

# Improved Robustness and Hyperparameter Selection in the Modern Hopfield Network

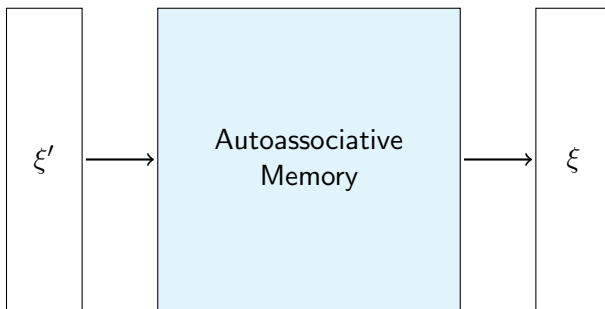
Hayden McAlister

School of Computing, University of Otago

Supervisors: Anthony Robins, Lech Syzmanski

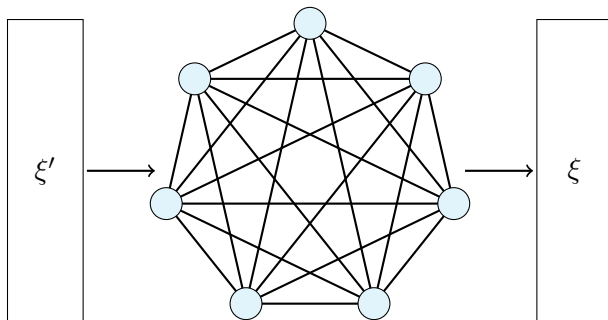
# Autoassociative Memories

- Learn to associate a state with itself.
- Relax probe towards a learned state.



# Autoassociative Memories

- Learn to associate a state with itself.
- Relax probe towards a learned state.



# Classical Hopfield Network

- Association by Hebbian learning.
  - Biological inspiration.
  - Easy to analyze.
- Relax by matrix multiplication.
  - Mean field approximation.
  - Nonlinearity keeps states in bipolar domain.
  - Energy guaranteed to achieve a minima (under sensible conditions).

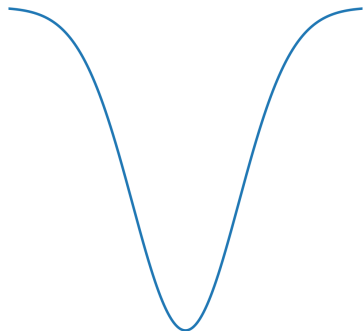
$$W = \sum_k \xi_k \otimes \xi_k \quad (1)$$

$$\xi_{t+1} = \text{Sign}(W \cdot \xi_t) \quad (2)$$

$$E(\xi) = -\frac{1}{2} \xi^T W \xi \quad (3)$$

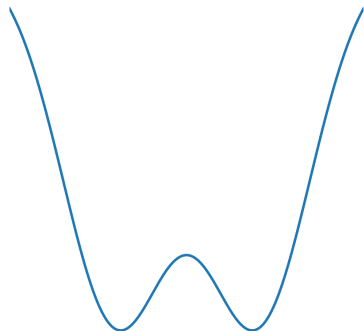
# Modern Hopfield Network

- Classical energy wells are too shallow.



# Modern Hopfield Network

- Classical energy wells are too shallow.



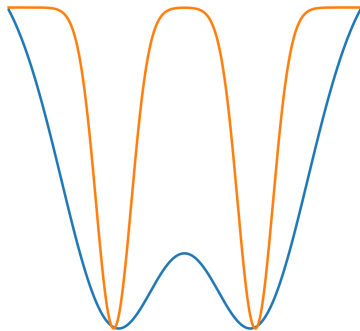
# Modern Hopfield Network

- Key trick: Replace quadratic energy with general polynomial.
  - Heck, anything with a vaguely polynomial shape.

$$f_n(x) = x^n$$

$$f_n(x) = \begin{cases} x^n & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

$$f_n(x) = \begin{cases} x^n & \text{if } x \geq 0 \\ -\epsilon x & \text{if } x < 0 \end{cases}$$

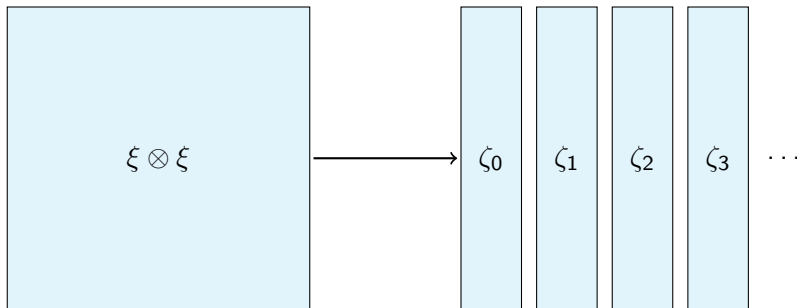


# Modern Hopfield Network

## $n$ – The Interaction Vertex

- Controls the range of influence that memories have.
- However, also radically alters the network architecture.

Memory matrix replaced by list of memory states – vectors of same dimension as data.





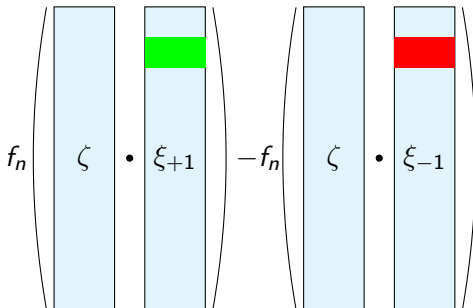
# Modern Hopfield Network

## $n$ – The Interaction Vertex

- Controls the range of influence that memories have.
- However, also radically alters the network architecture.

Relaxation no longer uses mean field – now a contrastive difference.

- Negative energy no longer means “stable” – the energy *difference* between a neuron clamped on and off indicates stability.



# Modern Hopfield Network

## $n$ – The Interaction Vertex

- Controls the range of influence that memories have.
- However, also radically alters the network architecture.

Learning no longer supports Hebbian – now requires gradient descent.

$$W = \sum_k \xi_k \otimes \xi_k$$

$$\text{Loss}(\xi) = \tanh \left[ \beta \sum_{\mu} \left( f_n \left( \zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) - f_n \left( -\zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) \right) \right]$$

# Properties – Network Capacity

Larger network capacities with higher interaction vertices:

$$K_{\max} = \frac{1}{2(2n-3)!!} \frac{N^{n-1}}{\ln(N)} \quad (4)$$

Notably, super-linear for  $n > 2$ .

# Properties – Training Times

Faster training times with higher interaction vertices:

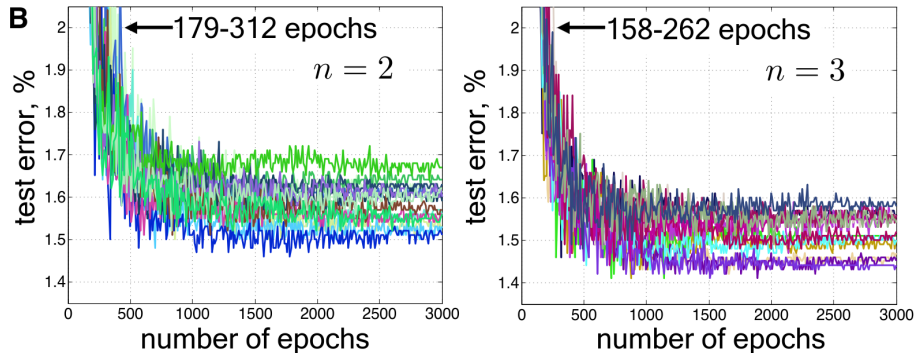


Figure: Krotov and Hopfield 2016, Figure 01

# Properties – Feature to Prototype Transition

Low interaction vertices result in memories that look like features, while higher interaction vertices result in memories that look like prototypes:

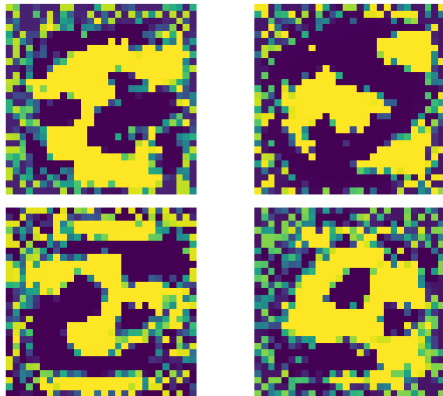


Figure: Feature-like Memories,  $n = 2$

# Properties – Feature to Prototype Transition

Low interaction vertices result in memories that look like features, while higher interaction vertices result in memories that look like prototypes:

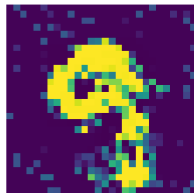
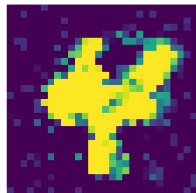
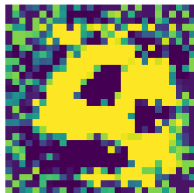
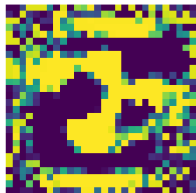
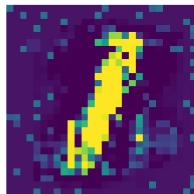
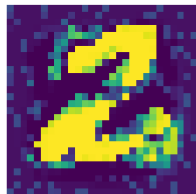
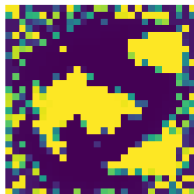
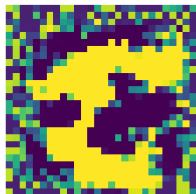


Figure: Feature-like Memories,  $n = 2$

Figure: Prototype-like Memories,  $n = 20$

# Properties – Feature to Prototype Transition

Low interaction vertices result in memories that look like features, while higher interaction vertices result in memories that look like prototypes:

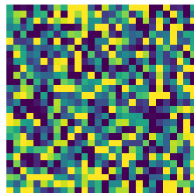
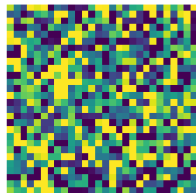
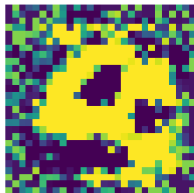
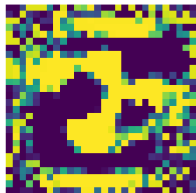
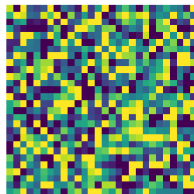
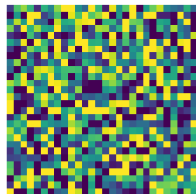
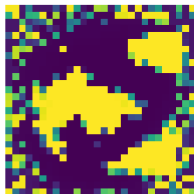
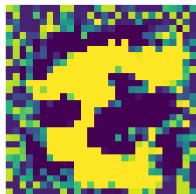


Figure: Feature-like Memories,  $n = 2$

Figure: Prototype-like Memories,  $n = 20$

# Network Dynamics Step by Step

$$\tanh \left[ \beta \sum_{\mu} \left( f_n \left( \zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) - f_n \left( -\zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) \right) \right]$$

- 1 Calculate similarities  $\zeta \cdot \xi_{+1}$ ,  $\zeta \cdot \xi_{-1}$
- 2 Pass similarities through interaction function  $f_n$
- 3 Sum the result over all memories  $\sum_{\mu}$
- 4 Multiply by a scaling factor  $\beta$
- 5 Pass through activation function (e.g. Sign or tanh)



# Network Dynamics Step by Step

$$\tanh \left[ \beta \sum_{\mu} \left( f_n \left( \zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) - f_n \left( -\zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) \right) \right]$$

- 1 **Calculate similarities**  $\zeta \cdot \xi_{+1}, \zeta \cdot \xi_{-1}$
- 2 Pass similarities through interaction function  $f_n$
- 3 Sum the result over all memories  $\sum_{\mu}$
- 4 Multiply by a scaling factor  $\beta$
- 5 Pass through activation function (e.g. Sign or tanh)

$$\zeta, \xi \in [-1, 1]^N \implies \zeta \cdot \xi \in [-N, N]$$

# Network Dynamics Step by Step

$$\tanh \left[ \beta \sum_{\mu} \left( f_n \left( \zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) - f_n \left( -\zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) \right) \right]$$

- ① **Calculate similarities**  $\zeta \cdot \xi_{+1}, \zeta \cdot \xi_{-1}$
- ② **Pass similarities through interaction function**  $f_n$
- ③ **Sum the result over all memories**  $\sum_{\mu}$
- ④ **Multiply by a scaling factor**  $\beta$
- ⑤ **Pass through activation function** (e.g. Sign or tanh)

$$\zeta, \xi \in [-1, 1]^N \implies \zeta \cdot \xi \in [-N, N]$$
$$f_n(\zeta \cdot \xi) = (\zeta \cdot \xi)^n$$

# Network Dynamics Step by Step

$$\tanh \left[ \beta \sum_{\mu} \left( f_n \left( \zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) - f_n \left( -\zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) \right) \right]$$

- 1 **Calculate similarities**  $\zeta \cdot \xi_{+1}, \zeta \cdot \xi_{-1}$
- 2 **Pass similarities through interaction function**  $f_n$
- 3 Sum the result over all memories  $\sum_{\mu}$
- 4 Multiply by a scaling factor  $\beta$
- 5 Pass through activation function (e.g. Sign or tanh)

$$\begin{aligned} \zeta, \xi \in [-1, 1]^N &\implies \zeta \cdot \xi \in [-N, N] \\ f_n(\zeta \cdot \xi) &= (\zeta \cdot \xi)^n \\ &= N^n \end{aligned}$$

# How large is too large?

Network parameters	Interaction function value
$N = 100, n = 2$	$10^4$
$N = 100, n = 5$	$10^{10}$
$N = 100, n = 10$	$10^{18.89}$
$N = 100, n = 20$	$10^{40}$

# How large is too large?

Network parameters	Interaction function value
$N = 100, n = 2$	$10^4$
$N = 100, n = 5$	$10^{10}$
$N = 100, n = 10$	$10^{18.89}$
$N = 100, n = 20$	$10^{40}$

Maximum value of a float32 is  $\approx 3.4 \cdot 10^{38}$ .

# How large is too large?

Network parameters	Interaction function value
$N = 100, n = 2$	$10^4$
$N = 100, n = 5$	$10^{10}$
$N = 100, n = 10$	$10^{18.89}$
$N = 100, n = 20$	$10^{40}$

Maximum value of a float32 is  $\approx 3.4 \cdot 10^{38}$ .

Maximum value of a float64 is  $\approx 1.8 \cdot 10^{304}$ .

# Fixing the Problem

$$\tanh \left[ \beta \sum_{\mu} \left( f_n \left( \zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) - f_n \left( -\zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) \right) \right]$$

# Fixing the Problem

$$\tanh \left[ \beta \sum_{\mu} \left( f_n \left( \zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) - f_n \left( -\zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) \right) \right]$$



# Fixing the Problem

$$\tanh \left[ \sum_{\mu} \left( \beta f_n \left( \zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) - \beta f_n \left( -\zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) \right) \right]$$

# Fixing the Problem

$$\tanh \left[ \sum_{\mu} \left( f_n \left( \beta \left( \zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) \right) - f_n \left( \beta \left( -\zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) \right) \right) \right]$$

If  $f_n$  is homogenous:  $f(\alpha x) = \alpha^k f(x)$ .

Weaker than linear – includes Polynomial and Rectified Polynomial.

# Fixing the Problem

$$\tanh \left[ \sum_{\mu} \left( f_n \left( \frac{\beta}{N} \left( \zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) \right) - f_n \left( \frac{\beta}{N} \left( -\zeta_{\mu,i} + \sum_{j \neq i} \zeta_{\mu,j} \xi_j \right) \right) \right) \right]$$

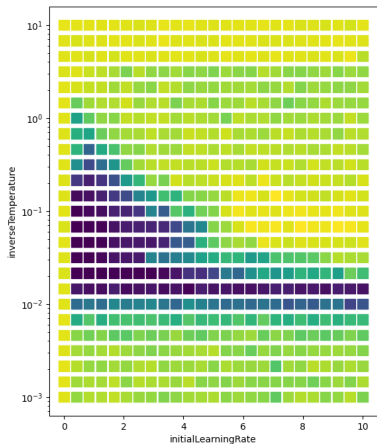
If  $f_n$  is homogenous:  $f(\alpha x) = \alpha^k f(x)$ .

Weaker than linear – includes Polynomial and Rectified Polynomial.

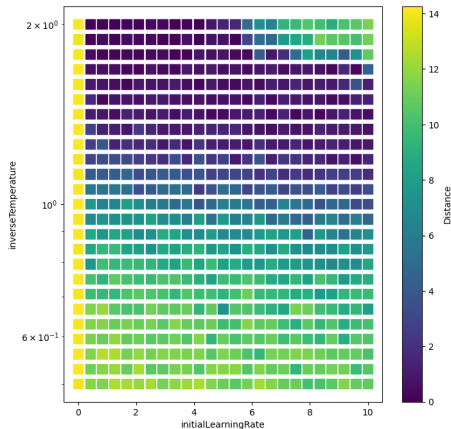
Provably doesn't alter network properties (e.g. capacity)

# Results of Modifications – Autoassociative Memory

$$n = 2$$



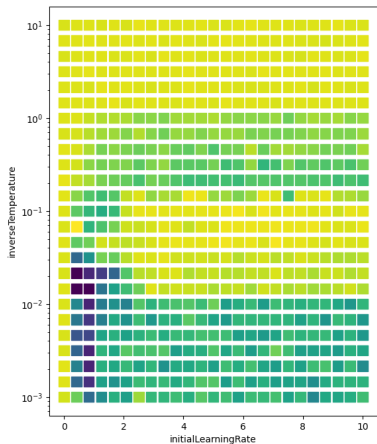
Original



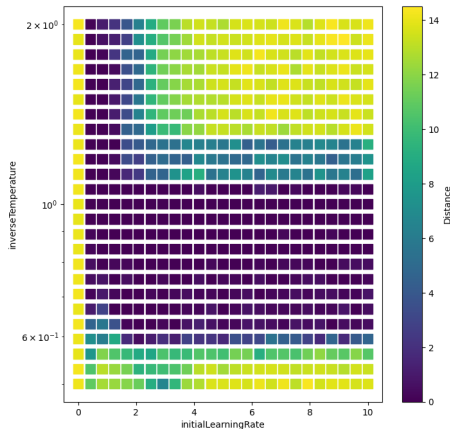
Modified

# Results of Modifications – Autoassociative Memory

$$n = 10$$



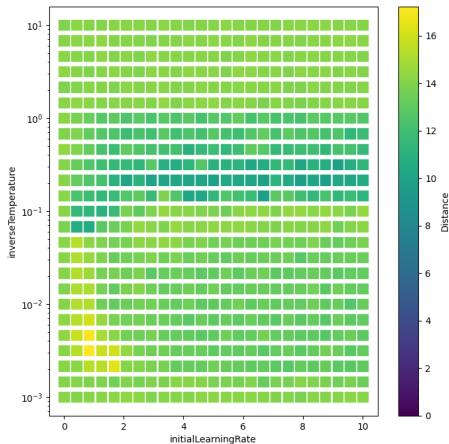
Original



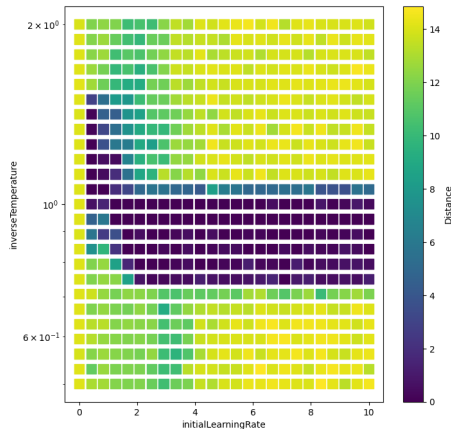
Modified

# Results of Modifications – Autoassociative Memory

$$n = 20$$



Original



Modified

# Conclusion

- Modern Hopfield Network generalizes Classical Hopfield Network.
- Network capacity increases with the interaction vertex, even super-linearly.
- The original network (Krotov and Hopfield, 2016) has very unstable behavior for larger interaction vertices.
- The original network also has wildly shifting optimal hyperparameter regions.
- Our modifications solve both the instability and shifting hyperparameter regions at no additional cost.