Università degli Studi di Padova

Dreaming Hopfield Networks

Giancarlo Saran Gattorno
Università degli Studi di Padova, Dipartimento di Fisica e Astronomia

January 21, 2025

Overview

1. How can a Hopfield network sleep? Unlearning and reinforcement
2. Local Hebbian learning and Dreaming sleep algorithm
3. Daydreaming sleep algorithm and correlated patterns
4. Final considerations and future development

Motivation

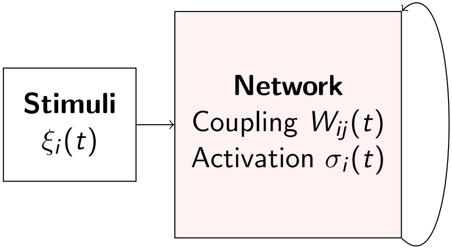
Theorem (Storage bound [Gardner, 1988])

For a symmetric network the maximum critical storage capacity with respect to uncorrelated patterns is $\alpha_c = 1$

- Hopfield Network with Hebbian weights has suboptimal storage ($\alpha_c = 0.138$).
- Could be improved by implementing some form of reinforcement of useful memories and deletion of unimportant ones
- In fact, this is what happens to mammals during sleep (REM and SW phases)

Locality

- A model of associative memory has to be local in order to be biologically plausible
- There has to be a learning phase depending only on the activations and the patterns



$$\Delta W_{ij}(t+1) = \phi(\xi(\tau \leq t), \sigma(\tau \leq t))$$

Local Hebbian algorithm

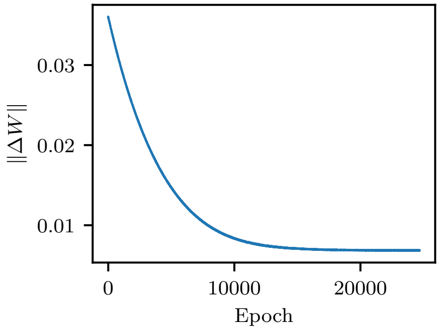


Figure: Hebbian online training curve. The opnorm of the change in weights goes to zero after convergence (all patterns have been learned)

To make Hebbian weights the result of a local learning rule:

- Present slightly noisy versions of the patterns (stimuli) to the network repeatedly as external field
- Update the weigths according to the Hebbian rule (neurons that fire together wire together)

Algorithm 1 Local Hebbian algorithm

Require: Weight matrix $\mathbf{W}$, patterns $\{\vec{\xi^{\mu}}\}$, learning rate η, threshold $\vec{\theta}$, flip fraction p_{flip}, stimulus strength λ_{stim}, weight decay α

$\vec{\sigma} \in \{-1, 1\}^N$ ▷ Initial network activation
while not converged **do** ▷ If update curve flattens or all patterns are retrieved
  Select random noisy pattern $\xi^{\mu}(p_{flip})$
  for $t = 1$ **to** T **do** ▷ Average over T activations
    for each neuron i **do** ▷ Async update
      $h_i \leftarrow \sum_j W_{ij}(\sigma_{t-1})_j + \xi_i^{\mu}(p_{flip})$
      $(\sigma_t)_i \leftarrow \text{sign}(h_i + \theta_i)$
    end for
    $(S_t)_{ij} \leftarrow (\sigma_t)_i(\sigma_t)_j$ ▷ Store correlation
  end for
  $W_{ij} \leftarrow W_{ij} + \frac{\eta}{N^2}\sum_t (S_t)_{ij} - \alpha W_{ij}$ ▷ Hebbian + weight decay update
end while
  return $\mathbf{W}$

Local Hebbian learning

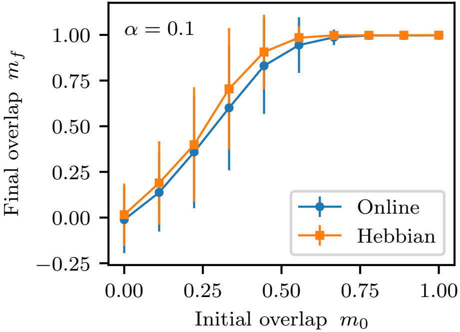


Figure: Retrieval map with subcritical load comparing Hebbian weights with weights trained for 25k epochs.

- Symbolizes the network being awake (the network sees various noisy representations of the patterns and learns them)
- Very sensitive to parameter tuning. Specifically η, T and λ_{stim}
- Finally converges to the correct weight matrix $W_{ij} = \dfrac{1}{N}\sum_{\mu=1}^{P} \xi_i^{\mu}\xi_j^{\mu}$

Dreaming algorithm

- Proposed [Fachechi et al., 2018] to achieve optimal storage of uncorrelated patterns
- Issue, already noted by Hopfield, is mixed memories are way more than pure ones. They should be unlearned
- Many modified Hamiltonians (or equivalently coupling rules) have been proposed over the years. In this implementation coupling depends on the sleep duration t and pattern correlation matrix $C_{\mu\nu} = \frac{1}{N}\sum_{i=1}^{N} \xi_i^\mu \xi_i^\nu$

$$W_{ij}(t) = \frac{1}{N} \sum_{\mu\nu} \xi_i^\mu \xi_j^\nu \left(\frac{1+t}{1+tC} \right)_{\mu\nu} \tag{1}$$

Dreaming algorithm

- The coupling can be obtained as a result of an iterative (but non local) learning rule with Hebbian initial conditions
- The discrete learning rule depends on unlearning rate ϵ

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \frac{\epsilon}{1+\epsilon t}[\mathbf{W}(t) - \mathbf{W}(t)^2] \tag{2}$$

- ϵ has to be tuned, there are critical values above which unlearning fails
- If the weight matrix is not normalized every few steps it can diverge

Daydreaming algorithm

More recently, another sleep algorithm [Serricchio et al., 2024] was proposed, with several improvements:

- No parameter tuning needed
- No initial assumptions on the weight matrix beyond symmetry
- Fully local
- Works well with correlated patterns

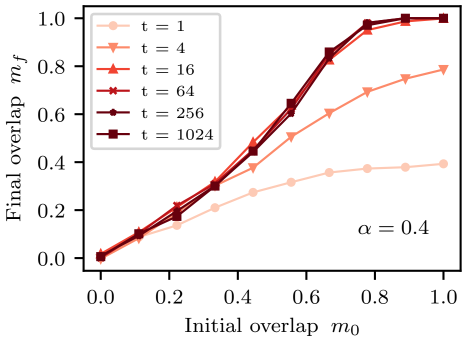


Figure: Retrieval map evolution during sleep. Even with supercritical load all pure memories become stable. Simulated with $N = 250$

Algorithm 2 Daydreaming

Require: Weight matrix $\mathbf{W}$, patterns $\{\vec{\xi}^{\mu}\}$, inverse learning rate τ

for $t = 1$ **to** T **do**
    for each neuron σ_i **do**
        Select random pattern $\vec{\xi}^{\mu}$
        Initialize randomly $\sigma_i \in \{-1, 1\}$
        Evolve until fixed point
        $W_{ij} \leftarrow W_{ij} + \frac{1}{N\tau}(\xi_i^{\mu}\xi_j^{\mu} - \sigma_i\sigma_j)$
    end for
    Normalize $\mathbf{W}$
end for
return $\mathbf{W}$

Daydreaming algorithm

- The two sleep rules give equivalent results on uncorrelated data
- The finite size of the network means α_c is slightly smaller than the theoretical value of 1
- At $\alpha = 1$ some pure memories become locally unstable

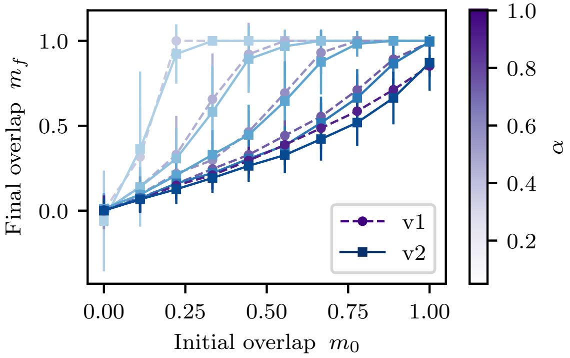


Figure: Simulation of Hebbian learning + dreaming (v1) and daydreaming (v2) with $N = 250$ at different loads.

Correlated data

- Previous consolidation algorithms were only applied on uncorrelated patterns
- Using the random feature model we can construct a correlated dataset from uncorrelated features $f_i^k \sim \mathcal{U}(\{-1, 1\})$ and a random matrix $c_{k\mu} \sim \mathcal{N}(0, 1)$

$$\xi_i^\mu = \operatorname{sign}\left(\sum_{k=1}^{D} c_k^\mu f_i^k\right) \tag{3}$$

- Another parameter controls correlation and phase behavior $\alpha_D = D/N$. At high α_D the data is uncorrelated

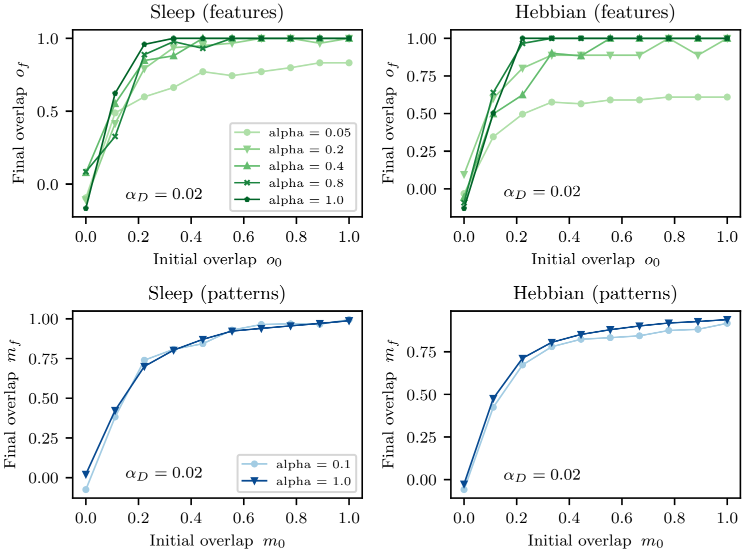


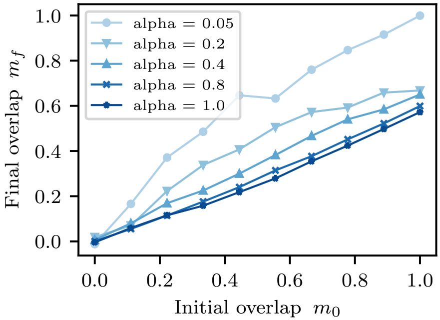
Figure: Simulation with correlated data and $N = 250$.

Correlated data

- At very high correlation (only 5/250 features) patterns are learned even at high load
- What is interesting is after many representations of patterns generated by the features are presented (high load, high correlation) features become attractors
- Daydreaming performs only slightly better than Hebbian at this correlation, but the discrepancy is expected to increase with α_D

Spurious patterns

- Spurious patterns are locally stable in Hebbian Hopfield networks
- We test only the 3-combinations $\psi_i^{\mu\nu\kappa} = \pm \text{sign}(\xi_i^\mu + \xi_i^\nu + \xi_i^\kappa)$
- At very low load the spurious patterns are locally stable but have a vanishing basin, this can be attributed to the finite size of the simulation



Conclusion

- "Sleeping" allows Hopfield networks to reach optimal performance by consolidating pure memories and weakening mixed states
- Hopfield networks can be used for feature learning. This makes them applicable on real correlated data
- In the paper this property was tested on MNIST
- Feature learning capabilities exploited to perform digit classification. The network learns combinations of images of the same digit (prototypes), similarly to features in the random feature model

Future development

- For both correlated and uncorrelated memories only zero temperature was analyzed
- More complete analysis of phase behavior (adding nonzero temperature)
- Study of feature retrieval in supercritical (high load) phase for non Hebbian couplings
- Use the network on more complex datasets

References

Fachechi, A., Agliari, E., and Barra, A. (2018).
Dreaming neural networks: forgetting spurious memories and reinforcing pure ones.
CoRR, abs/1810.12217.

Gardner, E. (1988).
The space of interactions in neural network models.
Journal of Physics A: Mathematical and General, 21(1):257.

Serricchio, L., Bocchi, D., Chilin, C., Marino, R., Negri, M., Cammarota, C., and Ricci-Tersenghi, F. (2024).
Daydreaming hopfield networks and their surprising effectiveness on correlated data.