FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Investigation of Hierarchical Temporal Memory Spatial Pooler’s Noise Robustness and Specificity

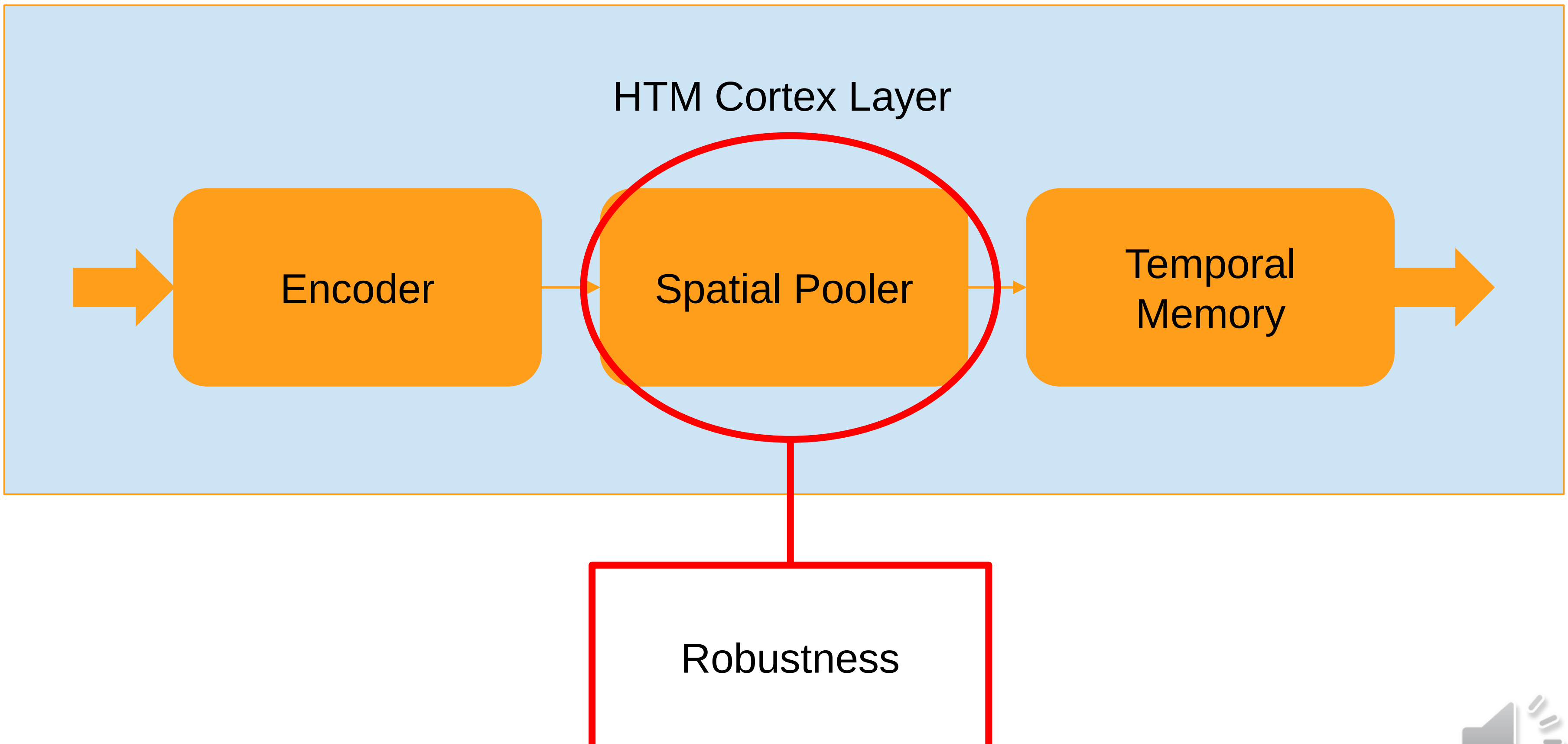
Sang Nguyen
phuocsangnguyen97@gmail.com

Duy Nguyen
ngthanhduy7@gmail.com

Fachbereich 2 Informatik und Ingenieurwissenschaften

Contents

Introduction



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Methods – making training data

$$f(x) = 10 \cdot \cos(0.01\pi \cdot x) \cdot \cos(0.05\pi \cdot x)$$

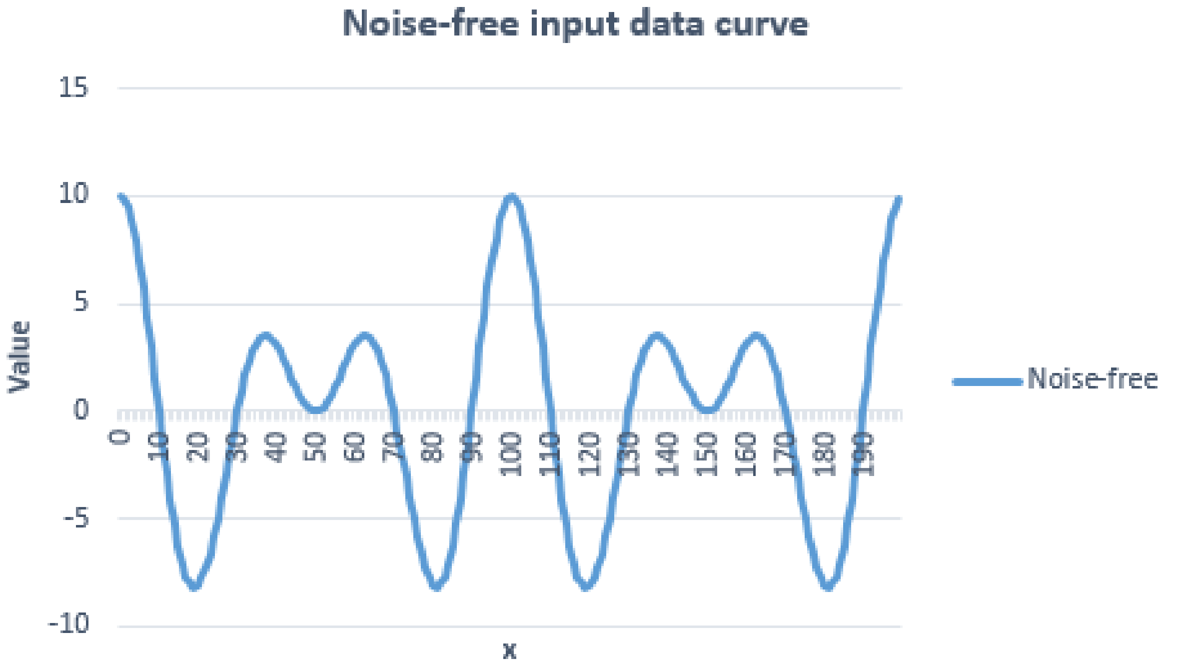


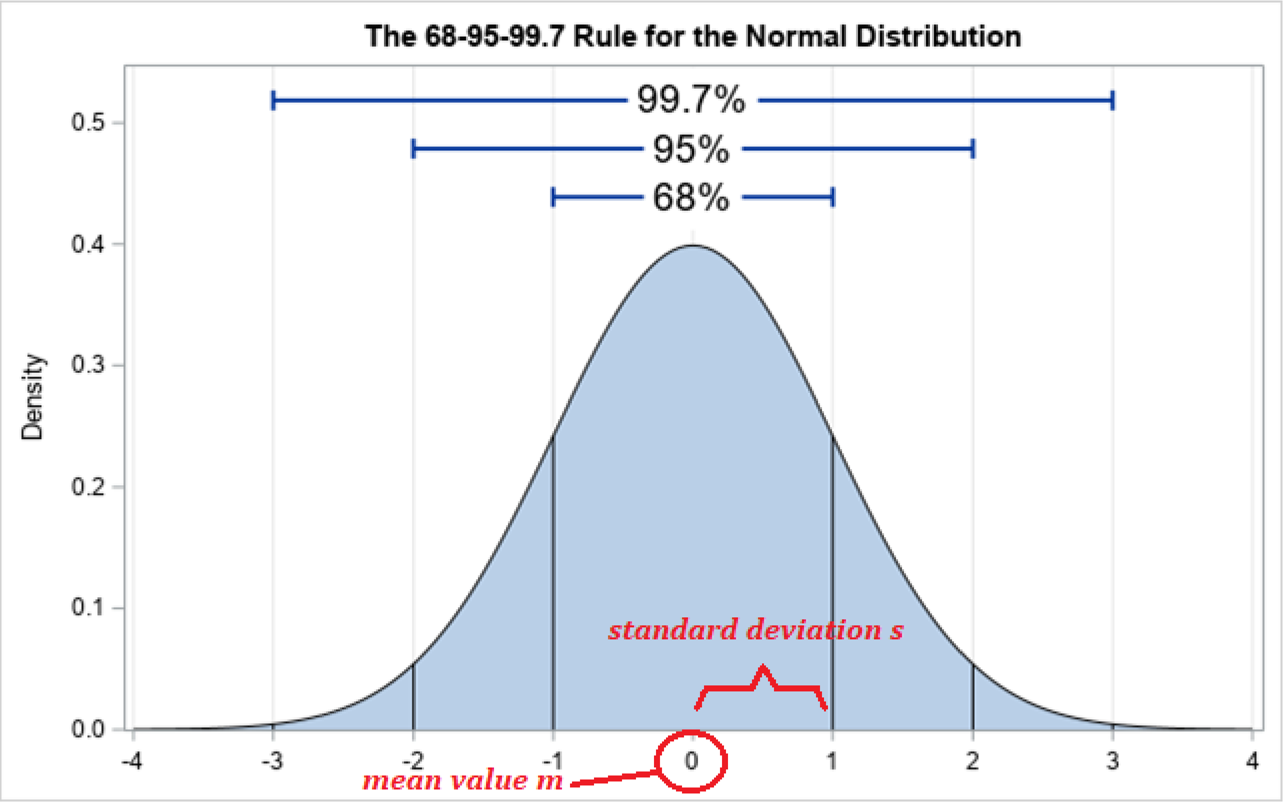
Figure 1. The 200 samples from the original input data set

Methods – making training data

	A	B
1	0	10
2	1	9.9
3	2	9.5
4	3	8.9
5	4	8
6	5	7
7	6	5.8
8	7	4.4
9	8	3
10	9	1.5
11	10	0
12	11	-1.5
13	12	-2.9
14	13	-4.2
15	14	-5.3
16	15	-6.3
17	16	-7.1
18	17	-7.7
19	18	-8
20	19	-8.2
21	20	-8.1
22	21	-7.8
23	22	-7.3

sinusoidal

Methods – making noisy data



(Source: https://blogs.sas.com/content/iml/2019/07/22/extreme-value-normal-data.html#prettyPhoto)

Methods – making noisy data

NORM.INV(RAND(), m, s)

Methods – making noisy data

C1 | =B1 + NORM.INV(RAND(),0,1)

	A	B	C
1	0	10	10.25059
2	1	9.9	9.644184
3	2	9.5	10.09109
4	3	8.9	9.75371
5	4	8	8.421129
6	5	7	7.540633
7	6	5.8	6.060184
8	7	4.4	3.776054
9	8	3	5.711293
10	9	1.5	1.836882
11	10	0	-1.50151
12	11	-1.5	-0.70024
13	12	-2.9	-4.55239
14	13	-4.2	-5.40321
15	14	-5.3	-5.66433
16	15	-6.3	-6.51344
17	16	-7.1	-6.25993
18	17	-7.7	-7.47766
19	18	-8	-5.5223
20	19	-8.2	-8.79862
21	20	-8.1	-7.8666
22	21	-7.8	-6.93527
23	22	-7.3	-7.12223

sinusoidal

READY

Methods – making noisy data

B1 | 10.3

	A	B	C	D	E	F
1	0	10.3				
2	1	9.6				
3	2	10.1				
4	3	9.8				
5	4	8.4				
6	5	7.5				
7	6	6.1				
8	7	3.8				
9	8	5.7				
10	9	1.8				
11	10	-1.5				
12	11	-0.7				
13	12	-4.6				
14	13	-5.4				
15	14	-5.7				
16	15	-6.5				
17	16	-6.3				
18	17	-7.5				
19	18	-5.5				
20	19	-8.8				
21	20	-7.9				
22	21	-6.9				
23	22	-7.1				

Noisy_N-0-1_sinusoidal

READY

Methods – making noisy data

Noise level: The ratio between standard deviation s and input resolution

Methods – making noisy data

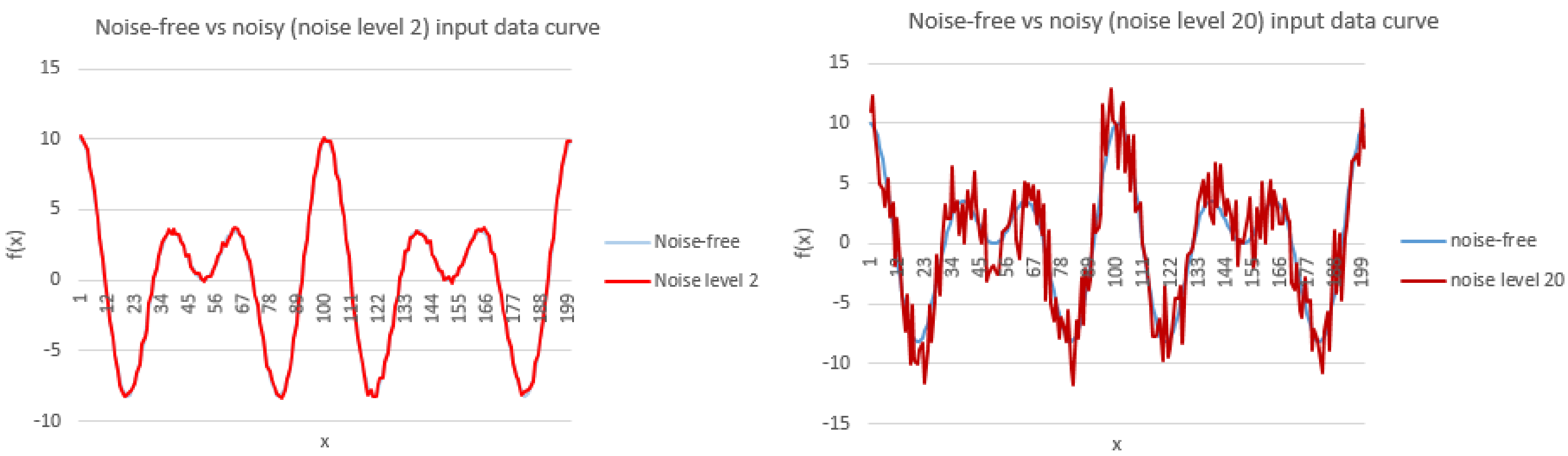


Figure 2. Comparison between original and noisy input data sets

Methods – Scalar Encoder‘s settings

Table 1. Scalar Encoder’s Settings

Parameter	Value
W	65
N	465
MinVal	-20.0
MaxVal	20.0
Periodic	false
ClipInput	true
Offset	108

=> Resolution 0.1

Methods – Spatial Pooler‘s settings

Table 2. Spatial Pooler’s Settings

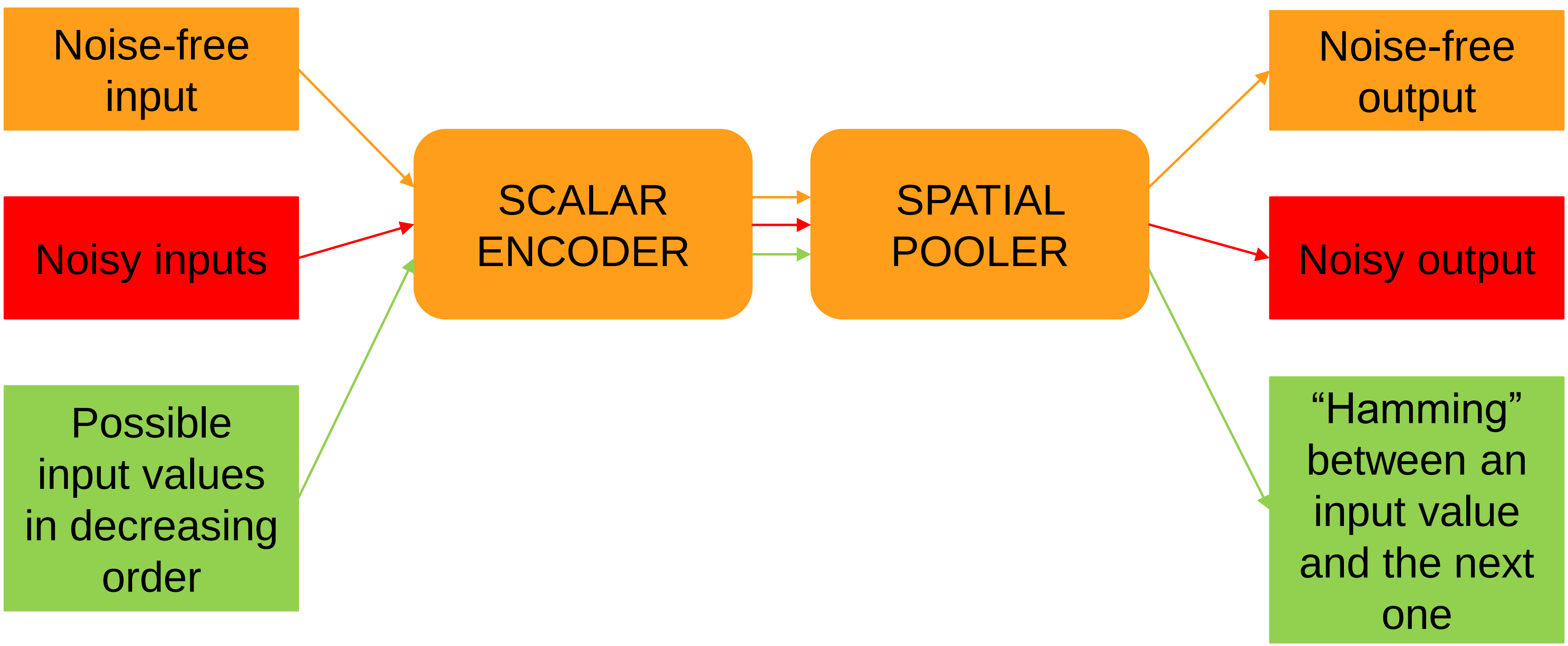
Parameter	Value
inputDimensions	465
comlumnsDimension	2048
potentialRadius	-1
potentialPct	1
globalInhibition	true
numActiveColumnsPerInhArea	0.02*2048 (2%)
stimulusThreshold	0.5
synPermInactiveDec	0.008
synPermActiveInc	0.01
synPermConnected	0.1
dutyCyclePeriod	100
maxBoost	10

Methods – Comparing function

Comparison function

```
public static double GetHammingDistance(int[] originArray,
int[] comparingArray, bool countNoneZerosOnly = false)
```

Methods - summarization



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Result and Discussion - Robustness

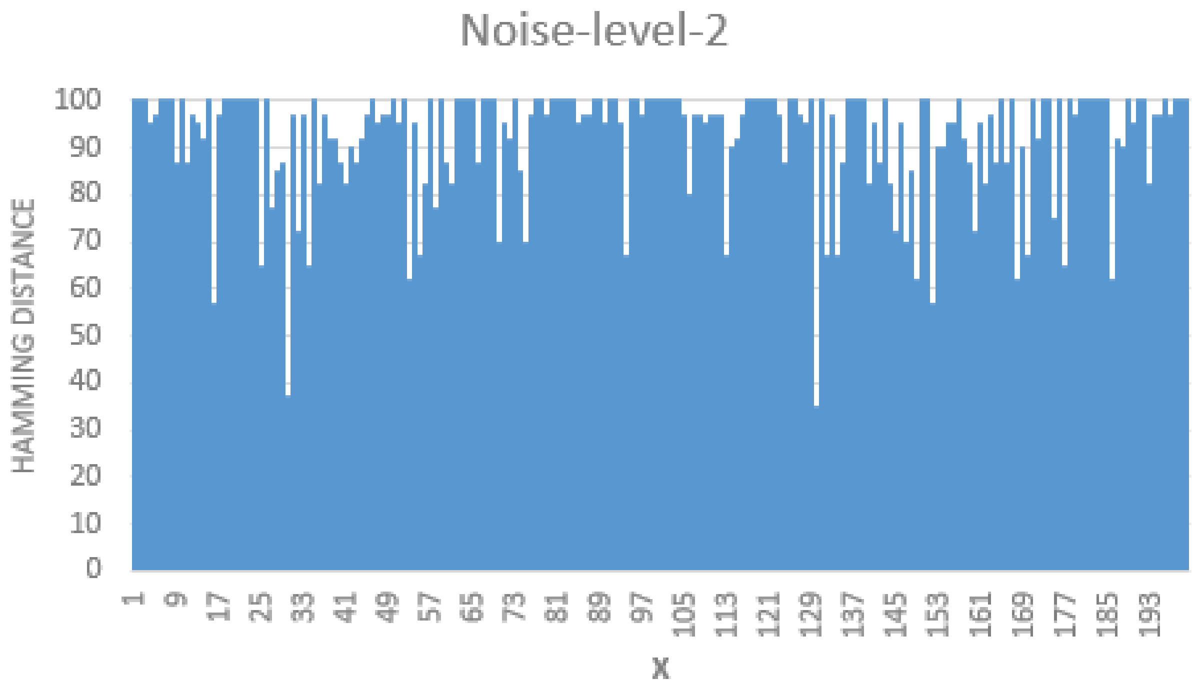


Figure 3. "Hamming distance" between original and noisy (noise-level-2) Spatial Pooler output data sets

Result and Discussion - Robustness

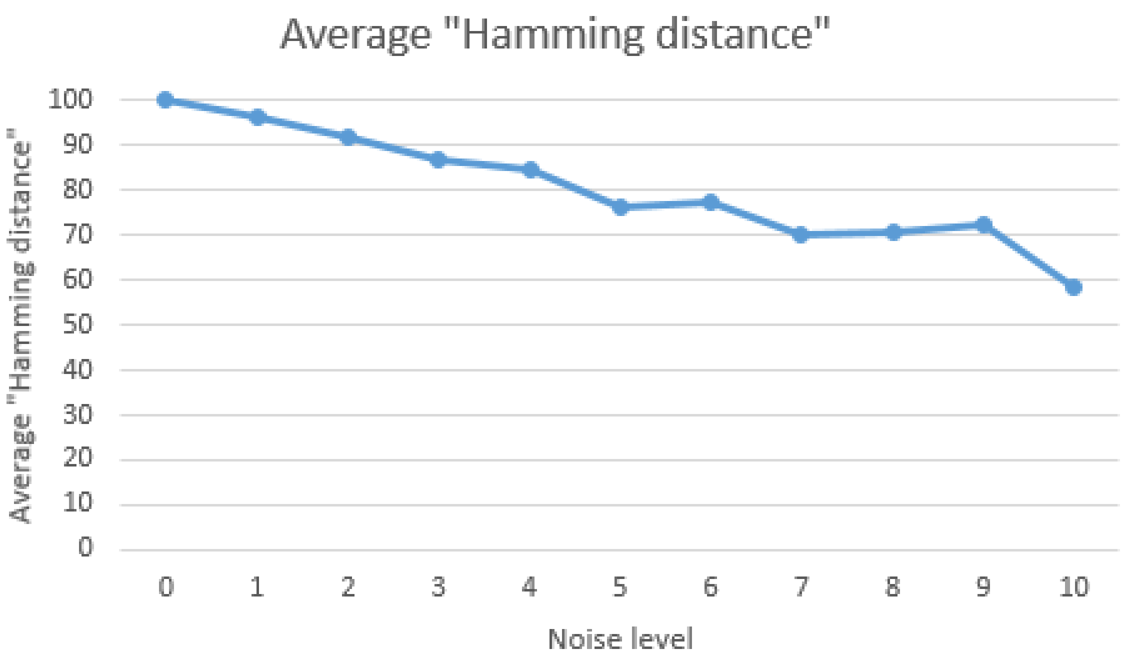
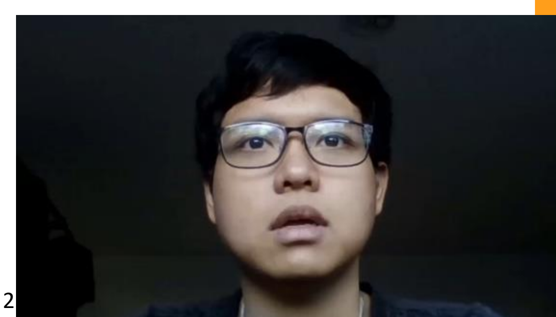


Figure 4. Average similarity between original and different levels of noisy input data

Result and Discussion

381	380	18
382	381	18.1
383	382	18.2
384	383	18.3
385	384	18.4
386	385	18.5
387	386	18.6
388	387	18.7
389	388	18.8
390	389	18.9
391	390	19
392	391	19.1
393	392	19.2
394	393	19.3
395	394	19.4
396	395	19.5
397	396	19.6
398	397	19.7
399	398	19.8
400	399	19.9
401	400	20

1. 2 data sets: Training set and testing set.
2. Training set: Integer numbers only, ranging from -20 to 20 with step of 1.
3. Testing set (noisy set): Decimal numbers, same range as above with step of 0.1.
4. SP learns only about training set then will have to predict testing set (decimal numbers).
5. Calculate average hamming distance for every numbers from every 0.1 step to the original integer to see how different the patterns are.



Result and Discussion

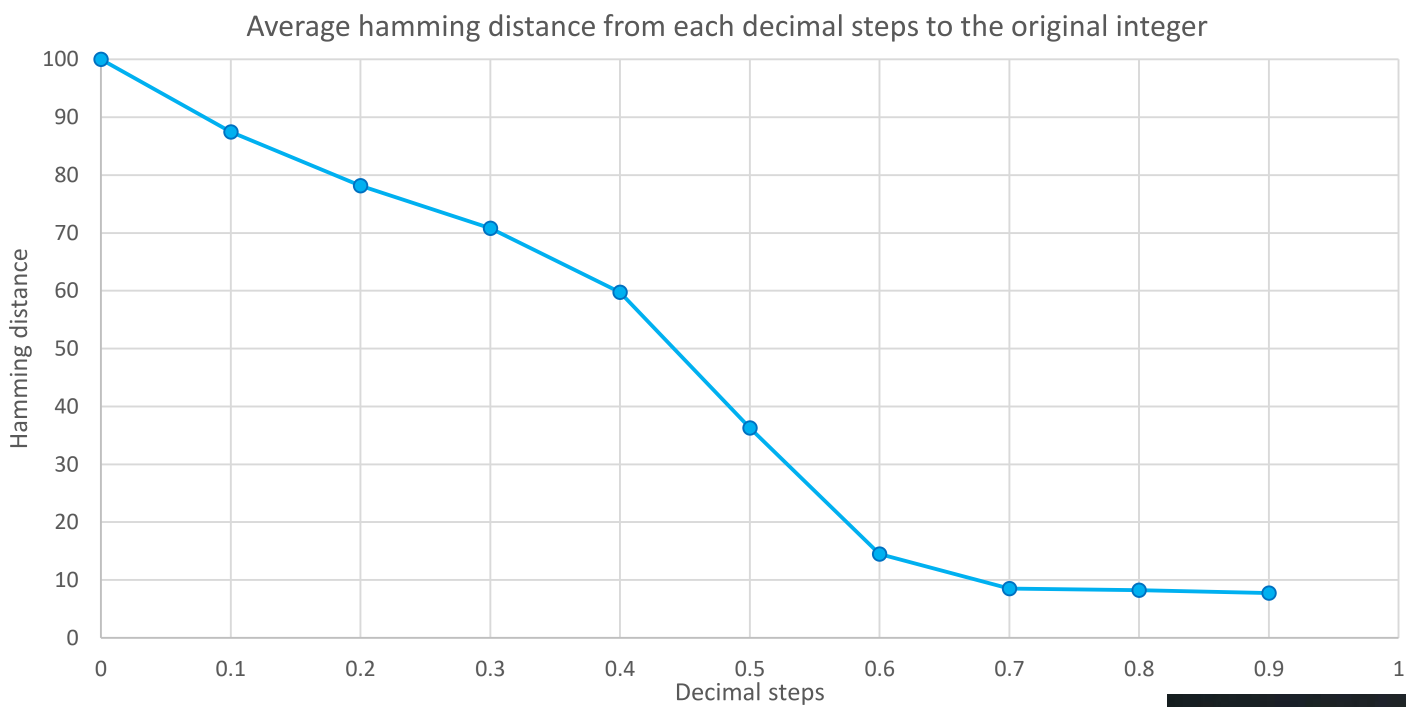


Figure 6. Average "Hamming distance" between each 0.1 decim
the integer number

Conclusion

- **Noise robustness:** robust against relatively low levels of noise

- **Specificity:** Moderate ability to differentiate two consecutively incremental input values

FRANKFURT
UNIVERSITY
OF APPLIED SCIENCES

Reference

[1] J. Hawkins and S. Blakeslee, *On intelligence*. New York: Times Books/Henry Holt, 2008.

[2] J. Hawkins and C. Maver, “Introduction Chapter,” in *Biological and Machine Intelligence*, Release 0.4., Numenta, 2016.

[3] J. Hawkins and C. Maver, “HTM Overview Chapter,” in *Biological and Machine Intelligence*, Release 0.4., Numenta, 2016.

[4] S. Purdy, “Encoding Data for HTM Systems,” in *Biological and Machine Intelligence*, Release 0.4., Numenta, 2016.

[5] S. Ahmad, M. Taylor, and Y. Cui, “Spatial Pooling Algorithm Details,” in *Biological and Machine Intelligence*, Release 0.4., Numenta, 2016.

[6] A. Lavin, S. Ahmad, and J. Hawkins, “Sparse Distributed Representations,” in *Biological and Machine Intelligence*, Release 0.4., Numenta, 2016.

[7] S. Ahmad and M. Lewis, “Temporal Memory Algorithm Details,” in *Biological and Machine Intelligence*, Release 0.4., Numenta, 2016.

[8] D. Dobric, “Influence of input sparsity to Hierarchical Temporal Memory Spatial Pooler noise robustness,” 2019.

[9] D. Dobric, “NeoCortexApi,” 2019. [Online]. Available: https://github.com/ddobric/neocortexapi/

Thank you for your time