Theoretical Neuroscience II
Computational Neuroscience II
Associative Memory
(10 points)

Until now our investigations focused on analyzing the behaviour of networks with respect to stability and fix points. Within this exercises of auto associative memory we will follow the reverse approach: we introduce an algorithm which creates for a given set of activity patterns, which shall be memorized, an auto associative network with a corresponding set of discrete fix points. These kind of networks not only provide the ability to simply recall the stored patterns but also show great capabilities in pattern completion/correction, which means that partial/incomplete representations of the patterns can be used to recall the corresponding prototypical full pattern.
For getting a deeper insight, please make yourself familiar with the chapter *Associative Memory* in the book*.

Assignments:

1. (2p) In order to reproduce the example from the book, please study the provided Matlab code (Memory.m), which implements an auto associative memory by applying the formulas from the text. The corresponding parameters are:

   $c = 18.731,\ N_v = 50,\ \alpha = 0.25,\ \lambda = 1.25$ and n being a vector of N_v ones

   Further, $M = \frac{\lambda \sum_{i=1}^{Nmem} (v^i - \alpha c n)(v^i - \alpha c n)^T}{c^2 \alpha N_v (1-\alpha)} - \frac{nn^T}{\alpha N_v}$ and $F(I) = 150 \left[tanh \left(\frac{I+20}{150} \right) \right]_+$

   Show that $c = 18.731$ fulfills the requirements of equation (7.49):
   $F(-c(1+\alpha\lambda)) = 0$ and $c = F(c(\lambda - 1 - \alpha\lambda))$

2. (2p) Your next task is to adapt the $N_{mem} = 4$ patterns v^i which shall later be memorized according to the example. Each pattern consists of N_v components where $N_v\alpha$ of the units show activity c, while the remaining ones are inactive with zero activity. Until now, the code generates 4 of such sparse patterns with randomly distributed activity. Change the $1st$ pattern in a way that now units 18-30 being active, and for the $2nd$ pattern every $4th$ unit being active. Patterns 3 and 4 can remain as they are. Check whether the network is able to reproduce patterns 1 and 2 qualitatively. Use the variable $vector$ in the test section of the code to select the desired test pattern.

3. (6p) Investigate the behavior of the auto associative memory in terms of its pattern completion/correction ability. First, replace the random patterns 3 and 4 by two new static patterns according to your own taste. Make sure that they are sparse with respect to activity c. Then, find out whether your network is able to reproduce all 4 patterns (like in the previous task). If this is not the case, they are probably too similar. Hence, make them more unique by adapting the activity patterns until the network is able to recall all patterns.
   For each of the 4 patterns investigate empirically the robustness against noise/partial input for 50 time steps by applying $v_c^{t+1} = F(Mv_c^t)$: by changing the value of the variable *strength* in the test section of the code you are able to distort a selected test pattern by adding Gaussian distributed noise. The higher the value of $strength$ the higher the distortion will be. Zero means no distortion.
   Now, for each pattern and each value of $strength$ repeat the experiment 10 times and notate whether the output resembles the correct pattern or not. For instance, if pattern 1 was distorted and the output activities equal qualitatively those of the original pattern 1 then it was mapped correctly onto its attractor ($v_c^\infty = v^i$). Start with $strength = 0$ in steps of 0.5 until your success level gets worse than 50%. Visualize your results in appropriate plots!

4. **(6p) Optional task:** Describe possible influences of α. Which value for α theoretically allows to create the maximum number of different activity patterns? By adding new patterns investigate the memory's capacity, i.e. the number storable of patterns.

* Theoretical Neuroscience
P. Dayan and L. F. Abbott
MIT Press
ISBN: 978-0-262-04199-7
pages 261-265