How Theta Oscillations Can Train Neural Networks and Punish Competitors

Kenneth Norman, Ehren Newman, Greg Detre, & Sean Polyn

Department of Psychology and Center for the Study of Brain, Mind, and Behavior, Princeton University

Princeton Computational Memory Lab

Summary

We present a new learning algorithm that leverages oscillations in the strength of neural inhibition to train neural networks.

Raising inhibition can be used to identify weak parts of target memories, which are then strengthened (by increasing weights into those units).

Conversely, lowering inhibition can be used to identify competitors, which are then punished (by reducing weights into those units).

We use the learning rule to account for behavioral data regarding how competition at retrieval affects subsequent memory. We also show that the learning algorithm's capacity for storing patterns increases steadily as a function of network size, and that the learning algorithm can memorize large numbers of correlated patterns without collapsing.

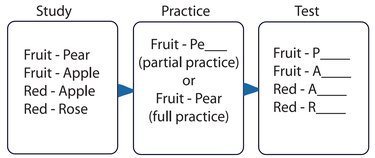
Finally, we relate this work to neural data on theta oscillations and learning.

Background: Competitors are Punished

Our original motivation for this research was to model data on competitive dynamics and memory. Across several domains, researchers have found that competitors are punished during memory retrieval.

More specifically: When a representation is activated by a retrieval cue, but that representation loses the competition to be retrieved, it suffers a lasting decrease in accessibility (on the order of hours and possibly longer).

This principle is illustrated very nicely by Michael Anderson's work on **retrieval-induced forgetting**, illustrated below (see Levy & Anderson, 2002, for a review).



Recall after practice, relative to baseline

Test Item	After Partial Practice (Fruit - Pe___)	After Full Practice (Fruit - Pear)
Fruit - P(ear)	BETTER	BETTER
Fruit - A(pple)	WORSE	SAME
Red - A(pple)	WORSE	SAME
Red - R(ose)	SAME	SAME

In other words, if given a partial practice -
- Recall of the practiced item improves (Fruit-Pear)
- Recall of competitors gets worse (Fruit-Apple), in a *cue-independent* fashion (Red-Apple)

and if given a full practice -
- Recall of the practiced item improves (Fruit-Pear)
- Other items are unaffected (Red-Rose)

Intuitive story: Partial practice affects "Apple" more than full practice because the cue is **more ambiguous** in the partial practice condition, which in turn leads to **more competition**. Because "Apple" **competes more strongly** in the partial practice condition (but still loses the competition), it accrues **more punishment**.

What are the brain mechanisms of competitor punishment? Existing accounts focus on the role of prefrontal cortex in resolving competition. These accounts help explain the **dynamics** of competition do not explain why competition has **lasting** effects on memory.

The goal of this research is to identify basic neural learning mechanisms that can:

- Account for retrieval-induced forgetting data, and other psychological findings showing how competitors are punished

- Train new patterns into the network

The goals are synergistic: The ability to push away competitors should help networks store new information.

Oscillation based learning rule (Norman, Newman, Detre, & Polyn, in preparation)

Extract structure of stored information with oscillations

Low inhibition to punish competitors

Oscillate between NORMAL - LOW - NORMAL inhibition (N-L-N)

Low inhibition = Less constraint on network activity
The network has more space to represent competitors

LOWERING inhibition lets the network identify competitors
(O'Reilly & McClelland, personal communication)

High inhibition to strengthen target

Oscillate between NORMAL - HIGH - NORMAL inhibition (N-H-N)

High inhibition = More constraint on network activity
Stress-test of target: Poorly supported units turn off, well-supported units remain active

RAISING inhibition lets the network identify weak parts of the target

Learn based on changes in activity

Changing activity during N-L-N = competitors popping up

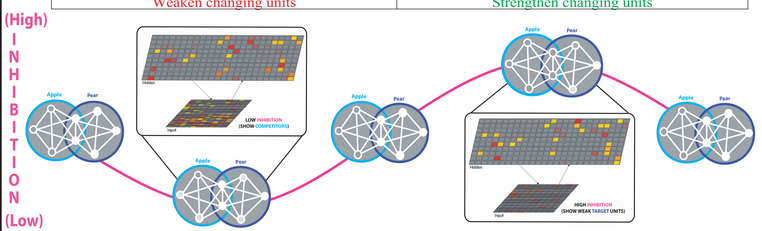
How to map changing activity to learning:
Inhibition decreases: Competitors become active
Therefore - increases in activity should trigger weakening
Inhibition returns to normal: Competitors back off
Therefore - decreases in activity should trigger weakening

Changing activity during N-H-N = target dropping out

How to map changing activity to learning:
Inhibition increases: Weak target units turn off
Therefore - decreases in activity should trigger strengthening
Inhibition returns to normal: Target turns back on
Therefore - increases in activity should trigger strengthening

INHIBITION APPLIED TO INPUT LAYER (CONSTRAINT)				
Normal	**Low**	**Normal**	**High**	**Normal**
Target on	Target on	Target struggles to stay on	Target comes back on	
Competitor allowed on	Competitor forced off	Competitor off	Competitor off	
Weaken changing units		Strengthen changing units		

(High) INHIBITION (Low)



Implementation

Only oscillate inhibition in the input layer

Input:
1. Calculate baseline inhibition to allow *k* active units
2. Add an oscillating component to this value

Hidden:
1. Calculate baseline inhibition to allow *k* active units
2. No oscillating component

Allow one full oscillation each trial

Compute weight change by applying the Contrastive Hebbian Learning rule (Movellan, 1990) **to successive time steps of the network (*t* and *t*+1).**

The sign of the learning rule changes as a function of the phase of the inhibitory oscillation (see equations to the right).

Weight changes are calculated at every time step, and applied at the end of each trial.

Learning rule as a function of phase of oscillation:
(Note: x_i = presynaptic neuron, y_j = postsynaptic neuron)

When inhibition is moving *away* from its midpoint:
Weight change = $lrate * ((x_i(t) * y_j(t)) - (x_i(t+1) * y_j(t+1)))$

Normal to Low Inhib: The rule weakens competing units that are coming on
Increases in receiving unit activation ($y_j(t+1) > y_j(t)$)
cause negative weight change from active senders

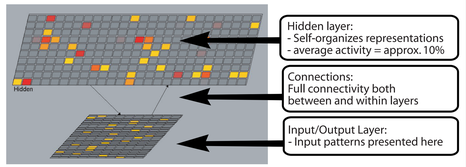
Normal to High Inhib: The rule strengthens target units that are turning off
Decreases in receiving unit activation ($y_j(t) > y_j(t+1)$)
cause positive weight change from active senders

When inhibition is moving *towards* its midpoint:
Weight change = $lrate * ((x_i(t+1) * y_j(t+1)) - (x_i(t) * y_j(t)))$

Low to Normal Inhib: The rule weakens competing units that are turning off
Decreases in receiving unit activation ($y_j(t+1) > y_j(t)$)
cause negative weight change from active senders

High to Normal Inhib: The rule strengthens target units that are coming on
Increases in receiving unit activation ($y_j(t+1) > y_j(t)$)
cause positive weight change from active senders

The Network



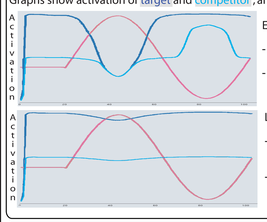
Retrieval-Induced Forgetting Simulations

Method

1. **Generate four patterns**
   A target pattern (presented at study and practice)
   A competitor pattern (50% similar to target , presented at study but not practice)
   and two controls (50% similar to each other, presented at study but not practice)
2. **Train the network on these patterns**
   Present the network with the complete patterns
   Update weights after each pattern
3. **Pretest the network's ability to pattern complete on all patterns**
   Present 1/8 units of the pattern as cue.
4. **Allow network to practice target pattern**
   In case of partial practice: 4/8 units presented
   In case of full practice: 8/8 units presented
5. **Test the network's ability to pattern complete on all patterns again**
   Compare to pretest performance to calculate practice effect

Network behavior during training

Graphs show activation of target and competitor , and the inhibitory oscillation on one trial



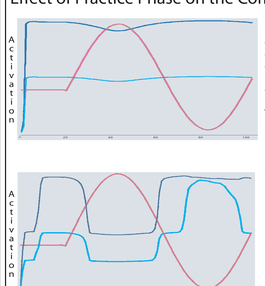
Early in training:

- High inhibition
  Target units decrease in activity
- Low inhibition
  Competing units increase in activity

Late in training:

- High inhibition
  Target units strong enough to stay on
- Low inhibition
  Competing units too weak to come on

Effect of Practice Phase on the Competitor



Full practice (8/8 units):

- Cue matches the target perfectly
- Just like "late in training" above
- Inhibitory oscillation does not affect network activity
- Because the competitor does not pop up, it does not get punished: The effect of full practice on competitor recall is negligible (.32 increase in sum squared error)

Partial practice (4/8 units):

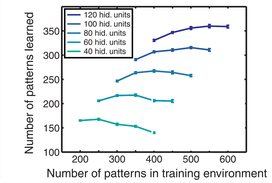
- Match between the cue and target (relative to match between the cue and competitor) is less good here
- The competitor turns on during the low inhibition phase
- This change in activity leads to substantial weakening of the competitor (4.56 increase in sum squared error).

Capacity Simulations

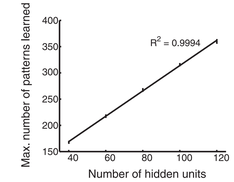
Uncorrelated Patterns

We trained the network on randomly generated input patterns with 8/80 units active. We tested the network after 25 epochs of training by presenting 7/8 units from trained patterns; the network had to activate the missing unit.

The graph on the left plots number of correct responses as a function of the number of patterns in the training set and the number of hidden units. The average number of active units in the hidden layer was held constant at k = 8.

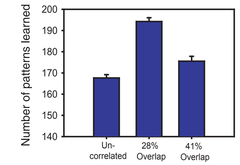


The graph on the right plots the maximum number of patterns learned as a function of the number of hidden units. The capacity of the network appears to increase in a linear fashion as a function of the number of hidden units.



Correlated Patterns

In this simulation, we generated correlated patterns by "flipping bits" away from a prototype pattern, and explored how varying the degree of correlation affects capacity. This simulation used 40 hidden units (k = 8) and 250 input patterns.



Other self-organizing learning algorithms (e.g., CPCA Hebbian learning; O'Reilly & Munakata, 2000) lose their memory for item-specific details when given large numbers of correlated patterns. However (up to a point) increasing overlap between patterns actually **increases** our model's capacity for recalling specific, non-prototypical features of individual patterns.

Our model avoids collapse because of its tendency to evenly space representations in the hidden layer. If representations get too close to each other, competitor-punishment mechanisms push them away; also, the model benefits from its ability to focus learning on features that are not already well-learned. Increasing overlap ends up boosting capacity because the model can exploit redundancies in correlated patterns in order to code them more efficiently.

Relation to Theta Oscillations

- There is extensive evidence that theta-frequency inhibitory oscillations are related to learning in cortex and hippocampus (e.g., Raghavachari et al., 2001; Rizzuto et al., 2003) but very little agreement regarding how, mechanistically, they contribute. This model shows how inhibitory oscillations can help train cortical attractor networks by alternately "stress-testing" target memories and revealing competitors.

- The fact that the sign of our learning rule depends on the phase of the inhibitory oscillation is reminiscent of Huerta & Lisman's (1996) finding that the "sign" of plasticity (LTP vs. LTD) depends on theta phase. However, much more work needs to be done to flesh out the details of this comparison. In future research, we will explore how our model relates to other, more biologically detailed models of how theta modulates learning, in the hippocampus and elsewhere (e.g., Hasselmo, Bodelon, & Wyble, 2002).

Applications of the Learning Rule

We are currently using the model to simulate:

- Other retrieval-induced forgetting findings (e.g., the effects of target and competitor strength/similarity)
- Cognitive dissonance reduction (Norman & Hovnanian, in preparation)
- Familiarity discrimination: Familiarity = the size of the dip in activation when inhibition increases above baseline
- Learning during sleep: How learning based on theta oscillations during REM can strengthen stored memories & help protect them from interference (Norman & Perotte, in preparation)

We are also planning to apply the model to other instances of competitor-punishment, e.g., negative priming effects.

Finally, although our model (as presented here) does not include prefrontal cortex (PFC), we think PFC plays a critical role in biasing competition when the correct response is not the dominant response. Future modeling work will directly address PFC contributions.

References

Hasselmo, M.E., Bodelon, C., Wyble, B.R. (2002). A proposed function for hippocampal theta rhythm: separate phase of encoding and retrieval enhance reversal of prior learning. *Neural Computation*, 14, 793-817.

Huerta, P.T. & Lisman, J.E. (1996). Synaptic plasticity during the cholinergic theta-frequency oscillation in vitro. *Hippocampus*, 6(1), 58-61.

Levy, B.J. & Anderson, M.C. (2002). Inhibitory processes and the control of memory retrieval. *TRENDS in Cognitive Sciences*, 6(7) , 299-305.

O'Reilly, R.C., & Munakata, Y. (2000). *Computational explorations in cognitive neuroscience*. Cambridge, MA: MIT Press.

Movellan, J.R. (1990). Contrastive Hebbian learning in the continuous Hopfield model. In D.S. Tourtezky, G.E. Hinton, & T.J. Sejnowski (Eds.), *Proceedings of the 1989 Connectionist Models Summer School* (pp. 10-17). San Mateo, CA: Morgan Kaufman.

Norman, K.A., Newman, E.L., Detre, G., & Polyn, S.M. (in preparation). How inhibitory oscillations can train neural networks and punish competitors.

Raghavachari, S., Kahana, M.J., Rizzuto D.S., Caplan, J.B., Kirschen, M.P., Bourgeois, B., Madsen, J.R., and Lisman, J.E. (2001). Gating of human theta oscillations by a working memory task. *J Neurosci*, 21(9), 3175-3183.

Rizzuto, D.S., Madsen, J.R., Bromfield, E.B., Schulze-Bonhage, A., Seelig, D., Aschenbrenner-Scheibe, R., and Kahana, M.J. (2003). Reset of human neocortical oscillations during a working memory task. *Proc Natl Acad Sci U S A*, 100(13), 7931-7936.

ELN was supported by an NIH Training Grant in Quantitative Neuroscience (MH65214) awarded to Princeton University

For a downloadable version, please visit: http://compmem.princeton.edu/publications.html