ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 199-204

Application of High-order Boltzmann Machines in OCR

A. de la Hera, M. Graña, A. D´Anjou, F.X. Albizuri

Dept. CCIA, UPV/EHU+ Apartado 649, 20080 San Sebastián
e-mail: ccpgrrom@si.ehu.es

Abstract: This work presents the empirical results of the application of High-order Boltzmann Machines (HOBM) to the classification of handwritten characters (digits) in the context of the form based OCR system developed at the NIST.

0 Introduction

The Hsfsys [1] System has been developed in the Computer Systems Laboratory of the National Institute of Standards and Technology (NIST) as a state of the art reference System for OCR. The task performed by Hsfsys is form-based Optical Character Recognition (OCR). The NIST freely distributes the C source code in a CD-ROM that can be obtained by sending a letter of request to NIST. Library utilities are provided with the recognition System for conducting form registration, form removal, field isolation, field segmentation, character normalisation, feature extraction, character classification, and dictionary-based postprocessing. A host of data structures and low-level utilities are also provided (computing vertical and horizontal signatures, connected components, Least-Squares fitting, Karhunen-Loeve (KL) feature extraction, Probabilistic Neural Networks (PNN) classification [2],etc). The System portability has been tested on many UNIX workstations.

Within Hsfsys, the classification task is performed by a variation of the Probabilistic Neural Networks (PNN) proposed in [2]. This decision is based upon a series of experiments [3,4,5] comparing the performance of various classifiers on data extracted from the forms that the NIST also makes available on CD-ROM as the Special Database 19. To support our approach, we have partially reproduced these experiments with HOBM.

Our group has been working for a time on theoretical and application aspects of HOBM [6,7,8]. From a theoretical view we have found that HOBM with binary units show convergence properties fairly superior to conventional Boltzmann Machines (BM) with hidden units, and that the distribution modelled (learned) by the HOBM has a nice analytical characterisation, whereas the distribution modelled by the BM with hidden units is less clearly characterised. The convergence properties of binary HOBM can be shown to hold for the case of finite discrete state units, and they seem to hold (we lack a formal proof) for the case of continuous state units. From the application point of view, we have applied the HOBM to several databases finding

+ Work supported by the Dept. Educación, Univ. e Inv. of the Gobierno Vasco, project GV9220,and project UPV140.226-EA166/94 of the UPV/EHU

that learning in HOBM is extremely robust. Reducing the spectra of applications to that concerned with classification we have also find that the training of HOBM can be several times faster than that of the conventional BM, because we can avoid the use of simulated annealing to estimate the connection statistics. The main problem, not addressed in this paper, for their general practical application is the combinatorial growth of connections. The efficient topological design of HOBM is an open problem related to the pruning methods of more conventional neural networks, but with some peculiarities that deserve a dedicated study. The approach in this paper has been quite straightforward. We have applied densely connected HOBMs of order 3 to the feature vectors as computed in the Hsfsys. The results obtained on the database provided with the Hsfsys show the adequacy of this topology. After that we have embedded a fully trained HOBM within the Hsfsys to compare the performance of the modified System with the standard one.

Section 1 describes how the standard Hsfsys classifies segmented characters. Section 2 summarises our notation for HOBM. Section 3 discusses the experimental results obtained. Finally, section 4 gives some conclusions and further work

1 Character classification in the Hsfsys

The initial processing of a form involves the form verification and the recovery from small distortions (form registration). An empty template is subtracted form the corrected form to obtain the handwritten information. The segmentation process involves the isolation of the fields, the removal of noise from previous steps and the detection of the blobs that will be assumed as the characters to be recognised. Character composed of not connected blobs will not be correctly segmented in the general case. Too small or too big blobs are ignored and do not enter the recognition process. Character images, once segmented from the form, are size-normalised to 32x32 binary images. This normalisation includes erosion/dilation morphological operations to account for variations on stroke width. Normalised character images are sheared to account for variations in slant.

To extract features for classification, the Karhunen Loeve (KL) (Principal Components) transform is used. For the computation of the KL transform, the character images are considered as vectors of 32^2 binary elements. These vectors are obtained by concatenation of the image rows. As it is well-known, the KL transform is a linear transformation given by a selected subset of the eigenvectors of the covariance matrix. The selected eigenvectors are those with the maximum eigenvalues associated that account for a desired percentage of the total variance (sum of the eigenvectors). If the number of eigenvectors is too small the resulting transformed vectors, which serve as input feature vectors for the classifier, do not allow for the correct definition of discriminating functions (the classification problem is "unsolvable"). If too may eigenvectors are used there is no reduction of the complexity of the problem, although feature independence is guaranteed. The approach taken in the specification of the number of eigenvectors used in Hsfsys has been empirical: the performance of several classifiers has been observed as the number of eigenvectors used increases. The final selected number is 64. The standard Hsfsys is provided with a precomputed KL transform. The basis (eigenvectors) has been obtained from a database of character images (also provided) using standard diagonalisation methods.

The Hsfsys designers have decided to use the Probabilistic Neural Networks (PNN) originally proposed by Specht [2] as the classification System. Grossly speaking, PNN fall in the category of k-NN classifiers. The bare k-NN stores all the training patterns (training is therefore trivial) and classifies a pattern by majority voting among the k nearest neighbours to it. In PNN this voting is filtered exponentially. Formally, for each class a discriminant function is computed.

$$D_i(y) = \frac{p(i)}{M_i} \sum_{j=1}^{M_i} \exp\left(-\frac{1}{2\sigma^2} \left\| y - x_j^i \right\|^2 \right)$$

Where M_i is the number of training patterns in class i, σ^2 is the assumed variance around the training patterns and $\left\{ x_j^i \middle| j = 1..M_i \right\}$are the training patterns of class i. The main disadvantages of k-NN approaches, including PNNs, are the big storage needs and the high cost of computing the discriminant functions. The designers of Hsfsys have done some optimisations to alleviate the cost of the computation of the discriminant functions. They impose a tree structure to the database of training patterns. In the computation of the discriminant functions they use a pruning rule that excludes the less significant patterns. They also preempt the computation of very large Euclidean distances. In the way to alleviate the storage problem, the Hsfsys designers provide the possibility of implementing Hsfsys with a reduced set of prototype vectors, selected beforehand to minimise the performance degradation incurred.

2 Summary of HOBM definitions and algorithms

For the application at hand we have used HOBM with continuous valued input units. Such an HOBM is described by the quadruplet (**U, L, W, R**), where **U** is the set of units, **L** is the set of connections, **W** denotes the weights of the connections, and **R** denotes the state spaces of the units. Each connection $\lambda \in \mathbf{L}$ is a subset of **U**. Units connected by λ belong to λ. The order of a connection is the number of units that belong to it, and the order of the HOBM is that of its highest order connection. We denote with ω_λ the weight associated with connection λ. As usual for BM, the dynamics of the HOBM is the (stochastic) maximisation of the consensus function defined over the set of global states (configurations) $C(k) = \sum_\lambda \omega_\lambda \prod_{u \in \lambda} k(u)$, where k(u) denote the state of the unit u determined by the configuration k.

The topology of HOBM is not easy to visualise, as its model is an hypergraph instead of the more usual graph that model conventional (order 2) neural networks. A way to represent them graphically is to define two kinds of nodes: unit nodes and high-order connection nodes. The nodes that represent high-order connections correspond to the so called sigma-pi units. A very general class of topologies is the class of densely connected of a given order. A densely connected topology contains all the connections up to the specified order. In this paper we will only consider densely connected topologies of order 3. That means that the connections considered are the bias of the output units (order 1), all the pairs formed by an input and an output unit (order 2) and all the triplets formed by a pair of input units and an output unit (order 3). The restriction of having one and only one output unit in each connection comes from the peculiarities of the learning algorithm applied, that we are about to discuss. In this paper we do not discuss any pruning or growing scheme of the connections.

Learning for Boltzmann Machines is defined by the minimisation of the Kullback-Leibler cross-entropy, which is a pseudo-distance (it is not symmetric). The Kullback-Leibler cross-entropy considered is that between the distribution that defines the visible behaviour of the BM and the distribution of the data. The gradient based minimisation of this distance leads to the well-known weight updating rule for Boltzmann Machines. For HOBM we use as the weight updating rule the estimation of the gradient modulated by a momentum term $\Delta_t \omega_\lambda = \alpha\left(\hat{a}^c{}_\lambda - \hat{a}^f{}_\lambda\right) + \mu \Delta_{t-1} \omega_\lambda$.

In this expression $\hat{a}^c{}_\lambda$ and $\hat{a}^f{}_\lambda$ denote the estimations of the mean activation of the connection under the clamped and free distributions, respectively. We do not perform fine tuning of the learning parameters; we simply set α=1 and μ=0.9. In the experiments reported below we performed batch learning, that is, the activation statistics were computed from the whole training set. The absence of hidden units and the fact that a classification task is modelled by orthogonal binary output units allows to avoid the use of simulated annealing for the estimation of activation statistics. Moreover, the statistics of the clamped phase can be computed once for all, and the free phase can be considered as computing the response of the HOBM to each of the training patterns. Samples of our software and previous reports can be accessed via anonymous ftp at the node ftp.sc.ehu.es/pub/unix/hobm. The interested reader may play with this software to verify the robustness of our approach (for example the insensitivity of the learning results to the value of α).

3 Experimental results of HOBMs on NIST data

The HOBMs used in this work have as many continuously valued input units as features taken from the KL transform. The output units are binary {0,1} units. The topologies are densely connected of order 3, already describe above. The state spaces of the units are the range of values that take the transformation coefficient. No normalisation is performed.

Our approach has been to test the feasibility of substituting the PNN, actually used by the Hsfsys to classify the characters, by HOBM. Therefore, we are not concerned with the remaining aspects of the Hsfsys system (segmentation, feature extraction, etc.). The set of experiments described in [3,4] seems to be the foundation for the decision of using PNNs. The data for these experiments was extracted from the Special database 3. The firsts 500 writers were considered. From the first 250 writers, 1/3 of the characters were randomly selected to compose the training set. The whole set of the 250 last writers was used as test set. This selection method seems to obey two ideas: (1) uncorrelated training and test sets, and (2) simulation of realistic operation requires the test set being much bigger than the training set. We think that this experimental design is markedly favourable for the k-NN approaches, and the results reported confirm that intuition. The experiments were done on images of digits. The training set had 7480 digits, and the test set 23140 digits. Both sets show an almost uniform distribution of the digit classes. Given the difficulties in reproducing the exact experimental settings, we have opted to perform a similar experimental design over the database provided with Hsfsys. This database is not explicitly related anywhere with Special Database 3, so our data and that used in [3,4] are, in principle, uncorrelated. Over this database we have randomly selected, without repetition, 1/6th for training and 3/6ths for testing. The training set has 10728 digits, and the test set has 30545 digits. Both sets show an almost uniform distribution of the digit classes. In table 1 we compare the two best error results reported in [3] with the results that we obtain with the HOBMs. The results from [3] correspond to the PNN and Multilayer Perceptron (MLP). The MLP was trained with a conjugate gradient minimisation method, and the number of hidden units was 64. The variance around the prototypes assumed by the PNN was 3. The table shows the improvement of performance of the classifiers as the number of features (KL coefficients) increases.

KL	12	16	20	24	28	32	36	40	44	48	52	56	60	64
PNN	4.3	3.3	2.9	2.7	2.7	2.6	2.6	**2.5**	2.6	2.6	2.6	2.6	2.5	2.5
MLP	6.2	5.3	4.9	4.6	4.5	4.6	4.5	4.5	4.5	4.5	**4.3**	4.5	4.4	4.5
HOBM	6.8	5.7	4.9	4.3	4.0	3.9	**3.8**	3.8	3.8	3.8	3.8	3.8	3.8	3.8

Table 1. Percentage of error on the test set of PNN, MLP [3] and HOBM varying the number of KL features.

Table 1 shows that the figures of the generalisation error of HOBMs fall between those of the PNN and MLP. perform better than the MLP and Having in mind the previous discussion on the databases used in both experiments, the only strong conclusion that we draw form the data in table 1 is that HOBM can be successfully applied to the character classification task with a considerable reduction of the computation time and storage required to classify each character. Our next step has been the the substitution in the Hsfsys of the PNN classifier by a HOBM of order 3. This HOBM was trained over the whole database of 64368 (KL transformed) digits provided with the Hsfsys. The HOBM uses only the 32 first features. The generalisation test has been performed on the first 500 forms of Special Database 1. We have applied both the original Hsfsys (classification by optimised PNN with the whole set of prototypes and using 64 features) and the modified Hsfsys (classification by HOBM of order 3 using 32 features). The results obtained that the PNN give a 97.5 correct classification (2.5 error) on the digit fields, whereas the HOBM give a 97.1 correct classification (2.9 error). The difference in performance was only 0.4 percent in this experiment, whereas in table 1 it was up to 1.9 percent. The data in table 1 gives a pessimistic estimation of the comparative behaviour of PNN and HOBM. Besides that, the storage requirements for the HOBM was much less than for the PNN, and the time to classify each character was reduced by a factor of 10 on the average. The timing results reported in [1] show that the Hsfsys systems invests in the classification task 30% of the time needed to process a form. The speed up obtained with the embedding of HOBM involves, thus, a significant acceleration of the whole Hsfsys system.

4 Conclusions and further work

The present work shows that HOBM can be successfully applied to the task of character recognition based on the features extracted applying the Karhunen Loeve (Principal Components) transformation. The performance of HOBM is comparable to that of the PNN, with much lesser computing requirements. Although the training of the HOBM is time consuming (as is the case for all connectionist approaches) we think that the speed up of the classification is enough to justify the embedding of the HOBM in the Hsfsys. Moreover, testing this embedding over the digit fields of the first 500 forms of Special Database 1, we have found that HOBMs performance is similar to that of PNNs and that they are ten times faster on the average.

Further work will be addressed to a thoroughly test of the embedding of the HOBM in the Hsfsys. An interesting goal is to obtain an adaptive form processing system, able to fit peculiarities of the writer. To this end, the computation of the eigenvectors for the Karhunen Loeve transform could be adapted using neural networks approaches, based in the work of Oja [9]. In this setting the HOBM can be training on-line giving a very robust adaptive classifier. We think that the combination of both neural network techniques could produce a truly adaptive and efficient form-based handwritten recognition system.

References

[1] Garris M.D., J.L. Blue, G.T. Candela, D.L. Dimmick, J. Geist, P.J. Grother, S.A. Janet, C.L. Wilson "NIST Form-Based Handprint Recognition System" NISTIR 5469 (1994) National Institute of Standards and Technoloy, Gaithersburg, MD.

[2] Specht D.F. "Probabilistic Neural Networks" Neural Networks vol 3(1) (1990) pp.109-119

[3] Grother P.J., G.T. Candela "Comparison of Handprinted Digit Classifiers" Tech. Rep. NISTIR 5209, National Institute of Standards and Technology (1993)

[4] Wilson C.L. "Evaluation of Character Recognition Systems" in Neural Networks for Signal Porcessing III pp.485-496, IEEE, New York, (1993)

[5]Blue J.L., G.T. Candela, P.J. Grother, R. Chelappa, C.L. Wilson "Evaluation of Pattern Classifiers for Fingerprint and OCR Applications" Pattern Recognition vol 27(4) pp.485-501

[6] Graña M. , V. Lavin, A. D'Anjou, F.X. Albizuri, J.A. Lozano "High-order Boltzmann Machines applied to the Monk's problems" ESSAN'94, DFacto press, Brussels, Belgium, pp117-122

[7] Albizuri F.X. , A. D´Anjou, M. Graña, F.J. Torrealdea, M.C. Hernandez "The High Order Boltzmann Machine: learned distribution and topology" IEEE Trans. Neural Networks vol 6(3) pp.767-771 (1995)

[8] M. Graña, A. D´Anjou, Albizuri F.X. , A. de la Hera, I. Garcia (1995) High Order Boltzmann Machines with continuous units: some experimental results in J. Mira, F. Sandoval (eds) From Natural to Artificial Computation LNCS 930 Springer-Verlag pp.144-150

[9] Oja E. "Principal Components, Minor Components and linear Neural Networks" Neural Networks vol 5 pp.927-935 (1992)