ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 87-92

Model clustering by deterministic annealing

Bart Bakker and Tom Heskes

Foundation for Neural Networks
Geert Grooteplein 21
6525 EZ Nijmegen
The Netherlands*

Abstract

Although training an ensemble of neural network solutions increases the amount of information obtained from a system, large ensembles may be hard to analyze. Since data clustering is a good method to summarize large bodies of data, we will show in this paper how to use clustering on instances of neural networks. We will describe an algorithm based on deterministic annealing, which is able to cluster various types of data. As an example, we will apply the algorithm to instances of three different types of MLP's, trained to predict the time of death of ovarian cancer patients.

1 Introduction

In neural network analysis a growing trend exists to not only train a network to find the best solution, but to create an ensemble of good solutions. For example, bootstrapping [1] leads to many different models by optimizing each model on a different sample of the training set, whereas the Bayesian approach creates a probability distribution of all solutions, from which may be sampled by Markov Chain Monte Carlo (MCMC) procedures [2]. In order to catch the fine nuances of the system, the number of network representations, or models, can be of considerable size. For large problems, the amount of models may even be too high to keep a good overview of the solution, and special transformations will be required to summarize the set of networks, preferably without loss of information.

Clustering can be seen as the representation of a large data set W by a smaller data set M. The components of W and M may be any kind of data carrying elements; the type of the elements may vary from set W to set M, or even within sets. The only requirement is that there exists a distance function $D(W, M)$, indicating how much sets W and M differ from each other, or rather how much information is lost in the conversion from W to M.

Since W and M may contain any kind of data carrying elements, the method of clustering may well be used to find a workable representation of any oversized

*This research was supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 87-92

solution space of a neural network. Taking the elements of W to be the networks in the original (large) solution, represented by their weights, biases and overall structure of the network (number of hidden layers, transfer functions, etc.), M is the (smaller) set of networks best representing the features contained in W. It would be logical for $D(W, M)$ to be based on the outputs of the networks, since these best represent the nature of each network. The form of the distance function can well be chosen as the error function used in the original training, since this function already expresses the distance between a "perfect" solution and an approximation.

In this paper we will review the clustering method first constructed by Rose *et al* [3]. Their method, based on the principles of deterministic annealing as first described by Jaynes [4], was shown to yield good results for the clustering of two-dimensional data with a Euclidean distance function. We will generalize this method for use with other data types and distance functions, and use it to cluster models (e.g. neural networks).

2 Theory

2.1 Notation

Suppose we have N models, fully characterized through their parameters w_i, $i = 1 \ldots N$. We define M other models, which we will refer to as cluster centres, denoted m_α, $\alpha = 1 \ldots M$. $D(w_i, m_\alpha)$ is the distance of model i to cluster centre α. The distance need not be symmetric, i.e., $D(w, m)$ may be different from $D(m, w)$[1] We do however require that $D(m, w) \geq 0$ and $D(m, m) = 0$. In the following we will assume distances of the form

$$D(w_i, m_\alpha) = \sum_\mu d(y(w_i, x^\mu), y(m_\alpha, x^\mu)) \ ,$$

for some distance measure $d(y_1, y_2)$, where $y(w_i, x^\mu)$ is the output of the network with weights w_i on the input x^μ. Each network is supposed to have the same input. Since, once the inputs x^μ are given, the only dependency of our models w_i on the result of the clustering procedure is through the outputs $y(w_i, x^\mu)$, we can compute these in advance.

2.2 The derivation of the "free energy"

We define variables $p_{i\alpha}$ as the probability that model i belongs to cluster α. The distances between models w_i and cluster centres m_α are given by $D(w_i, m_\alpha)$. We assume that the models w_i are given, but that the probabilities $p_{i\alpha}$ and cluster centres m_α are still free to choose. One of the goals of clustering is to put the cluster centres such as to minimize the average distance of the models to the cluster centres, i.e., to find a low average energy

$$E(m, p) = \sum_{i\alpha} p_{i\alpha} D(w_i, m_\alpha) \ .$$

[1] Note that $D(w, m)$ is not a distance function in the mathematical sense, but rather a measure of the difference between w and m. However, since the concept of clustering is intuitively best understood in terms of positions and distances, we will still refer to $D(m, w)$ as a distance.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 87-92

In this framework the average energy is a weighted average over the distances between models and cluster centres, where the weight of $D(w_i, m_\alpha)$ is equal to the probability that model w_i belongs to cluster α. For fixed cluster centres m_α, minimizing the average energy would correspond to assigning each model to its nearest cluster centre with probability one. A proper way to regularize this is through a penalty term of the form

$$S(p) = -\sum_{i\alpha} p_{i\alpha} \ln p_{i\alpha} \ ,$$

the discrete version of the Shannon entropy, which is the only quantity which is positive, increases with increasing uncertainty, and is additive for independent sources of uncertainty [4]. Maximizing $S(p)$ therefore favours a state of total chaos, i.e. $\forall_{i\alpha,j\beta}\ p_{i\alpha} = p_{j\beta}$, which corresponds to the notion that we have no prior knowledge about the structure of the clusters.

We introduce a regularization parameter T, weighting the two different terms, to arrive at the "free energy"

$$F(m,p) = E(m,p) - TS(p) \ .$$

Minimizing $F(m,p)$ can be seen as a search for the best compromise between a low average distance (minimizing $E(m,p)$) and keeping a reasonable amount of chaos in the system (maximizing $S(p)$). For any choice of the model centres m_α, the probabilities $p_{i\alpha}$ minimizing the free energy $F(m,p)$ read (under the constraint $\sum_\alpha p_{i\alpha} = 1\ \forall_i$)

$$p_{i\alpha}(m) = \frac{\mathrm{e}^{-\beta D(w_i, m_\alpha)}}{\sum_\gamma \mathrm{e}^{-\beta D(w_i, m_\gamma)}} \tag{1}$$

with $\beta = 1/T$. Substitution of this result into the free energy then yields

$$F(m) = F(m, p(m)) = \sum_i \log \sum_\alpha \mathrm{e}^{-\beta D(w_i, m_\alpha)} \ . \tag{2}$$

Eq. (2) is equivalent to the result presented in [3]. The main difference between our derivation and the derivation made by Rose *et al* [3] is the role of the parameter β. Here it is just a regularization parameter, that can be chosen in advance. In [3] an average energy $\langle E \rangle$ is defined, such that (like in statistical physics theory) β is a Lagrange multiplier that must be tuned to ensure that $\langle E \rangle$ stays constant. Since we have certain reservations with respect to this approach (in information theory there usually is no way of knowing the average distance (energy), and the fact that β in this way becomes implicitly dependent on the cluster centres m is completely neglected) we prefer the derivation presented in this section.

2.3 Phase transition

For small β (high temperature) the free energy is dominated by its entropy term, leading to a solution with only one cluster centre or, equivalently, to a solution where all cluster centres are the same. Minimization of $F(m)$ is in this case equivalent to finding the global minimum of $\sum_i D(w_i, m)$. However, when

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 87-92

β reaches a certain level, β_c, the one-cluster solution will evolve into a multiple-cluster solution, corresponding to multiple local minima of $F(m)$. Since the Hessian of $F(m_0)$ passes from being positive-definite to non-positive-definite at $\beta = \beta_c$, the latter can be found by solving:[2]

$$\det\left[\left.\frac{\partial^2 F}{\partial m \partial m^T}\right|_{m=m_0}\right] = 0 \,.$$

Substituting (2) yields

$$\frac{\partial^2 F}{\partial m_\beta \partial m_\alpha^T} = \sum_i \left[\frac{1}{n_c}\frac{\partial^2 D_{i\alpha}}{\partial m_\alpha \partial m_\alpha^T}\delta_{\alpha\beta} - \frac{\beta}{n_c}\frac{\partial D_{i\alpha}}{\partial m_\alpha}\frac{\partial D_{i\alpha}}{\partial m_\alpha^T}\delta_{\alpha\beta} + \frac{\beta}{n_c^2}\frac{\partial D_{i\beta}}{\partial m_\beta}\frac{\partial D_{i\alpha}}{\partial m_\alpha^T}\right] \quad (3)$$

where $D_{i\alpha} \equiv D(w_i, m_\alpha)$ and $\delta_{\alpha\beta}$ is the Kronecker delta function. We have made use of the fact that at β_c all cluster centres are given by m_0, so $p(w_i, m_\alpha) = \frac{1}{n_c}$ $\forall_{i,\alpha}$ where n_c is the number of clusters.

In such a case, when the last term in Eq. 3 is positive-semidefinite (which it is, being a correlation matrix), the matrix S is positive-definite if and only if

$$\sum_i \frac{\partial^2 D_{i\alpha}}{\partial m_\alpha \partial m_\alpha^T} - \beta \sum_i \frac{\partial D_{i\alpha}}{\partial m_\alpha}\frac{\partial D_{i\alpha}}{\partial m_\alpha^T} \quad (4)$$

is positive-definite (see [3]). Therefore, assuming that the left part of (4) has an inverse, symmetry breaking occurs at $\beta_c = 1/\lambda_{\max}$ where $\lambda_{\max}$ is the largest eigenvalue of $\left[\sum_i \frac{\partial^2 D_{i\alpha}}{\partial m_\alpha \partial m_\alpha^T}\right]^{-1}\left[\sum_i \frac{\partial D_{i\alpha}}{\partial m_\alpha}\frac{\partial D_{i\alpha}}{\partial m_\alpha^T}\right]$. Inserting the sum-squared distance for $D_{i\alpha}$ yields the result already obtained by Rose *et al* [3]. An extension of this result to self-organizing maps is given in [5] by Graepel *et al*.

3 The algorithm

The problem of (model) clustering now consists of finding, (for each β) the cluster centres m_α minimizing the free energy $F(m_\alpha)$. This corresponds to solving (for each m_α):

$$\frac{\partial F}{\partial m_\alpha} = \sum_i p_{i\alpha}\frac{\partial D_{i\alpha}}{\partial m_\alpha} = 0 \,. \quad (5)$$

Since both $p_{i\alpha}$ and $\frac{\partial D_{i\alpha}}{\partial m_\alpha}$ are functions of m_α, and the different m_α are interrelated through their normalization (which is a function of all m_α), solving Eq. 5 is not very simple. However, because of its probabilistic structure, the problem does lend itself to be solved by a GEM (Generalized EM) algorithm. A full description of the (Generalized) EM algorithm can be found in [6].

In the first step (expectation step) of the GEM algorithm the probabilities $p_{i\alpha}$, as given by Eq. 1, are evaluated. In the second step (maximization step) we seek to find new cluster centres m'_α such that:

$$\sum_i p_{i\alpha} D(w_i, m'_\alpha) \leq \sum_i p_{i\alpha} D(w_i, m_\alpha) \,. \quad (6)$$

[2] In the following m^T denotes the transpose of m.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 87-92

A solution for m'_α can be obtained from any gradient descent algorithm on $\sum_i p_{i\alpha} D(w_i, m)$ starting from $m = m_\alpha$. However, for certain distance functions (e.g. the sum-squared error or the cross-entropy error[3]) "average" outputs $\bar{y}^\mu_\alpha$ can be computed[4] such that, up to irrelevant constants,

$$\sum_i p_{i\alpha} D(w_i, m_\alpha) = \sum_i p_{i\alpha} \sum_\mu d(y(w_i, x^\mu), y(m_\alpha, x^\mu)) = \sum_\mu d(\bar{y}^\mu_\alpha, y(m_\alpha, x^\mu)) \; .$$

In those cases, the M-step is nothing but minimizing the distance between $\bar{y}^\mu_\alpha$ and $y(m_\alpha, x^\mu)$. This is achieved by training the network on the outputs $\bar{y}^\mu_\alpha$ and the inputs used throughout the algorithm to obtain the actual model parameters m_α. However, if there are no restrictions on the outputs the network can produce, the model parameters m_α need not be retrieved in every M-step, but only after the last step, when all cluster centres are found in terms of their corresponding outputs $\bar{y}^\mu_\alpha$. This may lead to a tremendous speed-up in the clustering process. The thus obtained cluster centres can be used as an initialization for the clustering process with restrictions on the outputs.

4 Applications

Although clustering in weight space has been used in network analysis [7], model clustering as described in this paper has not. However, several applications do readily present themselves. One application can be found in the comparison of networks which do not have the same structure, but are trained to perform the same task. If the network outputs depend strongly on the network's structure, different models are likely to be assigned to different cluster centres. If however two differently structured networks produce similar outputs, there will be clusters inhabited by both types of networks.

Another possible application is the detection of symmetries in one type of network [7]: clustering based on a distance function dependent on the outputs of the networks will produce different clusters than clustering in weight space directly (e.g. with a Euclidean distance function). Comparison of the two sets of clusters (i.e. clusters based on the outputs and clusters in weight space) may help to gain insight in the symmetries of the system.

We have implemented a first application to compare three sets of feed-forward networks, trained to predict the time of death of ovarian cancer patients, based on the patients' medical information [8, 9]. Each set contained 80 instances (obtained by Hybrid Markov Chain Monte Carlo (HMCMC) sampling [2]) of a network with 1, 2 or 3 hidden units; all networks were trained on the same data. The clustering algorithm as described in this paper was applied to the union of these three sets. The results can be seen in Figure 1, where for each cluster the weight ($W_\alpha = \frac{1}{N} \sum_i^N p_{i\alpha}$) of each set is given. At $\beta < \beta_c$ all probabilities are equal, and no distinction can be made between the different clusters or the different types of models. However, at $\beta > \beta_c$, three clusters are formed. From the division of the three types of models over the clusters it is clear that the models do possess different properties: especially in clusters 1 and 3 the weights of the models vary significantly. In cluster 2 however, the weights are divided more equally, so although the three models do have different

[3] $d(y(w_i, x^\mu), y(m_\alpha, x^\mu)) = y(w_i, x^\mu) \log \frac{y(w_i, x^\mu)}{y(m_\alpha, x^\mu)} + [1 - y(w_i, x^\mu)] \log \frac{1 - y(w_i, x^\mu)}{1 - y(m_\alpha, x^\mu)}$

[4] For the sum-squared error or the cross-entropy error $\bar{y}^\mu_\alpha = \sum_i p_{i\alpha} y(w_i, x^\mu)$.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 87-92

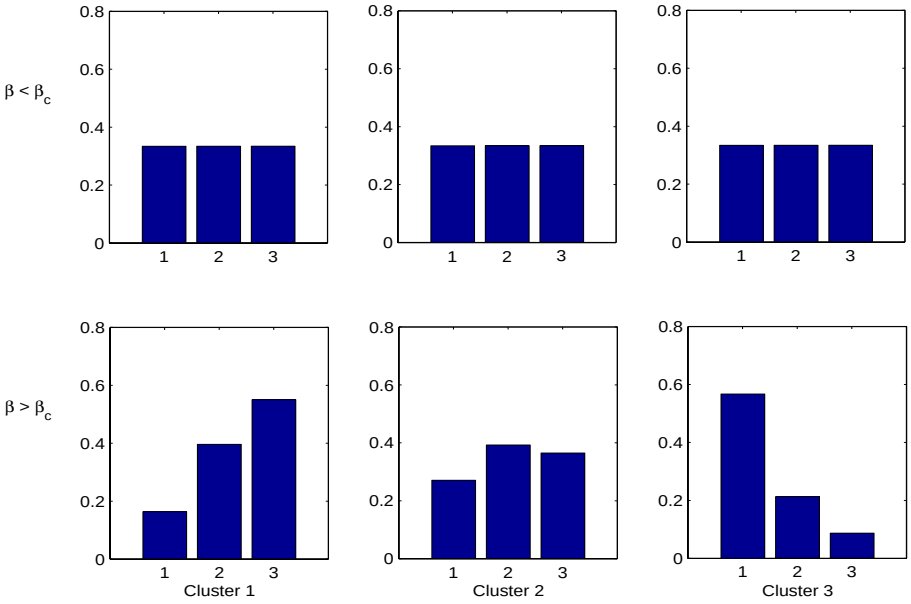

Figure 1: Weights of the three types of models inside the three clusters. Each bar plot represents one cluster, each bar represents one weight. In the upper row $\beta < \beta_c$ and all weights are equal. In the bottom row $\beta > \beta_c$ and a structure arises in the division of models over the clusters.

features, there also may exist a common denominator. With this information in mind, comparing the models inside one cluster may lead to new insights about the prediction problem.

References

[1] T. Heskes. Balancing between bagging and bumping. In M. Mozer, editor, *NIPS 9*, pages 466–472, Cambridge, 1997. MIT Press.

[2] R. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, New York, 1996.

[3] K. Rose, E. Gurewitz, and G. Fox. Statistical mechanics of phase transitions in clustering. *Physical Review Letters*, 65:945–948, 1990.

[4] E. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.

[5] T. Graepel, M. Burger, and K. Obermayer. Self-organizing maps: Generalizations and new optimization techniques. *Neurocomputing*, 21:173–190, 1998.

[6] Donald B. Rubin. EM and beyond. *Psychometrika*, 56(2):241–254, 1991.

[7] S. Rüger and A. Ossen. Clustering in weight space of feedforward nets. In C. Von der Malsburg, editor, *ICANN'96*, pages 83–88, Berlin, 1996. Springer.

[8] H. Kappen and J. Neijt. Neural network analysis to predict treatment outcome. *The Annals of Oncology*, 4:S31–S34, 1993.

[9] B. Bakker and T. Heskes. Model clustering by deterministic annealing. In *ESANN'99*, 1999.