APPLICATIONS OF MLP AND RBFN AS CLASSIFIERS
MODULE 12

Dr. Rajesh B. Ghongade
Professor, Vishwakarma Institute of Information Technology,
Pune-411048

Agenda

- Pattern Classifiers
- Classifier performance metrics
- Case Study : Two-Class QRS classification with MLP
- Data Pre-processing
- MLP Classifier MATLAB DEMO
- Case Study : Two-Class QRS classification with RBFN
- RBFN Classifier MATLAB DEMO

Pattern Classifiers

- Pattern recognition ultimately is used for classification of a pattern
- Identify the relevant features about the pattern from the original information and then use a feature extractor to measure them
- These measurements are then passed to a classifier which performs the actual classification, i.e., determines at which of the existing classes to classify the pattern
- Here we assume the existence of natural grouping, i.e. we have some a priori knowledge about the classes and the data
- For example we may know the exact or approximate number of the classes and the correct classification of some given patterns which are called the **training patterns**
- ***This type of information and the type of the features that may suggest which classifier to apply for a given application***

Parametric and Nonparametric Classifiers

- A classifier is called a ***parametric classifier*** if the discriminant functions have a well-defined mathematical functional form (Gaussian) that depends on a set of parameters (mean and variance)
- In ***nonparametric classifiers***, there is no assumed functional form for the discriminants. Nonparametric classification is solely driven by the data. These methods require a great deal of data for acceptable performance, but they are free from assumptions about the shape of discriminant functions (or data distributions) that may be erroneous.

Minimum-Distance Classifiers

- If the training patterns seem to form clusters we often use classifiers which use distance functions for classification.
- If each class is represented by a single prototype called the cluster center, we can use a minimum-distance classifier to classify a new pattern.
- A similar modified classifier is used if every class consists of several clusters.
- The nearest-neighbor classifier classifies a new pattern by measuring its distances from the training patterns and choosing the class to which the nearest neighbor belongs
- Sometimes the a priori information is the exact or approximate number of classes c.
- Each training pattern is in one of these classes but its specific classification is not known. In this case we use algorithms to determine the cluster (class) centers by minimizing some performance index and are found iteratively and then a new pattern is classified using a minimum-distance classifier.

Statistical Classifiers

- Many times the training patterns of various classes overlap for example when they are originated by some statistical distributions.
- In this case a statistical approach is appropriate, particularly when the various distribution functions of the classes are known
- A statistical classifier must also evaluate the risk associated with every classification which measures the probability of misclassification.
- The Bayes classifier based on Bayes formula from probability theory minimizes the total expected risk

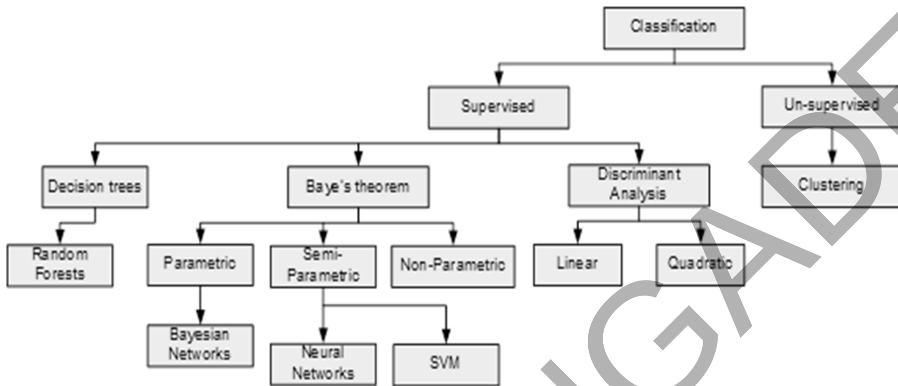
Fuzzy Classifiers

- Quite often classification is performed with some degree of uncertainty
- Either the classification outcome itself may be in doubt, or the classified pattern x may belong in some degree to more than one class.
- We thus naturally introduce fuzzy classification where a pattern is a member of every class with some grade of membership between 0 and 1
- For such a situation the crisp k-Means algorithm is generalized and replaced by the Fuzzy k- Means and after the cluster centers are determined, each incoming pattern is given a final set of grades of membership which determine the degrees of its classification in the various clusters.
- These are the non parametric classifiers.

Artificial Neural Networks

- The neural net approach assumes as other approaches before that a set of training patterns and their correct classifications is given
- The architecture of the net which includes input layer, output layer and hidden layers may be very complex
- It is characterized by a set of weights and activation function which determine how any information (input signal) is being transmitted to the output layer.
- The neural net is trained by training patterns and adjusts the weights until the correct classifications are obtained
- It is then used to classify arbitrary unknown patterns
- There are several popular neural net classifiers, like the **multilayered perceptron (MLP), radial basis function neural nets (RBFN), self-organizing feature maps (SOFM) and support vector machine (SVM)**
- These belong to the semi-parametric classifier type

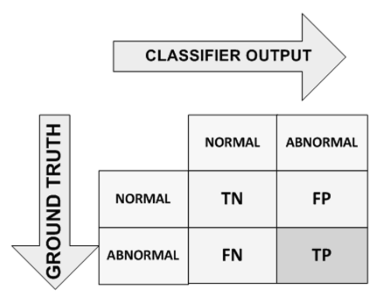
Pattern Classifiers



Classifier Performance Metrics

The Confusion Matrix

- The confusion matrix is a table where the true classification is compared with the output of the classifier
- Let us assume that the true classification is the row and the classifier output is the column
- The classification of each sample (specified by a column) is added to the row of the true classification
- A perfect classification provides a confusion matrix that has only the diagonal populated
- All the other entries are zero
- The classification error is the sum of off-diagonal entries divided by the total number of samples



TP: True positive
TN: True negative
FP: False positive
FN: False negative

Classifier Performance Metrics

1. **Sensitivity (Se) is the fraction of abnormal ECG beats that are correctly detected among all the abnormal ECG beats**

$$\text{Sensitivity (Se)} = \frac{TP}{TP + FN}$$

2. **Specificity (SPE) is the fraction of normal ECG beats that are correctly classified among all the normal ECG beats**

$$\text{Specificity (Spe)} = \frac{TN}{TN + FP}$$

3. **Positive predictivity is the fraction of real abnormal ECG beats in all detected beats**

$$\text{Positive Predictivity (PP)} = \frac{TP}{TP + FP}$$

4. **False Positive Rate is the fraction of all normal ECG beats that are not rejected**

$$\text{False Prediction Rate (FPR)} = \frac{FP}{TN + FP} = 1 - Spe$$

Classifier Performance Metrics continued…

5. **Classification rate (CR) is the fraction of all correctly classified ECG beats, regardless of normal or abnormal among all the ECG beats**

$$\text{Classification Rate (CR)} = \frac{TN + TP}{TN + FP + FN + TP}$$

6. **Mean squared error (MSE) is a measure used only as the stopping criterion while training the ANN**
7. **Percentage average accuracy is the total accuracy of the classifier**

$$\text{Percentage Average Accuracy} = \left(\frac{TP_{(NORMAL)}}{TOTAL_{(NORMAL)}} + \frac{TP_{(ABNORMAL)}}{TOTAL_{(ABNORMAL)}} \right) \times 100$$

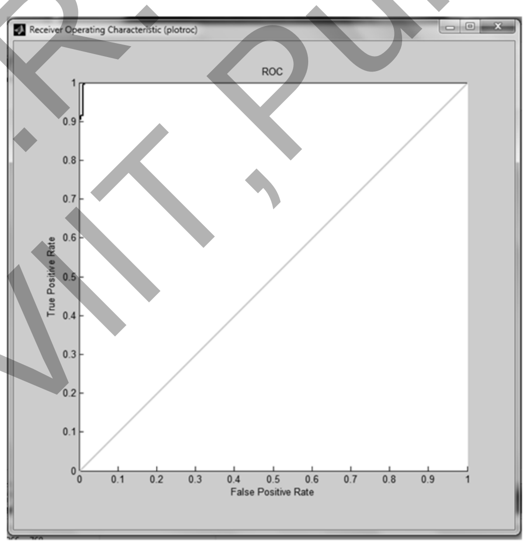
8. **Training Time is the CPU time required for training an ANN described in terms of time per epoch per total exemplars in seconds**
9. **Pre-processing time is the CPU time required for generating the transform part of the feature vector in seconds**

10. **Resources consumed for the ANN topology is the sum of weights and biases for the first layer and the second layer also called as adjustable or free parameters of the network**

Receiver Operating Characteristics (ROC)

- The receiver operating characteristic is a metric used to check the quality of classifiers
- For each class of a classifier, roc applies threshold values across the interval [0,1] to outputs.
- For each threshold, two values are calculated, the True Positive Ratio (the number of outputs greater or equal to the threshold, divided by the number of one targets), and the False Positive Ratio (the number of outputs less than the threshold, divided by the number of zero targets)
- ROC gives us the insight of the classifier performance especially in the high sensitivity and high selectivity region

A typical ROC

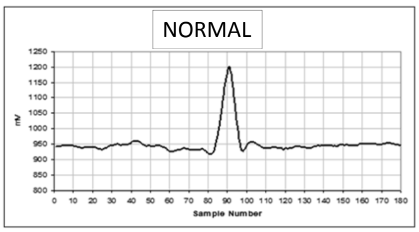


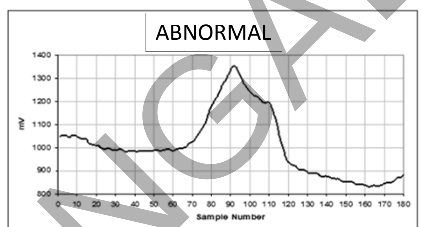
- **For an ideal classifier the area under curve of ROC =1**

Case Study : Two-Class QRS classification with MLP

Problem Statement:

Design a system to correctly classify extracted QRS complexes into TWO classes: NORMAL and ABNORMAL using MLP





Data Pre-processing

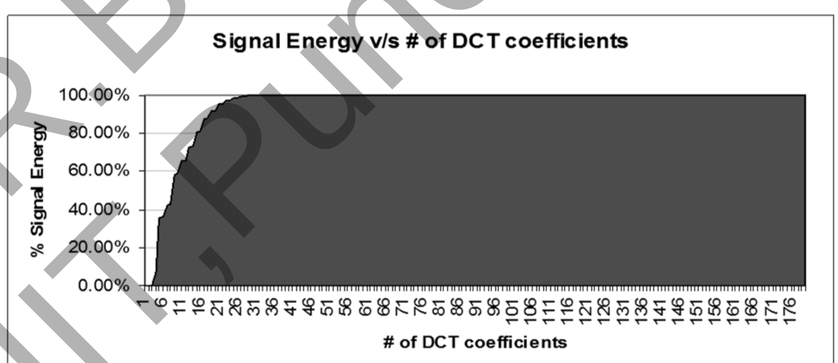
- Mean adjust the data to remove the DC component
- We can use all 180 points for training but it poses a computational overhead
- Using features for training minimizes the response of ANN to noise present in the signal
- Training time of the ANN reduces
- Overall accuracy of the ANN improves
- Generalization of the network improves
- Feature extraction in transform domain can use
  - Discrete Cosine Transform (DCT)
- One-hot encoding technique for the target

Selection of significant components

- Metrics for component selection
  - Components that retain 99.5% of the signal energy
  - Percent Root Mean Difference (PRD)

$$PRD = \sqrt{\frac{\sum_{i=0}^{n}\left[x_{original}(i) - x_{reconstructed}(i)\right]^2}{\sum_{i=0}^{n} x_{original}^2}} \times 100$$

Discrete Cosine Transform

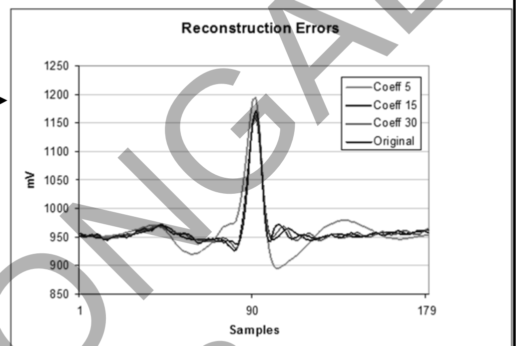


- Thirty DCT components contribute to 99.86% of the signal energy
- PRD(30 components)=0.5343%

Discrete Cosine Transform- selection of significant coefficients

Sr.	Transform	Coefficients	PRD %	Signal Energy (%)
1	DCT	5	9.8228	35.81
2		15	5.4096	80.53
3		**30**	**0.5343**	**99.86**
4		40	0.3134	99.93

Effect of truncating DCT coefficients



Dataset Creation

- It is always desirable that we have equal number of exemplars from each class for the training dataset
- This prevents "favoring" any class during training
- If the number of exemplars are unequal we have to de-skew the classifier decision
- De-skewing simply scales the output according to the probability of the input classes
- Data randomization before training is a must other wise repetitive training by same class may not allow the network to converge, remember that the error gradient is important for achieving global minima , if it exists!

- Partition the data into THREE disjoint sets
  - Training set
  - Cross-validation set
  - Testing set
- Before the data is presented to the net for training we have to ***normalize*** the data in range [-1,1] , this helps in faster network learning
- Amplitude and Offset are given as:

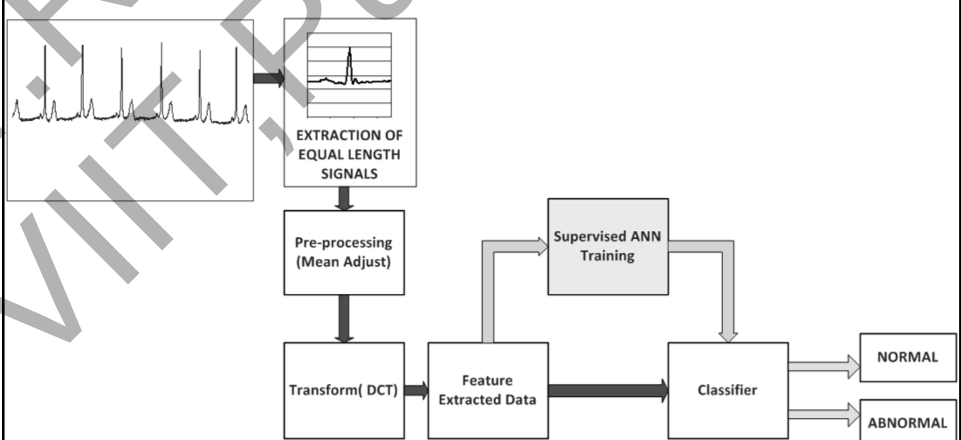
$$Amp(i) = \frac{[UpperBound - LowerBound]}{\max(i) - \min(i)}$$

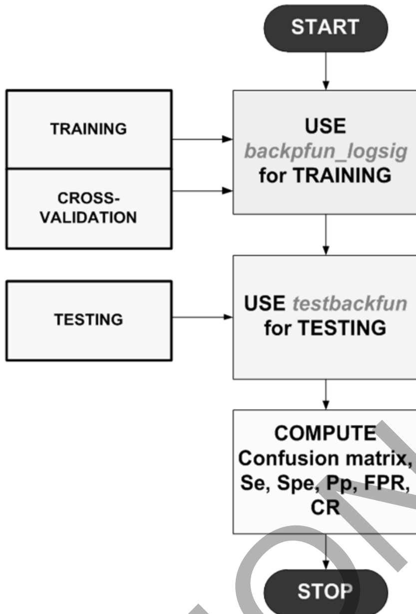
$$Off(i) = UpperBound - Amp(i) \cdot \max(i)$$

To normalize data: $$Data(i) = Amp(i) \cdot Data(i) + Off(i)$$

To de-normalize data: $$Data(i) = \frac{Data(i) - Off(i)}{Amp(i)}$$

Methodology



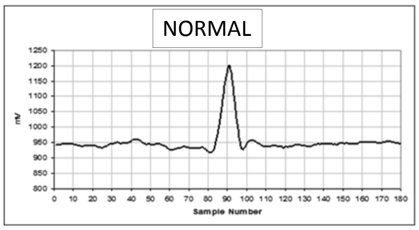


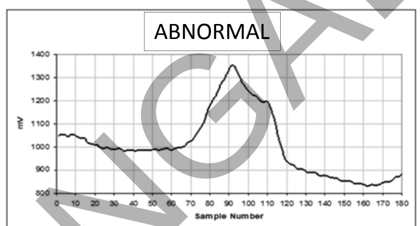
MLP Classifier MATLAB DEMO

Case Study : Two-Class QRS classification with RBFN

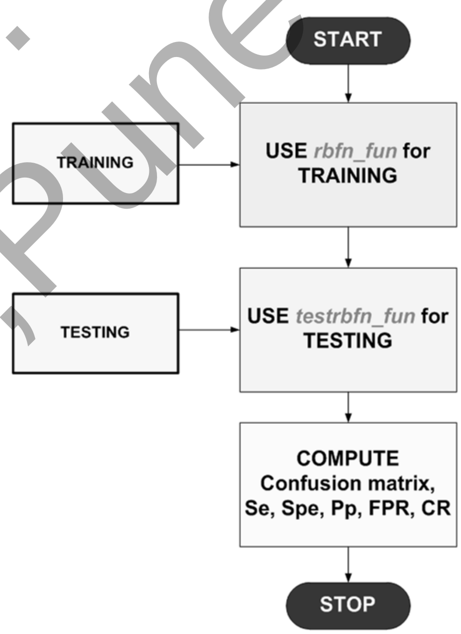
Problem Statement:

Design a system to correctly classify extracted QRS complexes into TWO classes: NORMAL and ABNORMAL using RBFN





We use the same pre-processed data from previous case study



RBFN Classifier MATLAB DEMO

Thank You!