RNN

Cadmus Yuan

2022.03.20

Content

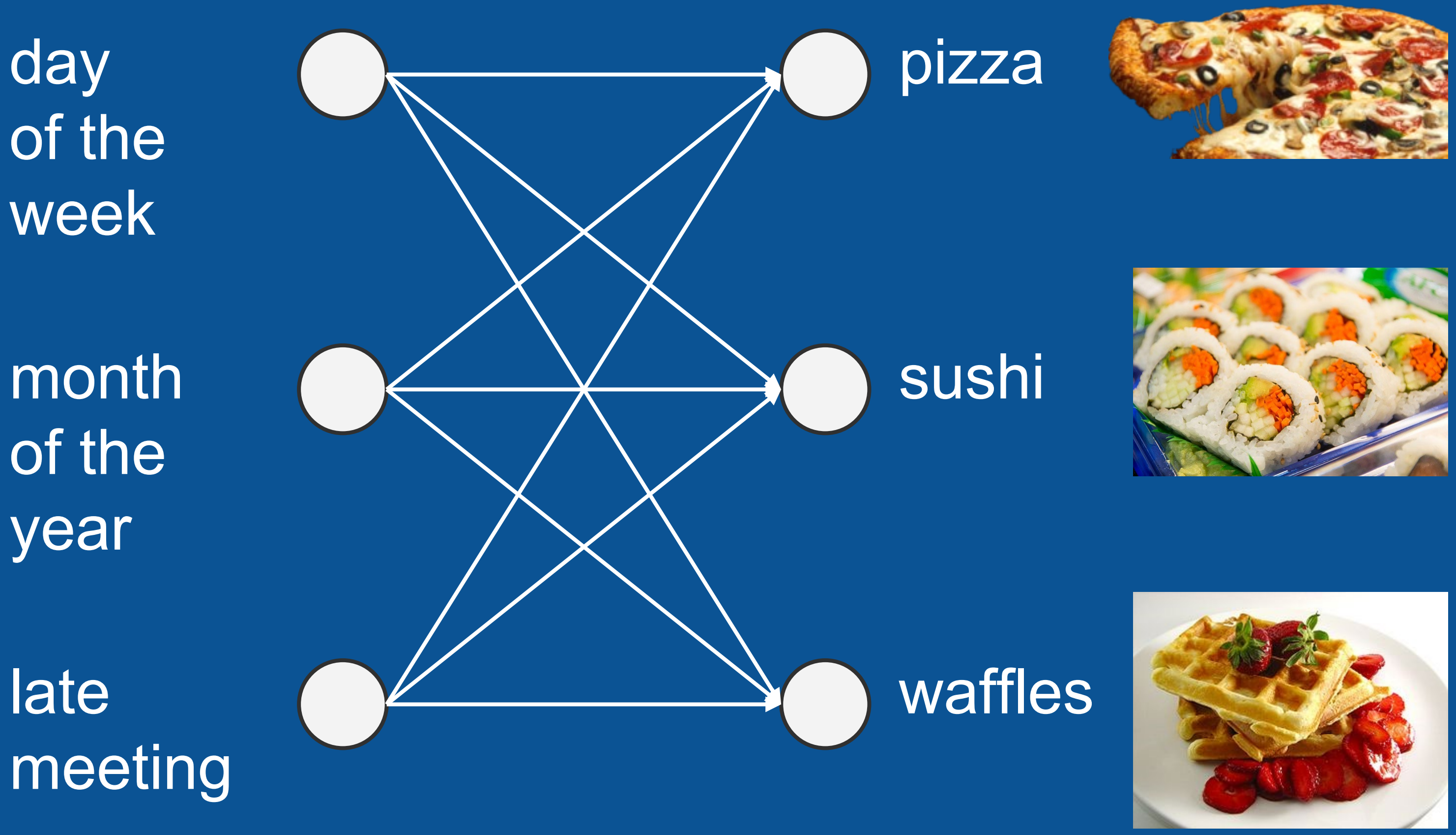
- What is the characteristics in RNN

- What is the difference in ML point of new
  - Different types of inputs
  - Different learning sequence

- Examples

- Continous with LSTM

What’s for dinner?



day of the week

month of the year

late meeting

pizza

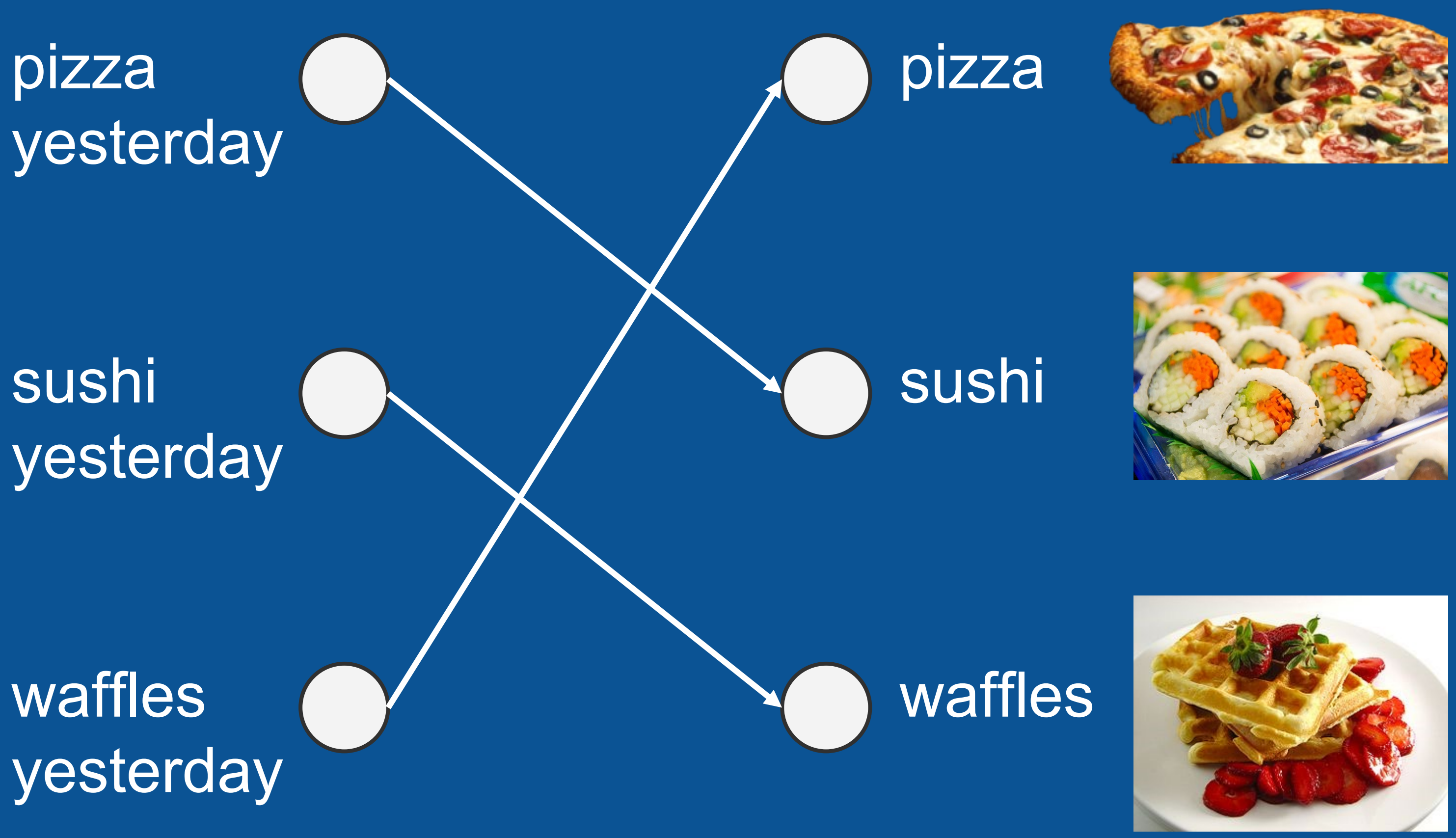
sushi

waffles

Taking from Brandon Rohrer
“How Long Short-Term Memory (LSTM)
and Recurrent Neural Networks (RNNs) work”

e2eml.school

What’s for dinner?



pizza yesterday

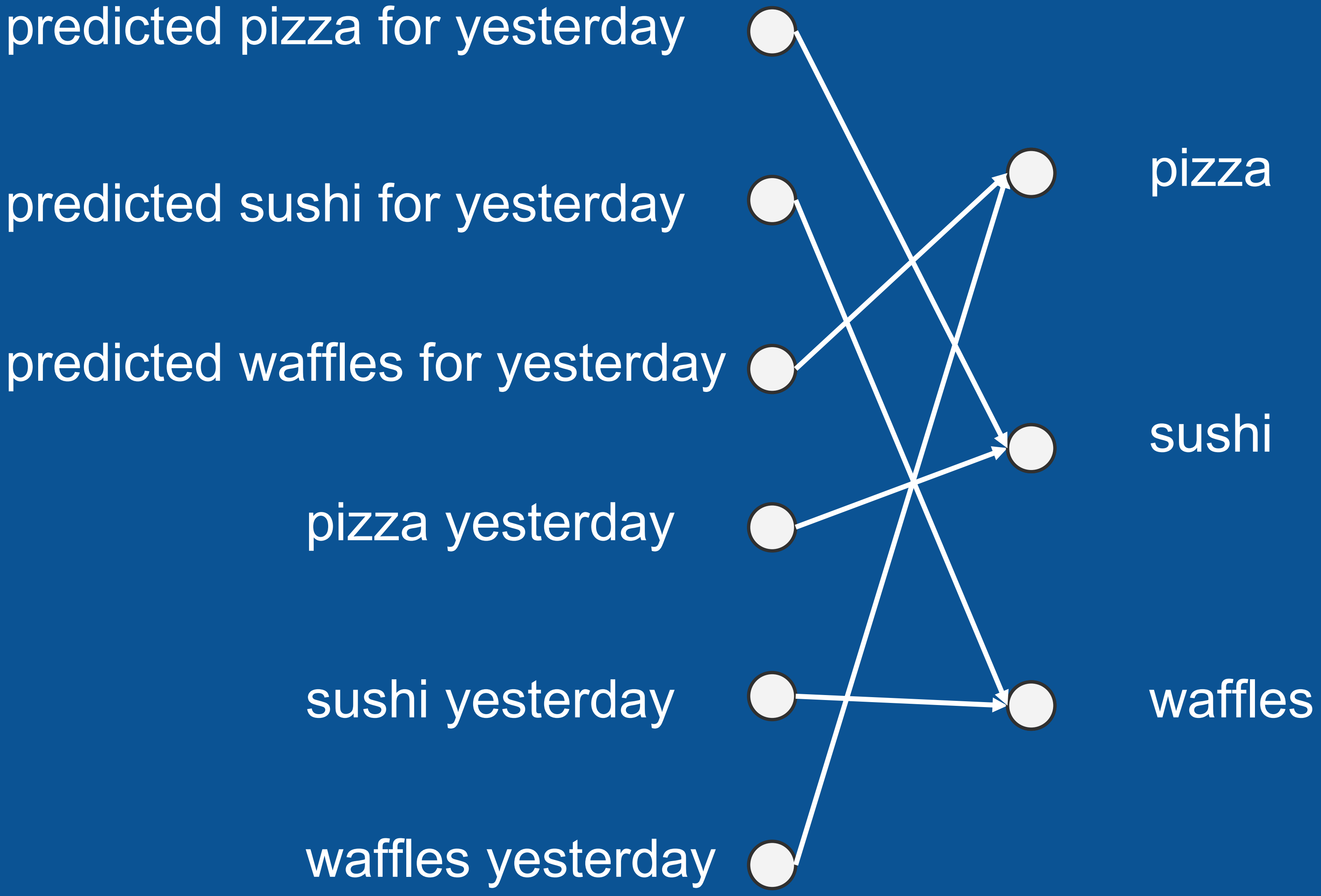
sushi yesterday

waffles yesterday

pizza

sushi

waffles



predicted pizza for yesterday

predicted sushi for yesterday

predicted waffles for yesterday

pizza yesterday

sushi yesterday

waffles yesterday

pizza

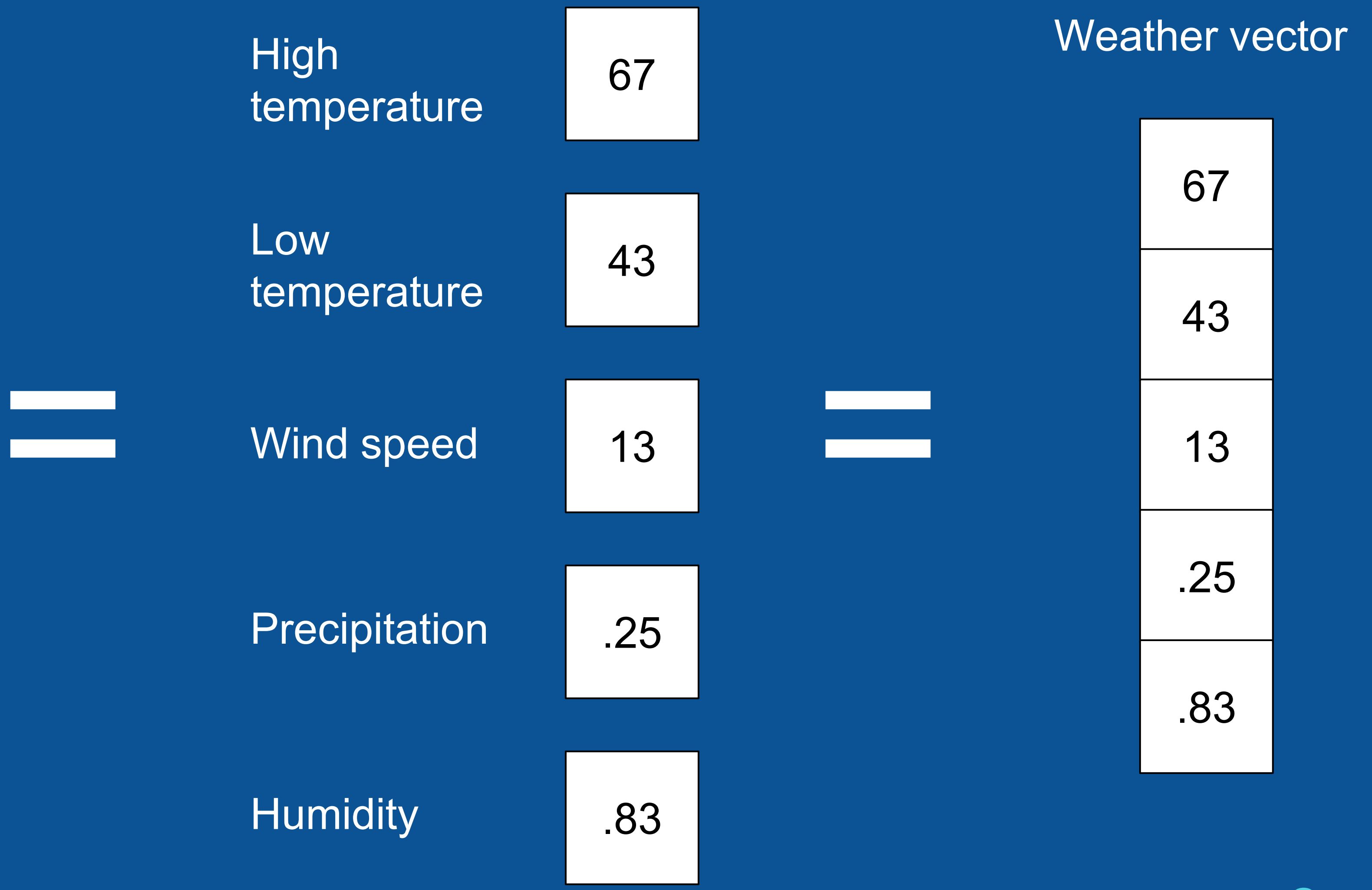
sushi

waffles

e2eml.school

A vector is a list of values

“High is 67 F.
Low is 43 F.
Wind is 13 mph.
.25 inches of rain.
Relative humidity
is 83%.”



=

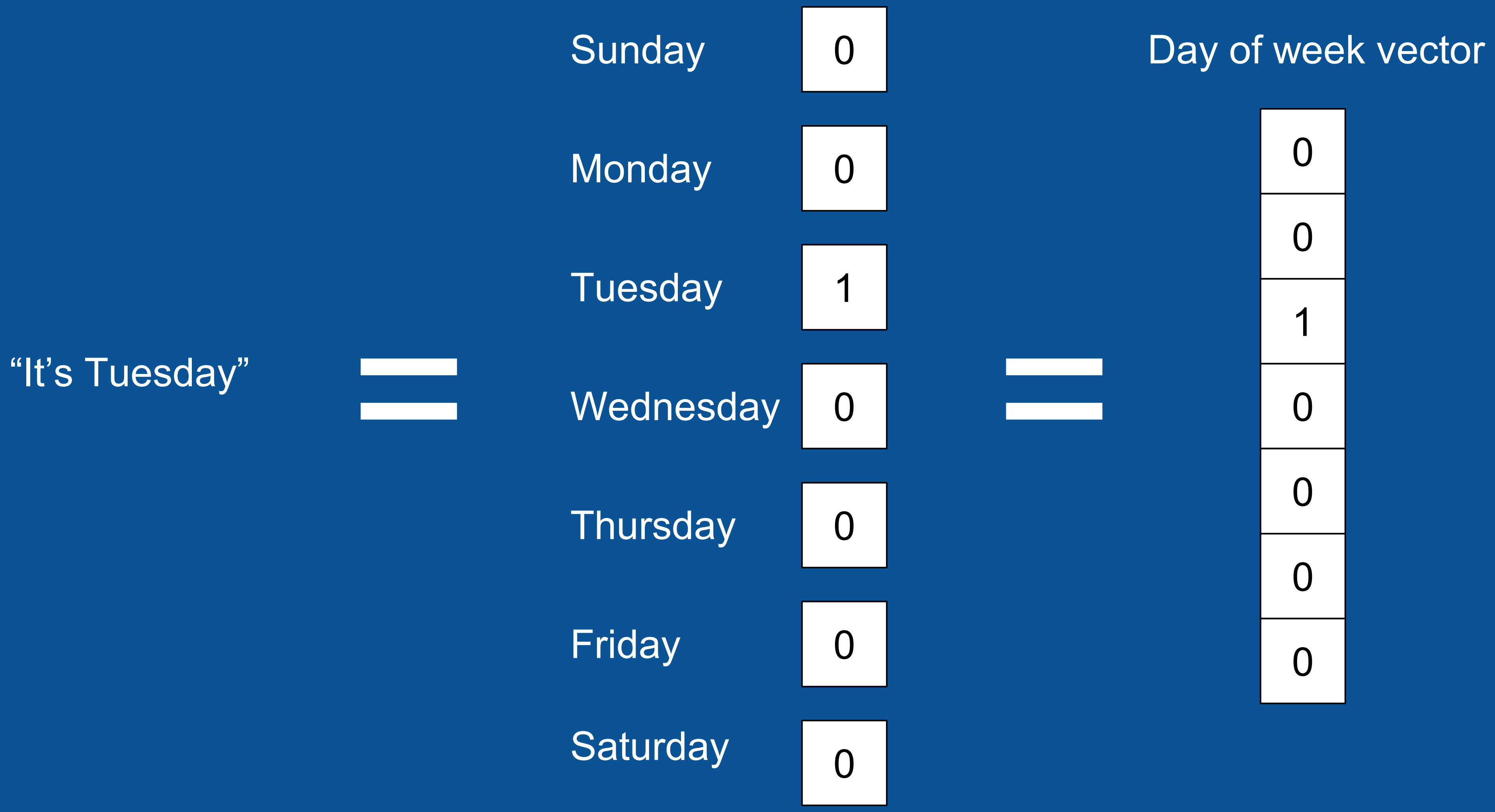
High temperature	67
Low temperature	43
Wind speed	13
Precipitation	.25
Humidity	.83

=

Weather vector

67
43
13
.25
.83

A vector is a list of values



“It’s Tuesday” =

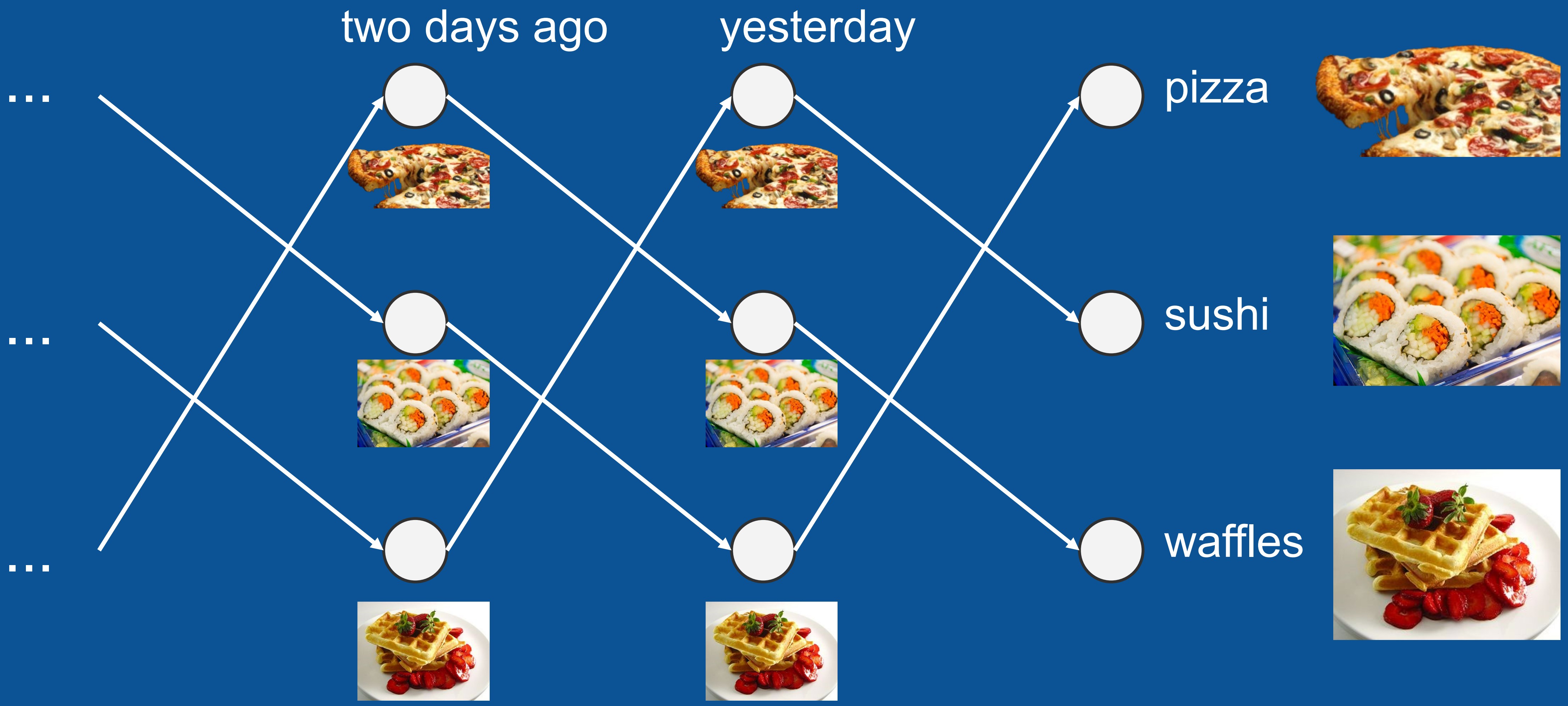
Sunday	0
Monday	0
Tuesday	1
Wednesday	0
Thursday	0
Friday	0
Saturday	0

=

Day of week vector

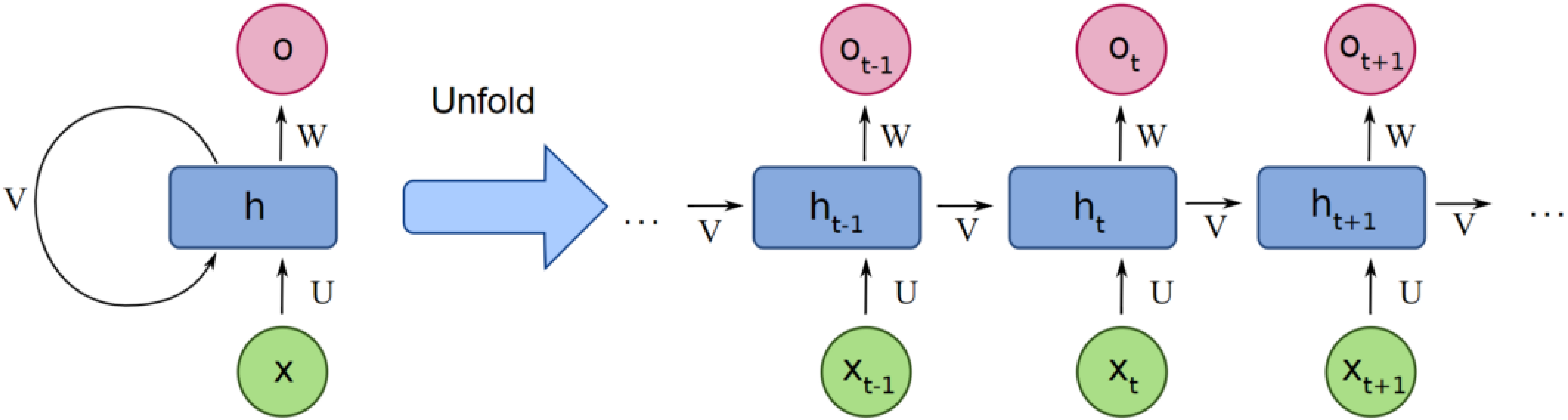
$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Unrolled predictions



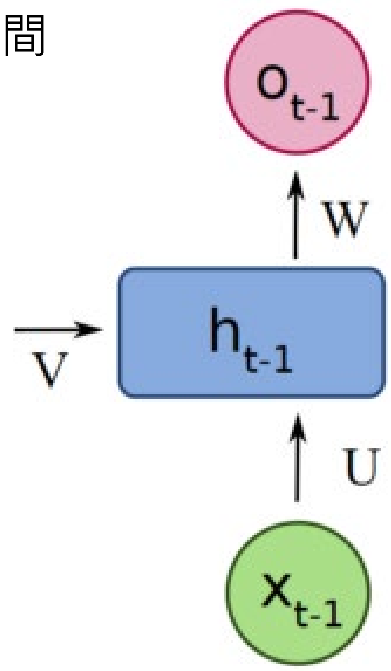
e2eml.school

The structure of RNN



What is the difference of RNN: the inputs

在ＲＮＮ中，把不同的時間考慮成時間區段



If this is the core NN, what are the inputs??

V and x_{t-1}. There are 3 different kind of inputs:

- Constant through all time
- Known varible through time
- The outputs from previous time

What is the learning technique (1)

- I use eligible trace to accelerate the computation

$w_{new} = w_{old} - \eta\delta h^n h^{n-1}$, if I use $\delta h = \frac{\delta h_{t-1}}{\delta h_{t-2}} \cdot \delta h$

Sounds if in each cycle, the δh is increasing, then increasing.

What is the learning technique (2)

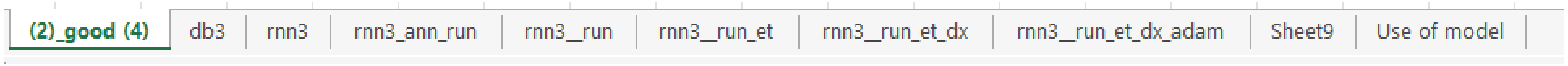
- I also can use the δx from the previous step to modify this step

$$o_t = o_t + \delta x_{t-1}$$

the delta from the previous should be included by output of this step

Excel demo

- Overview of all sheets



How to use

Base of NN

Summary

database

Rnn +et + dx + adam

Rnn way

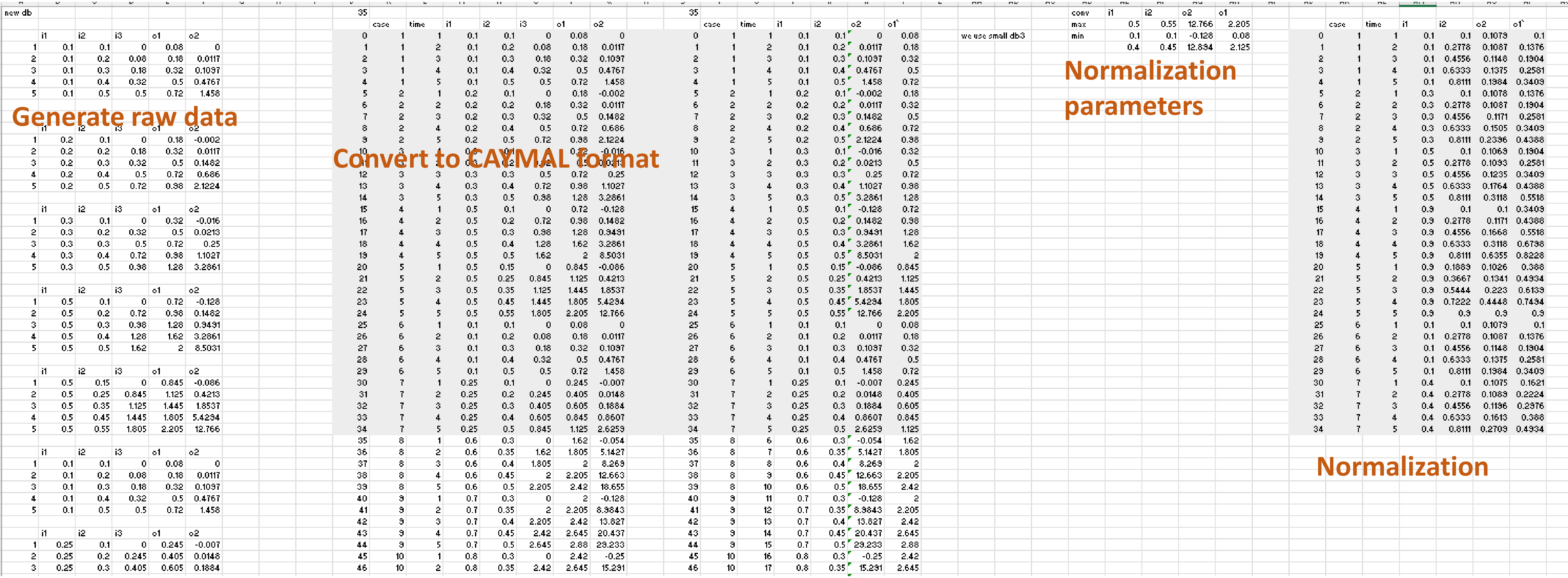
Basic rnn

Run rnn using ann

db

- Parameter
  - i1: it is fixed for all time
  - i2: it will change by time
  - i3: it is taken from the previous step
- I generate more datasets than I use for training
- I use CAYMAL to obtain the first set

- In the beginning , O1 is taken to i3
- But in CAYMAL, the recurrent para should be at last parameter
- So I swap O1 and O2 for CAYMAL input



new db

Generate raw data

	i1	i2	i3	o1	o2
1	0.1	0.1	0	0.08	0
2	0.1	0.2	0.08	0.18	0.0117
3	0.1	0.3	0.18	0.32	0.1097
4	0.1	0.4	0.32	0.5	0.4767
5	0.1	0.5	0.5	0.72	1.458
	i1	i2	i3	o1	o2
1	0.2	0.1	0	0.18	-0.002
2	0.2	0.2	0.18	0.32	0.0117
3	0.2	0.3	0.32	0.5	0.1482
4	0.2	0.4	0.5	0.72	0.686
5	0.2	0.5	0.72	0.98	2.1224
	i1	i2	i3	o1	o2
1	0.3	0.1	0	0.32	-0.016
2	0.3	0.2	0.32	0.5	0.0213
3	0.3	0.3	0.5	0.72	0.25
4	0.3	0.4	0.72	0.98	1.1027
5	0.3	0.5	0.98	1.28	3.2861
	i1	i2	i3	o1	o2
1	0.5	0.1	0	0.72	-0.128
2	0.5	0.2	0.72	0.98	0.1482
3	0.5	0.3	0.98	1.28	0.9491
4	0.5	0.4	1.28	1.62	3.2861
5	0.5	0.5	1.62	2	8.5031
	i1	i2	i3	o1	o2
1	0.5	0.15	0	0.845	-0.086
2	0.5	0.25	0.845	1.125	0.4213
3	0.5	0.35	1.125	1.445	1.8537
4	0.5	0.45	1.445	1.805	5.4294
5	0.5	0.55	1.805	2.205	12.766
	i1	i2	i3	o1	o2
1	0.1	0.1	0	0.08	0
2	0.1	0.2	0.08	0.18	0.0117
3	0.1	0.3	0.18	0.32	0.1097
4	0.1	0.4	0.32	0.5	0.4767
5	0.1	0.5	0.5	0.72	1.458
	i1	i2	i3	o1	o2
1	0.25	0.1	0	0.245	-0.007
2	0.25	0.2	0.245	0.405	0.0148
3	0.25	0.3	0.405	0.605	0.1884

Convert to CAYMAL format

35

	case	time	i1	i2	i3	o1	o2
0	1	1	0.1	0.1	0	0.08	0
1	1	2	0.1	0.2	0.08	0.18	0.0117
2	1	3	0.1	0.3	0.18	0.32	0.1097
3	1	4	0.1	0.4	0.32	0.5	0.4767
4	1	5	0.1	0.5	0.5	0.72	1.458
5	2	1	0.2	0.1	0	0.18	-0.002
6	2	2	0.2	0.2	0.18	0.32	0.0117
7	2	3	0.2	0.3	0.32	0.5	0.1482
8	2	4	0.2	0.4	0.5	0.72	0.686
9	2	5	0.2	0.5	0.72	0.98	2.1224
10	3	1	0.3	0.1	0	0.32	-0.016
11	3	2	0.3	0.2	0.32	0.5	0.0213
12	3	3	0.3	0.3	0.5	0.72	0.25
13	3	4	0.3	0.4	0.72	0.98	1.1027
14	3	5	0.3	0.5	0.98	1.28	3.2861
15	4	1	0.5	0.1	0	0.72	-0.128
16	4	2	0.5	0.2	0.72	0.98	0.1482
17	4	3	0.5	0.3	0.98	1.28	0.9491
18	4	4	0.5	0.4	1.28	1.62	3.2861
19	4	5	0.5	0.5	1.62	2	8.5031
20	5	1	0.5	0.15	0	0.845	-0.086
21	5	2	0.5	0.25	0.845	1.125	0.4213
22	5	3	0.5	0.35	1.125	1.445	1.8537
23	5	4	0.5	0.45	1.445	1.805	5.4294
24	5	5	0.5	0.55	1.805	2.205	12.766
25	6	1	0.1	0.1	0	0.08	0
26	6	2	0.1	0.2	0.08	0.18	0.0117
27	6	3	0.1	0.3	0.18	0.32	0.1097
28	6	4	0.1	0.4	0.32	0.5	0.4767
29	6	5	0.1	0.5	0.5	0.72	1.458
30	7	1	0.25	0.1	0	0.245	-0.007
31	7	2	0.25	0.2	0.245	0.405	0.0148
32	7	3	0.25	0.3	0.405	0.605	0.1884
33	7	4	0.25	0.4	0.605	0.845	0.8607
34	7	5	0.25	0.5	0.845	1.125	2.6259
35	8	1	0.6	0.3	0	1.62	-0.054
36	8	2	0.6	0.35	1.62	1.805	5.1427
37	8	3	0.6	0.4	1.805	2	8.269
38	8	4	0.6	0.45	2	2.205	12.663
39	8	5	0.6	0.5	2.205	2.42	18.655
40	9	1	0.7	0.3	0	2	-0.128
41	9	2	0.7	0.35	2	2.205	8.9843
42	9	3	0.7	0.4	2.205	2.42	13.827
43	9	4	0.7	0.45	2.42	2.645	20.437
44	9	5	0.7	0.5	2.645	2.88	29.233
45	10	1	0.8	0.3	0	2.42	-0.25
46	10	2	0.8	0.35	2.42	2.645	15.291

35

	case	time	i1	i2	o2	o1`
0	1	1	0.1	0.1	0	0.08
1	1	2	0.1	0.2	0.0117	0.18
2	1	3	0.1	0.3	0.1097	0.32
3	1	4	0.1	0.4	0.4767	0.5
4	1	5	0.1	0.5	1.458	0.72
5	2	1	0.2	0.1	-0.002	0.18
6	2	2	0.2	0.2	0.0117	0.32
7	2	3	0.2	0.3	0.1482	0.5
8	2	4	0.2	0.4	0.686	0.72
9	2	5	0.2	0.5	2.1224	0.98
10	3	1	0.3	0.1	-0.016	0.32
11	3	2	0.3	0.2	0.0213	0.5
12	3	3	0.3	0.3	0.25	0.72
13	3	4	0.3	0.4	1.1027	0.98
14	3	5	0.3	0.5	3.2861	1.28
15	4	1	0.5	0.1	-0.128	0.72
16	4	2	0.5	0.2	0.1482	0.98
17	4	3	0.5	0.3	0.9491	1.28
18	4	4	0.5	0.4	3.2861	1.62
19	4	5	0.5	0.5	8.5031	2
20	5	1	0.5	0.15	-0.086	0.845
21	5	2	0.5	0.25	0.4213	1.125
22	5	3	0.5	0.35	1.8537	1.445
23	5	4	0.5	0.45	5.4294	1.805
24	5	5	0.5	0.55	12.766	2.205
25	6	1	0.1	0.1	0	0.08
26	6	2	0.1	0.2	0.0117	0.18
27	6	3	0.1	0.3	0.1097	0.32
28	6	4	0.1	0.4	0.4767	0.5
29	6	5	0.1	0.5	1.458	0.72
30	7	1	0.25	0.1	-0.007	0.245
31	7	2	0.25	0.2	0.0148	0.405
32	7	3	0.25	0.3	0.1884	0.605
33	7	4	0.25	0.4	0.8607	0.845
34	7	5	0.25	0.5	2.6259	1.125
35	8	6	0.6	0.3	-0.054	1.62
36	8	7	0.6	0.35	5.1427	1.805
37	8	8	0.6	0.4	8.269	2
38	8	9	0.6	0.45	12.663	2.205
39	8	10	0.6	0.5	18.655	2.42
40	9	11	0.7	0.3	-0.128	2
41	9	12	0.7	0.35	8.9843	2.205
42	9	13	0.7	0.4	13.827	2.42
43	9	14	0.7	0.45	20.437	2.645
44	9	15	0.7	0.5	29.233	2.88
45	10	16	0.8	0.3	-0.25	2.42
46	10	17	0.8	0.35	15.291	2.645

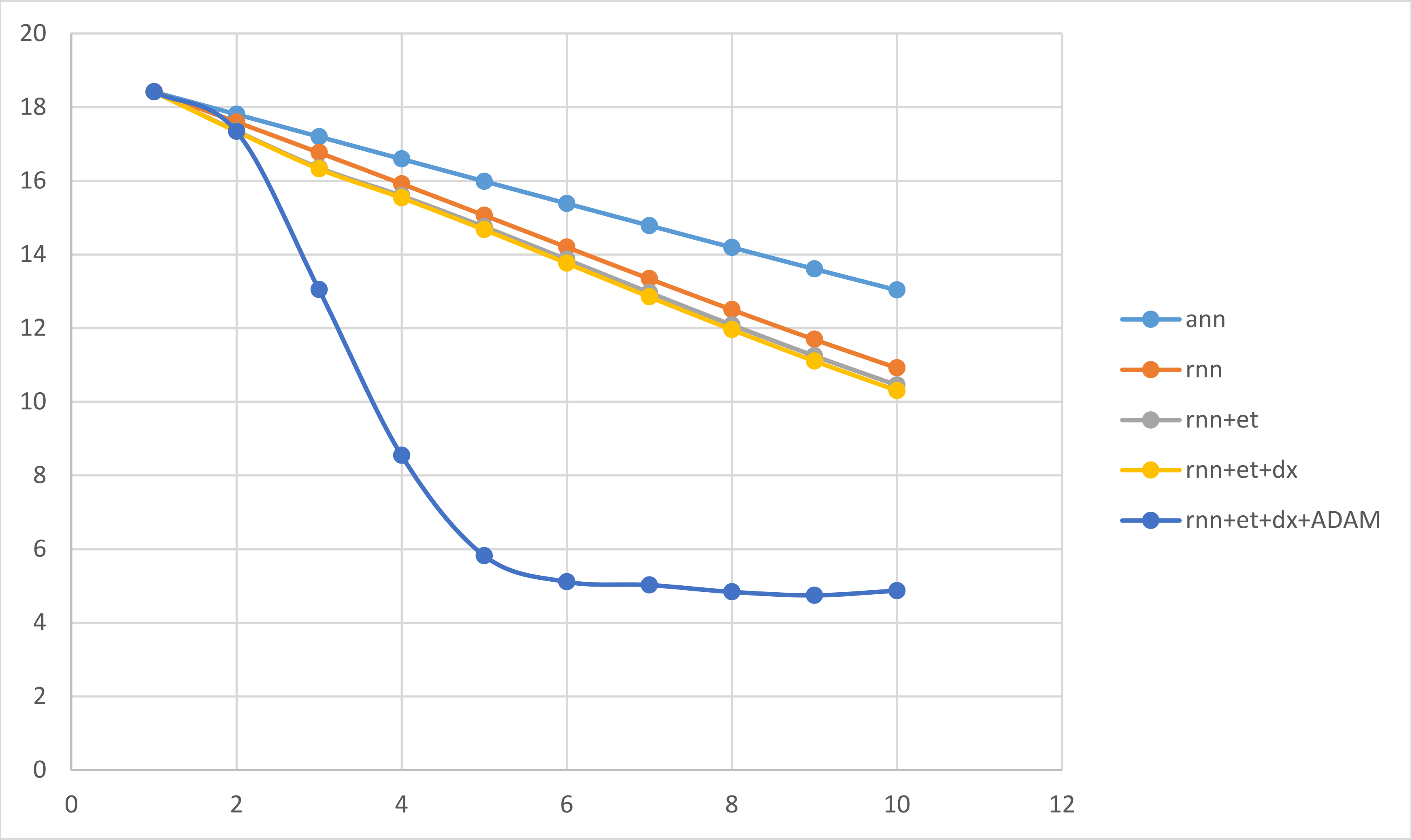
	conv	i1	i2	o2	o1
	max	0.5	0.55	12.766	2.205
we use small db3	min	0.1	0.1	-0.128	0.08
		0.4	0.45	12.894	2.125

Normalization parameters

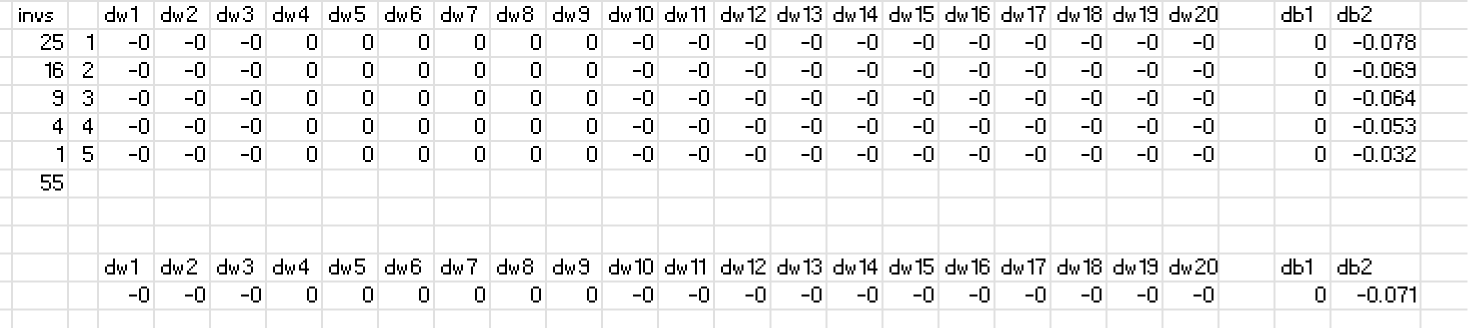
	case	time	i1	i2	o2	o1`
0	1	1	0.1	0.1	0.1079	0.1
1	1	2	0.1	0.2778	0.1087	0.1376
2	1	3	0.1	0.4556	0.1148	0.1904
3	1	4	0.1	0.6333	0.1375	0.2581
4	1	5	0.1	0.8111	0.1984	0.3409
5	2	1	0.3	0.1	0.1078	0.1376
6	2	2	0.3	0.2778	0.1087	0.1904
7	2	3	0.3	0.4556	0.1171	0.2581
8	2	4	0.3	0.6333	0.1505	0.3409
9	2	5	0.3	0.8111	0.2396	0.4388
10	3	1	0.5	0.1	0.1069	0.1904
11	3	2	0.5	0.2778	0.1093	0.2581
12	3	3	0.5	0.4556	0.1235	0.3409
13	3	4	0.5	0.6333	0.1764	0.4388
14	3	5	0.5	0.8111	0.3118	0.5518
15	4	1	0.9	0.1	0.1	0.3409
16	4	2	0.9	0.2778	0.1171	0.4388
17	4	3	0.9	0.4556	0.1668	0.5518
18	4	4	0.9	0.6333	0.3118	0.6798
19	4	5	0.9	0.8111	0.6355	0.8228
20	5	1	0.9	0.1889	0.1026	0.388
21	5	2	0.9	0.3667	0.1341	0.4934
22	5	3	0.9	0.5444	0.223	0.6139
23	5	4	0.9	0.7222	0.4448	0.7494
24	5	5	0.9	0.9	0.9	0.9
25	6	1	0.1	0.1	0.1079	0.1
26	6	2	0.1	0.2778	0.1087	0.1376
27	6	3	0.1	0.4556	0.1148	0.1904
28	6	4	0.1	0.6333	0.1375	0.2581
29	6	5	0.1	0.8111	0.1984	0.3409
30	7	1	0.4	0.1	0.1075	0.1621
31	7	2	0.4	0.2778	0.1089	0.2224
32	7	3	0.4	0.4556	0.1196	0.2976
33	7	4	0.4	0.6333	0.1613	0.388
34	7	5	0.4	0.8111	0.2709	0.4934

Normalization

Training technique



The rnn way

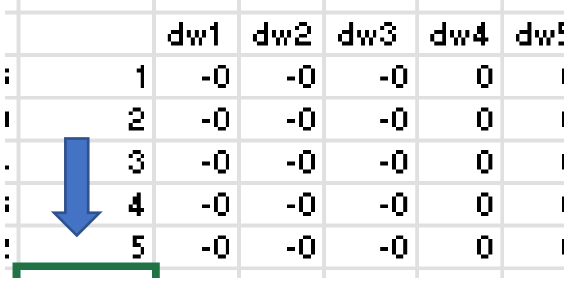


invs		dw1	dw2	dw3	dw4	dw5	dw6	dw7	dw8	dw9	dw10	dw11	dw12	dw13	dw14	dw15	dw16	dw17	dw18	dw19	dw20		db1	db2
25	1	-0	-0	-0	0	0	0	0	0	0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0		0	-0.078
16	2	-0	-0	-0	0	0	0	0	0	0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0		0	-0.069
9	3	-0	-0	-0	0	0	0	0	0	0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0		0	-0.064
4	4	-0	-0	-0	0	0	0	0	0	0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0		0	-0.053
1	5	-0	-0	-0	0	0	0	0	0	0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0		0	-0.032
55																								

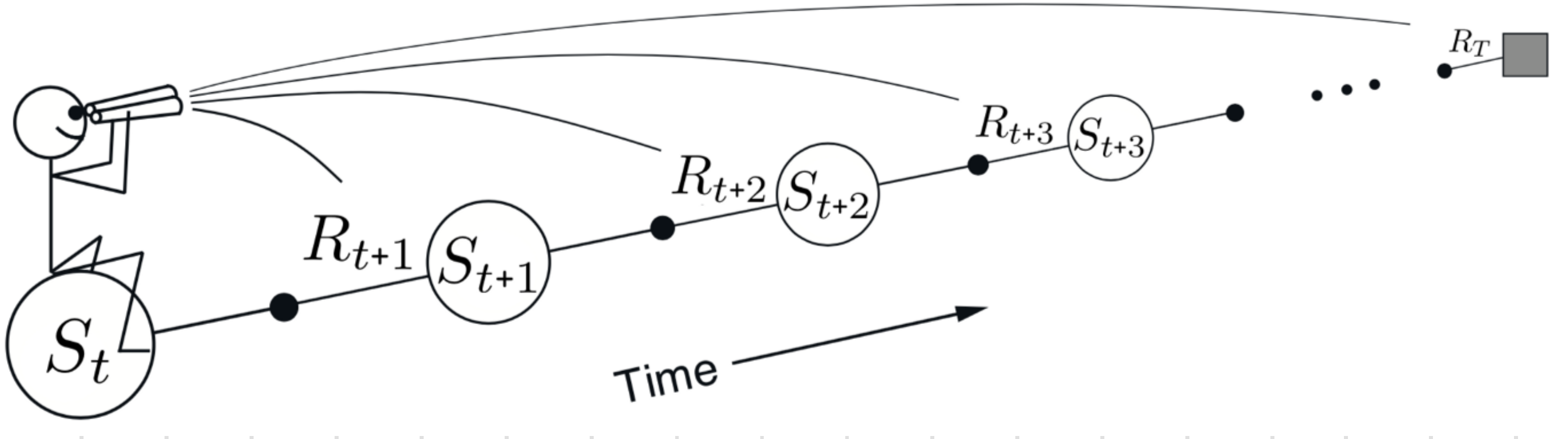
dw1	dw2	dw3	dw4	dw5	dw6	dw7	dw8	dw9	dw10	dw11	dw12	dw13	dw14	dw15	dw16	dw17	dw18	dw19	dw20		db1	db2
-0	-0	-0	0	0	0	0	0	0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0		0	-0.071

- The dw values are computed at each cycle, it is an average of the datapairs
- Different time has different weighting, t=1 has the highest weighting (why)??

Rnn+et



	dw1	dw2	dw3	dw4	dw5
1	-0	-0	-0	0	
2	-0	-0	-0	0	
3	-0	-0	-0	0	
4	-0	-0	-0	0	
5	-0	-0	-0	0	

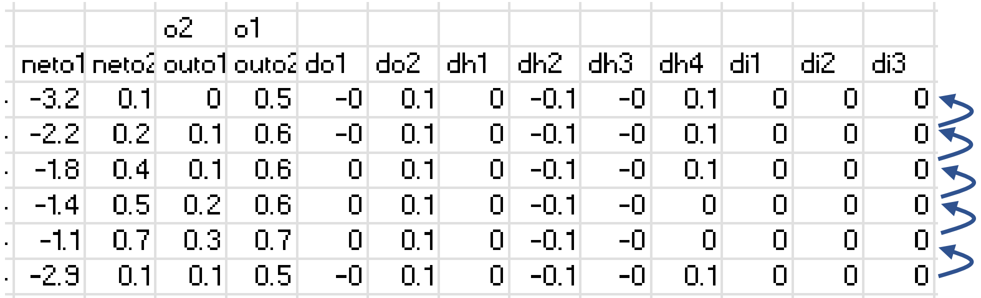


- $dw_{1,t=3} = (dw_{1,t=3}) * (\frac{dw_{1,t=2}}{dw_{1,t=1}})$

- If there is a fixed trend that found in previous 2 steps, the next step will be accelerated.

Rnn+et+dx

During our calculation, we have propagate the error through the network. But how about the dx?
In RNN, the dx is actually the output from the previous computation.



		o2	o1									
neto1	neto2	outo1	outo2	do1	do2	dh1	dh2	dh3	dh4	di1	di2	di3
-3.2	0.1	0	0.5	-0	0.1	0	-0.1	-0	0.1	0	0	0
-2.2	0.2	0.1	0.6	-0	0.1	0	-0.1	-0	0.1	0	0	0
-1.8	0.4	0.1	0.6	0	0.1	0	-0.1	-0	0.1	0	0	0
-1.4	0.5	0.2	0.6	0	0.1	0	-0.1	-0	0	0	0	0
-1.1	0.7	0.3	0.7	0	0.1	0	-0.1	-0	0	0	0	0
-2.9	0.1	0.1	0.5	-0	0.1	0	-0.1	-0	0.1	0	0	0