

Zeitreihen Datenvorbereitung

Karin Pröll
DSE/IEM

Zeitreihenvorhersage mit RNN

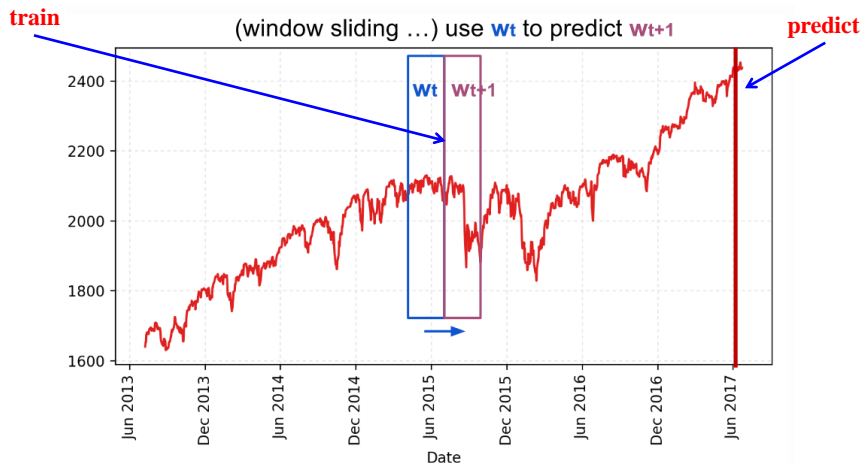
- Eine Zeitreihe ist eine Reihe von Beobachtungen, die gewonnen werden, indem eine Variable regelmäßig über einen Zeitraum beobachtet wird.
- Die Reihenfolge dieser Werte ist wichtig !



- Die Aufgabe der Zeitreihenvorhersage besteht darin, den nächsten Wert einer Zeitreihe basierend auf den zuvor beobachteten Werten vorherzusagen.
- Historische Zeitreihen-Informationen, um den nächsten Wert oder die nächsten Werte in der Zeitreihe vorherzusagen, sind für ein NN notwendig (Input und Output für Training)
- Generiere Input und Output mit Sliding Window Technik über die gegebene Zeitserie

Zeitreihe

Data Preparation – Sliding Window



Univariate Zeitserie - Beispiele

univariate
Zeitserie
gegeben

0
1
2
3
4
5
6
7

Input Vergangenheit timesteps=3				Output Zukunft timesteps=1
	$\text{var1}(t-3)$	$\text{var1}(t-2)$	$\text{var1}(t-1)$	$\text{var1}(t)$
3	0.0	1.0	2.0	3
4	1.0	2.0	3.0	4
5	2.0	3.0	4.0	5
6	3.0	4.0	5.0	6
7	4.0	5.0	6.0	7
8	5.0	6.0	7.0	8
9	6.0	7.0	8.0	9

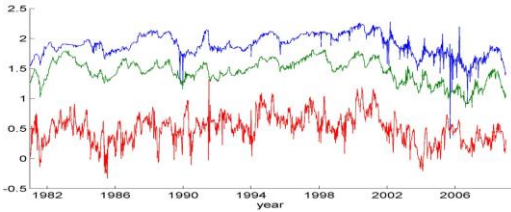
Beispiel 1: Fenstergröße $L=3$
Dimension Input =3, Output=1

Input Vergangenheit timesteps=2			Output Zukunft timesteps=2	
	$\text{var1}(t-2)$	$\text{var1}(t-1)$	$\text{var1}(t)$	$\text{var1}(t+1)$
2	0.0	1.0	2	3.0
3	1.0	2.0	3	4.0
4	2.0	3.0	4	5.0
5	3.0	4.0	5	6.0
6	4.0	5.0	6	7.0
7	5.0	6.0	7	8.0
8	6.0	7.0	8	9.0

Beispiel 2: Fenstergröße $L=2$
Dimension Input = 2, Output=2

Multivariate Zeitserie

Beispiel



multivariate
Zeitreihe
gegeben

var1	var2	var3
1	10	100
2	20	200
3	30	300
4	40	400
5	50	500
6	60	600
7	70	700
8	80	800
9	90	900
10	100	1000

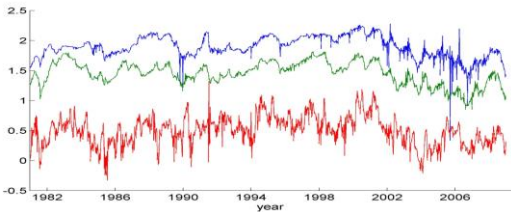


Input Vergangenheit timesteps=1			Output Zukunft timesteps=1		
var1(t-1)	var2(t-1)	var3(t-1)	var1(t)	var2(t)	var3(t)
1.0	10.0	100.0	2	20	200
2.0	20.0	200.0	3	30	300
3.0	30.0	300.0	4	40	400
4.0	40.0	400.0	5	50	500
5.0	50.0	500.0	6	60	600
6.0	60.0	600.0	7	70	700
7.0	70.0	700.0	8	80	800
8.0	80.0	800.0	9	90	900
9.0	90.0	900.0	10	100	1000

Beispiel 3: Fenstergröße L=1
Dimension Input = 3, Output=3

Multivariate Zeitserie

Beispiel



multivariate
Zeitreihe
gegeben

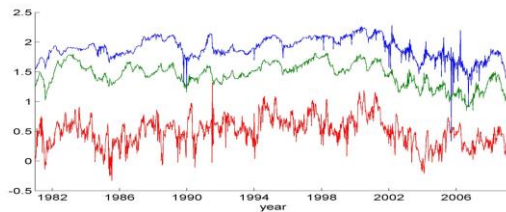
var1	var2	var3
1	10	100
2	20	200
3	30	300
4	40	400
5	50	500
6	60	600
7	70	700
8	80	800
9	90	900
10	100	1000



Input Vergangenheit timesteps=1			Output Zukunft timesteps=1
var1(t-1)	var2(t-1)	var3(t-1)	var1(t)
1.0	10.0	100.0	2
2.0	20.0	200.0	3
3.0	30.0	300.0	4
4.0	40.0	400.0	5
5.0	50.0	500.0	6
6.0	60.0	600.0	7
7.0	70.0	700.0	8
8.0	80.0	800.0	9
9.0	90.0	900.0	10

Beispiel 3: Fenstergröße L=1
Dimension Input = 3, Output=1

Multivariate Zeitserie Beispiel



multivariate
Zeitserie
gegeben

var1	var2	var3
1	10	100
2	20	200
3	30	300
4	40	400
5	50	500
6	60	600
7	70	700
8	80	800
9	90	900
10	100	1000



Input Vergangenheit timesteps=2						Output Zukunft timesteps=1		
var1(t-2)	var2(t-2)	var3(t-2)	var1(t-1)	var2(t-1)	var3(t-1)	var1(t)	var2(t)	var3(t)
1.0	10.0	100.0	2.0	20.0	200.0	3	30	300
2.0	20.0	200.0	3.0	30.0	300.0	4	40	400
3.0	30.0	300.0	4.0	40.0	400.0	5	50	500
4.0	40.0	400.0	5.0	50.0	500.0	6	60	600
5.0	50.0	500.0	6.0	60.0	600.0	7	70	700
6.0	60.0	600.0	7.0	70.0	700.0	8	80	800
7.0	70.0	700.0	8.0	80.0	800.0	9	90	900
8.0	80.0	800.0	9.0	90.0	900.0	10	100	1000

Beispiel 4: Fenstergröße $L=2$
Dimension Input = 6, Output=3

Funktion series_to_supervised

Aufbereitung der Zeitserie mit gewünschtem Input und Output

- Funktioniert für univariate und multivariate Zeitserien
- Fenstergröße wird berücksichtigt
- Berücksichtigt Anzahl der Zeitschritte pro Feature beim Input und beim Output
- Spalten, die nicht benötigt werden, können nach Aufruf einfach entfernt werden

```
def series_to_supervised(data, n_in=1, n_out=1, dropnan=True):
    """
    Frame a time series as a supervised learning dataset.
    Arguments:
        data: Sequence of observations as a list or NumPy array.
        n_in: Number of lag observations as input (X).
        n_out: Number of observations as output (y).
        dropnan: Boolean whether or not to drop rows with NaN values.
    Returns:
        Pandas DataFrame of series framed for supervised learning.
    """
```

Umwandlung von 2D Matrix in 3D Matrix für die Verwendung bei rekurrenten Netzwerktypen

- Separiere Output von Matrix
- Verwende Methode „reshape“ zum Umwandlung von 2D in 3D Tensor in folgender Form <samples><timesteps><features> für den `input_shape` Parameter des RNN/LSTMs Netzwerks

timesteps=3

	var1(t-3)	var1(t-2)	var1(t-1)
	0.0	1.0	2.0
	1.0	2.0	3.0
	2.0	3.0	4.0
	3.0	4.0	5.0
	4.0	5.0	6.0
	5.0	6.0	7.0
	6.0	7.0	8.0

Samples =7

samples=7
timesteps=3
features=1
array.reshape (7,3,1)

timesteps=1

	var1(t-1)	var2(t-1)
	0.0	50.0
	1.0	51.0
	2.0	52.0
	3.0	53.0
	4.0	54.0
	5.0	55.0
	6.0	56.0
	7.0	57.0
	8.0	58.0

Samples =9

samples=9
timesteps=1
features=2
array.reshape (9,1,2)

Umwandlung von 2D Matrix in 3D Tensor für die Verwendung bei rekurrenten Netzwerktypen (Beispiele)

var1(t-3)	var1(t-2)	var1(t-1)
0.0	1.0	2.0
1.0	2.0	3.0
2.0	3.0	4.0
3.0	4.0	5.0
4.0	5.0	6.0
5.0	6.0	7.0
6.0	7.0	8.0

train_X [7,3]:
1 Feature (var1)
3 Timesteps

timesteps

features

train_X = train_X.reshape((train_X.shape[0], 3, 1))
model = Sequential()
model.add(LSTM(50, input_shape=(3, 1))---- Anzahl der Samples
wird nicht angegeben

var1(t-1)	var2(t-1)
0.0	50.0
1.0	51.0
2.0	52.0
3.0	53.0
4.0	54.0
5.0	55.0
6.0	56.0
7.0	57.0
8.0	58.0

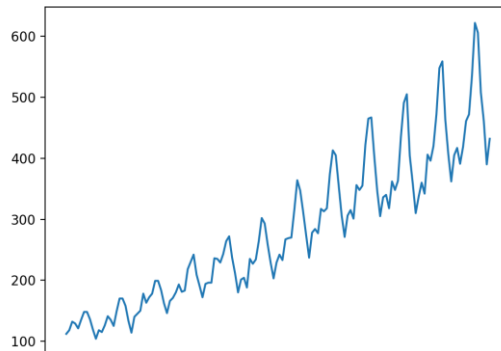
train_X [9,2]
2 Features: var1, var2
1 Timestep

timesteps

features

train_X = train_X.reshape((train_X.shape[0], 1, 2))
model = Sequential()
model.add(LSTM(50, input_shape=(1, 2))

Beispiel für Data Preparation und Verwendung von RNN/LSTM



```
"1949-01";112  
"1949-02";118  
"1949-03";132  
"1949-04";129  
"1949-05";121  
"1949-06";135  
"1949-07";148  
"1949-08";148  
"1949-09";136  
"1949-10";119  
"1949-11";104  
"1949-12";118  
.....
```

Siehe Code von `airline_pass_RNN.py`