP RNN



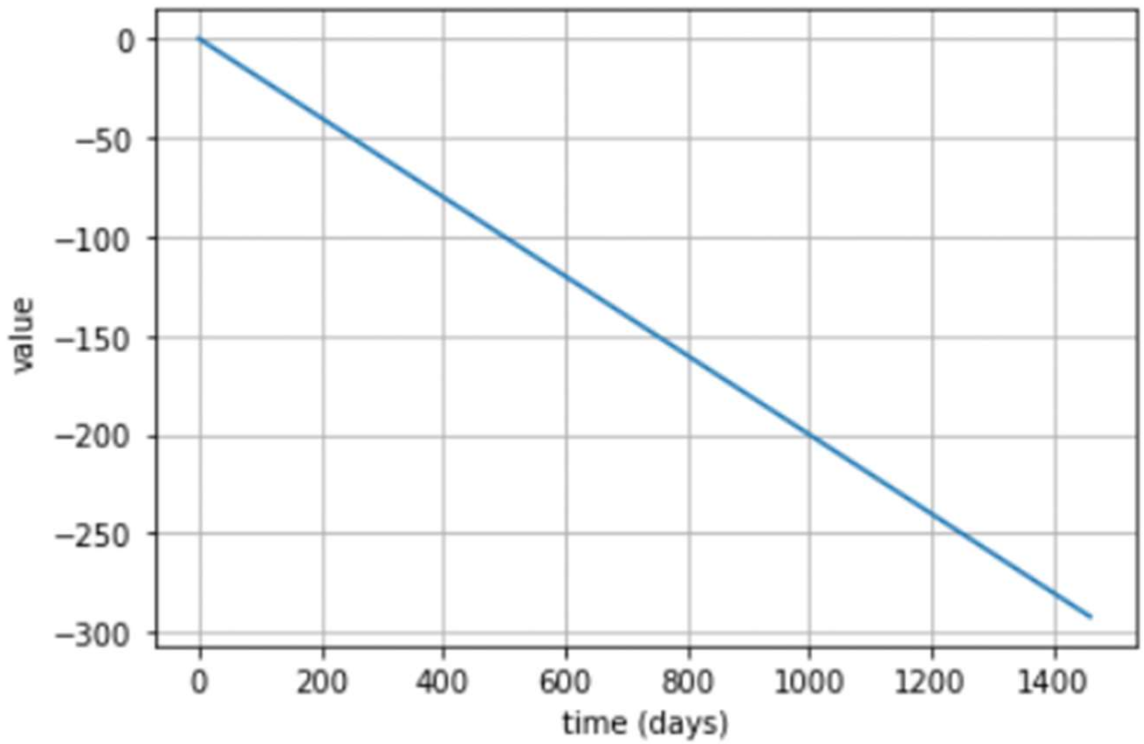
$\hat{y}^{<1>}$ $\hat{y}^{<2>}$ $\hat{y}^{<3>}$ $\hat{y}^{<4>}$ $\hat{y}^{<5>}$
$a^{[L]}$
$a^{[4]}$
$a^{[3]<1>}$ $a^{[3]<2>}$ $a^{[3]<3>}$ $a^{[3]<4>}$ $a^{[3]<5>}$
$a^{[2]<1>}$ $a^{[2]<2>}$ $a^{[2]<3>}$ $a^{[2]<4>}$ $a^{[2]<5>}$
$a^{[1]<1>}$ $a^{[1]<2>}$ $a^{[1]<3>}$ $a^{[1]<4>}$ $a^{[1]<5>}$
$x^{<1>}$ $x^{<2>}$ $x^{<3>}$ $x^{<4>}$ $x^{<5>}$

TIME SERIES PREDICTION

TIME SERIES COMPONENTS

- trend
- seasonal
- noise
- autocorrelation

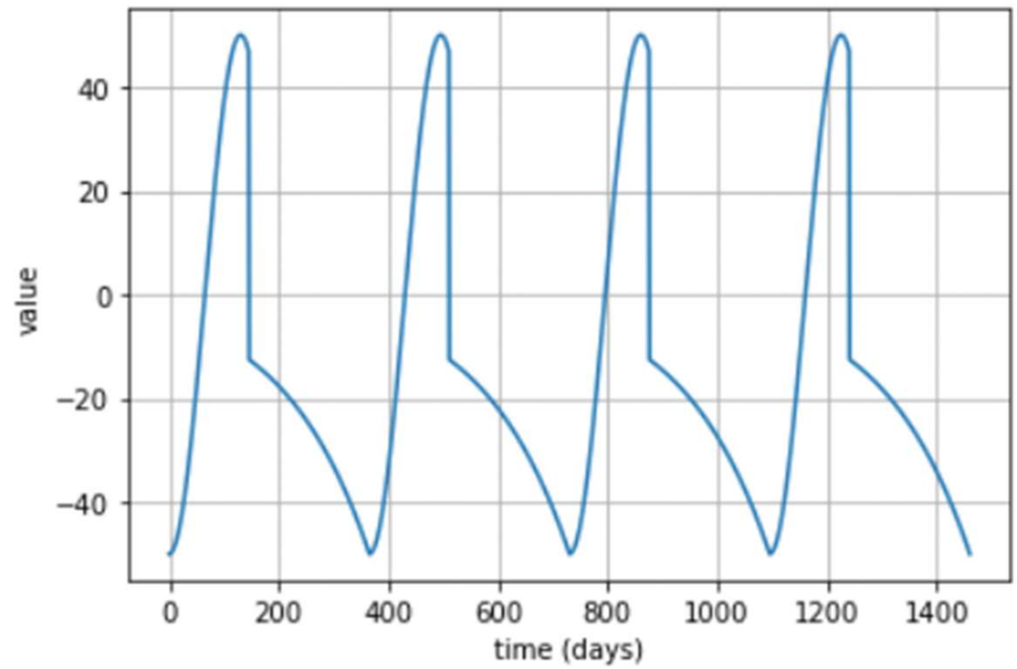
TREND




```
def trend(t, slope=0):
    return slope * t
```

```
time = np.arange(4 * 365 + 1)
y_trend = trend(time, -0.2)
plot_series(time, y_trend)
```

SEASONAL

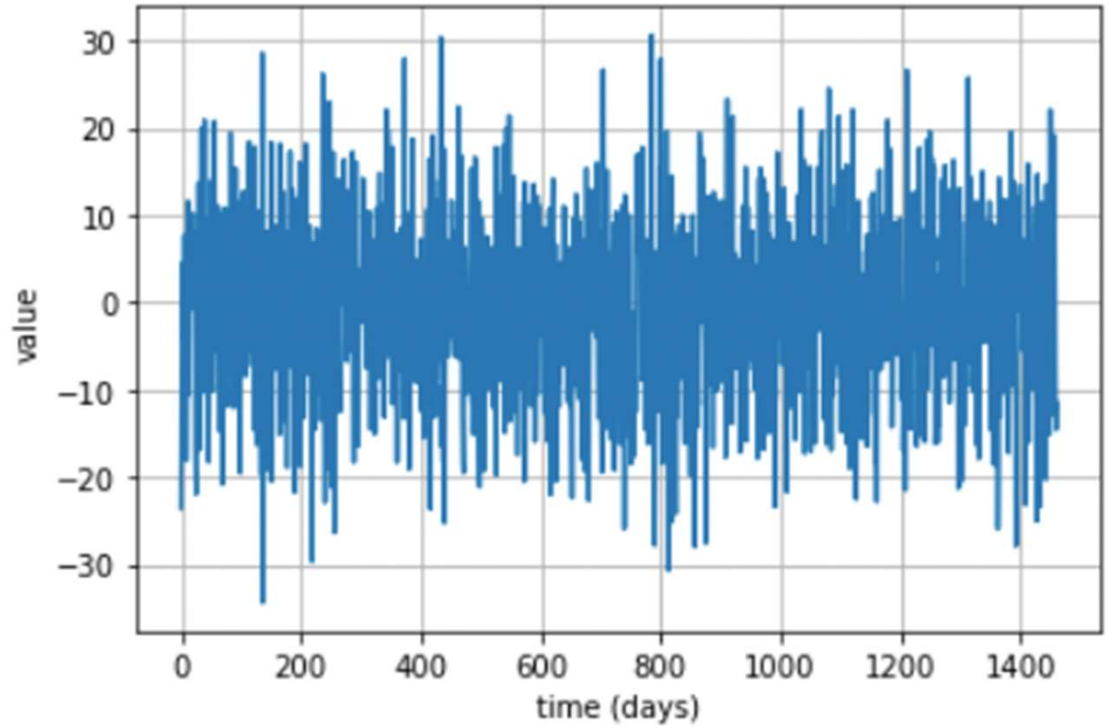



```
def seasonal_pattern(season_time):
    return np.where(season_time < 0.4,
                    -np.cos(season_time * 2.8 * np.pi),
                    -0.1*np.exp(2.3*season_time))

def seasonality(time, period, amplitude=1, phase=0):
    season_time = ((time + phase) % period) / period
    return amplitude * seasonal_pattern(season_time)
```

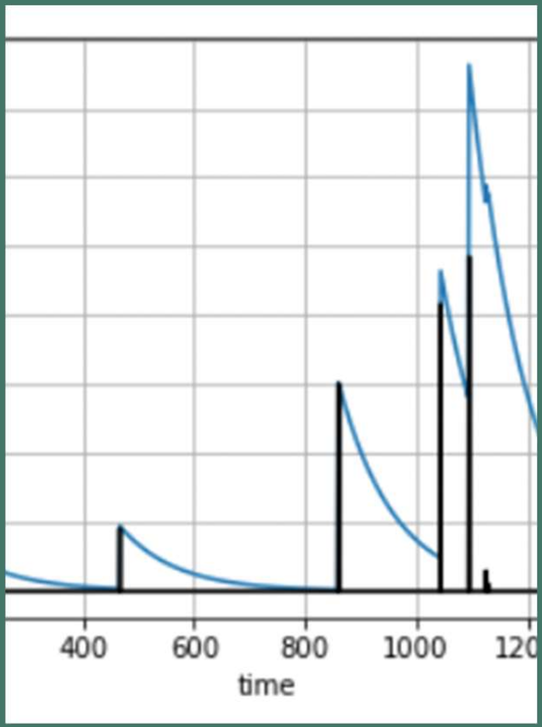
```
amplitude = 50
series = seasonality(time, period=365, amplitude=amplitude)
plot_series(time, series)
```

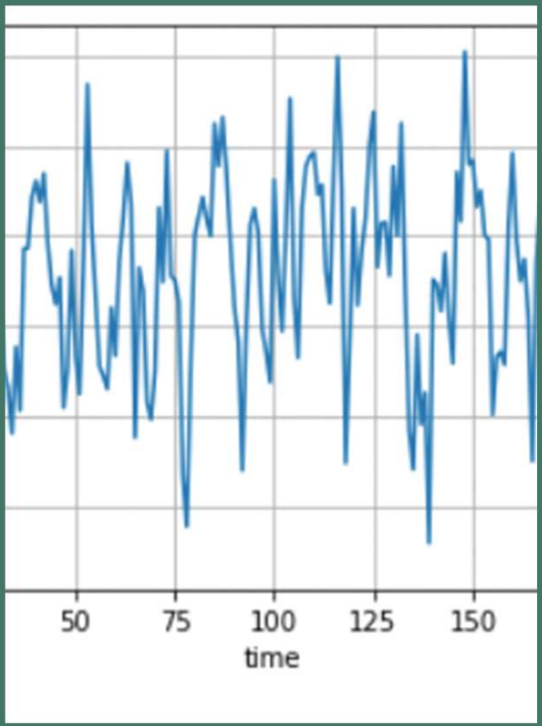
NOISE

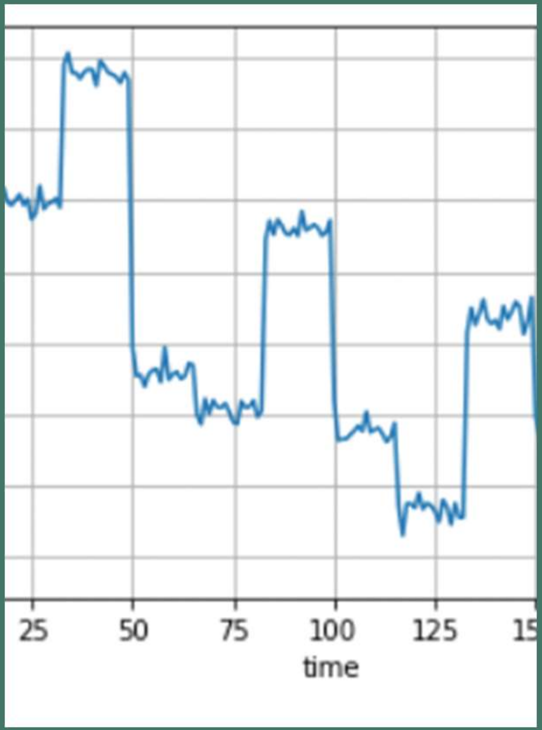



```
def noise(time, noise_level=1):
    return np.random.randn(len(time)) * noise_level
```

```
plot_series(time,noise(time, noise_level=10))
```

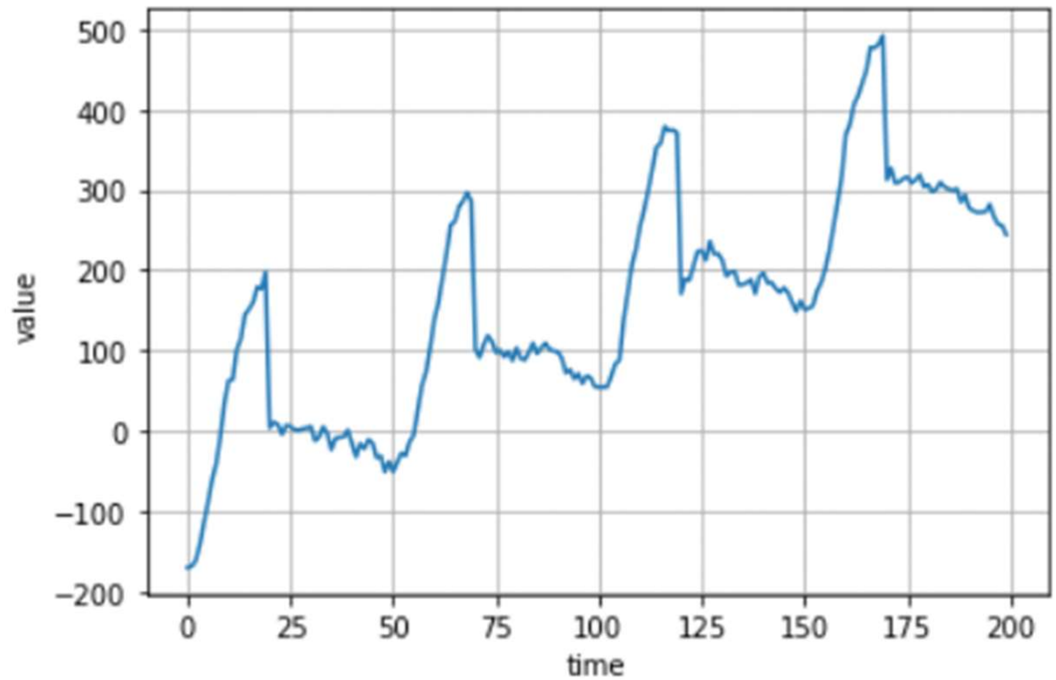






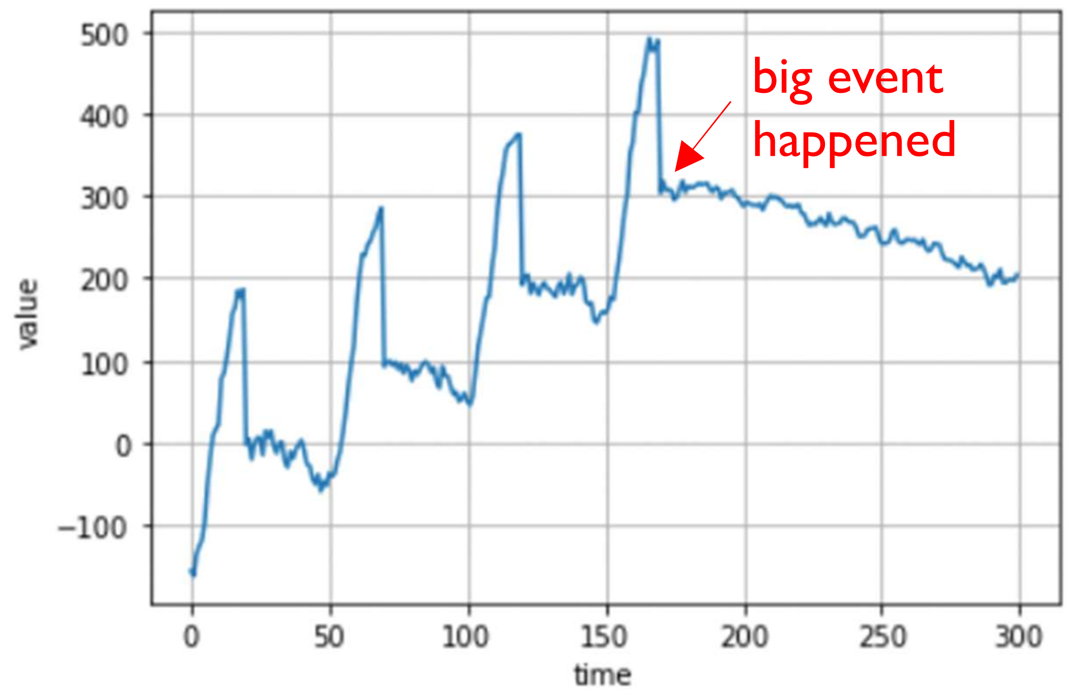
AUTO
CORRELATION
EXAMPLES

SERIES WITH TREND, SEASONAL, AND AUTOCORRELATION



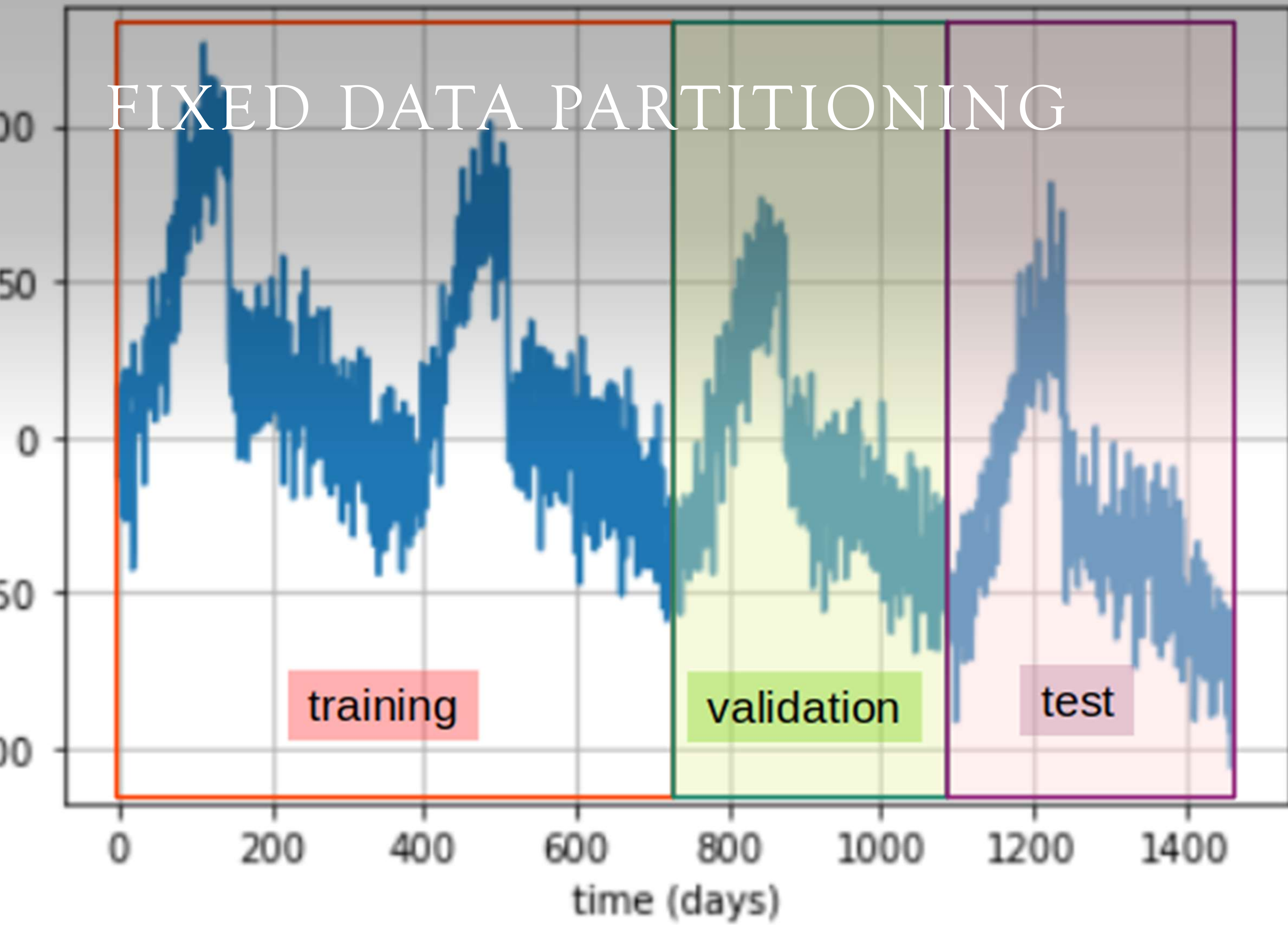

```
series = autocorrelation2(time, 10) + seasonality(time, period=50, amplitude=150) +
trend(time, 2)
plot_series(time[:200], series[:200],xlabel="time")
```

NONSTATIONARY TIME SERIES

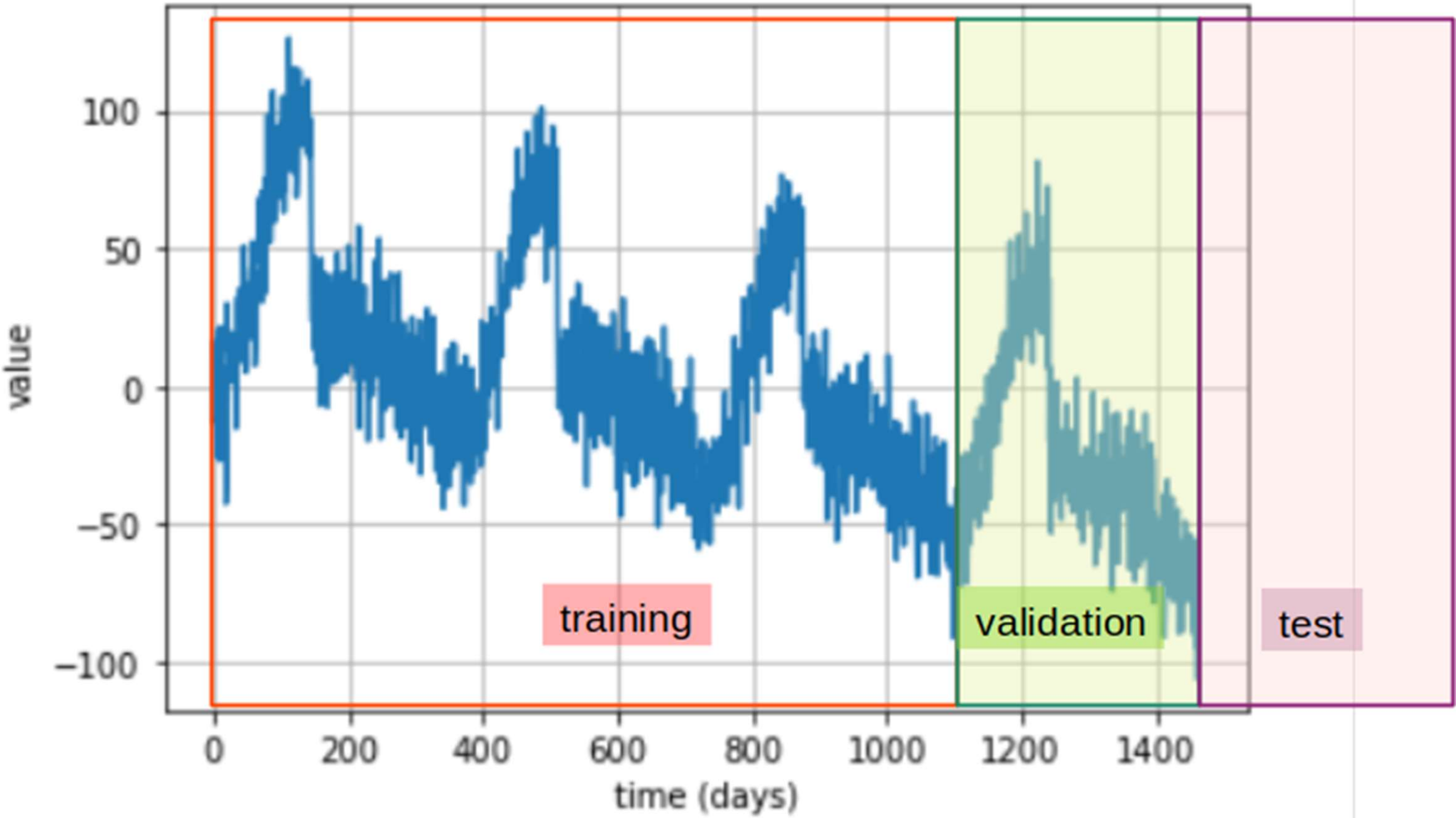



```
series = autocorrelation2(time, 10) + seasonality(time, period=50, amplitude=150) + trend(time, 2)
series2 = autocorrelation2(time, 5) + seasonality(time, period=50, amplitude=2) + trend(time, -1) + 500
series[180:] = series2[180:]
plot_series(time[:300], series[:300],xlabel="time")
```

FIXED DATA PARTITIONING



FIXED DATA PARTITIONING (ROLL FORWARD)

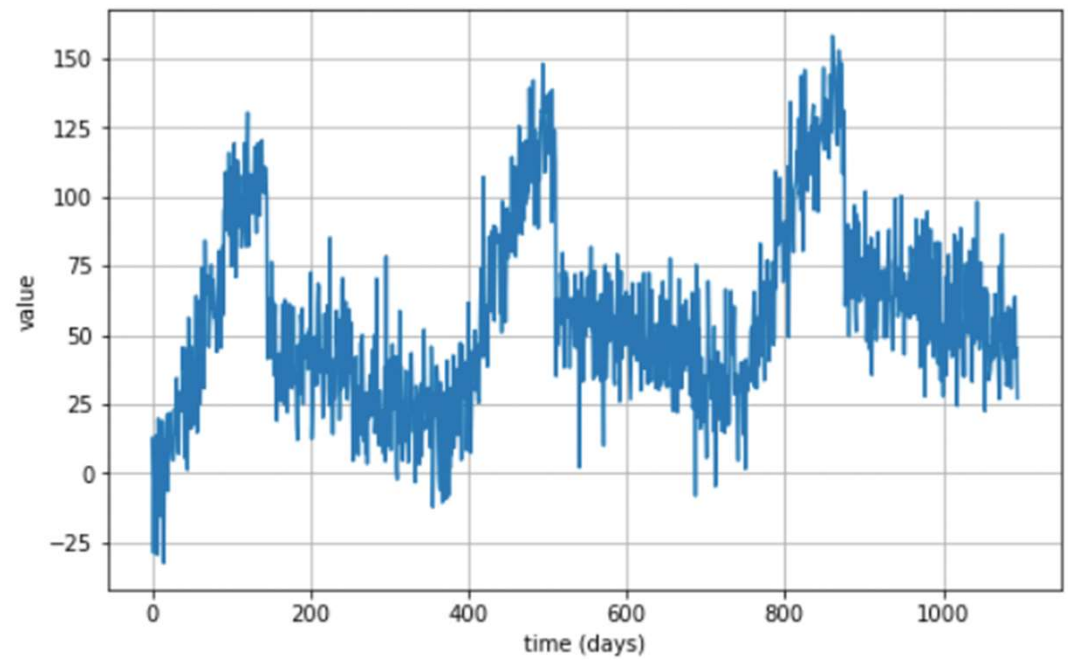


PERFORMANCE METRICS

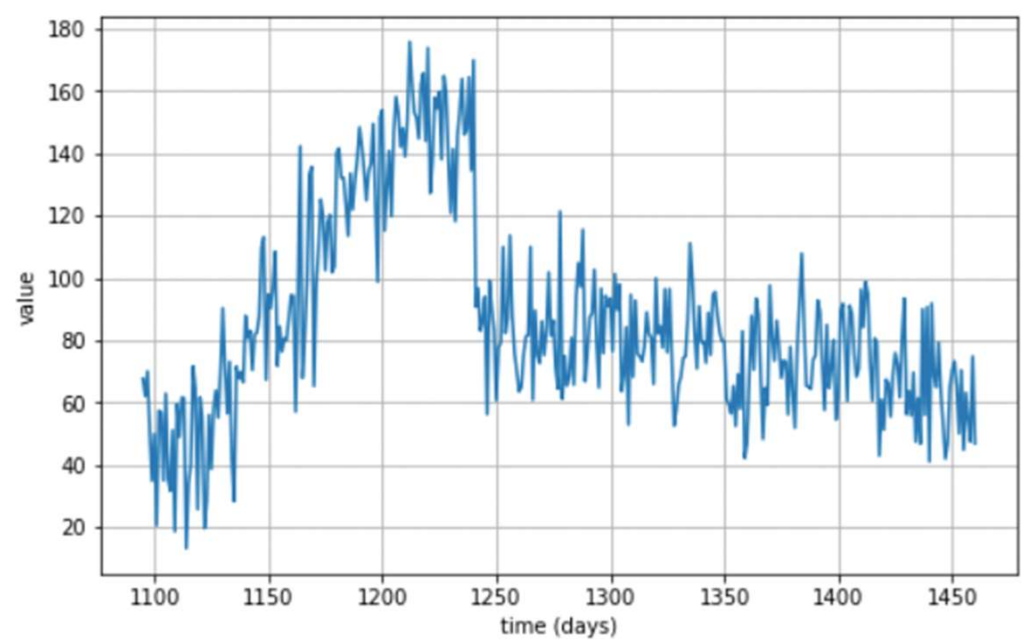
- mse = np.square(errors).mean()
- rmse = np.sqrt(mse)
- mae = np.abs(errors).mean()
- mape = np.abs(errors/x_valid).mean()

```
keras.metrics.mean_squared_error(x_valid, naive_forecast).numpy()
keras.metrics.mean_absolute_error(x_valid, naive_forecast).numpy()
```

CREATE SYNTHETIC DATA

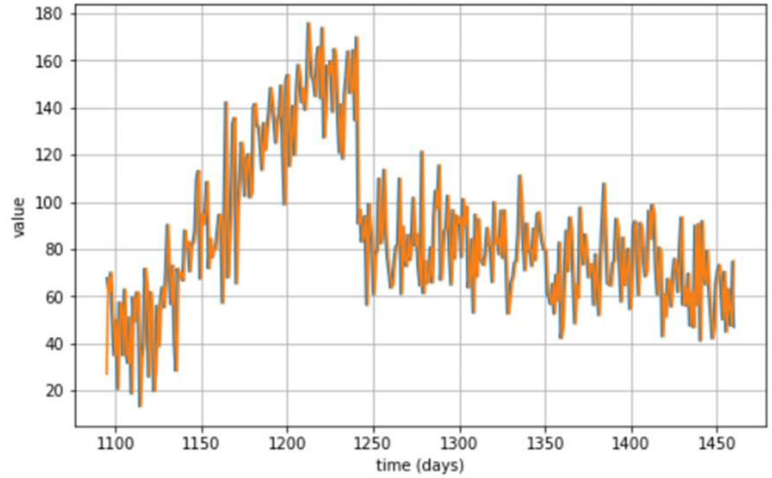


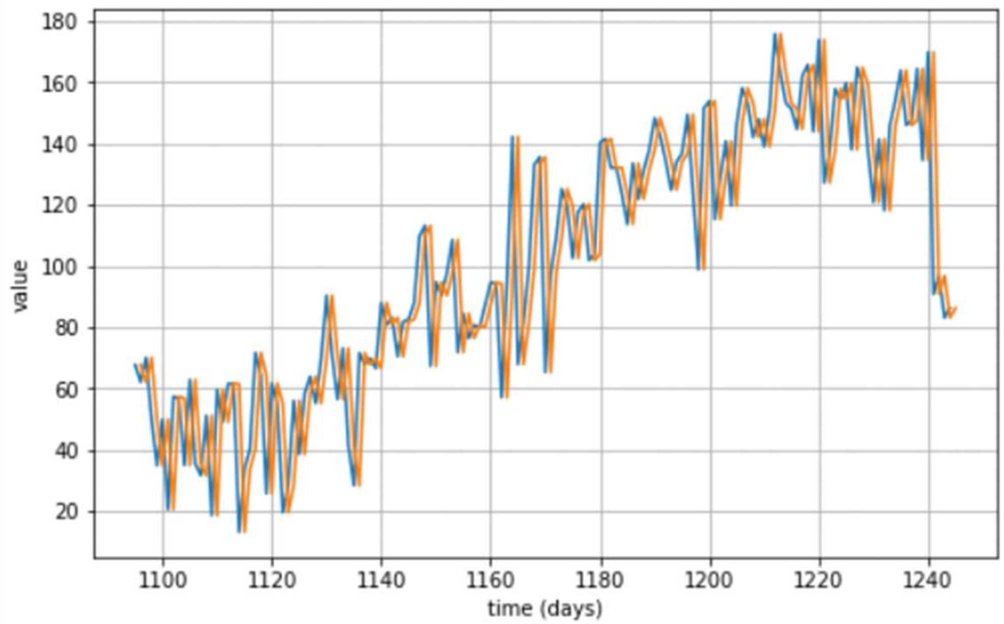
training



testing

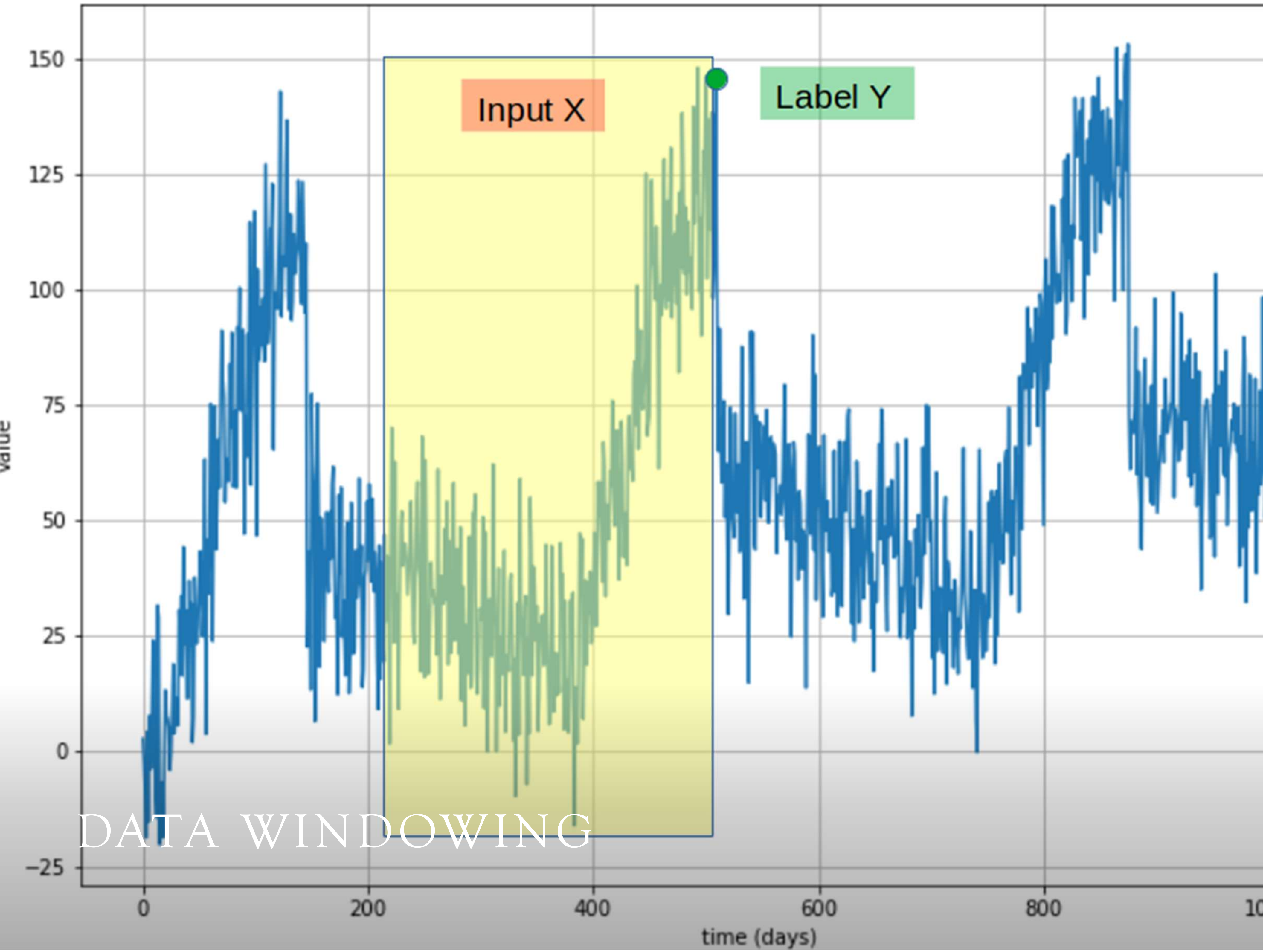
NAÏVE FORCAST (BASELINE)






```
print(keras.metrics.mean_squared_error(x_valid, naive_forecast).numpy())
print(keras.metrics.mean_absolute_error(x_valid, naive_forecast).numpy())

434.3780304104924
16.27624466936418
```



DATA WINDOWING

DEMO AND EXERCISE

- see time_series.ipynb
- Create a model for sunspot prediction