Universität Hamburg
Department Informatik, ABe NATS, WTM & TAMS
Wolfgang Menzel, Stefan Wermter, Jianwei Zhang

Kontakt: Cornelius Weber
Raum F-233
Tel 42883-2537

Exercises to the module: Algorithmisches Lernen

SS 2012 Sheet 6

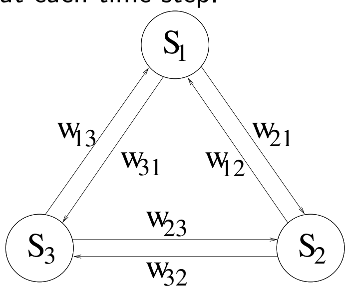
Due: 16.05.2012

Task 6.1 Tiny Hopfield Network

Consider an attractor network with three neurons $i = 1, 2, 3,$. The activation dynamics is:

$$S_i(t+1) = \text{sign}\big(\sum_j w_{ij} S_j(t) \, - \, \theta_i\big) \quad , \qquad \text{sign}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \tag{1}$$

Let there be no self-connections ($w_{ii} = 0$) and zero threshold ($\theta_i = 0$). The neurons are asynchronously updated, i.e. one neuron is updated at each time step.



The following patterns exist for this network:

Muster	S_1	S_2	S_3
$\vec{\sigma}_1$	-1	-1	-1
$\vec{\sigma}_2$	+1	-1	-1
$\vec{\sigma}_3$	-1	+1	-1
$\vec{\sigma}_4$	+1	+1	-1
$\vec{\sigma}_5$	-1	-1	+1
$\vec{\sigma}_6$	+1	-1	+1
$\vec{\sigma}_7$	-1	+1	+1
$\vec{\sigma}_8$	+1	+1	+1

(a) Let us store the pattern $\vec{\sigma}_7 = (-1, 1, 1)$ using the Hebbian learning rule. Which are the resulting weights w_{ij}?

(b) Show that pattern $\vec{\sigma}_7$ is stable.

(c) Calculate the energy

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} S_i S_j \; + \sum_i \theta_i S_i$$

for each of the 8 possible states. For which state(s) is the energy minimal?

— see reverse —

Wolfgang Menzel
(menzel@informatik. . .)
Stefan Wermter
(wermter@informatik. . .)
Jianwei Zhang
(zhang@informatik. . .)

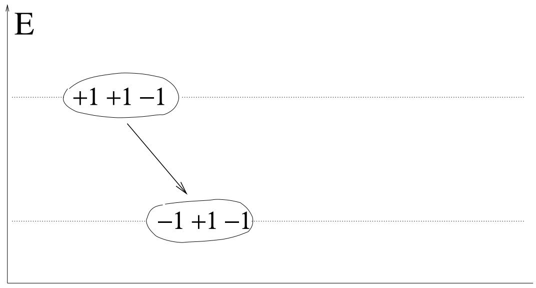
Vorlesung:
Mi 8:15-11:45, F-009
Übungen:
Mi 12:15-13:45, F-009

Dienstgebäude
Universität Hamburg
Vogt-Kölln-Str. 30
22527 Hamburg

Öffentliche Verkehrsmittel
Buslinien 181, 281
Haltestelle: Informatikum

(d) Now we wish to prefer state $\vec{\sigma}_7$ over its "inverse" $\vec{\sigma}_2$, that is, $\vec{\sigma}_7$ shall have a lower energy. To this end we apply a threshold $\theta_1 = 1$ to neuron 1. Again, calculate the energy E for every state.

(e) Draw the energy as a function of the state into a diagram of the following form.



Mark all state transitions, which are possible following Eq. 1, by an arrow.

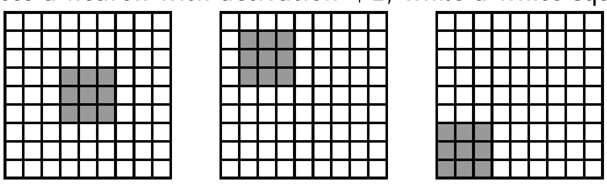
(f) The Hopfield network shall now store the following four patterns: $\vec{\sigma}_1, \vec{\sigma}_4, \vec{\sigma}_6, \vec{\sigma}_7$ (see table). Show that this is impossible — even when using a threshold.

Tip: In the 2-dim input space of the 3rd neuron, draw the inputs that belong to the four patterns, and their target output. Are the two regions where the output should be $+1$ or -1, respectively, linearly separable?

(g) How can the addition of a fourth neuron make the four given patterns stable?

Task 6.2 Large Hopfield Network

Regard an attractor network in which such kind of patterns as shown in the figure are stable. In the figure, a grey square denotes a neuron with activation $+1$, while a white square denotes activation -1.



(a) How do the weights of any given neuron look like qualitatively? How many non-zero weights are required (a threshold my be used)?

(b) How do we need to change the weights if we wish to achieve that the hill of activation moves one column to the right from one time step to the next?

Task 6.3

In the following exercise sheets 7 and 8 you will investigate either an Elman network (an elaboration of an MLP) or a SOM — dependent on your choice. Each network type should be dealt with at least by one group.

To this end, check out some software package and run any example of an MLP or a SOM, for example, from a tutorial. You may choose any package, such as the tool that you have used in the first part of the lecture, or software listed on exercise sheet 5, or the Python code of the Marsland book.

Describe your experience (installation, handbook, GUI, transparancy of the code, etc.).