Johns Hopkins Engineering

Applied Machine Learning for Mechanical Engineers

Popular Supervised Machine Learning Techniques, Part 1, B

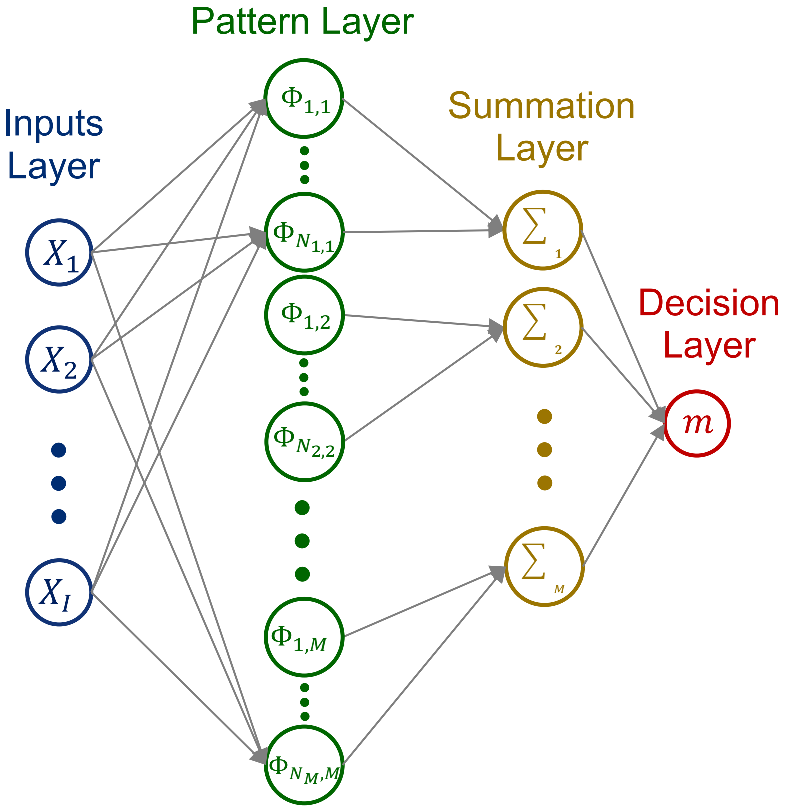
JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Probabilistic and Enhanced Probabilistic Neural Networks

- By the end of this lecture you will be able to:
  - Describe Probabilistic Neural Networks (PNN)
  - Describe Enhanced Probabilistic Neural Networks(EPNN)

Probabilistic and Enhanced Probabilistic Neural Networks

- Probabilistic Neural Networks* (PNN)
  - Four-Layer Neural Network
    - Input layer
    - Pattern Layer
    - Summation Layer
    - Decision Layer
  - Classification problems

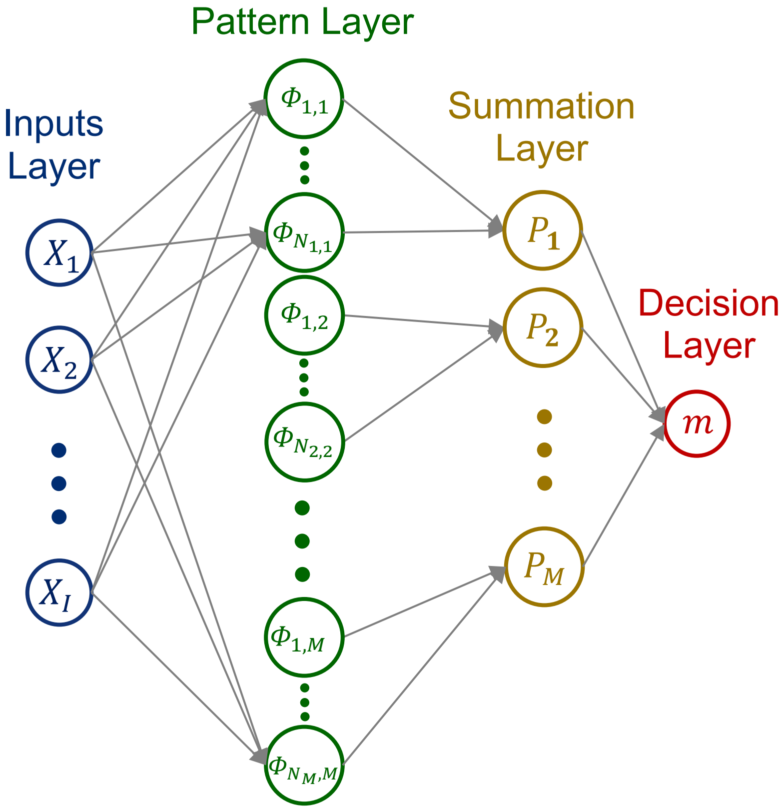


*Specht, D. F. (1990). Probabilistic neural networks. *Neural networks*, *3*(1), 109-118.

Probabilistic and Enhanced Probabilistic Neural Networks

- Probabilistic Neural Networks (PNN)
  - Pattern Layer
    - Compute the likelihood pattern of an I-dimensional vector features (e.g. testing) $\boldsymbol{X}$ being in the same class as $\boldsymbol{V}_{n,m}$, the n^{th} training data belonging to class m
    - Likelihood function
      $$\Phi_{n,m}(\boldsymbol{X}) = \frac{1}{(2\pi)^{\frac{I}{2}}\sigma^{I}} \exp\left(-\frac{\|\boldsymbol{X}-\boldsymbol{V}_{n,m}\|}{2\sigma^2}\right)$$

      where σ is the spread of the Gaussian function ($\sigma \in [0\ 1]$) chosen to minimize (minimization problem) the PNN estimation error for a given dataset and $\|\boldsymbol{X}-\boldsymbol{V}_{n,m}\|$ is the Euclidean distance between $\boldsymbol{X}$ and $\boldsymbol{V}_{n,m}$

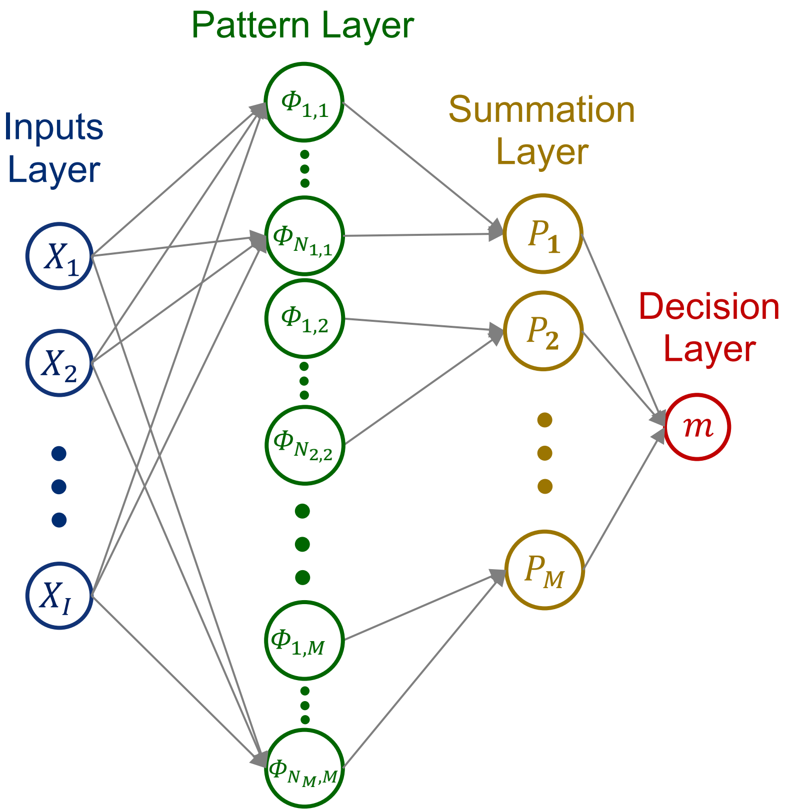


Probabilistic and Enhanced Probabilistic Neural Networks

- Probabilistic Neural Networks (PNN)
  - Summation Layer
    - Compute the average likelihood pattern of $\boldsymbol{X}$ being in the same class as all the training data belonging to class m
    - Average likelihood

$$P_m(\boldsymbol{X}) = \frac{1}{N_m} \sum_{n=1}^{N_m} \Phi_{n,m}(\boldsymbol{X})$$

where N_m is the number of training data of class m

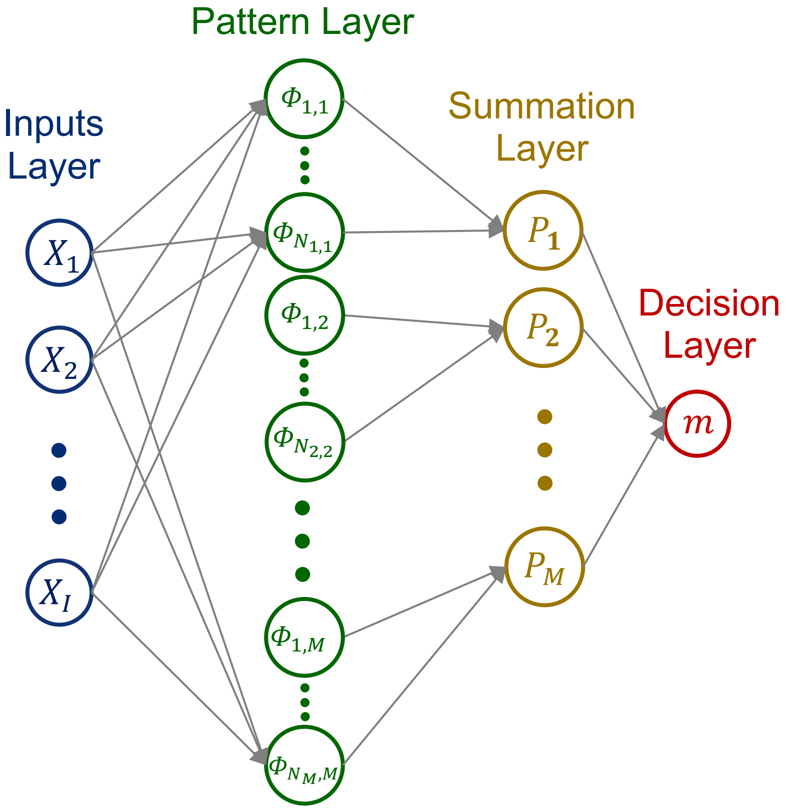


Probabilistic and Enhanced Probabilistic Neural Networks

- Probabilistic Neural Networks (PNN)

  - Decision Layer
    - The estimated class of $\boldsymbol{X}$ is the one with the maximum average likelihood:

$$Class(\boldsymbol{X}) = \arg\max_{m} \{P_m(\boldsymbol{X})\}$$



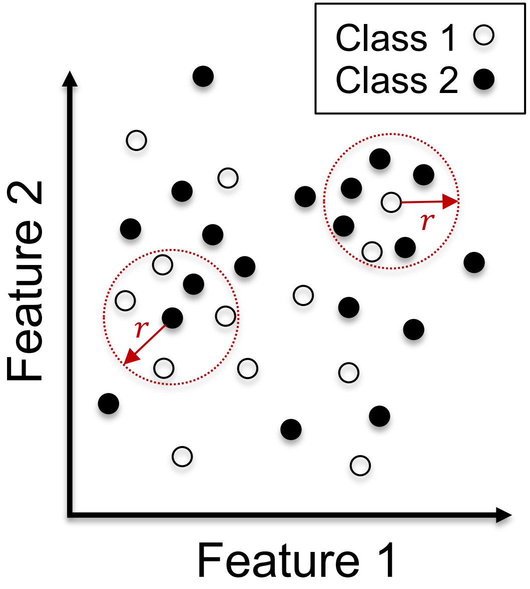
Probabilistic and Enhanced Probabilistic Neural Networks

- Enhanced Probabilistic Neural Networks[*] (EPNN)
  - Likelihood function
    - $\Phi_{n,m}(\boldsymbol{X}) = \frac{1}{(2\pi)^{\frac{I}{2}}\sigma^{I}} \exp\left(-\frac{\left\|\boldsymbol{X}-\boldsymbol{V}_{n,m}\right\|}{2\sigma^{2}}\right)$
    - Spread of Gaussian function, $\sigma \in [0\ 1]$, is the key variable
    - Low magnitude σ means higher $\Phi_{n,m}$ and vice versa
    - In PNN, optimum σ is being computed and fixed for all n and m
    - What if we can individualize σ for every n and m, and create $\sigma_{n,m}$ depending on where $\boldsymbol{V}_{n,m}$ is located in the Euclidean space?
    - Let's define a conditional probability multiplier, $0 < \alpha_{n,m} \leq 1$, where $\sigma_{n,m} = \alpha_{n,m} \times \sigma$. Let's talk about $\alpha_{n,m}$!

[*]Ahmadlou, M., & Adeli, H. (2010). Enhanced probabilistic neural network with local decision circles: A robust classifier. Integrated Computer-Aided Engineering, 17(3), 197-210..

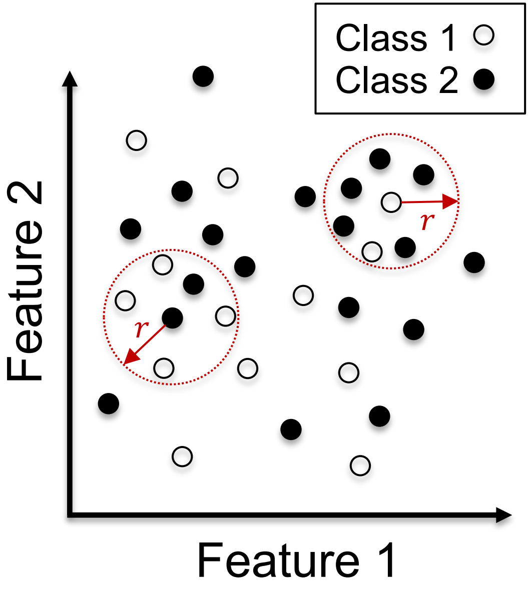
Probabilistic and Enhanced Probabilistic Neural Networks

- Enhanced Probabilistic Neural Networks (EPNN)
  - Local decision hyper-sphere to define an $\alpha_{n,m}$
    - Assume $S_{n,m}$ is an $I-$dimensional hyper−sphere with the radius $r \in [0\ 1]$ and center $\boldsymbol{V}_{n,m}$
    - The conditional probability of an embedded training data in $S_{n,m}$ to be in the same class as $\boldsymbol{V}_{n,m}$ (class m), denoted as $\alpha_{n,m} \in [0\ 1]$, is the ratio of the number of training input vectors with the same class as $\boldsymbol{V}_{n,m}$ in $S_{n,m}$, to the number of all training input vectors in $S_{n,m}$.



Probabilistic and Enhanced Probabilistic Neural Networks

- Enhanced Probabilistic Neural Networks (EPNN)
  - Likelihood function for EPNN
    - $\Phi_{n,m}(\boldsymbol{X}) = \frac{1}{(2\pi)^{\frac{I}{2}}\sigma_{n,m}^{I}} \exp\left(-\frac{\|\boldsymbol{X}-\boldsymbol{V}_{n,m}\|}{2\sigma_{n,m}^{2}}\right)$
    - $\sigma_{m,n} = \alpha_{m,n} \times \sigma$
  - Now we are to optimize both σ and r



Probabilistic and Enhanced Probabilistic Neural Networks

- In this lecture, you learned about:
  - Probabilistic Neural Networks (PNN)
  - Enhanced Probabilistic Neural Networks (EPNN)
- In the next lecture, we will talk about Convolutional Neural Networks

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING