Methods for Analyzing Internal Representations of Artificial Neural Networks

Yoshio Takane (McGill University)
Yuriko Oshima-Takane (McGill University)
Thomas R. Shultz (McGill University)

1. Introduction

Neural network models have recently been very popular in artificial intelligence, cognitive psychology, pattern recognition, etc. They appear to work remarkably well even for problems for which conventional statistical methods typically fail. However, how they achieve what they achieve is not understood sufficiently well. Feed-forward networks may be viewed as approximating (nonlinear) functions that connects inputs to outputs. We explore methods to analyze how the approximations are done. These methods range from a simple graphing technique to two-way and three-way constrained and unconstrained principal component analyses.

2. Preliminaries

Although most of the techniques we discuss apply to other network construction methods (e.g., back-propagation), all the results we present pertain to a particular method called cascade-correlation (CC) learning architecture (Fahlman & Lebiere, 1990). The following three features of the CC architecture make it particularly attractive:

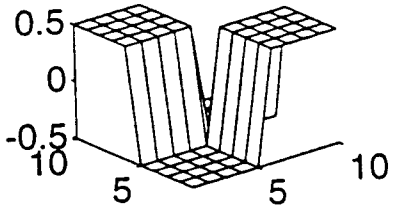
(a) No apriori net topology has to be specified. It always starts as a perceptron (i.e., a net with no hidden units), but it automatically adds hidden units to improve network performance until a satisfactory degree of performance is reached. Hidden units are added one at a time in such a way that all pre-existing units are connected to new ones.
(b) Bias and input units are always directly connected to output units (these connections are called cross connections) as well as to all hidden units. The cross connections often simplify the constructed net topology by capturing in the simplest possible way linear effects of bias and inputs, which often play crucial roles in learning.
(c) When a new hidden unit is recruited, activation patterns at that unit are determined by a heuristic method, and are fixed throughout the rest of learning process. This avoids the necessity of back-propagation, thus simplifies the learning algorithm, and leads to more stable and faster convergence.

There are a number of bench mark examples that we could use. For illustration we use a particular problem called continuous xor problem. This problem is simple enough to deduce what the optimal function is, and yet "complicated" enough to warrant network modelling. In particular, it requires an interaction effect between two input variables, and consequently no standard statistical methods work reasonably well unless the interaction effect is explicitly coded in.

The continuous xor problem has two input variables, x_1 and x_2, each ranging from .1 to 1.0, and one binary output variable, y. The problem is to discriminate two groups of input patterns. That is, whenever both x_1 and x_2 are greater than .55 or both smaller than .55, y=-.5, and whenever only one of the two variables is greater than .55 and the other less than .55, y=.5. The optimal discriminating function for this problem is

$$y=f(-a(x_1-.55)(x_2-.55))-.5,$$

Figure 1



where f is a sigmoid function (i.e., $f(t)=1/(1+\exp(-t))$, and $a \to \infty$. This function is depicted in Figure 1. A net is approximating this function,but how?

3. Contributions

The CC algorithm constructs a net topology and associated connection weights between units. A sample of 100 triplets of the form, (x_1, x_2, y), both x_1 and x_2 ranging from .1 to 1.0 in the step of .01, and y= -.5 or .5, were used in the training. For each input pattern a unit in the net sends contributions to units it is connected to. A contribution is defined as the product of the activation at

the sending unit and the connection weight. The receiving unit in turn forms an activation by summing contributions from its sending units and applying the sigmoid transformation to the summed contributions. An activation is "computed"at each unit in the net and for each input pattern in the training sample. Activations may be collected in the form of a matrix denoted by A. Let D_w represent a diagonal matrix of connection weights leading to the output unit. A matrix of contributions to the output unit (from which it computes its activations or output patterns) is obtained by

$$Z = AD_w.$$

This matrix plays an important role in the analyses to be suggested in the sequel. Let 1 denote a vector of ones. Let

$$z = Z1 = AD_w1 = Aw,$$

where $w=D_w1$. The sigmoid transformed z gives the net approximation to y.

4. Methods to Analyze Internal Structures of Net

(a) A simple graphing techniques. Elements of $z_j = Ze_j$ (j = 1,..., n, where n is the number of units in the net), where e_j is the n-element vector with the first j elements equal to one and the remaining elements equal to zero, may be plotted against x_1 and x_2 (Oshima-Takane, Shultz & Takane, in preparation). A plot of each z_j gives a successive approximation to y as more hidden units are added. This rather simple-minded graphing technique gives useful insights into how function approximations are done, when the number of input units is small (ideal for one or two, but perhaps one or more additional units). Similar techniques can be used for describing activations at hidden units as well as for so called developmental data. (Developmental data are collections of contribution matrices each taken immediately before a hidden unit is recruited.)

(b) Reduced-rank approximation. PCA may be applied to Z, and reduced-rank approximations to Z are obtained. Similar graphing techniques to the above maybe used to represent the approximated functions. In some cases reduced-rank approximations are found to give better approximations to y than the original Z matrix.

(c) Plotting component scores. Plots of component scores (after an appropriate simple structure rotation) are often informative for characterizing the nature of components, as done by Shultz and his collaborators (Shultz & Elman, 1994; Shultz, Oshima-Takane, 1994; Shultz, Oshima-Takane & Takane, 1994).

(d) Constrained PCA (CPCA). Matrix Z may contain known components. For example, it contains linear effects of input variables and bias. These known effects maybe eliminated before PCA is applied to the residuals. This will highlight more interesting aspects of Z, such as interactions among input variables. We also know Z1 is crucial for the net performance. We may set Z1 aside, and analyze the rest. More generally, Z may be decomposed into several submatrices, before each submatrix may be subjected to PCA (Takane & Shibayama, 1991). In general, partialling out known or trivial effects is an effective way of extracting unique contributions of particular units.

(e) Parafac. When there are more than one output unit, contributions to them are related by $Z_k = AD_{wk}$, where Z_k is the matrix of contributions to output unit k. These Z_k may be placed side by side and subjected to PCA as above, but a more interesting possibility is to apply a three-way PCA such as Parafac (Harshman, 1972). Parafac decomposes each Z_k by

$$Z_k = UD_kV',$$

where D_k is diagonal and specific to k, but U and V are common to all k. This makes sense because the basis matrix A is common to all Z_k. (However, diagonal elements of D_k could be negative.) The same technique may also be used for analyzing the developmental data, and for joint analyses of contributions for all units.

5. Results. Some results are presented at the meeting.

6. References.

Fahlman, S.E., & Lebiere, C. (1990). In Advances in neural network systems 2. Morgan Kaufmann.
Harshman, R. A. (1972). UCLA Phonetic Lab Working Paper.
Oshima-Takane, Y., Shultz, T.R., & Takane, Y. (in preparation). Understanding what a neural net does.
Shultz, T.R., & Elman, J.L. (1994). In Advances in neural network systems 6.
Shultz, T.R., Oshima-Takane, Y. (1994). Paper presented at the World Congress on Neural Networks, San Diego
Shultz, T.R., Oshima-Takane, Y., & Takane, Y. (1994). Paper submitted to NIPS.
Takane, Y., & Shibayama, T. (1991). Psychometrika, 97-120.