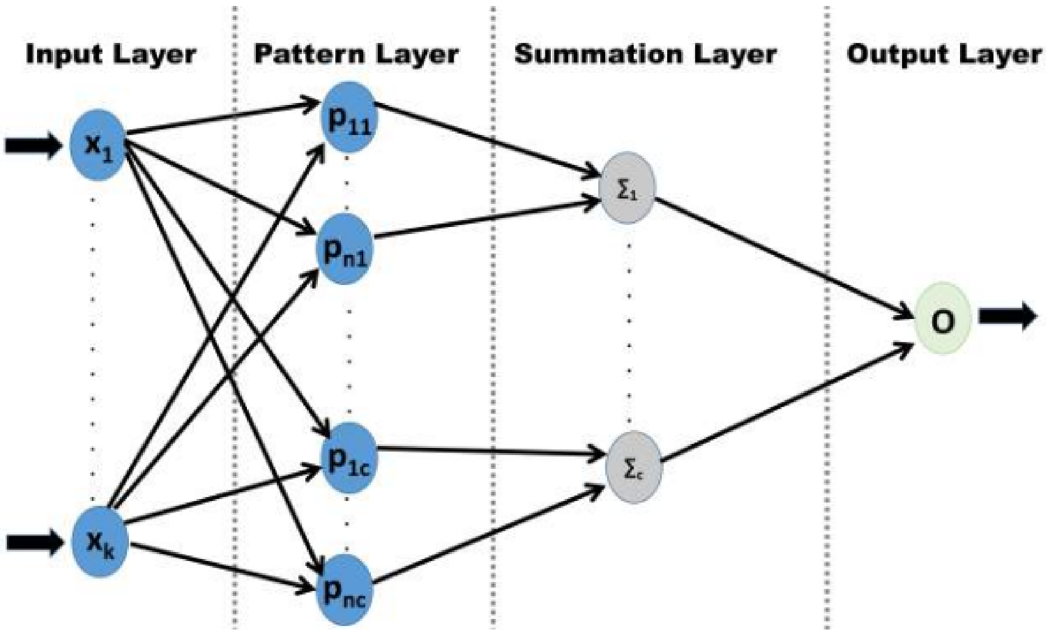
Probabilistic Neural Networks



1. If there are ***c*** *classes* in your data and ***n*** *samples* in the training set, there are a total of ***n×c*** *pattern layer neurons*.
2. Each *pattern layer* neuron calculates the probability of how well its sample input vector fits into the class represented by that neuron.
3. The third layer is the *summation layer*, in which the number of neurons is same as the number of classes present in our dataset. Assuming ***c*** *classes*, there will be ***c*** *summation layer* neurons. For each class of testing inputs the summation layer neurons sum the contributions of the previous layer output, and produce an output vector of probabilities related to the class they represent.

Pattern Layer computes its output using the *Gaussian Kernel* $\phi_{ic}(x)=\frac{1}{(2\pi\sigma^2)^{k/2}}\exp^{-\frac{(x-x_{ic})^2}{2\sigma^2}}$

Parameter σ is known as the *smoothing parameter* and controls the spread of the Gaussian function, smaller values of σ allows the estimated pdf to take non-gaussian shapes. Optimal σ value will be obtained using cross validation.

4. Finally, the *output layer* calculates the probabilities for each of the possible classes for which the input data can be classified. Output layer neurons classify the pattern vector x using Bayes's decision rule:-

$$C(x)=arg\,max_{i\leq c\leq C}f(c)$$