

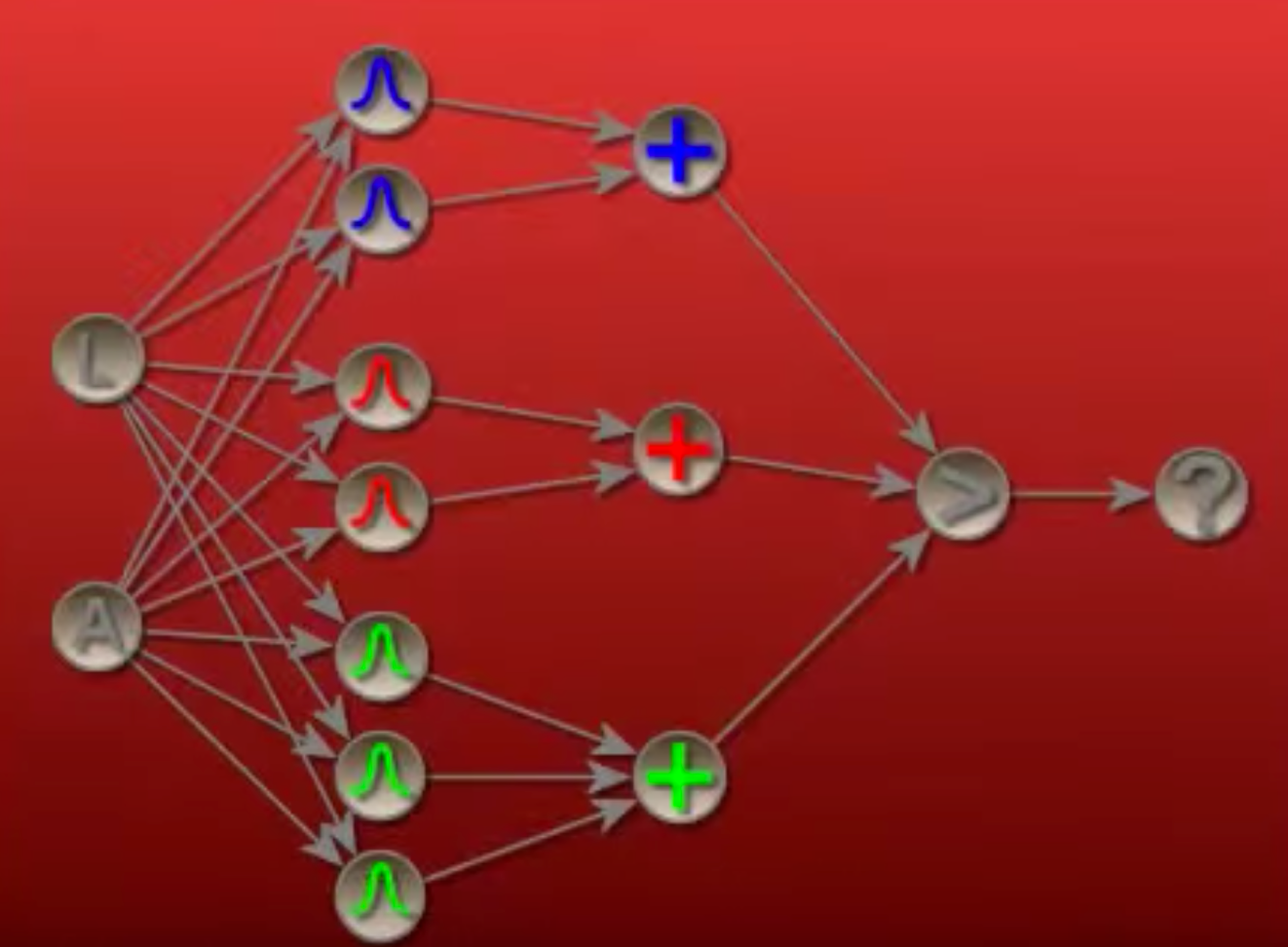
PNN in Python

Jae Duk Seo

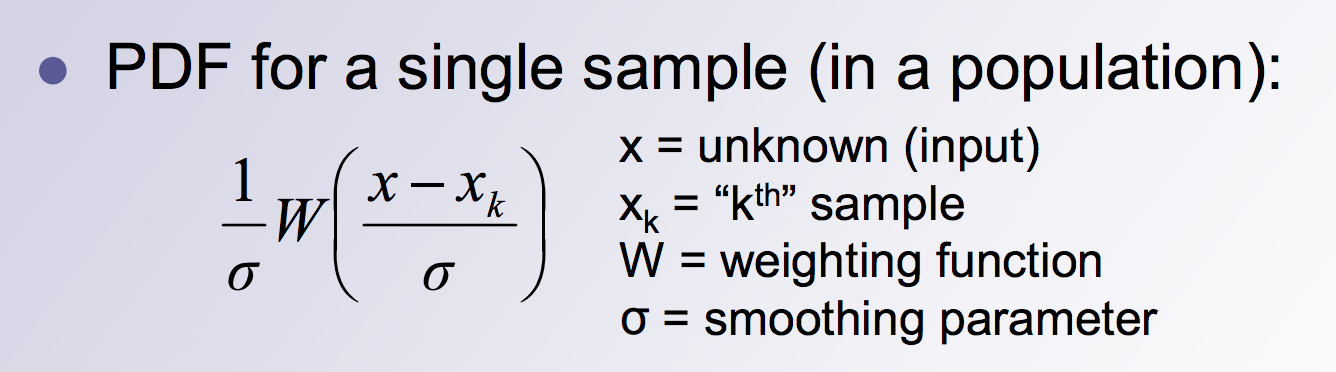
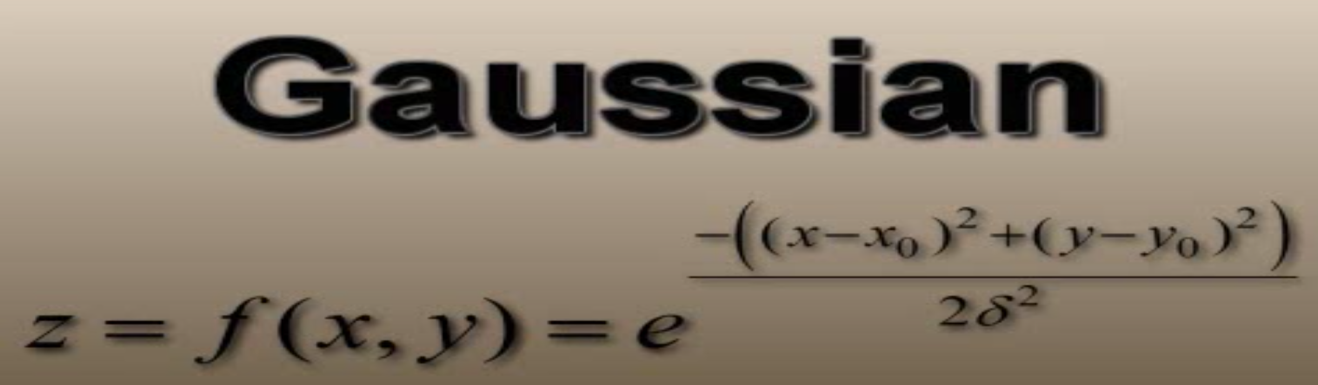
Jae Duk Seo – March 15

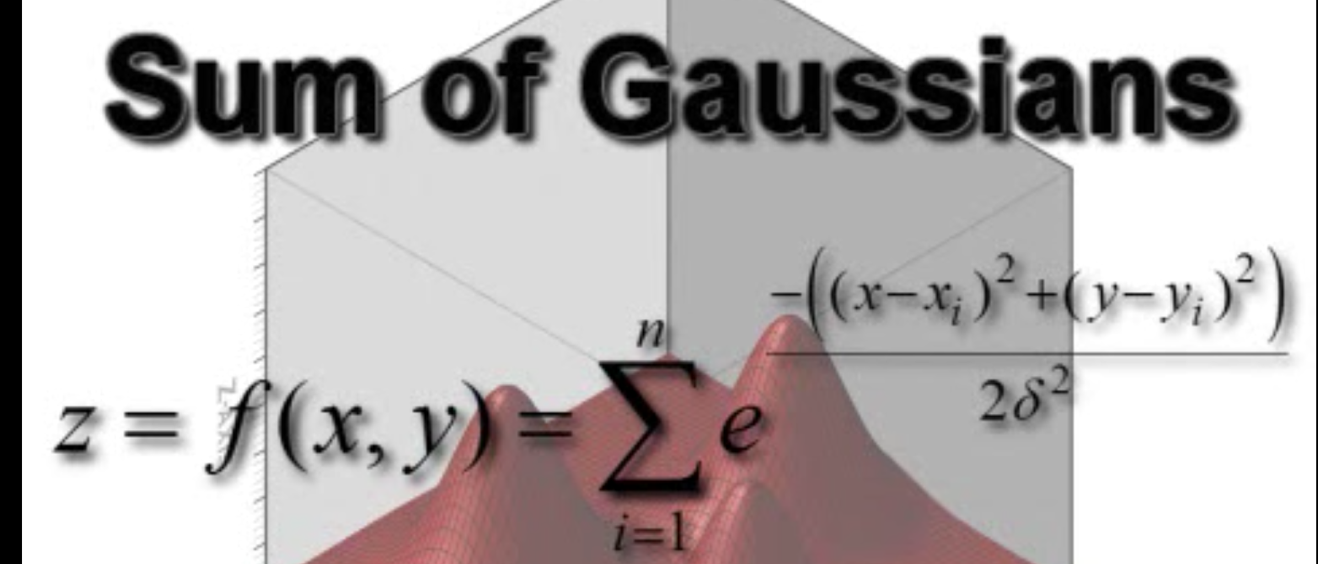
# Probabilistic Neural Networks

- Implementation of statistical algorithm called ‘kernel discriminant analysis’ in Neural Network Style.

- Have 4 layers   
a) Input Layer - Input the features of data points we wish to classify   
b) Pattern Layer – For each class calculate similarity  
c) Summation Layer - Sum up the calculated similarity (for simplicity) for each class  
d) Output Layer – Get the highest probability of class  
  
  
- There are three class – (Blue/Red/Green)

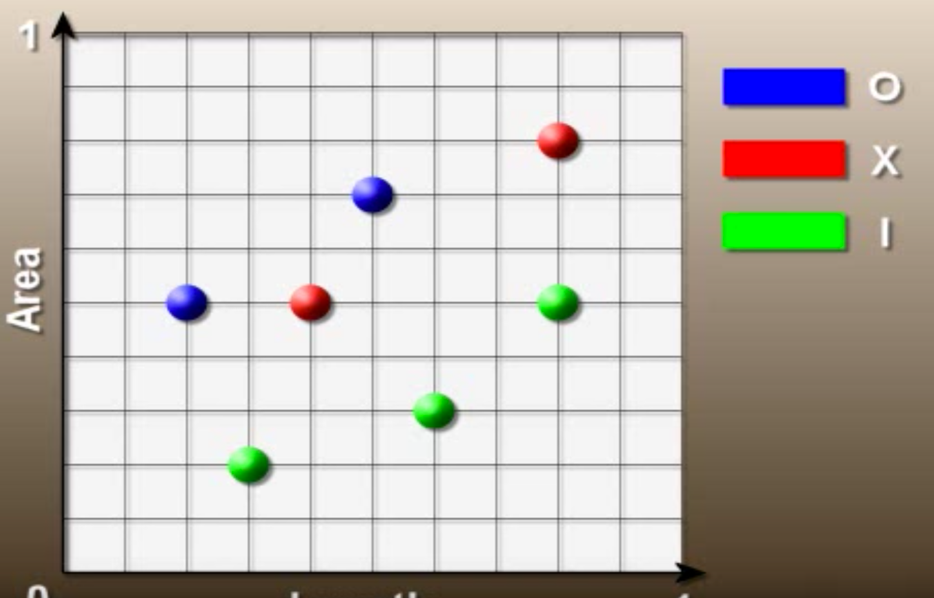
# probability density function

* Method to calculate “similarity”   
  
* Xk – this means the class k – Above example we have either (Blue/Red/Green)
* We can use **Gaussian function for Weighting Function**
* 
* The full transformation of the function

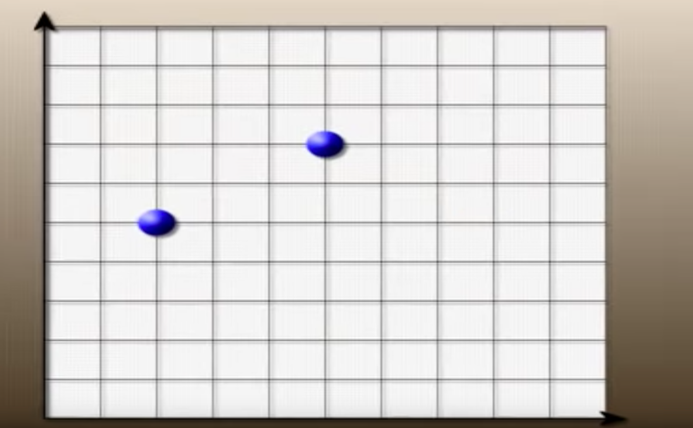


# Visualization of an example

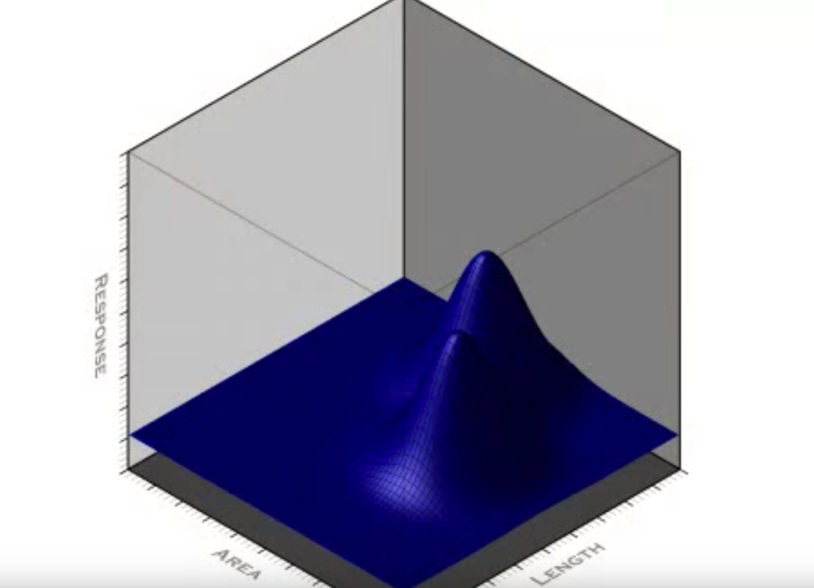
* Imagine a 2D plane with 3 class of data points



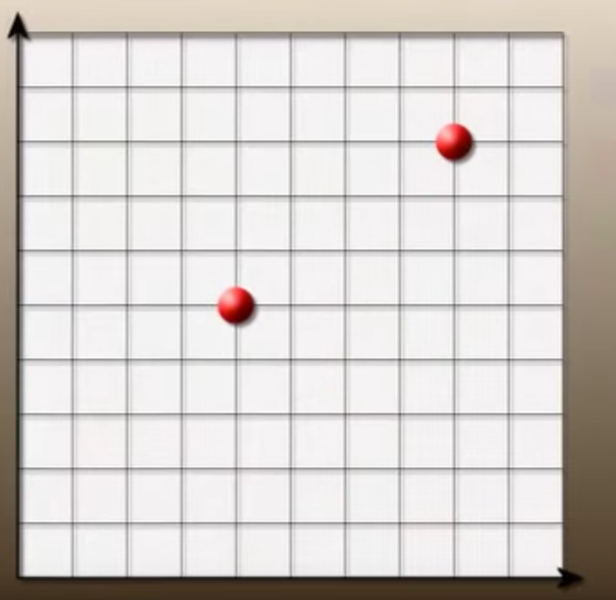
* Only take the blue class

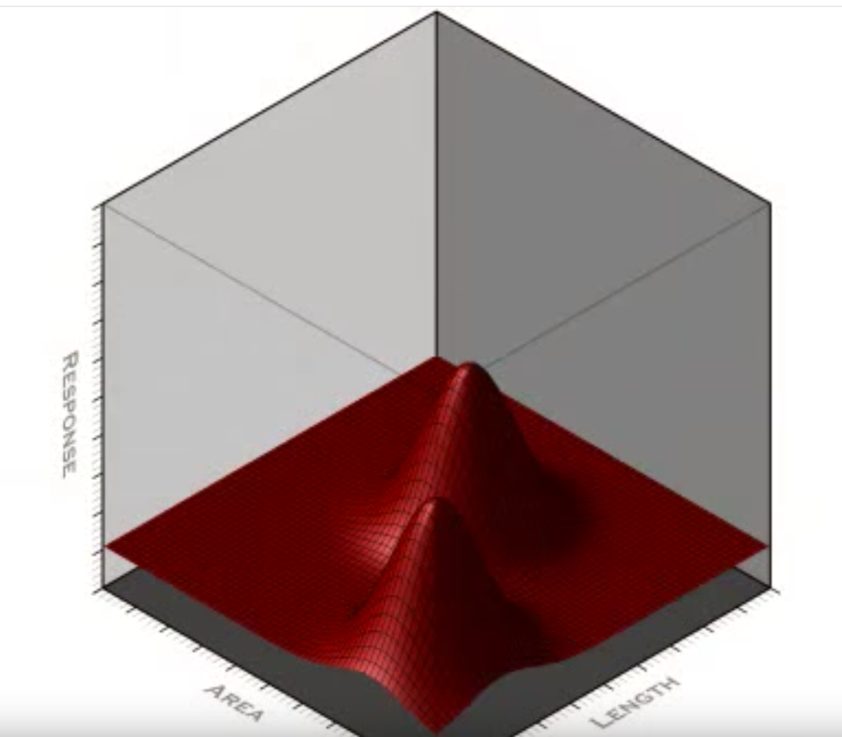
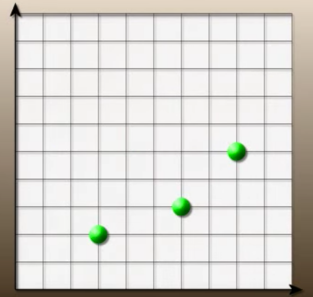
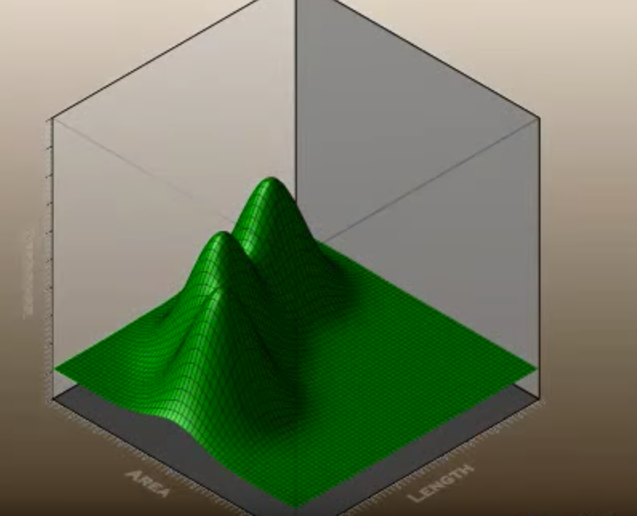


* Imagine that we are drawing a Gaussian distribution of the data point



* Now only take the red classes



* Exact same idea – Gaussian distribution   
  
* Finally the green point  
  
* 
* Now go back to the 2D plane -   
  