Storing messages with multipartite neural cliques

Nissim Zerbib[1], Vincent Gripon[2], 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 over.

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)

As the probability for i and j to be active in their respective clusters for one message is $\frac{a}{l}$, providing messages are independent, it follows that the probability for the edge (i,j) of not being added in the network for one message is $1 - \left(\frac{a}{l}\right)^2$. Thus the density d can be expressed like this :

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

Thanks to the memory effect γ, activated neurons in a non-erased cluster stays activated and are the only ones doing so in their cluster. For neurons in erased clusters, the maximum attainable score is $a(c - c_e)$. Neurons corresponding

1

to the original message achieve this score. Besides the probability for a neuron v to achieve the maximum score in such a cluster, that is to mean $a(c - c_e)$ is $d^{a(c-c_e)}$.

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

Since neurons representing messages are independent (as messages are themselves independent), the probability for none of the vertices of one erased cluster, excepting the correct ones, to achieve the maximum is $\left(1 - d^{a(c-c_e)}\right)^{l-a}$.

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

Whence error rate in retrieving messages is :

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

$$\eta_m = (P_{retrieve}m)\frac{2\left(c\log_2\binom{l}{a} - \log_2(m)+1\right)}{c(c-1)l^2} \times$$
$$\frac{c}{(c-c_e)\log_2\binom{l}{a}}$$
$$\eta_{max} = \max_x \eta_x$$

b Simulations

gap of theoretical density with density in simulations : under 1 percent

New rule : 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.
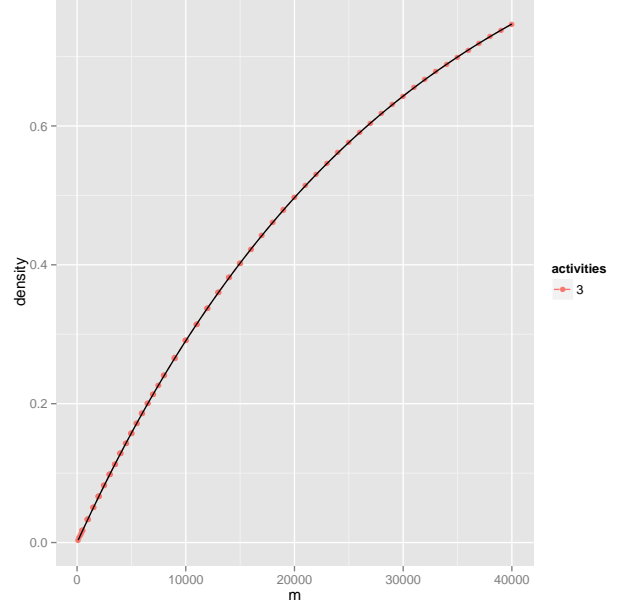

Figure 1: Theoretical and empirical densities

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.
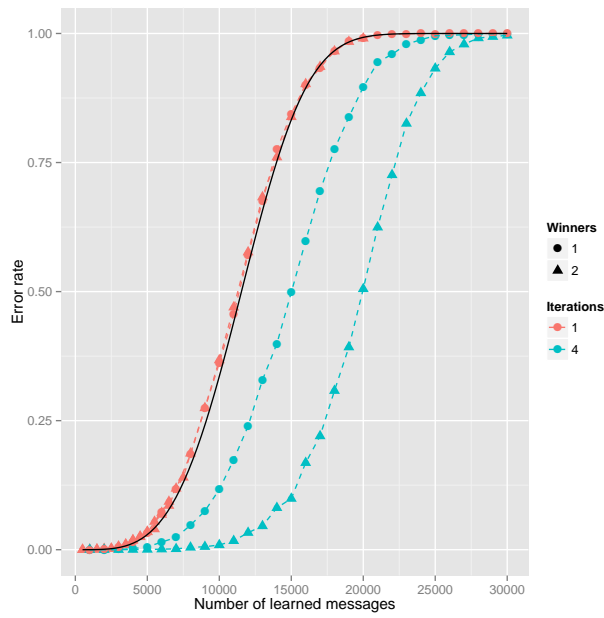
p

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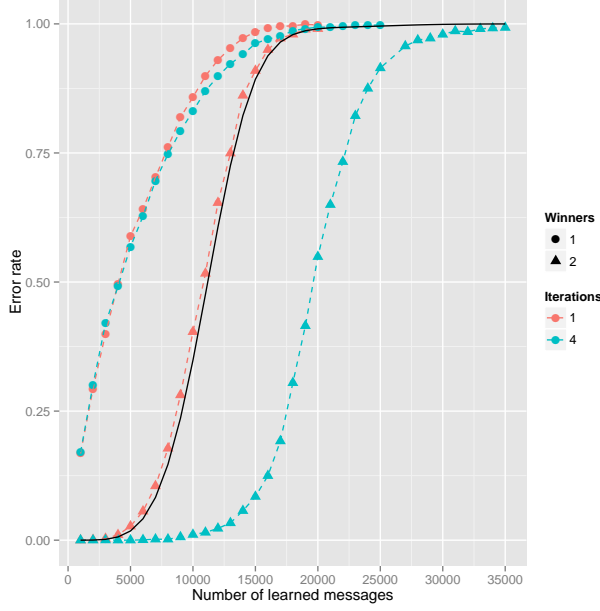
Figure 3: 2 activities per cluster, 4 clusters, 512 neurons per cluster, one erroneous cluster, each point is the mean of 5 networks with 1000 sampled messages, analytical result in continuous black

2 Retrieving corrupted messages

Formula for errors instead of erasures :

gc : good (à remplacer par right/wrong ?) neuron in correct cluster

ge : good neuron in erroneous cluster

abe : bad activated neuron in erroneous cluster

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

bc : bad neuron in correct cluster

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

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

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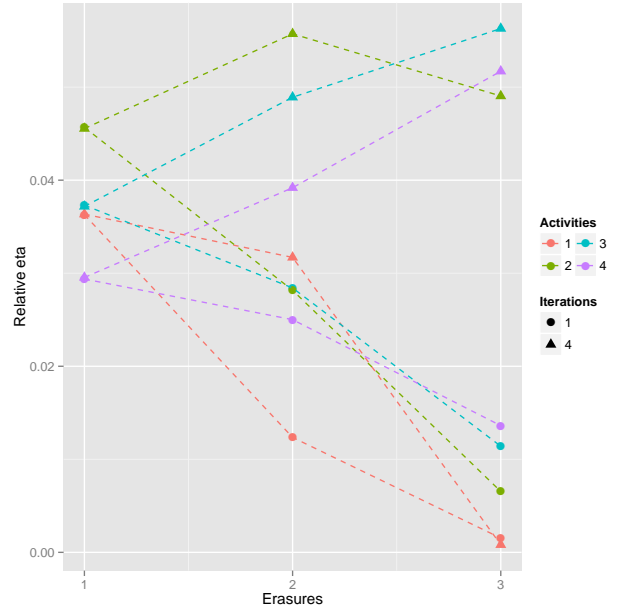
$P(n_{be} = x) = P(n_{bc} = x) = \binom{c-1}{x} d^x (1-d)^{c-1-x}$

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

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

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

generalisation to multiple activities per cluster

:

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

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

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

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

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

$$\times \left[\sum_{n=1}^{a(c-1)+\gamma} \left[\sum_{k=1}^{a} \binom{a}{k} P(n_{ge} = n)^k P(n_{ge} > n)^{a-k} \right] P(n_{bc} < n)^{l-2a} P(n_{abc} < n)^a \right]^{c_e}$$

$$P_{err} = 1 - P_{retrieve}$$

IV   Conclusion

V   Resilience

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.

a priori for noise on edges :

$P($no other vertex activated in this cluster$) =$

$\sum_{n=1}^{a(c-c_e)} \left[\sum_{k=1}^{a} \binom{a}{k} P(n_{v_c} = n)^k P(n_{v_c} > n)^{a-k}\right] \left[\sum_{x=0}^{n-1} P(n_v = x)\right]^{l-a}$

$P($no other vertex activated in any cluster$) =$

$\left(\sum_{n=1}^{a(c-c_e)} \left[\sum_{k=1}^{a} \binom{a}{k} P(n_{v_c} = n)^k P(n_{v_c} > n)^{a-k}\right] \left[\sum_{x=0}^{n-1} P(n_v = x)\right]^{l-a}\right)^{c_e}$

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.