MSPR9 Classification and Clustering

Dr. Hendrik Purwins,
Aalborg University Copenhagen

Parametric Classifiers

- The statistical theory behind a *parametric classifier* assumes that the data for each class is distributed according to a particular probability distribution (often a Gaussian).
- We fit this probability distribution (often a Gaussian) to the data of each class, just by estimating the *parameters* (e.g. mean and covariance matrix) of that function.
- Even if we are not sure, whether the data of each class follows a particular distribution, we may still use a parametric classifier and evaluate its performance, based on a test/cross-validation error.

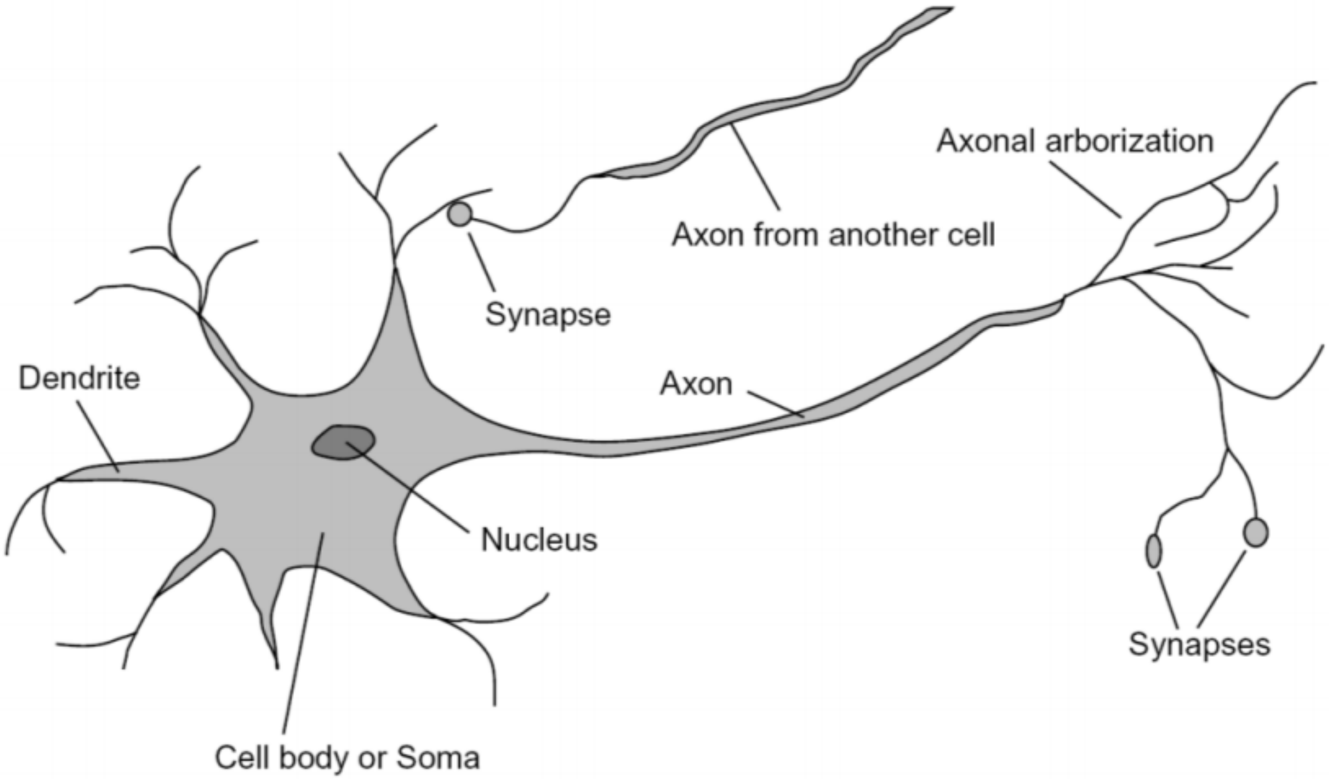
Parametric Classifiers: Examples

- Nearest mean classifier:
  - Assumption: each class is distributed according to a normal distribution with the same covariance matrix of Type $\Sigma = \sigma^2 I$ (no values on the off-diagonal, same value everywhere on the diagonal)
- Linear discriminant analysis (nearest Mahalanobis classifier):
  - Assumption: each class is distributed according to a normal distribution with the same covariance matrix of any form.
- Quadratic discriminant analysis
  - Assumption: each class is distributed according to a normal distribution with a class-specific covariance matrix Σ_k
- Regularized discriminant analysis
  - Assumption: each class is distributed according to a normal distribution where the covariance matrices of the two classes depend on each other to a certain degree, given by a regularization parameter (compromise between linear and quadratic DA)

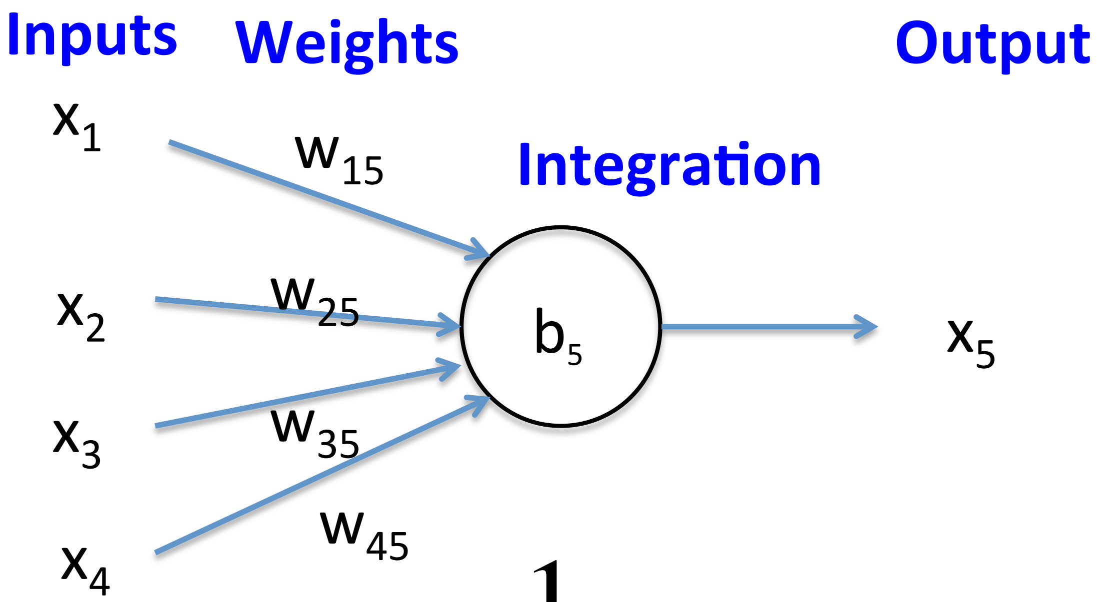
Non-parametric Classifiers

- The statistical theory behind a *non-parametric classifier* does not make any assumtion on the shape of the probability distribution where each class comes from
- Examples:
  – (Convolutional feed-forward) neural networks
  – K-nearest neighbor classifier
  – Support Vector machines

Inspiration: Neuron cells



Feedforward Activation in Artificial Neural Networks: Perceptron (Rosenblatt 1957)

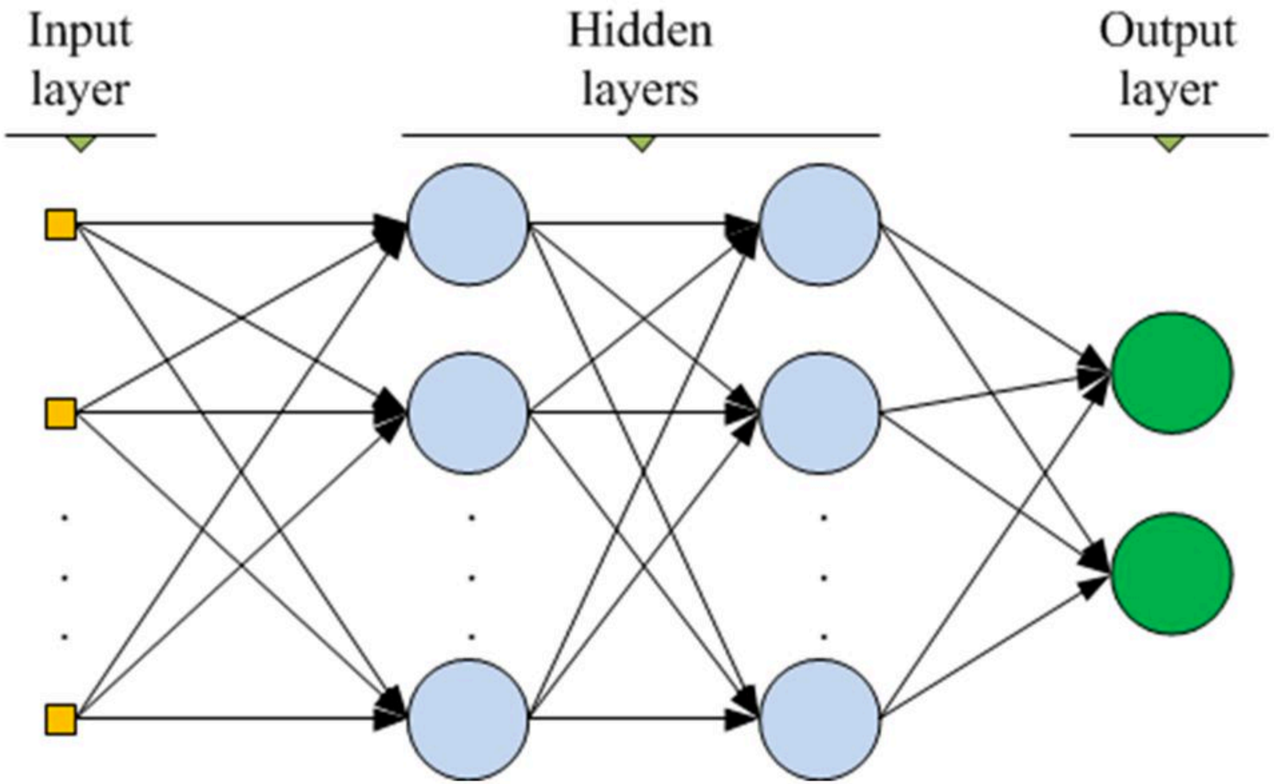


$$\frac{1}{1+e^{-(b_5+\sum_{i=1}^{4} w_{i\,5} x_i)}} = x_5$$

Feedforward Activation in Artificial Neural Networks: Perceptron (Rosenblatt 1957)

- "the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.”
- Minsky/ Papert (1969): cannot even learn XOR function!!
- BUT: Multilayer perceptron!

Feedforward Activation in Artificial Neural Networks: Multilayer Perceptron

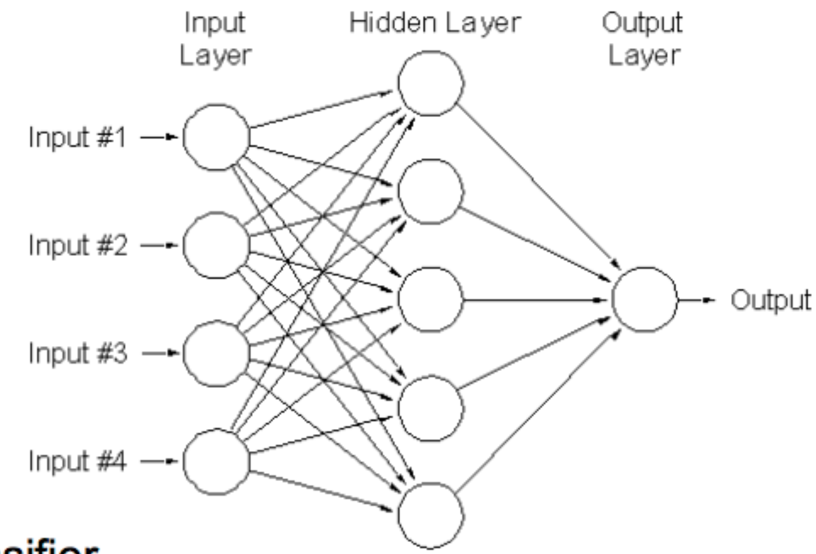


Exercise in Class

Input	Parameters	Output
$X_1=2$	$w_{14}=2$, $w_{46}=1$	$X_6=?$
$X_2=1$	$W_{24}=1$, $w_{56}=0$	
$X_3=0$	$b_4=1$, $b_6=0$	

$$\frac{1}{1+e^{-(b_j+\sum_{i=1}^{I} w_{ij}x_i)}} = x_j$$

Background: Multi-Layer Neural Networks



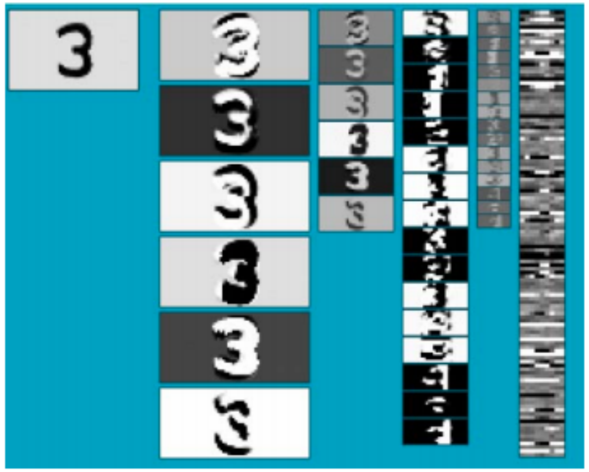
- Nonlinear classifier
- **Training:** find network weights **w** to minimize the error between true training labels y_i and estimated labels $f_{\mathbf{w}}(\mathbf{x}_i)$:

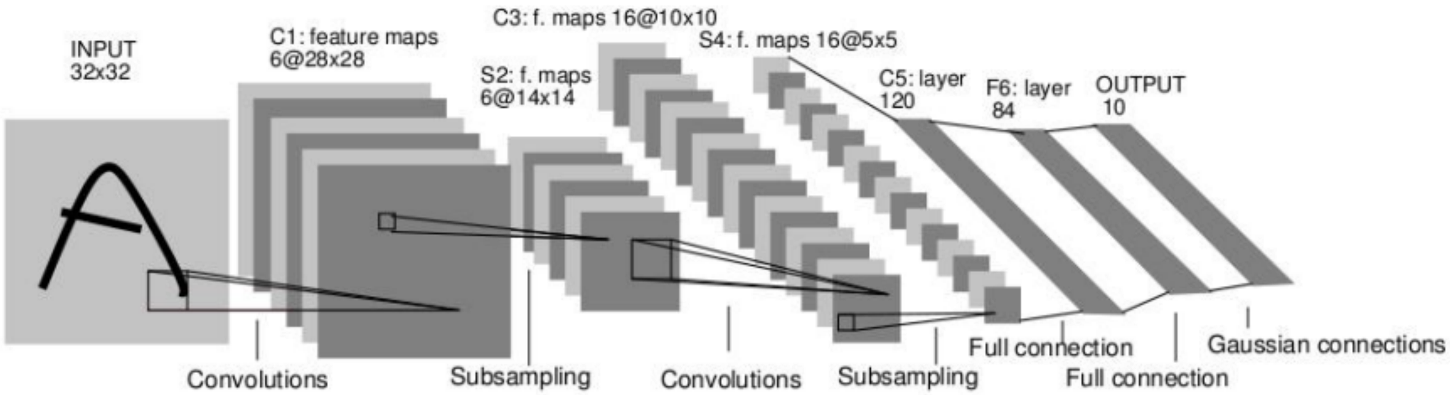
$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i)\right)^2$$

- Minimization can be done by gradient descent provided *f* is differentiable
  - This training method is called **back-propagation**

Convolutional Neural Networks (CNN, Convnet)

- Neural network with specialized connectivity structure
- Stack multiple stages of feature extractors
- Higher stages compute more global, more invariant features
- Classification layer at the end





Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86(11): 2278–2324, 1998.

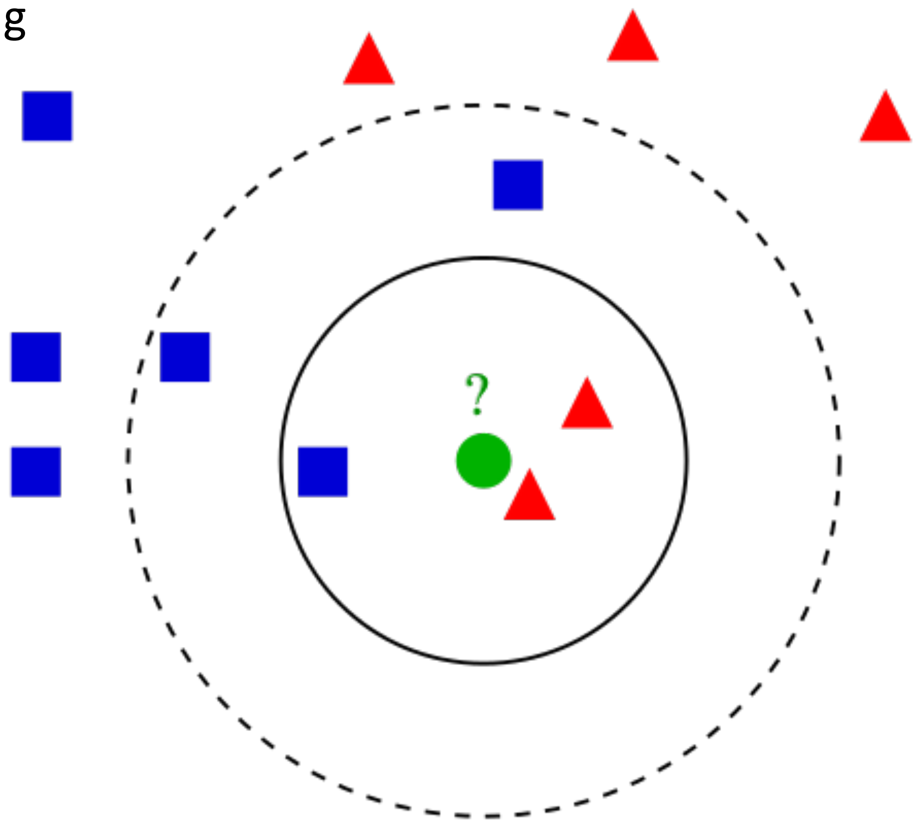
K-Nearest Neighbor Classification

- An object is classified by a majority vote of its neighbors
- Object assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small)
- Matlab: `w=knnc(z,k)`

Class Assignment

Classify the green point as a red triangle/blue square, according to k-nearest neighbor rule, use:

a) 1-nearest neighbors
b) 2-nearest neighbors
c) 5-nearest neighbors
d) 11-nearest neighbors



Support Vector Machines for Classification

- Classes that are not linearly separable are projected into higher dimensions, where they are linearly separable.
- Kernel: e.g. radial basis (Gauss) function with radius r

Complexity of the Classifier

- Complexity of classifier defined by the number of parameters to estimate
- Example:
  - NMC, LDA, QDA, RDA: Number of values in means, covariances
  - Neural networks: weights between layers
  - Support vector machines: weights