Incremental Learning of Limited Kernel Associative Memory

Yuji Aoyama (P)1, and Koiciro Yamauchi1

1 Department of information science, Chubu University
E-mail: tp110001-0732@sti.chubu.ac.jp, yamauchi@cs.chubu.ac.jp

Abstract—This paper proposes a limited kernel associative memory, where the number of kernels is limited to a certain number. This model aims to be used on embedded systems with a small amount of storage space. The learning algorithm for the kernel associative memory is an improved version of the limited general regression neural network, which was proposed by one of the authors. In the experiments, we show the LKAM is able to continue incremental learning of new instances by pruning redundant memory.

Keywords—limited general regression neural network, limited kernel associative memory, kernel method

1 Introduction

Recently, various new associative memory methods, which outperform the traditional correlation based associative memory, are proposed[1, 2]. The kernel associative memory solves the problem of the traditional one by converting input vectors to corresponding infinite dimensional vectors(e.g.[1]). It also supports incremental learning. The associative memory methods based on neural-gas also realize incremental learning of new instances[2]. However, these methods cannot be used if the number of units / kernels is fixed. In this paper, we propose an incremental learning algorithm for an associative memory, where the number of kernels is fixed to a certain number: Limited kernel associative memory(LKAM). The fundamental architecture of the proposed method is the same as the kernel associative memory, and its learning algorithm is an improved version of the limited general regression neural network(LGRRN)[3]. If a new instance is presented, LKAM append a new kernel to record the instance. If the number of the kernels reaches the upper bound, LKAM detects the most ineffective kernel by using a kernel method, and replace the kernel with a new kernel to record the new instance. Before the replacing process, the most ineffective kernel duty is substituted by modifying other kernel’s output weights. Yoshizawa et.al.[4] proposed a similar associative memory to realize incremental learning with limited number of units. The pruning method of their model is based on the importance of the new current input. On the other hand, the pruning method of the LKAM is based on the generalization capability. Moreover, LKAM has the ability to interpolate between memories if an unknown sample is stimulated.

2 Methods

The structure of the LKAM is similar to that of the limited general regression neural network. However, the number of output units is the same as the number of inputs. The outputs represents the recalled memory from the inputs. Output of LKAM is

\[ Y(x) = \sum_j W_j \Phi_j(x) / \sum_i R_i \Phi(x) \]  (1)

Here \( \phi_j(x) \) denotes the output value of the jth kernel. \( \phi_i(x) \) is shown by the following expression.

\[ \phi_j(x) = \exp \left( -\|x - u_j\|^2 / 2\sigma^2 \right) \]  (2)

Where \( u_j \) denotes the center of kernel. \( \sigma \) is standard deviation that represents the width of the parameter \( W_j \) and \( R_j \) show the weight of connection between middle unit and the j-th kernel and the number of learned samples respectively. When the number of kernels reaches the upper limit, LKAM basically replace the most ineffective unit with a new kernel for recording a new instance.

3 Detection of the most ineffective kernel

The most ineffective kernel is detected by using a kernel method. The Gaussian kernel function used in Eq(1) can be rewritten as \( \sigma_j(x) = \Phi(x)^T \Phi(u_j) \), where \( \Phi(x) \) denotes an infinite dimensional feature vector for \( x \) on Hirbert space. This means that if \( \Phi(u_j) \) is represented by a linear combination of other vectors, the \( i \)-th kernel is redundant. LKAM calculates a ratio of redundancy using below equation.

\[ \delta_i = \arg \min_a \| \Phi(u_i) - \sum_{j \neq i} a_{ij} \Phi(u_j) \| ^2 \]  (3)

Where \( \delta_i \) denotes the redundancy of the \( i \)-th kernel. If \( \delta_i \) is the smallest of all, the \( i \)-th kernel is the most ineffective kernel. The above equation can be easily calculated using the kernel trick. The coefficient \( a_{ij} \) is also derived by using the least square method.

4 Incremental learning

LKAM incrementally learns new instances by adding new kernels. However, if the number of kernels reaches the upper bound, the most ineffective unit is basically replaced with a new kernel for recording the new instance. Before the replacement, the other kernels substitute the duty of the redundant kernel by modifying their output connection weights. However, there are possibility that the substitution and replacement processes greatly destroy past memories stored in LKAM. To avoid the degradation of the past memory, LKAM selects the most appropriate learning option from the following four options.

(1)Substitution and pruning with replacement: The most ineffective kernel is pruned with replacement to
a new kernel. Before the replacement, the duty of the kernel is substituted by other kernels. If the \( i \)-th kernel is the most ineffective kernel, the substitution is realized by

\[
W_j' = a_{ij}(W_i/R_i), \quad R_j' = a_{ij}R_i
\]  

(4)

Then, the \( i \)-th kernel parameters are set as follows.

\[
u_i = x, \quad W_i = x, \quad R_i = 1
\]  

(5)

(2) Pruning with replacement: The most ineffective unit is replaced with a new kernel, which records the new instance. This learning option is for avoiding the adverse effect of the substitution process. The replacement algorithm is the same as Eq (5).

(3) Modification: If the new instance is very closed to the existing kernel center, the replacement might be wasteful. In this option, the connection weights \( W_j \) and \( R_j \) of the nearest kernel is updated by the following equation:

\[
W_j' = a_{\text{new}j}x, \quad R_j' = a_{\text{new}j}
\]  

(6)

where \( a_{\text{new}j} \) is derived by Eq (3), whose \( \Phi(u_i) \) is replaced with \( \Phi(x) \).

(4) Ignore: If the adverse effect of the learning is very large, LKAM should not execute the learning. This learning option is for ignoring the new instance.

Due to the space limitation, we omit the detailed explanations for the LGRNN learning. Reference [3] describes the detailed algorithm.

5 Preliminary results

In the experiment, the number of kernels was limited to five. Here, the number of kernels was set up to be less than the number of input samples. The learning samples were alphabetic character of \( 10 \times 10 \) pixels. First, the LKAM learned three characters ABC, next the LKAM incrementally learned the deformed character of ABC. Then, the LKAM also incrementally learned D (see figure 2). After the learning, the responses of the LKAM for the test patterns in Figure 3 were estimated.

6 Discussion

If there are seven learning samples the network normally needs at least seven kernels to memorize them. However, LKAM with five kernels correctly recalled the seven characters as the four categories of characters. This result suggests that LKAM is a suitable associative memory system for embedded systems which have limited memory capacity.

References


