

Storing messages with multipartite neural cliques

Nissim Zerbib¹, Vincent Gripon², and ?³

¹*Département d'Informatique, École normale supérieure, Paris, France*

²*Département d'Électronique, Télécom Bretagne, Brest, France*

Abstract

We extend recently introduced associative memories based on cliques to multipartite cliques. We propose a variant of the classic retrieving rule. We study their relative performance for retrieving partially erased and corrupted messages. We provide both analytical and simulation results showing improvements of capacity.

Index terms— associative memory, error correcting code, cliques, multipartite cliques, neural networks

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

III Retrieval performance

1 Retrieving partially erased messages

a Analytical result

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)

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 : $1 - \left(\frac{a}{l}\right)^2$

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

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 : $(1 - d^{a(c-c_e)})^{l-a}$

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

Whence error rate is :

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

b 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.

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.

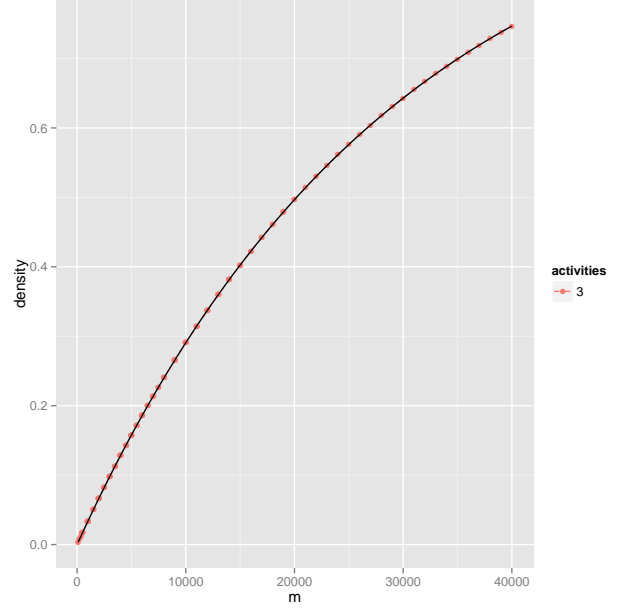


Figure 1: Theoretical and empirical densities

a

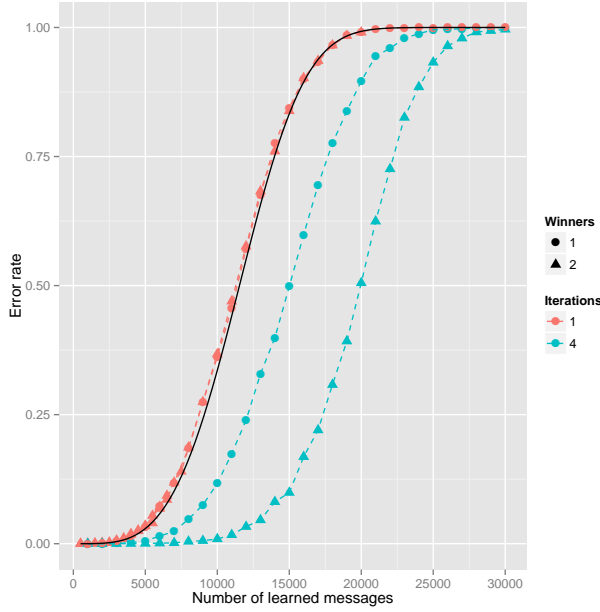


Figure 2: 2 activities per cluster, 4 clusters, 512 neurons per cluster, two erasures, each point is the mean of 5 networks with 1000 sampled messages, analytical result in continuous black

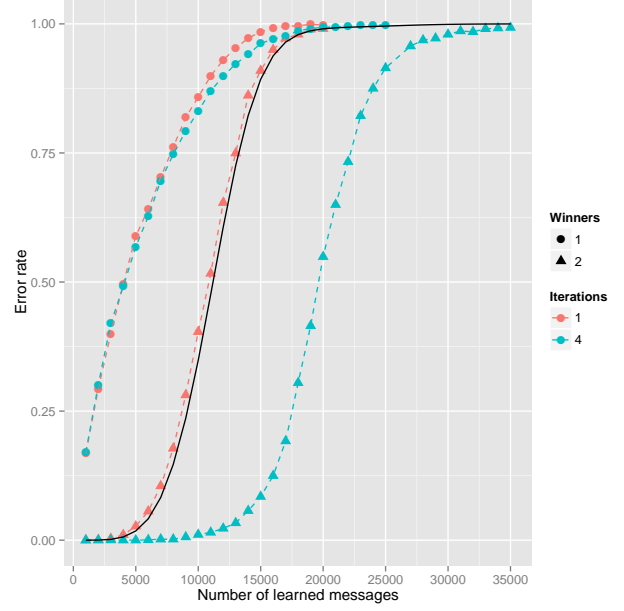


Figure 3: 2 activities per cluster, 4 clusters, 512 neurons per cluster, one erroneous cluster, each point is the mean of 5 networks with 1000 sampled messages, analytical result in continuous black

2 Retrieving corrupted messages

Spéculations en Français :

Peut-être un intérêt pour cluster 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.

a priori for noise on edges :

$$P(\text{no other vertex activated in this cluster}) = \sum_{n=1}^{ac_k} \left[\sum_{k=1}^a \binom{a}{k} P(n) \right]$$

$$P(\text{no other vertex activated in any cluster}) = \left(\sum_{n=1}^{ac_k} \left[\sum_{k=1}^a \binom{a}{k} P(n) \right] \right)^{ac_k}$$

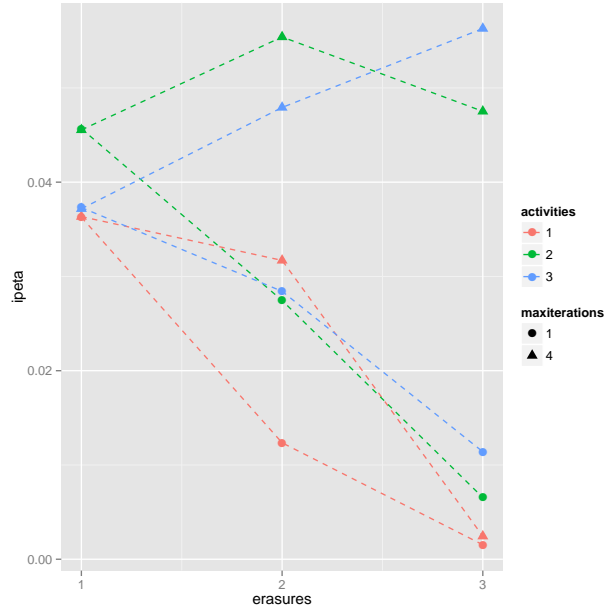


Figure 4: Comparison between different number of activities per cluster : 4 clusters, 512 neurons per cluster, 1000 sampled messages

a

Formula for errors insted of erasures :

gc : good (à remplaceur par right) neuron in correct cluster

ge : good neuron in erroneous cluster

abe : bad activated neuron in erroneous cluster

be : others bad (à remplaceur par wrong ou incorrect etc.) neurons in erroneous cluster

bc : bad neuron in correct cluster

$$P(n_{gc} = (c - c_e - 1) + \gamma + x) = \binom{c_e}{x} d^x (1 - d)^{c_e - x}$$

$$P(n_{ge} = (c - c_e) + x) = \binom{c_e - 1}{x} d^x (1 - d)^{c_e - 1 - x}$$

$$P(n_{abe} = \gamma + x) = \binom{c - 1}{x} d^x (1 - d)^{c - 1 - x}$$

$$P(n_{be} = x) = P(n_{bc} = x) = \binom{c - 1}{x} d^x (1 - d)^{c - 1 - x}$$

$$P_{retrieve} = \left[\sum_{n=1}^{c-1+\gamma} P(n_{gc} = n) \left[\sum_{x=0}^{n-1} P(n_{bc} = x) \right]^{l-1} \right]^{c-c_e} \\ \times \left[\sum_{n=1}^{c-1+\gamma} P(n_{ge} = n) \left[\sum_{x=0}^{n-1} P(n_{bc} = x) \right]^{l-2} \left[\sum_{x=0}^{n-1} P(n_{abc} = x) \right] \right]^{c_e}$$

generalisation to multiple activities per cluster

:

$$P(n_{gc} = a(c - c_e - 1) + \gamma + x) = \binom{ac_e}{x} d^x (1 - d)^{ac_e - x}$$

$$P(n_{ge} = a(c - c_e) + x) = \binom{a(c_e - 1)}{x} d^x (1 - d)^{a(c_e - 1) - x}$$

$$P(n_{abe} = \gamma + x) = \binom{a(c - 1)}{x} d^x (1 - d)^{a(c - 1) - x}$$

$$P(n_{be} = x) = P(n_{bc} = x) = \binom{a(c - 1)}{x} d^x (1 - d)^{a(c - 1) - x}$$

$$P_{retrieve} = \left[\sum_{n=1}^{a(c-1)+\gamma} \left[\sum_{k=1}^a \binom{a}{k} P(n_{gc} = n)^k P(n_{gc} > n)^{a-k} \right] \right. \\ \times \left[\sum_{n=1}^{a(c-1)+\gamma} \left[\sum_{k=1}^a \binom{a}{k} P(n_{ge} = n)^k P(n_{ge} > n)^{a-k} \right] P(n_{bc} < n) \right] \\ \left. P_{err} = 1 - P_{retrieve} \right]$$

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.