Storkey Learning Rules for Hopfield Networks

Xiao Hu
Bejing Technology Group

September 18, 2013

Abstract

We summarize the Storkey Learning Rules for the Hopfield Model, and evaluate performance relative to other learning rules. Hopfield Models are normally used for auto-association, and Storkey Learning Rules have been found to have good balance between local learning and capacity. In this paper we outline different learning rules and summarise capacity results. Hopfield networks are related to Boltzmann Machines: they are the same as fully visible Boltzmann Machines in the zero temperature limit. Perhaps renewed interest in Boltzmann machines will produce renewed interest in Hopfield learning rules?

1 Introduction

The Hopfield Model [1] is a deterministic neural network model commonly used for auto-association. It is related to spin systems in statistical physics and is the same as a fully visible Boltzmann Machine in the zero temperature limit. The Hopfield Model produces auto-association via dynamics that leads to attractive fixed points. The usual purpose of a Hopfield Model is to make sure the training data are fixed points of the Hopfield Network dynamics. The network can generalise in two ways: first certain patterns are not fixed points and have a dynamic that leads to another pattern - this gives an association. Second the network can make fixed points that are not the required memory patterns. These are called spurious patterns. However we can view spurious patterns as potential generalisations of the stored memories.

The weights of the Hopfield neural network are to be set to make the memories be fixed points of the dynamics. This is done with a **learning rule**. There are many learning rules for Hopfield Networks. In this paper we discuss learning rules in general, but focus on the Storkey Learning Rules [2, 3, 4, 5] first introduced in [6, 7, 8, 9, 10], and later discussed in [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21] among others. We choose this focus as we find that these rules have valuable properties, such as locality and, most importantly, online learning (called being incremental in [10]). However they also have a high capacity (defined later). They seem to have become the most desirable rules to use in the Hopfield network setting.

Let the variables to be modeled be s_i and take the values $+1$ or -1 as in statistical physics. The Hopfield Model is described by the dynamic update

$$s_i \to \text{sign}\left(\sum_{j=1}^{n} J_{ij} s_j\right) \tag{1}$$

where J is a symmetric weight matrix with zeros on the diagonal, and the updates are done asynchronously for $i = 1, 2, \ldots, n$, where n is the number of neurons in the network.

Given an initial setting for $(s_1, s_2, s_3, \ldots, s_n)$ we can run the network to find the pattern associated with this initial setting. Ideally we want the patterns to be found to be stored memories of the system.

2 Learning Rules

The learning rules are used to set the weight matrices to store memories in the network. The first learning rule proposed by John Hopfield himself [1] is a formulation of the Hebb rule, due to Donald Hebb. Consider the case of an incremental process where we have a weight matrix J_{ij} and we wish to update it to an new J_{ij} so that the new memory given by pattern $(s_1, s_2, s_3, \ldots, s_n)$ is stored. All the old memories must also remain stored.

2.1 Hopfield-Hebb rule

In this context the Hopfield-Hebb (or Hebb for short) rule states that the weight matrix J_{ij} is updated using

$$J_{ij} \to J_{ij} + s_i s_j \tag{2}$$

for all i different from j. The diagonal terms J_{ii} must be zero for all i.

2.2 Hopfield-Perceptron Rule

The Hopfield-Perceptron rule derives from the maximum likelihood learning algorithm for the Perceptron, by applying it to auto-association instead of the perceptron. It is an iterative, non-incremental gradient based rule, as opposed to the Hebb rule and the Storkey Rules below which are all one shot rules. The iterative update is

$$J_{ij} \to J_{ij} + \alpha \sum_k \left(s_{ki} - \text{sign}\left(\sum_{r=1}^{n} J_{ir} s_{kr}\right)\right) s_{kj} + \alpha \sum_k \left(s_{kj} - \text{sign}\left(\sum_{r=1}^{n} J_{jr} s_{kr}\right)\right) s_{ki} \tag{3}$$

where s_{ki} is the ith element on the kth training item, and α is a learning rate.

The Hopfield-Perceptron rule was used in [22], for Temporal Hidden Hopfield Networks. It maximizes the probability that each item maps to itself. A zero diagonal for J is necessary to ensure the Perceptron rule cannot just learn the identity matrix.

2.3 Boltzmann Machine Learning Rule

As the Hopfield Model is a special case of a Boltzmann Machine in the limit of zero temperature, it is sensible to use the Boltzmann Machine Learning Rule to train Hopfield Networks. There is one difficulty. The gradient in the Boltzmann machine learning rule involves expectations with respect to the model, usually approximated by sampling generatively from the Boltzmann machine. This is done with Gibbs MCMC. However in the zero temperature limit, the Boltzmann machine is not ergodic, so a Gibbs Markov chain does not produce a unique equilibrium.

Good mixing for Boltzmann machine sampling is hard even at finite temperature. One method is contrastive divergence, which makes a single step from each data point to approximate a sample from the distribution. The result is the contrastive divergence rule.

2.3.1 Contrastive Divergence Rule

The Contrastive Divergence Rule for Hopfield Networks is

$$J_{ij} \to J_{ij} + \alpha \left(\sum_k s_{ki} s_{kj} - \sum_k t_{ki} t_{kj} \right) \tag{4}$$

where $t_{kj} = \text{sign}\left(\sum_r J_{ir} s_r\right)$.

We can also use a form of contrastive divergence rule in an incremental online, rather than iterative, setting. The form

$$J_{ij} \to J_{ij} + \left(\sum_k s_{ki} s_{kj} - \sum_k t_{ki} t_{kj} \right) \tag{5}$$

performs poorly, but the form

$$J_{ij} \to J_{ij} + s_{ki} s_{kj} - h_{ki} h_{kj} \tag{6}$$

is a much better performing rule. We use this in the experiments below.

2.4 Pseudo-inverse Rule

Where the Hopfield-Perceptron rule is used to map each memory to itself (after thresholding with the sign function), the pseudo-inverse rule attempts to map a memory to itself before thresholding: that is it maps to the ± 1 activation levels rather than just $\pm$ activation levels. This turns the problem into a real valued regression problem which can be solved. The solution to linear regression is given by the pseudo-inverse, leading to the pseudo-inverse solution for the Hopfield Model. The Hebb rule is the special case of the pseudo-inverse rule for training vectors that are orthogonal.

$$J_{ij} = \frac{1}{n} \sum_{k,l} s_{ki} (C^{-1})_{kl} s_{lj} \tag{7}$$

where

$$C_{kl} = \frac{1}{n} \sum_i s_{ki} s_{li} \tag{8}$$

Note that this learning rule is a batch rule: it does not involve local computation; it does not involve incremental updates.

2.5 Storkey Rules

The motivation for various Storkey Rules, all derived in [10], can come from two directions. They can either be seen as an approximation to the pseudo inverse, or they can be seen as derived from the Hebb rule applied to residuals.

2.5.1 Storkey General Rule

Storkey notes [10] that the pseudo-inverse rules can be used to get an incremental update rule for the pseudo-inverse solution and proposes the general form of update rule

$$J_{ij} \to J_{ij} + \frac{c(k, s_k)}{n} \left(s_{ki} s_{kj} - s_{ki} h_{kj} - h_{ki} s_{kj} + h_{ki} h_{kj} \right) \tag{9}$$

where

$$h_{ki} = \sum_r J_{ir} s_{kr} \tag{10}$$

and states that if we use

$$c = \left(1 - \frac{1}{n} \sum_{ij} s_{ki} J_{ij} s_{kj} \right)^{-1} . \tag{11}$$

then we implement an incremental form of the pseudo-inverse rule via the partitioned inverse equations.

For the pseudo-inverse, this rule still involves non-local computation for computing c. This General Rule can be seen as the basis for the Storkey Learning Rules that follow. In this rule it is beneficial to maintain diagonal J_{ii} terms for the learning procedure, though we remove them when neural updates are made at test time. As the J_{ii} terms are local to an individual neuron, this does not affect the locality of the rule.

2.5.2 First Order Storkey Rules

We focus on the simplest form of rule, as that provides the most interesting properties. We refer the reader to [10] to see the variations. Suppose in (9), we write $c \approx 1$, and assume that h is smaller than s, so $h_{ki} h_{kj}$ terms can be neglected, we end up with the first order Storkey Learning Rule.

$$J_{ij} \to J_{ij} + \frac{1}{n} s_{ki} s_{kj} - \frac{1}{n} s_{ki} h_{kj} - \frac{1}{n} h_{ki} s_{kj} \tag{12}$$

2.5.3 Second Order Storkey Rules

The Extended Storkey Learning Rule, or second-order Storkey Learning Rule keeps the $h_{ki}h_{kj}$, but takes the $c \approx 1$, giving

$$J_{ij} \to J_{ij} + \frac{1}{n}s_{ki}s_{kj} - \frac{1}{n}s_{ki}h_{kj} - \frac{1}{n}h_{ki}s_{kj} + \frac{1}{n}h_{ki}h_{kj} \tag{13}$$

where

$$h_{ki} = \sum_r J_{ir}s_{kr} \tag{14}$$

2.5.4 Discussion

Compared with the pseudo inverse rule the Storkey Rules result in some patterns are overtrained and some are undertrained. However this can be an advantage: under the pseudo-inverse, later patterns would typically have smaller c than the larger patterns. However eventually the network would hit capacity and forget all the memories. By keeping c fixed, the network favors remembering recent memories at expense of earlier ones. The Storkey learning rules are called palimpsest rules for this reason. Alternatively the Extended Storkey Learning Rule can be seen as doing a Hebb update using the residual from the one step update (viewing s_{ki} as a real value).

$$J^+_{ij} \to J_{ij} + (s_{ki} - h_{ki})(s_{kj} - h_{kj}) \tag{15}$$

All the forms of Storkey Rules are fully local, in that the synaptic dynamics only need information available from the adjacent neurons. However they do require communication for computing the local fields.

2.6 Properies

Up to the point the various properties of learning rules have only been mentioned. The following are issues in Hopfield Networks:

- Generalisation and Spurious memories
- Capacity
- Locality of Learning
- Incrementality of learning
- Catastrophic Behaviour

2.6.1 Generalisation and Spurious Memories

A Hopfield Network with non-zero diagonal has the facility to learn the identity matrix. This is undesirable as it leads to an inability to generalize: the noisy

patterns will be mapped to noisy patterns and not to their noise-free memories. Hence larger basins of attraction for a learning rule is better.

Spurious memories are fixed points of the Hopfield network that are not original training vectors. Spurious memories can be good or bad depending on their characteristics. Spurious memories may generalize memories to new, but reasonable vectors that capture the distribution of the problem. However they may also be undesirable confounders. Which of these is true depends on the representation and the location of the spurious memories.

2.6.2 Capacity

Capacity refers to the number of memories that can be stored by a learning rule. In general higher capacity is better. The capacity of a network depends on the error rate that is allowed, how correlated patterns might be, and the learning rule. However capacity must not occur at the expense of generalization. Hence all capacity calculations must be computed for zero diagonal J_{ij}.

2.6.3 Local Learning

A learning rule is local if the learning process for a weight depends on information that can be made available at the two adjacent nodes. This information can be information from the patterns presented at the nodes, or from the result of the neural dynamics at the nodes.

2.6.4 Incremental Learning

Incremental learning is what we would normally call online learning: the update process can be done incrementally and does not require a batch process, or reference to previous memories. This is of great importance in modern systems for handling large data sets and streaming data. An incremental rule, in this context, achieves the required learning goal in an incremental way: it is not just using online streamed data as a proxy for multiple passes through a batch dataset.

2.6.5 Catastrophic Behaviour

As learning in Hopfield Networks progresses, it is possible for catastrophic forgetting to occur, where none of the previous learnt memories are stored in the system.

3 Experiments

We performed tests on storing image patches in a Hopfield Neural Network. Standard 16 by 16 image patches randomly sampled from natural images were truncated to 6 bit depth and represented in binary, resulting in a 1536 length binary vector. Each was then combined with a random binary vector of the

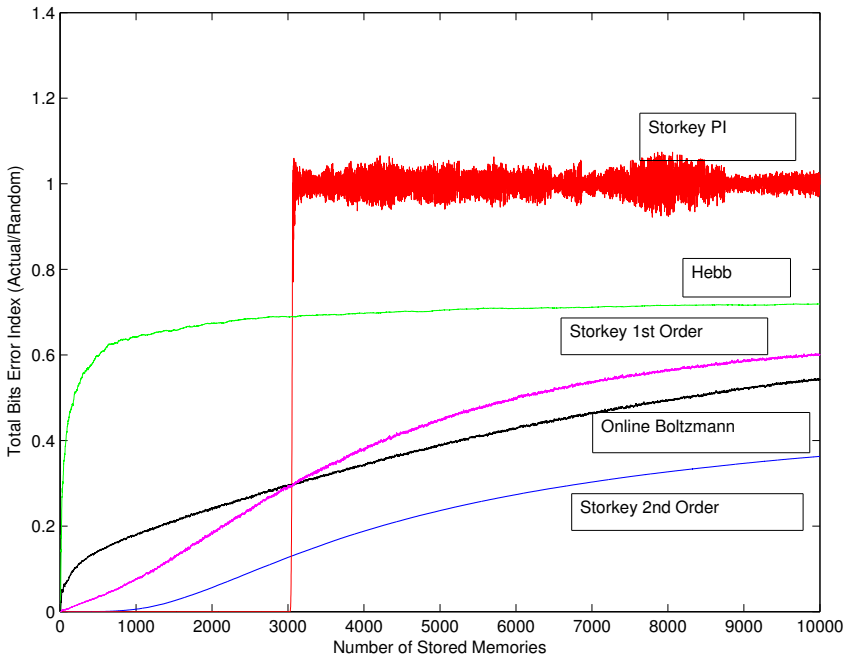

Figure 1: Capacity Calculations: Total number of erroneous bits on attempted recall of all patterns as a proportion of the number of erroneous bits expected for randomly generated recall vectors.

same length, and the resulting vectors were stored in a Hopfield model with full connectivity, using each of the above learning rules. The memories were then tested for recall and the total number of bits error was reported. This was compared with the bit error expected if recalled memories had just been random vectors. This was done for all incremental forms of Hopfield learning rules to compare like with like. We have not included the incremental form of the perceptron learning rule as that fails almost immediately.

The results are presented in Figure 1. We note that though the Storkey incremental version of the Pseudo Inverse works well for a number of patterns less than the number of neurons, it has a catastrophic failure afterwards. The Hebb rule is particularly poor. The second order Storkey Rule performs better than all other methods for all memory loadings considered.

4 Comments

It has been a long time since the original work by Hopfield, and since the improved learning schemes introduced by Storkey. Since then Hopfield Networks

have been of little interest in the machine learning and neural networks communities: interest has been primarily relegated to analogical use within certain areas of psychology and computational neuroscience. However given the close similarities between Hopfield Networks and Boltzmann machines, perhaps the new interest in Boltzmann machines within a deep learning context will produce with it increased awareness of Hopfield networks, and a renewed interest in the various learning algorithms associated with them.

References

[1] J.J.Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America: Biological Sciences*, 79:2554–2558, 1982.

[2] Hopfield networks. Wikipedia.

[3] C. Johansson. *Toward Cortex Isomorpic Attractor Neural Networks*. PhD thesis, KTH, Stochholm, Sweden, 2004.

[4] T. Hannagan. *Visual Word Processing: Holographic Representations and Attractor Networks*. PhD thesis, Université Paris 6, Paris, France, 2009.

[5] K. Swingler. On the capacity of Hopfield neural networks as EDAs for solving combinatorial optimisation problems. In *IJCCI*, pages 152–157, 2012.

[6] A.J. Storkey. Increasing the capacity of the hopfield network without sacrificing functionality. In *Proc. ICANN97*, 1997.

[7] A.J. Storkey and R. Valabregue. A new Hopfield learning rule with high capacity storage of correlated patterns. *Electronics Letters*, 33:1803–1804, 1997.

[8] A.J. Storkey. Palimpsest memories: A new high capacity forgetful learning rule for Hopfield networks. Technical report, Imperial College, 1998.

[9] A.J. Storkey and R. Valabregue. The basins of attraction of a new Hopfield learning rule. *Neural Networks*, 12:869–876, 1999.

[10] A.J. Storkey. *Efficient Covariance Matrix Methods for Bayesian Gaussian Processes and Hopfield Neural Networks*. PhD thesis, University of London, 1999.

[11] Neil Davey, Stephen P Hunt, and RG Adams. High capacity recurrent associative memories. *Neurocomputing*, 62:459–491, 2004.

[12] Neil Davey and Rod Adams. High capacity associative memories and connection constraints. *Connection Science*, 16(1):47–65, 2004.

[13] Neil Davey, Lee Calcraft, and Rod Adams. High capacity, small world associative memory models. *Connection Science*, 18(3):247–264, 2006.

[14] Jinde Cao, Anping Chen, and Xia Huang. Almost periodic attractor of delayed neural networks with variable coefficients. *Physics letters A*, 340(1):104–120, 2005.

[15] Petru Lucian Cur&scedil et al. Emergent states in virtual teams: a complex adaptive systems perspective. *Journal of Information Technology*, 21(4):249–261, 2006.

[16] Ben Goertzel, Joel Pitt, Matthew Ikle, Cassio Pennachin, and Liu Rui. Glocal memory: A critical design principle for artificial brains and minds. *Neurocomputing*, 74(1):84–94, 2010.

[17] Mahmood Amiri, Hamed Davande, Alireza Sadeghian, and Sylvain Chartier. Feedback associative memory based on a new hybrid model of generalized regression and self-feedback neural networks. *Neural networks*, 23(7):892–904, 2010.

[18] Hamed Davande, Mahmood Amiri, Alireza Sadeghian, and Sylvain Chartier. Auto-associative memory based on a new hybrid model of sfnn and grnn: Performance comparison with ndram, art2 and mlp. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1698–1703. IEEE, 2008.

[19] Christopher Johansson. *An attractor memory model of neocortex*. PhD thesis, Norwegian University of Science and Technology, 2006.

[20] JC Sylvester, JA Reggia, SA Weems, and MF Bunting. Controlling working memory with learned instructions. *Neural Networks*, 2013.

[21] Abbas Edalat and Federico Mancinelli. Strong attractors of hopfield neural networks to model attachment types and behavioural patterns. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2013.

[22] F.V. Agakov and D. Barber. Temporal hidden Hopfield models. *Institute for Adaptive and Neural Computation*, 2002.