

Title: Applications of Coding Theory to Increase the Storage Capacity of Neural Networks

Abstract: Memorizing patterns and correctly recalling them later is an essential ingredient of neural activity. In past 25 years, a number of neural networks has been invented memorize and recall patterns. Interestingly, some of these networks are able to recall the correct pattern even if the input pattern contains error and is partially corrupted because of noise. In this regard, these artificial networks resemble error correcting codes, i.e. they are able to recognize the correct pattern in presence of noise.

However, the storage capacity of these networks are quite small compared to their counterparts in coding theory. Given the fact that modern codes use the same basic structure to do error correction and the one used by neural networks, i.e. a bipartite graph with local message passing, it seems interesting to consider applications of modern coding theory to increase the storage capacity of neural networks by finding the appropriate weights for the neural graph.

Up to this point, some weighting schemes (including the Hebbian rule) were tested without much success. Considering more weighting schemes, such as the BCM rule, would be the next step toward the goal of increasing the storage capacity which is the main objective of this project.

The objectives of this project are:

- 1) To understand the principles of neural networks and modern codes.
- 2) Applying different weighting scheme to neural networks and evaluate the performance for storing codewords of a linear code.

This project is suitable for students who prefer doing simulations to find the appropriate framework for doing theoretical analysis.

The prerequisites are:

- 1) Basic knowledge of coding theory.
- 2) Being familiar with a suitable programming language (C/C++,MATLAB)