Linear Replicator in Kernel Space

Wei-Chen Cheng and Cheng-Yuan Liou*

Department of Computer Science and Information Engineering
National Taiwan University
Republic of China
cyliou@csie.ntu.edu.tw

Abstract. This paper presents a linear replicator \[2,4\] based on minimizing the reconstruction error \[8,9\]. It can be used to study the learning behaviors of the kernel principal component analysis \[10\], the Hebbian algorithm for the principle component analysis (PCA) \[8,9\] and the iterative kernel PCA \[3\].

Keywords: Replicator, Principal component analysis, Generalized Hebbian algorithm, Kernel Hebbian algorithm, Gaussian kernel.

1 Introduction

The replicator is constructed by the multilayer perceptron and has many applications \[2,4\]. This paper presents a linear replicator and its training algorithm based on minimizing the reconstruction error \[9,8\] in the kernel space. It can facilitate the study on the principal component analysis (PCA). The PCA projects data onto several selected orthogonal bases which preserve variational information. Those bases are called principal components. The projection process is a linear transformation. The kernel PCA \[10\] applies a nonlinear transformation, that projects data onto a very high dimensional space based on Mercer’s theorem \[6-1\], and finds principal components in that high space. The space complexity of the kernel PCA is the square of the number of data. This complexity is severe in many large scale applications. An iterative kernel PCA, called kernel Hebbian algorithm (KHA) \[3\], is devised for on-line learning to reduce the size of the storage. The technique of the generalized Hebbian algorithm (GHA) \[9\] is used in the KHA. The replicator is also constructed in the high dimensional space. Its energy function and training algorithm are formulated in the next section.

2 The Linear Replicator

The replicator is illustrated in Fig. 1. There are three layers, input layer, output layer and hidden layer. All neurons are linear elements. Both input and output layers have \(N\) neurons. The hidden layer has \(M\) neurons and \(M\) is less than \(N\) usually. The weight matrix of the synapses connecting the input layer and the hidden layer is \(W\) and the matrix connecting the hidden layer and the output layer is \(W^T\), where \(W^T\) is the transpose

* Corresponding author. Supported by National Science Council.
of the matrix \( W \). \( W \) is an \( N \)-by-\( M \) matrix. This replicator possesses the self-similarity structure in the two weight matrices.

According to the kernel PCA, each \( D \)-dimensional data, \( \mathbf{x}_p \in \{ \mathbf{x}_p; p = 1, \ldots, P \} \), is mapped to the \( N \)-dimensional space using a pre-designed mapping function \( \Phi, \Phi(\mathbf{x}_p) : \mathbb{R}^D \to \mathbb{R}^N \), where \( N \gg D \). \( \Phi(\mathbf{x}_p) \) is an \( N \)-dimensional column vector. Let the \( N \)-by-\( P \) matrix \( X \) contain all mapped data, \( X = \left[ \Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_P) \right] \). We plan to find the \( M \) principal components which contain large amounts of variational information in the \( N \)-dimensional space, \( \mathbb{R}^N \).

The weight matrix of the synapses connecting the input layer and the hidden layer is \( W = [w_1, w_2, \ldots, w_M] \). Each column vector \( w_q \) contains all weights of the \( q \)th hidden neuron. According to the kernel PCA, \( w_q \) is a linear combination of all mapped data,

\[
\mathbf{w}_q = \sum_{p=1}^{P} a_{pq} \Phi(\mathbf{x}_p) = X \mathbf{a}_q, q \in \{1, \ldots, M\}.
\]  

Let \( A \) be an \( P \)-by-\( M \) matrix whose elements are the coefficients of the linear combination,

\[
A = [\mathbf{a}_1, \ldots, \mathbf{a}_M].
\]  

We get \( W = XA \). Let the matrix \( Y \) contains the \( P \) outputs of the \( M \) hidden neurons,

\[
Y = [y_1, y_2, \ldots, y_P] = W^T X
\]