Probabilistic Neural Networks

Krishna Ganesula

CPSC 636: Neural Networks Spring 2010

Instructor: Dr. Ricardo Gutierrez-Osuna

Outline

- Context and Problem
- Bayesian Strategy
- Probabilistic Neural Network
- Comparison and Analysis
- Recent Work and Conclusions

Context

Author – Dr. Donald F. Specht

Lockheed Missiles & Space Company, Inc.

Published in 1989-90

Neural Networks - Volume 3

Radial Basis Function (RBF) Networks and Bidirectional Associative Memory (BAM) Networks were proposed about the same time.

Example – Cancer Diagnosis

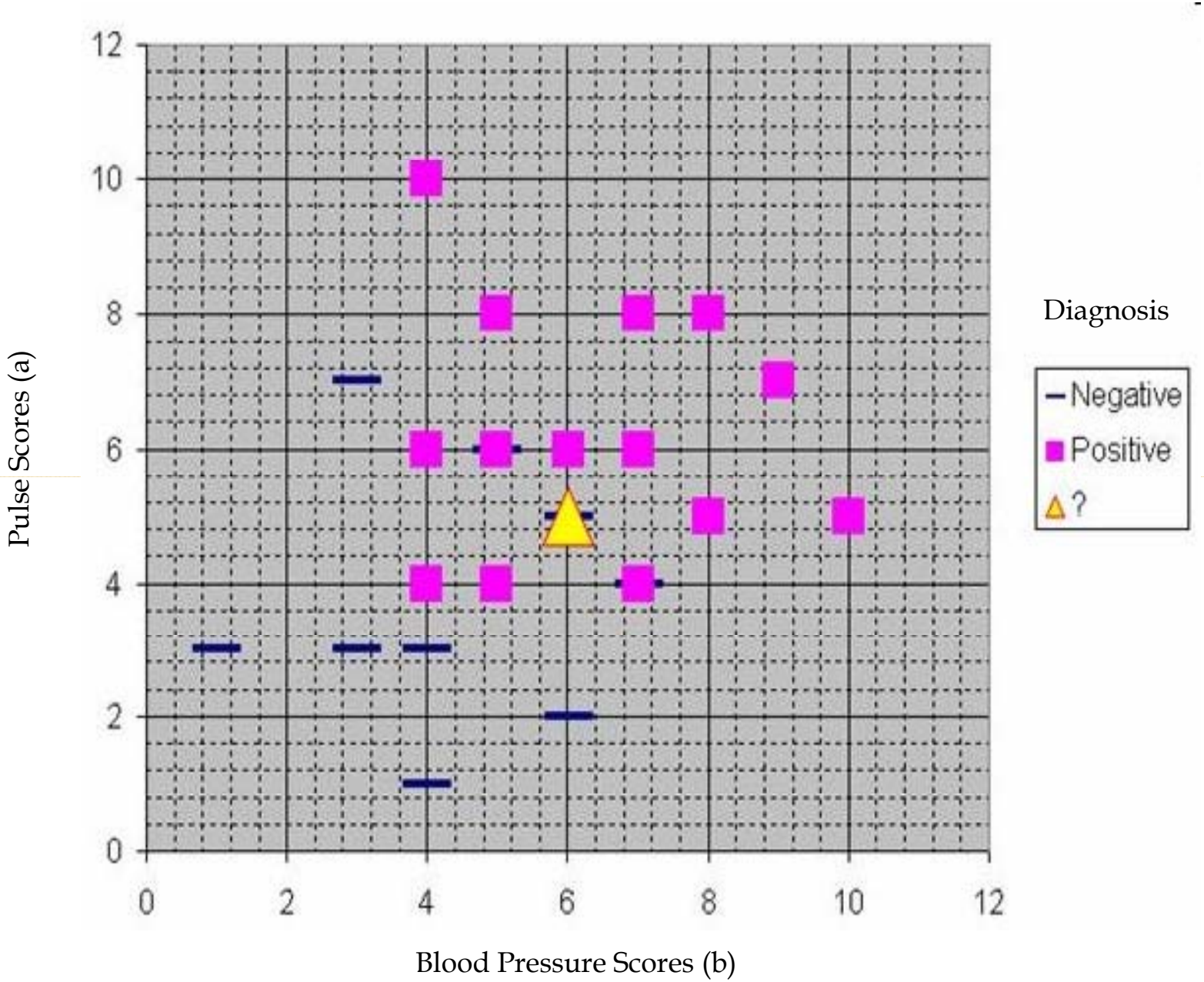
Prior Info –

(Pulse → a, Blood pressure → b, N samples, 1≤ i ≤N)

Past Scores	$\boldsymbol{X_i = (x_i(a), x_i(b))}$
True Diagnosis	$\mathbf{d_i}$

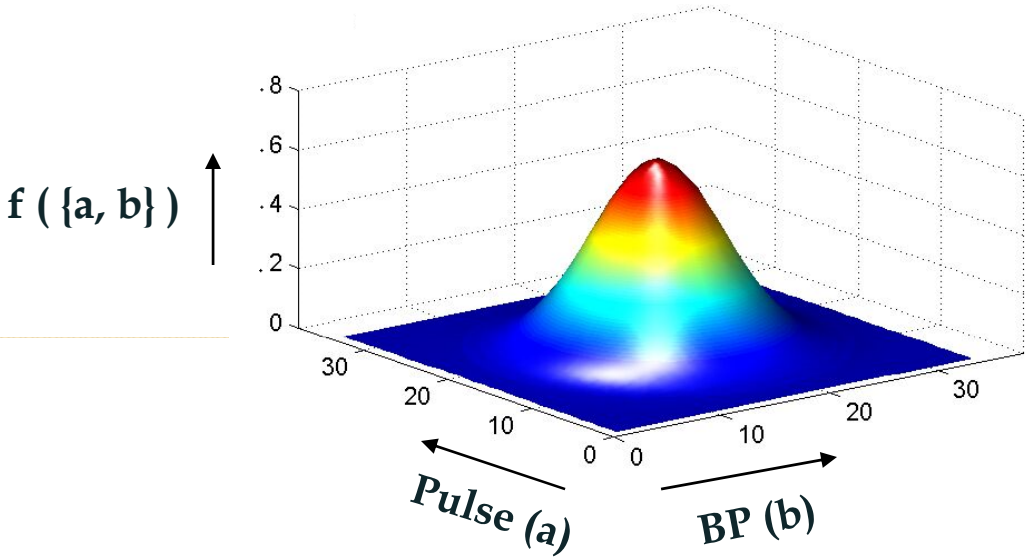
Task –

New score	$\boldsymbol{X = (x(a), x(b))}$
Predict	**d** (?)


Pulse Scores (a)
12
10
8
6
4
2
0
0
2
4
6
8
10
12
Blood Pressure Scores (b)
Diagnosis
Negative
Positive
?

KNN (1NN → [–] , 9NN → [+])

Probability Density Function (PDF)



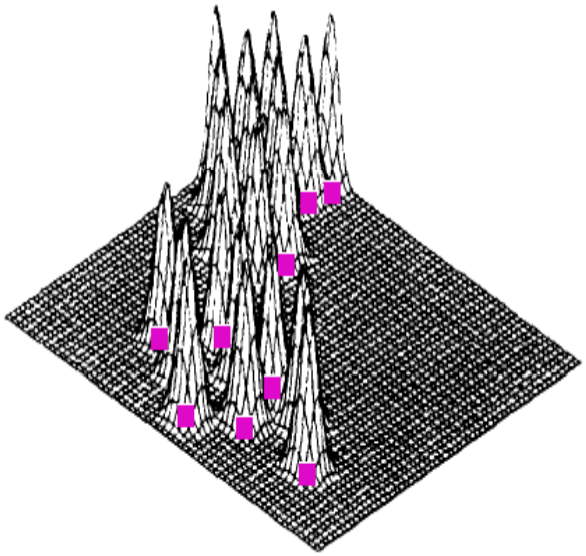
PDF for +ve samples

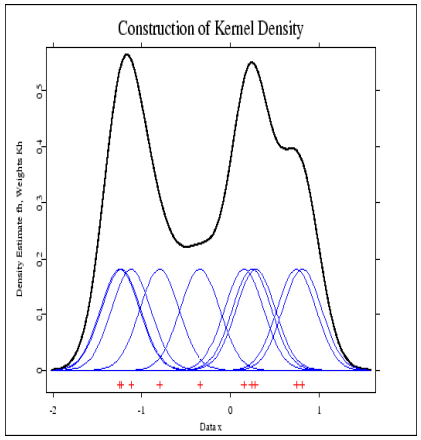
Parametric PDF

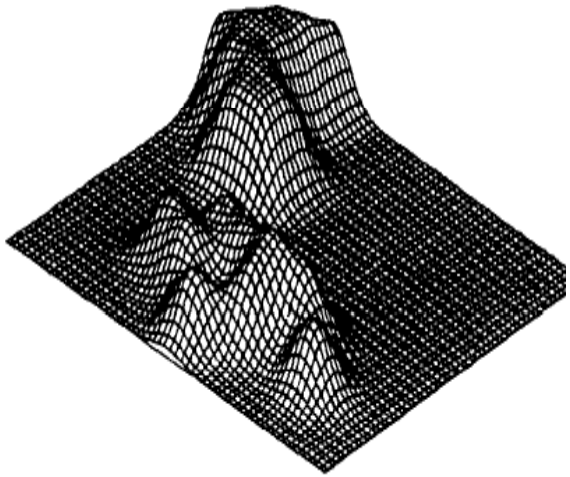
- Assume the PDF is Gaussian as above.
- Find f(X), $f_{[+]}(X) > f_{[-]}(X) \Rightarrow X \in [+]$

But the underlying distribution isn't always normal.

Kernel Density Estimation (Parzen Window)







$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

K → *kernel (weight) function*

h → *smoothing parameter*

(Probability distribution of X is continuous)

Kernel (Weight) → Gaussian

$$f(X) = \frac{1}{(2\pi)^{p/2}\sigma^p} \frac{1}{m} \sum_{i=1}^{m} \exp\left[- \frac{(X - X_i)^t (X - X_i)}{2\sigma^2} \right]$$

where

$X_i = i_{th}$ *training sample from category* [+]

σ = *smoothing parameter*

m = *number of training samples*

Is f(X) enough?

Prior Probabilities ($P_{[+]}$, $P_{[-]}$)

- Sample Inconsistency

Misclassification ($L_{[+]}$, $L_{[-]}$)

- Account for serious mistakes

$$\mathbf{X \in [+] \Leftrightarrow P_{[+]} L_{[+]} f_{[+]}(X) > P_{[-]} L_{[-]} f_{[-]}(X)}$$

where

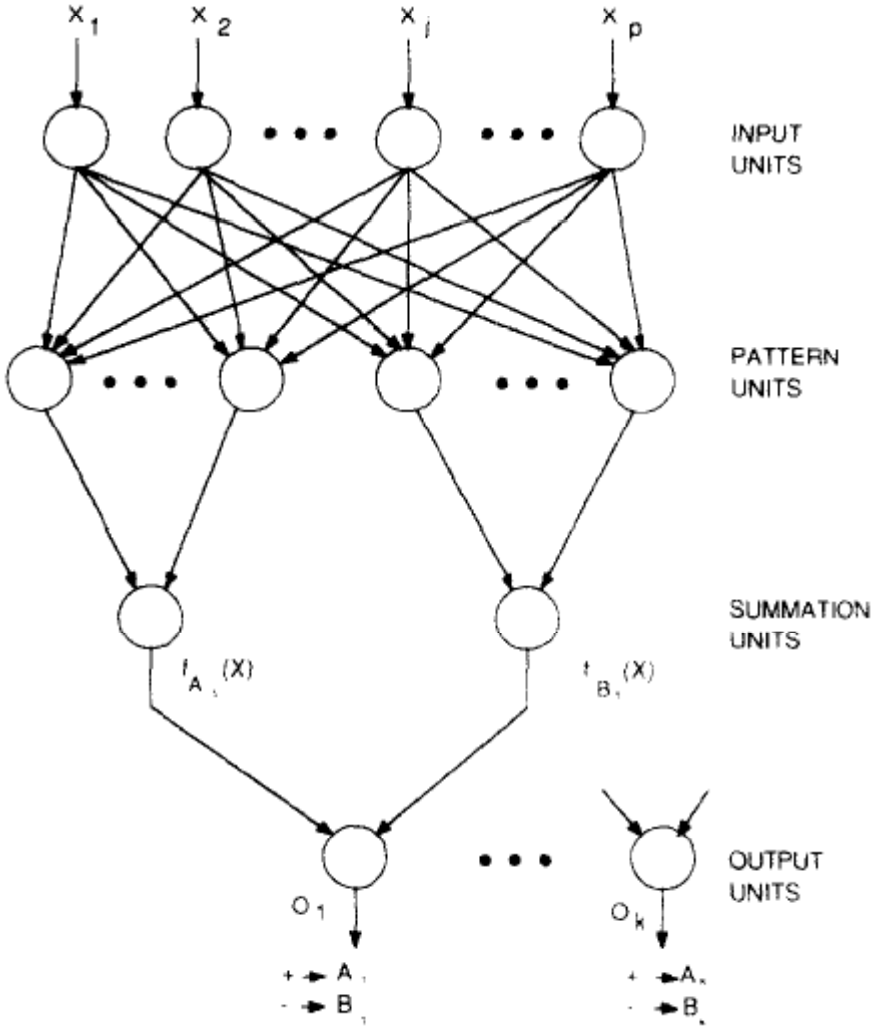
$f_{[+]}(X)$ *is the PDF values for* [+] *samples*

$L_{[+]}$ *is the loss function for the decision* $X \in [+]$ *when* $X \in [-]$.

$p_{[+]}$ *is the prior probability of positive diagnosis*

Probabilistic Neural Networks

- Architecture



Input Layer

Input vector

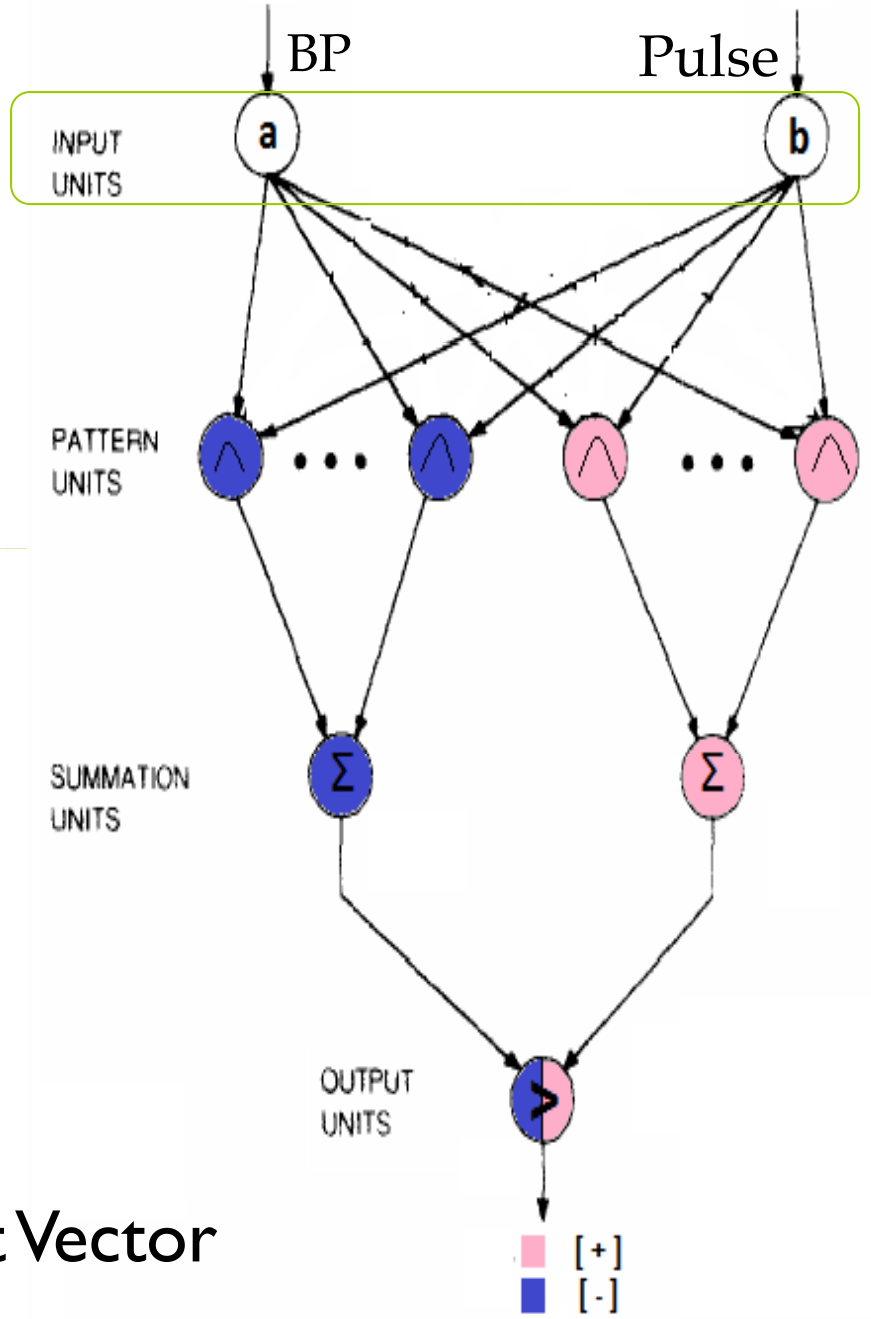
$\mathbf{X} = (\mathbf{X}_a, \mathbf{X}_b)$

where

a → Pulse score

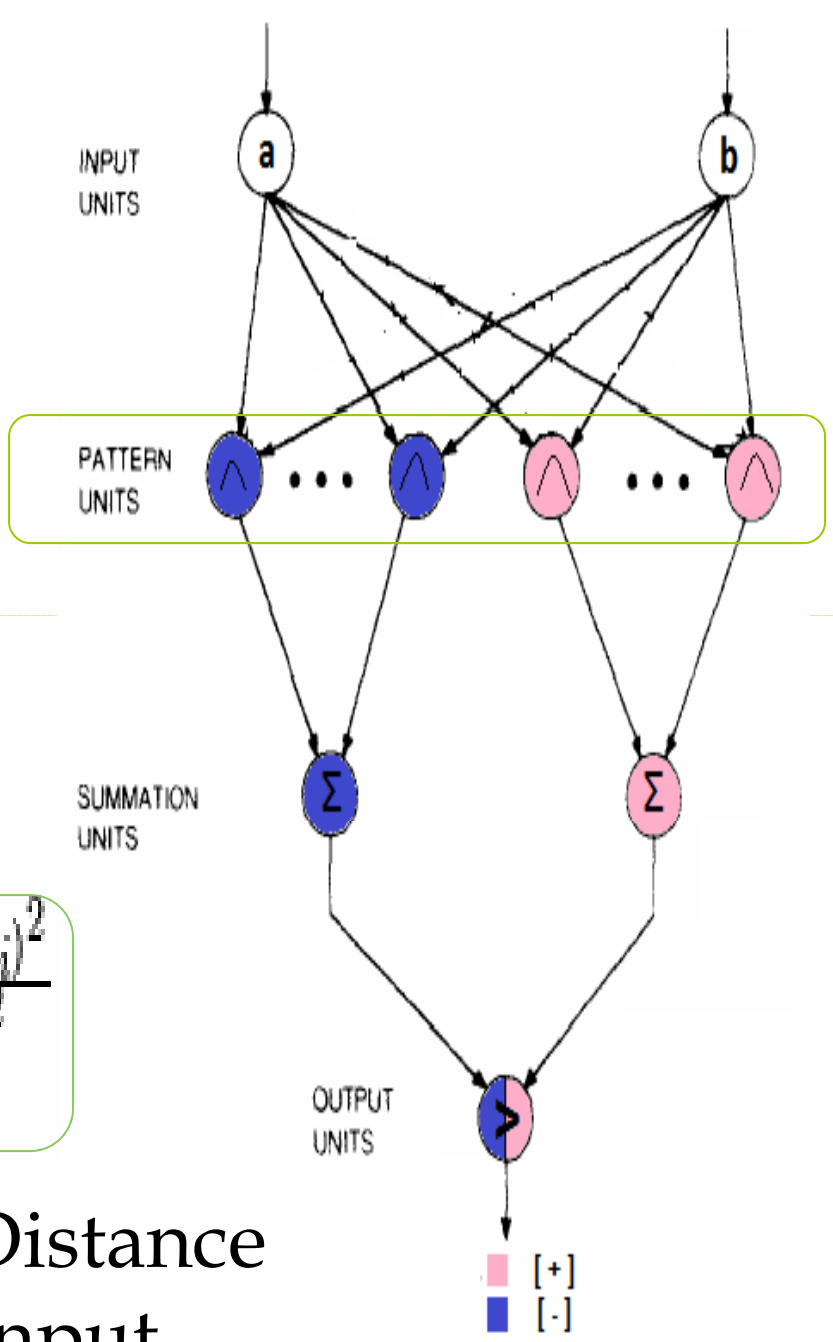
b → Blood Pressure score

Weights of edges = Input Vector



Pattern Layer

Each training sample has a corresponding pattern unit



Kernel Function K

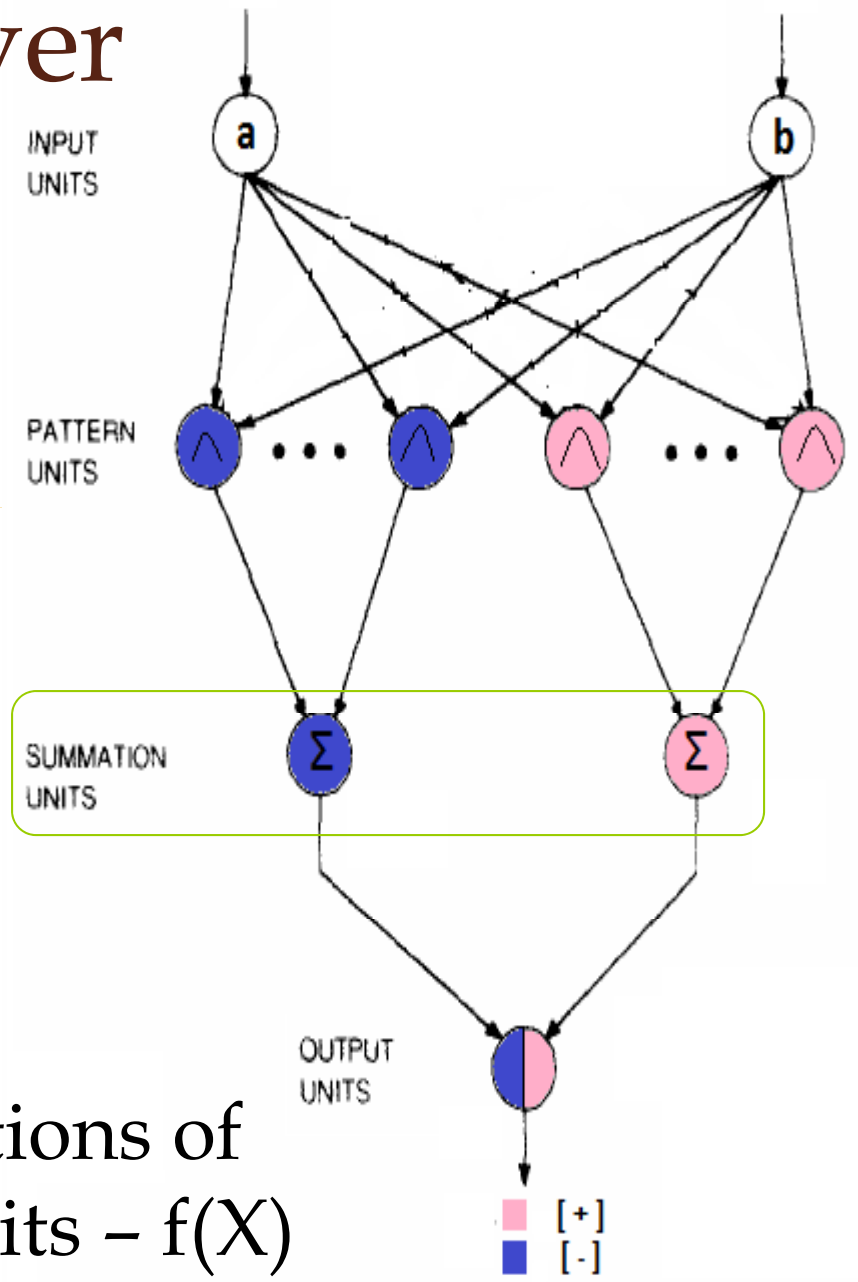
$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-x_i)^2}{2h^2}}$$

Computes Gaussian Distance of each sample from input.

Summation Layer

Each category has a corresponding summation unit

Only connected to pattern units of same category



Sums up kernel functions of connected pattern units – f(X)

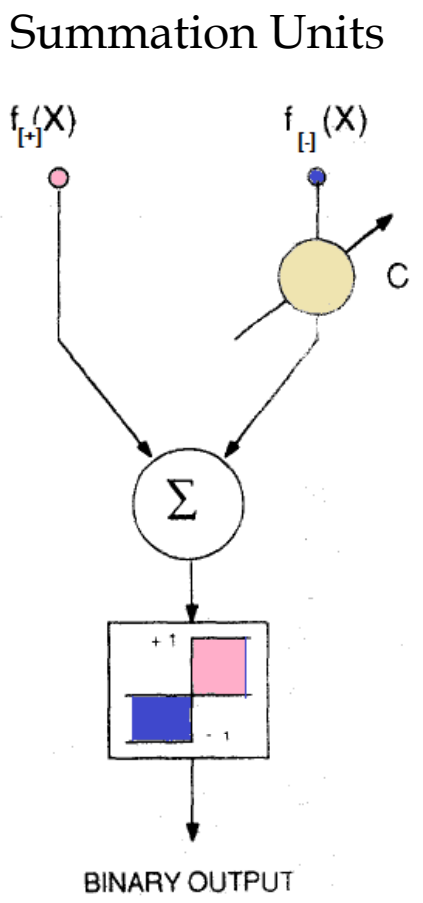
Output Layer

In this case we have binary output for two categories.

Prior probability and loss figures are added to the PDF as a weight

$$C = -\frac{P_{[-]}L_{[-]}}{P_{[+]}L_{[+]}} \times \frac{N_{[+]}}{N_{[-]}}$$

For proportional training C is ratio of losses. In case of no losses we get an inverter C = -1



Comparing with MLPs

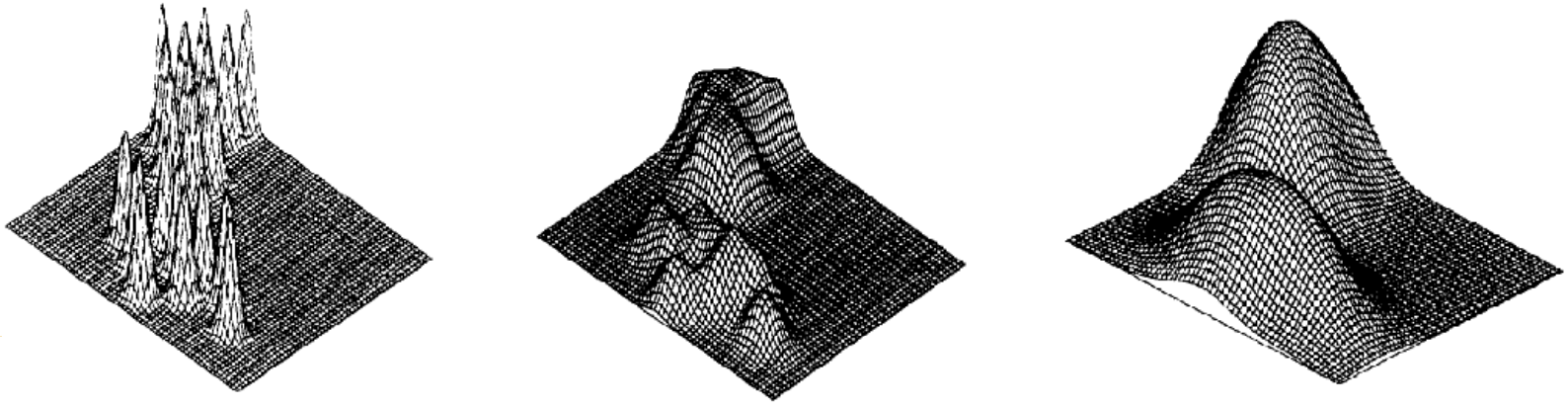
Advantages

- Virtually no time consumed to train.
- Relatively sensitive to outliers.
- Can generate probability scores.
- Can approach Bayes optimal.

Disadvantages

- Testing time is long for a new sample.
- Needs lot of memory for training data.

Smoothing effect (σ)



a. A small value of σ. b. A larger value of σ. c. An even larger value of σ.

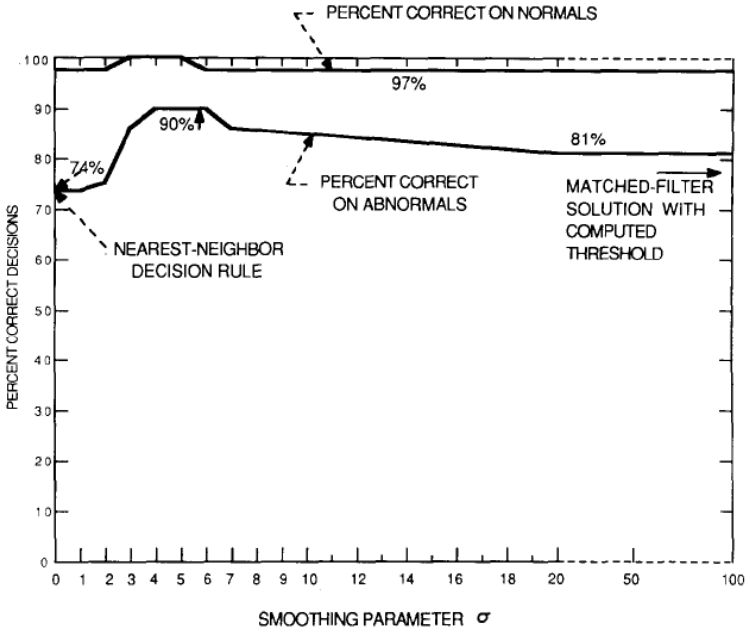
Nature of the PDF varies as we change σ.

- Small σ creates distinct modes
- Larger σ allows interpolation between points
- Very Large σ approximate PDF to Gaussian

So how do we choose σ ?

- neither limiting case

σ→0 or σ→∞ is optimal.



- No of neighbors to average should depend on the density of training samples
- Easy to find in practice, also low effect on error rate.
- Graph shows that any σ value between 4-10 gives close to optimal results.

Further Modifications

Alternate estimators

- Use Alternate univariant kernels
- Causes changes in activation function
- But returns same optimal result

Associative Memory

- Maximize PDF to estimate unknown input variable.
- For more than one unkown variable used a generalized global PDF f(X')

Recent Applications of PNNs

- *Ship Identification Using Probabilistic Neural Networks (PNN)* by LF Araghi , Proceedings of IMECS
- *Application of Probabilistic Neural Network Model in Evaluation of Water Quality* by Changjun Zhu, Zhenchun Hao, Environmental Science and Information
- *A probabilistic neural network for earthquake magnitude prediction* by H Adeli, Neural Networks Vol 22
- *Detection of Resistivity for Antibiotics by Probabilistic Neural Networks* by Fatma Budak and Elif Derya, Journal of Medical Systems

Conclusions

- PNNs are the neural network way of implementing non parametric PDF estimation for classification.
- PNNs are faster to train and approach the Bayes optimal as the training set increases.
- It is vital to find an accurate smoothing parameter σ.
- PNNs today are being widely researched to find more efficient classification solutions.

Questions?