CA-DSP'89

1989 International Symposium on

Computer Architecture and Digital Signal Processing

11-14 October, 1989
Hong Kong Convention and Exhibition Centre
Hong Kong

Symposium Proceedings
Volume 1 of 2 : Regular Session

IEE

Organised by
The Institution of Electrical Engineers
Hong Kong Centre

International Symposium on Computer Architecture and Digital Signal Processing. October. 1989. Hong Kong

Bernoulli Clamping in Alternating Projection Neural Networks

Kwan F. Cheung
Dept. Information Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
Seho Oh, Robert J. Marks II and Les E. Atlas
Interactive System Design Laboratory, FT-10 University of Washington, Seattle, Wa 98195., USA

Abstract

Alternating projection neural networks (APNN's) have been shown to be effective architectures for content addressable memories when a portion of a stored vector is known. Neural states corresponding to the known portion of the vector are imposed or *clamped* to the known values. The remaining *floating* neurons iteratively converge to the extrapolation of the vector. In this paper, this same notion is applied to binary vectors which, when presented for the content addressable operation, are corrupted by Bernoulli (or *flip*) noise. The procedure of the APNN is utilized with all of the neurons initially in the floating state. When the binary value of a neuron is the same in two consecutive iterations, the neuron is clamped to that value with the probability p. Otherwise, the neuron remains floating. The performance of the APNN using *Bernoulli clamping* is contrasted quite favorably with other associative memory artificial neural networks.

1. Introduction

The *alternating projection neural network* [?]-[?] is an effective architecture for a content addressable memory. That is, a memorized training vector may be reconstructed when an arbitrary portion of the vector is presented to the neural network. In this paper, we show that the APNN can also be effectively used as an associative memory when a library vector is corrupted by noise. The APNN is shown empirically to perform better than other associative memory artificial neural network [?]- [?].

2. The APNN

A brief review of the APNN is appropriate. More detailed discussions are available elsewhere [?]-[?].

Consider a set of N library vectors:

$$\Im = \{\vec{f}_n | 1 \leq n \leq N\}$$

Each library vector has a length $L > N$. We form the library matrix

$$\mathbf{F} = \left[\vec{f}_1 \vdots \vec{f}_2 \vdots \cdots \vdots \vec{f}_N\right]$$

and the neural network's interconnect matrix

$$\mathbf{T} = \mathbf{F}(\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T$$

where T denotes matrix transposition. Let $\vec{g} \in \Im$. We partition $\vec{g}$ into the vector $\vec{g}_p$ and $\vec{g}_q$ where $\vec{g}_p$ contains the first P elements of $\vec{g}$ and $\vec{g}_q$ the remaining $Q = L - P$. Let $\mathbf{T}_4$ denote the $Q \times Q$ matrix taken from the lower right hand corner of $\mathbf{T}$ and $\mathbf{T}_3$ the $Q \times P$ matrix taken from the lower left hand corner of $\mathbf{T}$. The iteration

$$\vec{s}_{m+1} = \mathbf{T}_4\vec{s}_m + \mathbf{T}_3\vec{g}_q$$

converges to the extrapolation $\vec{s}_\infty = \vec{g}_p$ for any initialization if the first P rows of $\mathbf{T}$ form a matrix of full rank. In the parlance of the APNN, the neural states corresponding to the known portion of the library matrix, $\vec{g}_q$, are *clamped* to the known values and the remainder of the neurons *float*. The role of a neuron as clamped or floated can change from application to application.

3. Bernoulli Clamping

By slightly perturbing the APNN algorithm, we can also recall library vectors when the known data is corrupted by noise. For this extension, we will limit our library to vectors containing ± 1's. Hence, each member of $\mathbf{F}$ is either $+1$ or -1. As before, let $\vec{g} \in \Im$. We will corrupt $\vec{g}$ with *flip noise*[1]. Specifically, with some "small" probability π, each element of $\vec{g}$ can be reversed. Let this corrupted library vector be $\vec{h}$. Then

$$h_n = (-1)^{\beta_n} g_n$$

where h_n is the n^{th} element of $\vec{h}$ and β_n is a Bernoulli random variable, i.e.

$$\begin{aligned} \text{Probability}[\beta_n = 1] &= \pi \\ \text{Probability}[\beta_n = 0] &= 1 - \pi \end{aligned}$$

[1] Also called Bernoulli noise.

We assume each β_n is independent from every other one. Typically, π is significantly below 1/2. Our goal is to recover $\vec{g}$ from $\vec{h}$ using a modified APNN. Our proposed algorithm is as follows:

1. Set $M = 1$ and $\vec{u}_M = \vec{h}$. Initially, all neurons are floating.

2. Compute

$$\vec{w}_M = \mathbf{sign}\, \mathbf{T}\vec{u}_M$$

   where **sign** is a pointwise operator which extracts the sign of each element of the vector on which it operates. (Hence, all the elements of $\vec{w}_M$ are ± 1). Let the n^{th} element of the vector $\vec{v}_M$ be equal to the n^{th} element of $\vec{w}_M$ if the neuron is floating and the n^{th} element of $\vec{u}_M$ if it is clamped.

3. If the n^{th} element of $\vec{v}_M$ is different than the n^{th} element of $\vec{u}_M$, then the corresponding neuron remains floating. If they are the same, the neuron is clamped to the value with probability p. (Thus, on the average, a fraction p of the neurons with the unchanged states are clamped and the remainder of the neurons with unchanged states remain floating.) Once a neuron is clamped, it remains clamped. We refer to this procedure *Bernoulli clamping*.

4. With the current partition of floating and clamped neurons, apply the conventional APNN until the sign of the neural states does not change. Let $\vec{u}_{M+1}$ be this steady state vectors of ± 1's.

5. Set $M = M + 1$, go to Step 2 and repeat until convergence.

Note that Step 3 is akin to simulated annealing [6]. We have here chosen, not to vary p with each iteration.

4. Examples

In this section, we present examples illustrative of the observed properties of this associative memory. In the examples to follow, the $N = 4$ letters, {I,S,D,L}, were coded as ± 1's in a 5×7 array. Thus, L=35.

Example 1: The letter L corrupted by flip noise is shown in Figure 1a. Figure 1b is a plot of the Hamming distance between this corrupted library vector and the other library vectors. Although closest to the letter L, $\vec{h}$ is also close to both S and D. Shown in Figure 1c and 1d are the results of $M = 1$ and $M = 4$ iterations using $p = 0.75$. The correct library vector is restored.

Example 2: This example illustrates that a choice of a large value of p is typically bad. The corruption is identical to that in Example 1. Use of $p = 1$ is shown in Figure 2 give an incorrect result. Use of $p = 0.75$ in Example 1 gave the correct result.

Example 3: In this example, we contrast the performance of the APNN using Bernoulli clamping with that of Hopfield's associative memory neural network [1]–[3]. Let

$$\wp = \{\vec{b}_n | 1 \le n \le N\}$$

denote a set of N binary $(0,1)$ library vectors. Let

$$\mathbf{B} = \left[\vec{b}_1 \vdots \vec{b}_2 \vdots \cdots \vdots \vec{b}_N\right]$$

We form the interconnect matrix

$$\underline{\mathbf{T}} = (2\mathbf{B} - \mathbf{1})(2\mathbf{B} - \mathbf{1})^T$$

where $\mathbf{1}$ is a matrix of 1's. Let $\vec{e} \in \wp$. Let $\vec{e}$ be corrupted with flip noise to form $\vec{k}$. In synchronous form [7], Hopfield's model claims restoration of $\vec{e}$ from $\vec{k}$ with the iteration

$$\vec{t}_{M+1} = U\,\underline{\mathbf{T}}\vec{t}_M$$

with initialization $\vec{t}_1 = \vec{k}$ and where the vector operator U takes the unit step of each element of the vector on which it operates.

Shown in Figure 3a is a flip noise corrupted version of the letter L. The APNN with Bernoulli clamping restored the object. Hopfield's neural network did not. Indeed, as shown in Figure 4, the Hopfield neural network can even converge to the improper result, in this case the library vector corresponding to the letter S.

5. Conclusion

We have demonstrated that when augmented with Bernoulli clamping, the APNN can be used as an associative memory for recall of inputs corrupted by flip noise. As was illustrated in the toy problem of recall of four stored letters, it can significantly outperform the Hopfield artificial neural network associative memory.

Acknowledgements

The authors gratefully acknowledge the support of this research by the Washington Technology Center. L. E. Atlas is also supported by a National Science Foundation Presidential Young Investigator's Award.

References

[1] R. J. Marks II, "A class of continuous level associative memory neural nets", **Applied Optics** v.26, 1987, pp.2005–2009.

[2] R. J. Marks II, S. Oh, L. E. Atlas and J. A. Ritcey, "Homogeneous and layered alternating projection neural networks", **Real–Time Signal Processing for Industrial Application**, edited by Brahram Javidi (SPIE Optical Engineering Press, Bellingham, Washington 1989) pp.217- -232.

[3] R. J. Marks II, S. Oh and L. E. Atlas, "Alternating projection neural networks," **IEEE Transactions on Circuits and Systems**, (in press).

[4] J. J. Hopfield, "Neural networks and physical system with emergent collective computational abilities," **Proc. of the Nat'l Academy of Science, USA**, v.79, 1982, pp.2554–2558.

[5] R. J. Marks II and L. E. Atlas, "Geometrical interpretation of Hopfield's content addressable memory neural network," *invited paper*, **Northcon/88 Conference Record, Volume II**, Seattle, Washington, (Western Periodicals Co., North Hollywood, CA) Oct 1988.

[6] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," **IEEE Trans. on Pattern Analysis and Machine Intelligence**, v. 6 1984, pp.721–741.

[7] K. F. Cheung and R. J. Marks II, "Synchronous versus asynchronous behavior of Hopfield's content addressable memory," **Applied Optics**, v.26, 1987, pp.4808–4813.

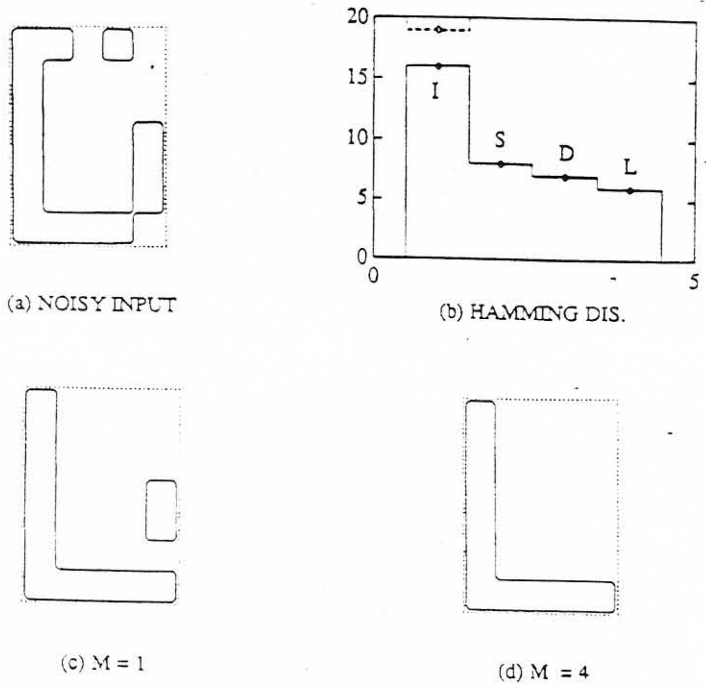

Figure 1 Four binary letters, ISDL, are coded on 5×7 grid. The letter L, corrupted by flip noise, is shown in (a). The Hamming distance between this perturbation and each letter is shown in (b). After four cycles in the APNN using Bernoulli clamping with $p = 0.75$, the letter is restored.

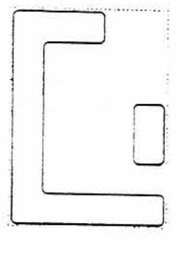

Figure 2 This is the result of the same procedure on the same problem as that described in the caption in Figure 1 except $p = 1$. The steady state result is clearly incorrect.

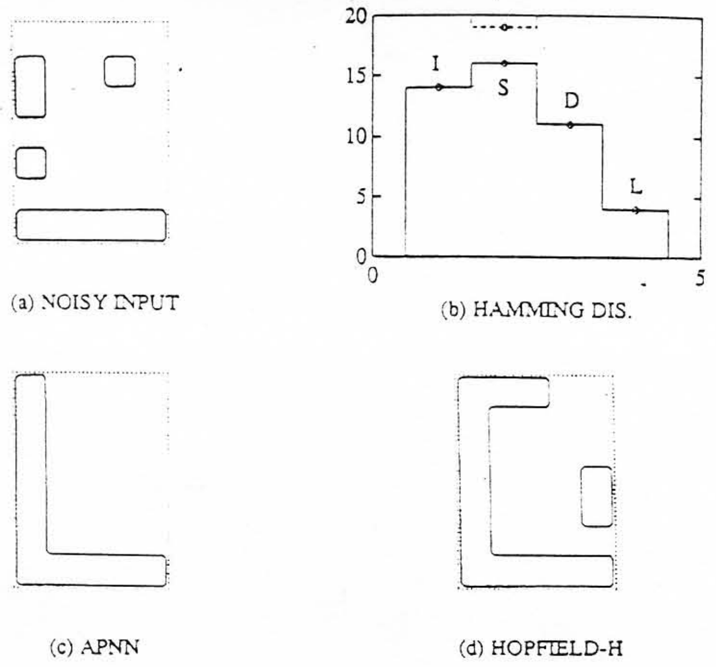

Figure 3 Comparison of the performance of a Hopfield artificial neural network to an APNN with Bernoulli clamping. The input in (a) is an L corrupted by flip noise. The APNN result, shown in (c), gives the proper answer. The Hopfield result in (d), does not.

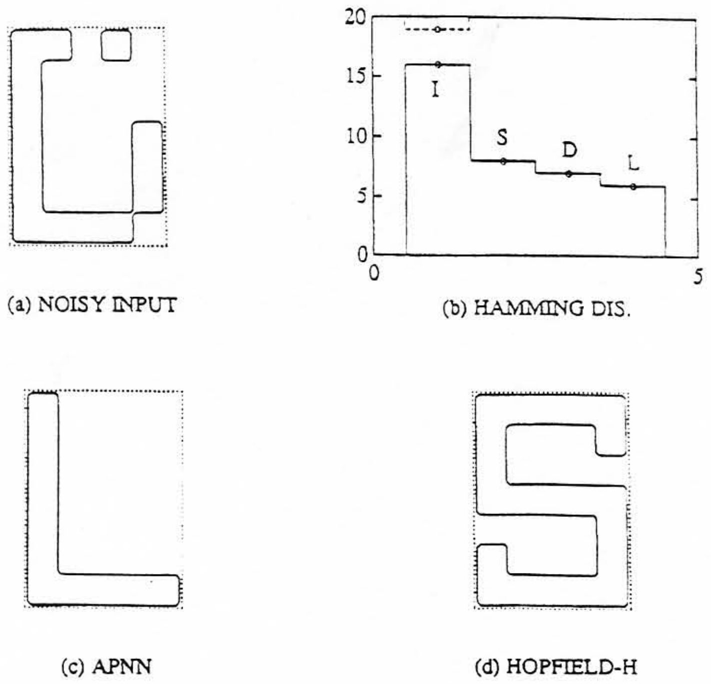

Figure 4 This is the same problem as is described in the caption of Figure 3, except using different noise on the input. The APNN with Bernoulli clamping again succeeded in restoring the perturbed input. The Hopfield model converged to the letter S which, although in the library, is not the proper result.