CHAPTER VI

BAYESIAN AND INCREMENTAL LEARNING FRAMEWORKS

In statistical regression estimation, there are two parameters that contribute to the generalization. Bias is the first parameter, which is characterized as a measure of a predictor's ability to generalize correctly to a test set once trained. The second parameter is variance, which can be characterized as a measure of the extent to which the same results would have been obtained if a different set of training data were used.

This chapter presents a new neural network architecture that can improve the generalization of the nonparametric regression (and also classification) error by exploiting the concept of multiresolution analysis. The new network architecture can also be called as *multiresolution committee of networks*, where it is very natural for image analysis. After a comprehensive introduction on Bayesian averaging framework, we point out several similarities between wavelet representation and Bayesian averaging framework. This results in the architecture that exploits the intuitive idea why the feature set of the multiresolution committee of networks can be split into several time-frequency populations (subsets), where they are considered to be associated with the combinations of the smoothness and edge characteristics in case of images.

Finally, the essential idea for building a collection of transform networks is by exploiting the interpolation power of the neural networks. This chapter provides a proof that the linear combination of individual network weights of a collection of transform networks is a more generalized representation for multiple classifier systems than other simple methods, e.g., constant or weighted sample mean of the weights. Moreover, the analogy between incremental learning and multiresolution learning method is used for our thoroughly discussion of the new neural network architecture.

6.1 Introduction

It is known that neural networks are powerful tools for handling problems of large dimension. Additionally, There are many studies reported the ability of neural networks to approximate nonlinear functions [99, 100]. Particularly, neural networks are widely used as a universal approximator for classification, prediction, and regression. One of the interesting properties of neural network is its low bias on approximation. The main reason underlying the low bias approximation of neural network approximation is that they are constructing predictors from a very large class of functions. These classes of functions are complete in

the sense that every sufficiently smooth function of the inputs can be well approximated by one of the functions in the class. For instance, every sufficiently smooth function can be well approximated by a single hidden layer feed-forward network. However, the price for achieving a small bias is large variance. Since neural networks are low biased classifiers, the key to increasing accuracy is in reducing the variance while keeping the bias low.

As also suggested by Haykin [29], if we are willing to purposely increase bias very little, it is possible to eliminate variance or reduce it significantly, which then improves the generalization. One of the methods to embedded bias in the network is by building network with some *prior knowledge*. For example, bias is designed and embedded in the network architecture in the design of constrained network architecture using weight sharing and local receptive fields. In other words, bias may take the form of prior knowledge built into the network design. The prior knowledge built into the neural network design using local receptive fields is that an image is two-dimensional and has a strong local structure. Moreover, these properties are also preserved in the wavelet transform of an image.

Finally, wavelets are also known to be better function estimators than classical nonparametric estimators as a result of their better local accuracy and faster convergence [101]. As mentioned in [102], at the coarser resolution the bias of the wavelet estimator increases, while the variance decreases. If multiple low biased estimators based on wavelet representation are used for the estimation, the only component left for us to taking care of is the remained variance, which can be reduced by combining of the estimators.

6.2 Local Discriminant Basis Neural Network Ensembles

In this chapter, the possibility of using an orthonormal basis to train a collection of artificial neural networks (ANNS) in a face recognition task is discussed. This orthonormal basis is selected from a dictionary of orthonormal bases consisting of wavelet packets. Our proposed method takes advantage of the fact that the dimensionality of the pattern recognition problem at hand is reduced, but the important information is still contained, and at the same time, some correlations between neighboring inputs are included. Furthermore, the performance of our proposed network scheme is improved over a single neural network as a result of multiple classifier systems.

As discussed before, there are a number of different ways of creating ensembles of neural networks: varying the set of initial random weights, varying the topology, or varying the data, etc. As mentioned in [23], varying the data is more effective than varying the set of initial conditions. Recently, the classification accuracy of test data is improved if a collection of networks is trained by a preprocessing set of data based on a full decomposition tree using multiresolution analysis technique. Such method is called *wavelet packet consensual neural network* (WPCNN) [87]. However, WPCNN is not efficient in terms of computation since the new versions of the preprocessed data used to trained an ensemble of networks are still the same dimension as the original data. Compared to single neural networks, WPCNN are

more computationally expensive since more computation is needed not only from the image decomposition using full wavelet packet decomposition tree, but also the longer training time of multiple neural networks.

The LDBNNE scheme proposed in this dissertation is an alternative network architecture to avoid the shortcomings of conventional back-propagation network and WPCNN. In the LDBNNE, a collection of slightly weaker learners is constructed using a set of neural networks, and a set of subbands selected by local discriminant basis (LDB) algorithm. Each neural network is responsible to learn on the same set of training data, but under different subband images. After all networks have been trained, their outputs are combined by simple averaging or weighted averaging. A major dimensionality reduction using LDB enables each network to reduce the training time, and at the same time, avoid the network being overtrained. Furthermore, our proposed method is expected to perform better than a single network because if each subband contains information with enough discriminant power, each network should find an acceptable decision boundary of the pattern recognition problem at hand. Finally, by simple averaging or weighted averaging of all decision boundaries of the set of neural networks, the classification error of the test data is expected to be reduced by the result of averaging.

Figure 6.1 displays the combining strategy between LDB and neural network ensembles. As illustrated, N networks are trained by N training sets of subband images. For each network, its training sets x_j are created from the projection of the original image onto one of the most discriminant basis vector chosen by the local discriminant bases algorithm presented in Chapter 3. The final decision is made from the plurality of all networks' decision.

We expect that a better generalization should occur, if we incorporate wavelets with neural networks in the nonparametric estimation task, such as in classification. We also expect that the predictor's variance should be reduced significantly while the bias term remains small. In particular, when we replicate training sets from the original images and use them to construct N predictors, it is possible that a new predictor constructed from the plurality vote of N predictors' outputs should produce variance that is going to be closed to zero. From this point forward, x_j will be called the *most discriminant subband* image. In the next section, we analyze LDBNNE that can be viewed as multiresolution committee of networks, especially in the context of Bayesian model averaging framework. As a result, the class of a collection of transform networks can be

6.3 Multiresolution Committee of Networks : Bayesian Model Averaging Framework

Here, we would like to emphasize that Local Discriminant Bases Neural Network Ensembles (LDBNNE) can also be interpreted as a committee of networks, naturally arisen from the framework of traditional Bayesian averaging [33, 103]. Assume that a statistical model allows the inference about the variable y in the form of the predictive probability

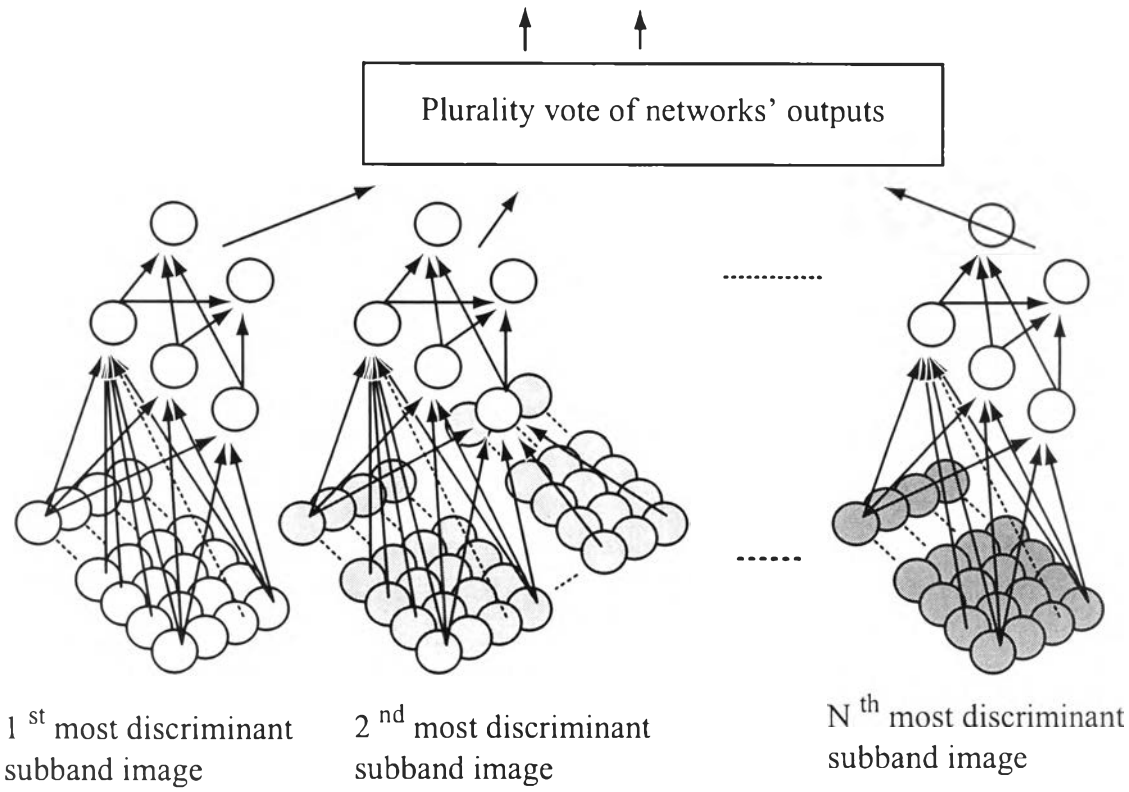

Figure 6.1: A collection of neural networks trained by N most discriminant subband images.

density $P(y/\mathbf{w})$, where $\mathbf{w}$ is a vector of model parameters. Furthermore, let's assume that we have a data set D which contains information about the parameter vector $\mathbf{w}$ in form of the probability density $P(\mathbf{w}/D)$. We then obtain

$$P(y/D) = \int P(y/\mathbf{w})P(\mathbf{w}/D)d\mathbf{w}. \tag{6.1}$$

As known before, there are several training procedures [23] that can be used for constructing an ensemble of classifiers. For example, training each ensemble classifier starting from different random initial weight configurations. This way, we will typically discover several different single, non-equivalent minima. Another approach is called *bagging*, which let each ensemble classifier be trained with a different data set in order to get the uncorrelated predictions between individual classifiers. Mixture of experts is one of the approach to ensemble classifiers with motivation that we design a system in which different classifiers are responsible for modeling different regions in input space. LDBNNE, random weights initialization, and bagging training procedures allow us to approximate the posterior distribution of the weights using a set of Gaussians, one centered on each local minimum, in which we assume that there is negligible overlap between Gaussians [33]. Hence, we can represent the posterior distribution of the weights as

$$P(\mathbf{w}/D) = \sum_i P(m_i, \mathbf{w}/D) \tag{6.2}$$

$$= \sum_i P(\mathbf{w}/m_i, D)P(m_i/D), \tag{6.3}$$

where m_i denotes one of the non-equivalent minima and all of its symmetric equivalents. From (6.1) and (6.3), we then get the posterior distribution of the outputs by integration over the weight space :

$$P(y/D) = \int P(y/\mathbf{w})P(\mathbf{w}/D)dw \tag{6.4}$$

$$= \sum_i P(m_i/D) \int_{\Gamma_i} P(y/\mathbf{w})P(\mathbf{w}/m_i, D)dw \tag{6.5}$$

$$= \sum_i P(m_i/D)P(y/m_i, D), \tag{6.6}$$

where Γ_i denoted the region of weight space surrounding the i^{th} local minimum. From (6.6), we see that the posterior distribution of the output is just a linear combination of the posterior distribution of the outputs made by each of the networks corresponding to distinct local minima, weighted by the posterior probability of that solution.

The above result can be further extended to a combination of different model H_j. Note that, here, different models mean networks with different simplicities, i.e., numbers of hidden nodes, number of hidden weights, or

$$P(y/D) = \sum_j P(H_j/D)P(y/H_j, D). \tag{6.7}$$

From (6.6) and (6.7), we can easily obtain other quantities such as the mean output predicted by the committee with respect to distinct local minima or models, which can be given by

$$\bar{y} = \sum_i P(m_i/D)\bar{y}_i, \tag{6.8}$$

and

$$\bar{y} = \sum_j P(H_j/D)\bar{y}_j, \tag{6.9}$$

where y_i and y_j are the corresponding network prediction averaged over the i^{th} local minimum and j^{th} models, respectively. Intuitively, a more accurate model for mean output can be provided if we combine (6.8) and (6.9), which can be given by

$$\bar{y} = \sum_j \sum_i P(H_j/D)P(m_i/D)\bar{y}_{ij}. \tag{6.10}$$

In other words, a more accurate model should be composed by a set of different submodels, where each submodel can further be composed as either a single Gaussian or a Gaussian mixture distribution. We known that classification is closely related to regression. As previously discussed, the generalization error is related to the *bias/variance* dilemma [104], where ensemble bias is the degree to which the averaged output of the committee of networks diverges from the true target function, and variance is the degree to which the committee members disagree. Generally, a low error requires both a low bias and variance. One of the potential approach is to combine multiple low biased classifiers in a way that the remained variance is reduced, leading to better generalization improvement. In other words, the candidate members of a committee of networks should be learners with rather strong discriminant power – i.e., a collection of classifiers composed of the learners with high classification accuracy will produce better results than random classification accuracy.

At this point, we have shown that the preference criteria for selecting candidate members of an ensemble classifier are that each member should have low bias, different simplicities, and a mixture Gaussian distribution. Coincidentally, after inspecting the properties and applications of the time–frequency representation (wavelets), we found that those criteria are usually met. First, the coarsest resolution subband of data set generated from the discriminant time–frequency transform are consistent with the discriminatory features in a classification problem, so the predicted classification functions trained by the coarsest subbands trend to have low bias.

Second and more importantly, the obtained resolution-specific subband of data exploit the intuitive idea that the feature subsets of network ensembles should be split into several population, where they are considered to be associated with the combinations of the smoothness and edge characteristics in case of images. In fact, the cares resolution subband of data is the feature subsets that usually represents information regarding on texture, where detail resolution-specific subbands play some important roles on capturing edges and other information regarding of the high discriminative frequency components. This way,

different feature subsets trend to be independent because they are capturing different feature characteristics. This way, the remained variance can be easily reduced by simple averaging such as in the Bayesian averaging framework.

Since different sizes of subbands can be generated by wavelet (packet) decomposition, so different network simplicities (hypotheses) are generated through the use of local discriminant bases algorithm. Evidently, each subband of data is derived from a specific basis in a way that is very similar to a Gaussian mixture. It is also easy to see that the analogy between wavelet reconstruction and Bayesian averaging can be used as a thorough explanation for the need of the new neural network architecture. In other words, wavelet representation preserves the properties of Gaussian mixture. Furthermore, the exploitation of wavelet transform for feature extraction is preferred because dimensionality reduction is achieved due to the subsampling functions implemented in the transform. Finally, it should be noted that there are several researches focused on the unification of wavelet transform, radial basis function and Gaussian mixture.

Next, one of the explanations on how to construct a multiresolution committee of networks is explained. In fact, we can use the following proof to describe the mechnism of the Bayesian averaging minima in (6.3).

6.4 A Collection of Transform Networks

Let f_k denote the activation function of layer k and $y_k(x)$ the vector of outputs from the node of layer k given x as input to the whole network. Let $W_{k,l}$ denote the matrix of weights of the connections leading from layer k to layer l (thus $l > k$). $W_{k,l}$ will be the zero matrix for disconnected layers. For the sake of simplicity, the notions k and l are used for this chapter only. The reader should not confuse these notations with those presented in the previous sections.

Definition 6.1 *A Transform Network (TN) is a pair $(\mathcal{G}, \mathcal{N})$, where $\mathcal{N}$ is a feedforward network satisfying the above conditions, and $g \in \mathcal{G}$ has its corresponding reversible inverse transform g^{-1}, which $A(g)$ and $A(g^{-1})$ are given as the matrix representation associated with g and g^{-1}, respectively.*

Lemma 6.1 *The output of a $TN(\mathcal{G}, \mathcal{N})$ is invariant under the linear transform of $\mathcal{G}$ on the input layer of $\mathcal{N}$.*

Proof We start by indexing the layers of $\mathcal{N}$ by $1, 2, \ldots, L$ on the understanding that layer 1 is the input layer and the output layers are those with the highest indices.

Let consider two (most discriminant subband) network inputs x_1 and x_j which are related by two linear transforms $g_1, g_j \in \mathcal{G}$. Suppose that A_j is the matrix representation associated with the j^{th} linear transform applied to original input x. Let layer 1 be the input layer, and layer L be the final output layer, then $P(j)$ be the statement

$$y_L(x_1) = y_L(x_j). \tag{6.11}$$

We now prove $P(j)$ for $j \in 2 \ldots J$. Without loss of generality, we assume that $W^1_{k,l} = W^j_{k,l}$, where $W^j_{k,l}$ denotes matrix of trained weights associated with the j^{th} linear transform, and $l > k$. Let assume $P(2), \ldots, P(j)$ are true, we then prove for the $(j+1)^{th}$ linear transform $\in \mathcal{G}$ from

$$y_1(x_1) = f_1\left(W^1_{1,2}\, x_1\right). \tag{6.12}$$

After applying the multiresolution weight transfer [105], we obtain

$$y_1(x_1) = f_1\left(A_1(g^{-1})W^1_{1,2}\, x\right). \tag{6.13}$$

Similarly, we can obtain

$$y_1(x_{j+1}) = f_1\left(A_{j+1}(g^{-1})W^{j+1}_{1,2}\, x\right). \tag{6.14}$$

Hence $y_1(x_1) = y_1(x_{j+1})$ is true, if the $(j+1)^{th}$ network can be trained such that

$$A_{j+1}(g^{-1})W^{j+1}_{1,2} = A_1(g^{-1})W^1_{1,2}. \tag{6.15}$$

If we multiply both sides by $A_{j+1}(g)$, we then obtain

$$A_{j+1}(g)A_{j+1}(g^{-1})W^{j+1}_{1,2} = A_{j+1}(g)A_1(g^{-1})W^1_{1,2}. \tag{6.16}$$

In the case of multiresolution analysis, it is easily to verify that $A_{j+1}(g)A_{j+1}(g^{-1}) = I$, where I is an identity matrix. In this case, we will obtain

$$W^{j+1}_{1,2} = A_{j+1}(g)A_1(g^{-1})W^1_{1,2}. \tag{6.17}$$

Furthermore, this result can be easily generalized to $l > 1$

$$W^{j+1}_{1,l} = A_{j+1}(g)A_1(g^{-1})W^1_{1,l}. \tag{6.18}$$

Since $y_1(x_1) = y_1(x_{j+1})$ when we impose the following constraint

$$\begin{array}{rcll} W^{j+1}_{k,l} & = & A_{j+1}(g)A_1(g^{-1})W^1_{k,l} & for\ \ k = 1\ \ and\ \ l > 1 \\ & = & W^1_{k,l} & for\ \ k, l > 1. \end{array} \tag{6.19}$$

Thus, we can generalize the above statement for $j \in 1, \ldots, K-1$ and all the output layers as

$$\begin{array}{rcl} y_L(x_1) & = & f_L\left(\sum_{k=0}^{L-1} W^1_{k,l}\, y_k(x_1)\right) \\ & = & f_L\left(\sum_{k=0}^{L-1} W^{j+1}_{k,l}\, y_k(x_{j+1})\right) \\ & = & y_L(x_{j+1}). \end{array} \tag{6.20}$$

This concludes the proof that the output of a $TN(\mathcal{G}, \mathcal{N})$ is invariant under the linear transform of $\mathcal{G}$ on the input layer of $\mathcal{N}$. In other words, concepts (hypotheses) of two networks are equivalent if the inputs of these two networks are both underlying the reversible linear transform condition and the necessary condition defined in (6.20). □

The reason underlying the weight transfer is that we can formulize this idea through the multiresolution analysis framework – i.e., If the two linear transform $g_1, g_j \in \mathcal{G}$ are related to each other through the multiresolution analysis framework, then we can construct a collection of the networks in which the overall performance can be benefited from the interpolation power of the training neural networks. In LDBNNE system, we usually address g_1 as the most discriminant transform basis, and $g_j, j = 1, \ldots, K$ are the subsequent power discriminant transform basis.

In practice, each $W_{k,l}^{j+1}$ is perturbed by small error and very closed to $A_{j+1}(g)A_1(g^{-1})W_{k,l}^1$ in different ways due to the random weight initialization and its learning local minima. Thus, the above Lemma can be easily further extended to a combination of different hypotheses models H_j. By using a 2-tuple set $\{(H_1, W_{k,l}^1), ..., (H_j, W_{k,l}^j), ..., (H_K, W_{k,l}^K)\}$ to approximating the posterior distribution of the outputs, a new input classification will be a weighted average of K network classification outputs as shown in (6.10), where $\{W_{k,l}^1, ..., W_{k,l}^K\}$ is a set of transferred weights of K networks generated by the LDBNNE. This way, the improvement of the generalization error is then followed nicely from the bias/variance [104] or bias/spread dilemma.

Furthermore, it is easy to show that LDBNNE is a generalized case of committee of networks than other simple methods, such as constant or weighted sample mean of the weights method [108].

Corollary 6.1 *Suppose that the hypothesis space of each network trained by* $\mathbf{x}_i$ *is large enough. Then either constant or weighted sample mean of the weights is one of the special case of LDBNNE.*

Proof Here, we impost a stronger constraint for a trivial case, which is that each input to the nodes of the first hidden layer of the base network are converges to the neighborhood of a local minimum. If the hypothesis space of each network trained by $\mathbf{x}_j$ is large enough. In other words, we can find $W_{k,l}^j$ from (6.20) that gives $\left|W_{k,l}^* - W_{k,l}^j\right| \leq \epsilon_i$, where $W_{k,l}^*$ is either constant or weighted sample mean of the weights, and ϵ_i is a small positive value. □

6.5 Relation to Incremental Learning Framework

It is well known that the generalization error of classifier is a result of appropriate network design; a rather small network size can make the network learn incomplete solutions, while an unnecessarily large size may lead the network learn only the specific training samples and noise. When the network is called *overtrained* if the size of the network is too large to classify correctly input data which were not included in its training set.

A typical approaches for good network design involves building prior knowledge into the network structure [29]. However, there are no well defined rules yet for building prior information into neural network design. Conventionally, there are some *ad-hoc* procedures that are known to yield useful results, especially when applied directly to image pixel values.

As proposed by the *ad-hoc* procedures so far, the network can be designed to include prior information by using a combination of two techniques: (1) constraining interconnection weights by the use of weight sharing and (2) restricting the network architecture through the use of local connections. On the contrary, another recent study [105] illustrates that a good network design can be obtained by incremental network growing using relative prior knowledge transferred from lower approximation subbands. The method is called *multiresolution learning* method, which is aimed for designing network in a more systematic way. After conducting literature survey, we found that multiresolution learning method is not only a good network design procedure but its interpretation is raised naturally in the incremental learning framework.

The ideas of incremental learning is developed for learning the new information without forgetting previously acquired knowledge. As pointed out by Grossberg [106], the *stability/plasticity* dilemma describes the fundamental problems in learning new information by stating that some information may have to be lost to learn new information, as learning new pattern will tend to overwrite formerly acquired knowledge.

Incremental learning has been referred to as diverse concepts as incremental network growing and pruning, on-line learning, or relearning of formerly misclassified instances. Furthermore, various other terms. such as constructive learning, lifelong learning, and evolutionary learning have also been used to imply learning new information. In general, good incremental learning algorithm, generates new decision clusters in response to new patterns that are sufficiently different from previously seen instances. Instead of generating new cluster nodes for each previously unseen (or sufficiently different) instance, the algorithm in [107] generate multiple new "weak classifiers" for previously unseen portions of the feature space. By combining the outputs of weak classifiers, the goal of new information learning or incremental learning is achieved.

In incremental learning, new information can be incrementally updated at different levels. There are two incremental learning approaches that are investigated here. For the first approach, new information is updated at the hidden layers, while new information is updated at the output levels in the second approach. The process of incremental learning at hidden layer level is by growing the number of hidden nodes for each new information. We aware that the original multiresolution learning method [105] is an approach where the incremental learning is corresponding to the first approach of the incremental learning. However, the new information incrementally learned by multiresolution learning method are different from the original definition of the incremental learning. In conventional definition, new information is come from previously unseen (or sufficiently different) instance, while in multiresolution learning method, the new information is come from the finer subband input data, which has larger dimension than the transformed input data.

The essential idea underlying the original multiresolution learning algorithm concept is based on the multiresolution weight transfer method [105]. To implement weight transfer, a constraint is imposed such that the inputs to the nodes of the first hidden layer of the

network trained by $\mathbf{x}_{l-1}$ be identical to the corresponding inputs of the nodes of the network trained by $\mathbf{x}_l$, where $\mathbf{x}_{l-1}$ is the detail subband of $\mathbf{x}_l$ at the l^{th} finer resolution level. On the implementation, we first train neural network with the lower approximation subband and then transfer these weights to the current resolution subband network. Next, we use the current resolution input data to train the current resolution network. This process can be implemented until reaching the original resolution or terminated at the desired resolution level. The reason for training in this way is that the network trained by the information from lower resolution level (LL, LH, HL, HH; in case of image) is not always meaningful; this is due to the fact that in most cases only some of the four subband images contain significant portion of the content of the original image.

As addressed before, the Learn++ algorithm [107] generates multiple new "weak classifiers" for previously unseen portions of the feature space. In other words, this algorithm is the second type of the incremental learning algorithms that updated the new information at the output levels; it is in this context that we generalize the concept of the second type of the incremental learning approach to the original multiresolution incremental learning method. Coincidentally, our LDBNNE can be fitted into the second type of the incremental learning algorithm, in which additional (new) information is updated at the output level. In LDBNNE system based on the best adapted bases algorithm, input data is composed of highly discriminative input data (usually the lower approximation subband data) and their subsequent discriminant detail input data (detail subband data).

As proven in [101],when signal is broken up into several frequency bands called subbands, the subband sample is uncorrelated at scales differing by more than 1, and has arbitrarily small correlation at scales differing by 1. Benefiting from such work, the collection of transform networks obtained from LDBNNE can be efficiently combined by the majority averaging method, if most of the selected subbands are at scales differing from one another more than 1. This is from the fact that different feature subsets for the collection of transform networks are trend to be independent. Additionally, we can generate a network to approximate an incremental network growing by training additional detail subband network, separately (or in parallel) from the coarsest resolution network.

Intuitively, learning in the original multiresolution learning method is incrementally updated after the weights of the coarser subband network are transferred to the next finer resolution network, and the new hidden nodes are added to this finer resolution network. In fact, this incremental learning procedure is performed every time the hidden layer nodes are growing. On the contrary, incremental learning in LDBNNE is performed every time the new classification network is ensembled to the collection of *transform networks*.

In summary, our new multiresolution learning method is suitably fitted to the incremental learning setting, inspiring from the incremental learning method [107] that have demonstrated major performance improvement. By incorporating the multiresolution learning concept with an idea of classifier ensembles, we can construct a new network architecture that performs rather well in practice. Due to the exploitation of the ensemble of classifiers,

the more the new transform networks are included, the better generalization error of the proposed method. This way, it is even trended to be close to the best generalization error for the problem in hand.

6.6 Experimental Results

We did a series of experiments on part of Yale face database to evaluate our proposed method. We modified the 13-class classification problem into a 4-class classification problem where the goal was to separate Subject A, B, C from the other 10 classes (in other words, we defined the remaining 10 classes as the Unknown Subject). Specifically, the original Yale face database of 143 faces consists of 13 subjects. There are 11 faces per subject, one for each of the following facial expressions or configurations: center-light, with glasses, happy, left-light, with no glasses, normal, right-light, sad, sleepy, surprised, and wink. In order to evaluate the generalization ability of our proposed classifier, a larger sample sets were created from the available faces. Specifically, four more faces per subject were created by zooming in and cropping on the original images. As a consequence, the total number of faces becomes $143 \times 4 = 572$. Examples of some faces from the Yale face database are shown in Figure 6.2.

For the simplicity of simulation, all faces were normalized and scaled down to the smaller size of 32x32. From our modified database, a set of 72 randomly picked faces was used as training samples, while another set of 500 faces was used to test the generalization ability of the networks. The Symmlet filters with $h(n)$ = [0, 0, -0.0707, 0.3536, 0.8485, 0.3536, -0.0707, 0, 0] and $g(n)$ = [0.0152, -0.3687, -0.0758, 0.8586, -0.3687, -0.0758, 0.0152] were used in the LDB algorithm. The first 10 most discriminant subbands (MDSB) are obtained for the set of training faces with 4 resolution levels. In fact, the preferred set of the most discriminant subbands is particularly obtained by the local discriminant bases algorithm proposed by Saito [55]. As shown in Figure 6.3, we have the MDSB indexed from 1 to 10, in which 1 indicates the first MDSB that has the highest discriminant cost. Then, we trained each network using 5 and 3 hidden units with one of the resolution-specific subbands selected from the 10 MDSB, respectively.

The performance of our proposed method is compared with two other methods; conventional backpropagation and WPCNN methods. Both methods were inputted with face at the original resolution. The reason for us to include WPCNN as the baseline method is that WPCNN also used the concept of ensemble of transform networks with wavelet packet transforms. In WPCNN, the number of networks for the ensemble was four, which corresponded to the number of levels in the full wavelet packet transform. We also set the number of resolution used in our proposed method equal to four.

Four sets of training samples created by full wavelet packet transform were used to train two versions of multiple classifier systems with one hidden layer in each neural network. The number of hidden nodes used in these two networks were 5 and 3 hidden

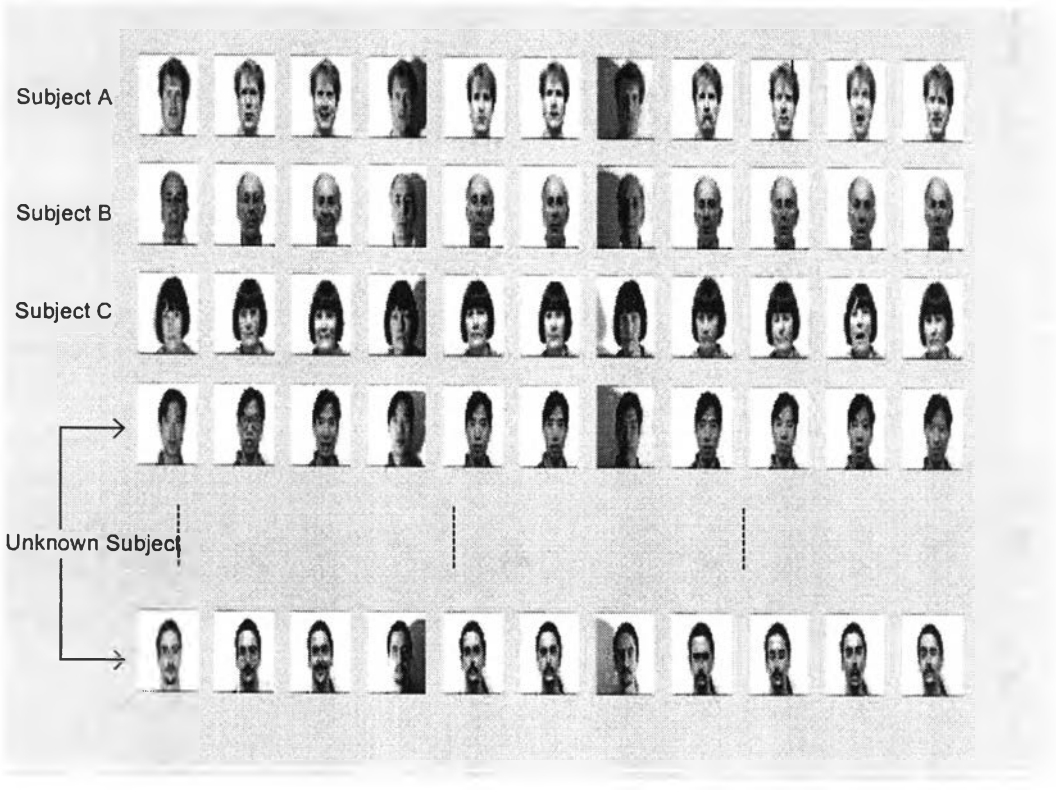

Figure 6.2 The Yale face database.

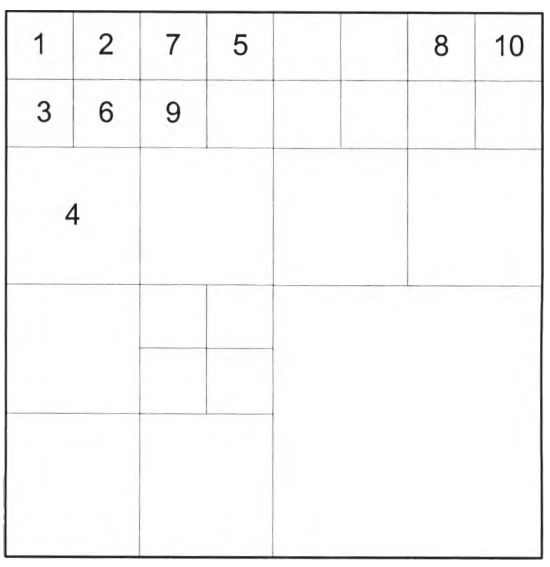

Figure 6.3 The first 10 most discriminant subbands.

units, respectively. For comparing the suboptimal and equal weighting methods of WPCNN with our proposed method, ten experiments for each method were performed. The average accuracy of WPCNN and our methods are shown in Figure 6.4.

Test accuracy, the number of iterations, and the computation time (also in ratio with respect to the proposed method and others) are reported in atu (addition time unit) in Table 6.1. Note that the computational complexity of all methods are calculated using the formula of atu given in Appendix B. From the experimental results, it can be seen that the LDBNNE methods outperformed both the WPCNN methods and conventional back-propagation network in terms of the classification accuracy of training and test data. For the issue of computational complexity, one of the arithmetic operations of WPCNN and LDBNNE based on the two-channel filter bank computation are comparable, but the computations in training transform networks are far more different. In particular, the computation of WPCNN is performed in two parts: the wavelet packet transform part, and the neural network ensembles part. The former computation is for computating the full tree wavelet packet transform of the 72 training images and 4 images created from the test image. The latter computation is the learning computation operations of the four networks.

In LDBNNE, the first computation is equal to the former computation in WPCNN. After MDSB of the training images is found, the test image is decomposed corresponding to the selected subbands. At this decomposition stage, the computation is usually negligible when it is compared to the decomposition of the training images. The latter computation of LDBNNE is the learning computation operations of smaller transform networks. From the experimental results, the computation complexity of the LDBNNE is less than all other methods in the order of magnitude. The reason regarding to the reduction in complexity is that the combination of all the numbers of connection weights of transform networks is far less than the number of connection weights of the network trained by high dimensional training set at the original size. Moreover, we can avoid the overtraining in LDBNNE, since the size of the each transform network is smaller than the network trained by training set at the original size. In comparison, the LDBNNE requires about 270.6 million operations (multiplications and additions). In contrast, the WPCNN and back-propagation networks require more than 16,087 and 3574 million operations, respectively. As a result, the proposed method outperformed both the WPCNN and conventional back-propagation methods in terms of the classification accuracy of test data, convergence speed, and computational complexity.

6.7 Conclusions

The LDBNNE architecture is based on a combination model of local discriminant basis algorithm and neural network ensembles. The LDNNE takes advantage of the fact that a collection of neural networks could be learned in parallel leading to a major reduction in learning times. Moreover, the learning times can also be improved because subband images obtained by this scheme can be done in an adaptive way (resulting from the best

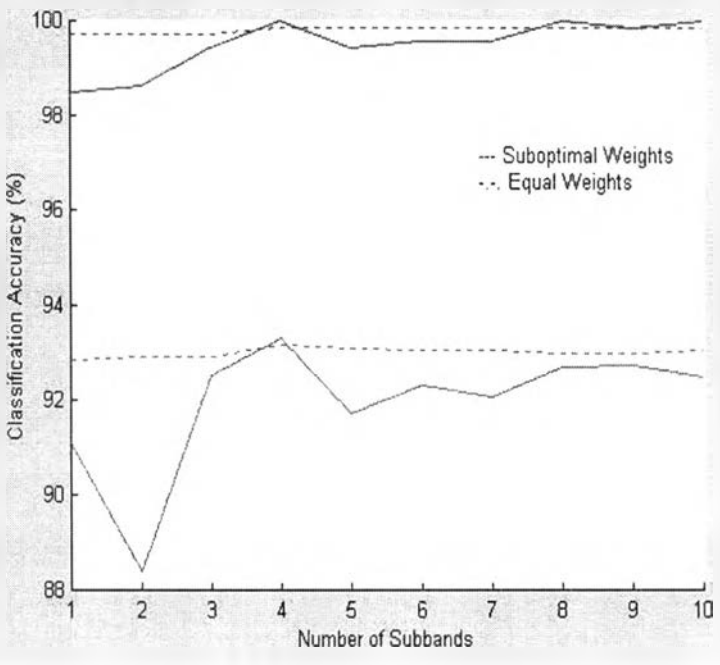

Figure 6.4: Average recognition accuracy of the LDBNNE. The upper curves represent training results and the lower curves test results.

Table 6.1: The performance of multiresolution committee of networks in overall percentage.

Method	Test Accuracy	Number of Iters.	Time ($x10^6$ atus)	Ratio
LDBNNE [21] 4 networks	**93.30**	800	270.60	**1.00**
LDBNNE 10 networks	92.43	800	542.58	2.01
WPCNN 10 networks	91.37	1800	16087.36	59.45
BP	90.97	1600	3574.31	13.21

discriminant bases selection algorithm) depending on the image classification characteristics, and at the same time, the dimensionalities of the subband images are often much less than the dimensionality of the original images. For the generalization ability of network ensembles, it can be improved because an ensemble of several good classifiers could be converted to be the best classifier.

LDBNNE is a method that not only reduce the generalization error but also:

- reduce overfitting in each classifier;
- reduce training times for the individual classifiers; and
- reduce the correlation among the classifiers.

More analytical study is discussed for our special type of multiple classifier systems, especially on how the above conditions can be achieved. In particular, the LDBNNE algorithm can be considered as a Bayesian committee networks. We also discuss on its relation to incremental learning, where additional (new) information is updated at the output level. For the completeness of the discussion, we finally provide a proof on a collection of transform networks that can be used as a supporting idea on how constant or weighted sample mean of the weights can be considered as one of the special cases of LDBNNE.

From the experiments, the results obtained showed that the LDBNNE outperformed wavelet packet parallel consensual and conventional neural networks in terms of both classification accuracy and computation complexity. Furthermore, it is not necessary that using as many subbands as possible can produce an optimal classifier. This is because networks in the ensemble may produce some classification outputs that are correlated to each other that would increase the generalization error. In order to remedy this problem, the algorithm suggested in Chapter 4 can be included to improve the overall generalization error.