Overlapping Clustered Neural Associative Memories

Amin Karbasi, Amir Hesam Salavati
E-mail: amin.karbasi@epfl.ch,hesam.salavati@epfl.ch

Algorithmics Laboratory (ALGO)
Ecole Polytechnique Federale de Lausanne (EPFL)

April 13, 2012

1 Background

In the paper we submitted to ISIT 2012, we considered a multi-level neural associative memory, shown in figure 1 in which each sub-network in the first level enforced some local constraints and the second level enforced some constraints globally on the pattern. Furthermore, the local sub-networks in the first level were non-overlapping. We proposed a learning algorithm for both local and global networks. The recall method was borrowed from [2], with slight modifications.

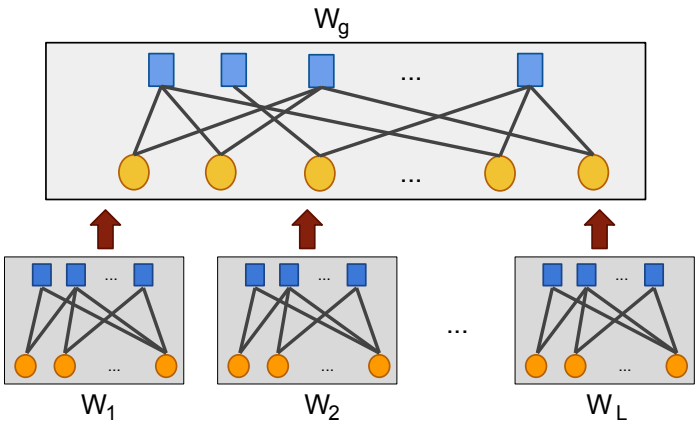

Figure 1: A two-level neural associative memory with non-overlapping local sub-networks.

The rationale behind this new architecture was due to two limiting aspects of the approach propoed in [2]:

1. The neural associative memory suggested in [2] is based on expander graphs. However, when desiging an iterative neural learning method, making the final graph sparse is difficult yet alone making it expander. Therefore, in [1] we proposed a learning agorithm to find sparse neural graphs and used this graph for error correction as well.

2. However, the error correction performance deteriorates when switching from expander to sparse graph. To compensate this loss, we modified the architecture so that we will hve multi-levels of error correction.

In the proposed model in our ISIT paper [1], each sparse neural graph is capable of correcting one error with high probability and more errors can be corrected but with some probability of making mistake. Therefore, when we have multiple local networks, each of them capable of correcting one single error, multiple errors in the pattern may fall in the domain of different local networks, resulting in their correction. If some of the errors remained uncorrected, the constraints in the second level may correct the remaining errors. Figure 2 illustrates the simulation results performed in [1] to compare the recall probability of error for a single-level network, a two-level network and the approach used in [2].

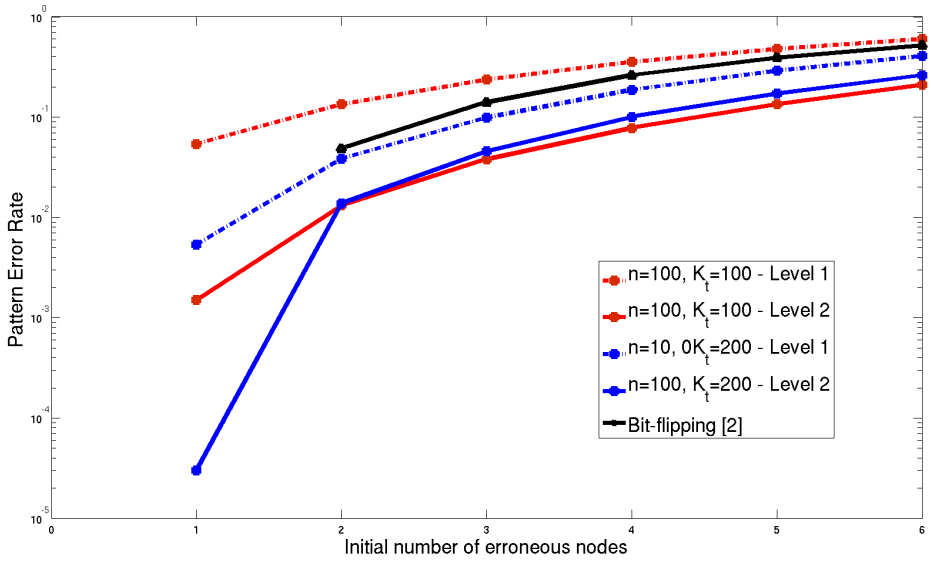

Figure 2: Block error rate for different architectures and network sizes [1].

The problem with architecure illustrated in figure 1 is that there is no guarantee that the second level corrects more than one remaining error either. In other words, given that the learning and recall methods behind the local and the global networks are the same, both of them are guaranteed to correct one error falling in their domain. The only difference is that the probability of correcting i errors increases as the network size n increases. Therefore, there is a better chance that the second level corrects i errors while local networks have failed to do so. And it is obvious from the results shown in figure 2 that the overall performance of the algorithm is better than that of [2]. [1] In the next section, we propose a novel architecure and formulate the multi-level neural associative memory in a way that its performance can be analytically assessed.

[1] Albeit the comparison conducted in [1] might not be fair as the total number of constraint nodes are larger than that of [2]. We address this issue in this report as well.

2 New architecture

In the new model and problem formulation, we assume that we have a budget in terms of the number of constraint nodes we can have access to. Now the question is what is the best architecture which yields the lowest recall error probability with the given number of constraint nodes. To be more specific, assume we have N pattern nodes and a total of M constraint nodes. How should we spread these M constraint nodes to get the best performance? In [2], we used all the M constraints in a single layer, resulting in a biprtite neural graph capable of correcting at least one error in the input pattern. In [1], we spread the constraint nodes into two levels and further clusteredthem in the first level so that we ended up with local non-overlapping sub-network, each capable of correcting one bit of error. Therefore, we reduced the probability of error as the first level could correct multiple errors *if* they were distributed among separate local networks.

Following the same line of thought, one might think about different ways of clustering neurons so that errors get separated from each other. Here, we remove the second level and instead, distribute its constraints among the neurons of the first level. Therefore, we will have *overlapping local* connections. The model is illustrated in figure 3

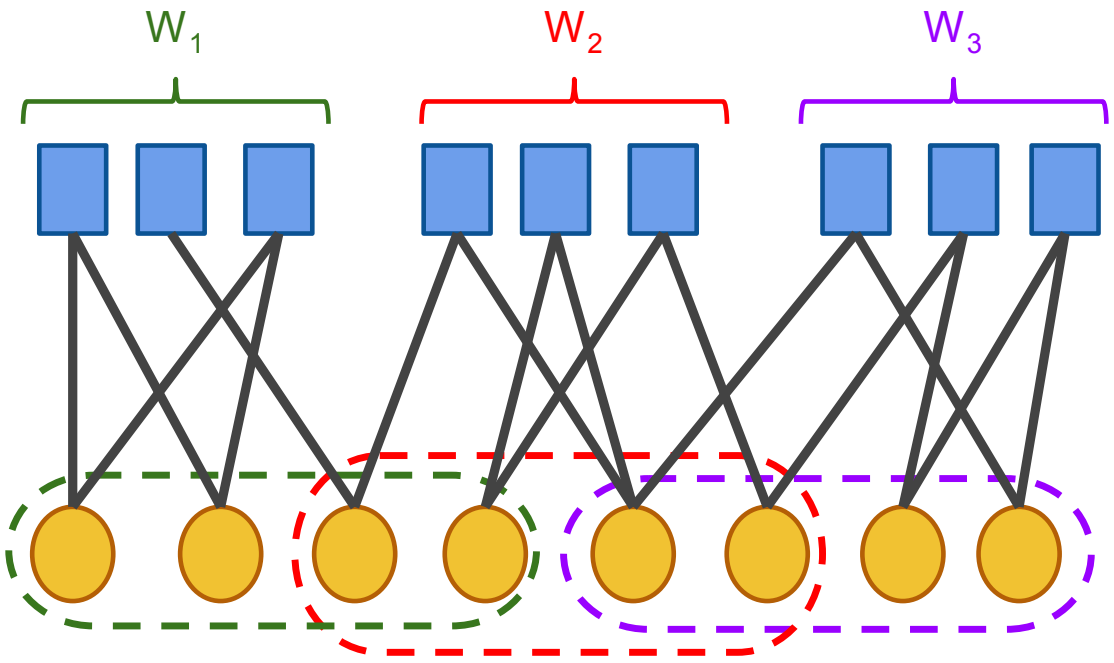

Figure 3: Overlapping clustered neural associative memory.

In the learning phase, we follow the same learning approach mentioned in [1] or [3]. Therefore, we expect each sub-network be a sparse weighted graph. During the recall phase, we utilize the bit-flipping or the winner-take-all algorithms mentioned in [1]. In order to analyze the error correction performance of the proposed method during the recall phase, we follow a similar approach to *peeling*

decoders in coding theory [4]:

1. If there is a cluster with only one bit of error in it, we assume it gets corrected after a few iterations. [2]

2. Removing the errors corrected in this round, we continue the above iteration until either all errors are corrected or no cluster with a single error remains, i.e. there are more than one error in all clusters. In the first case, we declare success, otherwise we have failed to correct the errors.

The advantage of the above model is its simplicity and a very rich analytical background in coding theory.

3 Remarks

In the above decoding procedure, we assumed that single errors get corrected after a few iterations. In fact if the noise value is ± 1, this happens in exactly one iteration. However, due to overlaps, we have to be creful with the effect of other neighboring clusters. For instance, it might be the case that two errors happen in a cluster and becuase it is not guaranteed to correct two errors, the cluster diverges from the original pattern and introduces even more errors. Therefore, we have to make sure that as long as the number of initial errors is less than a threshold, error correction operation does not increase the number of erroneous nodes.

4 Possible limitations and future works

The proposed model above assumes that the sub-patterns in each sub-cluster form a sub-space in order to be able to apply the learning and recall algorithms mention in [1]. Whether this is a very strict assumption for real-world patterns we do not know yet and have to check with neuroscientists. A possible remedy to this issue is to replace the sub-space assumption with the assumption that entries in each sub-pattern are correlated to each other in the way that the correlation matrix has several minor-components very close to zero. Now during the learning algorithm, we create random clusters and after the end of learning, those that had several minor components will form a cluster. Others, in which entries were not correlated, do not converge and we can dismiss them in the recall phase. Therefore, by looking for local correlations among patterns and simple learning and recall methods we could manage learning an exponential number of patterns while correcting many input errors.

References

[1] A. H. Salavati, A. Karbasi, *Multi-level error-resilient neural networks*, Submitted to the IEEE International Symposium on Information Theory (ISIT), 2012

[2] We can follow a sequential approach in which the first sub-network performs one iteration of error correction, then the second one and so on. The advantage of this scheduling is that as long as error alues are ± 1, single errors get corrected in one iteration

[2] K.R. Kumar, A.H. Salavati and A. Shokrollahi, *Exponential pattern retrieval capacity with non-binary associative memory*, Proc. IEEE Information Theory Workshop, 2011.

[3] K.R. Kumar, A.H. Salavati and A. Shokrollahi, *Exponential pattern retrieval capacity with non-binary learning associative memory*, Under preparation.

[4] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, V. Stemann, *Practical loss-resilient codes*, Proc. ACM Symp. on Theory of Computing, pp. 150159, 1997.