Self-Organizing Maps (SOM)

— Lecture Notes —

Laurenz Wiskott
Institut für Neuroinformatik
Ruhr-Universität Bochum, Germany, EU

27 May 2022

— Summary —

Self-Organizing Maps (SOM) are inspired from neural map formation in the visual system and mainly used for nonlinear dimensionality reduction and visualization. A set of nodes arranged in a predefined graph, often just a regular 2D-grid, are placed in high-dimensional space such that neighboring nodes in the graph are placed nearby in space and the graph as a whole covers the data distribution, much like in vector quantization.

— Lecture 1/1 —

1 The SOM Algorithm describes the SOM algorithm.

2 Applications shows some illustrative examples.

Contents

© 2011, 2017, 2022 Laurenz Wiskott (ORCID https://orcid.org/0000-0001-6237-740X, homepage https://www.ini.rub.de/PEOPLE/wiskott/). This work (except for all figures from other sources, if present) is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License, see https://creativecommons.org/licenses/by-sa/4.0/. If figures are not included for copyright reasons, they are uni colored, but the word 'Figure', 'Image', or the like in the reference is often linked to a freely available copy.
If applicable, core text and formulas are set in dark red, one can repeat the lecture notes quickly by just reading these; ♦ marks important formulas or items worth remembering and learning for an exam; ♢ marks less important formulas or items that I would usually also present in a lecture; + marks sections that I would usually skip in a lecture.
You can also download the teaching material of this topic as zip files and then view them locally on your computer.

LECTURE 1/0.5

1 The SOM Algorithm

Learning material:[1]

○ 15 min video 1 The SOM Algorithm by Laurenz Wiskott

■ Text below

[1] Generic instruction: Consider the (possibly nested) list of resources like a horizontal tree with an invisible root on the very left, and decide from left to right what you want to select to work through. The invisible root node has to be selected. For any selected parent node all children nodes marked with ■ or ● are mandatory and have to be selected. Children nodes marked with □ or ○ are optional and may be selected in addition to get a better understanding of the material. If a parent node has no mandatory child, then at least one optional child has to be selected. Children marked with + provide additional voluntary material that can be safely ignored, typically going beyond the scope of the section. Children of non-selected parents may be ignored. ■ and □ indicate children that cover (almost) the whole material of the section. Missing content might then be indicated by struck through references to the corresponding learning objectives. Items tend to be ordered by precedence and/or recommended temporal order from top to bottom, assuming that you prefer to first watch a video before reading through lecture notes. If a detailed table of content for videos or lecture notes is given, references to learning objectives might be provided in green, 1:30 should be read as 1 min and 30 seconds, and 1'30 should be read as page 1 at about 30% of the page. Video times may be linked directly to the indicated position in the video, but be aware that the video might be downloaded anew each time you click on a time. Resources without author name are usually authored by Laurenz Wiskott and his team.

In vector quantization and clustering the reference vectors (or cluster centers) are positioned entirely based on the structure of the data. However, it is sometimes desireable to also impose a topographic structure on the reference vectors themselves. For example, if the main purpose is visualization of the data, it is helpful to impose a two-dimensional structure on the reference vectors, because one can only plot in 2D. In such cases the reference vectors are typically arranged in a two-dimensional rectangular or hexagonal grid defining a neighborhood structure, and they are usually called units. Like in vector quantization the units are initially placed randomly in feature space. During learning they are updated and moved to where the data points lie with the additional objective to position neighboring units at nearby locations in feature space. The latter is achieved by always moving neighboring units together.

The most prominent algorithm of this kind is the self-organizing (feature) map (SOM) by Kohonen (Wikipedia, 2017). It works as follows:

1. Initialize all units with randomly drawn pointers. Alternatively, one can use the first stimuli to initialize the units to have a better starting configuration and save some learning.
2. Pick a data point at random.
3. Determine the unit that lies closest to the current data point. This is called the winning unit or, for short, the winner.
4. Move the winning unit a bit (let say 10%) towards the current data point. Also move the nearest neighbors of the winner towards the current data point but move them less (let say 5%). Maybe also move the second nearest neighbors but even less (let say 2%).

5. Repeat steps 2–4 over and over again to produce a SOM.

Moving the winning unit and its neighbors towards the current data point has two effects: Firstly, this process repeated over and over again moves all units into the data distribution, i.e. into the area where the data lie. Even a unit that is far away, so that initially it never becomes a winner, is dragged towards the data along with its neighboring units. Secondly, since not only the winner but also its neighbors are moved towards the current data point, neighboring units are dragged to similar locations, so that the imposed neighborhood structure is also preserved in feature space.

2 Demo

Figure 2.1 illustrates the self-organizing feature map in two examples.

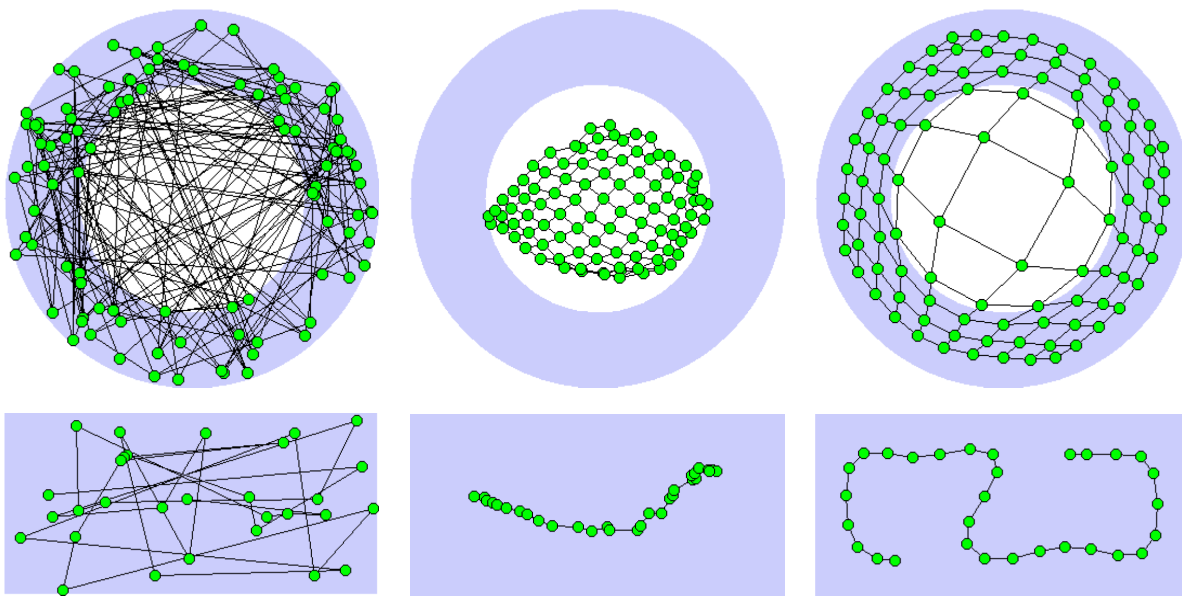


Figure 2.1: Two examples of a self-organizing map developing over time. The plots show a net of 10×10 units (top) and 1×30 units (bottom) after random initialization with data points (left), after 100 time steps (middle), and after convergence at 40000 time steps. One can see that the maps nicely try to cover the data distributions. But one sees also that the predefined structure does not fit the data, which leads to units sitting in empty regions (top) and nearby units that are not neighbors in the map (bottom). (Created with DemoGNG 1.5 written by Hartmut Loos and Bernd Fritzke, see https://www.demogng.de/js/demogng.html for a more recent version.)
© CC BY-SA 4.0

Run https://www.demogng.de/js/demogng.html and explore the model *Self-Organizing Map (Kohonen)*.

2 Applications

Learning material:

- Text below

SOM Analysis of the World Welfare

2/5 Figure: (Kaski & Kohonen (1996) 3rd Intl. Conf. on Neur. Netw. in the Capit. Markets, Fig. 1)

SOM analysis of the welfare of many countries in the world. The SOM is rectangular with an internal hexagonal structure. The feature vectors included 39 indicators of welfare related to health, education, consumption, and social services. Abbreviations show the countries that are associated with the nodes, those in lower case where not used during training, since too many indicators were missing. Grey values indicate local distances, with white indicating clustering nodes and black indicating distant nodes.
Figure: (Kaski and Kohonen, 1996, Fig. 1)

SOM Analysis of the World Welfare

Figure: (Kaski & Kohonen (1996) 3rd Intl. Conf. on Neur. Netw. in the Capit. Markets, Fig. 2)

(a) Life expectancy at birth; (b) Adult illiteracy; (c) Share of food in houshold consumption; (d) Share of medical care in houshold consumption; (e) Population per physician; (f) Infant mortality rate; (g) Tertiary education enrollment; (h) Share of the lowest-earning 20 percent in the total household income. White-large, black-low values.

3/5

Figure: (Kaski and Kohonen, 1996, Fig. 2)

SOM Analysis of the World Welfare

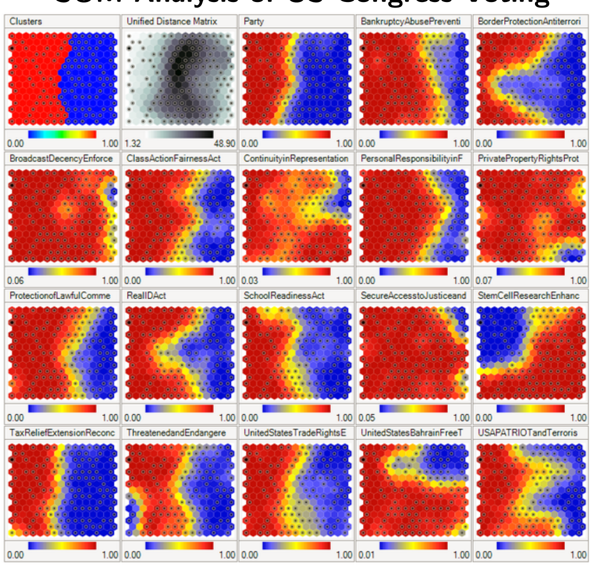
Figure: (Kaski & Kohonen (1996) 3rd Intl. Conf. on Neur. Netw. in the Capit. Markets, Fig. 3)

Distribution of GNP per Capita, which was not used during training.

4/5

Figure: (Kaski and Kohonen, 1996, Fig. 3)

SOM Analysis of US Congress Voting



5/5 Figure: (Wikimedia Synapse_Self-Organizing_Map.png 2010-11-29, © CC BY-SA 2.5, URL)

This is an analysis of the voting bahavior of the US Congress with a SOM. All panels show the square network with its internal hexagonal structure. The input vectors where yes/no (0/1) vectors of the voting behavior of members of the congress on different subjects. The first panel shows a clustering of the network nodes into two classes. The second panel shows a local measure of distance between nodes. All other panels show component planes, i.e. the values of a single component of the weight vector associated with the nodes. For the third panel red and blue indicate party membership for Republicans and Democrats, respectively. For all other panels red and blue indicate a yes- and no-vote, respectively.
Figure: (Wikimedia Synapse_Self-Organizing_Map.png 2010-11-29, © CC BY-SA 2.5, URL)[2.1]

References

Kaski, S. and Kohonen, T. (1996). Exploratory data analysis by the self-organizing map: Structures of welfare and poverty in the world. In *Neural networks in financial engineering. Proceedings of the third international conference on neural networks in the capital markets, London, England, 11-13 October 1095*, pages 498–507.

Wikipedia (2017). Self-organizing map. Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Self-organizing_map accessed 2017-01-28.

Notes

2.1 Wikimedia Synapse_Self-Organizing_Map.png 2010-11-29, © CC BY-SA 2.5, https://commons.wikimedia.org/wiki/File:Synapse_Self-Organizing_Map.png