ESANN'2006 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2006, d-side publi., ISBN 2-930307-06-4.

Magnification control for batch neural gas

Barbara Hammer[1], Alexander Hasenfuß[1], and Thomas Villmann[2]
1- Clausthal University of Technology, Institute of Computer Science, Clausthal-Zellerfeld, Germany, {hammer|hasenfuss}@in.tu-clausthal.de
2- University of Leipzig - Clinic for Psychotherapy and Psychosomatic Medicine, Leipzig, Germany, villmann@informatik.uni-leipzig.de

Abstract. It is well known, that online neural gas (NG) possesses a magnification exponent different from the information theoretically optimum one in adaptive map formation. The exponent can explicitely be controlled by a small change of the learning algorithm. Batch NG constitutes a fast alternative optimization scheme for NG vector quantizers which possesses the same magnification factor as standard online NG. In this paper, we propose a method to integrate magnification control by local learning into batch NG by linking magnification control to an underlying cost function. We validate the learning rule in an experimental setting.

1 Introduction

Vector quantization constitutes an important technical problem in different areas of application such as data mining, control, image compression, or information representation. Thereby, the tasks are diverse such as minimizing the quantization error, optimum information transfer, classification, visualization, or topographic map formation. Self-organizing quantization processes are a common property of many regions of the brain, including the visual, auditory, and somatosensoric cortex. Neural gas (NG) as proposed in [9] constitutes a particularly robust vector quantization method which dynamics can be interpreted as an overlay of standard vector quantization and diffusion. It can be extended towards topographic maps with data optimum topology [10].

A characteristic property of vector quantizers consists in a selective magnification of regions of interest. This corresponds to a specific connection of the density of prototypes and stimuli. An information theoretically optimum magnification factor one corresponds to an exact adjustment of the prototypes according to the underlying data distribution. This magnification law is achieved by approaches which explicitely optimize the information transfer or related quantities [8]. For a variety of popular alternatives, however, the magnification follows a power law with exponent different from one [3, 9, 16]. Starting with the work [1], schemes to control the magnification factor have been proposed in the literature, for recent results see e.g. [13]. Magnification control changes the learning scheme and allows to achieve a magnification factor one or beyond. An explicit control is particularly interesting for application areas where rare events should be suppressed or, contrarily, emphasized, which has proven beneficial in several tasks in robotics and image inspection [11, 15, 14]. In addition, this effect corresponds to biological phenomena such as the perceptual magnet effect which leads to an emphasis of rarely occurring stimuli [5, 7]. The magnification factor of online NG is $D/(D+2)$, D being the intrinsic (Hausdorff-)dimension of the data manifold of stimuli. This can be controlled using e.g. local learning, which changes the learning rate by a factor depending on the local data density [12].

Neural gas is a very robust, but computationally complex method since several thousands of learning cycles are necessary for convergence. For priorly known training examples, an alternative batch update scheme becomes possible [2]: the cost function of neural gas is optimized in turn for the prototype locations and hidden variables which correspond to the rank. Since each step takes all training patterns into account and directly moves into the next local optimum, much fewer training cycles are necessary. Unlike batch SOM, which easily suffers from topological mismatches [4], a data optimum topology is used.

Batch-NG follows the same power law for the magnification factor as online NG because of the same cost function. Here, we propose magnification control for batch NG by including local learning into the update formulas. The link becomes possible relating local learning to a modified cost function of NG which can be optimized in the batch mode. We demonstrate this strategy by a controlled example where the property of optimum information transfer can be tested.

2 Neural Gas

Assume data vectors $\boldsymbol{v} \in \mathbb{R}^d$ are given as stimuli, distributed according to $P(\boldsymbol{v})$. The data points should faithfully be represented by prototypes $\boldsymbol{w}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$. The objective of neural gas is a minimization of the cost function

$$E(W) = \frac{1}{2C(\lambda)} \sum_{i=1}^{n} \int h_\lambda(k_i(\boldsymbol{v}, W)) \cdot \|\boldsymbol{v} - \boldsymbol{w}_i\|^2 P(\boldsymbol{v}) d\boldsymbol{v}$$

where W denotes the set of prototypes, $k_i(\boldsymbol{v}, W) = |\{\boldsymbol{w}_j | \|\boldsymbol{v}-\boldsymbol{w}_j\|^2 < \|\boldsymbol{v}-\boldsymbol{w}_i\|^2\}|$ is the rank of prototype i, $h_\lambda(t)$ is a Gaussian shaped curve for $t \geq 0$ such as $h_\lambda(t) = \exp(-t/\lambda)$ with neighborhood range $\lambda > 0$, and $C(\lambda)$ is a normalization constant. The corresponding online adaptation rule is

$$\Delta \boldsymbol{w}_i = \epsilon \cdot h_\lambda(k_i(\boldsymbol{v}_j, W)) \cdot (\boldsymbol{v}_j - \boldsymbol{w}_i)$$

with learning rate $\epsilon > 0$, given a stimulus $\boldsymbol{v}_j$. This learning rule adapts all prototypes according to their rank given $\boldsymbol{v}_j$.

This adaptation can be applied in online scenarios such as a robot exploring an environment; however, usually several thousand steps are necessary for convergence, and the procedure can become quite costly. If data are given priorly, an alternative batch adaptation scheme can be applied. For a given (finite) training set $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p$ the cost function of NG becomes

$$\hat{E}(W) = \frac{1}{2C(\lambda)} \sum_{i=1}^{n} \sum_{j=1}^{p} h_\lambda(k_i(\boldsymbol{v}_j, W)) \cdot \|\boldsymbol{v}_j - \boldsymbol{w}_i\|^2 P(\boldsymbol{v}^j)$$

where $P(\boldsymbol{v}^j)$ is usually estimated by $1/p$. In batch optimization, the term $k_i(\boldsymbol{v}_j, W)$ is substituted by a free variable k_{ij} which is to be optimized under the condition that $k_{1j}, \ldots, k_{nj}$ yields a permutation of 0 to $n-1$ for each j. For fixed $\boldsymbol{w}_i$, optimum variables k_{ij} are given by the rank $k_i(\boldsymbol{v}_j, W)$. In turn, for fixed k_{ij}, optimum assignments of the prototypes have the form

$$\boldsymbol{w}_i = \left(\sum_{j=1}^{p} h_\lambda(k_{ij}) \cdot \boldsymbol{v}_j \right) \Big/ \left(\sum_{j=1}^{p} h_\lambda(k_{ij}) \right) .$$

Batch NG consecutively performs these two steps until convergence, which can usually be observed after only few epochs [2]. Batch NG always converges to a fixed point of the assignments, which is a local optimum of the original (discrete) cost function of NG unless two prototypes have the same distance from one data point for the final prototype locations (with measure 0 for concrete settings).

3 Magnification control

Optimization schemes for the NG cost function result in a map formation which obeys a magnification power law with magnification exponent different from one as demonstrated in the literature [9]. In this argumentation, the effect of an average update $\langle \Delta w_i \rangle$ on the map behavior is investigated. Thereby, several properties of the map are used, such as the fact, that the neighborhood function $h_\lambda(k_i(\boldsymbol{v}, W)$ converges sufficiently fast to 0 such that terms of higher order can be neglected. In addition, the system is considered in the limit of many prototypes, such that a continuum can be assumed. Then, the weight density $\rho(\boldsymbol{w}_i)$ of the map is linked to the density $P(\boldsymbol{w}_i)$ given by the input space by $\rho(\boldsymbol{w}_i) \sim P(\boldsymbol{w}_i)^\alpha$ with magnification factor $\alpha = D/(D+2)$ where D is the effective data dimensionality $\leq d$ [9]. For a given finite number of prototypes and patterns this law approximately describes concrete maps.

Local learning introduces a learning rate for each training pattern:

$$\Delta \boldsymbol{w}_i = \epsilon_0 \cdot P(\boldsymbol{w}_{s(\boldsymbol{v}_j)})^m \cdot h_\lambda(k_i(\boldsymbol{v}_j, W)) \cdot (\boldsymbol{v}_j - \boldsymbol{w}_i)$$

where $\epsilon_0 > 0$ is the learning rate and $s(\boldsymbol{v}_j)$ is the winner for stimulus $\boldsymbol{v}_j$. $m > 0$ is a constant which controls the magnification exponent. For this learning rule, the power law $\rho(\boldsymbol{w}_i) \sim P(\boldsymbol{w}_i)^{\alpha'}$ where $\alpha' = (m+1) \cdot \alpha = (m+1) \cdot D/(D+2)$ results, as shown in [12]. An information theoretically optimum factor is obtained for $m = 2/D$. Larger values emphasize input regions with high density, whereas smaller values focus on regions with rare stimuli. To apply this learning rule, the distribution P as well as the effective data dimensionality D have to be estimated from the data (using e.g. Parzen windows resp. the box counting dimension). Here, we consider the similar learning rule

$$\Delta \boldsymbol{w}_i = \epsilon_0 \cdot P(\boldsymbol{v}_j)^m \cdot h_\lambda(k_i(\boldsymbol{v}_j, W))(\boldsymbol{v}_j - \boldsymbol{w}_i)\,.$$

The average can be formulated as integral

$$\langle \Delta \boldsymbol{w}_i \rangle \sim \int P(\boldsymbol{v})^m \cdot h_\lambda(k_i(\boldsymbol{v}, W)) \cdot (\boldsymbol{v} - \boldsymbol{w}_i) \cdot P(\boldsymbol{v}) d\boldsymbol{v}\,.$$

In the limit of a continuum of prototypes, $\boldsymbol{w}_{s(\boldsymbol{v})} = \boldsymbol{v}$ holds, thus, the average update yields exactly the same result as the original one proposed in [12] and the same magnification factor $(m+1) \cdot \alpha'$ results. This alternative update has the benefit that it constitutes a stochastic gradient descent of the cost function

$$E_m(W) = \frac{1}{2C(\lambda)} \sum_{i=1}^{n} \int P(\boldsymbol{v})^m \cdot h_\lambda(k_i(\boldsymbol{v}, W)) \cdot \|\boldsymbol{v} - \boldsymbol{w}_i\|^2 \cdot P(\boldsymbol{v}) d\boldsymbol{v}$$

as shown in Appendix A. Thus, learning schemes which optimize $E_m(W)$ yield a map formation with magnification factor α' as given above.

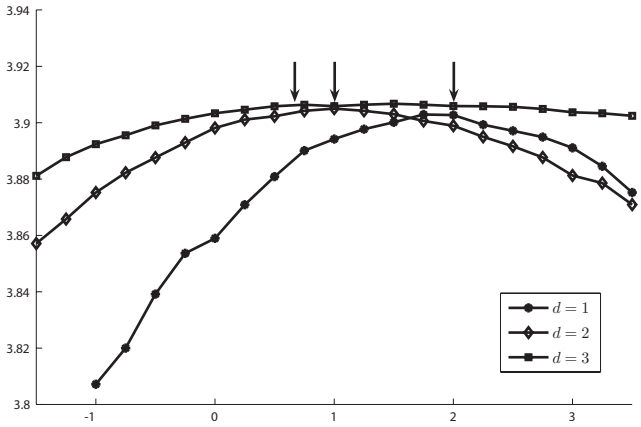

Fig. 1: Entropy of map formation for different values m of magnification control and training sets of intrinsic dimensionality $d \in \{1, 2, 3\}$. The arrows indicate the expected optima of the entropy according to the underlying theory.

The formulation of magnification control by means of a cost function opens the way towards an extension of control schemes to batch learning: For a given discrete set, the cost function becomes

$$\hat{E}_m(W) = \frac{1}{2C(\lambda)} \sum_{i=1}^{n} \sum_{j=1}^{p} h_\lambda(k_i(\boldsymbol{v}_j, W)) \cdot \|\boldsymbol{v}_j - \boldsymbol{w}_i\|^2 \cdot P(\boldsymbol{v}_j)^m \,.$$

As beforehand, we substitute the terms $k_i(\boldsymbol{v}_j, W)$ by hidden variables k_{ij} which are chosen from $\{0, \ldots, n-1\}$ such that $k_{1j}, \ldots, k_{nj}$ constitutes a permutation of $\{0, \ldots, n-1\}$. Batch optimization in turn determines optimum k_{ij}, given prototype locations, and optimum prototype locations, given values k_{ij}, as follows:

1. $k_{ij} = |\{\boldsymbol{w}_l \mid \|\boldsymbol{v}_j - \boldsymbol{w}_l\|^2 < \|\boldsymbol{v}_j - \boldsymbol{w}_i\|^2\}|$,
2. $\boldsymbol{w}_i = \left(\sum_j h_\lambda(k_{ij}) \cdot P(\boldsymbol{v}_j)^m \cdot \boldsymbol{v}_j\right) / \left(\sum_j h_\lambda(k_{ij}) \cdot P(\boldsymbol{v}_j)^m\right)$

It is shown in Appendix B that this procedure converges in a finite number of steps towards a fixed point. The fixed point is a local minimum of the original cost function $\hat{E}_m(W)$ if the distances of the final prototype locations from the data points are mutually different (which is almost surely the case in concrete settings). This offers a batch adaptation scheme with magnification coefficient $(m+1) \cdot D/(D+2)$ which can explicitely be controlled by the quantity m.

4 Experiments

Control experiments use the distribution $(x_1, \ldots, x_d, \prod_{i=1}^{d} \sin(\pi \cdot x_i))$ for $d \in \{1, 2, 3\}$ and x_i uniformly distributed in $[0, 1]$. The number of points is 2500 ($d = 1$), 5000 ($d = 1$), and 10000 ($d = 3$). Obviously, the intrinsic dimensionality is d, such that optimum information transfer can be expected for $m = 2$ ($d = 1$),

$m = 1$ ($d = 2$), and $m = 2/3$ ($d = 3$). All experiments have been performed for $m \in [-1.5, 3.5]$ (step size 0.25) such that the overall behavior of the local learning rule for different m can be observed. An NG network with 50 neurons, initial neighborhood size 25 (multiplicatively annealed to 0), and 200 epochs per training run has been used. The reported results have been averaged over 20 runs. The data density $P(\boldsymbol{v})$ has been estimated by a Parzen window estimator with bandwidth given by the average training point distances divided by 3.

The entropy of the winner counts of the map is reported in Fig. 1. The entropy should be maximum for optimum information transfer, i.e. we expect the optimum for $m = 2$, $m = 1$, and $m = 2/3$, respectively. As indicated by the arrows, the experimental optimum of the curves is very close to the expected theoretical values in all cases, thus confirming the theory presented in this paper.

5 Discussion

Linking local learning to a general cost function, we have transferred magnification control to batch NG and demonstrated the applicability in a controlled experiment. This method opens the way towards interesting applications of the fast batch-NG scheme: apart from an optimum information transfer, a magnification of rare events which is relevant for the classification of unbalanced classes, visualization of uncommon effects, or modeling attention becomes possible. We would like to mention that batch NG can naturally be transferred to proximity data for which no embedding into a euclidian vector space is available. Magnification control can immediately be applied to this important scenario.

Appendix A

The derivative of the cost function $E_m(W)$ is given by

$$\begin{aligned}\frac{\partial E_m(W)}{\partial \boldsymbol{w}_l} &= -\frac{1}{C(\lambda)} \int h_\lambda(k_i(\boldsymbol{v}, W)) \cdot (\boldsymbol{v} - \boldsymbol{w}_l) \cdot P(\boldsymbol{v})^{m+1} d\boldsymbol{v} \\ &\quad + \frac{1}{2C(\lambda)} \sum_{i=1}^{n} \int h'_\lambda(k_i(\boldsymbol{v}, W)) \cdot \frac{\partial k_i(\boldsymbol{v}, W)}{\partial \boldsymbol{w}_l} \cdot \|\boldsymbol{v} - \boldsymbol{w}_i\|^2 \cdot P(\boldsymbol{v})^{m+1} d\boldsymbol{v} \,.\end{aligned}$$

$k_i(\boldsymbol{v}, W) = \sum_{o=1}^{n} \theta(\|\boldsymbol{v} - \boldsymbol{w}_i\|^2 - \|\boldsymbol{v} - \boldsymbol{w}_o\|^2)$, where θ is the Heaviside function with symmetric and for inputs $\neq 0$ vanishing derivative δ. The first term yields the update rule. The second term equals the vanishing term

$$\begin{aligned}&\frac{1}{C(\lambda)} \int \left(\sum_{o=1}^{n} -h'_\lambda(k_l(\boldsymbol{v}, W)) \cdot \delta((\boldsymbol{v} - \boldsymbol{w}_l) - \|\boldsymbol{v} - \boldsymbol{w}_o\|^2) \cdot (\boldsymbol{v} - \boldsymbol{w}_l) \cdot \|\boldsymbol{v} - \boldsymbol{w}_l\|^2 + \right. \\ &\left. \sum_{i=1}^{n} h'_\lambda(k_i(\boldsymbol{v}, W)) \cdot \delta(\|\boldsymbol{v} - \boldsymbol{w}_i\|^2 - \|\boldsymbol{v} - \boldsymbol{w}_l\|^2)(\boldsymbol{v} - \boldsymbol{w}_l)\|\boldsymbol{v} - \boldsymbol{w}_i\|^2 \right) P(\boldsymbol{v})^{m+1} d\boldsymbol{v} \,.\end{aligned}$$

Appendix B

Consider the cost function

$$\hat{E}_m(W) = \frac{1}{2C(\lambda)} \sum_{i=1}^{n} \sum_{j=1}^{p} h_\lambda(k_i(\boldsymbol{v}_j, W)) \cdot \|\boldsymbol{v}_j - \boldsymbol{w}_i\|^2 \cdot P(\boldsymbol{v}_j)^m \,.$$

For each W, batch NG determines unique optimum assignments $k_{ij}(W) := k_i(\boldsymbol{v}_j, W)$ where we assume a fixed priority in case of ties. These values stem from a finite set. Conversely, for given $k_{ij}(W)$ unique optimum assignments W' are determined by batch NG. We consider the auxiliary function

$$Q(W', W) := \frac{1}{2C(\lambda)} \sum_{i=1}^{n} \sum_{j=1}^{p} h_\lambda(k_{ij}(W)) \cdot \|\boldsymbol{v}_j - \boldsymbol{w}'_i\|^2 \cdot P(\boldsymbol{v}_j)^m$$

which is connected to $\hat{E}_m(W)$ via $\hat{E}_m(W) = Q(W, W)$. Assume prototype locations W are given and new prototype locations W' are computed in one cycle of batch-NG. It holds $\hat{E}_m(W') = Q(W', W') \leq Q(W', W)$ because $k_{ij}(W')$ are optimum assignments for the k_{ij} given W'. In addition, $Q(W', W) \leq Q(W, W) = \hat{E}_m(W)$, because W' are optimum assignments of the prototypes given values $k_{ij}(W)$. Thus, $\hat{E}_m(W') - \hat{E}_m(W) = Q(W', W') - Q(W', W) + Q(W', W) - Q(W', W') \leq 0$, i.e. the value of the cost function decreases in each step. Because there exists only a finite number of different values k_{ij}, the procedure converges after a finite number of steps towards a fixed point W^*.

Assume that the distances of training points from W^* are mutually different. Then, the assignment $W \mapsto k_{ij}(W)$ is constant in a vicinity of W^*. Thus, $\hat{E}_m(\cdot)$ and $Q(\cdot, W^*)$ are identical in a neighborhood of W^* and a local optimum of $Q(\cdot, W^*)$ is also a local optimum of $\hat{E}_m$. Hence, W^* is a local optimum of $\hat{E}_m$.

References

[1] H. Bauer, R. Der, and M. Herrmann. Controlling the magnification factor of self-organizing feature maps. *Neural Computation*, 8(4):757–771, 1996.
[2] M. Cottrell, B. Hammer, A. Hasenfuß, and T. Villmann. Batch neural gas. *WSOM*, 2005.
[3] D. Dersch and P. Tavan. Asymptotic level density in topological feature maps. *IEEE TNN*, 6(1):230–236, January 1995.
[4] J.-C. Fort, P. Letrémy, and M. Cottrell. Advantages and drawbacks of the batch Kohonen algorithm. M. Verleysen, ed., *ESANN*, pages 223–230, 2002.
[5] M. Herrmann, H.-U. Bauer, and R. Der. The 'perceptual magnet' effect: a model based on self-organizing feature maps. *Neural Comp. and Psychology*, pages 107–116. 1994.
[6] T. Kohonen and P.Somervuo. How to make large self-organizing maps for nonvectorial data. *Neural Networks*, 15:945–952, 2002.
[7] P. Kuhl, K. Williams, F. Lacerda, K. Setevens, and B. Lindblom. Linguistic experience alters phonetic perception in infants by 6 months of age. *Science*, 255:606–608, 1992.
[8] R. Linsker. How to generate maps by maximizing the mutual information between input and output signals. *Neural Computation*, 1:402–411, 1989.
[9] T. Martinetz, S. Berkovich, and K. Schulten. 'Neural gas' network for vector quantization and its application to time series prediction. *IEEE TNN*, 4(4):558–569, 1993.
[10] T. Martinetz and K. Schulten. Topology representing networks. *Neural Netw.*, 7(2), 1994.
[11] E. Merényi and A. Jain. Forbidden magnification? M. Verleysen, ed., *ESANN*, 2004.
[12] T. Villmann. Controlling strategies for the magnification factor in the neural gas network. *Neural Network World*, 10(4):739–750, 2000.
[13] T. Villmann and J. Claussen. Magnification control in self-organizing maps and neural gas. *Neural Computation*, 2005. accepted.
[14] T. Villmann and A. Heinze. Application of magnification control for the neural gas network in a sensorimotor architecture for robot navigation. H.-M. Groß, K. Debes, and H.-J. Böhme, eds., *Selbstorganisation von Adaptiven Verfahren*, pages 125–134, 2000.
[15] T. Villmann, E. Merényi, and B. Hammer. Neural maps in remote sensing image analysis. *Neural Networks*, 16(3-4):389–403, 2003.
[16] P. Zador. Asymptotic quantization error of continuous signals and the quantization dimension. *IEEE Transactions on Information Theory*, 28:149–159, 1982.