Analysis of Autoassociative Mapping Neural Networks

M. Shajith Ikbal Hemant Misra* B. Yegnanarayana
Department of Computer Science and Engineering, IIT Madras, India-600 036.
Phone: 044–2354591, Fax: 044–2350509, E–mail: yegna@iitm.ernet.in

Abstract

In this paper we analyse the mapping behavior of an autoassociative neural network (AANN). The mapping in an AANN is achieved by using a dimension reduction followed by a dimension expansion. One of the major results of the analysis is that, the network performs better autoassociation as the size increases. This is because, a network of a given size can deal with only a certain level of nonlinearity. Performance of autoassociative mapping is illustrated with 2-D examples. We have shown the utility of the mapping feature of an AANN for speaker verification.

1. Introduction

Autoassociative mapping neural networks (AANN) are basically feedforward neural networks [1–3] with structure satisfying the requirements for performing restricted autoassociation. General structure of AANN is shown in Fig.1. It consists of an input layer, an output layer, and one or more hidden layers. Number of units in the input and output layers are equal. Number of units in one of the hidden layers is less than the number of units in the input/output layer. This layer can be called as *dimension compressing hidden layer* as this layer causes the input to go through a dimension reduction process before appearing at the output. Units in the input and output layers are linear. Units in the hidden layer are either linear or nonlinear. For nonlinear units, a 'tanh' function is used as output function.

In autoassociative mapping, the target output pattern is identical to the input pattern. Backpropagation algorithm can be applied for AANN to arrive at optimal parameter values [4]. It has been shown that AANNs are useful in various recognition tasks such as speech recognition [5] and image compression [6]. Kramer [7] used AANNs for suppressing variabilities in chemical variables for real-time applications.

*Department of Electrical Engineering, Indian Institute of Technology, Madras 600 036

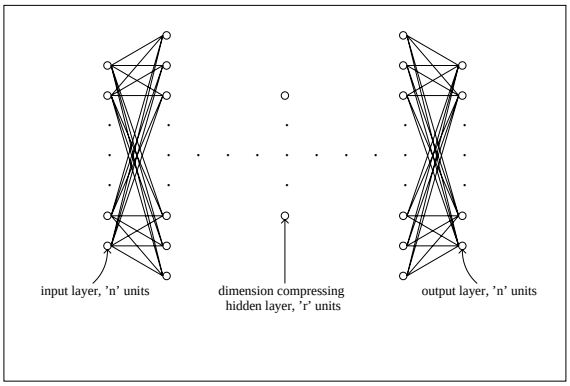

Figure 1: General structure of AANN.

An AANN with only linear units will perform linear autoassociation. A 3-layer linear AANN will perform PCA [3]. Bourlard and Kamp analysed linear autoassociative networks based on singular value decomposition [8]. Baldi [9] gave a complete description of the error landscapes in terms of principal component analysis for linear AANNs. Bishop analysed AANNs in the context of dimensionality reduction. He stated that a 5-layer network with nonlinear units in the hidden layers can perform better dimensionality reduction than a 3-layer network [3]. A detailed analysis of autoassociative mapping is still lacking. In this paper, we analyse in detail the feature of an autoassociative mapping. The analysis gives interpretation of the roles of different parameters of an AANN.

In Section 2, we give the analysis of an AANN. In Section 3, we apply the autoassociation feature to the problem of speaker verification, basically to show that the analysis is valid for real world problems also. Throughout the analysis 2-D examples are used to illustrate the higher dimensional analogies. Data used for such examples is shown in Fig.2. This data is generated artificially such that it will have certain number of clusters. Each cluster is generated such that its variance along the direction of maximum variance and the orthogonal direction are 0.6 and 0.2, respectively.

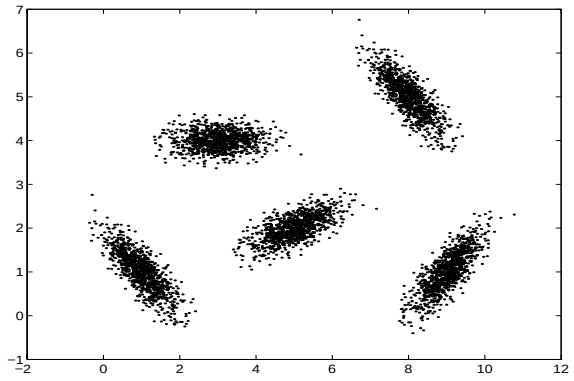

Figure 2: Artificially generated data for 2-D examples.

2. Analysis of AANN

Autoassociative mapping should reproduce an input vector at the output with least error. For example, for the case of 2-D data shown in the Fig.2, the outputs produced by mapping should be as close to the input as possible. Ideally, the output should be the same as the input.

AANN learns autoassociative mapping by training it in the autoassociative mapping mode with training patterns. In the autoassociative mode of training, the input patterns and the target output patterns should be the same. Let $Q = \mathbf{x_i}$, $i = 1, 2, ..., N$. be the set of training vectors.

Let $\mathbf{F}$ denote the autoassociative mapping learnt by the network. If $\{\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_N}\}$ is the set of output vectors produced by the AANN when the training vector set $\{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}\}$ is given as input, then $\mathbf{F}$ minimizes the mean square error given by the equation

$$E = \sum_{i=1}^{N} ||\mathbf{x_i} - \mathbf{y_i}||^2 = \sum_{i=1}^{N} ||\mathbf{x_i} - \mathbf{F}(\mathbf{x_i})||^2 \qquad (1)$$

For the network of our interest, the mapping function $\mathbf{F}$ can be separated into $\mathbf{F_1}$ and $\mathbf{F_2}$, so that

$$\mathbf{F}(.) = \mathbf{F_2}(\mathbf{F_1}(.)) \qquad (2)$$

where

$\mathbf{F_1}$ is the transformation in the network from the input layer upto the dimension compressing hidden layer, and

$\mathbf{F_2}$ the transformation from the dimension compressing hidden layer upto the output layer.

Assuming that the number of units in the input layer is n and the number of units in the dimension compressing hidden layer is r (where $r < n$), $\mathbf{F_1}$ transforms vectors in space R^n onto the space R^r. That is,

$$\mathcal{R}^n \xrightarrow{\mathbf{F_1}} \mathcal{R}^r$$

Likewise, $\mathbf{F_2}$ transforms vectors from the lower dimensional space R^r back to the space R^n at the output.

$$\mathcal{R}^r \xrightarrow{\mathbf{F_2}} \mathcal{R}^n$$

Since $r < n$, $\mathbf{F_1}$ is basically a dimension reduction process and $\mathbf{F_2}$ a dimension expansion process. Dimension reduction is achieved by projecting the vectors in the input space onto a subspace captured by the set of weights in the network part for $\mathbf{F_1}$. Dimension of the subspace is equal to the number of units in the dimension compressing hidden layer. Dimension expansion is achieved by mapping the lower dimensional vectors onto a hypersurface in the higher dimensional output space. Hypersurface is captured by the set of weights in the network part for $\mathbf{F_2}$. Subspace and hypersurface are in general nonlinear, because of the nonlinear units in the hidden layers.

Dimension expansion by mapping onto a hypersurface is obvious because, a set of lower dimensional vectors can not produce higher dimensional vectors of intrinsic dimensionality larger than the dimension of the lower dimensional space. Capturing the nonlinear subspace by the network part for $\mathbf{F_1}$ is explained as follows: Let us consider a network of structure x_iL-x_2N-...-x_{d-1}N-x_dL-x_{d+1}N-...-x_oL, where L denotes the linear units and N the nonlinear units. x_i, x_d, and x_o denote the number of units in the input layer, the dimension compressing hidden layer, and the output layer, respectively. If $x_i < x_{d-1}$, transformation $\mathbf{F_1'}$, done in the part of the network from the input layer upto the output of the layer just before the dimension compressing hidden layer, will map the input space onto a nonlinear hypersurface in the x_{d-1}-D space. The dimension compressing hidden layer will transform vectors in x_{d-1}-D space into x_d-D space, by projecting the higher dimensional vectors onto a linear subspace. Image of this subspace in the input space will be nonlinear. This can be generalized easily for the networks of any structure.

The nonlinearity level of the subspace and hypersurface depends on the size of the network. By increasing the size of the network in the part for $\mathbf{F_1}$, level of nonlinearity of the subspace can be increased. Similarly, by increasing the size of the network in the part for

$\mathbf{F_2}$, level of nonlinearity of the hypersurface can be increased.

As a result of dimension reduction and dimension expansion, autoassociative mapping as a whole is nothing but the projection of vectors in the input space onto a hypersurface in the same space. This is illustrated with the 2-D example as follows: For a network of structure 2L-1N-2L trained with the 2-D data shown in the Fig.2, the outputs generated when the training patterns themselves are given as inputs is shown in Fig.3. In the figure, the input patterns are also shown along with the output patterns. It can be seen from the figure that, outputs are on a straight line. This is because, the hypersurface is along that straight line for this network.

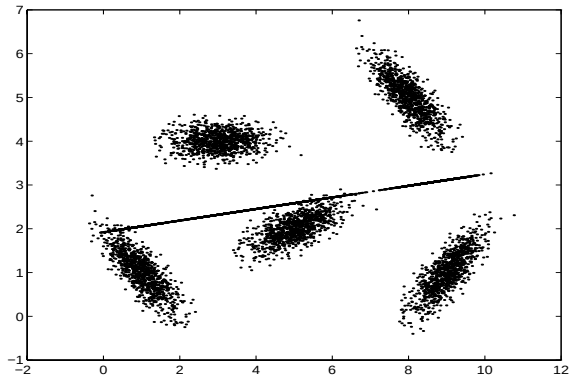

Figure 3: Mapping by a 3-layer network.

In order to achieve better autoassociation, the input vectors and the corresponding projected locations in the hypersurface must be as close to each other as possible. This can be achieved only if positional information about the higher dimensional vectors are retained to maximum extent during the dimension reduction, and such information is reproduced back in the higher dimensional space during the dimension expansion. (Positional information here refers to the relative spacing information, with respect to each other, of the higher dimensional vectors). For a subspace and hypersurface of specific dimensionality this will happen only when their spread in the space R^n are such that they are along the surface of maximum variance. (i.e., when the variance of the training vector set is maximum along their surfaces).

Training of the network in autoassociative mapping mode with the training set will make sure that the subspace and hypersurface will be captured along the surface of maximum variance, because, the training error will be least in such case. During training, the network will adjust the subspace and hypersurface so that they will finally capture the surface of maximum variance. From the 2-D autoassociation example shown in Fig.3, it can be seen that the outputs generated are along the line of maximum variance.

For a given training vector set, the surface of maximum variance of particular dimensionality differs for different level of nonlinearities. For example,

- If the nonlinearity is nil, i.e., for linear case, the surface of maximum variance is the principal subspace spanned by the principal eigenvectors of the correlation matrix given by,

$$\mathbf{C} = \mathcal{E}[\mathbf{x}\mathbf{x}^{\mathbf{T}}] \tag{3}$$

  where, $\mathcal{E}$ represents the expectation operator and $\mathbf{x}$ the training vectors.

- If infinite nonlinearity is possible then, the surface of maximum variance is a surface passing through all the points in the training set.

- If the possible nonlinearity is in between the above two extreme cases, the surface of maximum variance will be a surface satisfying the constraints on its nonlinearity.

Variance of the training vector set along the surface of maximum variance increases with increase in the level of nonlinearity. Variance along the principal subspace, which corresponds to the surface of maximum variance for zero nonlinearity, is the least. Variance along the surface passing through all the points, which corresponds to the surface of maximum variance for infinite nonlinearity, is the largest. Since the variance gives a measure of positional information, positional information along the surface of maximum variance will increase if the nonlinearity is increased. Hence, dimension reduction will be able to retain more positional information if the level of nonlinearity of the subspace is increased. Similarly, dimension expansion will be able to reproduce information back to the higher dimensional space effectively if nonlinearity in the hypersurface is increased. We have seen earlier that, nonlinearities in the subspace and hypersurface can be increased by increasing the sizes of the network, respectively in the parts for $\mathbf{F_1}$ and $\mathbf{F_2}$. Hence, the performance of autoassociation should improve with increase in the size of the network.

As we have seen already, the autoassociative mapping performed by a 3-layer network of structure 2L-1N-2L is linear because, there is no hidden layer between the dimension compressing hidden layer and the output layer. Fig.4 shows the results of the autoassociation performed by a 5-layer network of structure 2L-5N-1N-5N-2L. Since sizes of the network in the parts for $\mathbf{F_1}$ and

$\mathbf{F_2}$ are large for this network than the 3-layer network, subspace and hypersurface can be nonlinear. Hence, this network performs better autoassociation than the 3-layer network. It can be seen from the figure that, the outputs are brought closer to the corresponding inputs than in the previous case. Fig.5 shows the results of autoassociation performed by a 9-layer network of structure 2L-5N-5N-5N-1N-5N-5N-5N-2L. Since sizes of the network in the parts for $\mathbf{F_1}$ and $\mathbf{F_2}$ are large here than in the 5-layer network, the level of nonlinearity in the subspace and hypersurface can be large. As a result, it can be seen from the figure that, the inputs and the corresponding outputs are brought further close.

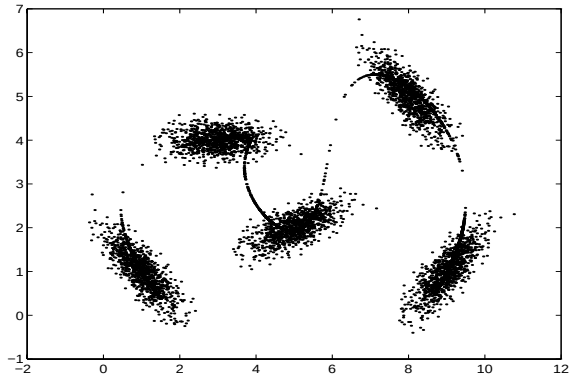

Figure 4: Mapping by a 5-layer network.

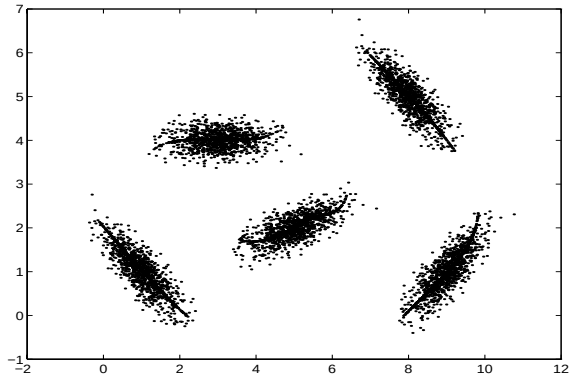

Figure 5: Mapping by a 9-layer network.

Mean square errors of different networks performing autoassociative mapping are given in Table 1. It can be seen from the table that, as the network size increases, the mean square error decreases, and hence the performance of the autoassociative mapping improves.

3. Speaker verification using AANN

Speaker verification is a process of accepting or rejecting the identity claim made by a speaker, based on the voice information in the speech signal. Uniqueness in the voice characteristics due to the uniqueness in the anatomical structure of the vocal tract for speakers is utilized for discriminating speakers. Generally, feature vectors representing the acoustic characteristics of the vocal tract are used for speaker verification because acoustic characteristics will be different for different shapes of the vocal tract.

Table 1: Mean square errors of 2-D AANNs of different sizes.

Network configuration	Mean square error
2L-1N-2L	4.058
2L-5N-1N-2L	3.942
2L-1N-5N-2L	2.798
2L-5N-1N-5N-2L	1.301
2L-5N-1N-5N-5N-2L	1.056
2L-5N-5N-1N-5N-5N-2L	0.699
2L-5N-5N-1N-3N-5N-5N-2L	0.692
2L-5N-5N-5N-1N-5N-5N-5N-2L	0.670

AANN trained with feature vectors derived from speech signal of a single speaker will capture subspace and hypersurface along the surface of maximum variance of the feature vectors. Since the acoustic characteristics are unique, the surface of maximum variance will be unique, and hence the subspace and hypersurface captured will also be unique. If a test feature vector is given to the network, it will give small error if the speaker of the test utterance is same as the speaker used for training (genuine testing). Otherwise (imposter testing) test error will be large. By fixing a threshold for test error, we can discriminate genuine speaker from the impostors.

A speaker verification system using AANN can be developed as follows: A separate network (AANN) is assigned for every speaker enrolled. These networks are trained in autoassociative mapping mode with the feature vectors derived from the training utterances. During the testing phase, feature vectors derived from test utterance will be tested against the model of the target speaker. The test error will be compared with a preset threshold to accept or reject the speaker. The speaker is accepted if the error is less than the preset threshold.

Performance of the speaker verification system is given by a measure called equal-error-rate (EER). EER is the value of the false rejection / false acceptance rates at a particular value of threshold for errors, where they become equal. False rejection is the case where the

genuine speaker is rejected and false acceptance is the case where imposter is accepted.

We have seen from the analysis that, the performance of autoassociative mapping will improve as the sizes of the network in the parts for $\mathbf{F_1}$ and $\mathbf{F_2}$ are increased. Hence, the performance of speaker verification also should improve with the size of the network used for the speaker model. This is because, as the nonlinearity of the surface of maximum variance increases, the uniqueness in the acoustic characteristics will be utilized well. Table 2 shows the results of the experiments conducted to demonstrate this point. Experiments are conducted with 38 speakers data from the 'dialect region 1' of TIMIT database. A 14^{th} order linear prediction cepstral coefficients with index weighting are used as feature vectors (19-D).

Table 2: Effect of network size on AANN-based speaker verification system performance.

Network configuration	Equal error rate
19L-6N-19L	15.79%
19L-30N-6N-19L	15.79%
19L-6N-25N-19L	7.97%
19L-6N-30N-19L	7.89%
19L-30N-6N-30N-19L	6.69%
19L-30N-6N-20N-30N-19L	5.90%
19L-30N-20N-6N-20N-30N-19L	5.94%

Each row in the table corresponds to a single experiment. Structures of speaker models used for the experiments are given in the first column of the table. From the table, it can be seen that EER decreases and hence verification performance improves as the network size is increased. Performance tends to reach a saturation value as the size is increased to a large value. This is because, as the size is increased, the number of patterns required to achieve good generalization also increases.

4. Summary and conclusions

In this paper, we have analysed the mapping behavior of autoassociative mapping neural networks. The network performs better autoassociation as the size is increased. The reason for this is that a network of particular size can deal with only a certain level of nonlinearity. This was illustrated with 2-D examples. The validation of the analysis is demonstrated for a real-world application, by applying the AANN to speaker verification problem.

References

[1] S. Haykin, *Neural Networks.* New York: Macmillan, 1994.

[2] B. Yegnanarayana, *Artificial neural networks.* New Delhi: Prentice-Hall of India Pvt. Ltd., 1999.

[3] C. M. Bishop, *Neural networks for pattern recognition.* Oxford: Clarendon press, 1995.

[4] D. E. Rumelhart, G. E. Hinton, and R. T. Williams, "Learning internal representations by error propagation," in *Parallel distributed processing: Explorations in the microstructure of cognition* (D. E. Rumelhart, J. L. McClelland, and the PDP research group, eds.), vol. 1, pp. 318–362, Cambridge: MA: MIT press. Reprinted in Anderson and Rosenfield, 1988.

[5] J. L. Elman and D. Zipser, "Learning the hidden structure of speech," *J. Acoust. Soc. Amer.*, vol. 83, pp. 1615–1626, 1987.

[6] G. W. Cottrell, P. W. Munro, and D. Zipser, "Image compression by back propagation: a demonstration of extensional programming," in *Advances in cognitive sciences* (N. E. Sharkey, ed.), vol. 2, Norwood (NJ): Abbex, 1988.

[7] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChe Journal*, vol. 37, pp. 233–243, 1991.

[8] H. Bourlard and Y. Kamp, "Auto–association by multilayer perceptrons and singular value decomposition," *Biol. Cybernet.*, vol. 59, pp. 291–294, 1988.

[9] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from from examples without local minima," *Neural Networks*, vol. 2, pp. 53–58, 1989.