ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 245-250

Decoding Functions for Kohonen Maps

Alvarez M. and Varfis A.

CEC - Joint Research Center, Ispra Establishment
Institute for System Engineering and Informatics (ISEI).
Neural Network Laboratory - TP 361, 21020 ISPRA (VA), Italy.

Abstract. Let us refer to as a *decoding function* of a trained Kohonen Map (KM) a function that estimates an input vector from the knowledge of the unit it activates (the *winning* unit). The reference vector associated to a winning unit usually provides the decoded value, but it is not necessarily optimal. For a typical class of interactivities between the units of the competitive layer of the KM, and assuming that the finite training sample of the KM can be seen as the set of equiprobable elementary events of a discrete probability distribution, we derive a first order approximation to a decoding function with an improved mean reconstruction error. The enhanced decoder is successfully tested on synthetic data.

1. Introduction

The Kohonen Maps (KMs) connectionist model [1] is an emerging technique for unsupervised data analysis which maps a feature space onto a low-dimensional lattice of reference vectors, while attempting to preserve some topological properties of the original sample distribution. The vector quantization performances *over the training set* could also be a quantity of interest, aware though one may be that the topological constraints over the reference vector layout will entail a fairly suboptimal mean reconstruction error. In this paper we address the issue of defining a suitable decoding function: from a trained KM of which the reference vectors and the number of training patterns that win on each cell are known, we estimate the mean value of the training samples belonging to the receptive field of any given cell. It should be pinpointed that the reference vector itself is not placed at the centroid of its receptive field, unless the algorithmic topological constraints have faded out during a lengthy late stage of the learning process, in which case a the KM algorithm ends up by implementing a stochastic version of the unconstrained K-means algorithm [2].

2. The Reference Vectors Layout

In this section the connectionist algorithm will be briefly described and then a useful characterization (5) (8) for the reference vectors of a KM will be recalled [3]. It states that if the finite training sample S of a KM is seen as the set of equiprobable elementary events of a discrete probability distribution, then the map's reference vectors can be interpreted in terms of weighted means of the elements of S, with weighting coefficients that correspond to the interactivities h (3) that have been used at the end of the learning phase. We will use the same notations as Kohonen [1] for easier cross-reference.

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 245-250

2.1. Kohonen map

Let us consider a fully connected feedforward network with two layers of units. The input layer is fed with a feature vectors $\boldsymbol{x} = \boldsymbol{x}(t) \in S \subset \Re^n$. The formal neurons in the output layer are organized in a two-dimensional *lattice* (or *grid*) and are referred to by their position vector $\boldsymbol{r}_{i=(i_1,i_2)}$, $1 \le i \le N$. A topology is defined on the grid by means of some distance $d(\boldsymbol{r}_i,\boldsymbol{r}_j) = d_{ij}$, like the Euclidean or Manhattan distance ($d_{ij} = \max\{|i_1 - j_1|, |i_2 - j_2|\}$). Each output cell also represents, through its fan-in weight vector, a variable *reference* (or *codebook*) vector $\boldsymbol{m}_i(t) \in \Re^n$.
At step t, either during the learning or the retrieval phase, the squared Euclidean distance E between current pattern $\boldsymbol{x}(t)$ and each reference vector $\boldsymbol{m}_i(t)$ is computed:

$$E(\boldsymbol{x}(t),\boldsymbol{m}_i(t)) = \| \boldsymbol{x}(t) - \boldsymbol{m}_i(t) \|^2 = \sum_{l=1}^{n} [\mathbf{x}_l(t) - \mathbf{m}_{il}(t)]^2 \qquad (1)$$

Let us denote by $c \in \{1,..,N\}$ the index of the best matching reference vector:

$$c(\boldsymbol{x}) = \arg\min_i \| \boldsymbol{x}(t) - \boldsymbol{m}_i(t) \|^2 \qquad (2)$$

The subset of input space where $c(\boldsymbol{x}) = i$ is referred to as the *receptive field* RF_i of $\boldsymbol{m}_i$. A step of the learning algorithm consists of presenting an input pattern $\boldsymbol{x}(t)$ and updating every reference vector $\boldsymbol{m}_i(t)$ proportionally to the *interactivity* $h(\boldsymbol{r}_i,\boldsymbol{r}_c) = h_{ic}$ between unit $\boldsymbol{r}_i(t)$ and the *winner* unit $\boldsymbol{r}_c(t)$:

$$\mathbf{m}_i(t) \longleftarrow \mathbf{m}_i(t) + \varepsilon(t) h_{ic} [\mathbf{x}(t) - \mathbf{m}_i(t)] \qquad (3)$$

The interactivity h is often a positive decreasing function of the grid's distance d. Normally, the *learning rate* ε(t) decreases during the learning process. This holds true for h as well, but in the following sections we will have to consider a stabilized h^τ value - the *target* interactivity - corresponding to the final steps of the learning phase.

2.2. Characteristic equation

If *the input data have been generated by a discrete probability distribution*, then (almost surely) no reference vector layout $\boldsymbol{m} = \{\boldsymbol{m}_i, 1 \le i \le N\}$ will be such that an $\boldsymbol{x}$ falls on the border of adjacent receptive field. Typical examples are problems where generalization over unseen cases is not included, so that the available finite training sample S represents the whole probability distribution. Assuming - for simplicity - that S consists of T equiprobable inputs $\boldsymbol{x}$, we can easily deduce from (3) that the average change of $\boldsymbol{m}_i$ on a single learning step is:

$$\overline{\Delta \mathbf{m}_i} = \frac{\varepsilon}{T} \sum_{\mathbf{x} \in S} h^{\tau}_{ic(\mathbf{x})} [\mathbf{x} - \mathbf{m}_i] \qquad (4)$$

Possible stable solutions need $\overline{\Delta \mathbf{m}_i}$ to be zero, yielding Kohonen map's reference vectors μ_i which obey to the following characteristic equation:

$$\mu_i = \frac{1}{\sum_{\mathbf{x} \in S} h^{\tau}_{ic(\mathbf{x})}} \sum_{\mathbf{x} \in S} h^{\tau}_{ic(\mathbf{x})} x \qquad (5)$$

Given the above restrictions for the input probability distribution, strong evidence has been shown in [3] for supporting the validity of characterization (5) which expresses

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 245-250

that *the codebook vector associated to a neuron r_i should be close to the barycentre of the input set S, weighted by the interactivity coefficients $h_{ic(x)}$.*
In the remaining part of this paper we will consider a single typical class of target interactivities H^α:

$$H_{ij}^{\alpha} = \mathbf{I}(d_{ij} = 0) + \alpha\, \mathbf{I}(d_{ij} = 1) \tag{6}$$

I is the indicator function and $0 \le \alpha \le 1$. For $\alpha = 0$, equation (3) simply implements a stochastic version of the K-means algorithm [2], and equation (5) then states that any reference vector should be close to the centroid of its receptive field RF (t_i denotes the cardinal of RF_i):

$$\mu_i = \frac{1}{t_i} \sum_{\mathbf{x} \in RF_i} \mathbf{x} \tag{7}$$

Whatever the distance d, $\alpha > 0$ in (6) means that the receptive fields of nearest adjacent neurons to a given cell are somehow taken into account when the algorithm settles down. If $A_i = \{j, d_{ij} = 1\}$ denotes the set of indexes for these neurons, then the characteristic equation (5) becomes:

$$\mu_i = \frac{1}{t_i + \alpha \sum_{j \in A_i} t_j} \left[\sum_{\mathbf{x} \in RF_i} \mathbf{x} + \alpha \sum_{j \in A_i} \sum_{\mathbf{x} \in RFj} \mathbf{x} \right] \triangleq \frac{1}{t_i + \alpha \sum_{j \in A_i} t_j} \left[\mathbf{S}_i + \alpha \sum_{j \in A_i} \mathbf{S}_j \right] \tag{8}$$

where we have denoted by S_i the sum of the inputs belonging to RF_i.

3. Decoding Functions

The following problem is addressed: Assume that from a trained KM we know *only* the reference vectors m_i ($\approx \mu_i$) and the corresponding numbers t_i of training patterns that would win on each cell. The underlying distribution of interest is the uniform one over the training sample *S*, which however is no more available. Given an observed $c(x) = i$ value, what is the best guess for x ?
The KM model is put into the vector quantization frame, briefly described below. Then we show that the straightforward and natural estimator $m_{c(x)}$ can be improved (in terms of mean square error) for the H^α target interactivity family (6)

3.1. Vector quantization

Vector quantization is a data compression method where the vectors $x(t) \in \mathbf{S}$ are encoded onto a smaller set of reference vectors $\mu_k \in \Re^n$, $1 \le k \le K$. As with Kohonen's algorithm, the code index $c(x)$ corresponds to a nearest neighbour assignment (2). The customary goal, however, consists of minimizing the mean *reconstruction error* $R(\mu)$:

$$R(\mu) = \frac{1}{T} \sum_{\mathbf{x} \in \mathbf{S}} \left\| \mathbf{x} - \varphi(\mu, c(\mathbf{x})) \right\|^2 \tag{9}$$

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 245-250

The most commonly used *decoding function* φ corresponds to mere nearest neighbour assignment $\varphi(\mu,c(\boldsymbol{x})) = \mu_{c(\boldsymbol{x})}$. While nearest neighbour decoding is optimal for K-means reference vectors (7), one may realize from (5) that over a Kohonen network it will perform badly in terms of reconstruction error as and when the target interactivity function spreads out [4]. In the following section we will derive a first order approximation to the correct decoding function ϕ^{α} for an H^{α} interactivity (6).

3.2. A decoding function for the Kohonen map

If S_i were known, then the best estimator of φ for the quadratic reconstruction error criterion (9) would obviously be the mean value S_i/t_i. However, the S_i sums may be approximately recovered from the m_i ($\approx \mu_i$) values by "inverting" the reference vector layout equation for H^{α} (8).
For a zero order approximation with respect to α, (8) yields of course $S_i \approx t_i \mu_i$ (7). A first order approximation is then obtained by substituting in equality (8) the S_j terms by $t_j \mu_j$ ($j \in A_i$):

$$\mu_i \approx \frac{1}{t_i + \alpha \sum_{j \in A_i} t_j} \left[S_i + \alpha \sum_{j \in A_i} t_j \mu_j \right] \tag{10}$$

When $t_i \neq 0$ - this is true for the empirical input distribution when a realization $c(\boldsymbol{x}) = i$ has been observed - we can derive the desired decoder ϕ^{α}:

$$\frac{S_i}{t_i} \approx \mu_i - \alpha \sum_{j \in A_i} \frac{t_j}{t_i} \left(\mu_j - \mu_i \right) = \phi^{\alpha}(\mu, c(\mathbf{x}) = i) \tag{11}$$

With respect to formula (7), *the neighbouring reference vectors μ_j ($j \in A_i$) act as a repeller for the mean value S_i/t_i of the receptive field RF_i*, proportionally to the relative firing rates t_j/t_i.

4. Experiments

The decoding function ϕ^{α} has been successfully tested on artificial datasets. Ten training samples consisting of 100 vectors regularly distributed on the unit square of $\Re^2$ have been mapped onto square $k \times k$ grids, with k = 4,6,8,10 or 12 (T = 100, N = $k \times k$). Such synthetic data, while reminding us a traditional benchmark example where the Kohonen map has to represent the unit's square uniform distribution, have a training sample size T that is small enough to induce notably different discrete input distributions. The underlying continuous distribution consents to replicate the tests under similar conditions, and eventually average out irrelevant statistical fluctuations.
The results are summerized on table 1. Fifteen different target interactivies from the H^{α} family have been used for every network's size and training set. The corresponding α values are provided in the first column of the table. For a given α, any learning session consisted of two phases. First, widely spread initial interactivities (to foster the reference vector expansion) are progressively shrunk over 30 epochs toward the desired H^{α} value. Then, 10 additional leaning epochs are carried out with the target

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 245-250

interactivity, to push the reference vectors into getting closer to a characteristic layout (5) (8).

For each $k \times k$ network, the first column gives the mean reconstruction error R (9) with respect to the actual reference vectors, and the second column corresponds to the algebraic gain when the proposed decoder ϕ^{α} (11) is used for computing R. The figures are averaged values over the ten experiments sharing identical α and k parameters, and have been multiplied by 10.000 to ease the reading and condense their format.

Useful lessons can be learned from table 1. As a function of α, the gain for using $\phi^{\alpha}(\mu,i)$ in place of μ_i is increasing for low α values, reaches a maximum for α = 0.167 and then decreases to end up in a loss. The overall behaviour is easy to understand: for low α, the quantity $[\phi^{\alpha}(\mu,i) - \mu_i]$ is small (11) and the decoders cannot differ by much; for larger α, the first order approximation will progressively fail. The range of values for which the gain is positive has been highlighted with a grey background. As though it made up for overfitting, *the relevance of formula (11) appears to increase with the N/T ratio*: not only the use of ϕ^{α} becomes safer - larger α values still lead to an improvement for R -, but it also results in increasingly appreciable relative gains.

α	Network 4X4		Network 6x6		Network 8x8		Network 10x10		Network 12x12	
0.000	807	0	437	0	280	0	171	0	120	0
0.063	879	41	491	51	334	52	215	52	159	47
0.118	949	78	540	74	360	73	247	76	185	70
0.167	1001	82	571	74	393	79	268	77	209	77
0.211	1043	64	599	60	416	74	288	77	225	74
0.250	1075	31	621	42	435	66	302	73	241	75
0.286	1101	-12	642	19	446	42	315	67	252	73
0.318	1128	-62	654	-11	468	32	330	57	262	62
0.348	1148	-98	665	-37	473	22	334	44	272	50
0.375	1168	-159	673	-65	477	-1	346	39	283	47
0.400	1179	-224	684	-91	485	-29	348	21	282	35
0.423	1197	-267	692	-123	495	-58	361	18	289	30
0.444	1205	-312	698	-153	496	-64	362	-10	293	15
0.464	1216	-345	706	-174	504	-83	369	-21	302	1
0.483	1230	-377	710	-206	511	-108	367	-22	304	-20

Table 1. For each network, the left column gives the average mean reconstruction error (9). The right column gives the average gain obtained by using the proposed decoding function.

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 245-250

To check further the validity of formula (11), R has also been computed with several ϕ^{β} decoding functions, for β values that do not correspond to actual H^{α} target interactivities ($\beta \neq \alpha$). Plotting R against β obviously yields a parabola, of which the minimum was always obtained for $\beta \approx \alpha$.

5. Conclusion

While it is admittedly unusual to consider the quantization performances of a KM, the fact is that a nearest neighbour decoder does not properly take into account the constraints induced by the learning algorithm. For a commonly used family of interactivities (6), it is possible to define a first order approximation to a decoding function that entails a smaller mean reconstruction error over the training set. Experiments on synthetic data have confirmed the viability of the approximation, and suggest that the proposed decoder should be particularly useful when the training sample size is small with respect to the number of available reference vectors.

6. References

1. T. Kohonen: The Self-Organizing Map. Proceedings of the IEEE, 78, N° 9, 1464-1480 (1990).
2. Y. Linde, A. Buzo, R.M. Gray: An algorithm for vector quantizer design. IEEE Trans. Comm., 28, 84-95 (1980).
3. A. Varfis, C. Versino: An Intuitive Characterization for the Reference Vectors of a Kohonen Map. Proc. of ESANN'93, M. Verleysen ed., 229-234 (1993).
4. A. Varfis, C. Versino: Clustering of socio-economic data with Kohonen Maps. Neural Network World, 2, N° 6, 813-834 (1992)