Hierarchical Fast Learning Artificial Neural Network: Progressive Learning in High Dimensional Spaces

Wong Lai Ping[1], Alex Tay Leng Phuan[2], Xu Jian[3]
[1,2] Nanyang Technological University, Singapore
[3] A* Star Singapore Institute of Manufacturing Technology
Email: ASLPTay@ntu.edu.sg

Abstract: This paper presents the Hierarchical Fast Learning Artificial Neural Network (HieFLANN) as a new model for learning high dimensionality data. The HieFLANN utilizes collaborating networks of K-Mean Fast Learning Artificial Neural Network (KFLANN) subnets [9] and a Canonical Covariance Feature Compression (C2FeCom) process. The HieFLANN is an inductive and unsupervised artificial neural network (ANN) that incorporates a hierarchical approach to address pattern classification for high dimensional data. The HieFLANN embeds individual KFLANN units (subnets) that produce a self-learning hierarchical network. The individual KFLANN subnet autonomously derives the essential localized network parameters from the input data and in the process, builds a hierarchical network that solves the larger problem. The C2FeCom feature compression process is also discussed in detail. It provides the HieFLANN with the capability of extracting the independent parameters in compact representations from subnets, allowing clusters of features to be formed. The proposed algorithm is experimentally evaluated using benchmark datasets.

1. Introduction

Hierarchical ANN models have been applied in many areas, ranging from social sciences and biological taxonomy to computer science and engineering. Some early work by Fukushima using hierarchical structures, gave rise to the NEOCOGNITION. A more recent model, the LAMINART derived from the 6 laminar-layered model of the biological vision model [6] explores the ability to discriminate varying orientations in a given scene. The LAMINART accounted for the complex visual task and was consistent with several physiological experiments conducted on the Inferior Temporal Cortex. Hierarchical ANN models allow complex learning problems to be solved by dividing the problems into a set of sub-problems. Another example is The Growing Hierarchical Self Organized Map (GHSOM) [2] that constructed layers automatically. Each network was composed of independent SOMs that adjust their size according to the requirements of the input space.

Hierarchical ANN models have its roots in biological systems. The fan out structure of each of the neuron in the neocortex provides evidence that the human brain processes information in a massively hierarchical way. The development of HieFLANN was motivated by the way brain processes information and experimental and modeling work published by other researchers [5], [6]. It utilizes the KFLANN as a subnet, connected as a hierarchical system to form a hierarchical framework of KFLANNs. In the proposed HieFLANN, clusters are grouped using similarity (dissimilarity) distance comparisons that are resident in the KFLANN clustering process. The topological organization of HieFLANN draws its architecture from the neurobiological layering structure. The inter-cluster correlation is generated by a canonical correlation profile of the independent KFLANN subnets in the HieFLANN. The eigen-decomposition of canonical correlation profile is the statistical eigen-decomposition on the respective canonical correlation matrix (CCM).

Although the solitary implementation of KFLANN, is already capable of good clustering results [1], [8], [9] and [10], the HieFLANN is investigated for the scalability of the KFLANN to handle higher dimensional problems. The HieFLANN architecture is shown in Figure 1. The construction of the HieFLANN is not a direct concatenation of KFLANN networks, but it is necessary to conduct some unsupervised preprocessing. The major advantages of the HieFLANN compared to the standard KFLANN are as follow:

- The individual unsupervised KFLANN subnet profile in the HieFLANN can be used for training multi-layer perceptron (MLP) learning algorithms. The overall MLP training time will be reduced since only the sub-set from the entire input space is used.
- The HieFLANN is able to uncover the hierarchical structure of the input space and thus provide more convenient way to analysis the input space. The explorative analysis of the small dimensional input space certainly easier as compared to a larger dimensional input space.
- Every subnet layering in HieFLANN produces clusters in small dimensions that ease the cluster analysis.
- Parallel processing can be implemented for each independent KFLANN subnet, reducing the computation load of the processor and improving the speed of the entire process.

The HieFLANN construction process consists of four major steps.

Step 1: Divide the input feature space into Homogenous Feature Subspaces (HFS). This is known as the feature partitioning stage. It involves the partitioning of the original feature space into multiple mutually exclusive sets of HFS. KFLANN is used as partitioning tool implement the Mahalanobis similarity measure. This similarity is used since it able to capture the distance of cluster with elongated variance with the implicit covariance. The HFS formed contains a set of features that are more likely to exhibit similar characteristic. An inherent property of the feature partitioning stage is the automated feature extracting capability.

Step 2: Processing each subnet assigned by HFS such that they occur synchronously and independent of each other. This is some form of localized distribution analysis. According to the distribution and coverage of the feature subset at a particular subnet, KFLANN is able to analysis the grouping of the presented data. Each KFLANN performs its independent clustering based on statistically obtained parameters.

Step 3: KFLANN subnet clustered results are then used to converge onto the third step where C2FeCom analysis is performed to compress the converging information. The goal behind the initial feature partitioning is for each subnet to align its feature space in preparation for the C2FeCom level. The proposed feature compression C2FeCom is motivated by the Canonical and Covariance analysis. C2FeCom feature compression method is purely statistical driven and can be viewed as feature extraction for feature dimensionality reduction. Canonical analysis employs the notion of elaborating discriminant function between k groups in the presented data set. Covariance tends to associate the correlation amongst features so that increase the efficiency of discrimination.

Step 4: The final step is yet another KFLANN clustering process, but the input space is now the compressed feature spaces obtained from step 3. This is the final adjustment to classify the presented data set on the compressed feature space.

HieFLANN can be viewed as unified framework for approaching problem of machine learning on manifolds of classification problem sets. KFLANN clustering subnet resident within HieFLANN handles this issue. High dimensionality of classification problems can lead to severe side effects, thus it is useful to adopt a divide and conquer approach to reduce the dimensionality without losing too much information. C2FeCom approach is introduced to tackle the problem. It extracts intrinsic dimension lying in the very high dimensional space of the problem set.

The organization of the paper is as follow. Section 2 discusses the HieFLANN algorithm and its architecture. In section 3, the KFLANN clustering algorithm is illustrated. This is followed by C2FeCom description in section 4. Section 5 gives the explanation on experimental evaluation. The conclusion is finally given in last section.

2. Hierarchical Fast Learning Artificial Neural Network

The HieFLANN operates on the divide and conquer principle. Similar decomposition concepts implemented in HieFLANN are often found when dealing with a complex task. It is a characteristic of intelligent behaviour to solve a complex problems by decomposing it into managable tasks. HieFLANN works with the C2FeCom and employs the unsupervised clustering method KFLANN that has been verified through several studies [1], [9], [10].

2.1. HieFLANN Architecture

The HieFLANN architecture is depicted in Figure 1. It is a formation of dynamic subnets made up of KFLANN units during the HFS generation. No assumption needs to be made on domain knowledge or known specification of parameters to construct the network model. The only parameter involved in KFLANN is the straight forward statistical computation of feature standard deviations as its tolerance setting. As these calculations follow a standardized process, the network can perform self-configuration of each feature tolerance. A detailed investigation of the tolerance setting and details can be found in [9].

The HieFLANN architecture, though seemingly complex, is built using duplicated KFLANN components. A parallel implementation of the network is therefore possible. KFLANN subnets act as independent processing modules that break the input space into smaller dimensions. These smaller dimensions allow the problem to be partitioned and later combined through a hierarchical network.

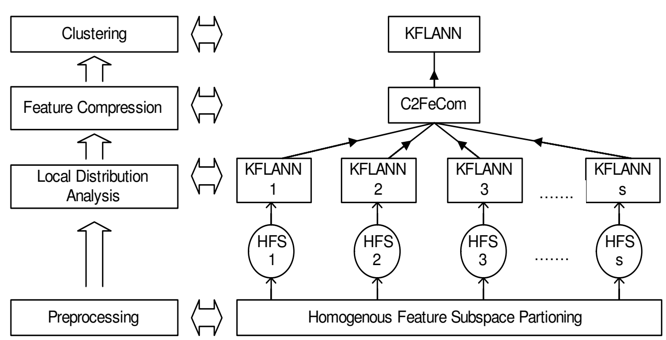

Figure 1. HieFLANN Architecture

2.2. HieFLANN Algorithm

The algorithm of HieFLANN follows.

Step 1 *Create homogenous feature subspace*
Step 2 *Perform KFLANN clustering on every homogenous feature subspace created*
Step 3 *Form canonical correlation matrix*
Step 4 *Perform eigen-decomposition on the canonical correlation matrix*
Step 5 *Compress feature space*
Step 6 *Perform KLFANN clustering using the compressed feature space in Step 5*

Preprocessing

In Step 1, the HFSs are created based on the similarity profile amongst the features. This is done by similarity computation on the feature space and it produces the feature similarity matrix. The Mahalanobis similarity is used with the equation:

$$D(X_i, X_j) = |X_i - X_j|^T * \mathrm{cov}(X) * |X_i - X_j| \qquad (1)$$

* $|a|$ *denote the absolute value of a.* X_i *is the* i^{th} *feature.*

Covariance encapsulate in Mahalanobis similarity is a measurement of the relationship between two ranges of data. The rational of having this step is to create HFSs that mitigate the effect of possible redundant features. KFLANN is then used to cluster the feature similarity matrix to form the Homogenous Feature Subspaces (HFSs). Features that are more similar bear higher changes to be grouped in the same HFS. Pictorial view is depicted in Figure 2, the raw data set at the bottom left hand side contributes to the feature similarity matrix as shown in right hand side. Equation (1) determines the similarity element of the feature similarity matrix. KFLANN clustering uses the feature similarity to partition the subnet space resulting the clusters denoted as HFSs.

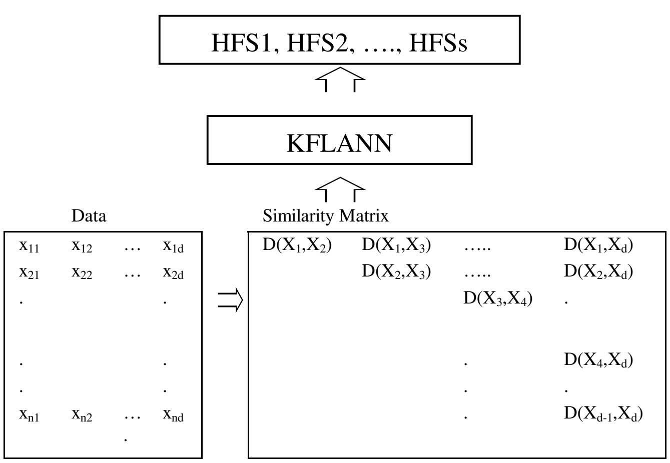

Figure 2. HFS Construction Process

Local Distribution Analysis

In step 2, the KFLANN is used to cluster the assigned input features within the localized HFS. The KFLANN subnet clustering process may executed simultaneously as every HFS grouping is independent. Section 3 provides detail description of KFLANN clustering process. Feature analysis is performed by KFLANN subnet on the HFS. The competitive learning of KFLANN using the properties of nearest neighborhood and vigilance properties of the Adaptive Resonance Theory (ART) are used to analyze the regularities of the feature subset. The KFLANN is able to analysis the grouping of the presented HFS data according to the subspace of the data distribution

The combinations of the input features by competitive layer ensure the feasibility of the higher order encoding of input features. The individual competitive neuron learns to specialize on a set of similar patterns and thus become “feature detectors”.

Feature Compression

Feature compression is performed through the proposed Canonical Covariance Feature Compression (C2FeCom), the canonical correlation matrix (CCM) is formed up using all HFSs and their corresponded clusters resolution. Eigen-decomposition is used to extract the eigenvectors from CCM which are the canonical vectors used for feature compression. Step 3 generates the CCM and step 4 extracts the canonical variates using the eigen-decomposition method. This is followed by feature compression process in step 5. The detail is given in section 4.

Clustering

Step 6 is the final step of the HieFLANN algorithm is the use of yet another KFLANN to cluster the compressed feature space. This portion can be supervised to provide the correct cluster label so that the label identity is independent of the feature compression clustering in the KFLANN subnets.

3. K-Mean Fast Learning Artificial Neural Network

KFLANN is used as the clustering algorithm that retrieves the localized salient information. Every KFLANN subnet of HieFLANN plays the role as categorizer clustering the inputs feeding the specific HFS. The KFLANN subnets have been designed as networks that categorize clusters within the HFS input space.

FLANN [3] was designed with concepts found in ART but imposed the Winner Take All (WTA) property within the algorithm. Further improvement was done to take in numerical continuous value in FLANN II [8]. The original FLANN II was restricted by its sensitivity to the pattern sequence. This was later overcome by incorporating the K-

Means centroid location which gave rise to KFLANN [1]. The KFLANN improves the sensitivity to pattern sequence but still did not have a solution for cluster stability.

The latest improvement on KFLANN [9] includes data point reshuffling which resolves the data sequence sensitivity that creates stable clusters. Clusters are said to be stable if the cluster formation is complete after some iterations and the cluster centroids remain consistent. The KFLANN is an unsupervised clustering algorithm that is

- A self-organized clustering ANN model.
- Autonomous in the determination of optimal cluster number. This enables it to generate new categories from the training exemplars dynamically.
- Not reliant of calibrated data set and there is no side effect from random data presentation sequences.
- Able to deal with stability-plasticity dilemma.
- Using competitive learning that favours the fault tolerance ability.
- Trained within a small definite number of epochs.

The above characteristics contribute to the merits of KFLANN clustering algorithm.

3.1. KFLANN Architecture

The KFLANN architecture is shown in Figure 3. KFLANN consists of a single input layer that integrates the source of the patterns. The output layer grows dynamically as new groups are formed during the clustering process. The weight connections between the input node and output node are the direct mapping of each element of input vectors. The dynamic formation of output nodes yield KFLANN is a self-organized clustering ANN model.

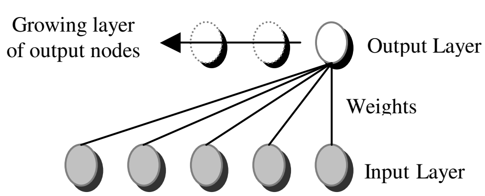

Figure 3. KFLANN Architecture

3.2. KFLANN Algorithm

The algorithm of KFLANN is as follow:

Step 1 *Initialize the network parameters*

Step 2 *Present the pattern to the input layer. If there is no output node, GOTO Step 6*

Step 3 *Determine all possible matches output node using*

$$\frac{\sum_{i=1}^{d} D\left[\delta_i^2 - \left(W_{ij} - X_i\right)^T * \mathrm{cov}(X) * \left(W_{ij} - X_i\right)\right]}{d} \geq \mu$$

$$D[a] = \begin{Bmatrix} 1, & a > 0 \\ 0, & a \leq 0 \end{Bmatrix}$$

D is the discriminant function

Step 4 *Determine the winner from all matches output nodes using*

$$winner = \min\left[\sum_{i=1}^{d} \left(W_{ij} - X_i\right)^T * \mathrm{cov}(X) * \left(W_{ij} - X_i\right)\right]$$

Step 5 *Match node is found. Assign the pattern to the match output node GOTO Step 2*

Step 6 *Create new output node. Perform direct mapping of the input vector into weight vectors*

Step 7 *If complete a single epoch compute centroids*
If centroid points of all clusters unchanged
Terminate
Else
GOTO Step 8
End if
Else
GOTO Step 2
End if

Step 8 *Find closest patterns to the centroids and reshuffle them to the top of the dataset list. GOTO Step 2*

δ_i is the tolerance for i^{th} feature of the input space, W_{ij} used to denote the weight connection of j^{th} output to the i^{th} input, X_i represent the i^{th} feature of the current pattern . d is feature dimensionality. μ is vigilance.

Apart from the initialization, the KFLANN clustering algorithm comprises of the following iterative steps.

Initialization

The input neurons at KFLANN's input layer integrate with input vector directly and no data set manipulation is required.

The initialization phase in Step 1, prior to the clustering process is needed to determine the vigilance μ and tolerance δ. The computation complexity is dependent on the size of the data set and the dimensionality of the input space. Vigilance setting is usually set between 0.5 and 1.0. Since the δ value determines the ratio between the number of features to be within tolerance and the total number of features, the clustering transformation becomes more stringent as the vigilance setting approaches 1. The Tolerance setting δ for a feature is the measurement of the allowable fluctuations. Several methods were investigated to deal with this tolerance setting [9] and the standard deviation computation was amongst the most optimal method.

Competitive Learning

The clustering process begins with the sequential presentation of data patterns. The tolerance filtering procedure is then performed on each input feature and an overall pattern vigilance screening is incorporated within Step 3.

Tolerance filtering can be viewed as localized screening that determines if the feature is within acceptable tolerance. It is implemented using the discriminating function *D[a]* in Step 3. If this discriminating function *D[a]* returns a value *1*, it indicates that the particular input attribute is within the bounds of the corresponding attribute of the cluster in question.

The vigilance screening, also known as global screening determines the number of features within the localized input space that have passed the tolerance screening. The vigilance screening is important as it enables the network to possess fault tolerance to missing value, outliers or noise feature.

After satisfying the screening criteria, there are still possibilities of exemplar matches that map into multiple clusters. The final decision of ownership is decided using the WTA competitive selection found in Step 4. Eventually, only the cluster centroid bearing the strongest weight will absorb the exemplar into its cluster. The winning neuron is chosen based on the closest distance from the data point to cluster centroid.

Step 5 is the Cluster Membership Assignment (CMA) that assigns the exemplar as a new cluster member of the selected winner centroid.

If a new exemplar pattern does not meet the two screening criteria, a new output centroid is created to represent the new cluster (Step 6).

Cluster Stability Checking

After an epoch (a full list of pattern has been processed), the cluster stability is checked. This is to determine if the cluster centroids represent the exemplars in the cluster. The algorithm has a dual way of creating cluster centroids. The first is a straight assignment when the exemplar is not represented in any available cluster, while the second is a deliberate shuffling of the data point to the top of the list in Step 7 of the KFLANN algorithm. The data point reshuffling process is performed only if cluster stability is not achieved. Re-shuffling the data points closest to cluster centroids to the top of pattern list create new pattern sequence for the subsequent clustering process (Step 8). KFLANN iterative process is limited to 5 iterations [9]. It has been shown experimentally that the cluster stability stabilizes after 3 iterations.

4. Feature Compression Using Canonical Covariance Analysis

Apart from the KFLANN, the feature compression method known as Canonical Covariance Feature Compression (C2FeCom) is key to the operation of the network. The C2FeCom is used to prepare salient features for clustering at final stage. It produces a combination of the input space with the purpose of maximizing the spread of data for different categories and minimizing the dispersion amongst similar categories.

The proposed feature compression process embedded in HieFLANN employs the canonical and covariance analysis. With the use of CCM, commonly studied in classical statistics, it is possible to extend the KFLANN clustering subnets into a hierarchical structure for a more complex, but robust ANN computation model. The theory of Canonical Analysis was originally developed to analyze relationship of a set of predictors to the presented set of criterion variable. The reasons of using the canonical analysis for feature compression method were because it was the only multivariate approach that is able to handle more than one metric criteria (dependent) variable and it was the only multivariate method that works with both criteria and predictor (independent) variables simultaneously.

The covariance analysis is attached to impress the feature variation in the compression process. Covariance between i^{th} feature (X_i) and j^{th} feature (X_j) is as follow:

$$\operatorname{cov}(X_i, X_j) = \frac{1}{n}\sum_{a=1}^{n}\left(x_{ia} - \mu_{X_i}\right)\left(x_{ja} - \mu_{X_j}\right) \qquad (2)$$

$$\operatorname{cov}(X) = \frac{1}{n}\sum_{a=1}^{d}\sum_{b=1}^{n}\left(x_{ab} - \mu_{X_a}\right)$$

With d is the feature dimensionality and n as the number of samples. μ_{X_a} *is the mean of feature* X_a *spanning over n data patterns.*

Notations used to describe the C2FeCom explained as follow. In this example, let $\mathcal{X}=\{x_1, x_2, x_3, x_4, x_5\}$, be the input feature space of dimensionality 5. Assume that the input feature space is divided into 2 HFSs namely HFS1 and HFS2. HFS1 may consist of $X_1=\{x_1, x_2, x_3\}$ and HFS2 consists of $X_2 =\{x_4, x_5\}$. As the inputs of each HFS are clustered by the KFLANN algorithm, the transformed clustering output $\mathcal{Y}$ is generated, such that $\mathcal{X}$ is transformed into $\mathcal{Y}$. In this example the two KFLANN subnets produce output the clustering outcome, where the first subnet deals with HFS1 (X_1) resulting Y_1, (represent the cluster index) while second subnet handles HFS2 (X_2) output Y_2.

4.1. Formation of Canonical Correlation Matrix

If $\mathcal{X}$ and $\mathcal{Y}$ are matrices, the CCM of 2 HFSs are showed in Table 1. The computation equation is given in (3), (4), (5) and (6) respectively.

Table 1. (a) CCM (b) Example of predictor sets $\mathcal{X}$ and criteria set $\mathcal{Y}$ for 2 HFSs

	X_1, X_2	Y_1, Y_2
X_1, X_2	R_{xx}	R_{xy}
Y_1, Y_2	R_{yx}	R_{yy}

(a)

x_1	x_2	x_3	Y_1
1	0.4	1.1	2
0.5	0.8	0.4	1
0.4	0.5	0.3	1
0.7	1.0	0.8	2
0.8	0.2	0.2	2

x_4	x_5	Y_2
0.6	0.7	2
0.5	0.2	1
0.1	0.2	1
0.8	0	3
0.7	0.3	1

(b)

$$R_{xx} = \sum (X_1 - \mu_{x1})(X_2 - \mu_{x2})^T \tag{3}$$

$$R_{yy} = \sum (Y_1 - \mu_{y1})(Y_2 - \mu_{y2})^T \tag{4}$$

$$R_{xy} = \sum (X - \mu_x)(Y - \mu_y)^T \tag{5}$$

$$R_{yx} = R_{xy}{}^T \tag{6}$$

R_{xx}, R_{yy}, R_{xy} and R_{yx} are in matrix form. The inter-correlation between predictor variables is given in R_{xx}, the cross-correlation between predictor and criteria variables is represented by R_{xy} and its transpose R_{yx}, and the inter-correlation between criteria variables is R_{yy}. To obtain the maximum correlation between the set of variables in HFSs, eigen-decomposition on the CCM is involved.

4.2. Eigen-decomposition of Canonical Correlation Matrix

Let *U* and *V* be the canonical variates to be determined such that the correlation between $\mathcal{X}' = U^T\mathcal{X}$ and $\mathcal{Y}'=V^T\mathcal{Y}$ is maximized. The canonical variate is the eigenvector of the CCM. It is the principal component of the respective matrix. Eigenvectors obtained from eigen-decomposition process are the decorrelated principal axes of the data set and grouping transformation. *U* is the canonical variate that exhibits the principal component of the feature variables, $\mathcal{X}$. *V* is the canonical variate for the subnet cluster representation variables, $\mathcal{Y}$. This forms up a pairwise canonical variate matrix contain of *U* and *V*.

4.3. Canonical Covariance Feature Compression

The combination of canonical analysis and covariance form the proposed feature compression method in this paper. The feature compression equation is given as follow:

$$C2FeCom\,(\mathcal{X}) = cov\,(\mathcal{X}) * \mathcal{X} * U \tag{7}$$

** is the multiplication symbol*

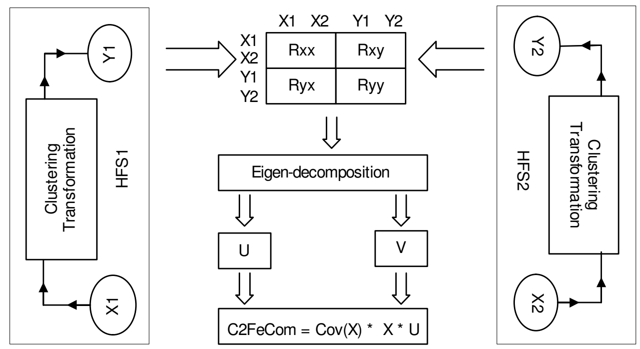

Figure 4. C2FeCom Process

Figure 4 depicts the feature compression process together with the formulation of CCM.

Diagonal elements of covariance matrix represent the feature variance. The covariance term used here is a factor loading in the compression process similar to the beta weights in a multiple regression equation.

The covariance term represents the correlation amongst the feature while canonical variates tend to minimize the projection within-group variances that have been pooled over all groups in each HFS, and maximize the between-group variances in each HFS. The correlation between heterogeneous partitions is then finalized through the CCM.

This summarizes the rationale of incorporating the covariance in the proposed C2FeCom. Other reasons include:

- It accounts for ranges of acceptability betweens features (variability). This mitigates the effect of the feature value that fall out acceptable range.
- The features correlation is exhibited thru the covariance term enable the compression atones for the relation amongst features.
- The covariance imposes weights that are distributed in the feature space. This eliminates the co-linearity effect of the dataset.

Although canonical analysis has been a decade old solution for feature reduction purposes, the common practice has been to take in the entire input space and perform canonical analysis to extract significant principal components. The proposed feature compression contributes to certain merits comparing to the conventional canonical analysis feature reduction method. Further explanation follows.

C2FeCom is considered as a principal component based feature reduction approach. In principal component based approach, the eigenvectors of either the covariance or correlation matrix is used as the principal component for feature transformation. The drawback is that principal component-based method ignores the different distributions representing the statistical classes. However this drawback is not severe in HieFLANN as the local distribution analysis is performed in each HFS. The local distribution analysis of KFLANN subnet discovers the different classes distribution.

The conventional canonical analysis limits the reduction on feature dimensionality to *K-1*. In actual live data, this may not be the optimal cardinality for feature compression as the actual compression may vary from the expected feature. Such a limitation does not occur in the HieFLANN. This is because every combination from subnets contributes to a final stage compression that yields the dimension of the compressed feature.

If the difference between the means is small, the canonical analysis will encounter an undesirable situation. This issue will eclipse other computations of inter-correlations between criteria variables. In HieFLANN, each KFLANN subnet eliminates the eclipsing effect as each subnet is localized and isolated. Each KFLANN subnet contains most similar feature subset resulting acceptable range of difference in mean amongst the HFSs.

There are other advantageous of C2FeCom that contribute to its success within the HieFLANN model. The characteristics of C2FeCom benefits from both covariance and canonical analysis by becoming invariant to scaling and producing a singular transformation

In other words, linear combination of the basis vectors is mapped by the transformation into zero vectors. A vector that is mapped into the zero vectors through transformation is said to be in the *kernel* of that transformation. The kernel is a subspace of the domain in the transformation form. In additional to the properties delineated, there are no stringent implicit assumptions made in C2FeCom.

5. Experiment Result and Analysis

To verify the performance of the proposed HieFLANN and the effectiveness of the C2FeCom feature compression, several benchmark datasets were used in this paper. These were downloadable from UCI Repository of Machine Learning Databases [7]. The data was randomly divided into test sets and training sets. The datasets were the donated to the repository from real life datasets. These datasets are widely used by other researchers as bench marks.

In this paper, the Receiver Operating Characteristics (ROC) was adopted to organize the performance measurement of the HieFLANN classification model. The average errors on each class in the presented data set can be derived from ROC [4] using equation (8).

$$E = \frac{1}{K}\sum_{m=1}^{K} e_m \qquad (8)$$

$$e_m = \frac{\text{number of pattern misclassified in class m}}{n_m}$$

Where K is the number of classes, e_m is the classification error for class m and n_m is the number of pattern in class m.

There has been an increasing in the use of ROC due to its usefulness for domains with skewed class distribution (unbalanced classes) and unequal classification error costs. Detail information of ROC is described in [4]. Complementing the use of ROC measurement is the Area Under Curve (AUC):

$$AUC = 1 - E \qquad (9)$$

AUC can be viewed as the classification accuracy measurement. AUC is used in this paper as the classification performance is comparable to the classification accuracy adopted by most of researchers.

Apart from the ROC performance measurement, the compression ratio (CR) was also included to indicate the efficiency of the proposed feature compression method. Actual dimension (AD) is the dimensionality of raw feature space while reduced dimension (RD) is dimensionality of compressed feature space. Equation in (10) is the compression ratio used in the experiments represents the degree of compression made that is the actual degree of feature dimensionalities compressed. The higher the ratio, the better the compression but it still depends on the characteristic of problem domain. Not all problems can produce high CR due to the feature discriminating level. If all features are needed in order to provide higher class discrimination analysis, then CR will be low and little compression is possible.

$$CR = 1 - \frac{\text{RD}}{\text{AD}} \qquad (10)$$

Classification performance of HieFLANN on different data set is given in Table 2. The first column in Table 2 lists the type of real life domain extracted from UCI Repository.

The second column represents the actual dimension of problem set while third column is the reduced dimension using C2FeCom. The fourth column is the compression ratio computed using equation (10).

The compression ratios of greater than 0.4 were obtained for all problem sets except the Waveform data set. This is shown in Table 2. This discovery shows that the C2FeCom method is able to map the given problem set into a compact representation. The Iris, Wisconsin Breast Cancer and ionosphere problem sets achieved greater than 75% compression. This is followed by two columns list the AUC for training set and test set respectively.

Table 2. HieFLANN classification performance

Data set	AD	RD	CR	Train AUC	Test AUC	KFLANN Accuracy
Iris	4	1	0.7500	0.9361	0.9524	0.9500
Thyroid	5	3	0.4000	0.8781	0.9500	0.9067
DB	8	4	0.5000	0.7503	0.7213	0.7300
WBC	9	2	0.7778	0.8634	0.9723	0.9657
Wine	13	5	0.6154	1.0000	0.9667	0.9651
Heart	13	5	0.6154	0.8493	0.8753	0.7833
WF	21	13	0.3809	0.9079	0.8196	0.7328
Iono	33	7	0.7878	0.9564	0.8340	0.7712
DM	34	19	0.4412	0.8333	0.8540	0.8915

DB: diabetes, WBC: Wisconsin breast cancer, WF: waveform, Iono: ionosphere, DM: dermatology.

The AUC for training refers to the performance of the HieFLANN classifier when the training set was used, while AUC on test set represents its prediction ability. From the experimental testing, it is shown that problem domains either obtained better AUC value on training set than test set or comparable AUC value between training and test set.

The Iris dataset achieved an AUC value (in percentage) of 93.61% on training set and 95.24% on the test set. WBC data set however had the best predictor amongst all others. The training AUC was 86.34% and test set AUC was 97.23%. The results indicate that HieFLANN is an acceptable architecture that is able to split the input space into smaller independent modules and later combine the results to yield acceptably accurate results.

Last column indicates the KFLANN classification accuracy on test set using actual dimensionality feature space. Comparison between the accuracy obtained on KFLANN and HieFLANN (AUC on test set) revealed that classification on compressed feature space significantly better if not comparable to the classification on actual dimensionality feature space. Most of the tested problem domains achieved better classification accuracy on compressed feature space than actual dimensionality feature space except for Diabetes and Dermatology data set. Classification accuracy for Diabetes data set in compressed feature space is 72.13%, only slightly differ from the accuracy on the actual dimensionality feature space that is 73%. Dermatology data set displayed the comparable classification accuracy on both the actual dimensionality feature space (85.4%) and compressed feature space (89.15%).

Experimental results demonstrate the merits of the proposed HieFLANN for pattern learning on large dimensionality problem domains. With the overall compression degree of minimum 40% on the tested problem domain verify the feasibility of the compression method proposed. Classification using the compressed feature space reduces the computational workload, minimize memory usage and shorten the computational time.

Table 3 summarizes the class distribution of the benchmark datasets used in experiments. They are listed in increasing order of feature dimensionality. All datasets contain continuous feature values. The dimensionality ranges from 4 to 34. All feature values were standardized with maximum normalization. The normalized values range between –1 and 1. Missing values in problem domains were excluded from the experiment. It is important to note that Class labels were not provided during training process and used only for evaluation on the experimental testing.

Table 3. Datasets description for the benchmarks datasets used. Numbers enclosed in parentheses represent the number of pattern samples for that particular class

Data set	Description
Iris	3 classes of Iris flowers, Class 1-Setosa (50), Class 2-Virginica (50) and Class 3-Versicolor (50). The length and width of flower's petal and sepal determine the flower type.
Thyroid	2 classes of thyroid disease, Class 1- (49), Class 2- (91) comprise of laboratory test indicators as the attributes information.
DB	5 classes of Eryhemato-Squamous Disease, Class 1- (112), Class 2- (61), Class 3- (72), Class 4- (49), Class5- (20). Individual health factors determine the diseases type.
WBC	2 classes indicate benign (241) and malignant (458) breast cancer cell. Attributes are the properties of cells measured.
Wine	3 types of wine based on the chemical contain. Class 1- (59), class 2- (71), class 3- (48).
Heart	2 classes indicate heart and non-heart disease based on the health factors, class 1- (1399), class 2- (164).
WF	3 classes of waveform with attributes are generated from a combination of 2 of 3 "base" waves. Class 1- (1657), class 2-(1647), class 3- (1696).
Iono	2 classes of radar reflection from ionosphere to detect free electrons in the ionosphere. Class 1- (225), class 2- (126).
DM	6 classes of skin disease. Class psoriasis (112), class seboreic dermatitis (61), class lichen planus (72), class pityriasis rosea (49), class cronic dermatitis (52), class pityriasis rubra pilaris (20). The clinical attributes and histopathological attributes determine type of disease.

6. Conclusion

This paper presents the Hierarchical Fast Learning Artificial Neural Network (HieFLANN) as a new model for learning high dimensionality data. It is a promising hierarchical ANN pattern learning model that uses the canonical covariance based feature compression as a means to process data. The proposed model was tested extensively on several benchmark datasets and it yielded promising results. A further note to the HieFLANN model is that it can be executed in a totally unsupervised mode.

HieFLANN addresses large problem sets by partitioning it into smaller manageable sets for independent processing subnets. The inherent attribute relationships can be coded between the subnets and the level of significance the input sub-space plays in determining the global accuracy. It initially uses similarity theory to generate the HFSs and subsequently using independent KFLANN clustering processing on each HFS. The computation of canonical variates thru eigen-decomposition on the canonical correlation matrix (CCM) is carried out prior to C2FeCom. The principal components obtained from the eigen-decomposition on CCM together with covariance components of the underlying unknown distribution leads to a convenient representation for generalization. The KFLANN was used to cluster the compressed feature space in the final stage.

The results indicate that the C2FeCom feature compression method has been highly reliable in all the experiments performed. Experimental obtained results showed that the classification performance on test data is generally similar to the training set. It was also discovered that the majority of the collected benchmark datasets seem to contain redundant, irrelevant features that do not significantly contribution to the successful classification of the data set. It is therefore possible to obtain a compressed representation of original feature space with minimal loss of discriminative power.

The HieFLANN is advantageous in that it

- provides excellent generalization power as showed in experiments results.
- is a facile self-organized neural model that is easy to duplicate.
- Can be fully unsupervised as it does not require any parameter settings.

The proposed HieFLANN is a totally unsupervised driven classification neural model provides a novel approaches. It process the feature transformation and classification within a single ANN framework.

Motivated by the self-organized learning way adopted in biological nervous system, the HieFLANN actualizes the unsupervised self-learning concept through its self-organized structure. No intervention is required to set up the parameters. Each KFLANN subnet determines the overall stability of HieFLANN model. The HieFLANN also takes into consideration the plasticity dilemma issue, thus it does not suffer from the severe effect of the interference when learning new knowledge.

A further study will be done to determine how inherent inter-correlation between subnets can be encoded for rule interpretation. This may be useful for extracting rules needed for deterministic software environments where useful predictive rules can be incorporated into decision systems.

A formalization of the HieFLANN is also needed to determine how the C2FeCom actually assists in the compressed feature space encoding, providing a mathematical model for the network.

Since the HieFLANN is able to provide an effective clustering platform that partitions the global problem into packet sized local problems, the reduced input feature space can be used to train the conventional multilayered perceptron (MLP) models. Instead of remaining as a HieFLANN, the internal networks can theoretically be replaced by the conventional MLP neural networks. This may proof useful as the pre-partitioning of high dimensional problems to create a hierarchical structure of minute MLP networks.

References

1. Alex T. L. P., Prakash S., K-Means Fast Learning Artificial Neural Network, An Alternate Network For Classification, ICONIP, (2002).
2. Dittenbach M., The Growing Hierarchical SOM (GHSOM), IJCNN, (2000).
3. Evans D.J., Tay L. P., Fast Learning Artificial Neural Network for continuous Input Applications, Kybernetes. 24 (3), (1995).
4. Fawcett W., ROC Graphs: Notes and Practical Considerations for Researchers, HP Labs Tech Report HPL-2003-4 (2004).
5. Fukushima K., Neocognition: A Hierarchical neural network capable of visual pattern recognition, Neural Network, vol. 1, pp. 119-130, (1988).
6. Grossberg S., How does the Cerebral Cortex Work, development, learning, attention and 3D vision by Laminar circuits of visual cortex, Technical Report CAS/ CNS TR-2003-005, (2003).
7. Merz, C.J., Murphy, P.M. UCI Repository of machine learning databases, Irvine, CA: University of California, Department of Information and Computer Science, (1998).
8. Tay L. P., Evans D. J., Fast Learning Artificial Neural Network Using Nearest Neighbors Recall, Neural Parallel and Scientific Computations, 2(1), (1994).
9. Wong L. P., Alex T. L. P., Centroid Stability with K-Mean Fast Learning Artificial Neural Networks, IJCNN, (2003).
10. Wong L. P., Xu J., Alex T. L. P., Liquid Drop Photonic signal using Fast Learning Artificial Neural Network, ICICS, (2003).