IFAC Workshop on Adaptation and Learning in Control and
Signal Processing, and IFAC Workshop on Periodic Control
Systems, Yokohama, Japan, August 30 – September 1, 2004

SELF-ORGANISING ASSOCIATIVE KERNEL MEMORY FOR MULTI-DOMAIN PATTERN CLASSIFICATION

Tetsuya Hoya

Laboratory for Advanced Brain Signal Processing,
BSI RIKEN, 2-1, Hirosawa, Wakoh-City, Saitama 351-0198, Japan

ABSTRACT

This paper proposes a novel self-organising associative neural network model in terms of kernel memory. The objective of this paper is not to give a sophisticated learning scheme and its rigorous mathematical accounts but rather attempt to address a paradigm shift, which could potentially answer a number of critical issues related to the current artificial neural network architectures. In the new memory model, the notion of 'weights' between the nodes is totally different from that as in ordinary neural network models, in which the weights simply represent the strengths of the connections between pairs of nodes in the kernel memory each realised by a kernel unit. Hence any arduous and iterative tuning of weight parameters is not involved and thereby the neural memory does not inherently suffer from any numerically-related problems. The associative memory is constructed via a simple unsupervised learning algorithm motivated from the traditional Hebbian principle. In the simulation study, both the plasticity and performance of the novel neural network architecture are discussed within the pattern classification context through single and simultaneous multi-domain classification tasks.

1. INTRODUCTION

The scientific study of human learning and memory in psychology has established and provided us with a wealth of data [1]. In the study, a great variety of models that take these data into account have been proposed. These models suggest that the human memory system is a central part of the cognitive process. In general, creatures have a far more complex and flexible memory system, in both the structural and functional sense, than the simple connectionist models (or, artificial neural networks) which have as yet been proposed. Moreover, it is generally known that their memory systems exhibit capability in both remembering and forgetting (undesirable) events/objects and can therefore deal with various problems which are essentially unavoidable for living.

In the artificial neural network field, multilayered perceptron neural networks (MLP-NNs), which were pioneered in the early 1960's [2, 3], have played a central role in the study of pattern recognition tasks [4]. In MLP-NNs, sigmoidal functions are used for the nonlinearity, and the network parameters, such as the weight vectors between the input and hidden layers and those between hidden and output layers, are usually adjusted by the backpropagation (BP) algorithm [5, 6, 7]. However, it is now well-known that in practice the learning of the MLP-NN parameters by BP type algorithms quite often suffers from becoming stuck in local minima and requiring long period of learning, both of which are good reason for detracting their utility in on-line processing. This account also holds for training the ordinary radial basis function type networks [7] or self-organising feature maps (SOFMs) [8], since the method for network parameters resorts to a gradient-descent type algorithm, which normally requires iterative and long training. In addition, such networks normally need for training from scratch, i.e., when new training data is arrived (i.e., incremental training). The conventional MLP-NNs, radial basis function neural networks (RBF-NN)s [9], and SOFMs therefore do not seem to be attractive candidates for elucidating the learning mechanism of the brain (for more critical arguments in the existing artificial neural networks, the see also [10]). Moreover, similar to the aforementioned connectionist models, most of the recent works in support vector machines (SVMs) (for a concise survey, see e.g., [11]) have not been focused upon the issues of plasticity, albeit providing a vast amount of the mathematical / theoretical accounts and/or performance improvement.

In the early 1990's, Specht rediscovered the effectiveness of kernel discriminant analysis [12] within the context of artificial neural networks. This led him to establish the notion of a probabilistic neural network (PNN) [13]. Subsequently, Nadaraya-Watson kernel regression [14, 15] was reformulated as a generalised regression neural network (GRNN) [16]. In the neural network context, both PNNs and GRNNs have layered (though fixed) structures as in MLP-NNs and are categorised into a family of RBF-NNs [9] in which the hidden neurons are represented by Gaussian response functions (or, Gaussian kernels). From the statistical point of view, regardless of minor exceptions, it is intuitively considered that the selection of a Gaussian response function is reasonable for the general description of the real-world data, as represented by a consequence of the central limit theorem. While the roots of PNNs and GRNNs differ from each other, in practice, the only difference between PNNs and GRNNs (in the strict sense) is confined to their implementation; for PNNs the weights between the RBFs and the output neuron(s) (which are given identical to the target values for both the PNNs and GRNNs) are normally fixed to binary (0/1) values, whereas GRNNs generally do not hold such restriction in the weight setting.

The advantage of PNNs/GRNNs against such commonly used neural networks as MLP-NNs, RBF-NNs, SOFMs, or SVMs is that they are essentially free from tuning a number of network parameters to obtain a reasonable convergence rate or worrying about any numerical instability such as local minima or long and iterative training of the network parameters. By exploiting the property of GRNNs and PNNs, simple and quick incremental learning is possible due to their inherent memory-based architecture, whereby the network growing (i.e. incremental training) / shrinking is straightforwardly performed [17]. Moreover, in [18] it is reported that these types of networks exhibit the capability in accommodation of new classes, by exploiting the property of the straightforward network growing / shrinking.

2. THE SELF-ORGANISING ASSOCIATIVE KERNEL MEMORY

The concept of kernel memory is originally inspired from the PNNs/GRNN models, in which the network has a memory-based architecture and inherits the attractive property of straightforward network growing / shrinking. However, as described later, the kernel memory is more flexible in that it can be self-organised via an unsupervised learning algorithm and thereby essentially has no fixed structure, with allowing any kind of lateral connections between the kernel units.

Fig. 1 depicts the kernel unit used to construct the self-organising associative memory (SAKM). Although any distance metric be-

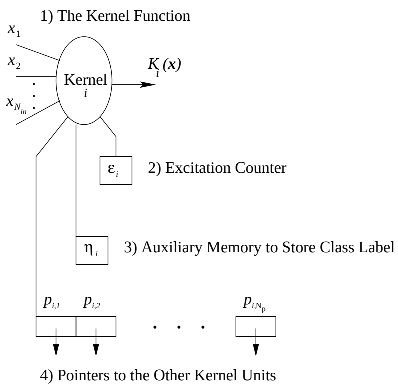

Fig. 1. The kernel unit.

tween two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ could be exploited for representing a kernel function within the kernel memory concept, we hereafter limit ourselves to consider a Gaussian response function as a kernel function, without loss of generality. Then, in Fig. 1, the activation of the kernel function $K(\boldsymbol{x})$ is defined as

$$K_i(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{c}_i) = \exp(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_i\|_2^2}{\sigma_i^2}) \qquad (1)$$

In the family of RBF-NNs, this is nothing more than an RBF where $\boldsymbol{x} = [x_1, x_2, \ldots, x_{N_{in}}]^T$ (T: vector transpose) is the input vector to the network, $\boldsymbol{c}_i$ is the centroid vector, and σ_i is the radius. However, as implicitly indicated in the figure, the output (or activation) of the i-th kernel function $K_i(\boldsymbol{x})$ is not directly transferred to other nodes in the post-layer(s) as in ordinary RBF-NNs (i.e., the output layer). In other words, the weight value itself is not directly used for the computation of the forwarding nodes; in the RBF-NNs, each output neuron is calculated as the total sum of the weight value *times* the activation of the hidden (i.e., the RBF) units, which in general yields the final results. Instead, where necessary, the links between the kernel and others are firstly established via the addressing pointers $p_{i,1}, p_{i,2}, \ldots, p_{i,N_p}$ which indicate the adjacent kernels of K_i. Then, the 'link weight' w_{ij} between the kernel K_i and K_j ($j = 1, 2, \ldots, N_p, i \neq j$) is assigned, the value of which represents the strength of the connection in between.

As stated earlier, since the actual data is stored within the template (centroid) vector $\boldsymbol{c}_i$ the change in the values of the link weights does not affect the data stored within the template vector at all. Therefore, within the SAKM context, even a single kernel can exhibit a generalisation capability; in the application to pattern classification, a single kernel is to a certain extent capable of classifying the patterns (for a Gaussian kernel, cf. [19]). Moreover, unlike conventional layered-type neural networks, there is no constraint within the structure of SAKM, e.g., sparse representations or any lateral connections are allowed, whilst modelling the 'dense' structures similar to correlation matrix memory [7] or SOFMs is also possible, by exploiting the addressing pointer(s) of a kernel unit in Fig.1. Nevertheless, within the SAKM context, both the number of neurons and the weight connections are dynamically varied during the learning (or construction) phase.

In another respect, the proposed kernel concept lies between the symbolic connectionism and artificial neural networks. However, compared to a conventional symbolic approach such as the Minsky's Knowledge-Line (or, simply denoted as K-line) concept [20], the kernel memory can replace the ordinary symbolic approaches in that each node (kernel) can have a generalisation capability which could mitigate the 'curse-of-dimensionality' to a greater extent. In contrast, since the kernel concept inherits the properties of PNNs/GRNNs, the self-organising memory does not involve any numerically-related problems, which are in general of crucial to the overall performance, i.e., the convergence rate or numerical instability, as commonly found in the conventional artificial neural network literature.

In Fig. 1, apart from the kernel function and the link weights $\boldsymbol{w}_i = [w_{i,1}, w_{i,2}, \ldots, w_{i,N_p}]^T$, the kernel unit has both the excitation counter ϵ_i and auxiliary memory η_i to store the class label (or ID). The excitation counter ϵ_i is incremented every time the kernel is excited, i.e., when the kernel function satisfies the relation

$$K_i(\boldsymbol{x}) \geq \theta_K \qquad (2)$$

where θ_K is a given threshold. In this paper, the class label η_i is fixed when the input vector is assigned to the template (centroid) vector of the (Gaussian) kernel.

2.1. An Algorithm for Updating Link Weights Between the Kernels

In [21] (p.62), Hebb postulated that "*When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.*". In this paper, the 'link weights' (or simply, 'weights') between the kernels are defined in this neurophysiological context. Namely, the following conjecture can firstly be drawn:

> **Conjecture 1**: When a pair of kernels K_i and K_j ($i \neq j$) in the SAKM are excited repeatedly, a new link weight w_{ij} between K_i and K_j is formed. Later, if this is occurred intermittently, the value of the link weight w_{ij} is increased.

In the above, the Hebb's original postulate for the physical locations of the adjacent cells A and B is not considered, since, in the actual implementation of the proposed scheme (to the memory system of a robot, etc), it is not always necessary for such place adjustment of the kernels. Secondly, the Hebb's postulate implies that the excitation of the cell A may be occurred due to the *transfer* of activations from other cells via the synaptic connections. This leads to the following conjecture:

> **Conjecture 2**: When a kernel K_i is excited and one of the link weights is connected to the kernel K_j, the excitation of K_i is transferred to K_j via the link weight w_{ij}. However, the amount of the excitation depends upon the value of the link weight.

Based upon the conjectures 1 and 2 above, the following algorithm for updating the link weights between a pair of the kernels K_i and K_j is given:

[The Link Weight Update Algorithm]

1) If there is already established the link weight w_{ij}, decrease the value according to:

$$w_{ij}(t) = w_{ij}(t-1) \cdot \exp(-\xi) \qquad (3)$$

(This simulates the time-wise synaptic decay.)

2) If the subsequent excitation of a pair of kernels K_i and K_j ($i \neq j$) is occurred (the excitation is judged by (2) given earlier) and repeated for p times, the link weight w_{ij} is updated as

$$w_{ij} = \begin{cases} w_{init} & ; \text{if } w_{ij} \text{ does not exist} \\ w_{max} & ; \text{else if } w_{ij} > w_{max} \\ w_{ij} + \delta & ; \text{otherwise} \end{cases} \qquad (4)$$

where ξ, w_{init}, w_{max}, and δ are all positive constants. Both the conditions **1)** and **2)** within the link weight update algorithm also agree with the rephrase of the Hebb's principle [22, 23]:

1. If two neurons on either side of a synapse are activated asynchronously, then that synapse is selectively weakened or eliminated.

2. If two neurons on either side of a synapse (connection) are activated simultaneously (i.e., synchronously), then the strength of that synapse is selectively increased.

Note that, to meet the second rephrase above, a decaying factor is introduced within the link weight update algorithm (in Condition **1)** above), to simulate the synaptic elimination (or decay). In this paper, the second rephrase above is extended and interpreted such that 1) the decay can always be occurred in time-wise (though the amount of such decay is considered to be slight in a short period of time) and 2) the synaptic decay can also be caused when the other kernel(s) is/are activated via the transmission of the activation of the kernel. In the link weight decay of the SAKM, the former is represented by ξ, whereas the latter is on the assumption that the potential of other end may be (slightly) lower than the one. At the neuro-anatomical level, it is known that a similar situation occurs due to the changes in the transmission rate of the spikes [21, 24] or the decay represented by e.g., the long term depression (LTD) [25]. These can lead to the modification of the second rephrase and thus the following conjecture can be also drawn:

Conjecture 3: When the kernel K_i is excited by the input $\boldsymbol{x}$ and has the connection to the kernel K_j via the link weight w_{ij}, the activation of K_j is computed by the relation

$$K_j(\boldsymbol{x}) = \gamma w_{ij} I_i \tag{5}$$

where γ ($0 << \gamma \leq 1$) is the decay factor and I_i is defined as an indicator function:

$$I_i = \begin{cases} 1 & \text{; If the kernel } K_i \text{ is excited (given in (2))} \\ 0 & \text{; otherwise} \end{cases}$$

In the above, the indicator function I_i is sufficient to describe the situation where an impulsive spike (or the action potential) generated from one neuron is transmitted to the other via the synaptic connection (for a thorough discussion, see, e.g., [24]), due to the excitation of the kernel K_i, within the context of modelling SAKM. The above also indicates that, apart from the regular input vector $\boldsymbol{x}$, the kernel can be excited by the secondary input, i.e., the transfer of the activations from other nodes, unlike the conventional neural architectures.

Consequently, both the construction of an SAKM (or the training phase) and the manner of testing the SAKM for pattern classification tasks are summarised as follows:

[Summary of Constructing A Self-Organising Kernel Memory]

Step 1) Initially ($cnt = 1$), there is only a single kernel in the SAKM, with the template vector identical to the first input vector presented, namely, $\boldsymbol{c}_1 = \boldsymbol{x}(1)$ (and setting η_1 to the corresponding class ID). If a Gaussian kernel is chosen, the unique setting of the radius σ (in (1)) may be determined *a priori*.

Step 2) For $cnt = 2$ to {num. of input data to be presented}, do the following:

Step 2.1) Calculate all the activations of the kernels K_i ($\forall i$) in the SAKM by the input data $\boldsymbol{x}(cnt)$, (as given by (1)). Then, if $K_i(\boldsymbol{x}(cnt)) \geq \theta_K$ (as in (2)), the kernel K_i is excited. Check the excitation of kernels via the link weights $\boldsymbol{w}_i$, by following the principle in **Conjecture 3**. Mark all the excited kernels.

Step 2.2) If there is no kernel excited by the input vector $\boldsymbol{x}(cnt)$, add a new kernel into the SAKM, with setting its template vector to $\boldsymbol{x}(cnt)$ (and η_i: the corresponding class ID).

Step 2.3) Update all the link weights within the SAKM by following **[The Link Weight Update Algorithm]** given above.

[Summary of Testing the Self-Organising Kernel Memory]

Step 1) Present the input data $\boldsymbol{x}$ to the SAKM, and compute all the kernel activations by (1) within the SAKM. Check also the activations via the link weights $\boldsymbol{w}_i$, by following the principle in **Conjecture 3**. Mark all the excited kernels.

Step 2) Obtain the maximally activated kernel K_{max} by

$$K_{max} = \max(K_i(\boldsymbol{x})) \tag{6}$$

amongst all the excited kernels within the SAKM. Then, if a classification task is performed, the classification result can be obtained by simply restoring the class label η_{max} from the auxiliary memory.

In the above, the excitation counters ϵ_i are not exploited, without loss of generality, though such exploitation can lead to more flexible learning algorithms.

2.2. Simultaneous Multi-Domain Data Processing

Fig. 2 shows an illustrative example of the SAKM which can process multi-domain data simultaneously.

In a systematic view point, the SAKM is a multi-(domain-)input multi-output (MIMO) system, whilst conventional layered type neural networks can be regarded as a single-(domain-)input multi-output (SIMO) system, since essentially only a single domain input is treated. In this example, three different domain vectors $\boldsymbol{x}^n = [x^n(1), x^n(2), \ldots, x^n(N_n)]^T$ ($n = 1, 2, 3$, and the length N_n can be varied, according to the modality or the feature extraction mechanism) are simultaneously presented to the SAKM and the same number of outputs (three) is obtained (thereby a three-input three-output system is considered). In the figure, $K_i^n(\cdot)$ denotes the i-th kernel which is responsible for the n-th domain input vector, and the bi-directional connections between the kernels or those between the kernels and output neurons represent the link weights w_{ij}. (Note that the three output neurons, o_1, o_2, and o_3, are depicted, instead of the auxiliary memory η_i attached to the respective kernel units, for the clarity.) It is also of notice that, as described earlier, the activation from the kernel K_1^3 can occur due to the excitation of the kernel K_1^1 via the link weight in between,

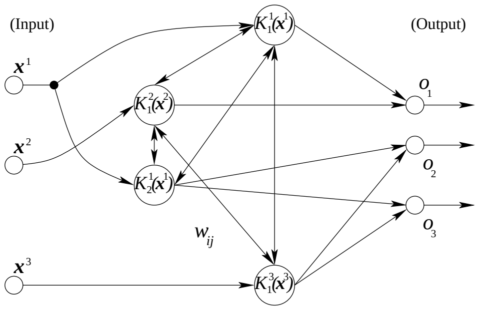

Fig. 2. An example of the SAKM for multi-domain data processing.

even if the regular input vector $\boldsymbol{x}^3$ does not cause such activation. (In this paper, this sort of link is also referred to as an 'associative link'.) For instance, provided that the input vector $\boldsymbol{x}^1$ is given as the feature vector of the voice sound uttered by a particular person and given to the kernel K_1^1, it is possible to design an SAKM such that, without presenting the feature vector $\boldsymbol{x}^3$ extracted from, say, the corresponding facial image, the kernel K_1^3 can be simultaneously activated by other sensory input. However, this simultaneous activation is occurred only when K_1^1 is excited, but not directly by the activation of the kernel K_2^1 (because there is no link between K_2^1 and K_1^3). Moreover, suppose that an appropriate built-in decoding mechanism of the template vector $\boldsymbol{c}_1^3$ is already given to the system, the facial image can eventually be restored from $\boldsymbol{c}_1^3$ by only presenting the feature vector of the voice sound $\boldsymbol{x}^1$.

As discussed earlier, the integrated information processing of this kind has not in general been considered by the conventional approaches and cannot be carried out by considering a simple mixture of the pattern classifiers, each of which is responsible for a particular domain classification task (or, in other words, the agents), as in typical modular approaches (see, e.g., [7]).

3. SIMULATION STUDY

In this paper, the objective of the simulation study is firstly (1) to validate the performance of the SAKM for single-domain pattern classification tasks using the datasets extracted from three different domain databases, i.e., Speech Filing System (SFS) [26], OptDigit, and PenDigit, where the latter two can be available from the 'UCI Machine Learning Repository' of the University of California. Then, the second objective is (2) to perform multiple (dual) domain pattern classification tasks, using a combination of the two datasets, the SFS and PenDigit, and observe how the kernels in one domain can subsequently excite (some of) the kernels in the other via the associative link weights so formed in between.

3.1. Parameter Settings

The SFS dataset contains a total of 900 patterns, whereas both the OptDigit and PenDigit sets respectively consist of a total of 1600 patterns. The description of the data sets is summarised in Table 1. For both the objectives (1) and (2) above, the parameters chosen arbitrarily for performing the actual simulations are also summarised in Table 2. As in Table 2, the combination of the parameters was chosen as uniquely as possible for all the three datasets, in order to perform the simulations in a similar condition. It was empirically confirmed that, as for the PNNs/GRNNs, though the selection of the radii σ_i yields a significant impact upon the generalisation capability, a unique setting of the radii value still

Data Set	Length of Each Pattern Vector	Total Num. of Patterns in the Training Set	Total Num. of Patterns in the Testing Sets
SFS	256	540	360
OptDigit	64	1200	400
PenDigit	16	1200	400

Table 1. Data sets used for the simulation study.

Parameter	Data Set		
	SFS	OptDigit	PenDigit
Decaying Factor for Excitation γ	0.95	0.95	0.95
Unique Radius for Gaussian Kernel σ	8.0	5.0	2.0
Link Weight Adjustment Constant δ	0.02	0.02	0.02
Synaptic Decaying Factor ξ	0.001	0.001	0.1
Threshold Value for Link Weights p	5	5	5
Initializing Value for Link Weights w_{init}	0.7	0.7	0.6
Maximum Value for Link Weights w_{max}	1.0	1.0	0.9

Table 2. Parameters chosen for the simulation study.

gives a reasonable performance for each dataset, which is similar to the case of PNNs/GRNNs, whereas the other parameters were arbitrarily chosen. Thus, during the construction phase of the SAKM, the settings $\sigma_i = \sigma$ $(^\forall i)$ and $\theta_K = 0.7$ were used. In addition, without loss of generality, the excitation of the kernels via the link weights was restricted only to the nearest neighbours (i.e., 1-nn) throughout the simulation study of this paper.

3.2. Single-Domain Pattern Classification

The objective of the single-domain pattern classification is to observe how the SAKM is organised and demonstrate that the SAKM can reasonably function as a pattern classifier.

Figs. 3 and 4 show respectively the variations in the monotonically growing number of the kernels and the link weights formed within the SAKM. To check the relative growing numbers for the

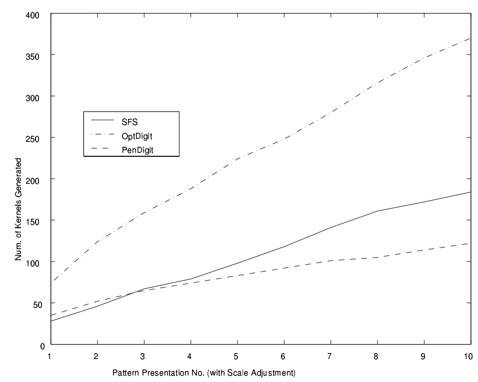

Fig. 3. Simulation results of single-domain pattern classification tasks - number of kernels generated.

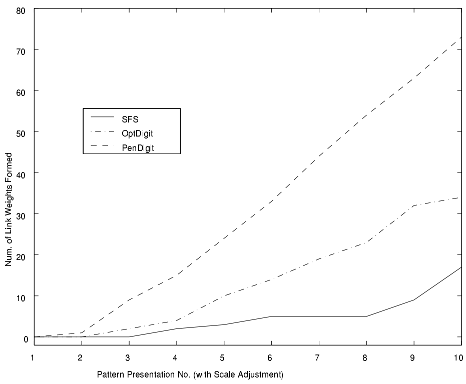

Fig. 4. Simulation results of single-domain pattern classification tasks - number of links formed.

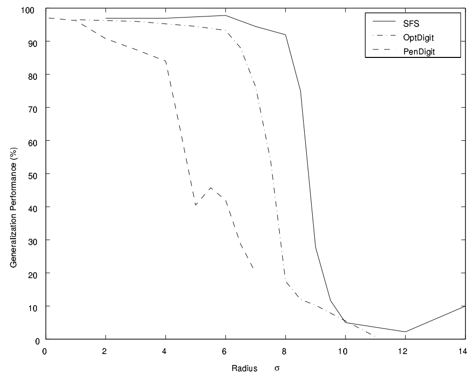

Fig. 5. Simulation results of single-domain pattern classification tasks - variations in the generalization performance of the SAKM with varying σ.

three different domain datasets, a normalised scale of the pattern presentation number is used (in the x-axis). In the figures, each number $x(i)$ $(i = 1, 2, \ldots, 10)$ in the x-axis thus corresponds to the relative number of the pattern presentation, i.e., $x(i) = i \times$ {the total number of patterns in the training set}/10.

From the observation of Figs. 3 and 4, it can be said that the data structure of the PenDigit dataset is relatively simple, compared to the other two, since the number of the kernels generated is always the smallest, whereas that of the link weights is the largest. As in Table 1, this is also considered due to the fact that, since the length of each pattern vector (16) is the shortest among the three, the pattern space can be constructed with a smaller number of data points than the other datasets.

	Total Num. of Kernels Generated within SAKM	General. Perf. of SAKM	General. Perf. of PNN with *k*-means
SFS	184	91.9%	88.9%
OptDigit	370	94.5%	94.8%
PenDigit	122	90.8%	88.0%

Table 3. Comparison of generalisation performance between the SAKM and a PNN using the *k*-means clustering algorithm.

3.2.1. Impact of the Selection σ Upon the Performance

As aforementioned, similar to PNNs/GRNNs, the behaviour of the SAKM (with Gaussian kernels) is particularly dependent upon the selection of the radius σ, amongst all the parameters. To investigate this, the value of σ was varied from the minimum Euclidean distance between a pair of pattern vectors in the training data set and the maximum. Then, it was empirically found that the number of the kernels generated as well as the overall generalization capability of the SAKM is dramatically varied, according to the value σ, as shown in Fig. 5; when σ is close to the minimum distance, the number of the kernels is almost the same as the number of patterns in the dataset. In other words, almost all the training data were exhausted during the construction of the SAKM for such cases, which is computationally expensive. However, it was found that in turn the decrease in the number of the kernels does not immediately corresponds to the relative degradation in terms of the generalisation performance. This tendency was also confirmed by examining the number of correctly connected link weights (i.e., the num. of the link weights which establish connections between the kernels with the same class labels, though the results are not shown in this paper).

Table 3 summarises the performance comparison between the SAKM constructed (i.e., the SAKM when the pattern presentation for the construction terminated) using the parameters given in Table 2 and a PNN with the centroid vectors found by the well-known MacQueen's *k*-means clustering algorithm. To make a fair comparison as much as possible, the numbers of the RBFs in the PNN responsible for the respective classes were fixed to those of the kernels within the SAKM. It was confirmed that for the three datasets the overall generalisation performance of the SAKM is almost the same/slightly better than the PNN + *k*-means approach, which is considered to be reasonable if the SAKM is applied to pattern classification tasks. However, the more advantageous point is that, unlike ordinary clustering schemes, the number of kernels necessary to yield a reasonable generalisation performance can be automatically determined by the unsupervised algorithm described in Section 2.1 and thus in this sense the SAKM approach is more flexible.

3.3. Simultaneous Dual-Domain Pattern Classification

The dual-domain pattern classification task in this paper was designed to imitate the situation where a specific voice sound (auditory) input to a particular area of memory excites not only the area responsible for the auditory modality but (in parallel) the visual, on the ground that appropriate built-in feature extraction mechanisms for the respective modalities are provided within the system. This is thus somewhat relevant to the issues of modelling associations between different cognitive modalities or, in a more general context, the 'concept formation' [21] / imagery, in which several perceptual processes are united together and thereby some sort of the integrated notion or, what is called, *Gestalt* (or 'data fusion') is formed.

For the actual simulation, both the SFS and PenDigit datasets were chosen, each of which constituted a sub-SAKM responsible for the specific domain data, and the cross-domain link weights (or, the associative links) established between a certain number of kernels within the sub-SAKMs were formed by the algorithms given in Section 2.1. The parameters for updating the link weights for the dual-domain task were almost the same as those of OptDigit case for the single-domain as in Table 2, except $\sigma = 8.0(2.0)$ for SFS(PenDigit) (which yielded the reasonable generalisation capability) and $w_{init} = 0.75$. For the formation of the associative links between the two sub-SAKMs, the same values as those for the ordinary links (i.e., the link weights within the sub-SAKM) given in Table 2 were used, except the synaptic decay factor $\xi = 0.0005$. Moreover, for imitating such cross-modality, it is natural to con-

	General. Perf. (GP) / Num. Excited Kernels via the Associative Links (NEKAL)	
	GP	NEKAL
SFS	86.7%	N/A
PenDigit	89.3%	N/A
Sub-SAKM(1) → (2)	62.4%	141
Sub-SAKM(2) → (1)	88.0%	125

Table 4. Generalisation performance of the dual-domain pattern classification task.

sider that the way of presentation may affect the formation of the associative links. In this paper, the patterns were presented alternatively across the two training data sets (viz., the pattern vector, SFS #1, PenDigit #1, SFS #2, PenDigit #2, . . .).

In Table 4, the overall generalisation performance of the dual-domain pattern classification task is summarised (besides the generalisation performance of each SAKM appeared in the first two rows). In the table, the title 'Sub-SAKM(1) → Sub-SAKM(2)' (Sub-SAKM(1) is responsible for the SFS data set, whereas Sub-SAKM(2) is PenDigit) denotes the overall generalisation performance obtained by the excitations of the kernels within Sub-SAKM (2) via the associative links from Sub-SAKM(1).

4. CONCLUSION

In this paper, a novel self-organising associative kernel memory (SAKM) has been proposed together with the Hebbian motivated learning algorithm between the kernel units. It should be noted that the proposed learning scheme is based upon relatively a straightforward rule and yet can construct a dynamic associative memory, whilst having the attractive features of PNNs/GRNNs, e.g. easy network growing/shrinking and the robust performance, which has not been possible/considered by the aforementioned conventional neural architectures. An attempt has then been made to achieve a paradigm shift from the conventional neural network concept by altering the notion of the 'weights' and thereby the issue of multi-domain data processing has been addressed. In this paper, though the discussion of the SAKM has been limited only within the pattern classification context, more general concept of kernel memory will be given as the neural foundation for modelling various psychological functions associated with the artificial mind system [27]. Future work includes a further analysis of the behaviour, the practical utility of the novel neural memory model to sensor fusion (e.g. [28]) direction of research, which is also related to the simultaneous multi-domain classification tasks, and the application to language processing.

5. REFERENCES

[1] Anderson, J. A. (1995). *Learning and Memory: an Integrated Approach* (2nd ed.). New York:John Wiley.

[2] Rosenblatt, F., (1962). *Principles of Neurodynamics*. Washington, DC:Spartan Books.

[3] Widrow, B. (1962). Generalization and information storage in networks of adaline 'neurons'. In M. C. Yovitz, G. T. Jacobi, & G. D. Goldstein (Eds.), *Self-Organizing Systems* (pp.435-461). Washington, D.C.: Sparta.

[4] Bishop, C. M. (1996). *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press.

[5] Werbos, P. J. (1974). Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph.D dissertation, Harvard University.

[6] Rumelhart, D.E. Hinton, G. E. & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1, Chapter 8, Cambridge, MA:MIT Press.

[7] Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. New York:Macmillan.

[8] Kohonen, T. (1997). *Self-Organizing Maps*. Springer-Verlag:Berlin.

[9] Darken, C. & Moody, J. (1992). Towards faster stochastic gradient search. In *Advances in Neural Information Processing Systems 4*, 1009-1016, San Mateo, CA: Morgan Kaufmann.

[10] Roy, A. (2000). Artificial neural networks - a science in trouble. *SIGKDD Explorations*, 1(2), 33-38.

[11] Hearst, M. A., Scholkopf, B., Dumais, S., Osuna, E. & J. Platt., J. (1998). Trends and controversies - support vector machines. *IEEE Intelligent Systems*, 13-4, 18-28.

[12] Hand, D. J. (1984). *Kernel Discriminant Analysis*. Research Studies Press.

[13] Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks*, 3, 109-118.

[14] Nadaraya, E. A. (1964). On estimating regression. *Theory Probab. Applic.*, 10, 186-190.

[15] Watson, G. S. (1964). *Smooth Regression Analysis*. Sankhy, Series A, 26, 359-372.

[16] D. F. Specht, (1991). A generalized regression neural network. *IEEE Trans. Neural Networks*, 2-6, 568-576.

[17] Hoya, T. (2004). Notions of intuition and attention modeled by a hierarchically arranged generalized regression neural network. *IEEE Trans. Systems, Man, and Cybernetics - Part B: Cybernetics*, 34-1, 200-209.

[18] Hoya, T. (2003). On the capability of accommodating new classes within probabilistic neural networks. *IEEE Trans. Neural Networks*, 14-2, 450-453.

[19] Poggio, T. & Edelman, S. (1990). A network that learns to recognize three-dimensional objects. *Nature*, 343-18, 263-266.

[20] Minsky, M. (1985). *The Society of Mind*, Simon & Schuster: New York.

[21] Hebb, D. O. (1949). *Organization of Behavior*. New York: Wiley.

[22] Stent, G. S. (1973). A physiological mechanism for Hebb's postulate of learning. *Proc. National Academy of Sciences of the U.S.A.*, 70, 997-1001.

[23] Changeux, J. P., & Danchin, A. (1976). Selective stabilization of developing synapses as a mechanism for the specification of neural networks. *Nature*, 264, 705-712.

[24] Gazzaniga, M. S., Ivry, R. B., & Mangun, G. R., (2002). *Cognitive Neuroscience - the Biology of the Mind*, 2nd Ed., New York:W. W. Norton & Company.

[25] Dudek, S. M., & Bear, M. F. (1992). Homosynaptic long-term depression in area CA1 of hippocampus and effects of N-methyl-D-aspartate receptor blockade. *Proc. Natl. Acad. Sci. USA*, 89: 4363-4367.

[26] Huckvale, M. (1996). *Speech Filing System Vs3.0 - Computer Tools for Speech Research*, London, U.K.: University College.

[27] Hoya, T. *Artificial Mind System - Kernel Memory Approach*. to appear from Springer-Verlag: Heidelberg.

[28] Wolff, G. J., Prasad, K. V., Stork, D. G., & Hennecke, M. (1993). Lipreading by neural networks: visual preprocessing, learning and sensory integration. In *Advances in Neural Information Processing Systems 5*, 1027-1034, San Mateo, CA: Morgan Kaufmann.