COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS
DEPARTMENT OF APPLIED INFORMATICS

Introduction to Artificial Intelligence
Exercise 11 – SOM: Self-Organizing Map

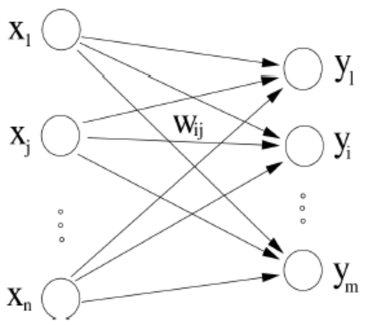
December 4, 2025

11. SOM: SELF-ORGANIZING MAP

We will implement another model of a neural network: SOM (Self-Organising Map), which is able to approximate the shape of input data.

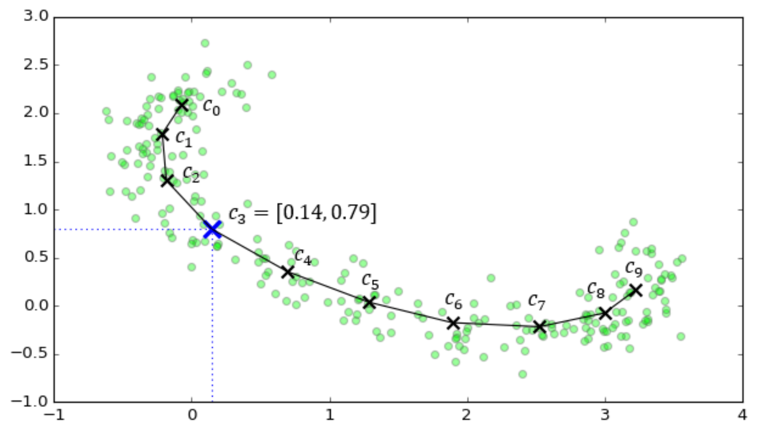
Model:

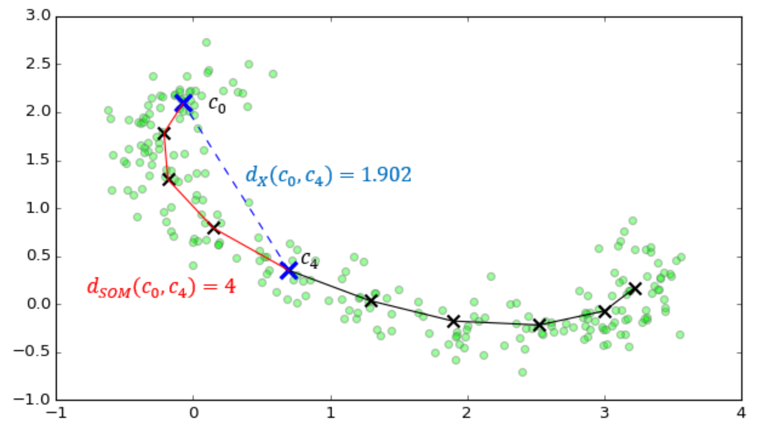
SOM is a single layer network, the number of inputs n depends on data (in our case 2D data => 2 input neurons) and the number of output neurons m is arbitrary, so the weight matrix $\boldsymbol{W}$ is of shape $m \times n$.



This network is different from the standard single layer perceptron. The first difference is that individual neurons are not independent, but they form a structure – grid: in our case it is a 1D grid, so kind of a "snake".
Every neuron has its respective weight vector $w_i = \boldsymbol{W}_{i,:}$. This vector has n components (based on the input dimensionality) and determines the position of the neuron in the input space. This position is also called the neuron centre: $c_i = w_i$. For example neuron no.3 has position $\boldsymbol{W}_{3,:} = w_3 = c_3 = [0.14, 0.79]$.
Every neuron has its position in the input space and its position in the SOM grid. Therefore, we distinguish two types of distances between neurons:

1. $d_X(c_i, c_j)$ is the distance in the input space, i. e. standard Euclidean distance $||c_i - c_j||$
2. $d_{SOM}(c_i, c_j)$ is the distance in the grid, in our case of 1D grid it is just the difference between indices $|i - j|$.





Similarly as for two centres, distance in the input space can be also computed between a centre and an input vector x : $d_X(c_i, x) = ||x - c_i||$.

Network training:

Unlike perceptron, which is trained using supervised learning, SOM is trained unsupervised, so there are no labels d or training/testing error e. Learning is based only on the input data. Learning parameters α_t (learning rate) and λ_t (neighbourhood coefficient) for SOM are changed in time, values in the first epoch are α_s, λ_s and they are gradually decreased down to α_f, λ_f in the last epoch. Gradual change is achieved according to the equations:

$$\alpha_t = \alpha_s . \left(\frac{\alpha_f}{\alpha_s}\right)^{\frac{t}{t_{max}-1}} \qquad \lambda_t = \lambda_s . \left(\frac{\lambda_f}{\lambda_s}\right)^{\frac{t}{t_{max}-1}}$$

To capture the interaction between neighbouring neurons i. e. how much a change in neuron i^* affects a different neuron i, we use the neighbourhood function. This function returns a number close to 1 for two neurons close to each other in the grid, and a number close to 0 for neurons that are far apart.

$$h(i^*, i) = exp\left(-\frac{d_{SOM}(i^*, i)^2}{\lambda_t^2}\right)$$

The learning runs through multiple, fixed number of epochs, in each epoch we iterate through all inputs x in a random order. For each input x:

- find the winner neuron: $i^* = argmin_i ||x - w_i||$

- adjust weights of **all** neurons: $w_i = w_i + \alpha_t.h(i^*, i).(x - w_i)$

Alternatives: In different implementations, some parts of SOM may be altered a bit. There exist different functions to compute α_t, λ_t, neighbourhood function $h(i^*, i)$, grid distance $d_{SOM}(c_i, c_j)$, etc.

Dimensionality reduction:

One of the practical uses of SOM is the dimensionality reduction of the input data, for example for visualisation in 2D or classification. Data may have 20 or even more dimensions, SOM grid is 2D or 3D. Our data have 2 dimensions, so the only sensible reduction is from 2D to 1D. Then we could for example classify the data into four classes according to colours.
The program has prepared dimensionality reduction using SOM (for which you need to finish a couple of functions) and also using PCA, so you can see their comparison.

Assignment: Finish function *find_winner(x)*, in the prepared code, which takes input x, finds the closest neuron and returns its **index**, and finish the *train(...)* function which trains the SOM. Weights initialisation is already prepared, and there is also some visualisation of the learning process and the results.