ELSEVIER

Neurocomputing 26–27 (1999) 237–245

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Towards the memory capacity of neurons with active dendrites

Panayiota Poirazi*, Bartlett W. Mel

Department of Biomedical Engineering, University of Southern California, Los Angeles, CA 90089, USA

Accepted 18 December 1998

Abstract

Active dendrites can be viewed as linear classifiers augmented by a few second-order product terms representing multiplicative synaptic interactions [4]. To quantify the degree to which local synaptic interactions could augment the memory capacity of a neuron, we have studied the family of "subsampled quadratic" (SQ) classifiers. Each SQ classifier is a linear classifier augmented by a subset k of the $K = O(d^2)$ second-order product terms available in d dimensions. Using a randomized classification task, we show that the error rate of an SQ classifier depends only on: (1) the product term ratio $p = k/K$, which identifies a family of isomorphic classifiers, and (2) the number of bits contained in the SQ classifier's specification. Finally, we quantify the increase in memory capacity of any SQ classifier relative to its linear counterpart.

Keywords: Neural modeling; Subsampled quadratic classifiers

1. Introduction

Empirical studies indicate that many neuron types are electrically "active", i.e. their dendrites contain voltage-dependent ionic conductances that can lead to full regenerative propagation of action potentials and other dynamical phenomena [9]. In previous work, we developed a single-neuron model, which includes location-dependent multiplicative synaptic interactions [4,5] and showed that active dendrites could boost the pattern discrimination/memory capacity of a neuron. These results suggest that an active dendritic tree behaves like a high-dimensional quadratic classifier which

* Corresponding author.
E-mail addresses: poirazi@lnc.usc.edu (P. Poirazi), mel@lnc.usc.edu (B.W. Mel)

0925-2312/99/$ – see front matter
PII: S0925-2312(99)00078-8

contains only a small subset of the possible second-order interaction terms. Though it is known that higher-order terms can increase the power of a learning machine for both regression and classification [1,8,3,7], existing theory regarding the learning capabilities of quadratic classifiers is limited.

In this work, we try to quantify the degree to which inclusion of varying numbers of non-zero quadratic terms augments the capacity of a "subsampled quadratic" (SQ) classifier relative to its linear counterpart. We study the relations between (i) the input space dimensionality, (ii) the number of non-zero product terms available to an SQ classifier, (iii) the difficulty of the classification problem, and (iv) the resulting boost in classification performance relative to a linear classifier on a randomized benchmark learning problem.

2. Methods

2.1. The classification problem

We adopt the classification problem in which N patterns are drawn randomly from a d-dimensional zero-mean unit-variance Gaussian distribution G, and half of the patterns are randomly assigned to each of the two classes T_{pos} and T_{neg}. Since all N patterns are drawn from a single distribution, discrimination between T_{pos} and T_{neg} becomes arbitrarily difficult as N grows large. We quantify the memory capacity of a classifier on this benchmark task by a graph of the classification error versus the training set size N.

2.2. Subsampled-quadratic classifiers

We considered the family of SQ classifiers which contain only a subset k of the $K = (d^2 + d)/2$ available second-order terms in d dimensions. When $k = 0$ we have a pure linear classifier, while $k = K$ corresponds to a "full" quadratic classifier. In all simulations reported here, the coefficients for the k non-zero quadratic terms were determined as follows. A conjugate gradient algorithm was used to train a full quadratic classifier to minimize mean squared error (MSE) over the training set. Given the sphericity of the training set distribution, the selection of the k "best" product terms after training could be made based on weight magnitude. It was verified empirically that the increase in MSE or classification error when a single weight was set to zero grew monotonically with the weight magnitude. The pruned classifier was then retrained to minimize MSE and the output was passed through a sigmoidal thresholding function $(1 - e^{-x/s})/(1 + e^{-x/s})$ with slope $s = 0.51$.

3. Results

3.1. The linear case

We derived an analytical expression for the performance curve of a pure linear classifier in the limit of large N, based on the assumption that the positive and

negative training sets T_{pos} and T_{neg} were themselves spherical, unit-variance Gaussian blobs G_{pos} and G_{neg}, with means $\bar{X}_{pos}$ and $\bar{X}_{neg}$ slightly shifted from the origin. Assuming the optimal discrimination surface was a hyperplane cutting perpendicularly halfway between the means, we found that in the limit of high dimension

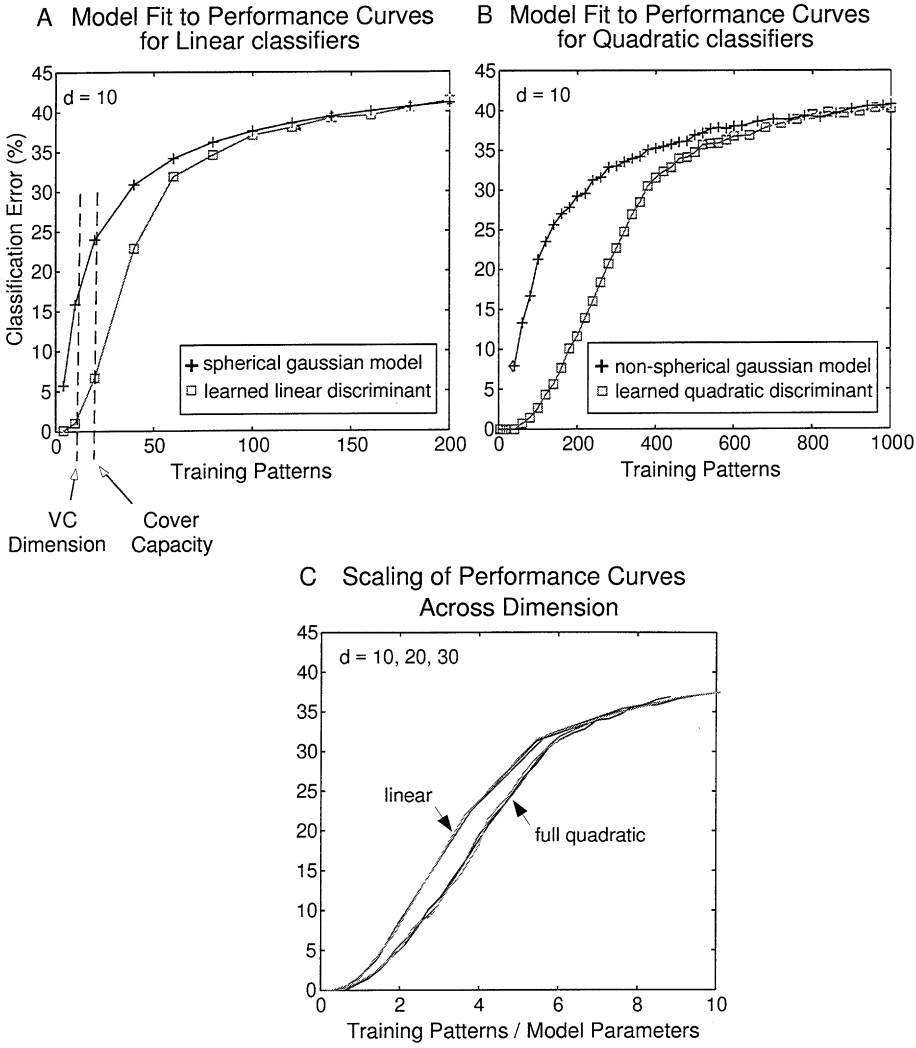


Fig. 1. A. Analytical model (crosses) fit numerical simulations results (squares) in the asymptote of large N. The dashed lines indicate VC dimension ($N = 1 + d$) and Cover capacity ($N = 2d$) for the Perceptron, which in this problem corresponds to approximately 1% and 7% error rates, respectively. B. Bayes-optimal classifier (crosses) fit full-quadratic classifier (squares) in the limit of large N. C. Scaling of linear and quadratic performance curves across dimension, when plotted against training patterns per model parameter ($1 + d$ for linear, $1 + d + (d^2 + d)/2$ for quadratic).

with $N \gg d$, the expected classification error is

$$CE = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{d}{2N}}\right), \tag{1}$$

where σ^2 is the variance of the original generating distribution ($\sigma = 1$ in this case) and the complementary error function is defined by $\mathrm{erfc}(z) = (2/\sqrt{\pi})\int_z^\infty e^{-t^2}\,dt$, with $z = x/\sqrt{2}\sigma$. Pearlmutter has previously analyzed the similar "deja vu" learning problem [6].

This expression was compared to the results of computer simulations for 10-dimensional random training sets of various sizes. As expected, the expression proved valid in the asymptote of large training sets (i.e. $N > 10d$) as shown in Fig. 1A. Departures from the analytical model for small training sets were due to violation of the spherical Gaussian assumption, i.e. where the optimization routine could adjust the discriminating hyperplane to capitalize on geometric idiosynchracies of the sparse training set. As is evident in Eq. (1), the dependence of classification error on N and d appears only as a ratio, consistent with the linear dependence on d of the VC dimension ($N = 1 + d$) and Cover capacity ($N = 2d$) of a Perceptron. Each of these scalar capacities is indicated by a dashed line in Fig. 1A.

3.2. The full-quadratic case

As a geometric control for the full quadratic classifier, we modeled T_{pos} and T_{neg} as non-spherical Gaussian blobs by estimating their covariance matrices from the data. In this way, we could test the Bayes-optimal hyperquadric discriminant function [2] to compare with the results of gradient-based parameter optimization on the raw data. The results were similar to the linear case (Fig. 1B).

3.3. Scaling of linear and quadratic performance curves with dimension

Consistent with the invariance of Eq. (1) to the ratio of training patterns to classifier parameters in the limit of large d, the trained hyperplane performance curves for 10, 20, and 30 dimensions fall into precise superposition when plotted as a function of training patterns per model parameter $N/(1 + d)$, even for $N \ll d$, in the region where the Gaussian assumption breaks down (Fig. 1C).

Similarly for the full-quadratic case, we found empirically that the performance curves fell into superposition across dimension when plotted as a function of patterns per total parameters, in this case $N/(1 + d + (d^2 + d)/2)$. This scaling relation again held up for small N where the Gaussian assumption was not valid (Fig. 1C). We also noted that the linear and full-quadratic performance curves did not scale to each other in a simple way, such as by normalizing by the number of classifier parameters – their shapes were fundamentally different (Fig. 1C).

3.4. Subsampled-quadratic classifiers

Empirical performance curves for SQ classifiers for a range of k values in 10 dimensions are shown in Fig. 2A. As expected, the SQ error curves fall within the

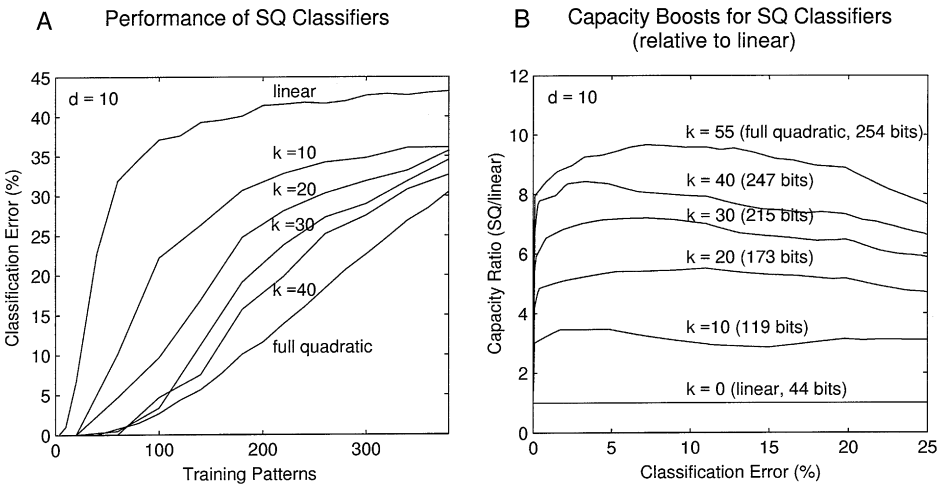


Fig. 2. Performance of SQ classifiers. A. Performance curves for various values of k in a 10-dimensional input space. B. Ratio of the memory capacity of SQ classifiers in 10 dimensions relative to a linear classifier $k = 0$, plotted as a function of the error rate.

upper and lower bounds provided by the linear and full-quadratic cases. The capacity boost achieved by addition of second-order terms to a linear classifier can be read off the graph by choosing a fixed error rate, and reading off the associated storage capacities for the linear versus various SQ curves. Ratios of storage capacity relative to the linear classifier ($k = 0$) are shown in Fig. 2B. For example, at fixed 1% error rate in 10 dimensions, the addition of the 10 best quadratic terms to the 11 linear and constant terms increases the trainable memory capacity by more than a factor of 3.

3.5. *Scaling relations for SQ classifiers*

Given that the error surface for SQ classifiers is three-dimensional over the parameters N, d and k, we sought scaling relations or other invariances that would allow this error surface to be described by fewer underlining variables. Both linear and full-quadratic performance curves could be brought into superposition across dimension by a simple normalization of the x-axis to reflect "patterns per classifier parameter". Unfortunately, this strategy failed for intermediate SQ classifiers. We found empirically, however, that two SQ performance curves could be consistently brought into register across dimension with an x-axis scaling, but only when their product term ratios were equal, i.e. $p = k_1/K_1 = k_2/K_2$, indicating that a key geometric invariance in the SQ family involves the proportion of available second-order terms included in the classifier (Fig. 3A). It remained to determine whether the value of this scaling factor could be consistently defined for the entire SQ family. In a search for the correct scaling factor, we hypothesized that it involved the number of bits needed to

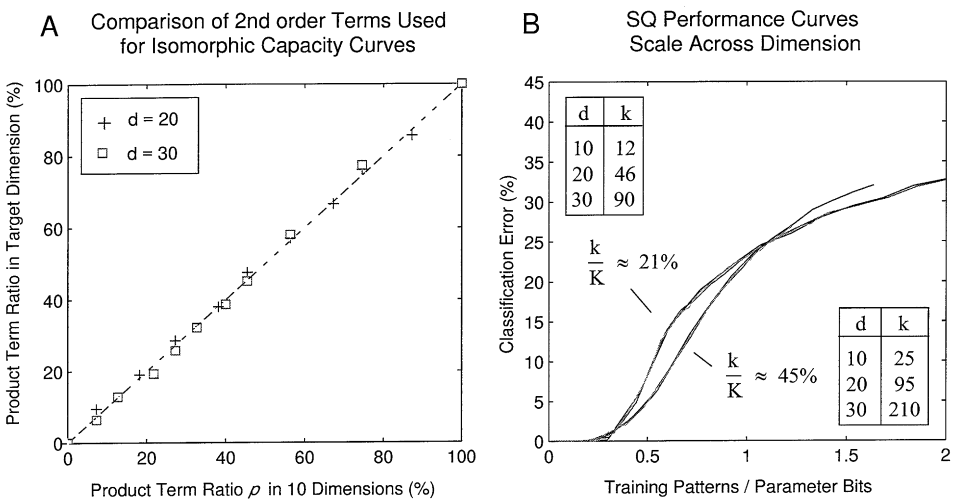


Fig. 3. A. Relationship between SQ classifiers in different dimensions whose performance curves were isomorphic. A linear relation (equality) holds between the product term ratios $p = k/K$ for two isomorphic SQ classifiers. B. Superimposed scaled performance curves for 10, 20 and 30 dimensional input spaces, for $p \approx 21\%$ and 45%. (Approximation arises from quantization of k values.) When training patterns are drawn uniformly from the unit hypercube, SQ classifiers with equal p values continue to scale across dimension according to their bit totals. However, Gaussian and uniform curves have different shapes (not shown). For all curves shown here, the x-axis was scaled by the total number of classifier bits, as given by Eq. (2).

specify an SQ classifier:

$$B(k,d) = (1 + d + k)w + \log_2\binom{K}{k}, \tag{2}$$

where the first term specifies the number of bits needed to encode the explicit weight values (with w indicating bits per weight), and the second term specifies the number of bits needed to specify *which* k of the available K product terms is used. We found empirically that by choosing $w = 4$, Eq. (2) gave the correct scaling factor relating SQ classifier performance curves for any k and d. This value for w was confirmed in a separate experiment where we found that quantization of weights with fewer than four bits resulted in a sharp increase in the classification error. Examples of normalized performance curves are shown in Fig. 3B for two values of p. Conveniently, we note that for $k = 0$ and K the combinatorial term in Eq. (2) drops out, so that the capacity scaling factor becomes directly proportional to the number of *explicit* weight parameters. This explains the success of simple per-parameter scaling for linear and full-quadratic classifiers, such as was used to generate the inter-dimensional correspondences shown in Fig. 1C.

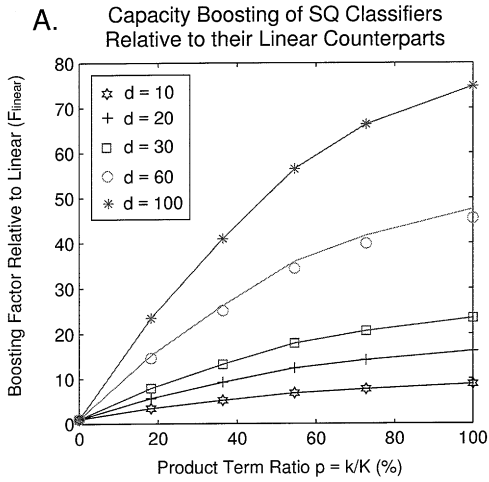


Fig. 4. Boosting of SQ memory capacity relative to linear counterpart at a fixed classification error rate $\varepsilon = 1\%$. Plot of F_{linear} shows the increase in capacity of an SQ classifier relative to the corresponding linear classifier. For fixed p-ratio, the boost factor growth is O(d).

3.6. SQ classifier performance relative to linear: how large the boost?

Based on a single set of SQ performance curves in a reference dimension d_r, we derived an expression to predict the boost factor that relates the memory capacity of any $SQ_{k,d}$ classifier in any dimension to the capacity of a linear classifier in the same dimension, for a given error rate ε:

$$F_{\text{linear}}(p, d, \varepsilon) = \hat{F}_{\text{linear}}(p, d_r, \varepsilon)\,\frac{1 + d_r}{1 + d}\,\frac{B(k,d)}{B(k_r,d_r)}, \tag{3}$$

where $\hat{F}_{\text{linear}}$ is measured empirically from a reference curve; for example, a range of values of $\hat{F}_{\text{linear}}$ for $d_r = 10$ are shown in (Fig. 2B). This boost factor is valid for any $SQ_{k,d}$ classifier with p-ratio equivalent to that of an SQ classifier in the reference dimension d_r. As seen in the above equation, the boost factor follows the ratio of classifier bits consumed by the SQ vs. linear classifiers and grows as O(d) since the capacity of the SQ classifier with fixed p-ratio grows as O(d^2), while that of the linear classifier grows as O(d) (Fig. 4A).

4. Discussion

Our main result is that, for the learning problem posed here, the error rate of a subsampled quadratic classifier with k non-zero product terms in d dimensions depends only on 2 variables: (1) the product term ratio $p = k/K = 2k/(d^2 + d)$, which specifies the family of geometrically equivalent learning machines, and (2) the number

of training patterns per classifier bit, which keys the absolute performance level to the classifier capacity. One clear conclusion by examining this representation of the SQ family is that significantly different error rates can arise from classifiers even when fully normalized for capacity (as measured by total classifier bits), depending on geometric aspects of the learning problem. Another result involves the expression for the total bits needed to specify an SQ classifier. As seen in Eq. (2), the absolute capacity of an SQ classifier depends on having a choice as to which coefficients to include in the classifier, in addition to the *values* of the coefficients. Furthermore, we have shown that we may estimate based on the graph in Fig. 4A an upper limit on the degree to which active dendrites could augment the capacity of a neuron relative to its Perceptron-like counterpart.

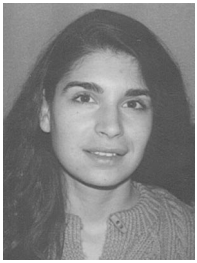
Our results take us a step closer to our goal of understanding how multiplicative interactions among neighboring synapses on a dendritic tree can increase the learning capacity of a neuron.

Acknowledgements

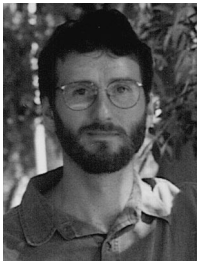
Thanks to Barak Pearlmutter and Dan Ruderman for helpful discussions on this work.

References

[1] R.L. Barron, A.N. Mucciardi, F.J. Cook, J.N. Craig, A.R. Barron, Adaptive learning networks: development and applications in the united states of algorithms related to gmdh, in: S.J. Farlow (Ed.), Self-Organizing Methods in Modeling, Marcel Dekker, New York, 1984.
[2] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[3] I. Guyon, B. Boser, V. Vapnik, Automatic capacity tuning of very large vc-dimension classifiers, in: S.J. Hanson, J.D. Cowan, C.L. Giles (Eds.), Neural Info. Process. Systems, vol. 5, Morgan Kaufmann, San Mateo, CA, 1993, pp. 147–155.
[4] B.W. Mel, The clusteron: toward a simple abstraction for a complex neuron, in: J. Moody, S. Hanson, R. Lippmann (Eds.), Advances in Neural Information Processing Systems, vol. 4, Morgan Kaufmann, San Mateo, CA, 1992, pp. 35–42.
[5] B.W. Mel, D.L. Ruderman, K.A. Archie, Translation-invariant orientation tuning in visual 'complex' cells could derive from intradendritic computations, J. Neurosci. 17 (1998) 4325–4334.
[6] B.A. Pearlmutter, How selective can a linear threshold unit be?, in: International Joint Conference on Neural Networks, Beijing, PRC, November 1992, IEEE.
[7] T. Poggio, Optimal nonlinear associative recall, Biol. Cybernet. 9 (1975) 201.
[8] J. Schurmann, Pattern Classification: A Unified View of Statistical and Neural Approaches, Wiley, New York, 1996.
[9] G. Stuart, N. Spruston, B. Sakmann, M. Hauser, Action potential initiation and backpropagation in neurons of the mammlian cns, Trans. Inst. Neurosci. 20 (1997) 125–131.



Panayiota Poirazi received the Diploma in Mathematics from the University of Cyprus, Nicosia, Cyprus, in 1996. She obtained the M.S. degree in Biomedical Engineering in 1998 from the University of Southern California (USC), Los Angeles, and is currently pursuing the Ph.D. degree in Biomedical Engineering at the same institution. Her general research interests lie in the area of computational modeling with emphasis in the storage capacity of single-neuron models.



Bartlett Mel received his Ph.D. in Computer Science from the University of Illinois in 1989, followed by 5 years as a post-doctoral fellow at Caltech. He has been an Assistant Professor of Biomedical Engineering at USC since 1994. His research interests involve computational modeling of single cell function, and neurally-inspired algorithms for visual recognition.