Task:

Develop a Kohonen network where each unit takes 3 inputs. Your network should allow you to change the number of units to be 2, 3 or 4. Test your network with data for which you know the answers, showing your test results and explaining your test strategy.

Answer:

a) An introduction.

I developed 3 Kohonen networks where each unit takes 3 inputs, weights were randomised.

First 2-clusters Kohonen network (for 2 neurons).

Second 3-clusters Kohonen network (for 3 neurons).

Third 4-clusters Kohonen network (for 4 neurons).

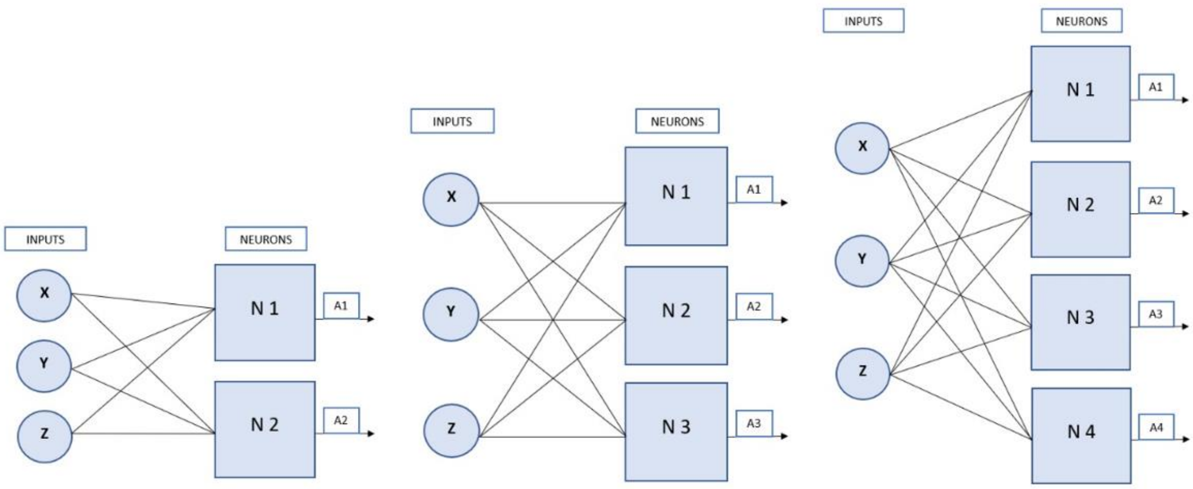


Figure 1. Example of my 2, 3, 4 neurons Kohonen networks

I have a Dataset with x, y, z coordinates for 24 point. This Dataset is the same for all 3 Kohonen networks.

To avoid a confusion in calculations, I named 24 points with a letters from A to X accordingly (A=1, B=2...X=24).

		DATASET		
3		X	Y	Z
4	A	-0.82	0.49	0.29
5	B	-0.77	0.47	0.43
6	C	-0.73	0.55	0.41
7	D	-0.71	0.61	0.36
8	E	-0.69	0.59	0.42
9	F	-0.68	0.6	0.42
10	G	-0.66	0.58	0.48
11	H	-0.61	0.69	0.4
12	I	0.28	0.96	0.01
13	J	0.31	0.95	0.07
14	K	0.35	-0.06	0.94
15	L	0.36	0.91	-0.22
16	M	0.37	0.93	0.09
17	N	0.43	0.83	-0.34
18	O	0.43	0.9	0.01
19	P	0.47	-0.1	0.88
20	Q	0.47	-0.07	0.88
21	R	0.48	0.82	-0.3
22	S	0.53	-0.23	0.82
23	T	0.54	-0.21	0.82
24	U	0.58	-0.38	0.72
25	V	0.58	0.81	0
26	W	0.65	-0.04	0.76
27	X	0.74	-0.05	0.67

Figure 2. Example of Dataset.

Learning rate was randomised as well and = 0,8.

b) A description of your program or spreadsheet and how you developed and tested it.

STEP 1. Normalise Dataset.

For each **input** in a dataset I calculated a **length** using formula: SQUAREROOT of $x^2 + y^2 + z^2$. Also, using formula: x ÷ length, y ÷ length, z ÷ length, I **normalised inputs** x, y, z on x', y', z'.

31		DATASET						
32		X	Y	Z	LEN	X'	Y'	Z'
33	A	-0.82	0.49	0.29	0.9983	-0.8214	0.49084	0.29049
34	B	-0.77	0.47	0.43	0.99935	-0.7705	0.47031	0.43028
35	C	-0.73	0.55	0.41	1.00175	-0.72873	0.54904	0.40928
36	D	-0.71	0.61	0.36	1.0029	-0.70795	0.60824	0.35896
37	E	-0.69	0.59	0.42	1.0003	-0.68979	0.58982	0.41987
38	F	-0.68	0.6	0.42	0.9994	-0.68041	0.60036	0.42025

Figure 3. Example of Dataset with length and normalised inputs.

STEP 2. Initialise neurons.

I set a random weights for 2, 3, 4 neurons.

STEP 3. Normalise neurons' weights.

For each **neuron** I calculated a **length** using formula: SQUAREROOT of $x^2 + y^2 + z^2$.
Also, using formula: x ÷ length, y ÷ length, z ÷ length, I **normalised neurons' weights from** x, y, z to x', y', z'.

NEURONS							
X	Y	Z		LEN	X'	Y'	Z'
-15	-20	-20	N1	32.0156	-0.46852	-0.6247	-0.6247
10	-10	10	N2	17.3205	0.57735	-0.57735	0.57735

Figure 4. Example of neuron's random weights, length and normalised neuron's weights.

STEP 4. Calculate NET values for all neurons.

Using formula: (normalised input x' × normalised weight x') + (normalised input y' × normalised weight y') + (normalised input z' × normalised weight z').

STEP 5. Determine "winning" neuron.

The winning neuron always should have a highest value.

	NET	ACT
N1	-0.10325	1
N2	-0.5899	0

Figure 5. Example of Net and "winning neuron.

STEP 6. Update weight for "winning" neuron only.

Using formula:
$Weight_{new} = Weight_{old} + Learning\ Rate \times (x' - Weight_{old})$
$Weight_{new} = Weight_{old} + Learning\ Rate \times (y' - Weight_{old})$
$Weight_{new} = Weight_{old} + Learning\ Rate \times (z' - Weight_{old})$

<u>STEP 7. Renormalise weight for “winning” neuron.</u>

For “winning” **neuron** I calculated a **length** using formula: SQUAREROOT of $x^2 + y^2 + z^2$. Also, using formula: x ÷ length, y ÷ length, z ÷ length, I **normalised “winning “neurons’ weights from** x, y, z to x’, y’, z’.

NEURONS							
X	Y	Z		LEN	X'	Y'	Z'
-0.75082	0.26773	-0.01748	N1	0.79732	-0.94168	0.33579	-0.02193
0.57735	-0.57735	0.57735	N2	1	0.57735	-0.57735	0.57735

Figure 6. Example of renormalised weights for "winning neuron".

c) Sets of results.

1. <u>2-clusters Kohonen network:</u>

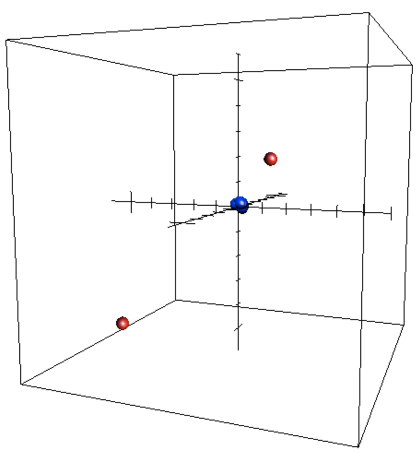
Start point.

NEURONS							
X	Y	Z		LEN	X'	Y'	Z'
-15	-20	-20	N1	32.01562	-0.46852	-0.6247	-0.6247
10	-10	10	N2	17.32051	0.57735	-0.57735	0.57735

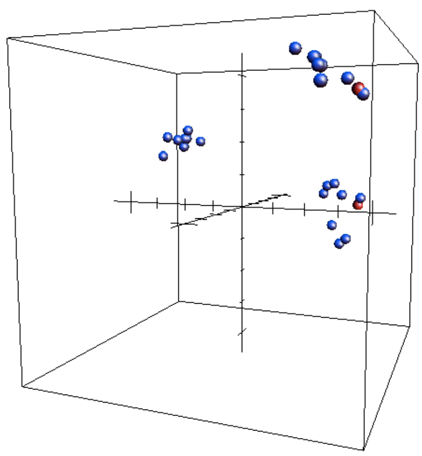
End Point.

NEURONS							
X	Y	Z		LEN	X'	Y'	Z'
0.564116	0.824142	-0.05062	N1	1	0.564116	0.824142	-0.05062
0.713033	-0.0552	0.694988	N2	0.997232	0.715012	-0.05536	0.696917

Start point.



End Point.



2. <u>3-clusters Kohonen network:</u>

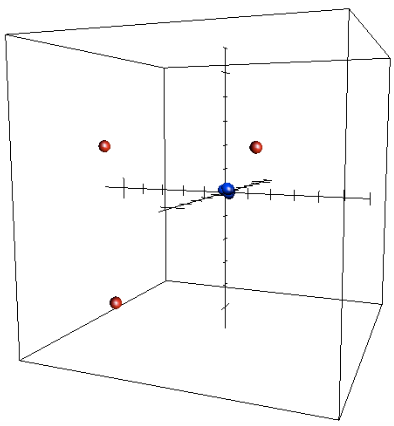
Start point.

NEURONS							
X	**Y**	Z		**LEN**	**X'**	**Y'**	**Z'**
-15	-20	-20	N1	32.01562	-0.46852	-0.6247	-0.6247
10	-10	10	N2	17.32051	0.57735	-0.57735	0.57735
-20	-15	10	N1	26.92582	-0.74278	-0.55709	0.371391

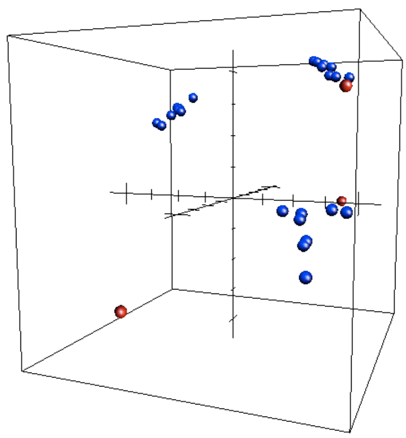
End Point.

NEURONS							
X	**Y**	Z		**LEN**	**X'**	**Y'**	**Z'**
-0.46852	-0.6247	-0.6247	N1	1	-0.46852	-0.6247	-0.6247
0.720109	-0.06044	0.688737	N2	0.998283	0.721348	-0.06055	0.689921
0.542253	0.835453	-0.08933	N3	1	0.542253	0.835453	-0.08933

Start point.



End Point.



3. <u>4-clusters Kohonen network:</u>

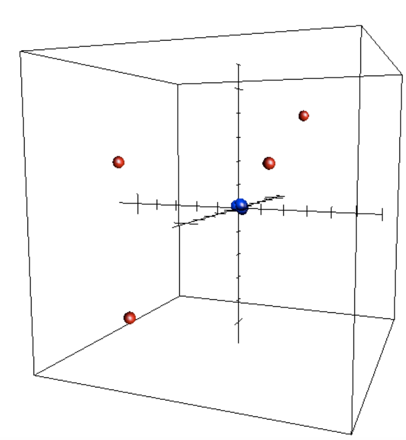
Start point.

NEURONS							
X	**Y**	Z		**LEN**	**X'**	**Y'**	**Z'**
-15	-20	-20	N1	32.01562	-0.46852	-0.6247	-0.6247
10	-10	10	N2	17.32051	0.57735	-0.57735	0.57735
-20	-15	10	N1	26.92582	-0.74278	-0.55709	0.371391
10	15	20	N2	26.92582	0.371391	0.557086	0.742781

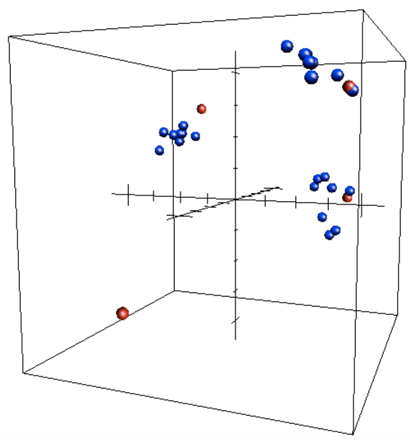
End Point.

NEURONS							
X	Y	Z		LEN	X'	Y'	Z'
-0.46852	-0.6247	-0.6247	N1	1	-0.46852	-0.6247	-0.6247
0.720152	-0.06045	0.6887	N2	0.998289	0.721386	-0.06055	0.68988
-0.51664	0.605508	0.605348	N3	1	-0.51664	0.605508	0.605348
0.564119	0.824141	-0.05062	N4	1	0.564119	0.824141	-0.05062

Start point.



End Point.



d) An analysis of the results obtained including any surprising or interesting observations that you made.

I noticed that with 2 neurons I do not have enough neurons for settle a dataset, because the dataset is split in 3 inputs.

With 3 neurons the dataset will settle in a perfect way, because the dataset is split exactly in 3 inputs.

With 4 neurons the dataset will be settled only with 3 neurons. The forth neuron will never “win”.

e) A conclusion.

For 3 inputs dataset, 2 neurons are not enough, 4 neurons have an extra one, and only 3 neurons will settle in a perfect way.