

# Storing messages with multipartite neural cliques

???

## Abstract

???

**Index terms**— associative memory, error correcting code

probability for  $i$  and  $j$  to be active in their respective clusters for one message :  $\frac{a}{l}$

probability of the edge not being added in the network for one message :  $1 - \left(\frac{a}{l}\right)^2$

$$d = 1 - \left(1 - \left(\frac{a}{l}\right)^2\right)^m$$

## I Introduction

## II Networks of neural cliques

### 1 Learning messages

### 2 Retrieving messages

#### a The "winner takes all" rule

#### b The " $a$ winners take all" rule

less biologically plausible rule

same algorithmic complexity

gap with density in simulations : under 1 percent

probability for a vertex  $v$  to achieve the maximum score, that is to mean  $a(c - c_e)$  :  $d^{a(c-c_e)}$

probability for a vertex  $v$  not to achieve the maximum :  $1 - d^{a(c-c_e)}$

probability for none of the vertices of one erased cluster, excepting the correct ones, to achieve the maximum :  $\left(1 - d^{a(c-c_e)}\right)^{l-a}$

probability for none of the vertices in any erased cluster, excepting the correct ones, to achieve the maximum in their respective cluster :  $\left(1 - d^{a(c-c_e)}\right)^{c_e(l-a)}$

Whence error rate is :

$$P_{err} = 1 - \left(1 - d^{a(c-c_e)}\right)^{c_e(l-a)}$$

## III Retrieval performance

### 1 Analytical results

number of neurons per cluster  $l$

number of clusters  $c$

number of erased clusters  $c_e$

number of messages :  $m$

number of activities per cluster :  $a$

density :  $d$

edge :  $(i, j)$

### 2 Simulations

Improves the retrieval rate for multiple iterations and/or corrupted messages.

The simulations agree with the analytical results for density and error rate.

For one iteration (for erasures, not for errors), winner takes all is the same as  $a$ -winners take all.

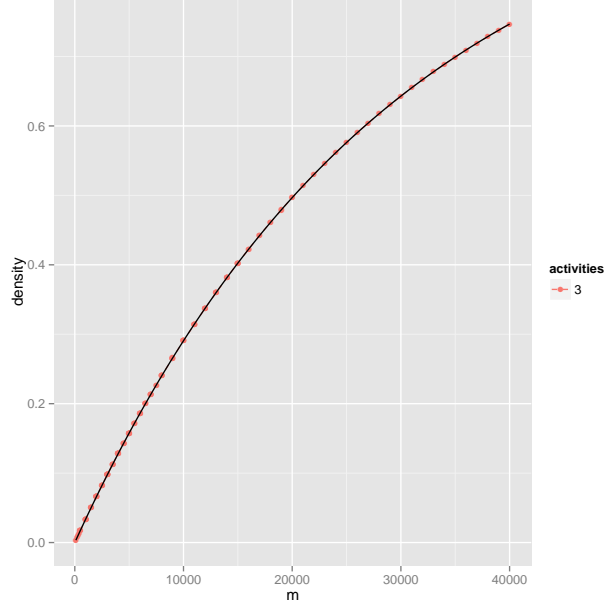


Figure 1: Theoretical and empirical densities

For multiple iterations  $a$ -winners take all is a significant improvement.

" $a$ -winners take all" better than "winner takes all" with errors instead of erasures, even for one iteration.

For errors : only better than only 1 activity per cluster if multiple iterations of the  $a$ -winners take all rule. For one iteration, it is less efficient.

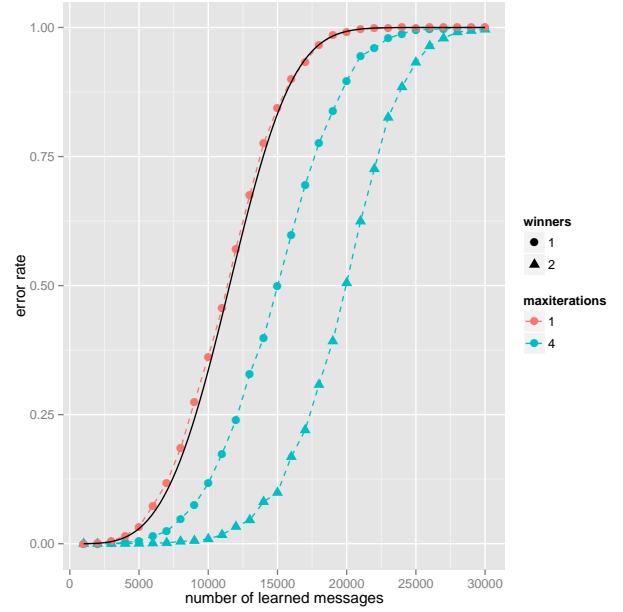


Figure 2: 2 activities per cluster, 4 clusters, 512 neurons per cluster, two erasures, each point is the mean of 5 networks with 1000 sampled messages, analytical result in continuous black

Spéculations en Français :

Peut-être un intérêt pour cluter based associative memories build from unreliable storage : si une arête porte moins d'info, on peut peut-être en supprimer plus (mais rajout ?!)

Marche mieux pour des bruits importants

$$P(n_{v_c} = n_0) = \binom{ac_k}{n_0} (1 - \psi)^{n_0} \psi^{ac_k - n_0}$$

$$P_+ = \psi(1 - d) + (1 - \psi)d$$

$$P(n_v = x) = \binom{ac_k}{x} P_+^x (1 - P_+)^{ac_k - x}$$

Sans tenir compte du choix au hasard si plus sont activés.

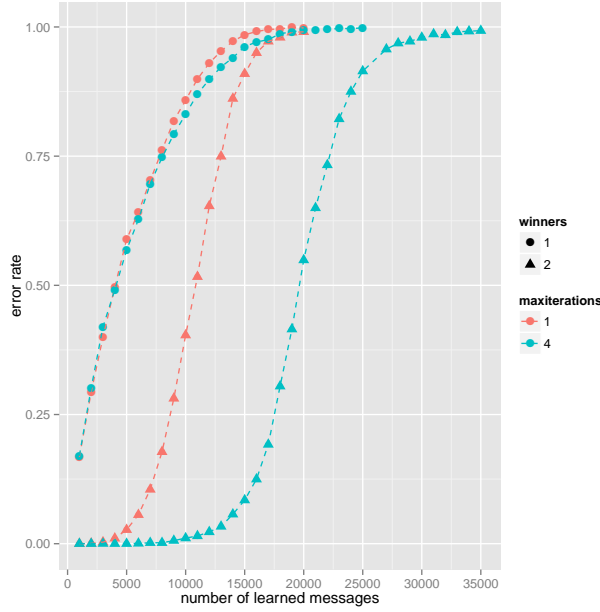


Figure 3: 2 activities per cluster, 4 clusters, 512 neurons per cluster, two erasures, each point is the mean of 5 networks with 1000 sampled messages, analytical result in continuous black

Formula for errors insted of erasures :

$$\begin{aligned}
 P(n_{gc} = (c - c_e - 1) + \gamma + x) &= \binom{c_e}{x} d^x (1 - d)^{ce - x} \\
 P(n_{ge} = (c - c_e) + \gamma + x) &= \binom{c_e - 1}{x} d^x (1 - d)^{c_e - 1 - x} \\
 P(n_{abe} = \gamma + x) &= \binom{c - 1}{x} d^x (1 - d)^{c - 1 - x} \\
 P(n_{be} = x) &= P(n_{bc} = x) = \binom{c - 1}{x} d^x (1 - d)^{c - 1 - x}
 \end{aligned}$$

## IV Conclusion

## References

- [LPGRG14] François Leduc-Primeau, Vincent Gripon, Michael Rabbat, and Warren Gross. Cluster-based associative memories built from unreliable storage. In *ICASSP*, May 2014. To appear.